

## KONZORCIJ IJPP

### projekt

### »Uvedba integriranega javnega potniškega prometa v RS«

#### Delovni sklop 4: Integracija IS sistemov prevoznika

#### STANDARD IJPP



Ljubljana, junij 2023

Verzija: 07062023



## DS 4: Integracija IS sistemov prevoznika

Naziv projekta SLO/ANG:	Uvedba integriranega javnega potniškega prometa v RS		
faza izdelave	Integracija IS sistemov prevoznika		
NAROČNIK:	Ministrstvo za infrastrukturo, Langusova 4, 1535 Ljubljana		
Številka pogodbe:	2430-15-100042		
IZDELOVALEC:	KONZORCIJ IJPP:		
	<ul style="list-style-type: none"><li>- SŽ-Potniški promet, d.o.o., vodilni partner</li><li>- JAVNO PODJETJE LJUBLJANSKI POTNIŠKI PROMET, d.o.o., partner</li><li>- Javno podjetje za mestni potniški promet MARPROM d.o.o., partner</li><li>- Slovenske železnice, d.o.o., partner</li><li>- Prometni institut Ljubljana d.o.o., partner</li></ul>		
Sodelavci:	Imovation	<ul style="list-style-type: none"><li>- dr. Amor Chowdhury,</li><li>- dr. Janja Svečko,</li><li>- mag. Bor Prihavec,</li><li>- dr. Saša Klampfer,</li><li>- dr. Peter Podbreznik</li><li>- Peter Šamperl, dipl. inž. elektrotehnike,</li><li>- Franc Horvat, univ. dipl. inž. elektrotehnike,</li><li>- dr. Matjaž Fras,</li><li>- Gregor Jančič, univ. dipl. inženir računalništva in informatike,</li><li>- dr. Iztok Blazinšek,</li><li>- dr. Nikolaj Semenič,</li><li>- Ciril Sem, univ. dipl. gospodarski inženir,</li><li>- dr. Peter Rulič,</li><li>- Jernej Kroflič, univ. dipl. inž. elektrotehnike,</li><li>- dr. Matej Kseneman,</li><li>- Nenad Šikaleski, univ. dipl. inž. elektrotehnike,</li><li>- mag. Andrej Medved,</li><li>- Stanko Goličnik, univ. dipl. inž. elektrotehnike,</li><li>- Jože Gorjanc, univ. dipl. inž. elektrotehnike,</li><li>- dr. Dalibor Igrec</li></ul>	
		mag. Bojan Žlender	
Število izvodov:	1		
Izvod številka:	1/1		
Kraj in datum:	Ljubljana, maj 2021		

Vodja projekta:  
**mag. Boštjan Koren**

Vodilni partner:  
**SŽ-Potniški promet, d.o.o.**  
**mag. Boštjan Koren**

## DS 4: Integracija IS sistemov prevoznika

---

Vsebinski vodja projekta:

**dr. Primož KRANJEC**

Vodja delovnega skopa 4:

**dr. Janja SVEČKO**



# KAZALO

<b>KAZALO .....</b>	<b>1</b>
<b>KAZALO SLIK .....</b>	<b>17</b>
<b>KAZALO TABEL .....</b>	<b>19</b>
<b>1 UVOD IJPP .....</b>	<b>20</b>
1.1 SPLOŠNO .....	20
1.2 PREGLED PODROČJA STANDARDA .....	23
1.3 NAMEN STANDARDA .....	26
1.4 SLOVAR .....	28
1.5 OZNAKE .....	31
1.6 OMREŽJE INTEGRIRANEGA JAVNEGA POTNIŠKEGA PROMETA - IJPP .....	33
1.7 INFRASTRUKTURA OMREŽJA JAVNEGA POTNIŠKEGA PROMETA .....	34
1.7.1 Definicije osnovnih gradnikov poenostavljenega modela javnega potniškega prometa .....	35
1.7.2 Prenos poenostavljenega modela na infrastrukturo omrežja javnega potniškega prometa .....	37
1.7.2.1 Osnovne entitete infrastrukturnega omrežja .....	38
1.7.2.2 Relacije med entitetami infrastrukturnega omrežja .....	38
1.7.3 Tip transporta v javnem potniškem prometu .....	39
1.7.4 Prevozni sistem IJPP .....	40
1.7.5 Prestopne točke znotraj omrežja .....	41
1.7.6 Režimi vožnje vozil .....	42
1.8 PREVOZNIKI .....	43
1.9 UPRAVLJAVEC SISTEMA .....	44
1.10 INFORMACIJSKI SISTEM IJPP .....	47
1.10.1 Brezstična kartica .....	49
1.10.2 Terminalska oprema na vozilih .....	50
1.10.3 Terminalska oprema na prodajnih mestih .....	51
1.10.4 Zaledni sistem .....	53
1.10.5 Pridobivanje podatkov IJPP .....	54
1.10.6 Poročanje podatkov IJPP .....	56
1.11 REFERENCE .....	58
<b>2 SISTEM IJPP .....</b>	<b>59</b>
2.1 POTNIK .....	60
2.1.1 Postopki potnika v sistemu IJPP .....	60
2.1.1.1 Inicializacija IJPP kartice .....	60
2.1.1.2 Nakup IJPP vozovnic .....	61
2.1.1.3 Validacija potnika pri prevozniku .....	62
2.1.1.4 Prijava izgube kartice .....	62
2.1.1.5 Zahteva za duplikat IJPP kartice .....	62
2.1.1.6 Predaja potekle kartice .....	62
2.1.2 Življenjski cikel brezstične kartice znotraj sistema IJPP .....	63
2.2 PREVOZNIK .....	64
2.2.1 Postopki prevoznika .....	64
2.2.1.1 Izdaja kartice .....	65
2.2.1.2 Prodaja vozovnic .....	65

## DS 4: Integracija IS sistemov prevoznika

2.2.1.3	Izdaja duplikatov.....	66
2.2.1.4	Validacija potnikov na prevoznem sredstvu .....	66
2.2.1.5	Kontrola ustreznih vozovnic potnikov .....	66
2.2.1.6	Blokiranje kartice.....	67
2.2.1.7	Blokiranje vozovnice.....	67
2.2.1.8	Odvzem kartice.....	67
2.2.1.9	Vpis novih terminalov v sistem IJPP.....	68
2.2.1.10	Pridobitev osnovnih podatkov zalednega sistema (tarifa,...).....	68
2.2.1.11	Sporočanje podatkov zalednemu sistemu (batch) .....	68
2.2.2	Terminalska oprema za inicializacijo ali izdajo kartic.....	69
2.2.3	Terminalska oprema za zunanja prodajna mesta .....	69
2.2.4	Terminalska oprema za validiranje potnikov .....	71
2.2.5	Terminalska oprema za kontrolo potnikov.....	73
2.3	UPRAVLJAVEC .....	75
2.3.1	Administrativni del sistema IJPP.....	75
2.3.2	Zaledni sistem IJPP .....	77
2.4	POSTOPKI UPRAVLJAVCA .....	78
2.4.1	Upravljanje tarife .....	79
2.4.2	Upravljanje črne liste.....	79
2.4.3	Upravljanje izvršne liste.....	80
2.4.4	Upravljanje vozniških redov.....	80
2.4.5	Pravice za prestop .....	80
2.4.6	Seznam prodajnih mest .....	80
2.4.7	Seznam kartic .....	80
2.4.8	Inicializacija kartice .....	81
2.4.9	Nakup vozovnic .....	81
2.4.10	Validacija/aktivacija vozovnic.....	81
2.4.11	Izdaja dublikata.....	81
2.4.12	Blokada kartice/vozovnice .....	81
2.4.13	Deblokada kartice/vozovnice.....	82
2.5	KOMUNIKACIJA NA RELACIJI PREVOZNIK-UPRAVLJAVEC .....	82
2.6	RELACIJA MED ZALEDNIM SISTEMOM IN UPRAVLJAVCEM .....	83
<b>3</b>	<b>BREZSTIČNA PAMETNA KARTICA .....</b>	<b>84</b>
3.1.	BREZSTIČNA PAMETNA KARTICA MIFARE DESFIRE EV1 .....	84
3.1.1.	Lastnosti kartice .....	84
3.1.1.1.	Fizične lastnosti kartice.....	84
3.1.1.2.	Komponente na kartici.....	85
3.1.1.3.	RF vmesnik .....	85
3.1.1.4.	Pomnilnik .....	85
3.1.1.5.	Varnostni mehanizmi .....	86
3.1.2.	Grafična podoba kartic drugih prevoznikov .....	86
3.1.2.1.	Definirati način dodajanja manjkajočih elementov (dotisk, nalepke).....	86
3.1.2.2.	Definirati obvezne lastniške podatke.....	87
3.1.2.3.	Definirati obliko CUID .....	87
3.1.2.4.	Definirati pozicijo oznake IJPP na karticah drugih prevoznikov .....	88
3.2.	ČITALNIK BREZSTIČNE PAMETNE KARTICE.....	89
3.2.1.	Lastnosti čitalnika.....	90
3.3.	KOMUNIKACIJA Z UKAZI .....	90

## DS 4: Integracija IS sistemov prevoznika

3.3.1. Komunikacijski ukazi ISO/IEC 14443-3 .....	90
3.3.2. Komunikacijski ukazi ISO/IEC 14443-4 .....	91
3.3.2.1. Ukaz RATS in ATS .....	92
3.3.2.2. Ukaz PPS zahteva in PPS odgovor .....	92
3.3.3. Varnostni ukazi, ukazi za upravljanje s kartico, aplikacijami in PODATKI .....	92
3.4. POSTOPKI KOMUNIKACIJE MED ČITALNIKOM IN KARTICO .....	95
3.4.1. Protokol komunikacije pri prijavi .....	96
3.4.1.1. Stanje pripravljenosti .....	96
3.4.1.2. Stanje izberi aplikacijo .....	97
3.4.1.3. Stanje avtentikacija .....	97
3.4.1.4. Stanje branje verzije (datoteka INFO) .....	98
3.4.1.5. Stanje branje info (datoteka CARD INFO) .....	98
3.4.1.6. Stanje branje indeksov (datoteka DATA INDEX) .....	98
3.4.1.7. Stanje branje statusa (datoteka DATA FILE) .....	99
3.4.2. Protokol komunikacije med čitalnikom in kartico pri branju informacij o obstoječih vozovnicah na kartici .....	100
3.4.2.1. Stanje pripravljenosti .....	103
3.4.2.2. Stanje izbiri aplikacijo .....	103
3.4.2.3. Stanje avtentikacija .....	103
3.4.2.4. Stanje branje verzije (datoteka INFO) .....	104
3.4.2.5. Stanje branje info (datoteka CARD INFO) .....	104
3.4.2.6. Stanje branje indeksov (datoteka DATA INDEX) .....	105
3.4.2.7. Stanje branje statusa (datoteka DATA FILE) .....	105
3.4.2.8. Stanje branje kontrolnih podatkov (datoteka CHECK) .....	106
3.4.2.9. Stanje branje statusov vozovnic (datoteka PRODUCTS) .....	106
3.4.2.10. Stanje branje vozovnic (datoteka PRODUCTS) .....	106
3.4.2.11. Stanje branje statusov relacij (datoteka LINE TARIFF DATA) .....	107
3.4.2.12. Stanje branje relacij (datoteka LINE TARIFF DATA) .....	107
3.4.2.13. Stanje branje zgodovine validacij (datoteka HISTORY-VALID) .....	107
3.4.3. Protokol komunikacije med čitalnikom in kartico pri NAKUPU vozovnice z gotovino .....	108
3.4.3.1. Stanje branje zgodovine validacij (datoteka HISTORY-VALID) .....	108
3.4.3.2. Stanje zapis statusa (datoteka PRODUCTS) .....	108
3.4.3.3. Stanje zapis podatkov vozovnice (datoteka PRODUCTS) .....	108
3.4.3.4. Stanje zapis naključnega števila (datoteka CHECK) .....	109
3.4.3.5. Stanje zapis časa zadnje transakcije (datoteka CHECK) .....	109
3.4.3.6. Stanje potrdi zapis podatkov .....	109
3.4.4. Protokol komunikacije med čitalnikom in kartico pri VALIDACIJI / AKTIVACIJI obstoječe vozovnice .....	110
3.4.4.1. Stanje branje zgodovine validacij (datoteka HISTORY-VALID) .....	111
3.4.4.2. Stanje zapis statusov vozovnic (datoteka PRODUCTS) .....	112
3.4.4.3. Stanje zapis vozovnic (datoteka PRODUCTS) .....	112
3.4.4.4. Stanje zapis validacije v zgodovino validacij (datoteka HISTORY-VALID) .....	112
3.4.4.5. Stanje zapis naključnega števila (datoteka CHECK) .....	113
3.4.4.6. Stanje zapis časa zadnje transakcije (datoteka CHECK) .....	113
3.4.4.7. Stanje potrdi zapis podatkov .....	113
3.4.5. Protokol komunikacije med čitalnikom in kartico pri preverjanju vozovnice .....	116
3.4.6. Protokol komunikacije med čitalnikom in kartico pri STORNACIJI vozovnice .....	118
3.4.6.1. Stanje branje zgodovine validacij (datoteka HISTORY-VALID) .....	118
3.4.6.2. Stanje zapis statusa (datoteka PRODUCTS) .....	118
3.4.6.3. Stanje zapis naključnega števila (datoteka CHECK) .....	119
3.4.6.4. Stanje zapis časa zadnje transakcije (datoteka CHECK) .....	119
3.4.6.5. Stanje potrdi zapis podatkov .....	119

## DS 4: Integracija IS sistemov prevoznika

<b>3.4.7. Protokol komunikacije med čitalnikom in kartico ob POPOLNEM VRAČILU - IZPLAČILU vozovnice</b>	<b>121</b>
3.4.7.1. Stanje pripravljenosti	122
3.4.7.2. Stanje izberi aplikacijo	123
3.4.7.3. Stanje avtentikacija	123
3.4.7.4. Stanje branje časa zadnje transakcije in naključnega števila	123
3.4.7.5. Stanje zapis statusa	123
3.4.7.6. Stanje zapis naključnega števila	124
3.4.7.7. Stanje zapis časa zadnje transakcije	124
3.4.7.8. Stanje potrdi zapis podatkov	124
<b>3.4.8. Protokol komunikacije med čitalnikom in kartico ob DELNEM VRAČILU – DELNEM IZPLAČILU vozovnice</b>	<b>124</b>
<b>3.4.9. Protokol komunikacije med čitalnikom in kartico ob BLOKADI aplikacije IJPP</b>	<b>125</b>
3.4.9.1. Stanje pripravljenosti	125
3.4.9.2. Stanje izbiri aplikacijo	125
3.4.9.3. Stanje avtentikacija	126
3.4.9.4. Stanje branje verzije (datoteka INFO)	126
3.4.9.5. Stanje branje info (datoteka CARD INFO)	127
3.4.9.6. Stanje potrdi zapis podatkov	127
<b>3.4.10. Protokol komunikacije med čitalnikom in kartico ob BLOKADI vozovnice</b>	<b>128</b>
3.4.10.1. Stanje pripravljenosti	131
3.4.10.2. Stanje izberi aplikacijo	131
3.4.10.3. Stanje avtentikacija	131
3.4.10.4. Stanje branje časa zadnje transakcije in naključnega števila	131
3.4.10.5. Stanje zapis statusa	132
3.4.10.6. Stanje zapis naključnega števila	132
3.4.10.7. Stanje zapis časa zadnje transakcije	132
3.4.10.8. Stanje potrdi zapis podatkov	132
<b>3.4.11. Protokol komunikacije med čitalnikom in kartico ob DEBLOKADI aplikacije IJPP</b>	<b>132</b>
3.4.11.1. Stanje pripravljenosti	133
3.4.11.2. Stanje izberi aplikacijo	134
3.4.11.3. Stanje avtentikacija	134
3.4.11.4. Stanje branje verzije kartice	134
3.4.11.5. Stanje zapis info	135
3.4.11.6. Stanje potrdi zapis podatkov	135
<b>3.4.12. Protokol komunikacije med čitalnikom in kartico ob DEBLOKADI vozovnice</b>	<b>135</b>
<b>3.4.13. Protokol komunikacije med čitalnikom in kartico ob preverjanju uspešnosti zapisa vozovnice</b>	<b>136</b>
3.4.13.1. Stanje pripravljenosti	137
3.4.13.2. Stanje izbiri aplikacijo	138
3.4.13.3. Stanje avtentikacija	138
3.4.13.4. Stanje branje kontrolnih podatkov (datoteka CHECK)	138
3.4.13.5. Stanje branje statusov vozovnic (datoteka PRODUCTS)	139
3.4.13.6. Stanje branje vozovnic (datoteka PRODUCTS)	139
3.4.13.7. Stanje branje statusov relacij (datoteka LINE TARIFF DATA)	140
3.4.13.8. Stanje branje relacij (datoteka LINE TARIFF DATA)	140
3.4.13.9. Stanje branje zgodovine validacij (datoteka HISTORY-VALID)	140
3.4.13.10. Stanje zapis statusov vozovnic (datoteka PRODUCTS)	140
3.4.13.11. Stanje potrdi zapis podatkov	141
<b>3.5. ČASOVNE OMEJITVE PRI POSTOPKIH KOMUNIKACIJE MED ČITALNIKOM IN KARTICO</b>	<b>142</b>
<b>3.6. PODATKOVNA STRUKTURA</b>	<b>143</b>
3.6.1. Pozicioniranje aplikacije IJPP	143
3.6.2. Struktura posameznih datotek	143

## DS 4: Integracija IS sistemov prevoznika

3.6.3. Opis posameznih datotek .....	149
3.6.3.1. FID 0x00 (INFO) .....	149
3.6.3.2. FID 0x01 (CARD INFO) .....	150
3.6.3.3. FID 0x02 (DATA INDEX) .....	150
3.6.3.4. FID 0x03 (DATA FILE) .....	150
3.6.3.5. FID 0x07 (CHECK) .....	151
3.6.3.6. FID 0x08 (PRODUCTS) .....	151
3.6.3.7. FID 0x09 (HISTORY-VALID) .....	152
3.6.3.8. FID 0x0A (RELATIONS) .....	153
3.7. VARNOSTNA SPECIFIKACIJA .....	154
3.7.1. Opisi varnostnih / kriptografskih protokolov .....	154
3.7.1.1. Izbor kriptografskih algoritmov .....	154
3.7.1.2. Diverzifikacija ključev .....	154
3.7.1.3. Avtentikacija .....	155
3.7.2. Uporaba varnostnega elementa pri delu s IJPP kartico (oz. aplikacijo na kartici) .....	156
3.7.2.1. Opis SAM modula .....	158
3.7.3. Specifikacija varnostnega elementa .....	162
3.7.3.1. Ključi na varnostnem elementu .....	162
3.7.4. Življenjski cikel varnostnega elementa .....	163
3.7.4.1. Dobava .....	163
3.7.4.2. Personalizacija .....	163
3.7.4.3. Distribucija .....	163
3.7.4.4. Namestitve in aktivacija .....	163
3.7.4.5. Delovanje .....	164
3.7.4.6. Deaktivacija / blokada .....	165
3.7.4.7. Uničenje .....	165
3.8. PROTOKOL DELA .....	165
3.8.1. Podatki, ki se procesirajo pri branju INFO kartice .....	165
3.8.1.1. Podatki, ki se preverjajo na kartici ob branju INFO kartice .....	165
3.8.2. Podatki, ki se procesirajo pri NAKUPU vozovnice .....	167
3.8.2.1. Nakup enkratne vozovnice .....	171
3.8.2.2. Nakup enosmerne vozovnice .....	172
3.8.2.3. Prodajna mesta IJPP .....	172
3.8.2.4. Inicializacija kartice .....	172
3.8.2.5. Postopek za zapis datuma .....	173
3.8.2.6. Podatki na kartici, ki so potrebni za nakup vozovnic .....	174
3.8.3. Podatki, ki se procesirajo pri VALIDACIJI/AKTIVACIJI vozovnice .....	175
3.8.3.1. Podatki, ki se preverjajo na kartici ob validaciji/aktivaciji .....	175
3.8.3.2. Podatki, ki se zapišejo ob validaciji .....	176
3.8.3.3. Podatki, ki se zapišejo ob aktivaciji .....	177
3.8.3.4. Podatki, ki se pošljejo proti centru ob validaciji/aktivaciji .....	178
3.8.4. Podatki, ki se procesirajo pri STORNACIJI vozovnice .....	178
3.8.5. Podatki, ki se procesirajo pri BLOKADI aplikacije/vozovnice .....	179
3.8.6. Podatki, ki se procesirajo pri DEBLOKADI aplikacije/vozovnice .....	179
3.9. LISTNA VOZOVNICA KOT ALTERNATIVA BREZSTIČNI PAMETNI KARTICI .....	179
3.9.1. Opis barkode .....	180
3.9.2. Generiranje hash kode in kriptiranega podpisa .....	180
3.9.2.1. SHA 224 .....	181
3.9.2.2. DSA 2048 .....	181
3.9.3. Ključi DSA .....	181

## DS 4: Integracija IS sistemov prevoznika

3.9.4. AZTEC barkoda .....	181
3.10. REFERENCE .....	182
<b>4 ZGRADBA TARIFE .....</b>	<b>183</b>
4.1 KONCEPTUALNI OPIS TARIFE .....	183
4.2 CENOVNI MODEL TARIFE .....	185
4.2.1 Osnovni podatki tarife – struktura <i>TariffLocationSubLocation</i> .....	185
4.2.2 Osnovni podatki tarife – struktura <i>ServiceProviderTariff</i> .....	186
4.2.3 Vrste vozovnic – struktura <i>TariffProduct</i> .....	186
4.2.4 Statusi – struktura <i>TariffStatus</i> .....	196
4.2.5 Tarifni razredi – struktura <i>TariffClass</i> .....	197
4.3 PODATKOVNA STRUKTURA TARIF NA TERMINALIH .....	197
4.4 DISTRIBUCIJA TARIF .....	200
4.5 VEČRAZREDNE MESTNE TARIFE .....	202
4.6 PRAVILA ZA TARIFE .....	202
4.7 UPRAVLJANJE S TARIFAMI - ADMINISTRACIJA .....	203
4.7.1 Kreiranje nove tarife .....	203
4.7.2 Kreiranje pravil .....	204
4.7.3 Pomembni napotki pri upravljanju s tarifami .....	205
<b>5 ZGRADBA IZVRŠNE LISTE .....</b>	<b>207</b>
5.1 KONCEPTUALNI OPIS IZVRŠNE LISTE .....	207
5.2 DELOVANJE IZVRŠNE LISTE .....	208
5.2.1 Offline način delovanja izvršne liste .....	209
5.2.2 Online način delovanja izvršne liste .....	210
5.3 UPRAVLJANJE Z IZVRŠNO LISTO - ADMINISTRACIJA .....	211
5.4 PODATKOVNI MODEL IZVRŠNE LISTE .....	211
5.1 INKREMENTALNI PRENOS IZVRŠNE LISTE .....	212
<b>6 INTEGRACIJA Z ZALEDNIM SISTEMOM .....</b>	<b>213</b>
6.1 PODPRTE FUNKCIONALNOSTI .....	213
6.1.1 Identifikacija stanja transakcijskih podatkov v zalednem sistemu .....	215
6.1.2 Časovni intervali ažuriranja .....	215
6.1.3 Distribucija črne liste .....	215
6.1.4 Ugotavljanje statusa kartice .....	216
6.1.5 Nastavitev statusa kartice .....	216
6.1.6 Distribucija izvršne liste .....	216
6.1.7 Distribucija tarif .....	216
6.1.8 Distribucija podatkov o prehodih med področji .....	216
6.1.9 Prenos uspešno izvedenih transakcij za procesiranje na zaledni sistem .....	216
6.1.9.1 Prodaja vozovnice .....	216
6.1.9.2 Validacija vozovnice .....	216
6.1.9.3 Stornacija nakupa vozovnice .....	217
6.1.9.4 Permanentna blokada kartice .....	217
6.1.9.5 Online prenos vozovnice na kartico .....	217
6.1.9.6 Online brisanje vozovnice na kartici .....	217
6.1.9.7 Online nastavitev novega statusa na kartici .....	217
6.1.9.8 Zavrnitev transakcije .....	217
6.1.9.9 Prijava vožnje .....	217

## DS 4: Integracija IS sistemov prevoznika

6.1.9.10	Prijava blagajnika .....	217
6.1.9.11	Odjava blagajnika .....	217
6.1.9.12	Aktivacija vozovnice .....	217
6.1.9.13	Inicializacija kartice .....	218
6.1.9.14	Duplikat kartice .....	218
6.1.9.15	Kontrolorjeva transakcija .....	218
6.1.9.16	Prijava Kontrolorja .....	218
6.1.9.17	Dodajanje medkrajevne relacije na kartico .....	218
6.1.9.18	Brisanje medkrajevne relacije na kartici .....	218
6.1.9.19	Vračilo vozovnice .....	218
6.1.9.20	Zamenjava vozovnice - stornacija .....	218
6.1.9.21	Ročna validacija vozovnice .....	218
6.1.9.22	Blokiranje vozovnice .....	218
6.1.9.23	Odblokiranje vozovnice .....	218
6.1.9.24	Neuspelo branje kartice .....	218
6.1.9.25	Zamenjava vozovnice - nakup .....	219
6.1.9.26	Posodobitev relacije na izvršni listi .....	219
6.1.9.27	Transakcija za razhroščevanje .....	219
<b>7</b>	<b>METODE SPLETNE STORITVE IJPP - TRANSAKCIJSKI SISTEM .....</b>	<b>221</b>
7.1	IDENTIFYTRANSACTIONPARAMETERS .....	222
7.2	GETBLACKLIST .....	222
7.3	GETCARDSTATUS .....	224
7.4	SETNEWCARDSTATUS .....	226
7.5	GETEXECUTABLELIST .....	227
7.6	GETTARIFF .....	228
7.7	GETTARIFFCLASSTRANSITIONMATRIX .....	230
7.8	PROCESSTRANSACTIONS .....	231
7.9	GETIJPPDATE TIME .....	232
7.10	GETIJPPDATE TIME_PLADENJ .....	233
7.11	DEFINICIJE PODATKOVNIH STRUKTUR .....	233
7.12	TIPI IJPP TRANSAKCIJ .....	249
7.12.1	Nakup vozovnice .....	250
7.12.2	Validacija vozovnice .....	252
7.12.3	Stornacija nakupa vozovnice .....	255
7.12.4	Blokirana kartica .....	257
7.12.5	Prenos vozovnice na kartico iz izvršne liste .....	260
7.12.6	Brisanje vozovnice na kartici iz izvršne liste .....	262
7.12.7	Nastavitev novega statusa na kartici iz izvršne liste .....	264
7.12.8	Zavrnitev transakcije .....	267
7.12.9	Prijava vožnje .....	271
7.12.10	Prijava blagajnika .....	273
7.12.11	Odjava blagajnika .....	275
7.12.12	Aktivacija vozovnice .....	277
7.12.13	Inicializacija kartice .....	280
7.12.14	Duplikat kartice .....	282
7.12.15	Kontrolorjeva transakcija .....	284
7.12.16	Prijava kontrolorja .....	287
7.12.17	Dodajanje relacije na kartico .....	289



## DS 4: Integracija IS sistemov prevoznika

7.12.18	Brisanje relacije iz kartice.....	291
7.12.19	Vračilo vozovnice.....	294
7.12.20	Zamenjava vozovnice – stornacija .....	296
7.12.21	Ročna validacija vozovnice.....	299
7.12.22	Blokiranje vozovnice na kartici iz izvršne liste .....	301
7.12.23	Odblokiranje vozovnice na kartici iz izvršne liste .....	304
7.12.24	Neuspelo branje kartice .....	306
7.12.25	Zamenjava vozovnice – nakup .....	308
7.12.26	Posodobitev relacije na kartici iz izvršne liste .....	311
7.12.27	Transakcija za razhroščevanje .....	313
7.12.31	Pogojna validacija vozovnice .....	315
7.12.36	Posodobitev vloge na kartici iz izvršne liste .....	318
7.12.37	Posodobitev vloge na kartici .....	320
7.12.38	Vračilo neprevzete vozovnice.....	322
7.12.39	Prenos koordinat iz postajne točke .....	325
7.12.40	Reklamacija.....	327
7.12.41	Vračilo dobropisa .....	330
7.12.42	Koriščenje dobropisa .....	332
7.12.43	Pripis dobropisa .....	334
7.12.47	Podaljšanje subvencioniranega statusa na kartici.....	337
<b>8</b>	<b>METODE SPLETNE STORITVE IJPP - ADMINISTRACIJA .....</b>	<b>339</b>
8.1	SETPASSENGER .....	340
8.2	GETPASSENGER.....	341
8.3	SETCARD .....	342
8.4	UPDATECARD .....	343
8.5	GETCARD.....	344
8.6	UPDATECARDPASSENGERSTATUS .....	345
8.7	GETCARDPASSENGERSTATUS.....	346
8.8	GETRELATION .....	347
8.9	GETCARDPRODUCTS .....	348
8.10	GETPRODUCTPRICE .....	349
8.11	SETPOS .....	350
8.12	GETPOS.....	351
8.13	SETTERMINAL .....	352
8.14	GETTERMINAL .....	353
8.15	SETBLACKLIST .....	354
8.16	SETEXECUTABLELIST.....	355
8.17	DELEXECUTABLELIST .....	356
8.18	REGISTRATIONTERMINAL .....	357
8.19	ACTIVATIONSAM .....	358
8.20	GETDUPLICATEDATA.....	359
8.21	GETTRANSACTIONS.....	360
8.22	GETSUBTRANSACTIONSBSUBSIDY .....	362
8.23	GETCUIDSBYIDENTITYNUMBER.....	363
8.24	GETIJPPCATALOGS.....	364
8.25	GETIJPPCATALOGS_PLADENJ .....	365
8.26	DEFINICIJE PODATKOVNIH STRUKTUR .....	366



## DS 4: Integracija IS sistemov prevoznika

<b>9</b>	<b>POSTOPKI PRI UPORABI SPLETNIH STORITEV IJPP.....</b>	<b>379</b>
9.1	REGISTRACIJA TERMINALA .....	379
9.2	SPREMEMBA PODATKOV O TERMINALU .....	380
9.3	AKTIVACIJA SAM .....	380
9.4	REGISTRACIJA PRODAJNEGA MESTA .....	380
9.5	SPREMEMBA PODATKOV O PRODAJNEM MESTU .....	380
9.6	REGISTRACIJA VALIDATORJA .....	381
9.7	REGISTRACIJA UPORABNIKA .....	381
9.8	REGISTRACIJA NEPERSONIFICIRANE KARTICE .....	381
9.9	REGISTRACIJA PERSONIFICIRANE KARTICE .....	381
9.10	NASTAVITEV PRENOSA VOZOVNICE NA KARTICO Z IZVRŠNO LISTO .....	382
9.11	NASTAVITEV BRISANJA VOZOVNICE NA KARTICI Z IZVRŠNO LISTO .....	382
9.12	NASTAVITEV NOVEGA STATUSA UPORABNIKA Z IZVRŠNO LISTO .....	382
9.13	NASTAVITEV ZAČASNO ONEMOGOČENE KARTICE Z IZVRŠNO LISTO .....	383
9.14	NASTAVITEV ZAČASNO ONEMOGOČENE VOZOVNICE Z IZVRŠNO LISTO .....	383
9.15	ODSTRANITEV NEIZVRŠENEGA OPRAVILA IZ IZVRŠNE LISTE.....	383
<b>10</b>	<b>POSTOPKI ZA VERIFIKACIJO STROJNE IN PROGRAMSKE OPREME TER BREZSTIČNE KARTICE.....</b>	<b>385</b>
10.1	SPECIFIKACIJA PROCEDURE ZA PRISTOP K VERIFIKACIJSKIM POSTOPKOM.....	385
10.1.1	<i>Pridobitev dokumentacije standarda IJPP.....</i>	<i>386</i>
10.1.2	<i>Prilagoditev strojne in programske opreme standardu IJPP.....</i>	<i>386</i>
10.1.3	<i>Spletni obrazec za priglasitev opravljanja verifikacijskih postopkov v IJPP laboratoriju.....</i>	<i>386</i>
10.1.4	<i>Odgovor laboratorija IJPP.....</i>	<i>386</i>
10.1.5	<i>Potrditev termina izvajanja testov v laboratoriju IJPP s strani odgovorne (registrirane) osebe tehnološkega ponudnika.....</i>	<i>387</i>
10.1.6	<i>Postavitev opreme tehnološkega ponudnika v okolje laboratorija.....</i>	<i>387</i>
10.1.7	<i>Izvedba specifikiranih testov za priglaseeno opremo tehnološkega ponudnika.....</i>	<i>387</i>
10.1.8	<i>Pogoji, kdaj mora periferija tehnološkega ponudnika opraviti test.....</i>	<i>388</i>
10.2	DEFINICIJA IN OSNOVNI POJMI UPORABLJENI V POSTOPKU VERIFICIRANJA .....	388
10.3	PROCES TESTIRANJA IN NJEGOVI SKLOPI .....	389
10.4	SPLOŠNI TESTNI NAČRT.....	390
10.5	ZASNOVA TESTNIH SCENARIJEV PO SKLOPIH TESTIRANIH SEGMENTOV.....	391
10.5.1	<i>Test poenotenega delovanja validirnih enot različnih ponudnikov.....</i>	<i>391</i>
10.5.1.1	<i>Test indikacije nepripravljenosti validirne enote.....</i>	<i>392</i>
10.5.1.2	<i>Test indikacije pri vklopu validirne enote.....</i>	<i>392</i>
10.5.1.3	<i>Test indikacije pripravljenosti validirne enote na sprejem kartice .....</i>	<i>393</i>
10.5.1.4	<i>Test indikacije procedure branja kartice na validirni enoti.....</i>	<i>393</i>
10.5.1.5	<i>Test indikacije procedure neuspešnega branja kartice na validirni enoti.....</i>	<i>394</i>
10.5.1.6	<i>Test avdio indikacije na validirni enoti .....</i>	<i>395</i>
10.5.2	<i>Pravila in procedure za validacijo terminalske opreme .....</i>	<i>396</i>
10.5.2.1	<i>Fizična namestitev varnostnega elementa .....</i>	<i>396</i>
10.5.2.2	<i>Logični test preizkusa protokola T=1 .....</i>	<i>397</i>
10.5.2.3	<i>Validacija avtentikacijskega postopka brezstične kartice .....</i>	<i>398</i>
10.5.3	<i>Test NFC modula .....</i>	<i>398</i>
10.5.3.1	<i>Test prisotnosti NFC polja naprave z NFC modulom .....</i>	<i>399</i>
10.5.3.2	<i>Logični test preizkusa protokola NFC.....</i>	<i>399</i>
10.5.3.3	<i>Test razdalje (dometa) delovanja kartice NFC .....</i>	<i>400</i>
10.5.3.4	<i>Test prisotnosti dveh kartic NFC.....</i>	<i>401</i>

## DS 4: Integracija IS sistemov prevoznika

10.6	TESTI BREZSTIČNE IJPP KARTICE .....	402
10.6.1	Testi znotraj datoteke INFO .....	402
10.6.1.1	Test zapisane verzije podatkovne datoteke INFO (VERSION_INFO) .....	402
10.6.1.2	Test zapisane verzije podatkovne datoteke CARD INFO (VERSION_CARD_INFO) .....	403
10.6.1.3	Test zapisane verzije podatkovne datoteke DATA INDEX (VERSION_DATA_INDEX) .....	404
10.6.1.4	Test zapisane verzije podatkovne datoteke DATA FILE (VERSION_DATA_FILE) .....	405
10.6.1.5	Test zapisane verzije podatkovne datoteke CHECK (VERSION_CHECK) .....	406
10.6.1.6	Test zapisane verzije podatkovne datoteke PRODUCTS (VERSION_PRODUCTS) .....	406
10.6.1.7	Test zapisane verzije podatkovne datoteke HISTORY (VERSION_HISTORY_VALIDATION) .....	407
10.6.1.8	Test zapisane verzije podatkovne datoteke RELATIONS (VERSION_RELATIONS) .....	408
10.6.2	Testi znotraj datoteke CARD INFO .....	409
10.6.2.1	Test zapisanega statusa kartice v podatkovni datoteki CARD INFO (STATUS) .....	409
10.6.2.2	Test zapisanega datuma pričetka veljave kartice v podatkovni datoteki CARD INFO (DATUM PREČETKA VELJAVE KARTICE) .....	411
10.6.2.3	Test zapisanega datuma konca veljave kartice v podatkovni datoteki CARD INFO (DATUM KONCA VELJAVE KARTICE) .....	412
10.6.2.4	Test zapisanega tipa personifikacije v podatkovni datoteki CARD INFO (TIP PERSONIFIKACIJE) brezstične IJPP kartice 412	
10.6.2.5	Test zapisanega posebnega statusa v podatkovni datoteki CARD INFO (POSEBNI STATUS) brezstične IJPP kartice 414	
10.6.2.6	Test zapisanih osebnih podatkov v podatkovno datoteko CARD INFO (IME, PRIIMEK, NASLOV) brezstične IJPP kartice .....	415
10.6.2.7	Test zapisane EMŠO številke v podatkovno datoteko CARD INFO brezstične IJPP kartice .....	416
10.6.2.8	Test zapisane DAVČNE številke v podatkovno datoteko CARD INFO brezstične IJPP kartice .....	417
10.6.2.9	Test zapisanega DATUMA INICIALIZACIJE kartice v podatkovno datoteko CARD INFO brezstične IJPP kartice 418	
10.6.2.10	Test zapisanega datuma personifikacije kartice v podatkovno datoteko CARD INFO brezstične IJPP kartice 419	
10.6.3	Testi znotraj datoteke DATA INDEX .....	420
10.6.3.1	Test zapisanega števila ponudnikov storitev (SP_ID) v podatkovni datoteki DATA INDEX .....	420
10.6.3.2	Test zapisanega formata podatkovne datoteke ponudnika v datoteki DATA INDEX .....	421
10.6.3.3	Test zapisane informacije o zamiku (OFFSET) podatkov ponudnika storitev v datoteki DATA INDEX ..	422
10.6.3.4	Test zapisane informacije o dolžini (LENGTH) podatkov ponudnika storitev v datoteki DATA INDEX ..	423
10.6.4	Testi znotraj datoteke DATA FILE .....	424
10.6.4.1	Test zapisanih podatkov ponudnika v podatkovni datoteki DATA FILE .....	424
10.6.5	Testi znotraj datoteke CHECK – kontrola stanja .....	426
10.6.5.1	Test zapisanih podatkov v CHECK datoteki .....	427
10.6.6	Test datoteke PRODUCTS .....	428
10.6.6.1	Test zapisanih podatkov v PRODUCTS datoteki .....	428
10.6.7	Test datoteke HISTORY VALID .....	429
10.6.7.1	Test zapisanih podatkov v HISTORY-VALID datoteki .....	429
10.6.8	Test datoteke RELATIONS .....	431
10.6.8.1	Test zapisanih podatkov v RELATIONS datoteki .....	431
10.6.9	Test vozovnice .....	432
10.6.9.1	Test uspešne prodaje vozovnice za relacijo A-B .....	432
10.6.9.2	Test neuspešne prodaje vozovnice za relacijo A-B .....	433
10.6.9.3	Test prodaje vozovnice za relacijo A-B s kartico, ki je na črni listi .....	434
10.6.10	Test validatorja .....	435
10.6.10.1	Test ujemanja diverzificiranih ključev na kartici s ključi validirne enote .....	435
10.6.10.2	Test preverjanja seje v proceduri pisanja in spremembe podatkov na kartici s strani validirne enote 436	

## DS 4: Integracija IS sistemov prevoznika

10.6.10.3	Test odziva validatorja v primeru manjkajočega podatka na brezstični kartici.....	437
10.6.10.4	Test zaklenitve validatorja .....	438
10.6.10.5	Test zavrnitve koriščenja produktov v režimu popolne zaklenitve validatorja .....	439
10.6.10.6	Test koriščenja specifičnih produktov v offline režimu delovanja validatorja .....	440
10.6.10.7	Test pošiljanja in upravljanja batchev pri offline koriščenju produktov .....	441
10.6.10.8	Test uspešne validacije vozovnice .....	442
10.6.10.9	Test validacije vozovnice, ki je na črni list .....	444
10.6.10.10	Test validacije vozovnice s kartico, ki je na črni listi.....	445
10.6.10.11	Test validacije vozovnice zunaj časa njene veljavnosti .....	446
10.6.10.12	Test validacije vozovnice po preteku maksimalnega časa prestopanja .....	447
10.6.10.13	Test validacije vozovnice na neustrezni postaji, ki ne pripada relaciji A-B.....	448
10.6.11	<b>Testi verifikacij vozovnice na kartomatu oz. prodajnem mestu.....</b>	<b>450</b>
10.6.11.1	Test uspešne verifikacije kartice in zapisanih podatkov .....	450
10.6.11.2	Test verifikacije izven časa veljavnosti enkratne vozovnice.....	451
10.6.11.3	Test verifikacije po uspešni validaciji in znotraj dovoljenega časa prestopanja .....	452
10.6.11.4	Test verifikacije po uspešni validaciji in izven dovoljenega časa prestopanja .....	453
10.6.11.5	Test verifikacije vozovnice s kartico, ki je na črni listi .....	454
10.6.11.6	Test verifikacije vozovnice, ki se nahaja na črni listi .....	456
10.6.12	<b>Testi nadzorov vozovnic z mobilnim validatorjem (kontrolor) .....</b>	<b>457</b>
10.6.12.1	Test kontrole z mobilnim validatorjem vozovnice, ki je bila uspešno validirana .....	457
10.6.12.2	Test kontrole z mobilnim validatorjem vozovnice, ki ni bila validirana .....	458
10.6.12.3	Test kontrole vozovnice z mobilnim validatorjem znotraj območja veljavnosti .....	459
10.6.12.4	Test kontrole vozovnice z mobilnim validatorjem izven območja veljavnosti .....	461
10.6.12.5	Test istovetnosti imetnika IJPP kartice .....	462
10.6.13	<b>Testi obnove vozovnic .....</b>	<b>463</b>
10.6.13.1	Test obnove neuporabljene enkratne vozovnice.....	463
10.6.13.2	Test preverjanja novega datuma veljavnosti po uspešni obnovi .....	464
10.6.13.3	Test obnove uporabljene enkratne vozovnice.....	465
10.6.14	<b>Testi refundacije (povračila) zneska vozovnice .....</b>	<b>466</b>
10.6.14.1	Test obnove uporabljene enkratne vozovnice.....	467
10.6.14.2	Test obnove neuporabljene enkratne vozovnice.....	468
10.6.14.3	Test blokade enkratne vozovnice v sistemu .....	469
10.6.14.4	Test deblokade enkratne vozovnice v sistemu .....	470
10.6.14.5	Test odstranitve enkratne vozovnice iz kartice.....	471
10.7	<b>TESTI METOD SPLETNE STORITVE IJPP .....</b>	<b>473</b>
10.7.1	<b>Testi metode IdentifyTransactionParameters.....</b>	<b>473</b>
10.7.2	<b>Testi metode GetBlackList.....</b>	<b>474</b>
10.7.3	<b>Testi metode GetCardStatus .....</b>	<b>475</b>
10.7.4	<b>Testi metode SetNewCardStatus.....</b>	<b>476</b>
10.7.5	<b>Testi metode GetExecutableList.....</b>	<b>477</b>
10.7.6	<b>Testi metode GetTariff.....</b>	<b>478</b>
10.7.7	<b>Testi metode GetTariffClassTransitionMatrix .....</b>	<b>479</b>
10.7.8	<b>Testi metode ProcessTransactions .....</b>	<b>480</b>
10.8	<b>VERIFIKACIJA METOD SPLETNE STORITVE IJPP - ADMINISTRACIJA .....</b>	<b>481</b>
10.8.1	<b>Testi metode SetPassenger .....</b>	<b>481</b>
10.8.2	<b>Testi metode GetPassenger .....</b>	<b>482</b>
10.8.3	<b>Testi metode SetCard.....</b>	<b>483</b>
10.8.4	<b>Testi metode UpdateCard .....</b>	<b>484</b>
10.8.5	<b>Testi metode GetCard .....</b>	<b>485</b>
10.8.6	<b>Testi metode UpdateCardPassengerStatus.....</b>	<b>486</b>

## DS 4: Integracija IS sistemov prevoznika

10.8.7	Testi metode GetCardPassengerStatus.....	487
10.8.8	Testi metode GetRelation .....	488
10.8.9	Testi metode GeCardProducts.....	489
10.8.10	Testi metode GetProductPrice .....	490
10.8.11	Testi metode SetPos.....	491
10.8.12	Testi metode GetPos .....	492
10.8.13	Testi metode SetTerminal .....	493
10.8.14	Testi metode GetTerminal.....	494
10.8.15	Testi metode SetBlackList .....	495
10.8.16	Testi metode SetExecutableList.....	496
10.9	REFERENCE.....	497
<b>11</b>	<b>VARNOSTNA SHEMA SISTEMA .....</b>	<b>498</b>
11.1	RAVNANJE Z OBČUTLJIVIMI PODATKI .....	498
11.1.1	Ključni in certifikati.....	499
11.1.2	Osebnostni podatki .....	501
11.1.3	Transakcijski podatki.....	501
11.2	VAROVANJE PODATKOV ZNOTRAJ SISTEMA IJPP .....	501
11.2.1	Zaledni sistem IJPP .....	502
11.2.2	Terminalna oprema .....	502
11.2.3	Brezstična kartica.....	503
11.2.4	Varnostni element.....	503
11.2.5	Brezstična IJPP aplikacija .....	503
11.2.6	IJPP aplikacija na terminalski opremi.....	504
11.3	VAROVANJE POVEZAV Z ZUNANJIMI SISTEMI .....	504
11.3.1	Zaledni sistem prevoznika.....	504
11.4	VAROVANJE POVEZAV PROTI KONČNIM UPORABNIKOM (ADMINISTRACIJA, SPLETNI PORTAL) .....	504
<b>12</b>	<b>PODPORA SISTEMA.....</b>	<b>505</b>
12.1	DEFINICIJA PODPORE SISTEMA .....	505
12.2	DEFINICIJE .....	505
12.3	NIVOJI PODPORE .....	505
12.3.1	Prvi nivo .....	506
12.3.1.1	Naloge .....	506
12.3.1.2	Potek delovanja .....	506
12.3.1.3	Odgovorna oseba .....	506
12.3.2	Drugi nivo.....	506
12.3.2.1	Naloge .....	506
12.3.2.2	Potek delovanja .....	507
12.3.2.3	Odgovorna oseba .....	507
12.3.3	Tretji nivo .....	507
12.3.3.1	Potek delovanja .....	507
12.3.3.2	Odgovorna oseba .....	507
12.3.4	Četrty nivo.....	508
12.3.4.1	Potek delovanja .....	508
12.3.4.2	Odgovorna oseba .....	508
12.4	DEFINICIJE RAVNI RESNOSTI PROBLEMA .....	508
12.4.1	Kritični poslovni učinek.....	508
12.4.1.1	Primeri kritičnih učinkov.....	508

## DS 4: Integracija IS sistemov prevoznika

12.4.2	Resni poslovni učinek .....	508
12.4.2.1	Primeri resnih učinkov .....	509
12.4.3	Srednji poslovni učinek .....	509
12.4.3.1	Primeri srednjih učinkov .....	509
12.4.4	Manjši oziroma ničelni poslovni učinek .....	509
12.4.4.1	Primeri manjših, ničelnih učinkov .....	509
12.4.5	Dodatna zahteva .....	509
12.5	ODZIVNI ČASI PODPORE .....	510
12.6	RAZDELITEV ODGOVORNOSTI .....	511
12.6.1	Prvi nivo .....	511
12.6.2	Drugi nivo .....	511
12.6.3	Tretji nivo .....	511
12.6.4	Četrty nivo .....	511
12.7	PREHOD MED NIVOJI PODPORE .....	511
12.8	SLEDENJE ZAHTEV .....	512
12.9	NAČINI KOMUNIKACIJE MED NIVOJI .....	512
12.10	DOTIČNA TOČKA .....	512
12.11	VZOREC OBRAZCA NA KLICNEM CENTRU .....	513
<b>13</b>	<b>POSTOPEK SERVISIRANJA TERMINALOV IJPP .....</b>	<b>513</b>
13.1	ENTITETE VKLJUČENE V POSTOPEK SERVISA .....	513
13.2	POŠKODBE, UNIČENJE ALI IZGUBA IN/ALI KRAJA TERMINALA TER POVRAČILO ŠKODE .....	514
13.3	SPLOŠNI POTEK DOGODKOV .....	514
13.4	UGOTAVLJANJE GARANCIJ .....	515
13.5	CIKEL TERMINALA IJPP .....	515
13.6	SKLADIŠČE NOVIH TERMINALOV .....	516
13.7	VARNA SOBA .....	516
13.8	POSTOPEK TESTIRANJA TERMINALA .....	517
13.8.1	Stanja na zaslonu ob testiranju .....	517
13.8.1.1	Normalno stanje .....	517
13.8.1.2	Branje servisne kartice .....	517
13.8.2	Zagon terminala .....	517
13.8.3	Čitanje ostalih kartic .....	517
13.9	SKLADIŠČE IJPP TERMINALOV .....	518
13.10	MONTAŽA .....	518
<b>14</b>	<b>SISTEMI ZA PROMETNO PLANIRANJE .....</b>	<b>519</b>
14.1	OPIS PROCESOV .....	519
14.1.1	Opis procesov za Enotni daljinar .....	519
14.1.2	Opis procesov za Nacionalni vozni red .....	520
<b>15</b>	<b>DEFINICIJE METOD IN PODATKOVNIH STRUKTUR ZA SPLETNE STORITVE – PROMETNI SISTEMI IJPP ...</b>	<b>522</b>
15.1	DEFINICIJE METOD ZA SPLETNE STORITVE – PROMETNI SISTEMI IJPP .....	522
15.1.1	GetPrevozniki .....	522
15.1.2	GetPostajalisca .....	523
15.1.3	GetPostajalisca_Pladenj .....	524
15.1.4	GetRelacije .....	525
15.1.5	GetPostajneTocke .....	526

## DS 4: Integracija IS sistemov prevoznika

15.1.6	GetCone .....	527
15.1.7	GetLokacije.....	528
15.1.8	GetLinijskiOdseki .....	529
15.1.9	GetCasovniRezimi.....	530
15.1.10	GetPrestopneTocke .....	531
15.1.11	GetPrestopi .....	532
15.1.12	GetRezimi .....	533
15.1.13	GetRezimiZaPrevoznika.....	534
15.1.14	GetVozniRediZaPrevoznika .....	535
15.1.15	GetVozniRediIndexZaPrevoznika.....	537
15.1.16	GetPostajneTockeZaPrevoznika .....	538
15.1.17	SetDelovnaNalogaZaPrevoznika .....	539
15.1.18	SpremeniVeljavnostDelovneNaloge .....	541
15.1.19	GetNearestStations.....	542
15.1.20	GetNearestStations_Pladenj .....	543
15.1.21	SetLocation.....	544
15.2	DEFINICIJE PODATKOVNIH STRUKTUR – PROMETNI SISTEMI IJPP.....	546
15.2.1	Prevoznik.....	546
15.2.2	Postajališče .....	547
15.2.3	Postajališče_Pladenj .....	549
15.2.4	Relacija.....	551
15.2.5	Postajna točka .....	552
15.2.6	Postajna točka – Ext.....	553
15.2.7	Geometrija .....	554
15.2.8	Cona .....	555
15.2.9	Lokacija .....	556
15.2.10	Linijski odsek .....	557
15.2.11	Časovni režim .....	558
15.2.12	Časovni faktor .....	558
15.2.13	Prestopna točka .....	559
15.2.14	Prestop .....	560
15.2.15	Vozni redi – Index.....	561
15.2.16	Vožnja – Index.....	562
15.2.17	Vožnja opis – Ext .....	566
15.2.18	Vozni red – Ext .....	563
15.2.19	Vožnja – Ext.....	565
15.2.20	Vozni red – Emb .....	566
15.2.21	Vožnja – Emb .....	568
15.2.22	Vožnja – Grp.....	569
15.2.23	Vožnja opis – Emb .....	570
15.2.24	Režim – Ext.....	571
15.2.25	Režim termin .....	572
15.2.26	Režim interval .....	573
15.2.27	Delovna naloga .....	574
15.2.28	StationDetails.....	575
15.2.29	StationDetails_Pladenj.....	576
15.2.30	PackageStatus.....	577

## DS 4: Integracija IS sistemov prevoznika

15.2.31	Package.....	578
<b>16</b>	<b>APLIKACIJA ZA VALIDACIJO IN NAKUP.....</b>	<b>579</b>
16.1	FUNKCIONALNOST (SPLOŠNO) .....	580
16.2	KOMUNIKACIJA Z ZALEDNIM SISTEMOM.....	581
16.3	PRIJAVA/ODJAVA .....	581
16.4	IZBIRA PARAMETROV DELOVANJA.....	582
16.5	BRANJE INFO KARTICE.....	582
16.6	VALIDACIJA .....	582
16.6.1	VALIDACIJA CONSKI PROMET.....	582
16.6.2	VALIDACIJA RELACIJSKI PROMET.....	583
16.7	NAKUP VOZOVNIC .....	583
16.7.1	NAKUP CONSKI PROMET.....	583
16.7.1.1	BREZSTIČNA KARTICA .....	583
16.7.1.2	LISTNA .....	583
16.7.2	NAKUP RELACIJSKI PROMET.....	584
16.7.2.1	BREZSTIČNA KARTICA .....	584
16.7.2.2	LISTNA .....	584
16.8	STORNO .....	584
16.9	BLOKADA APLIKACIJE .....	585
16.10	BLOKADA VOZOVNICE .....	585
16.11	PREVERJANJE USPEŠNOSTI ZAPISA VOZOVNICE .....	585
16.12	FINANČNA POROČILA IZMENE.....	585
16.13	GUI.....	585
16.13.1	Definicije .....	586
16.13.2	GUI - mobilni terminal ne razpolaga s SIM kartico .....	587
16.13.3	GUI - vnos SIM PIN .....	588
16.13.4	GUI - vnos SIM PUK .....	589
16.13.5	GUI - registracija mobilnega terminala v GPRS omrežje .....	591
16.13.6	GUI - avtentikacija mobilnega terminala z zalednim sistemom - servisna storitev .....	592
16.13.7	GUI - mobilni terminal ne razpolaga z aktualno tarifo .....	594
16.13.8	GUI - mobilni terminal ne razpolaga z aktivnim seznamom kartic na črni listi .....	595
16.13.9	GUI - mobilni terminal ne razpolaga z aktivnim seznamom vozovnic na črni listi .....	596
16.13.10	GUI - mobilni terminal ne razpolaga z dejanskim časom .....	597
16.13.11	GUI - prijava voznika prevoznega sredstva IJPP .....	599
16.13.12	GUI - odjava voznika prevoznega sredstva IJPP in avtomatično tiskanje poročil .....	602
16.13.13	GUI – validacija – mestni, medkrajevni PROMET .....	605
16.13.14	GUI – podaljševanje terminskih vozovnic na mobilnem terminalu, mestni in medkrajevni promet .....	609
16.13.15	GUI – stornacija vozovnic na mobilnem terminalu, mestni, medkrajevni promet.....	613
16.13.16	GUI – Poizvedba o vsebini brezstične ijpp kartice (Informacija kartice) .....	616
16.13.17	GUI – NAKUP LISTNE VOZOVNICE v TISKANI OBLIKI oz. KOT PRODUKT SHRANJEN NA IJPP KARTICO .....	621
16.13.18	POMEN STATUSNIH IKON .....	626
16.13.19	SPOROČILA NAPAK .....	627
16.14	SISTEMSKA FUNKCIONALNOST (PRAVILA/DIAGRAMI) .....	629
16.14.1	INICIALIZACIJA APLIKACIJE IN SISTEMSKIH PODATKOV.....	629
16.14.2	KOMUNIKACIJA Z ZALEDNIM SISTEMOM.....	630



## DS 4: Integracija IS sistemov prevoznika

16.14.3	SERVISNA TRANSAKCIJA .....	630
16.14.4	TARIFE .....	631
16.14.5	ČRNE LISTE .....	631
16.14.6	IZVRŠNE LISTE .....	631
16.14.7	KARTIČNE TRANSAKCIJE .....	632
16.14.8	SEZNAMI VOZNIH REDOV .....	632
16.14.9	SEZNAM PRAVIL PREHODOV MED CONAMI .....	633
16.14.10	POSODOBITVE .....	633
16.14.11	DOLOČITEV USTREZNOSTI CONSKE VOZOVNICE .....	633
16.14.12	DOLOČITEV USTREZNOSTI RELACIJSKE VOZOVNICE .....	633
16.14.12.1	Algoritem za preverjanje dvosmerne relacijske vozovnice .....	634
16.14.12.2	Algoritem za preverjanje enosmerne relacijske vozovnice .....	634
16.15	PROTOKOLI KOMUNIKACIJE MED ČITALNIKOM IN KARTICO .....	636
16.16	APLIKACIJSKI PROTOKOL KOMUNIKACIJE MED ČITALNIKOM IN KARTICO .....	638
16.16.1	PRIJAVA/ODJAVA .....	638
16.16.1.1	Protokol Prijave .....	638
16.16.2	BRANJE INFO KARTICE .....	641
16.16.2.1	Protokol Branje INFO kartice .....	644
16.16.3	VALIDACIJA IN AKTIVACIJA .....	648
16.16.3.1	Protokol VALIDACIJA in AKTIVACIJA .....	648
16.16.4	NAKUP VOZOVNICE .....	653
16.16.4.1	Protokol NAKUP VOZOVNICE .....	653
16.16.5	STORNO NAKUPA VOZOVNICE .....	655
16.16.5.1	Protokol STORNO NAKUPA VOZOVNICE .....	656
16.16.6	BLOKADA APLIKACIJE IJPP .....	659
16.16.6.1	Protokol BLOKADA APLIKACIJE IJPP .....	659
16.16.7	BLOKADA VOZOVNICE .....	662
16.16.7.1	Protokol BLOKADA VOZOVNICE .....	662
16.16.8	PREVERJANJE USPEŠNOSTI ZAPISA VOZOVNICE .....	663
16.16.8.1	Protokol PREVERJANJE USPEŠNOSTI ZAPISA VOZOVNICE .....	664
16.17	REFERENCE .....	669
17	DODATEK OBRAZCI ZA TESTIRANJE .....	670
18	DODATEK OBRAZEC ZA SERVISIRANJE .....	671



## KAZALO SLIK

SLIKA 1.1: OSNOVNI GRADNIKI INTEGRIRANEGA JAVNEGA POTNIŠKEGA PROMETA IJPP .....	33
SLIKA 1.2: POENOSTAVljena SHEMA INFRASTRUKTURE JAVNEGA POTNIŠKEGA PROMETA V POVEZAVI Z UPRAVLJAVCEM IN PREVOZNIKI .....	34
SLIKA 1.3: OSNOVNA TOČKA T1 ZNOTRAJ MODELA JAVNEGA POTNIŠKEGA PROMETA .....	36
SLIKA 1.4: POVEZAVA P1 MED TOČKO T1 IN TOČKO T2 S PRIPADAJOČIMI OZNAKAMI IN PARAMETRI .....	36
SLIKA 1.5: DEFINICIJA POTI L1 S PRIPADAJOČIMI OZNAKAMI IN PARAMETRI .....	37
SLIKA 1.6: DEFINICIJA CONE S PRIPADAJOČIMI TOČKAMI .....	37
SLIKA 1.7: OSNOVNA IN POENOSTAVljena SHEMA INFORMACIJSKEGA SISTEMA V POVEZAVI Z UPRAVLJAVCEM SISTEMA IN PREVOZNIKI .....	48
SLIKA 1.8: PONAORITEV POVEZAVE MED ZALEDNIM SISTEMOM IJPP, ZALEDNIM SISTEMOM PREVOZNIKOV IN TERMINALSKO OPREMO IJPP .....	54
SLIKA 2.1: OSNOVNI ELEMENTI IJPP SISTEMA S PRIPADAJOČIMI POSTOPKI IN PRIPADAJOČO TELEKOMUNIKACIJSKO OPREMO ZNOTRAJ SISTEMA IJPP .....	59
SLIKA 2.2: OSREDNJI POSTOPKI PRIDOBITVE KARTICE, NAKUPA VOZOVNIC TER NATO KORIŠČENJE PRI PREVOZNIKI DO IZTEKA ŽIVLJENSKE DOBE BREZSTIČNE KARTICE Z NAMEŠČENO IJPP APLIKACIJO. ....	63
SLIKA 2.3: ŽIVLJENJSKI CIKEL BREZSTIČNE KARTICE Z NAMEŠČENO APLIKACIJO IJPP .....	64
SLIKA 2.4: DELOVANJE SISTEMA ZA IZDAJO NOVIH IN PREGLED STARIH KARTIC TER PRODUKTOV .....	65
SLIKA 2.5: ŽIVLJENJSKI CIKEL SISTEMA ZA IZDAJO, NOVIH IN PREGLED STARIH KARTIC TER PRODUKTOV .....	69
SLIKA 2.6: SHEMA GRADNIKOV TERMINALA NA PRODAJNIH MESTIH PREVOZNIKA IN DRUGIH PRODAJNIH MESTIH .....	70
SLIKA 2.7: ŽIVLJENJSKI CIKEL TERMINALA NA PRODAJNIH MESTIH PREVOZNIKA IN DRUGIH PRODAJNIH MESTIH .....	71
SLIKA 2.8: OSNOVNI GRADNIKI TERMINALA ZA NAMENE VALIDACIJE PRODUKTOV .....	72
SLIKA 2.9: ŽIVLJENJSKI CIKEL TERMINALA ZA NAMENE VALIDACIJE PRODUKTOV IJPP .....	73
SLIKA 2.10: OSNOVNI GRADNIKI TERMINALA ZA NAMENE KONTROLE PRODUKTOV .....	73
SLIKA 2.11: ŽIVLJENJSKI CIKEL TERMINALA ZA NAMENE KONTROLE PRODUKTOV IJPP .....	74
SLIKA 2.12: SHEMA POMEMBNEJŠIH NALOG UPRAVITELJA SISTEMA IJPP .....	76
SLIKA 2.13: SHEMA ZALEDNEGA SISTEMA IJPP .....	77
SLIKA 2.14: ŽIVLJENJSKI CIKEL ZALEDNEGA SISTEMA IJPP .....	78
SLIKA 3.1: ZGRADBA KARTICE MIFARE DESFIRE .....	85
SLIKA 3.2: MINIMALNI PRAZNI PROSTOR IN VELIKOST, MREŽA 2 X 2 MM RAZMERJA .....	88
SLIKA 3.3: POSTOPEK PRIJAVE .....	100
SLIKA 3.4: BRANJE INFO KARTICE – SPLOŠEN PREGLED .....	101
SLIKA 3.5: BRANJE INFO KARTICE .....	102
SLIKA 3.6: NAKUP VOZOVNICE .....	110
SLIKA 3.7: POTEK POSTOPKA VALIDACIJE IN AKTIVACIJE .....	114
SLIKA 3.8: POSTOPEK VALIDACIJE .....	115
SLIKA 3.9: POSTOPEK AKTIVACIJE VOZOVNICE .....	116
SLIKA 3.10: POSTOPEK PREVERJANJA VOZOVNICE .....	117
SLIKA 3.11: STORNO NAKUPA VOZOVNICE .....	120
SLIKA 3.12: POSTOPEK PRI POLNEM VRAČILU - IZPLAČILU VOZOVNICE .....	122
SLIKA 3.13: BLOKADA APLIKACIJE IJPP .....	128
SLIKA 3.14: POSTOPEK BLOKADE VOZOVNICE .....	130
SLIKA 3.15: POSTOPEK DEBLOKADE APLIKACIJE IJPP .....	133

## DS 4: Integracija IS sistemov prevoznika

SLIKA 3.16: POSTOPEK DEBLOKADE VOZOVNICE .....	136
SLIKA 3.17: PREVERJANJE USPEŠNOSTI ZAPISA VOZOVNICE.....	142
SLIKA 3.18: BLOKOVNI DIAGRAM ALGORITMA DIVERZIFIKACIJE ZA KLJUČE 3KTDES .....	155
SLIKA 3.19: POTEK KOMUNIKACIJE MED AVTENTIKACIJO IJPP KARTICE .....	157
SLIKA 3.20: POTEK ZAPISA NA IJPP KARTICO.....	157
SLIKA 3.21: POTEK BRANJA IZ IJPP KARTICE .....	158
SLIKA 3.22: VELIKOSTI, KI JIH DEFINIRA STANDARD ISO 7810 IN VELIKOST ID-000 (SIM) .....	159
SLIKA 3.23: POZICIJA KONTAKTOV, KI JIH DEFINIRA STANDARD ISO 7810 .....	159
SLIKA 3.24: STRUKTURA PRENOSNEGA BLOKA PROTOKOLA T=1 .....	160
SLIKA 3.25: BLOKOVNA SHEMA SAM MODULA .....	161
SLIKA 3.26: VARNI KANAL SAC MED VARNOSTNIM ELEMENTOM IN APLIKACIJO IJPP .....	164
SLIKA 3.27: PRIMER POVEZAVE MED FID 0X02 IN FID 0X03 .....	168
SLIKA 3.28: SHEMATIČNI PRIKAZ GENERIRANJA PODPISA .....	181
SLIKA 4.1: PRIMER PODATKOVNEGA POLJA S CENAMI.....	183
SLIKA 4.2: PRIKAZ RAZLIČNIH TIPOV TARIF .....	184
SLIKA 4.3: DOLOČANJE NEAKTIVNEGA ALI AKTIVNEGA ČASA VOZOVNICE .....	191
SLIKA 4.4: DOVOLJENI RELATIVNI ČAS NAKUPA VOZOVNICE.....	194
SLIKA 4.5: ZGRADBA PODATKOVNE STRUKTURE TARIFE .....	199
SLIKA 4.6: HIERARHIJA PRI PRODAJI VOZOVNIC.....	200
SLIKA 4.7: PRAKAZ LOKACIJ UPORABE IN SEZNAMA TARIF, KI SE POŠLJEJO NA TERMINALE .....	201
SLIKA 5.1: OSNOVNI KONCEPT IZVEDBE ODDALJENIH OPRAVIL Z IZVRŠNO LISTO .....	207
SLIKA 5.2: OFFLINE NAČIN DELOVANJA IZVRŠNE LISTE .....	209
SLIKA 5.3: ONLINE NAČIN DELOVANJA IZVRŠNE LISTE .....	210
SLIKA 6.1: SHEMA INTEGRACIJE Z ZALEDNIM SISTEMOM.....	213
SLIKA 11.1: KOMUNIKACIJSKE POVEZAVE PRI MODELU A IN MODELU B .....	498
SLIKA 11.2: KOMUNIKACIJSKE POVEZAVE V SISTEMU IJPP (MODEL A); RDEČA POVEZAVA PONAZARJA INFORMACIJE, KI NISO DEFINIRANE ZNOTRAJ SISTEMA IJPP .....	502
SLIKA 11.3: KOMUNIKACIJSKE POVEZAVE V SISTEMU IJPP (MODEL B).....	502
SLIKA 13.1: ŽIVLJENJSKI CIKEL IJPP TERMINALA .....	516
SLIKA 16.1: POSTOPEK PRIČETKA USTREZNE TRANSAKCIJE .....	630
SLIKA 16.2: ALGORITEM ZA DOLOČANJE USTREZNOSTI RELACIJSKE VOZOVNICE .....	636
SLIKA 16.3: PRIJAVA .....	641
SLIKA 16.4: BRANJE INFO KARTICE – SPLOŠEN PREGLED .....	642
SLIKA 16.5: BRANJE INFO KARTICE .....	643
SLIKA 16.6: POTEK POSTOPKA VALIDACIJE IN AKTIVACIJE .....	651
SLIKA 16.7: POSTOPEK VALIDACIJE .....	652
SLIKA 16.8: POSTOPEK AKTIVACIJE VOZOVNICE .....	653
SLIKA 16.9: NAKUP VOZOVNICE .....	655
SLIKA 16.10: STORNO NAKUPA VOZOVNICE .....	658
SLIKA 16.11: BLOKADA APLIKACIJE IJPP .....	661
SLIKA 16.12: BLOKADA VOZOVNICE .....	663
SLIKA 16.13: PREVERJANJE USPEŠNOSTI ZAPISA VOZOVNICE .....	668

## KAZALO TABEL

TABELA 3.1: OPIS SESTAVNIH DELOV PROTOKOLA T=1 .....	160
TABELA 4.1: DEFINICIJA VREDNOSTI VRST VOZOVNIC .....	188
TABELA 4.2: DEFINICIJA ČASOVNIH VREDNOSTI .....	189
TABELA 4.3: DEFINICIJA VREDNOSTI ČASOVNIH ENOT .....	190
TABELA 4.4: DEFINICIJA STATUSOV .....	196
TABELA 4.5: TABELA S PREHODI MED OBMOČJI .....	202
TABELA 5.1: TIP PODATKOV ADDITIONALCARD OBJECT GLEDE NA VREDNOST ACTION .....	212
TABELA 7.1: KODE NAPAK PRI ZAVRNITVI TRANSAKCIJE .....	269
TABELA 10.1: POSTOPEK VERIFIKACIJE .....	390
TABELA 11.1: LASTNOSTI KLJUČEV IN CERTIFIKATOV V SISTEMU IJPP .....	500
TABELA 12.1: TABELA ODZIVNIH ČASOV .....	510
TABELA 12.2: PREHODI MED NIVOJI .....	511

# 1 UVOD IJPP

Dokument predstavlja standard sistema integriranega javnega potniškega prometa (v nadaljevanju projekt IJPP) v RS katerega cilj je uvedba enotne vozovnice ter poenotenje voznih redov na ravni avtobusnega in železniškega prometa. Projekt je ena izmed aktivnosti Ministrstva za infrastrukturo, ki prispeva k uresničitvi Resolucije o prometni politiki Republike Slovenije.

Uvedba integriranega javnega potniškega prometa pomeni združitev, nadgradnjo in integracijo različnih že obstoječih sistemov elektronske vozovnice pri prevoznikih v JPP v Sloveniji ter jih integrirati s sistemi upravljavca IJPP. V današnjem času, se pri javnem prevozu porablja vse več različnih tehnoloških in informacijskih sistemov, za zagotovitev učinkovitega delovanja. Poleg prvotne naloge, kot je prevoz potnikov na določenih poteh, je naloga javnega prevoza zagotovitev uporabnikom prijaznih storitev, zanesljivo in točno delovanje. Za zagotovitev prijaznih storitev potnikov in drugih uporabnikov sistema so potrebne pravilne in smiselne nastavitve urnikov in voznih redov, upravljanje voznih parkov, izdaja vozovnic, zagotovitev enostavne validacije in identifikacije potnikov in uporabnikov sistema.

Sedanji javni potniški promet v RS sestoji iz različnih sistemov različnih proizvajalcev in tehnologij. Cilj dokumenta je povezava, standardizacija povezave in nadgradnja obstoječih različnih sistemov v enoten sistem, ki bo omogočal komunikacijo in izmenjavo informacij na vseh nivojih integriranega sistema.

Za dosego usklajenega delovanja, že obstoječih sistemov, je potrebna nadgradnja, s katero se zagotovi vnaprej definirano delovanje podsistemov, struktur in pomen podatkov. To dosežemo s predpisano povezavo podsistemov znotraj organizacije, ki jo vodi en nadzorni organ (upravljavalec sistema).

Za določitev učinkovitega delovanja javnega prevoza je potrebna vzpostavitev standarda, ki zagotavlja okvir za opredeljevanje in delovanje celotnega območja javnega prevoza. Standard integriranega javnega prevoza zagotavlja, da lahko operaterji, prevozniki in ponudniki opreme, zagotovijo lažje delovanje in integracijo sistema javnega prevoza. Standard omogoča nadgradnjo v smislu posodabljanja in vključitve novejših tehnoloških rešitev. Prav tako določa nadgradnjo programske in/ali strojne opreme obstoječih in delujočih sistemov elektronske vozovnice pri prevoznikih v JPP v Sloveniji, oziroma implementacija enega od delujočih sistemov elektronskih vozovnic v primerljivem okolju. Primerljivo okolje pomeni, da so v sistemu integrirane vse tri vrste prevozov (železniški, medkrajevni avtobusni in mestni prevoz potnikov) in je obseg implementiranega in delujočega sistema najmanj takšen, kot je predviden v RS glede na število voznih sredstev in prodajnih mest, vključenih v ta delujoč sistem.

## 1.1 SPLOŠNO

#### DS 4: Integracija IS sistemov prevoznika

---

Standard integriranega javnega potniškega prometa IJPP v RS definira in specificira vse potrebne segmente za nadgradnjo obstoječih sistemom, razvoj novih aplikacij in njihove medsebojne integracije pri prevoznikih v celoten sistem s katerim upravlja upravljavec sistema.

Celotni sistem javnega potniškega prometa JPP na območju RS vključuje vse prevoznike in upravljavca sistema IJPP. Standard sistema integriranega javnega potniškega prometa v RS definira posamezne segmente sistema IJPP, ki so podrobno opisani v nadaljevanju. Pri razvoju standarda enotnega sistema javnega potniškega prometa na območju RS so se upoštevale različni rezultati predhodnih študij in projektov, izdelanih v okviru projekta Uvedba integriranega javnega potniškega prometa v RS in jih integrirati v enotni sistem.

Izhodišča za izdelavo standarda enotnega sistema slovenskega javnega potniškega prometa so:

- Uvedba mešanega relacijsko - conskega sistema. Conski sistem v okviru mestnega prometa in relacijski v medkrajevnem prometu zunaj conskega prometa.
- Uvedba enotnega tarifnega sistema.
- Implementacija prilagojenega bruto modela bruto model (prihodki od prodaje vozovnic pripadajo upravljavcu, ki ima popoln vpogled v vse transakcije, sredstva pa izvajalci GJS nakazujejo upravljavcu enkrat mesečno; prevoznik dobi plačilo po prevoženih km in dogovorjeno stimulacijo). Model mora biti ustrezno modificiran za povezave z mestnim potniškim prometom. Možnost nadgradnje bruto modela v kompleksnejši neto model.

Osnovni cilji standarda enotnega sistema slovenskega javnega potniškega so zagotovitev naslednjih funkcionalnosti:

- Postavitev koncepta IT komponent znotraj IJPP sistema za informacijsko podporo delovanja celotnega IJPP sistema.
- Možnost nadgradnje in postavitev zahtev za nadgradnjo obstoječih informacijskih sistemov (funkcionalnost, vsebina in ažurnost podatkov, tehnologija).
- Postavitev koncepta IT komponent za informacijsko podporo delovanja sistema z nadzorom upravljavca sistema.
- Določitev specifikacij potrebnih za nadgradnjo obstoječih sistemov in način medsebojne integracije in povezovanja.
- Določitev specifikacij funkcionalnosti novih sistemov, ki jih je potrebno razviti in integrirati v enotni IJPP sistem.
- Postavitev podrobnega načrta podatkovnih tokov med prevozniki oziroma prodajalci vozovnic in informacijskim sistemom upravljavca.
- Postavitev okvirjev za vzpostavitev in delovanje IT infrastrukture IJPP sistema, ki je potrebna za delovanje informacijskih sistemov upravljavca in povezovanje z informacijskimi sistemi / podsistemi posameznih prevoznikov znotraj JPP sistema.
- Vzpostavitev podpore upravljavcu sistema IJPP za možnost vzdrževanja in morebitne nadgradnje informacijskih sistemov.

#### DS 4: Integracija IS sistemov prevoznika

---

- Vzpostavitev podpore upravljavcu sistema IJPP za vzdrževanju in hranjenju podatkov v okviru implementiranih informacijskih sistemov.
- Definiranje postopkov za skrbništvo nad izdelavo in distribucijo varnostnih ključev sistema.
- Definiranje postopkov za preverjanja skladnosti opreme za potrebe IJPP in njegova izvedba.
- Definiranje enotne tarife sistema tarife ter določanje splošnih in posebnih pogojev uporabe, prodaje in obračunov produktov IJPP.
- Registracija opreme za validacijo in prodajo ter vključitev novih ali izključitev prodajnih mest sistema IJPP.
- Vključitev nove ali odstranitev neustrezne opreme za prodajo in validiranje iz sistema (avtomatiziran sprejem podatkov od prevoznikov).
- Registracija in identifikacija prevoznikov in prodajnih mest znotraj IJPP sistema.
- Vodenje evidence izdanih elektronskih brezstičnih kartic (avtomatiziran sprejem podatkov od izdajateljev elektronskih kartic).

Za vzpostavitev IJPP sistema v RS se uvede enotni tarifni sistem, ki temelji na relacijsko-conskem modelu prometno-geografske obravnave ozemlja RS. Vzpostavitev enotnega tarifnega sistema temelji na poenotenju tarifne sheme, tarifne politike in prometnega obravnavanja geografskega prostora RS, ki je osnova za določanje cen enotne vozovnice in posameznih produktov IJPP za avtobusni in železniški prevoz. Na urbanih mestnih območjih se uporabi conski sistem, ki ohrani sedanjo strukturo in geografsko razporeditev con, kjer se izvajajo mestni prevozi. Na ostalem območju RS, se kot osnova za oblikovanje novega tarifnega sistema uporabi relacijski kilometrski geografski model. To je območje, kjer se trenutno izvaja medkrajevni linijski avtobusni prevoz in železniški prevoz. Enotni tarifni sistem relacijskega območja temelji na poenotenju tarif posameznih kilometrskih relacijskih razredov IJPP medkrajevnega avtobusnega in železniškega prevoza.

IJPP predvideva nadgradnjo obstoječih sistemov skladno s standardom IJPP. Na vsako kartico prevoznikov, ki je že sedaj v uporabi in izpolnjuje zahteve Standarda IJPP se bo vpisala IJPP aplikacija (postopki upravljanja z IJPP kartično aplikacijo so zajeti v nadaljevanju dokumenta), ki je neodvisna od drugih aplikacij na kartici. Terminalska oprema IJPP mora biti skladna s standardom IJPP, torej mora omogočati operacije avtentikacija / pisanje / branje podatkov z IJPP aplikacijo na kartici. Prav tako mora terminalska oprema omogočati prejemanje podatkov (tarifa, vozni red, črna lista, izvršna lista,...) in sporočanje podatkov (validacija, nakup,...) proti sistemu IJPP.

Cilj standarda je implementacija in nadgradnja obstoječih sistemov prevoznikov na osnovi definiranega standarda v enoten IJPP sistem (končni izdelek), ki bo razvit in preizkušen v okviru pilotnih projektov, na območju celotne Slovenije. Izvajalec posameznega podsistema bo na osnovi standarda izvedel nadgradnjo ali na novo razvil opremo, ki bo nato preizkušena in potrjena na osnovi verifikacije, ki je prav tako predpisana v standardu. Uspešno verificirana oprema se bo lahko vključila v sistem IJPP. Upravljavac IJPP bo z razvitim sistemom imel možnost mehkega prehoda, pomeni, za začetek se prenese v polno funkcionalnost le omejen nabor produktov ter se pri tem ohrani obstoječi način poravnavanja subvencij in kompenzacij med obstoječimi subjekti (država-prevozniki-uporabniki),

dinamika širitve produktov in migracije zaledne poravnave bo popolnoma v rokah upravljavca, ki bo skladno z zakonodajo ter dogovori med vsemi zainteresiranimi subjekti upravljal z IJPP sistemom.

## 1.2 PREGLED PODROČJA STANDARDA

Standard zajema naslednje področja enotnega sistema slovenskega javnega potniškega prometa:

### 1. Uvod

Poglavje opisuje osnovne informacije o integriranem javnem potniškem prometu. Definirani so osnovni pojmi, grobe delitve sistema in entitet samega IJPP omrežja, ki so v nadaljevanju dokumenta specificirani v detajle.

### 2. Sistem IJPP

Prvi del poglavja opisuje osnovne elemente celotnega sistema slovenskega javnega potniškega prometa IJPP. Podane so osnovne definicije in specifikacije terminalske opreme na vozilih, terminalske opreme na prodajnih mestih prevoznika, definicija in specifikacije brezstične kartice, ki se bo uporabljala znotraj enotnega sistema javnega potniškega prometa. Podan so še specifikacije in opis zalednega sistema integriranega javnega potniškega prometa. Poglavje prav tako podaja definicijo življenjskega ciklusa celotnega sistema in kartice.

### 3. Brezstična pametna kartica

Poglavje podaja specifikacije in opis brezstične kartice, ki se bo uporabljala znotraj celotnega sistema slovenskega javnega potniškega prometa IJPP. Standard podaja opis fizičnih lastnosti kartice, podatkovne strukture na kartice za potrebe enotnega sistema IJPP, varnostne specifikacije, protokole za postopke in podatke za najrazličnejše operacije (validacija, nakup,...) ter na koncu še opis pogojev za zapis vzporednih aplikacij na brezstično kartico. Poglavje podaja tudi specifikacije in opis čitalnika, ki se bo uporabljal znotraj sistema IJPP.

### 4. Zgradba tarife

Četrto poglavje podaja specifikacije in opis zgradbe tarife. Na začetku poglavja se poda konceptualni opis same tarife znotraj sistema IJPP, nato se podajo opisi posameznih entitet

znotraj tarife. Poglavje prav tako poda definicije pravil, ki jih je potrebno uporabiti pri določanju cen ter na koncu še opis in pravila posodabljanja tarif.

## **5. Integracija z zalednim sistemom**

Ta poglavje definira kot prvo terminalsko opremo v povezavi z zalednim sistemom. V tem delu je definicija komunikacijskih protokolov z vso terminalsko opremo na osnovi Web Servisov (WS). Poglavje opisuje in definira zaledni sistema IJPP v povezavi z terminalsko opremo prevozniki ali zalednim sistemom prevoznikov.

## **6. Metode spletne storitve IJPP – transakcijski sistem**

Poglavje opisuje definicije metod spletne storitve IJPP, ki se nanašajo na transakcije med terminalsko opremo in zalednim sistemom na osnovi web servisov (WS).

## **7. Definicija podatkovnih struktur**

Poglavje podaja definicijo podatkovnih struktur, ki se uporabljajo za web servise (WS).

## **8. Metode spletne storitve – administracija**

To poglavje podaja seznam vseh metod spletnih storitev, ki so namenjene za administracijo sistema IJPP.

## **9. Postopki za verifikacijo strojne in programske kartice ter brezstične kartice**

To poglavje definira pogoje za strojno opremo, postopke testiranja za strojno opremo, postopke potrjevanja strojne opreme, pogoje za programsko opremo, postopke testiranja za programsko opremo, postopke potrjevanja programske opreme, pogoje za brezstično pametno kartico, postopke testiranja brezstične pametne kartice in postopke potrjevanja brezstične pametne kartice.

## **10. Varnostna shema sistema**

To poglavje opisuje ravnanje z občutljivimi podatki na različnih segmentih znotraj sistema, kot so podatki kartic ali osebni podatki. Prav tako definira ravnanje s podatki znotraj sistema ter



## DS 4: Integracija IS sistemov prevoznika

---

varnost povezav z zunanjimi sistemi in s končnimi uporabniki. Prav tako se definira varnostna politika in segmentacija dostopov do sistema.

### 11. Sankcioniranje

To poglavje definira postopke sankcioniranja in sicer v primerih kršitev predpisanih časovnih rokov sporočanja opravljenih transakcij v zaledni sistem, Kršitev pri uporabi neverificirane opreme pri prevoznikih, Kršitev pri uporabi napačne tarife in Kršitve pri uporabi personificirane kartice.

### 12. Podpora sistema

To poglavje opisuje in definira podporo sistema.

### 13. Postopek servisiranja terminalov IJPP

To poglavje definira postopke servisiranja za terminalsko opremo IJPP.

### 14. Sistem za prometno planiranje

V danem poglavju so zajeti koraki, ki so potrebni ob spremembi voznih redov (ob uvedbi / spremembi / ukinitvi linije, postajališča, voznega reda).

### 15. Aplikacija za validacijo in nakup

Poglavje podaja vse potrebne specifikacije in informacije, ki so potrebne za implementacijo storitev IJPP na mobilnih terminalih, ki se nahaja na IJPP prevoznem sredstvu (avtobus ali vlak). Upoštevanje specificiranih priporočil je ključnega pomena za končnega uporabnika, saj mora terminalska oprema različnih tehnoloških ponudnikov delovati usklajeno in po enakem principu.

### 16. Dodatek obrazec za testiranje

To poglavje podaja dodatne informacije in sicer obrazce za testiranje.

### 17. Dodatek obrazec za servisiranje

To poglavje podaja dodatne informacije in sicer obrazce za servisiranje.

### 1.3 NAMEN STANDARDA

Dokument definira vse potrebne segmente znotraj sistema integriranega javnega potniškega prometa na območju RS za potrebe nadgradnje obstoječega JPP sistema. Dokument specificira in standardizira vse potrebne procese kot komunikacija terminalske opreme prevoznikov z zalednim sistemom, definira enotni tarifni sistem IJPP znotraj celotnega sistema, definira procese povezane z delovanjem vozovnic na brezstični karticah za nakup in validacijo na terminalski opremi prevoznikov.

Namen standarda je definirati vse potrebne procese ter tudi samo programsko in strojno opremo, ki se bo uporabljala znotraj sistema IJPP. Standard je tako namenjen za nadgradnjo že obstoječe tako strojne kot programske opreme kot tudi za razvoj in vpeljavo nove strojne in programske opreme.

V primeru razvoja in nadgradnje strojne in programske opreme je standard namenjen dobaviteljem strojne in programske opreme ponudnikom storitev znotraj sistema IJPP – prevoznikom. Standard podaja vse potrebne informacije in definicije za posamezne sklope nadgradnje in razvoja programske in strojne opreme kot so:

- Standard za strojno opremo (NFC, Mifare DESFire,...)
- Standard za programsko opremo za namene komunikacije med brezstično kartico in čitalcem (proces validacije, proces nakup karte,...)
- Standard za terminalsko opremo
- Standard integracije terminalske opreme z zalednim sistemom
- Standard podatkovnih struktur, ki so potrebne za procese validacij in nakupa kart v sistemu IJPP (tarife, vozni redi, črne liste,...)
- Verifikacijski postopki za ugotavljanje primernosti opreme za sistem IJPP

Prav tako je standard namenjen tudi upravljavcu sistema IJPP. Na osnovi standarda upravljavec definira osnovne podatkovne strukture (npr. definicija enotne tarife), ki so potrebne za nemoteno delovanje sistema IJPP. Standard tako zajema definicije podatkovnih struktur (npr. tarifa), ki se prenesejo v sistem IJPP in se koristijo pri procesih nakupa ali validiranja na terminalski opremi. Standard upravljavcu omogoča naslednje osnovne funkcionalnosti:

- Določitev tarife (različne tipi kart, določanje cen)
- Določitev vozni redov (dodajanje linij, postaj,...)
- Vodenje črnih list (dodajanje ali odstranitev kartic iz črne liste)

#### DS 4: Integracija IS sistemov prevoznika

---

- Vodenje izvršnih list (dodajanje ali odstranitev kartic z določenimi parametri na izvršne liste liste)
- Določitev šifrantov (postaje, vozila,...)

## 1.4 SLOVAR

<b>Brezstična kartica</b>	Virtualno okolje na elektronski pametni kartici, ki omogoča validacijo ter nakup produktov IJPP znotraj sistema IJPP.
<b>Črne liste</b>	Črne liste predstavljajo seznam kartic, ki so bile umaknjene iz sistema IJPP iz različnih razlogov (npr. zloraba).
<b>Izvršne liste</b>	Izvršne liste predstavljajo seznam kartic z dopolnilnim parametrom, ki se bo prenašal iz zalednega sistema na terminalsko opremo.
<b>MIFARE</b>	Blagovna znamka najrazličnejših čipov podjetja NXP Semiconductors, ki se uporabljajo v brezstičnih pametnih karticah
<b>Postaja</b>	Postaja je skupina bližnjih postajnih točk, ki imajo enako ime in jih uporabniki zaznavamo kot eno, ter predstavlja njihovo geografsko težišče. Postaja/postajališče lahko pripada prestopni točki
<b>Postajna točka</b>	Postajna točka je skupina bližnjih postajnih točk, ki imajo enako ime in jih uporabniki zaznavamo kot eno, ter predstavlja njihovo geografsko težišče. Postaja/postajališče lahko pripada prestopni točki
<b>Potnik</b>	Subjekt, ki koristi storitve v javnem potniškem prometu.
<b>Prestop</b>	Prestop je mesto zamenjave linije ali prevoznega sredstva znotraj iste postaje oz. postajališča ali pešpot med dvema postajama oz. postajališčema znotraj iste prestopne točke, ki je potrebna za zamenjavo linije ali prevoznega sredstva.
<b>Prestopna točka</b>	Prestopna točka je pomembnejša postaja oz. postajališče ali skupina bližnjih postaj oz. postajališč, kjer potnik lahko zamenja linijo ali prevozno sredstvo.
<b>Prevoznik</b>	Prevozniki v sistemu IJPP predstavljajo gospodarsko javno službo, ki izvajajo prevoze potnikov in druge storitve znotraj sistema integriranega javnega potniškega prometa IJPP.
<b>Relacija</b>	Relacija je povezava med dvema zaporednima postajama oz. postajališčema v itinerarju, ki pa fizično nista nujno tudi sosednja.
<b>Tarifa</b>	Tarifa je podatkovna struktura, ki vsebuje podatke o kartah, ki jih je mogoče validirati ali kupiti prav tako določa pravila za prodajo in validacijo vozovnic v transakcijskem sistemu IJPP.

#### DS 4: Integracija IS sistemov prevoznika

<b>Terminalska oprema</b>	Programska in strojna oprema znotraj IJPP sistema, ki omogoča zagotovitev osnovnih funkcionalnosti, kot so validacija, nakup produktov, ter komunikacija z zalednim sistemom.
<b>Upravljavlec</b>	Upravljavlec sistema IJPP je nadzorni organ nad celotnim sistemom IJPP in skrbi za upravljanjem s celotnim sistemom.
<b>Varnostni element</b>	Elektronska naprava (ali komponenta), ki omogoča varno shranjevanje informacij, izvaja različne kriptografske operacije in omogoča varno komunikacijo z uporabo avtentikacijskih postopkov.
<b>Validacija</b>	Proces identifikacije potnika in brezstične kartice na terminalski opremi vozila
<b>Verifikacija</b>	Proces preverja strojna in programska oprema ustreza zahtevam, ki so bile postavljene v standardu.
<b>Vozni red</b>	Podatkovna struktura z naborom vseh vozniških redov in linij znotraj IJPP sistema.
<b>Zaledni sistem</b>	Zaledni sistem predstavlja osrednji del informacijskega sistema IJPP omrežja, ki omogoča upravljavcu sistema celoten vpogled in kontrolo nad informacijskim sistemom.
<b>Status</b>	Vsaka aplikacija IJPP (kartica ali mobilna aplikacija) ima znotraj sistema določen status, ki se upošteva tudi pri določitvi cene izbrane vozovnice.
<b>Vrsta vozovnice</b>	V osnovi poznamo dva tipa vozovnic terminske in količinske. Med terminske vozovnice sodijo enkratna, dnevna, mesečna, letna, poletna, itd. Količinske pa se razlikujejo po številu voženj, ki jih lahko potnik opravi z dano vozovnico.
<b>Tarifni razred</b>	Tarifni razred se razlikuje v mestnem in medkrajevnem prometu. V mestnem prometu predstavljajo tarifne razrede CONE, v medkrajevnem prometu pa so tarifni razredi določeni s km intervalom (načeloma na 5 km).
<b>Kartica IJPP</b>	Znotraj sistema IJPP ločimo dva tipa kartic; imenska in neimenska kartica.
<b>Imenska kartica</b>	Imenska kartica ima med ostalimi podatki zapisane tudi podatke o lastniku kartice. Ime in priimek ali naziv lastnika se zapiše fizično na kartico, kakor so ti podatki zapisani tudi na čip kartice in se vodijo v zalednem sistemu.

#### DS 4: Integracija IS sistemov prevoznika

---

<b>Neimenska kartica</b>	Za neimenske kartice v sistemu nimamo shranjenih podatkov o lastniku kartice. Takšne kartice imajo vedno status prenosna in se za njih ne mora izdelat duplikat.
<b>Enotni tarifni sistem</b>	Enotni tarifni sistem tvorijo enotna shema relacijskih prevozov, enotna conska tarifna shema in kombinirana relacijsko – conska tarifna shema.
<b>Incident, težava</b>	Situacija zaradi katere je stranka kontaktirala podporo sistema.
<b>Problem</b>	Dejanska napaka oziroma sum napake, ki ne povzroči spremembe delovanja izdelka.
<b>Napaka</b>	Napaka v produktu, ki povzroča, da izdelek ne deluje v skladu s specifikacijami.
<b>Popravek</b>	Sprememba v programski kodi, binarni kodi ali izvršni kodi, katera razreši napako v delovanju izdelka.
<b>Začasna rešitev</b>	Sprememba v izvajanju postopka, kateremu stranka sledi, brez bistvenega poslabšanja uporabe izdelka.

## 1.5 OZNAKE

<b>2KTDES</b>	trojni DES kriptografski algoritem z dvojnim DES ključem (ang. Double length triple DES)
<b>3KTDES</b>	trojni DES kriptografski algoritem s trojnim DES ključem (ang. Triple length triple DES)
<b>AES128</b>	napredni kriptografski algoritem (ang. Advanced encryption standard)
<b>AID</b>	identifikacijska številka aplikacije (ang. Application ID)
<b>ASK</b>	digitalna amplitudna modulacija (ang. Amplitude shift keying )
<b>CMAC</b>	MAC z uporabo kriptiranja (Cipher based MAC)
<b>CPU</b>	centralna procesna enota (ang. Central processing unit)
<b>CRC</b>	ciklično preverjanje redundance (ang. Cyclic redundancy check)
<b>DES</b>	kriptografski algoritem (ang. Data encryption standard)
<b>HSM</b>	Strojni varnostni modul za upravljanje s ključi (Hardware security module)
<b>HTTP</b>	Protokol za izmenjavo hiperteksta (Hypertext transfer protocol)
<b>HTTPS</b>	Varni HTTP (Hypertext transport protocol secure sockets)
<b>IJPP</b>	Integriran Javni Potniški Promet
<b>IP</b>	Internetni protokol (Internet protocol)
<b>IPSec</b>	Varnostni protokol za zaščito komunikacije po internetnem protokolu (IP) (Internet protocol security)
<b>MAC</b>	Overitvena oznaka sporočila (Message Authentication Code)
<b>MMU</b>	Enota za upravljanje pomnilnika (Memory management unit)
<b>NFC</b>	Near Field Communication
<b>PICC</b>	Proximity Integrated Circuit Card - Brezstična kartica
<b>PPTP</b>	Protokol za vzpostavitev navideznega zasebnega omrežja (Point to point tunneling protocol)
<b>RF</b>	radio frekvenčni vmesnik
<b>RS</b>	Republika Slovenija
<b>SAM</b>	varnostni element (ang. Secure access module)
<b>SHA</b>	Kriptografska zgoščevalna funkcija (Secure hash algorithm)

#### DS 4: Integracija IS sistemov prevoznika

---

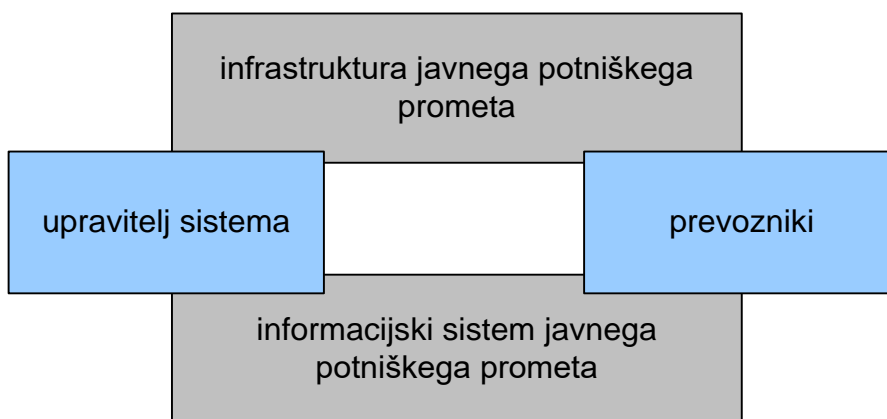
<b>SSL</b>	Sloj varnih vtičnic (Secure socket layer)
<b>TCP</b>	Protokol za upravljanje seje (Transmission control protocol)
<b>TLS</b>	Sloj varnega prenosa (Transport Layer Security)
<b>UID</b>	unikatna identifikacijska številka



## 1.6 OMREŽJE INTEGRIRANEGA JAVNEGA POTNIŠKEGA PROMETA - IJPP

Sistem javnega potniškega prometa na območju RS predstavlja osnovo za vzpostavitev in delovanje IJPP, ki vključuje vse prevoznike JPP in upravljavca sistema IJPP. Sistem javnega potniškega prometa na območju RS določa način izvajanja usklajenega voznega reda, financiranja JPP, tarifni sistem IJPP, enotno vozovnico, procese, povezane z delovanjem enotne vozovnice in model poravnave za delitev sredstev med prevozniki (kliring) in upravljavcem.

Omrežje javnega potniškega prometa lahko opišemo na več različnih načinov in sicer v odvisnosti potreb opisa in namen. Osnovi gradniki omrežja integriranega javnega prometa IJPP so prikazani na Slika 1.1.



SLIKA 1.1: OSNOVNI GRADNIKI INTEGRIRANEGA JAVNEGA POTNIŠKEGA PROMETA IJPP

Osnovni gradniki integriranega javnega potniškega prometa so:

- Infrastruktura javnega potniškega prometa

Predstavlja vse infrastrukturne točke in dele znotraj avtobusnega kakor tudi železniškega prometa.

- Upravljevec sistema

Predstavlja organ, ki skrbi za celotni integriran sistem javnega potniškega prometa. Osnovne naloge upravljavca sistema IJPP so zagotavljanje nemotenega, učinkovitega in potnikom prijaznega sistema javnega potniškega prometa.

- Prevozniki

Prevozniki izvajajo prevoze potnikov s koriščenjem infrastrukture omrežja po točno določenih voznih redih, ki jih določi upravljavec sistema.

## DS 4: Integracija IS sistemov prevoznika

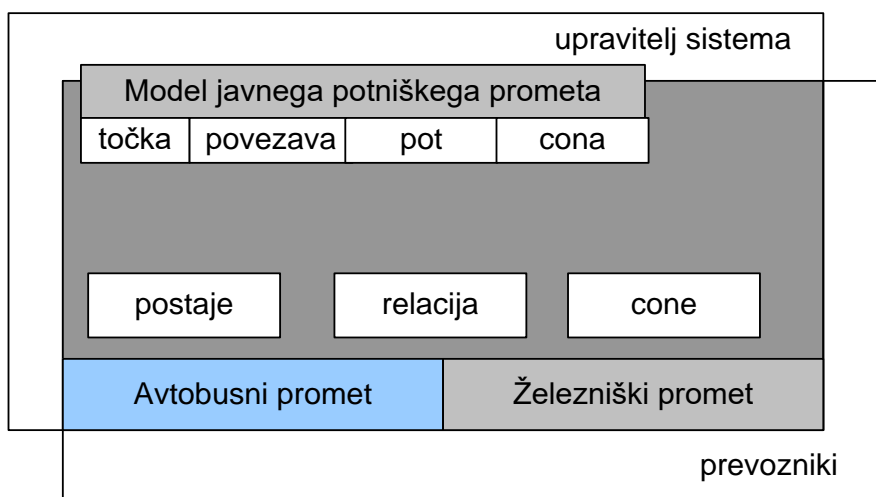
- Informacijski sistem javnega potniškega prometa

Informacijski sistem znotraj javnega potniškega prometa predstavlja poglavitno orodje tako upravljavca sistema kot prevoznikov, ki omogočajo pregled nad celotnim sistemom v smislu pridobitve informacij glede validacije potnikov, nakupa produktov, določitev enotnega tarifnega razreda ter pridobitev vseh potrebnih podatkov na enem mestu za namene poravnave in delitev sredstev med prevozniki (kliring) in upravljavcem.

V nadaljevanju bomo na kratko opisali vlogo posameznih segmentov v celotnem integrirane javnem potniškem prometu.

### 1.7 INFRASTRUKTURA OMREŽJA JAVNEGA POTNIŠKEGA PROMETA

V tem poglavju so poleg osnovne infrastrukture javnega potniškega prometa podane vse potrebne definicije tako poenostavljenega modela potniškega prometa, ki se nato prenesejo na infrastrukturne točke znotraj sistema javnega potniškega prometa. Te definicije so potrebne za nadaljnjo razumevanje dokumenta, ker nastopajo v različnih segmentih tako infrastrukturnega kot informacijskega sistema javnega potniškega prometa. Naslednja slika prikazuje poenostavljeno shemo infrastrukture javnega potniškega prometa.



SLIKA 1.2: POENOSTAVLJENA SHEMA INFRASTRUKTURE JAVNEGA POTNIŠKEGA PROMETA V POVEZAVI Z UPRAVLJAVCEM IN PREVOZNIKI

Model javnega potniškega prometa definira vse potrebne entitete za opis poti in relacij znotraj modela javnega potniškega prometa. Le te nato prenesemo na infrastrukturo omrežja in sicer tako na

avtobusni kot železniški promet. Za potrebe standarda so najprej definirani osnovni pojmi gradnikov modela javnega potniškega prometa.

### *1.7.1 DEFINICIJE OSNOVNIH GRADNIKOV POENOSTAVLJENEGA MODELA JAVNEGA POTNIŠKEGA PROMETA*

Osnovni gradniki gradnikov modela javnega potniškega prometa so:

- Točka
- Povezava
- Pot - linija
- Cone

**Točke** predstavlja osrednjo entiteto v modelu javnega potniškega prometa, saj predstavljata geografsko točko znotraj omrežja javnega potniškega prevoza, ki jo lahko nato povežemo s točko z infrastrukturnim pomenom, kot je npr. postaja.

Znotraj modelu javnega potniškega prometa ločimo več različnih tipov točk in sicer:

#### Vstopna točka:

Predstavlja začetno postajo znotraj modela omrežja javnega potniškega prometa iz katere bo prevozno sredstvo začelo pot do končne izstopne točke. Vstopna točka se lahko tudi nanaša na vmesno postajno (točka prestopa), kjer lahko potniki vstopajo in izstopajo.

#### Izstopna točka:

Izstopna točka znotraj modela javnega potniškega prometa predstavlja končno lokacijo postaje, kjer lahko potniki izstopijo iz prevoznega sredstva. Izstopna točka se nanaša tudi na vmesne postaje na poti do končne izstopne točke, kjer lahko potniki izstopajo in vstopajo.

#### Točka prestopa:

Točka prestopa znotraj modela javnega potniškega prometa predstavlja vmesno postajo med vstopno točko in izstopno točko, ki predstavljata določeno pot prevoznega sredstva.

Za točke znotraj modela javnega potniškega prometa so definirani osnovni parametri za opis le teh. Naslednja slika prikazuje osnovne oznake in parametre točke.

## DS 4: Integracija IS sistemov prevoznika

Točka T1



TOČKA T1

- oznaka
- ime
- ID točke
- zemljepisna dolžina
- zemljepisna širina

SLIKA 1.3: OSNOVNA TOČKA T1 ZNOTRAJ MODELA JAVNEGA POTNIŠKEGA PROMETA

### Povezava

Zvezo med dvema točkama (lahko različnih tipov) v modelu javnega prevoza predstavlja **povezava** - Link. Povezava je lahko različnega tipa in ima lahko različno število parametrov, kot je npr dolžina povezave ali ID same povezave. Prav tako lahko ima poveza tudi smer ali iz točke T1 v točko T2 ali obratno. Iz tega sledi da lahko imamo med dvema točkama T1 in T2 samo eno povezavo - link. Naslednja slika prikazuje osnovne oznake in parametre povezave.

T1

T2



Povezava P1

Povezava P1

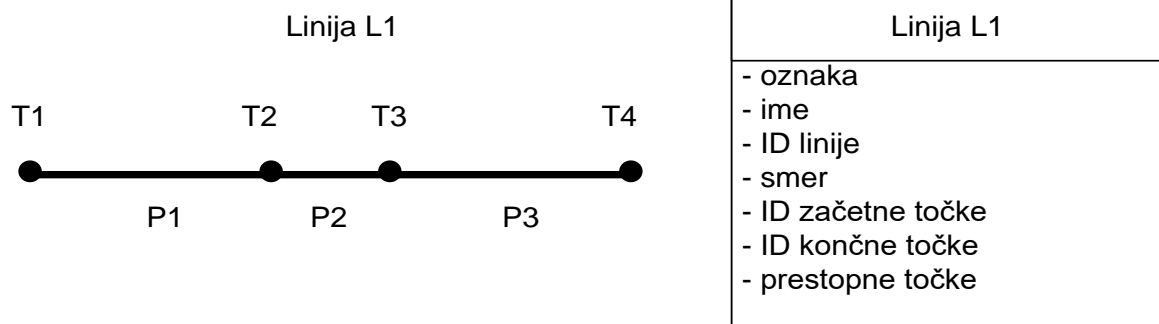
- oznaka
- ime
- ID poti
- smer
- ID začetne točke
- ID končne točke

SLIKA 1.4: POVEZAVA P1 MED TOČKO T1 IN TOČKO T2 S PRIPADAJOČIMI OZNAKAMI IN PARAMETRI

### Pot - linija

Pot predstavlja več zaporednih povezav. Definirana je lahko kot pot od vstopne do končne točke, ki jo bo opravilo prevozno sredstvo med dvema točkama. Lahko rečemo da je pot povezava med vstopno in izstopno točko, ki sta povezani z zaporedjem točk prestopa. Naslednja slika prikazuje osnovne parametre poti.

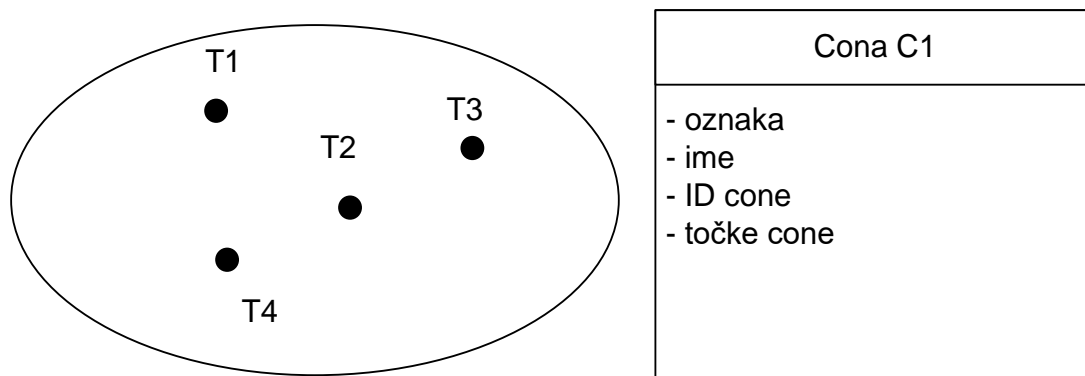
#### DS 4: Integracija IS sistemov prevoznika



SLIKA 1.5: DEFINICIJA POTI L1 S PRIPADAJOČIMI OZNAKAMI IN PARAMETRI

#### Cona

Cona, se lahko opredeli s skupino točk, ki pripadajo coni. Cona, se lahko opredeli tudi kot geometrično območje, ki je omejeno z povezavami v zaporedju. V takem primeru mora biti zaporedje povezav zaprto, kjer je začetna in končna točka skupna. Naslednja slika prikazuje osnovne parametre cone.



SLIKA 1.6: DEFINICIJA CONE S PRIPADAJOČIMI TOČKAMI

#### 1.7.2 PRENOS POENOSTAVLJENEGA MODELA NA INFRASTRUKTURO OMREŽJA JAVNEGA POTNIŠKEGA PROMETA

Po opisu in definiciji poenostavljenega modela javnega potniškega prometa na osnovi osnovnih gradnikov so definirani osnovni pojmi infrastrukture omrežja javnega potniškega prometa v RS. Tako lahko točka poenostavljenega modela javnega potniškega prometa predstavlja fizično mesto, kot je npr. območje postaje kjer potniki (praviloma) vstopajo na vozilo JPP. Postaja ali postajališče lahko opišemo z večimi točkami modela, ki imajo enako ime in jih uporabniki zaznavamo kot eno, ter

predstavlja njihovo geografsko težišče. V nadaljevanju definirajmo osnovne entitete infrastrukturnega omrežja javnega potniškega prometa.

#### 1.7.2.1 Osnovne entitete infrastrukturnega omrežja

Osnovne entitete infrastrukturnega omrežja javnega potniškega prometa so[1]:

- postajna točka,
- segment linijskega odseka,
- postaja/postajališče,
- linijski odsek,
- prestopna točka,
- prestop,
- relacija.

**Postajna točka** je točka, ki predstavlja fizično mesto oziroma težišče območja (peron), kjer potniki (praviloma) vstopajo na vozilo JPP. Postajna točka pripada postaji.

**Postaja/postajališče** je skupina bližnjih postajnih točk, ki imajo enako ime in jih uporabniki zaznavamo kot eno, ter predstavlja njihovo geografsko težišče. Postaja/postajališče lahko pripada prestopni točki.

**Prestopna točka** je pomembnejša postaja oz. postajališče ali skupina bližnjih postaj oz. postajališč, kjer potnik lahko zamenja linijo ali prevozno sredstvo.

**Segment linijskega odseka** je podrobno opisan gradnik cestne ali železniške mreže (odsekov in pododsekov), po kateri poteka določen linijski odsek. Segmentu linijskega odseka pripada polilinja.

**Linijski odsek** je usmerjena povezava med dvema zaporednima postajnama točkama v itinerarju, ki pa fizično nista nujno tudi sosednja, in poteka po osi ceste oz. proge. Linijskemu odseku pripada polilinja.

**Prestop** je mesto zamenjave linije ali prevoznega sredstva znotraj iste postaje oz. postajališča ali pešpot med dvema postajama oz. postajališčema znotraj iste prestopne točke, ki je potrebna za zamenjavo linije ali prevoznega sredstva.

**Relacija** je povezava, ki vsebuje vsaj začetno postajo in končno postajo potovanja. Relacija lahko vključuje tudi vmesno postajo (prestopna točka) za bolj natančno opredelitev veljavne smeri potovanja.

#### 1.7.2.2 Relacije med entitetami infrastrukturnega omrežja

## DS 4: Integracija IS sistemov prevoznika

---

Relacije med entitetami predstavljajo odnose med dvema ali več entitetami [1].

**Prestopna točka** vsebuje najmanj eno in največ N **postaj**. **Postaja** lahko logično pripada **prestopni točki**, lahko pa tudi ne.

**Prestopna točka** vsebuje najmanj eno in največ N **prestopov** med najmanj eno in največ dvema **postajama**.

**Prestopna točka** vsebuje najmanj eno in največ N **prestopov** med natančno dvema **postajama**, ki sta lahko isti ali različni.

**Postaja** vsebuje najmanj eno in največ N **postajnih točk**. Postajna točka logično pripada postaji, občini in tarifni coni.

**Linijski odsek** je element, ki se začne natančno pri eni **postajni točki** in konča natančno pri eni **postajni točki**.

**Linijski odsek** je sestavljen iz najmanj enega do N **segmentov linijskega odseka**.

**Relacija** se začne natančno pri eni **postaji** in konča natančno pri eni **postaji**

### 1.7.3 TIP TRANSPORTA V JAVNEM POTNIŠKEM PROMETU

Glede na tip transporta v sistemu ločimo dve skupini in sicer:

- Avtobusni promet
- Železniki promet

#### Avtobusni promet

Avtobusni promet predstavlja prevoz potnikov z avtobusi po cestnem omrežju. Avtobusni promet v RS ločimo v dve skupini in sicer:

- **mestni avtobusni promet**

Mestni avtobusni promet predstavlja avtobusni promet na področju mesta in bližnje okolice. Po navadi je mestni promet razdeljen na posamezne cone, ki pa je po navadi odvisno od velikosti samega mesta. Frekvenca prestopanja je pri mestnih avtobusih večja, razdalje, ki jih pa opravijo potniki pa krajše.

- **medkrajevni avtobusni promet**

Medkrajevni avtobusni promet predstavlja avtobusni med posameznimi mesti znotraj RS. Medkrajevni promet po navadi poteka po naprej določenih linijah. Frekvenca prestopanja je pri medkrajevnih avtobusih manjša, razdalje, ki jih pa opravijo potniki pa daljše.

## **Železniki promet**

Železniški promet predstavlja prevoz potnikov z vlaki po železniškem omrežju. Železniški promet v RS poteka po naprej definiranih linijah od končno do začetne postaje skozi vmesne postaje, ki so lahko tudi prestopne postaje.

### **1.7.4 PREVOZNI SISTEM IJPP**

Glede na različne tipe transportnih možnosti sistem IJPP vključuje tako prevoznike linijskega medkrajevnega avtobusnega prometa, železniškega prometa in mestnega prometa. Tako ločimo sledeče sisteme znotraj IJPP sistema:

#### **Conski sistem**

Conski sistem javnega potniškega prometa predstavlja sistem prevozov v geografsko-prometnem območju, ki lahko obsega območje v okviru mestnega prometa ali urbane regije. Geografsko območje cone je lahko sestavljeno iz večih ali ene same enote. Cono definira točno določeno geografsko zaključeno območje, v katerem velja enotna cena prevoza.

Znotraj conskega sistema se uporabljajo za prevoz potnikov conske vozovnice. Za ta tip vozovnice je značilno, da smer vožnje ter končna in začetna točka nista v naprej točno določena.

Vozila vozijo v conskem sistemu v conskem režimu.

#### **Relacijski sistem**

Relacijski prevozni sistem javnega potniškega prometa predstavlja je definiran kot sistem prevoza v geografsko-prometnem območju izven conskega območja, kjer je prevoz določen z relacijo potovanja in kjer sta znani začetna in končna točka potovanja, pri čemer vsaj ena od točk leži zunaj conskega območja oz. točki ne ležita v istem conskem območju.

V relacijskem prevoznem sistemu se za prevoz potnikov znotraj sistema uporabljajo tako relacijske kot tudi kombinirane vozovnice. Te vozovnice veljajo za točno določeno relacija-pot potovanja. Za to relacijo potovanja potnika je točno določena tako vstopna kot tudi izstopna postaja. S vstopno in izstopno točko je definirana tudi sam smer vožnje-potovanja.

Vozila vozijo v relacijskem sistemu v relacijskem režimu.



### **Mešani sistem v conskem območju**

Mešani sistem v conskem območju omogoča kombinacijo conskega sistema in relacijskega sistema. To omogoča, da lahko potnik s ustrezno vozovnico uporabljajo prevoz in prestopanje na vozilih, ki vozijo v conskem in relacijsko conskem režimu vožnje.

#### **1.7.5 PRESTOPNE TOČKE ZNOTRAJ OMREŽJA**

Tako avtobusno kot železniško omrežje je namenjeno za prevoz potnikov začetne do končne točke-postaje. Vmesne točke na liniji pa predstavljajo vmesne točke na vnaprej definirani poti. Te vmesne točke lahko predstavljajo tudi prestopne točke. Prestopna točka je definirana kot točka, kjer lahko potnik zamenja tako nosilca transporta kot tudi samo storitev. Prav tako prestopna točka predstavlja mesto, kjer se lahko sreča več različnih ponudnikov storitev z enako ali različnim tipom transporta.

V osnovi ločimo dva tipa prestopnih točk in sicer:

1. unimodalno prestopno točko – kjer potnik zamenja storitev in
2. multimodalno prestopno točko – kjer potnik zamenja tudi nosilca transporta.

### **Unimodalne prestopne točke**

Unimodalne prestopne točke definiramo kot tiste točke, kjer lahko uporabnik javnega omrežja – potnik zamenja storitev znotraj iste vrste javnega prevoza, pri tem pa ne spremeni tip transporta.

Glede na različne tipe avtobusnega in železniškega prometa ločimo tri različne tipe unimodalnih prestopnih točk in sicer:

- unimodalne prestopne točke mestnega avtobusnega prometa
- unimodalne prestopne točke medkrajevnega avtobusnega prometa
- unimodalne prestopne točke znotraj železniškega prometa

Uporabniki javnega prevoza se velikokrat srečajo s težavo, da jim ni na voljo direktna povezava od začetne točke do končne točke. V teh primerih so prisiljeni na določenih točkah opravijo prestop na drugo linijo-pot, da lahko dosežejo izbrano končno točko. Prav te točke, kjer so omogočeni prestopi imenujemo unimodalne točke, ki smo jih definirali od zgoraj.

Za prestop potnikov na določeni prestopnih točkah je potrebno zagotoviti ustrezen minimalni prestopni čas – interval, ki ga potnik potrebuje, da lahko zamenja prevozno sredstvo istega ali pa drugega ponudnika storitev. Prestopni čas je povezan z razdaljo, ki jo znotraj točke prestopa (razdalja med izstopno in vstopno točko ) opravi potnik.

V nasprotju z unimodalnimi prestopnimi točkami so multimodalne prestopne točke. Bimodalne prestopne točke definiramo kot prestopne točke med medkrajevnimi in mestnimi avtobusi in služijo

## DS 4: Integracija IS sistemov prevoznika

---

kot poveza med medkrajevnimi in primestnimi potovanji z mestnim prometom (ter obratno). Te prestopne točke se po navadi nahajajo v večjih krajih/mestih, ki imajo tako svoj mestni promet in povezave v druge kraje z medkrajevnim avtobusnim prometom.

### Multimodalne prestopne točke

Multimodalne prestopne točke definiramo kot tiste točke, kjer lahko uporabnik javnega omrežja – potnik zamenjata vrsto javnega prevoza. Cilj multimodalnih prestopnih točk je zagotovitev multimodalne prevozne verige, kar pomeni povezavo večih različnih tipov javnega transporta z omogočenim prestopom med posameznimi tipi v točno določenih točkah.

Multimodalne prestopne točke znotraj integriranega javnega potniškega prometa so zelo pomembne saj celoten koncept integriranega javnega prevoza temelji na ideji potovanje od začetne proti končni točki z različnimi načini prevoza (avtobus in železnica).

Multimodalne prestopne točke znotraj integriranega javnega potniškega prometa omogočajo prehod med tremi različnimi tipi storitev.

- prestopne točke medkrajevni avtobus– železnica
- prestopne točke medkrajevni avtobus – medkrajevni avtobus
- prestopne točke medkrajevni avtobus – mestni avtobus
- prestopne točke železnica – mestni avtobus

Podobno, kot v primeru unimodalnih prestopnih točk, je tudi za multimodalne prestopne točke določiti in zagotoviti minimalni prestopni čas – interval. To je čas, ki je potreben da potnik zamenja vrsto prometnega sredstva. Ta čas je odvisen od fizičnih razdalj znotraj multimodalne prestopne točke in sicer npr. razdalje med železniškim in avtobusnim postaje ali postajališča ali razdalja med avtobusno in železniško postajo.

### 1.7.6 REŽIMI VOŽNJE VOZIL

Režim vožnje vozil je določen znotraj sistema IJPP za vsako vozilo. Režim vožnje določa v kakšnem prevoznem sistemu vozijo. V sistemu IJPP poznamo sledeče režime vožnje vozil:

- **conski režim**  
Conski režim definira vožnjo vozil v conskem območju, ki prevažajo potnike v conskem sistemu.
- **relacijski režim**  
Relacijski režim definira vožnjo vozil na določeni relaciji, prevažajo potnike v relacijskem sistemu.
- **relacijsko-conski režim**

## DS 4: Integracija IS sistemov prevoznika

---

Relacijsko-conski režim definira vožnjo vozil, ki vozijo na določeni relaciji in prevažajo potnike v relacijskem sistemu, znotraj vnaprej določenih conskih območij pa prevažajo tudi potnike v conskem sistemu.

### 1.8 PREVOZNIKI

Prevozniki v sistemu IJPP predstavljajo gospodarsko javno službo, ki izvajajo prevoze potnikov in druge storitve znotraj sistema integriranega javnega potniškega prometa IJPP. Javni potniški promet JPP organizira prevoz potnikov v medkrajevnem linijskem avtobusnem prometu, železniškem prometu in mestnem avtobusnem prometu.

Prevoz potnikov znotraj sistema JPP se izvaja na linijah po voznem redu, s prevoznimi sredstvi, ki so vključena v IJPP in v skladu s splošnimi prevoznimi pogoji, ki jih določi upravljavec sistema IJPP in veljajo za celoten sistem IJPP. Prevoznik, ki ima z upravljavcem IJPP sklenjeno pogodbo o izvajanju prevozov v sistemu IJPP, je dolžan upoštevati tudi ostale pogoje prevoza, Standard IJPP in ostale predpise, ki jih izda upravljavec IJPP in so sestavni del sistema IJPP. Vsi prevozni pogoji, ki jih izda prevoznik in so namenjeni uporabi v sistemu IJPP, morajo biti izdani v soglasju z upravljavcem IJPP in z upravljavcem IJPP tudi predhodno usklajeni.

Izdaja kartic ostane v domeni vsakega prevoznika. V dogovoru z upravljavcem IJPP, se znotraj CGP-ja prevoznika uvrsti tudi logo IJPP. Seveda morajo biti kartice prevoznika tipa Mifare DESFire EV1 8K.

Prevoznik ne more izdati prevoznih pogojev, ki so v nasprotju s prevoznimi pogoji, ki jih izda upravljavec IJPP. V primeru nejasnosti ali neusklajenosti prevoznih pogojev prevoznika in upravljavcem IJPP prevladajo prevozni pogoji, ki jih izda up upravljavec IJPP.

Naloga prevoznikov znotraj sistema IJPP so:

- prevoz potnikov

Varen prevoz potnikov po naprej določenih poteh in časovnem terminu. Potnik pri tem mora upoštevati Splošne in posebne pogoje prevoza IJPP in vse pogoje prevoza.

- označevanje režima

Prevozniki morajo zagotoviti označevanje režima vožnje vozil za vsa vozila, ki sodelujejo pri prevozu potnikov znotraj omrežja IJPP.

- zagotovitev geografske pozicije

Prevozniki morajo zagotoviti geografsko pozicijo vozila v območju sistema IJPP

- validacija kart

## DS 4: Integracija IS sistemov prevoznika

---

Prevoznik znotraj IJPP sistema poskrbi za validacijo potnikov pri vstopu na vozilo. Validacija se izvede na validacijski napravi (validator), ki je nameščena na prevoznem sredstvu ali pri sprevodniku/vozniku.

- Zagotoviti prodajno shemo

Prevoznik zagotovijo organizirano mrežo stacionarnih prodajnih mest (blagajne) na katerih je možen nakup kart IJPP. Z upravljavcem IJPP se prevoznik dogovori, katera svoja prodajna mesta bo vključil v prodajno mrežo IJPP. Na prodajnih mestih mora prevoznik zagotoviti:

- Izdajanje kartic IJPP
- Nalaganje aplikacije IJPP na novo ali obstoječo kartico
- Nalaganje produktov- vozovnic na IJPP kartico

- Poročanje o prodaji in izdaji kartic

Naloga prevoznika je tudi redno poročanje o prodaji produktov IJPP na celotni terminalski opremi, ki je vključena v sistem IJPP. Poročanje je lahko izvedeno v on-line načina ali pa tudi v off-line načinu. Pri off-line načinu je točno določen čas, v katerem se mora poročanje o prodaji in izdaji kartic izvesti na zaledni sistem.

Transakcijski podatki, ki se prenašajo v sistemu IJPP morajo biti kriptirani s čimer se zagotovi zaupnost in celovitost informacij.

- Poročanje o validaciji kartic

Naloga prevoznika je tudi redno poročanje o validacijah IJPP kartic na celotni svoji terminalski opremi za validacijo IJPP produktov na vozilih, ki so vključena v sistem IJPP. Poročanje je lahko izvedeno v on-line načina ali pa tudi v off-line načinu. Pri off-line načinu je točno določen čas, v katerem se mora poročanje o validaciji IJPP kartic izvesti na zaledni sistem.

Transakcijski podatki, ki se prenašajo v sistemu IJPP morajo biti kriptirani s čimer se zagotovi zaupnost in celovitost informacij.

## 1.9 UPRAVLJAVEC SISTEMA

Upravljavec sistema IJPP je nadzorni organ nad celotnim sistemom IJPP in skrbi za upravljanje s celotnim sistemom. Upravljavec mora uporabnikom v IJPP sistemu zagotoviti Integracijo storitev javnega potniškega prometa, ki potujejo iz enega kraja v drug kraj, na podlagi enotne vozovnice. Pri tem mora omogočiti prestopanje iz enega prevoznega sredstva na drugega, ne glede na vrsto sredstva ali prevoznika, pri čemer pa morajo biti navezave prevozov med prevoznimi sredstvi časovno usklajene.

#### DS 4: Integracija IS sistemov prevoznika

---

Upravljavec sistema IJPP skrbi za celoten nadzor in delovanje sistema na različnih nivojih in sicer upravljanje sistema JPP, voznih redov, vozovnic, pogojev prevoza, prodajnih in validacijskih mest, informiranja potnikov, financiranja in informacijske podpore delovanju in upravljanju sistema vozovnic. Prav tako skrbi za integriran prostorski razvoj sistema (prestopne točke, oprema postaj in postajališč,...).

V nadaljevanju so naštetе poglavitne naloge upravljavca sistema IJPP, ki se nanašajo na informacijski sistem v IJPP:

- Upravljanje sistema

Upravljavec sistema skrbi za nemoteno delovanje integriranega javnega potniškega prometa. Prav tako skrbi za predloge spremembe samega sistema.

- Organizacija in koordinacija

Upravljavec skrbi za organizacijo in koordinacijo integriranega javnega potniškega prometa ter skrbi za njegovo nemoteno delovanje.

- Prodajna mreža

Upravljavec sistema IJPP se skupaj s prevozniki dogovori o vključitvi prodajnih mest prevoznikov v prodajno mrežo IJPP, ki omogoča izdajo, inicializacijo in nakup IJPP produktov.

- Vozni redi

Upravljavec sistema skrbi za organizacijo in vodenje voznih redov znotraj sistema IJPP. Upravljavec ima kadarkoli možnost do spremembe voznih redov znotraj sistema.

- Tarife

Tarife predstavljajo kombinacije cen vrste vozovnic, statusov in tarifnih razredov, ki se s časom spreminjajo in so določene s strani upravljavca sistema IJPP. Upravljavec sistema IJPP ima možnost kadarkoli možnost do odločitve, da posamezne parametre (tarifne razrede, produkte, statute) vključi, izključi ali pa spremeni v tarifi IJPP.

- Črne liste

Upravljavec vodi tudi črno listo kartic sistema IJPP. Upravljavec ima kadarkoli možnost da posamezne kartice IJPP uvrsti ali odstrani iz seznama črnih list.

- Izdajanje kartic IJPP

Upravljavec sistema IJPP ima pravico izdajanja kartic IJPP, ki se bodo uporabljale znotraj sistema. Upravljavec omogoča izdajanje kartic z aplikacijo na novih ali že obstoječih in uporabljenih karticah znotraj sistema IJPP.

- Inicializacija kartic IJPP

#### DS 4: Integracija IS sistemov prevoznika

---

Inicializacija kartice je v naprej določen postopek odobren s strani u upravljavca sistema. Inicializacija kartice se lahko izvaja na novih IJPP karticah ali karticah različnih prevoznikov, ki se trenutno že uporabljajo v različnih sistemih, kateri bodo postali del IJPP sistema.

- Namestitev varnostnih ključev

Upravljavca sistema prav tako zagotavlja varno namestitev varnostni ključ na različno terminalsko in drugo opremo znotraj IJPP sistema, s katero se zagotovi varno komunikacijo kartic IJPP s terminalsko opremo (prodaja, validacija, kontrola) in dostop do podatkov, ki so zapisani na kartici IJPP. Varnostni ključ so v obliki programske kode zapisani v varnostnem modulu v obliki modulu SIM kartice (SAM modul).

- Distribucija varnostnih ključev

Upravljavca prav tako poskrbi za varno distribucijo varnostnih ključev znotraj sistema IJPP.

- Reklamacijski postopek

Upravljavca je prav tako zadolžen za reševanje reklamacijskih postopkov. Vsak vložen reklamacijski postopek je dolžan le tega obravnavati v skladu s predpisanim postopkom reklamacije pri prevozniku.

- Klicni center in podpora uporabnikom

Upravljavca je prav tako zadolžen za podporo klicnega centra in podporo uporabnikom, ki bo omogočal predvsem podporo potnikom in podporo prevoznikom.

Produkte IJPP definira upravljavca IJPP. Kot je že ustaljena praksa se bodo uvedle tako terminske vozovnice (dnevna, mesečna, letna,...), kakor tudi količinske vozovnice (npr. za 10, 20 in 30 voženj). Vsak produkt ima svojo veljavnost in število voženj, ki jih lahko potnik opravi z njimi. Možne so tudi kombinacije conskih produktov z medkrajevnimi produkti, kjer se predvideva, da se bo ob dani kombinaciji conski produkt prodal z določenim popustom. Vsi podatki ki zajemajo produkte (naziv, veljavnost, cena, itd.) so zajeti v tarifi IJPP. Možna je tudi implementacija tarife, ki zajema še izbrane poenotene dodatne produkte IJPP za prevoz prtljage in izvajanje nadstandardnih prevoznih storitev. Prevozniki te dodatne produkte IJPP izvajajo v obsegu svojih tehnično-tehnoloških zmogljivosti. Kdaj in kako se bodo dodatni produkti vpeljali v sistem IJPP, je odvisno od upravljavca IJPP.

Upravljavca IJPP je zakonsko zavezan k varovanju osebnih podatkov zato so vsi postopki, ki obravnavajo skladni z zakonskimi omejitvami. V splošne prevozne pogoje se vključijo ustrezni popravki in dopolnila, ki potnike obvestijo o uporabi in prenosu podatkov do upravljavca IJPP. Upravljavca IJPP zagotavlja, da se podatki glede na status, ki ga ima potnik v sistemu IJPP, po določenem obdobju skladno z zakonodajo ireverzibilno anonimizirajo.

V sistemu IJPP morajo biti osebni podatki, ki se pošiljajo med različnimi sistemi, pretvorjeni v psevdonim oz. identifikacijsko številko.

## DS 4: Integracija IS sistemov prevoznika

---

Upravljalavec IJPP bo vzpostavil ustrezno več nivojsko shemo pomoči uporabnikom in prevoznikom, regulacija in delovanje slednje bo določeno s pravilnikom. Reklamacije se bodo izvajale po več nivojskem sistemu, tako bo moral potnik, ki bo želel izvesti reklamacijo na drugem prodajnem mestu, kakor je bil izveden nakup, problem oziroma težavo najprej prijaviti na klicni center. V veliki večini bo lahko dano težavo razrešil že sam klicni center. Ker pa klicni center NIMA vpogleda v transakcije sistema IJPP in so za dano težavo/reklamacijo potrebne dodatne informacije, se vključi drugi nivo. Osebe, ki so zadolžene za drug nivo, so zaposlene pri upravljavcu IJPP in imajo omejen vpogled v podatke zalednega sistema IJPP. Če je potrebna še dodatna pomoč, se vključi tretji nivo, itd.

### 1.10 INFORMACIJSKI SISTEM IJPP

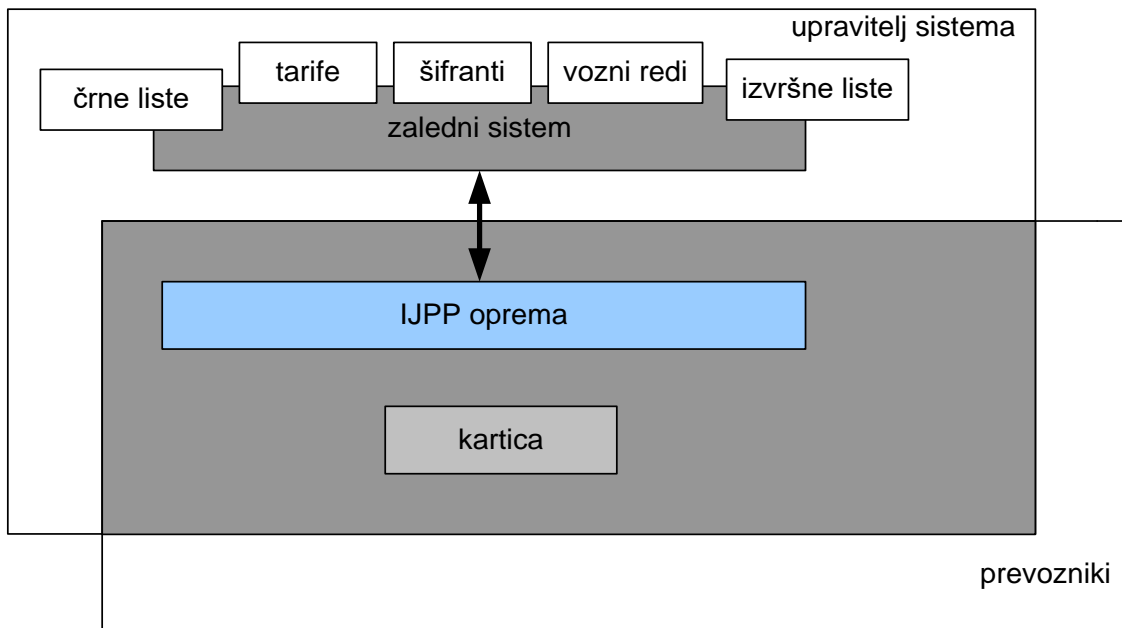
Ministrstvo za infrastrukturo želi z uvedbo projekta integriranega javnega potniškega prometa v RS (v nadaljevanju projekt IJPP) uvesti enotno vozovnico ter poenotenje voznih redov na ravni avtobusnega in železniškega prometa znotraj javnega avtobusnega železniškega omrežja v RS. Za doseg tega cilja je potrebno v celotnem omrežju nadgraditi informacijski sistem, ki že obstaja ali pa ga je potrebno postaviti na novo.

Informacijski sistem omrežja znotraj integriranega javnega potniškega prometa v RS bo pokrival več področij. Osnovni gradniki celotnega informacijskega sistema so:

- Brezstična kartica
- Terminalska oprema na vozilih
- Terminalska oprema na prodajnih mestih
- Zaledni sistem

Naslednja slika prikazuje osnovne gradnike celotnega informacijskega sistema IJPP v povezavi z upravljalavcem sistema in prevozniki.

#### DS 4: Integracija IS sistemov prevoznika



SLIKA 1.7: OSNOVNA IN POENOSTAVljena shema informacijskega sistema v povezavi z upravljavcem sistema in prevozniki

Vsi osnovni gradniki predstavljajo celotno informacijsko omrežje, ki zagotavlja uporabnikom-potnikom prijazno koriščenje storitev kot so nakup kart za potovanje od začetne do končne postaje, validiranje na prevoznih sredstvih ter ustrezno informiranje v času potovanja ali izven njega.

Prav tako sistem mora zagotoviti kakovostne storitve in prijazno delovanje sistema za prevozniki, ki upravljajo transport potnikov po v naprej določenih poteh tako na cestnih kot železniških odsekih. Prevozniki tako v železniškem kot avtobusnem javnem potniškem prometu bodo imeli možnost prijazne uporabe informacijskega sistema, ki ga bodo uporabljali pri validiranju potnikov kot samo informiranje.

Informacijski sistem bo omogočal celoten nadzor omrežja za upravljavca celotnega sistema. Ta bo imel popoln nadzor nad samim sistemom, ki mu bo omogočal spreminjanje voznih redov, izdaje kartic, dodeljevanje nove ali nadgradnjo obstoječe opreme na terminalih ter ostalih nalog, ki so definirane v nadaljevanju tega dokumenta.

Od terminalske opreme, ki je vključena v sistem IJPP, se zahteva aktivna povezava z zalednim sistemom IJPP. Hkrati mora vsa oprema, ki se vključi v sistem IJPP skozi verifikacijski postopek in delovati skladno s standardom IJPP. Kadar se izkaže, da sistem pri posameznem prevozniku ne deluje skladno s standardom IJPP, se bo le to ustrezno sankcioniralo.

Operativni sistem IJPP bo zasnovan kot offline sistem, kar pomeni, da za direktno izvajanje procesov (nakupi vozovnic, validacije) ne bo potrebna direktna online povezava s sistemom IJPP in bodo lahko mobilne naprave te procese izvajale avtonomno. Na ta način se zagotavlja robustnost sistema v primeru tehničnih težav in trenutkov, ko direktna povezava ni na voljo.



Standardu IJPP definira časovne parametre, ki bodo določali kdaj in kako je potrebno v sistem IJPP poročati dogodke in kakšnih intervalih je potrebno iz sistema v mobilnih napravah posodabljanje operativne podatke (črne liste, tarife, ...).

V določenih primerih, ki jih definira standard, bo potrebno za uspešno izvedbo določenih procesov (npr. za preverjanje, če je kartica na mehki črni listi ali za prevzemanje že kupljenih vozovnic) vzpostaviti direktno online povezavo med terminalske opremo in zalednim sistemom.

### 1.10.1 BREZSTIČNA KARTICA

Brezstična kartica predstavlja osnovni del informacijskega sistema, ki omogoča validiranje točno določenih produktov znotraj sistema IJPP s strani upravljalca celotnega sistema. Brezstična kartica z vsebovano brezstično kartično aplikacijo predstavlja virtualno okolje na elektronski pametni kartici, ki omogoča uporabo datotek in podatkov, ki so shranjeni na elektronski kartici. Brezstična kartica lahko ima naloženih več kartičnih aplikacij, poleg enotne slovenske transportne aplikacije (IJPP) še aplikacije drugih prevoznikov.

IJPP aplikacija se lahko naloži že na obstoječe kartice in sicer kot ločena aplikacija na brezstični kartici ali pa na novo kartico. Kadar prevoznik uporablja v svojem sistemu brezstično kartico Mifare-DESFire EV1 8K, je predvideno, da se lahko zraven aplikacije prevoznika, na isto kartico naloži tudi aplikacija IJPP. IJPP aplikacija bo imela svoj ID (unique), ki se bo ločevala od drugih aplikacij prevoznikov na kartici znotraj IJPP sistema. IJPP maksimalna velikosti aplikacije IJPP znaša 4kB, ki se doda že na obstoječo brezstično kartico prevoznika znotraj sistema IJPP (ob predpostavki, da je na kartici vsaj 4kB prostora in da na kartici še ni aplikacije z enakim ID kot je določena za IJPP aplikacijo).

Potnik na točno določenih prodajnih mestih znotraj IJPP sistema kupi karte za vožnjo, ki se nato preko brezstičnega čitalca naložijo v enotno slovensko transportno aplikacijo na brezstični kartici potnika. Le ta se ob prihodu validira na čitalcu, ki je nameščen na prevoznem sredstvu. Validacijo lahko fizično opišemo kot približanje brezstične kartice k čitalcu na razdaljo nekaj centimetrov. Pri tem čitalec pridobi podatke iz kartice o tipe kartice ter o naloženih kartah na brezstični kartici. Čitalec, ki je povezan z zalednim sistemom, na podlagi določenih pravil določi rezultat validacije, ki je lahko uspešna ali pa neuspešna. V primeru uspešne validacije lahko potnik začne ali nadaljuje pot na izbranem prevoznem sredstvu, v primeru neuspešne transakcije pa žal potnik nima ustrezne karte za izbrano vozilo.

V nadaljevanju dokumenta so definirani vsi potrebni parametri brezstične kartice, ki se bo uporabljala znotraj sistema IJPP in sicer:

- Fizične lastnosti kartice
- Komponente na kartici
- RF vmesnik
- Pomnilnik

## DS 4: Integracija IS sistemov prevoznika

---

- Varnostni mehanizmi
- Grafična podoba IJPP kartice

Pred samo izdajo kartic je potrebno da upravljaivec sistema opravi verifikacijo brezstičnih kartic, ki se bodo uporabljale znotraj IJPP sistema. Verifikacija je proces pri katerem se preverja ali brezstična kartica ustreza zahtevam, ki so bile postavljene v standardu. V nadaljevanju dokumenta so opisani vsi potrebi postopki za verifikacijo brezstične kartice.

Pri komunikaciji s kartico poteka avtentikacija po standardu Mifare DESFire (triple des). Komunikacija zalednega sistema s terminalsko opremo, pa poteka na osnovi varnostnega elementa v terminalski opremi.

### 1.10.2 TERMINALSKA OPREMA NA VOZILIH

Za validiranje potnikov z brezstičnimi karticami je potrebno vozila tako na avtobusnem prometu kot železniškem prometu nadgraditi ali pa opremiti z terminalsko opremo, ki bo omogočala validacije brezstične IJPP kartice. Terminalska oprema za validacijo na vozilih mora podpirati brezstično komunikacijo med čitalcem terminalske opreme in kartico v neposredni bližini. Prav zaradi nujno potrebne neposredne bližine med napravami je prav tako potrebno zagotoviti varnost podatkov, ki se prenašajo med kartico in čitalcem.

Terminalska oprema se uporablja, kjer vsebuje čitalec brezstičnih kartic tudi na prodajnih mestih, ki omogočajo nakup različnih kart-produktov za sistem IJPP. V primeru nakupa se na prodajnem mestu izbere karto ali produkt potnika, ki nato prisloni brezstično kartico k terminalu na katero se naloži karta ali izbran produkt, ki ga potnik kasneje validira na vozilu na izbrani poti.

V nadaljevanju dokumenta so podani zahteve za terminalsko opremo na vozilih in sicer:

- Lastnosti čitalnika
  - RF antena
  - NFC čip
  - Nosilni procesor
  - Bralnik SAM modulov
  - Varnostni element
- Komunikacija z ukazi
  - Komunikacijski ukazi ISO/IEC 14443-3
  - Komunikacijski ukazi ISO/IEC 14443-4
  - Ukaz RATS in ATS
  - Ukaz PPS request in PPS response
  - DESFire komunikacijski ukazi (varnostni ukazi, ukazi za upravljanje s kartico, aplikacijami in podatki)

## DS 4: Integracija IS sistemov prevoznika

---

Pred samo izdajo terminalske opreme za vozila je potrebno, da upravljavec sistema skupaj z dobaviteljem opravi verifikacijo terminalske opreme, ki se bo uporabljale znotraj IJPP sistema. Verifikacija je proces, pri katerem se preverja ali terminalska oprema ustreza zahtevam, ki so bile postavljene v standardu. V nadaljevanju dokumenta so opisani vsi potrebi postopki za verifikacijo terminalske opreme.

### 1.10.3 TERMINALSKA OPREMA NA PRODAJNIH MESTIH

Za mesta prodaje znotraj IJPP sistema je potrebno prodajna mesta tako na avtobusnem prometu kot železniškem prometu nadgraditi ali pa opremiti z terminalske opremo, ki bo omogočala tako inicializacijo, nalaganje IJPP aplikacije in nakup produktov na brezstično IJPP kartico. Terminalska oprema na prodajnih mestih mora podpirati brezstično komunikacijo med čitalcem terminalske opreme in kartico v neposredni bližini. Prav zaradi nujno potrebne neposredne bližine med napravami je prav tako potrebno zagotoviti varnost podatkov, ki se prenašajo med kartico in čitalcem.

Terminalska oprema se uporablja, kjer vsebuje čitalec brezstičnih kartic tudi na prodajnih mestih, ki omogočajo nakup različnih kart-produktov za sistem IJPP. V primeru nakupa se na prodajnem mestu izbere karto ali produkt potnika, ki nato prisloni brezstično kartico k terminalu na katero se naloži karta ali izbran produkt, ki ga potnik kasneje validira na vozilu na izbrani poti.

V nadaljevanju dokumenta so podani zahteve za terminalske opremo na prodajnih mestih in sicer:

- Lastnosti čitalnika
  - RF antena
  - NFC čip
  - Nosilni procesor
  - Bralnik SAM modulov
  - SAM modul
- Komunikacija z ukazi
  - Komunikacijski ukazi ISO/IEC 14443-3
  - Komunikacijski ukazi ISO/IEC 14443-4
  - Ukaz RATS in ATS
  - Ukaz PPS request in PPS response
  - DESFire komunikacijski ukazi (varnostni ukazi, ukazi za upravljanje s kartico, aplikacijami in podatki)

Pred samo izdajo terminalske opreme za prodajna mesta je potrebno, da upravljavec sistema skupaj z dobaviteljem opravi verifikacijo terminalske opreme, ki se bo uporabljale znotraj IJPP sistema. Verifikacija je proces, pri katerem se preverja ali terminalska oprema ustreza zahtevam, ki so bile postavljene v standardu. V nadaljevanju dokumenta so opisani vsi potrebi postopki za verifikacijo terminalske opreme.

## Mobilna prodaja in stacionarna prodaja

Znotraj sistema IJPP je potrebno zagotoviti dva načina prodaje in sicer:

- Stacionarna prodaja  
Pri stacionarnem načinu prodaje smo vezani na eno lokacijo prodaje IJPP produktov
- Mobilna prodaja  
Pri mobilnem načinu prodaje prodajalec ni vezan na eno lokacijo prodaje IJPP produktov in lahko prodajo izvaja na različnih lokacijah

## Komunikacije med čitalnikom in kartico

Postopki komunikacije med čitalnikom in kartico se sprožijo kadar prislonimo kartico v neposredno bližino čitalnika. Komunikacija se izvaja točno po določenem standardu, ki je opisan v nadaljevanju dokumenta. V nadaljevanju dokumenta so predstavljeni naslednje operacije med brezstično kartico in čitalcem:

- operacije nakup vozovnice
- operacije validacija vozovnice
- operacije preverjanje vozovnice
- operacije pregledovanja informacij vozovnic na kartici
- operacije stornacije vozovnice
- operacije ob popolnem vračilu vozovnice
- operacije ob delnem vračilu vozovnice
- operacije ob blokadi vozovnice
- operacije ob deblokadi vozovnice

Za uspešno komunikacijo med brezstično kartico in čitalcem je potrebno točno definirana struktura kartice, kar pomeni točno določena pravila zapisa podatkov na brezstično kartico. To omogoča pravilno in nemoteno branje kartic s strani čitalcev. V nadaljevanju dokumenta so podane definicije zapisa podatkov na brezstično kartico in sicer:

- Osebni podatki
- Tarifni razredi
- Produkti
- Relacije
- Zgodovina

V procesu verifikacije tako brezstične kartice in terminalske opreme se izvede tudi verifikacija komunikacije med brezstično kartico in čitalnikom terminalske opreme.

#### 1.10.4 ZALEDNI SISTEM

Zaledni sistem predstavlja osrednji del informacijskega sistema IJPP omrežja, ki omogoča upravljavcu sistema celoten vpogled in kontrolo nad informacijskim sistemom. Zalednemu sistemu upravljavca se poročajo vse transakcije vezane na sistem IJPP; inicializacija, nakup, validacija, stornacija, polno vračilo, delno vračilo, blokada, deblokada, itd. Točne specifikacije posamezne transakcije so zajete v nadaljevanju dokumenta.

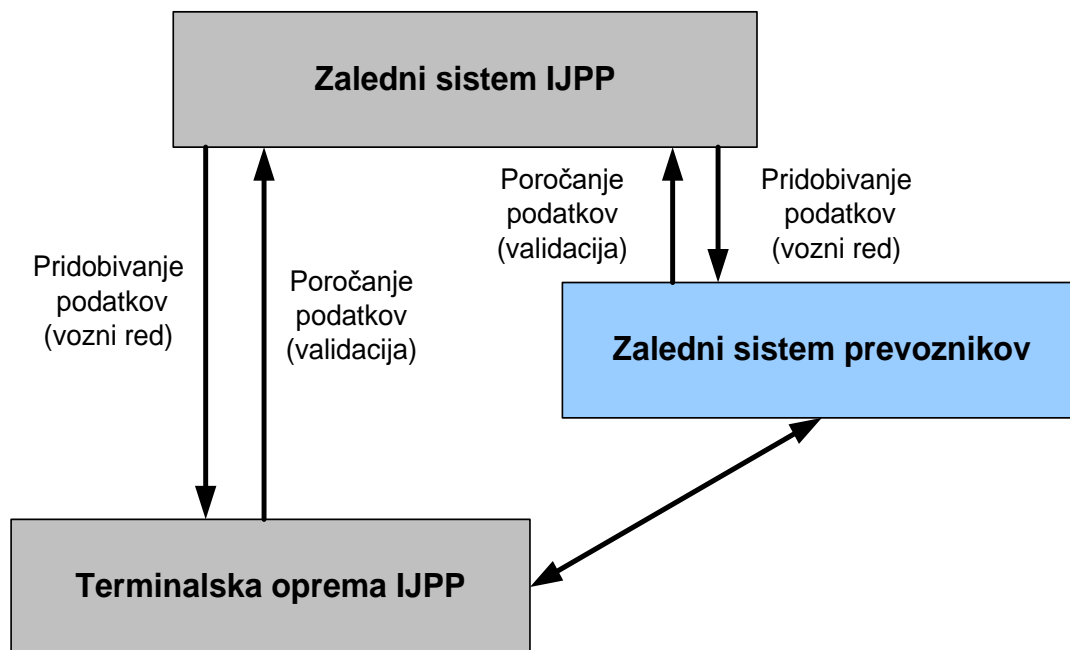
Zaledni sistem omogoča komunikacijo z naslednjimi sistemi in komunikacijsko opremo:

- Terminalsko opremo prevoznikov
- Zalednim sistem prevoznikov

Terminalska oprema komunicira z zalednim sistemom preko web servisov, ki so specificirani v standardu IJPP. Glede na tip komunikacije z zalednim sistemom ločimo dva tipa in sicer:

- Pridobivanje podatkov IJPP
- Poročanje podatkov IJPP

Slika prikazuje shematski prikaz povezave zalednega sistema z preostalo komunikacijsko opremo znotraj IJPP omrežja ter samim pretokom informacij.



## DS 4: Integracija IS sistemov prevoznika

SLIKA 1.8: PONAŽORITEV POVEZAVE MED ZALEDNIM SISTEMOM IJPP, ZALEDNIM SISTEMOM PREVOZNIKOV IN TERMINALSKO OPREMO IJPP

V primeru terminalske opreme IJPP le ta izvaja direktno povezavo na IJPP zaledni sistem. V primeru že obstoječega zalednega sistema prevoznika, lahko ves prenos podatkov poteka preko zalednega sistema prevoznika (prenos podatkov med IJPP zalednim sistemom in terminalsko opremo prevoznika).

### 1.10.5 PRIDOBIVANJE PODATKOV IJPP

Poleg komunikacije zalednega sistema z terminalsko opremo ali z drugimi zalednimi sistemi ponudnikov storitev zaledni sistem omogoča še pridobitev osnovnih informacij, ki so potrebne za nemoteno delovanje sistema in sicer:

- Vozni redi
- Tarife
- Črne liste
- Izvršne liste
- Šifranti
- Pravila za prestopanje v mestnem prometu

#### Vozni redi

Vozni red se izdaja s strani upravljalca sistema IJPP. Ta je namenjen tako potnikom, kot tudi prevoznikom in vsebuje vse potrebne informacije za potovanje. Trenutno se v RS uporablja več različnih prometno tehničnih podlag za izdelavo voznih redov (daljinarjev). Tako se za medkrajevni avtobusni daljinar uporablja podatkovni sistem imenovan AVRIS. Poleg tega sistema poznamo še daljinar Slovenskih železnic, ki je ločen od medkrajevnega avtobusnega daljinarja. Prav tako so ločeni tudi daljinarji, ki so namenjene mestnemu prometu.

Vozni red v integriranem javnem potniškem prometu bo temeljil na osnovi Nacionalnega voznega reda, kjer so zbrani vozni redi 14. sistemov mestnega prometa. Izhodiščni vir za 9 mestnih občin (Idrija, Koper, Krško, Maribor, Murska Sobota, Nova Gorica, Piran, Postojna in Velenje) je bila podatkovna baza postajnih točk BUSO. Na osnovi teh voznih redov so bili pripravljeni itinerarji na podlagi postaj in postajališč znotraj mestnega prometa. Vozni red IJPP tako za vsako linijo vsebuje podatke o vseh zaporednih postajališčih (itinerar) in časih prihodov in odhodov s postajališča v nekem dnevu. Vozni red prav tako sovпада s koledarjem voženj in velja za določen režim ali koledar vožnje (npr. vozni red vlaka na neki liniji ob delavnikih).

## Tarifa

Tarifa je podatkovna struktura, ki določa pravila za prodajo in validacijo vozovnic v transakcijskem sistemu IJPP. Pravila za prodajo in validacijo vozovnic se nanašajo na vse igralce, ki sodelujejo v sistemu IJPP. To so končni uporabniki in podjetja, ki tržijo vozovnice IJPP. Podjetja lahko prevzamejo dve različni vlogi v sistemu IJPP to sta vlogi ponudnika storitev in trgovca. Za končne uporabnike je cena vozovnic najpomembnejši podatek iz tarife. V osnovi lahko tarifo opišemo s tridimenzionalnim podatkovnim poljem, ki določa cene. Prvo dimenzijo predstavlja vrsta vozovnic, drugo dimenzijo predstavljajo statusi in tretjo dimenzijo predstavljajo tarifni razredi oziroma cone.

Tarife predstavljajo kombinacije cen vrste vozovnic, statusov in tarifnih razredov, ki se s časom spreminjajo. Zato so tarife definirane le za določena časovna obdobja. Ponudnik storitev ima lahko le eno aktivno tarifo. Zaradi tega moramo tarife unikatno identificirati. Tarifa, ki je trenutno aktivna se pošlje na terminale v naprej predpisani obliki. Podrobnejši opis strukture tarife, ki se pošlje na terminale je definiran v poglavju: Struktura tarife na terminalih.

Na izbranih prodajnih mestih lahko onemogočimo prodajo določenih vozovnic, medtem ko na preostalih prodajnih mestih, ki niso izbrana, prodajo dovolimo. To storimo z definiranjem pravil za tarife. S pravili za tarife eksplicitno določimo katere vrste vozovnic, statusi in tarifni razredi v tarifi so na voljo za prodajo na določenih lokacijah. Podrobnejši opis delovanja pravil za tarife je predstavljen v poglavju: Pravila za tarife.

V nadaljevanju dokumenta so definicije osnovnih delov tarife in sicer:

- **Cenovni model tarife**  
Opisuje cenovni model tarife, ki ga predstavljajo vrste vozovnic s pripadajoči parametri za opis le teh. Prav tako definira statute z unikatnim identifikatorjem statusa za določenega ponudnika storitev in različne tarifne razrede z vnaprej definiranimi parametri.
- **Tarife in ponudniki storitev**  
Ta del definira tarife za posamezne ponudnike storitev znotraj IPP omrežja za točno določena prodajna mesta ali mesta validacij. Pošiljanje tarif na izbrana prodajna mesta se izvedejo z nastavitvijo distribucijskih pogodb med ponudniki storitev in trgovci.
- **Pravila za tarife**  
Osnovna ideja pravil za tarife je v maskiranju cen za posamezne vrste vozovnic, statute in tarifne razrede. Pogosto želimo onemogočiti prodajo določenih vrst vozovnic glede na segment v prodajni verigi, kot so terminali, prodajna mesta, podjetja, ipd. Takšno onemogočanje prodaje dosežemo s pravili za tarife.
- **Podatkovna struktura tarif na terminalih**  
Terminali na katerih se lahko vozovnice kupijo, uporabijo ali preverijo, potrebujejo vse podatke iz tarife. Ker so terminali pogosto omejeni s pomnilnikom in z računsko zmogljivostjo, zaledna

## DS 4: Integracija IS sistemov prevoznika

---

stran pripravi kompaktno in preprosto podatkovno strukturo, ki se pošlje na terminale. Zgradba podatkovne strukture tarife za terminale je prikazana na sliki

- Upravljanje s tarifami

V tem poglavju bomo predstavili postopke upravljanja s tarifami kot so: kreiranje nove tarife, kreiranje pravil in napotke pri upravljanju s tarifami

### Črne liste

Črne liste predstavljajo seznam kartic, ki so bile umaknjene iz sistema IJPP iz različnih razlogov (npr. zloraba). Seznam kartic se vodi na zalednem sistemu in se prenaša na terminalsko opremo. Naloga terminalske opreme je, da ob prislonitvi IJPP brezstične kartice preverijo ali je prislonjena kartica na seznamu črnih list. V primeru, da kartica na seznamu črnih list se le to onemogoči in se na terminalu potniku sporoči, da je njegova kartica na seznamu črnih list. Nadaljnje reševanje kartice in umik kartice s črnih list ureja upravljavec sistema.

### Izvršne liste

Izvršne liste predstavljajo seznam kartic z dopolnilnim parametrom, ki se bo prenašal iz zalednega sistema na terminalsko opremo. Naloga terminalske opreme je, da ob prislonitvi IJPP brezstične kartice preverijo ali je prislonjena kartica na seznamu izvršnih list. V primeru, da kartica na seznamu izvršnih list se iz izvršne liste prebere še dodan parameter, ki sporoči določeno operacijo, ki se mora izvesti za prislonjeno kartico (npr. online preverjanje, onemogočanje produktov,...) .

### Šifranti

V sistemu IJPP se uporablja več različnih šifrantov (šifranti postaj, šifranti prevoznikov,...), ki se bodo prenašali iz zalednega sistema na terminalsko opremo prevoznikov.

### Pravila za prestopanje v mestnem prometu

Struktura podatkov in pravil za prestopanje v mestnem prometu.

#### 1.10.6 POROČANJE PODATKOV IJPP

Zaledni sistem prav tako skrbi za vodenje, pregled in sprejemanje podatkov iz terminalske opreme: pogledjmo nekaj najbolj osnovnih:

- Poročanje o validaciji
- Poročanje o izdaji in inicializaciji IJPP kartic
- Poročanje o nakupu produktov
- Izvajanje servisnih klicev



### **Poročanje o validaciji**

Poročanje o validaciji predstavljajo podatke o sami validaciji potnika na vozilu v sistemu IJPP, ki se nato v določenem časovnem okvirju sporočijo/prenesejo zalednemu sistemu IJPP sistema. Ob sami validaciji potnika se med kartico in terminalom izmenjajo določeni podatki za izvedbo same validacije in pa podatki, ki so potrebni za sporočanje validacije zalednemu sistem. Podatki o validaciji zajemajo vse potrebne informacije o potniku (npr. številka kartice) kot tudi podatki o sami terminalski napravi na kateri je bila validacija izvedena (npr. terminal ID). Definicija poročanja o validacijah so definirane v nadaljevanju dokumenta.

### **Poročanje o izdaji in inicializaciji IJPP kartic**

Poročanje o izdaji in inicializaciji IJPP kartic predstavljajo podatke o sami izdaji in inicializaciji IJPP kartic potnika na prodajnem mestu v sistemu IJPP, ki se nato v določenem časovnem okvirju sporočijo/prenesejo zalednemu sistemu IJPP sistema. Ob inicializaciji IJPP kartic se med kartico in terminalom izmenjajo določeni podatki za izvedbo same inicializaciji in pa podatki, ki so potrebni za sporočanje validacije zalednemu sistem. Podatki o inicializaciji zajemajo vse potrebne informacije o potniku (npr. številka kartice) kot tudi podatki o sami terminalski napravi na kateri je bila inicializaciji izvedena (npr. terminal ID). Definicija poročanja o inicializaciji in izdaji so definirane v nadaljevanju dokumenta.

### **Poročanje o nakupu produktov**

Poročanje o nakupu produktov predstavljajo podatke o samem nakupu produktov na prodajnem mestu v sistemu IJPP, ki se nato v določenem časovnem okvirju sporočijo/prenesejo zalednemu sistemu IJPP sistema. Ob nakupu produktov se med kartico in terminalom izmenjajo določeni podatki za izvedbo nakupa produktov in pa podatki, ki so potrebni za sporočanje validacije zalednemu sistem. Podatki o nakupu produktov zajemajo vse potrebne informacije o potniku (npr. številka kartice), kupljenih produktih kot tudi podatki o sami terminalski napravi na kateri je bil nakup produktov izveden (npr. terminal ID). Definicija poročanja o nakupu produktov so definirane v nadaljevanju dokumenta.

### **Izvajanje servisne storitve**

Servisna storitev je poizvedovanje terminalske opreme o novih verzijah tarifnega sistema, črnih list, izvršnih list in voznega reda. Na točno določeno periodo terminalska oprema izvede servisni klic s katerim sporoči na zaledni sistem ID tarifnega sistema, črnih list, izvršnih list in voznega reda. V primeru, da so na zalednem sistemu že nove verzije teh podatkovnih struktur, se le te prenesejo na terminal. S tem zagotovimo, da ima terminalska oprema vedno nove podatke o voznih redih, tarifah ter posodobljene tako črne kot izvršne liste.

## 1.11 REFERENCE

- [1] Izdelava enotnega daljinarja za integriran javni potniški promet - KONČNO POROČILO, Ministrstvo za infrastrukturo in prostor, Direktorat za promet, Ljubljana, avgust 2012.

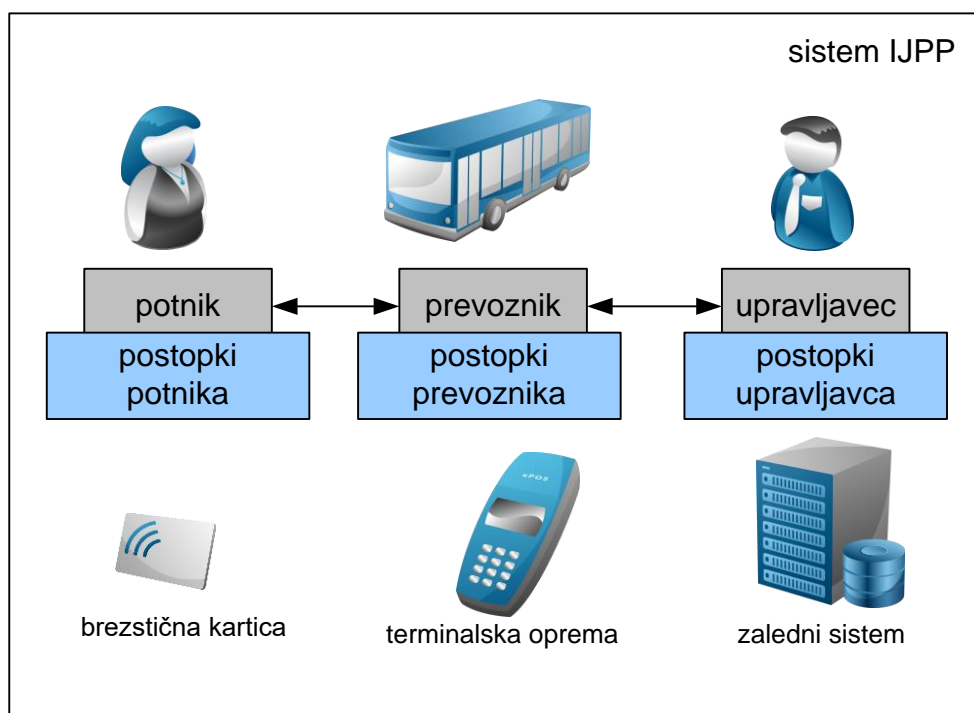
## 2 SISTEM IJPP

Sistem IJPP podpira uvedba enotne vozovnice ter poenotenje voznih redov na ravni avtobusnega in železniškega prometa znotraj RS. Uvedba integriranega javnega potniškega prometa pomeni združitev, nadgradnjo in integracijo različnih že obstoječih sistemov elektronske vozovnice pri prevoznikih v en sistem IJPP, s katerim bo upravljal upravljavec.

Za lažji opis zgradbe sistema dokument najprej opisuje osnovne elemente sistema IJPP ter nato še postopke posameznega elementa znotraj IJPP sistema.

Sistem IJPP je razdeljen na tri osnovne elemente:

- Potnik
- Prevoznik
- Upravljavce



SLIKA 2.1: OSNOVNI ELEMENTI IJPP SISTEMA S PRIPADAJOČIMI POSTOPKI IN PRIPADAJOČO TELEKOMUNIKACIJSKO OPREMO ZNOTRAJ SISTEMA IJPP

V nadaljevanju dokumenta so podane definicije osnovnih elementov IJPP sistema s pripadajočimi postopki znotraj sistema IJPP:

## DS 4: Integracija IS sistemov prevoznika

---

- Postopki potnika
- Postopki prevoznika
- Postopki upravljavca

Posamezni postopki potnika, prevoznika in upravljavca se izvajajo na točno specificirani telekomunikacijski opremi. Potniki uporabljajo brezstično pametno kartico z naloženo aplikacijo IJPP za namene koriščenje prevoza v sistemu IJPP. Prevozniki imajo svoja vozila in prodajna mesta opremljena s terminalsko opremo, ki služi za validacijo in nakup vozovnic potnikom. Upravljavec skrbi za celotno upravljanje s sistemom IJPP, ki s pomočjo orodij za planiranje določi vozne rede, tarifo, črno listo in vseh potrebnih podatkov shranjenih na zalednem sistemu do katerih dostopa terminalska oprema prevoznikov.

### 2.1 POTNIK

Potnik je oseba, ki uporablja sistem IJPP na osnovi IJPP aplikacije, nameščene na brezstični pametni kartici. V nadaljevanju so definirani posamezni postopki potnika znotraj IJPP sistema.

#### 2.1.1 POSTOPKI POTNIKA V SISTEMU IJPP

Definirajmo posamezne postopke potnika znotraj sistema IJPP:

- Inicializacija IJPP kartice
- Nakup IJPP vozovnic
- Validacija potnika pri prevozniku
- Prijava izgube kartice
- Zahteva za duplikat IJPP kartice
- Predaja potekle kartice

##### 2.1.1.1 Inicializacija IJPP kartice

Pred uporabo storitev IJPP mora potnik pridobiti brezstično kartico nad katero se izvede inicializacijski postopek, pri katerem se na kartico naloži aplikacija IJPP. Kartica mora biti skladna z zahtevami v tem standardu. IJPP aplikacija se lahko naloži, ali na novo brezstično kartico, ali na že obstoječe kartice in sicer kot ločena aplikacija. Kadar prevoznik uporablja v svojem sistemu brezstično kartico Mifare-DesFIRE EV1 8K, je predvideno, da se lahko zraven

## DS 4: Integracija IS sistemov prevoznika

aplikacije prevoznika, na isto kartico naloži tudi aplikacija IJPP. IJPP aplikacija bo imela svoj ID (unique), ki se bo ločil od ID drugih aplikacij prevoznikov na kartici znotraj IJPP sistema. IJPP maksimalna velikosti aplikacije IJPP znaša 4kB, ki se doda že na obstoječo brezstično kartico prevoznika znotraj sistema IJPP (ob predpostavki, da je na kartici vsaj 4kB prostora in da na kartici še ni aplikacije z enakim ID kot je določena za IJPP aplikacijo). Pred uporabo mora biti kartica IJPP z naloženo aplikacijo prav tako uspešno registrirana v sistem IJPP. Točno strukturo kartice, strukturo podatkov in pripadajočih datotek definira standard IJPP v poglavju 3.

Na vsaki kartici IJPP je zapisan status, ki je povezan z upravičenostjo do popustov posebnih skupin potnikov in z načinom uporabe kartice IJPP. V sistemu IJPP obstajajo osnovni statusi in dodatni statusi, ki so povezani z izdajo subvencioniranih vozovnic.

Nekaj primerov statusov kartic IJPP:

- Prenosna
- Splošna
- Otroci
- Šolar
- Mladi...

Pogoji za doseganje statusa, ki velja za posebne skupine potnikov, so določeni s Tarifo IJPP in v zakonu, ki ureja izdajo subvencioniranih vozovnic. Kartica je enolično povezana s statusom, kar pomeni, da je lahko na vsaki kartici IJPP hkrati veljaven le en status, razen izjem, ki bodo natančno definirane s strani upravljavca.

Vsak status kartice ima določeno veljavnost statusa. Po poteku veljavnosti statusa, se vsak status avtomatično preoblikuje v status »Splošna«. Vsak status je mogoče obnoviti na osnovi dokazila, na katerem je jasno razviden status, ki ga upravičenec uveljavlja in obdobje veljavnosti. Če je veljavnost statusa na dokazilu neomejena, je veljavnost statusa na kartici omejen z rokom veljavnosti kartice IJPP.

### 2.1.1.2 Nakup IJPP vozovnic

Potnik na točno določenih prodajnih mestih znotraj IJPP sistema kupi karte za vožnjo, ki se nato preko brezstičnega čitalca naložijo v enotno slovensko transportno aplikacijo na brezstični kartici potnika. Nakup določenih vozovnic na kartico je omejen, saj je v sistemu IJPP vpeljan pravilnik, ki določa katere vozovnice lahko kupi potnik s točno določenim tipom kartice (pravice- tarifni razred). Ta pravila so določena v tarifi, ki se nato upoštevajo ob nakupu produktov na prodajnem mestu. Tarifa skupaj s pravili so definirana v nadaljevanju tega dokumenta.

Pri nakupu ločimo tri tipe vozovnic in sicer:

- Vozovnice za mestni promet (zapis cone na kartico)
- Vozovnice za medkrajevni promet (zapis relacije na kartico)
- Vozovnice za kombiniran promet (zapis relacije in cone na kartico)

### 2.1.1.3 Validacija potnika pri prevozniku

Ob prihodu na vozilo potnik opravi validacijski postopek na čitalniku, ki je nameščen na prevoznem sredstvu. Validacijo lahko fizično opišemo kot približanje brezstične kartice k čitalcu na razdaljo nekaj centimetrov. Pri tem čitalec pridobi podatke iz kartice o tipu kartice ter o naloženih vozovnicah na brezstični kartici. Čitalec, ki je povezan z zalednim sistemom, na podlagi določenih pravil določi rezultat validacije, ki je lahko uspešna ali pa neuspešna. Ob validaciji se na terminalski opremi izpišejo tudi osnovni podatki o validaciji in vozovnici ter sproži zvočni signal ki označuje validacijo. V primeru uspešne validacije lahko potnik začne ali nadaljuje pot na izbranem prevoznem sredstvu, v primeru neuspešne transakcije potnik nima ustrezne vozovnice za izbrano vozilo.

### 2.1.1.4 Prijava izgube kartice

V primeru izgube kartice, je potnik dolžan izgubo kartice javiti prevozniku ali na pomoč uporabnikom. Ob prijavi izgube upravljavec vstavi kartico na seznam črnih list in s tem onemogoči nadaljnjo rabo izgubljene kartice. S tem se onemogoči nepooblaščen raba izgubljene kartice v sistemu IJPP.

### 2.1.1.5 Zahteva za duplikat IJPP kartice

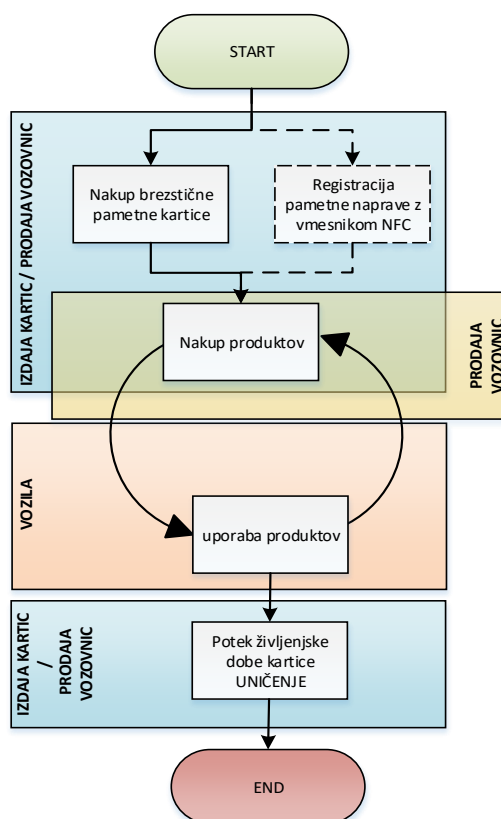
V primeru prijave izgubljene kartice, se lahko potniku izdela duplikat kartice. V tem primeru se izgubljena kartica vstavi v seznam črnih list, kar onemogoča nadaljnjo rabo izgubljene kartice, potniku pa se izdela duplikat kartice, ki omogoča nadaljnjo nemoteno uporabo kartice v sistemu IJPP.

### 2.1.1.6 Predaja potekle kartice

Vsaka kartica IJPP ima določeno veljavnost. Po preteku časa je kartica neveljavna, kar pomeni, da onemogoča validiranje in nalaganje novih vozovnic na kartico. To kartico potnik preda prevoznikom, ki nato skupaj z upravljavcem poskrbijo za uničenje poteklih kartic.

Spodnja Slika 2.2 prikazuje uporabo sistema IJPP s strani potnika z zgoraj opisanimi postopki. Potnik pridobi novo IJPP kartico ali opravi inicializacijo obstoječe brezstične kartice na katero se naloži aplikacija IJPP. Prevoznik nato na prodajnem mestu prevoznika, ki je opremljeno z verificirano terminalsko opremo, opravi nakup izbranih vozovnic. Pri nakupu se kupljene vozovnice naložijo na brezstično kartico potnika. Ob prihodu na vozilo prevoznika potnik opravi validacijo. Potnik kartico za validacijo in nakupe uporablja do izteka življenjske dobe kartice.

#### DS 4: Integracija IS sistemov prevoznika

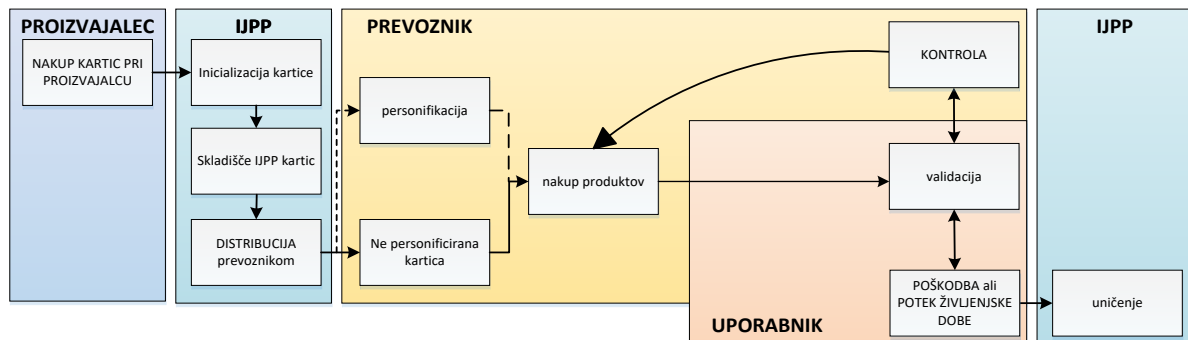


SLIKA 2.2: OSREDNJI POSTOPKI PRIDOBITVE KARTICE, NAKUPA VOZOVNIC TER NATO KORIŠČENJE PRI PREVOZNIKIH DO IZTEKA ŽIVLJENSKE DOBE BREZSTIČNE KARTICE Z NAMEŠČENO IJPP APLIKACIJO.

#### 2.1.2 ŽIVLJENJSKI CIKEL BREZSTIČNE KARTICE ZNOTRAJ SISTEMA IJPP

Brezstična pametna kartica predstavlja osnovni medij, ki se uporablja znotraj sistema IJPP s strani potnikov za validacijo na prevoznih sredstvih prevoznikov. Pred uporabo storitev IJPP mora potnik pridobiti brezstično kartico, ki ima naloženo aplikacijo IJPP. Omogočena je tudi uporaba več različnih aplikacij na eni brezstični kartici, kar specifičnemu prevozniku omogoča uporabo že obstoječega sistema za izvajanje lokalnih voženj. Pred uporabo mora biti kartica IJPP z naloženo aplikacijo prav tako uspešno registrirana v sistem IJPP. Potnik lahko kartico uporablja za validacijo in nakup novih vozovnic do izteka življenjske dobe kartice, ki je zapisana v aplikaciji IJPP naložena na brezstični kartici. Slika 2.3. prikazuje življenjski cikel brezstične kartice znotraj sistema IJPP, ki jo uporabljajo potniki.

## DS 4: Integracija IS sistemov prevoznika



SLIKA 2.3: ŽIVLJENJSKI CIKEL BREZSTIČNE KARTICE Z NAMEŠČENO APLIKACIJO IJPP

## 2.2 PREVOZNIK

Predstavlja končnega ponudnika IJPP storitev, ki jih potnik koristi. Prevozniki v sistemu IJPP predstavljajo gospodarsko javno službo, ki izvajajo prevoze potnikov in druge storitve znotraj sistema IJPP. Javni potniški promet IJPP organizira prevoz potnikov v medkrajevnem linijskem avtobusnem prometu, železniškem prometu in mestnem avtobusnem prometu.

Za uspešno integracijo v sistem IJPP mora prevoznik omogočati potnikom uporabo več različnih storitev, hkrati pa zagotavljati dosledno in nemoteno uporabo produktov IJPP, kar lahko zagotovi z uporabo naslednjih postopkov in sicer v relaciji s potniki in upravljavcem:

### 2.2.1 POSTOPKI PREVOZNIKA

Postopki prevoznika v relaciji s potniki

- Izdaja kartice
- Prodaja vozovnic
- Izdaja duplikatov
- Validacija potnikov na prevoznem sredstvu
- Kontrola ustreznih vozovnic potnikov
- Blokiranje kartice
- Blokiranje vozovnice



## DS 4: Integracija IS sistemov prevoznika

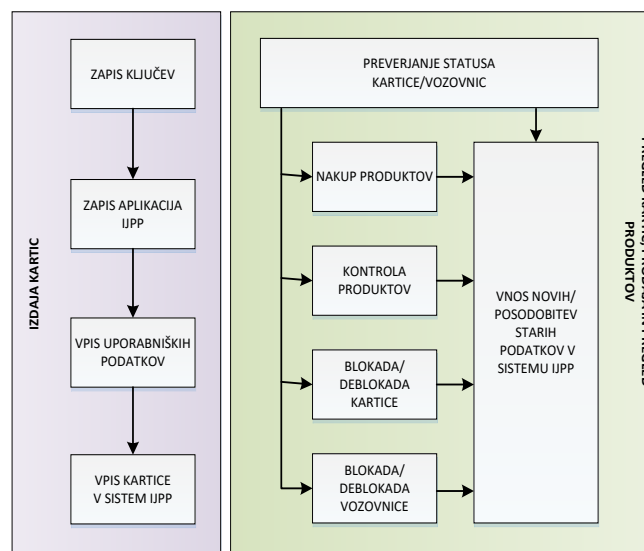
- Odvzem kartice

Postopki prevoznika v relaciji z upravljavcem

- Vpis novih prodajnih mest v sistem IJPP
- Pridobitev osnovnih podatkov zalednega sistema (tarifa,...)
- Sporočanje podatkov zalednemu sistemu (batch)

### 2.2.1.1 Izdaja kartice

V sistemu IJPP se lahko uporabijo nove kartice IJPP ali se na obstoječe brezstične kartice prevoznikov z inicializacijskim postopkom naloži aplikacija IJPP. V prvem primeru, se bodo izdajale kartice, ki so skladne z zahtevami v tem standardu. V drugem primeru pa morajo biti brezstične kartice prevoznikov prav tako skladne s standardom. Spodnja Slika 2.4. prikazuje izdajo novih kartic in inicializacija obstoječih kartic prevoznikov, na katere se bo naložila aplikacija IJPP.



Slika 2.4: Delovanje sistema za izdajo novih in pregled starih kartic ter produktov

### 2.2.1.2 Prodaja vozovnic

Prevoznik omogoča nakup vozovnic na točno določenih prodajnih mestih znotraj IJPP sistema na osnovi dogovora z upravljavcem sistema IJPP. Nakup je možen samo z brezstičnimi karticami z naloženo aplikacijo IJPP. Nakup določenih vozovnic na kartico je omejen, saj je v sistemu IJPP vpeljan pravilnik, ki določa katere vozovnice lahko kupi potnik s točno določenim tipom kartice (pravice- tarifni razred). Ta pravila so določena v tarifi, ki se nato upoštevajo ob nakupu

## DS 4: Integracija IS sistemov prevoznika

---

produktov na prodajnem mestu. Tarifa skupaj s pravili so definirana v nadaljevanju tega dokumenta.

Pri nakupu ločimo dva tipa vozovnic in sicer:

- Vozovnice za mestni promet (zapis cone na kartico)
- Vozovnice za medkrajevni promet (zapis relacije na kartico)

### 2.2.1.3 Izdaja duplikatov

V primeru prijave izgubljene kartice, se lahko potniku izdela duplikat izgubljene kartice. Prevoznik mora omogočati izdelavo duplikatov po standardu ob tem je potrebno izgubljeno kartico sporočiti upravljavcu, ki to kartico vstavi na seznam črnih list. S tem je onemogočena nadaljnja raba izgubljene kartice v sistemu IJPP.

### 2.2.1.4 Validacija potnikov na prevoznem sredstvu

Ob prihodu potnika na prevozno sredstvo vozila je le ta dolžan opraviti validacijo. Validacija je postopek, kjer potnik približa brezstično pametno kartico k čitalcu terminalske opreme vozila. Pri postopku validacije se med brezstično kartico in čitalcem izmenjajo podatki, ki so potrebni za določitev rezultata validacije. Rezultat validacije je lahko :

- Uspešna validacija

Uspešno validacija pomeni, da ima potnik pravico do izbrane vožnje

- Neuspešna validacija

Neuspešno validacija pomeni, da potnik nima pravico do izbrane vožnje

- Ponovitev

Pri ponovitvi pa mora potnik ponoviti postopek validacije, saj je pri predhodnem postopku prišlo do nepravilne izmenjave podatkov med brezstično kartico in čitalcem.

Rezultat validacije na terminalski opremi se izvede glede na definiran algoritem validacije, ki se izvede glede na podatke na kartici, tarife terminala, črne liste, izvršne liste, voznega reda in trenutne lokacije vozila. Natančna definicija algoritma validacija sledi v nadaljevanju dokumenta.

Vozovnice ob nakupu imajo status na IJPP kartici »Neaktivni«, ki se ob prvi validaciji aktivirajo.

### 2.2.1.5 Kontrola ustreznih vozovnic potnikov

## DS 4: Integracija IS sistemov prevoznika

---

Upravljavec skupaj s prevozniki skrbi za kontrolo ustreznih vozovnic za potnike. Kontrolo izvajajo pooblaščen osebje upravljavca IJPP, ki izvajajo občasne kontrole vozovnic na vozilih. Kontrolno osebje preverja veljavnost vozovnice in način uporabe pri potnikih v skladu s splošnimi in posebnimi pogoji prevoza IJPP. Kontrolno osebje kontrolo vozovnic izvaja avtomatsko z uporabo ročnih mobilnih terminalov. Kontrolo statusa, ki velja za posebne skupine potnikov, kontrolno osebje izvaja na osnovi osebne identifikacije. Kontrolor lahko od potnika zahteva, da izkaže svojo identiteto z veljavnim osebnim dokumentom s sliko. Če se potnik med prevozom ne ravna v skladu s Splošnimi in posebnimi pogoji prevoza, ga lahko kontrolor opozori, mu izreče globo ali ga napoti izven vozila. Kontrolor v skladu s pravili kontrole izreka globe za nepravilno uporabo sistema IJPP ali druge ukrepe, ki so predpisani v nadaljevanju dokumenta.

### 2.2.1.6 Blokiranje kartice

Pri validaciji potnika na terminalski opremi vozila se preveri ali je kartica na črni listi IJPP, ki jo terminalska oprema pridobi iz zalednega sistema. Kartica je lahko na črni listi iz 3 razlogov:

- Izguba/kraja kartice
- Dublikat kartice
- Zloraba kartice

V primeru, da validacijski algoritem ugotovi, da je kartica na črni listi, se izvede blokada IJPP aplikacije. To pomeni, da s to kartico ne bo mogoče opraviti validacije v sistemu IJPP ali opraviti nakup vozovnic za to kartico.

Blokiranje kartice pomeni zgolj blokiranje aplikacije IJPP na kartici. V primeru, da ima kartica več aplikacij (ob IJPP aplikaciji še aplikacijo prevoznika), ostanejo ostale aplikacije nespremenjene in se lahko uporabljajo naprej (o tem se odloča prevoznik).

### 2.2.1.7 Blokiranje vozovnice

Pri validaciji potnika na terminalski opremi vozila se preveri ali je vozovnica izbrana za validacijo na izvršni listi IJPP, ki jo terminalska oprema pridobi iz zalednega sistema.

V primeru, da validacijski algoritem ugotovi, da je vozovnica za pripadajočo brezstično kartico na izvršni listi, se izvede blokada vozovnice znotraj IJPP aplikacije. To pomeni, da te vozovnice v bodoče ne bo možno validirati do preklica.

### 2.2.1.8 Odvzem kartice

Odvzem kartice potniku se lahko izvede s strani kontrolorja, ki pri kontroli vozovnic na IJPP kartici ugotovi zlorabo, za katere je predpisan odvzem kartice. Odvzem kartice lahko prav tako izvede prevoznik, če ugotovi zlorabo kartice v sistemu IJPP.

#### **2.2.1.9 Vpis novih terminalov v sistem IJPP**

Pred samo uporabo terminalske opreme je potrebno izvesti vpis terminalske opreme v sistem. Terminalska oprema se doda v seznam terminalske opreme na zalednem sistemu, ki ima nato pravico pridobitve vseh potrebnih podatkov za normalno delovanje terminalske opreme na prodajnih mestih ali mestih za validiranje. Vpis novih terminalov v sistem IJPP se izvede prek definiranega web servisa v nadaljevanju dokumenta.

#### **2.2.1.10 Pridobitev osnovnih podatkov zalednega sistema (tarifa,...)**

Terminalska oprema pri prevoznikih za svojo pravilno in nemoteno delovanje potrebuje informacije upravljalca, ki jih pridobi iz zalednega sistema. Ti podatki so:

- Vozni redi
- Tarife
- Črne liste
- Izvršne liste
- Šifranti
- Pravila za prestopanje v mestnem prometu

Terminalska oprema mora pred začetkom delovanja pridobiti vse zgoraj naštetе podatke, šele nato lahko omogoči validiranje ali nakup produktov. Ob tem mora v točno določenih okvirjih preveriti ali so na zalednem sistemu na voljo novi podatki, ki jih mora prenesti terminal (npr. Nova tarifa). Dostop do podatkov na zalednem sistemu je omogočen preko definiranih web servisov. Standard natančno definira časovne okvirje kako in kdaj terminalska oprema pridobi ali pošlje poizvedbo o podatkih na zalednem sistemu.

#### **2.2.1.11 Sporočanje podatkov zalednemu sistemu (batch)**

Zaledni sistem skrbi za vodenje, pregled in sprejemanje podatkov iz terminalske opreme. Terminalska oprema mora v vnaprej določenih časovnih okvirjih sporočiti določene dogodke na terminalski opremi kot je npr. validacija ali nakup vozovnic potnika. Standard natančno definira časovne okvirje kako in kdaj terminslaka oprema sporoči podatke na zalednem sistemu. Ti podatki so:

- Poročanje o validaciji/aktivaciji
- Poročanje o izdaji in inicializaciji IJPP kartic
- Poročanje o nakupu produktov
- Poročanje o stornaciji

## DS 4: Integracija IS sistemov prevoznika

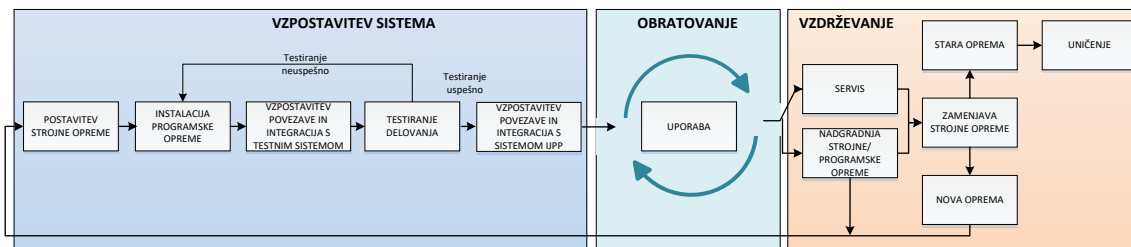
- Preverjanje veljavnosti podatkov (servisni klic)

### 2.2.2 TERMINALSKA OPREMA ZA INICIALIZACIJO ALI IZDAJO KARTIC

Za izvedbo inicializacijskega postopka ali same izdaje kartice je potrebna verificirana oprema skladna s standardom pri prevoznikih. Ta oprema je lahko nameščena tako na prodajnih mestih kot na samem vozilu. Za zagotavljanje osnovne funkcionalnosti kartičnega sistema in podporo vseh naštetih postopkov prevoznikov so potrebne naslednje entitete:

- bralno/pisalna enota za brezstične kartice,
- računalnik oz. drug sistem, na katerega je bralno/pisalna enota priključena in omogoča:
  - popolno podporo bralno pisalni enoti brezstičnih kartic
  - varno povezavo do IJPP zalednega sistema
  - uporabo programske opreme za vpis novih kartic, branje in pregled nad starimi karticami ter modifikacijo statusa kartice in posameznih produktov
  - uporabo programske opreme za nakup produktov
  - uporabo programske opreme, ki omogoča dostop do sistema IJPP (npr. programska oprema za dostop do portala IJPP - internetni brskalnik)
- enota, ki omogoča izdajo računov za prodane produkte.

Življenjski cikel sistema za izdajo, novih in pregled starih kartic ter produktov predstavlja Slika 2.5:

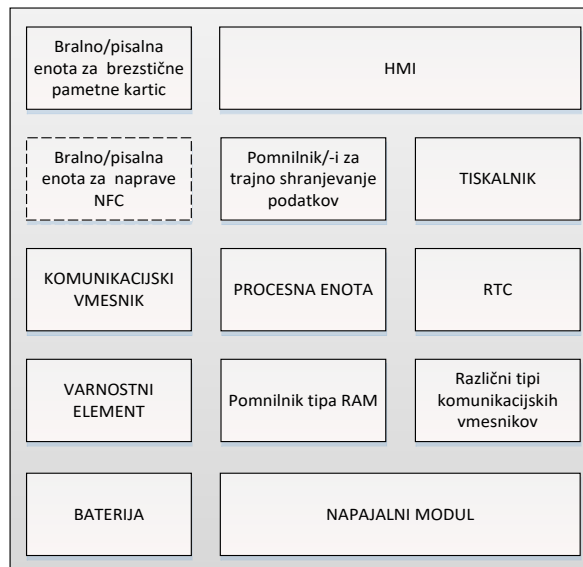


SLIKA 2.5: ŽIVLJENJSKI CIKEL SISTEMA ZA IZDAJO, NOVIH IN PREGLED STARIH KARTIC TER PRODUKTOV

### 2.2.3 TERMINALSKA OPREMA ZA ZUNANJA PRODAJNA MESTA

Prodajni terminal je del sistema prodaje IJPP, ki omogoča uporabo storitev sistema IJPP na prodajnih mestih posameznega prevoznika. Prodajni terminal omogoča izpis vsebine kartice in prodajo ter zapis kupljenih produktov (vozovnic) na brezstične kartice. Prodajno mesto z vključeno opremo mora omogočati tudi nakup listnih vozovnic. Ta entiteta mora podpirati tako delovanje v samostojnem režimu, kot morebitno integracijo z drugimi napravami (npr. blagajna). Osnovne gradnike prodajnega terminala prikazuje Slika 2.6.

#### DS 4: Integracija IS sistemov prevoznika



SLIKA 2.6: SHEMA GRADNIKOV TERMINALA NA PRODAJNIH MESTIH PREVOZNIKA IN DRUGIH PRODAJNIH MESTIH

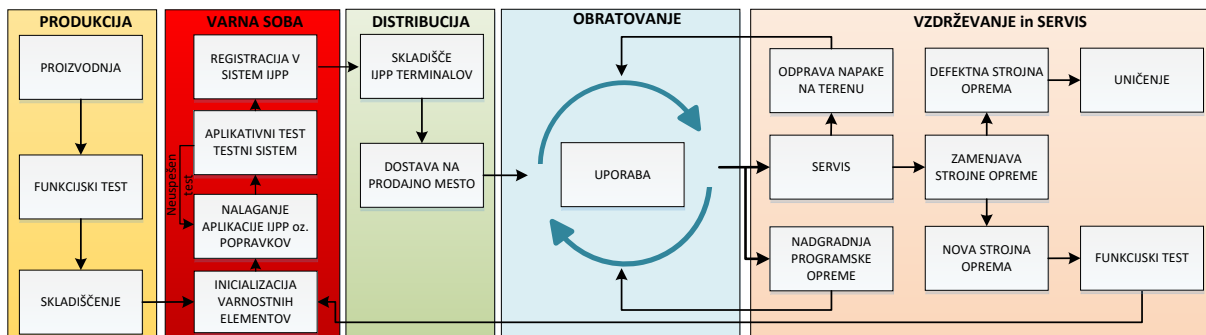
Najpomembnejši gradniki takšnega terminala so:

- Bralno/pisalna enota (čitalec) brezstičnih kartic – ta omogoča terminalu prodajnega mesta operacijo s karticami sistema IJPP
- čitalec kartic lahko OPCISKO omogoča še podporo napravam NFC (za morebitno kasnejšo nadgradnjo)
- komunikacijski vmesnik – služi za varno komunikacijo z zalednim sistemom IJPP
- varnostni element – služi za namene shranjevana varnostno občutljivih informacij (ključi za dostop do kartice, ključi za kodiranje transakcij, ipd...)
- procesna enota – je namenjena poganjanju programske opreme terminala
- pomnilnik tipa RAM (ang. Random Access Memory) – je lahko del procesne enote ali pa samostojen element in služi začasnemu shranjevanju podatkov procesne enote
- pomnilnik/-i za trajno shranjevanje podatkov – služijo trajnemu shranjevanju podatkov, kot so podatki o transakcijah, opravljenih na dotičnem terminalu
- različni tipi komunikacijskih vmesnikov – služijo za povezavo zunanjih naprav na terminal prodajnega mesta
- modul RTC (ang. Real Time Clock) – služi za zagotavljanje natančnega časa; ta se potrebuje pri prodaji in verifikaciji produktov, opravljanju transakcij in zapisovanju podatkov na brezstično kartico
- tiskalnik – služi za izdajanje računov in potrdil o opravljenih storitvah in tiskanje kupljenih listnih vozovnic.
- vmesnik za upravljanje terminala prodajnega mesta oz. HMI (ang. Human Machine Interface) – vmesnik omogoča operaterju uporabo terminala prodajnega mesta. Operaterju prikazuje informacije in hkrati od operaterja sprejema vhodne informacije (zaslon in tipkovnica ali zaslon na dotik, različni svetlobni in zvočni indikatorji...)

#### DS 4: Integracija IS sistemov prevoznika

- baterija – zagotavlja terminalu prodajnega mesta izvor energije za nemoteno delovanje
- napajalni modul – omogoča polnjenje baterije in delovanje terminala prodajnega mesta s priključitvijo na električno omrežje
- drugi opcijski gradniki, npr. čitalec črtne kode.

Slika 2.7 prikazuje primer življenjskega cikla terminala prodajnega mesta specifičnega prevoznika.



SLIKA 2.7: ŽIVLJENJSKI CIKEL TERMINALA NA PRODAJNIH MESTIH PREVOZNIKA IN DRUGIH PRODAJNIH MESTIH

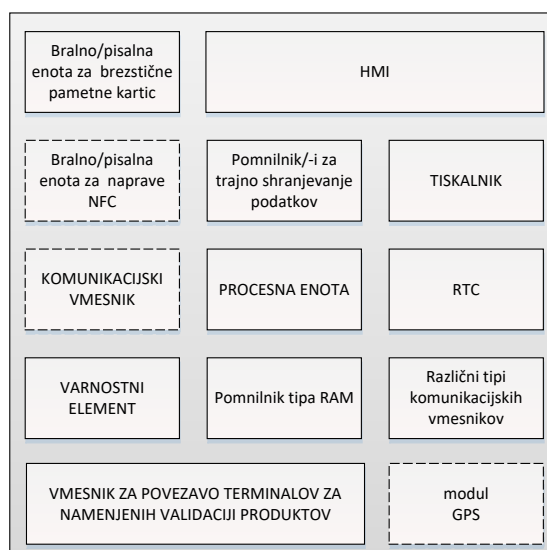
#### 2.2.4 TERMINALSKA OPREMA ZA VALIDIRANJE POTNIKOV

Primarna naloga sistema IJPP na vozilih je validacija in prodaja vozovnic potnikom. Pri prodaji vozovnic je potrebno zagotoviti še prodajo klasičnih listnih vozovnic. Skladno s pogojem izdaje listnih vozovnic je potrebno zagotoviti tiskanje le teh v definirani obliki (glej definicijo listne vozovnice). Glede na namen uporabe se lahko sistem na vozilih razdeli na dva dela, in sicer:

- sistem za validacije, in
- sistem za prodajo produktov.

Glavni gradnik takšnega sistema je terminal, ki omogoča proces validacije produktov IJPP, ki so naloženi na brezstično kartico sistema IJPP (opcijsko napravo z vmesnikom NFC). Slika 2.8 prikazuje gradnike, ki omogočajo terminalu opisano funkcionalnost.

## DS 4: Integracija IS sistemov prevoznika



SLIKA 2.8: OSNOVNI GRADNIKI TERMINALA ZA NAMENE VALIDACIJE PRODUKTOV

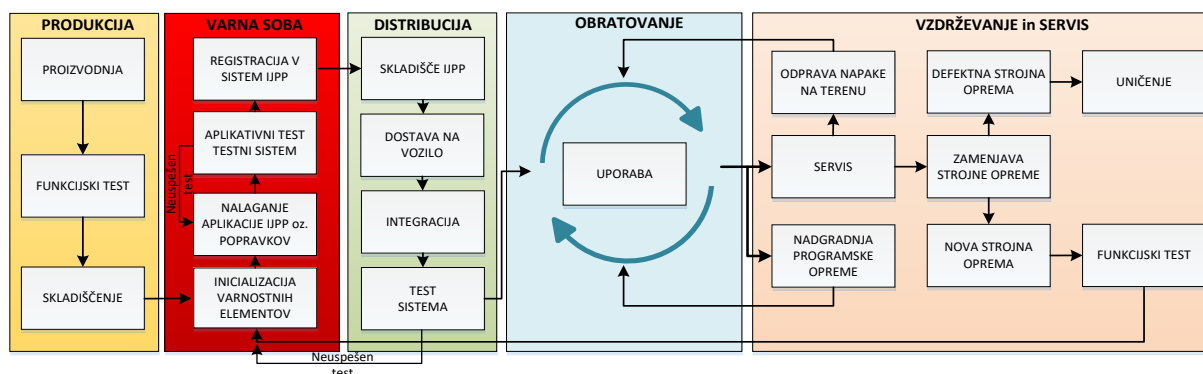
Najpomembnejši gradniki takšnega terminala so:

- Bralno/pisalna enota (čitalec) brezstičnih kartic – ta omogoča terminalu operacijo s karticami sistema IJPP
- čitalec kartic lahko **OPCIJSKO** omogoča podporo napravam NFC (za morebitno kasnejšo nadgradnjo)
- komunikacijski vmesnik – služi za varno komunikacijo z zalednim sistemom IJPP
- varnostni element – služi shranjevanju varnostno občutljivih informacij (ključi za dostop do kartice, ključi za kodiranje transakcij, ipd...)
- procesna enota – je namenjena poganjanju programske opreme terminala
- pomnilnik tipa RAM (ang. Random Access Memory) – je lahko del procesne enote ali kot samostojen element. Služi začasnemu shranjevanju podatkov procesne enote
- pomnilnik/-i za trajno shranjevanje podatkov – služijo trajnemu shranjevanju podatkov, kot so podatki o izvedenih transakcijah, opravljenih na dotičnem terminalu ipd.
- modul RTC (ang. Real Time Clock) – služi za zagotavljanje natančnega časa; ta se potrebuje pri prodaji in verifikaciji produktov, pri pisanju podatkov na brezstično kartico ter pri opravljanju transakcij
- vmesnik za upravljanje terminala oz. HMI (ang. Human Machine Interface) omogoča potniku prikaz stanja na kartici ter rezultate validacije
- modul GPS za določanje pozicije vozila (**OPCIJSKO**)
- drugi opcijski gradniki

Predviden življenjski cikel terminala za potrebe validacij in prodajo produktov IJPP prikazuje Slika 2.9.



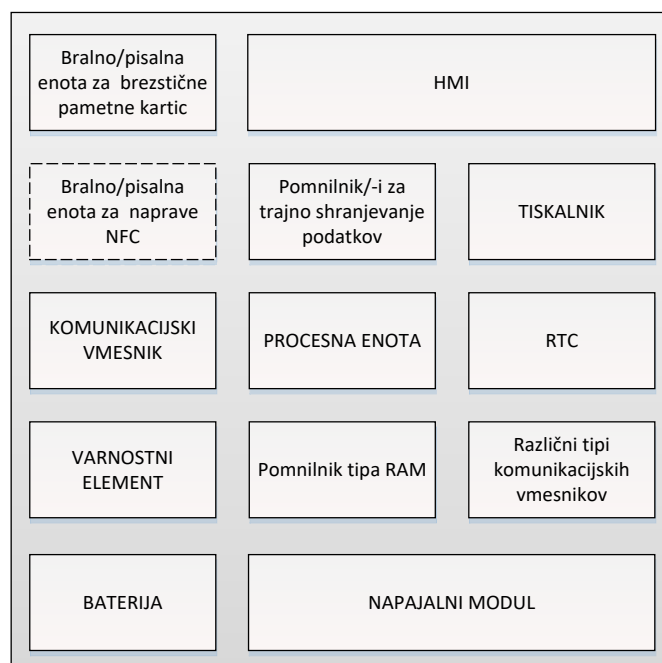
## DS 4: Integracija IS sistemov prevoznika



SLIKA 2.9: ŽIVLJENJSKI CIKEL TERMINALA ZA NAMENE VALIDACIJE PRODUKTOV IJPP

### 2.2.5 TERMINALSKA OPREMA ZA KONTROLO POTNIKOV

Glavni gradnik tega sistema je mobilni terminal, ki omogoča proces kontrole validacije produktov IJPP, ki so naloženi na brezstični kartico sistema IJPP (opcijsko z napravo, ki premore vmesnik NFC) določenega potnika. Zraven kontrole produktov mora terminal omogočati še funkcionalnost za nakup produkta in funkcionalnost za izdajanje kazni/opozorila potniku (uporabnikom), ki nimajo ustreznega produkta. Arhitektura gradnikov, ki terminalu omogočajo opisano funkcionalnost prikazuje Slika 2.10.



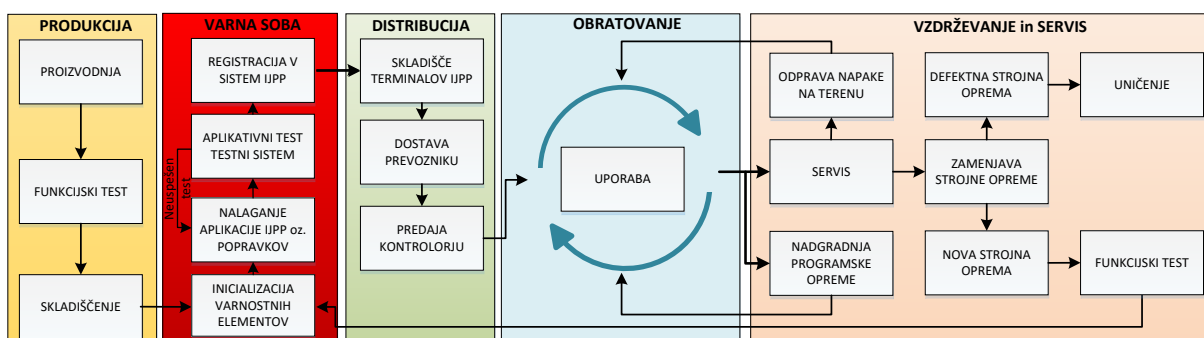
SLIKA 2.10: OSNOVNI GRADNIKI TERMINALA ZA NAMENE KONTROLE PRODUKTOV

## DS 4: Integracija IS sistemov prevoznika

### Osnovni gradniki:

- Bralno/pisalna enota (čitalec) pametnih brezstičnih kartic – ta omogoča terminalu operacije s karticami sistema IJPP.
- čitalec kartic lahko **OPCIJSKO** omogoča podporo napravam NFC (za morebitne kasnejše nadgradnje)
- komunikacijski vmesnik – služi za varno komunikacijo z zalednim sistemom IJPP
- varnostni element – služi shranjevanju varnostno občutljivih informacij (ključi za dostop do kartice, ključi za kodiranje transakcij, ipd...)
- procesna enota – je namenjena poganjanju programske opreme terminala
- pomnilnik tipa RAM (ang. Random Access Memory) – je lahko del procesne enote ali kot samostojen element, služi začasemu shranjevanju podatkov procesne enote,
- pomnilnik/-i za trajno shranjevanje podatkov služijo trajnemu shranjevanju podatkov, kot so podatki o transakcijah opravljenih na dotičnem terminalu ipd.
- različni tipi komunikacijskih vmesnikov služijo za povezavo zunanjih naprav na terminal
- modul RTC (ang. Real Time Clock) – služi za zagotavljanje natančnega časa; ta se potrebuje pri prodaji in verifikaciji produktov, pri pisanju podatkov na brezstično kartico ter beleženju opravljenih transakcij
- tiskalnik – služi za izdajanje kazni, opozoril ali potrdil o opravljenih storitvah
- vmesnik za upravljanje terminala oz. HMI (ang. Human Machine Interface) omogoča uporabniku rokovanje s terminalom. Uporabniku prikazuje informacije in hkrati od uporabnika sprejema vhodne informacije (zaslon in tipkovnica ali zaslon na dotik, različni svetlobni in zvočni indikatorji, itd.)
- baterija zagotavlja avtonomno delovanje terminala
- napajalni modul omogoča polnjenje baterije iz omrežne napetosti
- drugi opcijski gradniki.

Predviden življenjski cikel terminala za namene kontrole produktov IJPP prikazuje Slika 2.11.



SLIKA 2.11: ŽIVLJENJSKI CIKEL TERMINALA ZA NAMENE KONTROLE PRODUKTOV IJPP

## 2.3 UPRAVLJAVEC

Informacijski sistem upravljavca je namenjen in prilagojen uvajanju in rabi enotne vozovnice na osnovi poenotenih voznih redov na ravni avtobusnega in železniškega javnega prometa znotraj javnega avtobusno-železniškega omrežja v RS. Sistem IJPP mora zagotavljati kakovostne storitve in s tem omogočiti potnikom in prevoznikom tako v železniškem kot avtobusnem javnem potniškem prometu, učinkovito in prijazno uporabo.

Upravljavec skrbi za celotno delovanje sistema in ga sestavlja:

- Administrativni del

Upravljavec sistema IJPP skrbi za celoten nadzor in delovanje sistema na različnih nivojih in sicer upravljavec sistema JPP, voznih redov, vozovnic, pogojev prevoza, prodajnih in validacijskih mest, informiranja potnikov, financiranja in informacijske podpore delovanju in upravljanju sistema vozovnic. Prav tako skrbi za integriran prostorski razvoj sistema (prestopne točke, oprema postaj in postajališč...).

- Zaledni sistem

Zaledni sistem predstavlja osrednji del informacijskega sistema IJPP omrežja, ki omogoča upravljavcu sistema celoten vpogled in kontrolo nad informacijskim sistemom. Zalednemu sistemu upravljavca se poročajo vse transakcije vezane na sistem IJPP; inicializacija, nakup, validacija, stornacija, polno vračilo, delno vračilo, blokada, deblokada, itd. Točne specifikacije posamezne transakcije so zajete v nadaljevanju dokumenta.

### 2.3.1 ADMINISTRATIVNI DEL SISTEMA IJPP

Upravljavec sistema IJPP je nadzorni organ, ki skrbi za upravljanje celotnega sistema IJPP. Skrbi za celoten nadzor in delovanje sistema na različnih nivojih, in sicer: upravljanje sistema IJPP, voznih redov, vozovnic, produktov, tarif, pogojev prevoza, prodajnih in validacijskih mest, informiranja potnikov, financiranja, informacijsko podporo delovanju in upravljanju sistema vozovnic ipd. Prav tako skrbi za integriran prostorski razvoj sistema (prestopne točke, oprema postaj in postajališč, itd.).

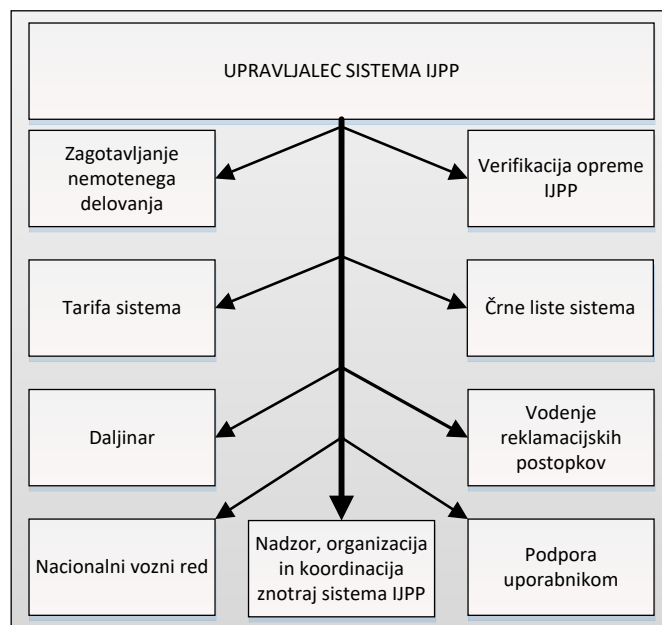
Naloge upravljavca so:

- Zagotavljanje nemotenega delovanja sistema – primarna naloga upravitelja.
- Upravljanje sistema – nadzor, organizacija in koordinacija različnih subjektov (prevoznikov, prodajnih mest, produktov, itd.).

#### DS 4: Integracija IS sistemov prevoznika

- Nacionalni vozni red – upravljavec mora poskrbeti, da se vse spremembe, ki so bile dogovorjene in usklajene s prevozniki, pravilno odražajo na nacionalnem voznom redu in so razvidne potnikom oziroma uporabnikom sistema.
- Tarife sistema – upravitelj ima nadzor nad tarifo sistema IJPP in s tem možnost dodajanja novih tipov vozovnic, statusov, cen itd.
- Črna lista kartic – upravitelj mora nadzirati črno listo kartic. To so kartice, ki so bile zaradi takšnega ali drugačnega razloga (izguba, sankcija) postavljene na črno listo sistema IJPP.
- Črna lista vozovnic – upravitelj mora nadzirati črno listo vozovnic. To so vozovnice, ki so bile zaradi takšnega ali drugačnega razloga (nepravilna uporaba, manipulacija, itd.) postavljene na črno listo vozovnic sistema IJPP.
- Upravitelj IJPP sistema je odgovoren za verifikacijo opreme IJPP.
- Upravitelj IJPP vodi morebitne reklamacijske in servisne postopke opreme sistema IJPP.
- Upravitelj IJPP sistema mora zagotavljati uporabnikom prijazno podporo (npr. klicni center).

Pomembnejše naloge upravljavca sistema IJPP prikazuje Slika 2.12.



SLIKA 2.12: SHEMA POMEMBNEJŠIH NALOG UPRAVITELJA SISTEMA IJPP

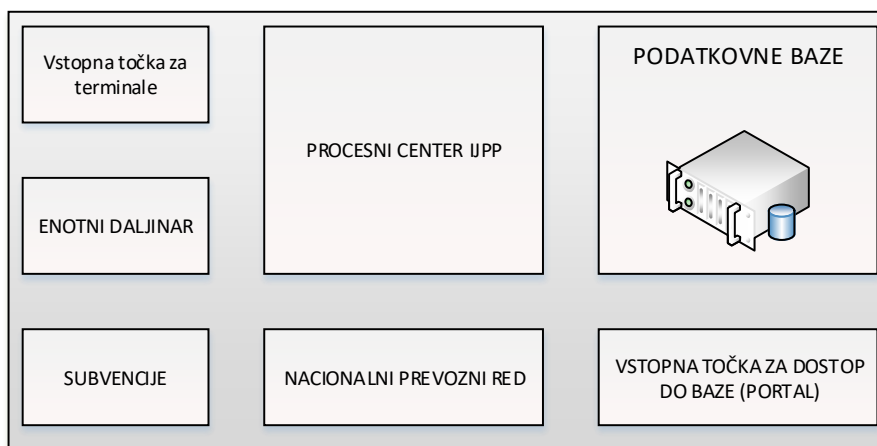
Sistem upravljavca IJPP ni del standarda, zato standard ne vsebuje podrobnega opisa tega podsistema.

### 2.3.2 ZALEDNI SISTEM IJPP

Zaledni sistem predstavlja osrednji del informacijskega sistema IJPP, ki omogoča upravljavcu IJPP celoten vpogled in kontrolo nad informacijskim sistemom. Zaledni sistem je ključni del, saj se preko njega izvajajo vse operacije, ki potekajo znotraj IJPP. Te operacije med drugim vključujejo:

1. operacije inicializacije kartic (registracije naprav NFC),
2. operacije upravljanja s karticami,
3. operacije izdajanja produktov,
4. operacije upravljanja s produkti,
5. operacije s transakcijami,
6. izmenjava podatkov med prevozniki in IJPP,
7. prenos zahtev za servis/modifikacijo/nadgradnjo,
8. shranjevanje in procesiranje podatkov,
9. pripravo poročil
10. pripravo osnov za finančna poročila
11. itd.

Zaledni sistem IJPP se lahko razdeli na posamezne bloke, kot jih prikazuje Slika 2.13.



SLIKA 2.13: SHEMA ZALEDNA SISTEMA IJPP

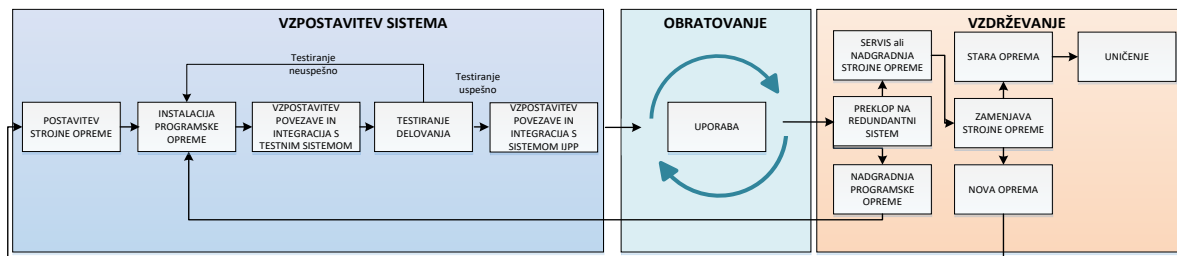
Opis funkcionalnosti po posameznih blokih:

12. Vstopna točka za terminale – jo predstavlja primerno zaščiten (APN, VPN, SSL, SSH) IP naslov skupaj s številko vrat. Namen vstopne točke je omogočanje povezave terminalov do procesnega sistema, ki procesira zahteve/transakcije, prejete iz terminalov. Zaledni sistem ponuja več vstopnih točk s čimer zagotavlja nemoteno delovanje sistema v primeru okvare.
13. Procesni center – je jedro sistema. Sestavljajo ga:
  - a. programska oprema (sistem), katera je namenjena služi obravnavanju transakcijskih zahtevkov terminalov,

#### DS 4: Integracija IS sistemov prevoznika

- b. mehanizem upravljanja s potniki, karticami, produkti in terminali sistema IJPP,
  - c. mehanizem za shranjevanje in prenos servisnih in drugih zahtev,
  - d. mehanizem ustvarjanja voznih redov in nacionalnega voznega reda,
  - e. drugo....
14. Enotni DALJINAR,
  15. Subvencije – so del sistema, katere omogočajo izvajanje subvencioniranega programa za šolarje, študente, starejše občane, upokoјence ipd.
  16. Nacionalni vozni red se ustvarja na podlagi podatkov iz enotnega daljinarja in podatkov o omogočenih subvencijah.
  17. Vstopna točka za povezavo zunanjih sistemov (kot npr. portal IJPP).

Predviden življenjski cikel zalednega sistema IJPP prikazuje Slika 2.14



SLIKA 2.14: ŽIVLJENJSKI CIKEL ZALEDNA SISTEMA IJPP

## 2.4 POSTOPKI UPRAVLJAVCA

Relacija med zalednim sistemom in prevozniki predstavlja izmenjavo podatkov med njima. Za ta namen so definirani postopki, ki omogočajo pridobitev in sporočanje podatkov iz enega ali drugega sistema. Zaledni sistem predstavlja osrednji del informacijskega sistema IJPP omrežja v katerem so shranjeni vsi potrebni podatki o delovanju celotnega sistema, kot npr. podatek o validaciji potnika na določenem prevoznem sredstvu prevoznika.

S stališča povezave prevoznika do zalednega sistema se komunikacija med njima deli na:

- **Pridobivanje podatkov IJPP**  
V tem primeru terminalska oprema iz zalednega sistema pridobi osnovne informacije, ki so potrebne za nemoteno delovanje terminalske opreme pri prevoznikih. Te informacije so vozni redi, tarife, črne liste, izvršne liste, šifranti, pravila za prestopanje v mestnem prometu,...
- **Poročanje podatkov IJPP**  
V tem primeru terminalska oprema na zaledni sistem sporoči osnovne informacije o dogodkih, ki so se zgodili na terminalski opremi prevoznikov. Te informacije so poročanje o validaciji, poročanje o izdaji in inicializaciji IJPP kartic, poročanje o nakupu produktov,...

## DS 4: Integracija IS sistemov prevoznika

---

Postopki pridobivanja podatkov:

- Upravljanje tarife
- Upravljanje črne liste
- Upravljanje izvršne liste
- Upravljanje voznih redov
- Pravice za prestop
- Seznam prodajnih mest
- Seznam kartic

Postopki poročanja podatkov:

- Inicializacija kartice
- Nakup vozovnic
- Validacija/aktivacija vozovnic
- Izdaja duplikata
- Blokada kartice/vozovnice

### 2.4.1 UPRAVLJANJE TARIFE

Tarifa IJPP je dokument v sistemu IJPP, ki predstavlja kombinacijo cen vrste vozovnic, statusov in tarifnih razredov, ki se s časom spreminjajo in so določene s strani upravljavca sistema IJPP. Upravljavec sistema IJPP ima možnost kadarkoli do odločitve, da posamezne parametre (tarifne razrede, produkte) vključi, izključi ali pa spremeni v tarifi IJPP. Po vsaki spremembi tarife upravljavca se na zaledni sistem vstavi nova tarifa, ki jo nato pridobijo terminali v celotnem omrežju IJPP. Veljavnost tarife je predpisana s časovnimi omejitvami, ki se nanašajo na uporabo le te na terminalski opremi znotraj sistema IJPP.

### 2.4.2 UPRAVLJANJE ČRNE LISTE

Upravljavec vodi črno listo kartic sistema IJPP. Upravljavec ima kadarkoli možnost da posamezne kartice IJPP uvrsti ali odstrani iz seznama črnih list.

Upravljavec IJPP izdaja seznam črnih list IJPP, ki ga nato pridobijo prevozniki (terminalska oprema prevoznikov) znotraj sistema IJPP.

#### 2.4.3 UPRAVLJANJE IZVRŠNE LISTE

Upravljavec vodi tudi črno listo kartic sistema IJPP. Upravljavec ima kadarkoli možnost da posamezne kartice IJPP uvrsti ali odstrani iz seznama črnih list.

Upravljavec IJPP izdaja seznam izvršnih list IJPP, ki ga nato pridobijo prevozniki (terminalska oprema prevoznikov) znotraj sistema IJPP.

#### 2.4.4 UPRAVLJANJE VOZNIH REDOV

Upravljavec upravlja z voznimi redi, ki se določijo na osnovi orodja za prometno planiranje. Vozni red IJPP za vsako linijo vsebuje podatke o vseh zaporednih postajališčih (itinerar) in časih prihodov in odhodov s postajališča v nekem dnevu. Vozni red je povezan s koledarjem voženj in velja za določen režim ali koledar vožnje (npr. vozni red vlaka na neki liniji ob delavnikih).

Upravljavec IJPP izdaja Vozni red IJPP, ki ga nato pridobijo prevozniki (terminalska oprema prevoznikov) znotraj sistema IJPP.

#### 2.4.5 PRAVICE ZA PRESTOPE

Upravljavec prav tako določa pravice za prestopne. Pravice za prestopne je seznam prestopnih točk, kjer potnik prestopa z enega na drugo vozilo, pri tem pa ni pomembno če je prestopanje med različnimi vrstami prevoza (intermodalna prestopna točka), med vozili v različnih režimih vožnje (med vozili v conskem in relacijskem režimu) ali vozili v istem režimu vožnje.

Upravljavec IJPP izdaja pravice za prestopne IJPP, ki ga nato pridobijo prevozniki (terminalska oprema prevoznikov) znotraj sistema IJPP.

#### 2.4.6 SEZNAM PRODAJNIH MEST

Upravljavec sistema IJPP se skupaj s prevozniki dogovori o vključitvi prodajnih mest prevoznikov v prodajno mrežo IJPP, ki omogoča izdajo, inicializacijo in nakup IJPP produktov. Upravitelj vodi seznam terminalske opreme za vsa prodajna mesta znotraj sistema IJPP. Ob vključitvi novih terminalov v sistem IJPP se izvede avtomati vpis terminala v seznam vseh terminalov IJPP sistema.

#### 2.4.7 SEZNAM KARTIC



## DS 4: Integracija IS sistemov prevoznika

---

Upravljaivec vodi seznam vseh kartic z nameščeno IJPP aplikacijo z vsemi potrebnimi podatki o kartici. Ob inicializaciji kartice se na kartico doda aplikacija IJPP ob tem pa terminalska oprema skupaj z zaledni sistemom poskrbi da se kartica doda na seznam vseh kartic IJPP.

### 2.4.8 INICIALIZACIJA KARTICE

Inicializacija kartice se lahko izvaja na novih IJPP karticah ali karticah različnih prevoznikov, ki se trenutno že uporabljajo v različnih sistemih, kateri bodo postali del IJPP sistema.

Terminalska oprema poskrbi za poročanje podatkov o inicializaciji za vsako kartico na zaledni sistem na osnovi definiranih web servisov.

### 2.4.9 NAKUP VOZOVNIC

Nakup vozovnic se izvede na prodajnih mestih. Poročanje nakupa vozovnic se lahko izvede v on-line načina ali pa tudi v off-line načinu. Pri off-line načinu je točno določen čas, v katerem se mora poročanje o prodaji in izdaji kartic izvesti na zaledni sistem.

Terminalska oprema poskrbi za poročanje podatkov o nakupu vozovnic za vsako kartico na zaledni sistem na osnovi definiranih web servisov.

### 2.4.10 VALIDACIJA/AKTIVACIJA VOZOVNIC

Z validacijo/aktivacijo vozovnic potnik aktivira vozovnico na vozilu, pri čemer se z validacijo ugotavlja upravičenost dovoznje. Validacija vozovnice se izvede na terminalski napravi (validator), ki je nameščena v vozilu,

Terminalska oprema poskrbi za poročanje podatkov o validaciji/aktivaciji za vsako kartico na zaledni sistem na osnovi definiranih web servisov.

### 2.4.11 IZDAJA DUBLIKATA

V primeru prijave izgubljene kartice, se lahko potniku izdela duplikat izgubljene kartice. Prevoznik mora omogočiti izdelavo duplikatov po standardu pa vse potrebne podatke o duplikatu kartice sporoči zalednemu sistemu upravljavca.

Terminalska oprema poskrbi za poročanje izdaje duplikata kartice na zaledni sistem na osnovi definiranih web servisov.

### 2.4.12 BLOKADA KARTICE/VOZOVNICE

## DS 4: Integracija IS sistemov prevoznika

---

Pri validaciji potnika na terminalski opremi vozila se preveri ali je kartica na črni listi IJPP in ali je izbrana vozovnica na izvršni listi.

V primeru, da je kartica na črni listi terminalska oprema poskrbi za blokiranje aplikacije IJPP na brezstični kartici in se opravi poročanje terminalske opreme na zaledni center s definiranimi web servisi. V primeru, da je izbrana vozovnica na izvršni listi se opravi blokiranje vozovnice v aplikaciji IJPP na kartici. V tem primeru prav tako terminalska oprema sporoči zalednemu sistemu blokiranje vozovnice na osnovi definirane web servisa.

### 2.4.13 DEBLOKADA KARTICE/VOZOVNICE

Za določene blokirane kartice/vozovnice se lahko izvede deblokasa kartice ali deblokada vozovnice. Deblokada se izvede na osnovi osnovne pozitivne odločbe Upravljavca IJPP, ki je izdana v postopku reklamacije na blagajniškem prodajnem mestu poda zahtevek za deblokado (odblokiranje) vozovnice ali kartice IJPP.

## 2.5 KOMUNIKACIJA NA RELACIJI PREVOZNIK-UPRAVLJAVEC

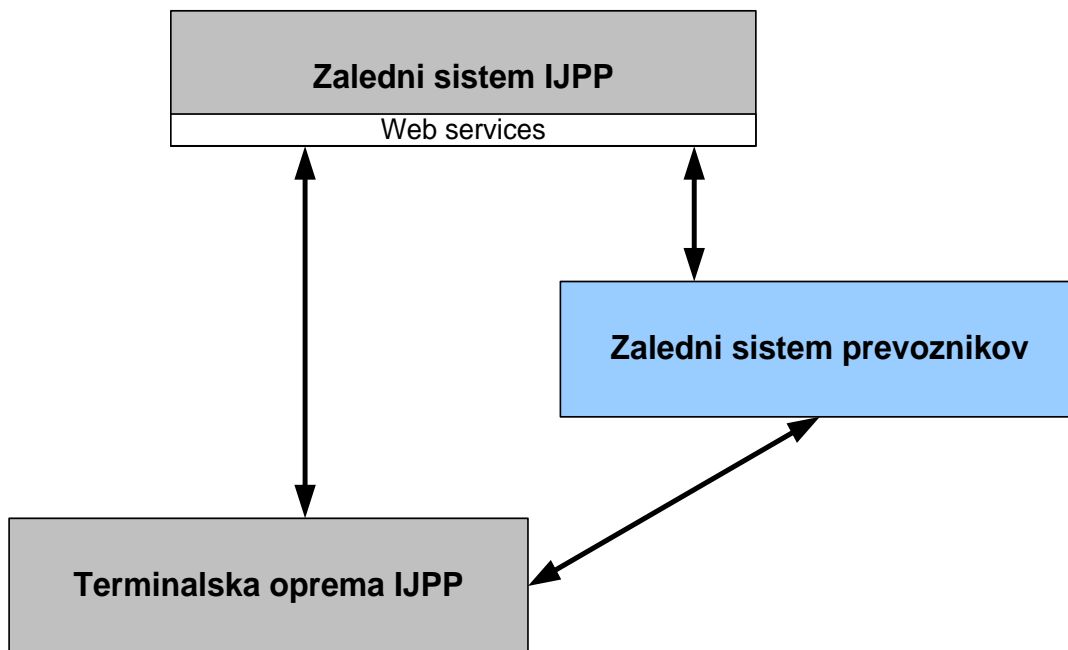
Komunikacija med prevozniki in zalednim sistemom je določena na osnovi web servisov, ki so natančno definirani v nadaljevanju standarda. Prav tako standard v nadaljevanju definira določena pravila za uporabo web servisov, kot ne npr. časovna omejitev preverjanja nove tarife na zalednem sistemu.

Zaledni sistem omogoča komunikacijo z naslednjimi sistemi in komunikacijsko opremo prevoznikov:

- Terminalsko opremo prevoznikov
- Zalednim sistem prevoznikov

Slika prikazuje shematski prikaz povezave zalednega sistema s preostalo komunikacijsko opremo znotraj IJPP omrežja ter samim pretokom informacij.

#### DS 4: Integracija IS sistemov prevoznika



SLIKA 2.15: PONAZORITEV POVEZAVE MED ZALEDNIM SISTEMOM IJPP, ZALEDNIM SISTEMOM PREVOZNIKOV IN TERMINALSKO OPREMO IJPP

V primeru terminalske opreme IJPP le ta izvaja direktno povezavo na IJPP zaledni sistem preko specificiranih web servisov.

**V primeru že obstoječega zalednega sistema prevoznika lahko že obstoječ zaledni sistem prevoznikov povežemo na zaledni sistem IJPP na osnovi specificiranih web servisov. V tem primeru komunikacija med terminalsko opremo IJPP in zalednim sistemom prevoznika ni vnaprej definirana in ni del standarda.**

## 2.6 RELACIJA MED ZALEDNIM SISTEMOM IN UPRAVLJAVCEM

Upravljevec sistema IJPP skrbi za celoten nadzor in delovanje sistema na različnih nivojih in sicer upravljevec sistema JPP, voznih redov, vozovnic, pogojev prevoza, prodajnih in validacijskih mest, informiranja potnikov, financiranja in informacijske podpore delovanju in upravljanju sistema vozovnic. Prav tako skrbi za integriran prostorski razvoj sistema (prestopne točke, oprema postaj in postajališč...).

Relacija med zalednim sistemom in upravljavcem ni del standarda zato standard ne vsebuje podrobnejšega opisa.

## 3 BREZSTIČNA PAMETNA KARTICA

### 3.1. BREZSTIČNA PAMETNA KARTICA MIFARE DESFIRE EV1

MIFARE je blagovna znamka najrazličnejših čipov podjetja NXP Semiconductors, ki se uporabljajo v brezstičnih pametnih karticah. Pod okrilje znamke MIFARE spadajo različne vrste pametnih kartic MIFARE Classic, MIFARE DESFire, MIFARE Plus, MIFARE Ultralight, MIFARE SAM (varnostni element SAM modul). MIFARE Plus je naslednik MIFARE Classic, ki je ob naprednejših varnostnih mehanizmov primeren za enostavnejše aplikacije. MIFARE Ultralight predstavlja poseben tip pametnih vozovnic, modul MIFARE SAM pa nudi kriptografske mehanizme in tako omogoča nadgradnjo obstoječih čitalnikov v smislu izboljšane varnosti.

MIFARE DESFire predstavlja družino brezstičnih pametnih kartic za zahtevnejše aplikacije elektronskih vozovnic, javnega prevoza, cestninjenja cest, omejevanje ali nadzorovanja pooblaščenega dostopa. Tako fizične karakteristike kot varnostni mehanizmi družine DESFire temeljijo na globalnih standardih. Fizične lastnosti in potek komunikacije med čitalnikom definira ISO/IEC 14443A deli 1-4 s skupino določenih ukazov, ki se uporabljajo za komunikacijo in so natančno opredeljeni v ISO/IEC 7816-4. Podprti kriptografski algoritmi 2KDES, 3KDES in AES128 omogočajo varno komunikacijo med karticami in čitalnikom. Sistem integriranega vezja na osnovi mikroprocesorja in programske opreme je certificiran s splošnim informacijsko tehnološkim standardom varnosti CC EAL4+ (ISO/IEC 15408). Brezstična pametna kartica MIFARE DESFire EV1 lahko sočasno vsebuje 28 različnih aplikacij. Vsaka posamezna aplikacija pa lahko vsebuje največ 32 različnih datotek.

#### 3.1.1. LASTNOSTI KARTICE

##### 3.1.1.1. Fizične lastnosti kartice

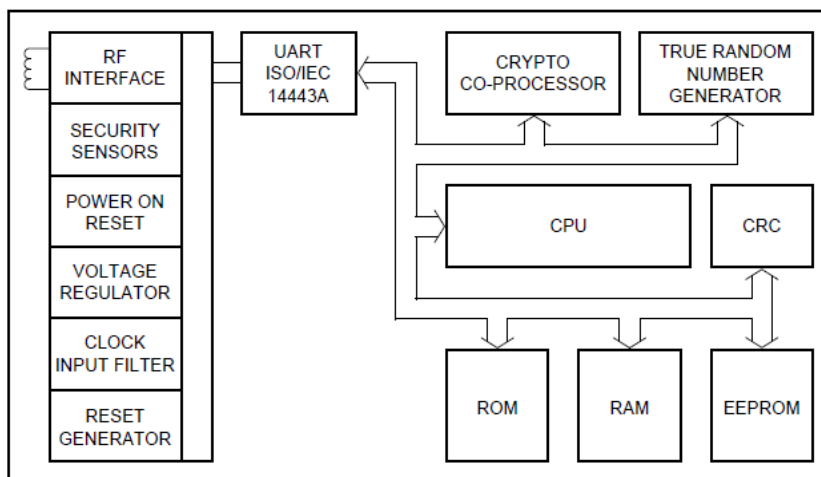
Fizične karakteristike pametne kartice so natančneje obravnavane v prvem delu ISO/IEC 14443 [1]. Dimenzije so definirane v standardu 7810 in znašajo 85,72 x 54,03 x 0,76 mm. Zraven dimenzij so obravnavane še druge lastnosti:

- Odpornost proti UV žarčenju
- Odpornost proti radioaktivnosti
- Odpornost proti upogibanje natančneje podaja standard ISO/IEC 10373
- Odpornost proti torzijskem upogibanju
- Odpornost proti izmeničnim in statičnim magnetnim valovanjem
- Odpornost proti izmeničnim in statičnim električnim poljem
- Delovna temperatura (0-50 °C)

## DS 4: Integracija IS sistemov prevoznika

### 3.1.1.2. Komponente na kartici

Na kartici se nahajajo vsi osnovni gradniki računalniškega mikroprocesorskega sistema, centralna procesna enota (CPU), ROM, RAM, EEPROM 8 kB, kripto procesor, generator naključnih števil, CRC modul, UART modul, RF vmesnik za komunikacijo s čitalnikom. Proizvajalec jamči za zapis 500.000 ciklov v EEPROM, kjer lahko podatki ostanejo do 10 let.



SLIKA 3.1: ZGRADBA KARTICE MIFARE DESFIRE

### 3.1.1.3. RF vmesnik

Elektro magnetne lastnosti so definirane v drugem delu ISO/IEC 14443 [1]. Za kartico MIFARE DESFire je značilen vmesnik tipa A, pri katerem se uporablja ASK modulacija in modificirano Millerjevo kodiranje za komunikacijo v smeri med čitalnikom in kartico ter Manchesterjevo fazno kodiranje v smeri med kartico in čitalnikom. Posledično kartica ne potrebuje lastnega napajanja, saj za kratek čas aktivnega delovanja energijo zagotavlja elektromagnetno RF valovanje. Druge pomembne lastnosti kartice so:

- Območje delovanja do 100 mm (odvisno od brezstičnega napajanja, ki ga zagotavlja čitalnik in zgradbe RF antene)
- Frekvenca delovanja: 13.56 MHz
- Hiter prenos podatkov: 106 kbit/s, 212 kbit/s, 424 kbit/s, 848 kbit/s

### 3.1.1.4. Pomnilnik

EEPROM 8 kB pomnilnik je organiziran s fleksibilnim datotečnim sistemom, ki omogoča največ 28 različnih aplikacij, kjer lahko posamezna aplikacija vsebuje do 32 različnih datotek. Vsaki aplikaciji je dodeljen unikaten identifikator AID (ang. Application IDentifier). Posamezne datoteke se lahko ustvarijo v fazi produkcije kartic ali na terenu na posameznem terminalu oz. čitalniku kartic. Če se določena datoteka ne uporablja več, se lahko odstrani. Določeni ukazi s katerimi lahko upravljamo s

## DS 4: Integracija IS sistemov prevoznika

---

vsebino in strukturo datotek omogočajo zaščito posameznih datotek z mehanizmom, ki omogoča, da posamezno datoteko povrne v prvotno stanje. Datotečni sistem omogoča naslednje tipe datotek:

- Standardna podatkovna datoteka
- Podatkovna datoteka z varnostno kopijo
- Datoteka numeričnih vrednosti z varnostno kopijo
- Linearna datoteka vnosov z varnostno kopijo
- Ciklična datoteka vnosov z varnostno kopijo

### 3.1.1.5. Varnostni mehanizmi

S pomočjo 7 B številke UID predstavlja vsaka posamezna kartica neodvisno entiteto v sistemu. Unikatna številka je vstavljena v vsako posamezno kartico v času proizvodnje in je kasneje ni mogoče spremeniti. Kartica omogoča varno izmenjavo podatkov s pomočjo naslednjih naprednih kriptografskih algoritmov:

- DES 56 bitni ključ
- 2K3DES, trojni DES z dvojnim 112 bitnim ključem
- 3K3DES, trojni DES s trojnim 168 bitnim ključem
- AES

**V sistemu IJPP se bo uporabljal 3K3DES, trojni DES s trojnim 168 bitnim ključem.**

### 3.1.2. GRAFIČNA PODOBA KARTIC DRUGIH PREVOZNIKOV

#### 3.1.2.1. Definirati način dodajanja manjkajočih elementov (dotisk, nalepke)

Znotraj grafične podobe prevoznikov, sta možne dve integraciji parametrov IJPP:

- Parametri se zajamejo znotraj celostne grafične podobe (CGP) prevoznika in se ob naslednjem tiskanju predvidi prostor za vse zahtevane parametre.
- Parametri se na obstoječe kartice, ki so trenutno v obtoku dodajo v obliki nalepke (logotip IJPP) ali se do tiskajo (lastniški parametri). Pri čemer je možno tudi te dodati na kartico v obliki nalepke.

Ne glede na način integracije kartice prevoznikov, morajo imeti le te prikazane lastniške parametre in logotip IJPP na jasn in berljiv način.

Pozicija parametrov mora biti usklajena in odobrena s strani upravljavca IJPP.

### 3.1.2.2. Definirati obvezne lastniške podatke

Na kartici morajo biti na viden in berljiv način ter v skladu s pravili CGP-ja sistema IJPP ter prevoznika, uvrščeni in prikazani naslednji podatki:

- Edinstvena številka brezstične pametne kartice - CUID
- Logotip IJPP
- Ime in priimek lastnika kartice v primeru imenske kartice

### 3.1.2.3. Definirati obliko CUID

CUID je 7 bajtna številka, ki predstavlja edinstveno številko brezstične kartice in je na kartici zapisana v zaklenjen del spomina, namenjenega izdelovalcu čipa.

Ko jo čitalnik bere, jo bere v dveh korakih znotraj antikolizijske zanke, najprej se preberejo bajti B1 do B3, nato pa še bajti B4 do B7:

0x04						
B1	B2	B3	B4	B5	B6	B7

Če je izdelovalec čipa NXP, bo prvi bajt, torej bajt B1 vseboval vrednost 0x04, ki pomeni identifikacijsko kodo izdelovalca. Pri branju lahko ta vrednost služi kot orientacija za pravilno nadaljnje formatiranje izpisa.

CUID se na kartico zapiše v desetiškem formatu tako, da najprej prebrane bajte B1 do B7 zložimo 64 bitno celo število tako, da je B1 najmanj pomemben bajt (LSB) v njem, B7 pa drugi najbolj pomemben bajt v njem. Najbolj pomemben bajt (MSB) – B8 pa se postavi na 0x00:

0x00							0x04
B8	B7	B6	B5	B4	B3	B2	B1

Tako oblikovan CUID nato pretvorimo v desetiško število in dobimo 17 mestno desetiško število, ki ga potem zapišemo na kartico.

#### Primer:

S kartice preberemo sledeči CUID

0x04	0x85	0xC2	0x12	0xA4	0x28	0x80
B1	B2	B3	B4	B5	B6	B7

Iz tega naredimo sledeče 64 bitno celo število zapisano v šestnajstiškem formatu:

#### DS 4: Integracija IS sistemov prevoznika

0x8028A412C28504

S pretvorbo v desetiški format dobimo:

36073482173449476

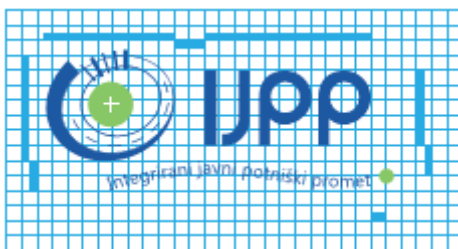
To je zapis, ki ga je potrebno zapisati na kartico.

#### 3.1.2.4. Definirati pozicijo oznake IJPP na karticah drugih prevoznikov

Pri umestitvi logotipa IJPP je potrebno upoštevati minimalni prazni prostor in minimalno velikost logotipa IJPP ter pravila o sodelovanju znamk.

Če je pobudnik (in plačnik!) komuniciranja eden od IJPP partnerjev - z pod-licenco ali licenco, če se pokaže kot ponudnik storitev IJPP- bo spoštovana njihova vizualna identiteta in pravila blagovne znamke. V tem primeru je potrebno vse osnovne standarde IJPP znamke (barve, razmerja, minimalen prazen prostor in velikost, itd), spoštovati - toda sam položaj znamke bo opredeljen s standardi blagovne znamke partnerja!

Prazen prostor okrog IJPP blagovne znamke mora biti vsaj 1X na obeh straneh, kjer je X višina črke O (Slika 3.2). Prazen prostor se prilagodi sorazmerno glede na velikost znamke. Sprejemljivo in običajno zaželeno je, da se omogoči več praznega prostora.



SLIKA 3.2: MINIMALNI PRAZNI PROSTOR IN VELIKOST, MREŽA 2 X 2 MM RAZMERJA

**Opomba:** Ne postavljajte nobene druge grafike, besedila ali slike v minimalno dovoljen prazen prostor.

Minimalna velikost IJPP blagovne znamke je 11mm širine za tiskanje in 44px širine za aplikacije na zaslonu.

**Opomba:** Nikoli ne reproducirajte znamke na velikosti manjše od velikosti navedene zgoraj.



### 3.2. ČITALNIK BREZSTIČNE PAMETNE KARTICE

NFC (near field communication) je tehnologija, ki omogoča brezstično komunikacijo med napravami, ki so v neposredni bližini. Prav zaradi nujno potrebne neposredne bližine med napravami je prisluškovanja izredno oteženo, kar ima očitne prednosti kar se tiče varstva podatkov. Naslednja prednost, ki je prav tako povezana z nujno neposredno bližino med posameznimi napravami je lastnost, da je možna komunikacija tudi med aktivno in pasivno napravo. Kar pomeni, da je lahko ena naprava brez lastnega napajanja in posledično za to poskrbi nasprotna naprava. Pasivna naprava se tako napaja s pomočjo induktivnosti in elektromagnetnega valovanja, ki služi za prenos komunikacije. NFC tehnologija omogoča tri različne načine komunikacije:

- **Način Branja/Pisanja**

V načinu branja in pisanja lahko naprava oz. čitalnik prejema podatke od druge naprave ali pa ji podatke pošilja. Ta način delovanja je značilen za čitalnik, ki lahko komunicira z brezstično pametno kartico, značko NFC (ang. NFC tag) ali z mobilnim telefonom, ki podpira tehnologijo NFC in je v načinu delovanja emuliranja brezstične pametne kartice (ang. Card Emulation Mode).

Pri tem načinu plačevanja je ključno, da naprava, ki vzpostavi iniciativo pri komunikaciji prevzame generiranje RF elektromagnetnega polja. Kar pomeni, da je druga naprava pasivna in za svoje delovanje ne potrebuje lastnega napajanja.

Za varno komunikacijo med napravami se lahko uporabi varnostni element (SAM - ang. Secure Access Module), ki služi kot kriptoprosesor in lahko poskrbi za avtentikacijo med napravami in kodiranjem podatkov pri nadaljnji komunikaciji.

- **Način Peer to Peer**

Ta način komunikacije se uporablja za vzpostavitev dvosmernega komunikacijskega kanala med dvema napravama NFC, kjer sta lahko obe napravi aktivni ali pa je ena naprava aktivna druga pa pasivna. Način komunikacije služi za kompleksnejše oblike prenosa podatkov.

- **Način emuliranja brezstične kartice**

V tem načinu naprava deluje na takšen način, da neposredno posnema delovanje brezstične pametne kartice. Takšen način delovanja je značilen za mobilne telefone, kar omogoča, da se lahko naprave uporabijo v obstoječih pametnih sistemih, ki delujejo na osnovi brezstičnih pametnih kartic. Prednost delovanja v tem načinu je majhna poraba energije, saj deluje kot pasivna naprava.

**V sistemu IJPP se bodo čitalniki uporabljali način delovanja Branje/Pisanje.** S tem bodo komunicirali z brezstičnimi pametnimi karticami, ki predstavljajo pasivne naprave. Pasivna komunikacija je natančneje definirana v drugem delu ISO/IEC 14443[1]:

## DS 4: Integracija IS sistemov prevoznika

---

- Pobudnik komunikacije oz. čitalnik generira elektromagnetno polje oz. 13.56 MHz nosilni signal. Elektromagnetno polje se uporablja za napajanje pasivne naprave.
- Prenos podatkov s strani čitalnika proti pasivni napravi oz. brezstični pametni kartici poteka z direktno amplitudno modulacijo (ASK) nosilnega signala.
- Prenos podatkov s strani brezstične pametne kartice v smeri čitalnika poteka s pomočjo modulacije bremena, kar se pozna na strani čitalnika v obliki padcev elektromagnetnega polja.

### 3.2.1. LASTNOSTI ČITALNIKA

Čitalnik omogoča s pomočjo modulacije elektromagnetnega polja omogoča komuniciranje po komunikacijskem protokolu ISO/IEC 14443A/MIFARE. Zgrajen je iz naslednjih komponent:

- **Antena**  
omogoča vzpostavitev ustreznega elektromagnetnega polja preko katerega se napaja pasivna naprava. S pomočjo modulacije elektromagnetnega poteka prenos podatkov.
- **Namensko integrirano vezje NFC**  
je vmesnik med RF anteno in nosilnim procesorjem, ki omogoča izvedbo ustreznih modulacij za pošiljanje in prejemanje podatkov po elektromagnetnemu nosilnemu signalu.
- **Procesna enota**  
zagotavlja komunikacijo z NFC čipom po protokolu ISO/IEC 14443.
- **Reža (podnožje) za varnostni element (SAM modul),**  
je namenjena neposredni povezavi in komunikaciji z varnostnim elementom.
- **Varnostni element (SAM modul)**  
je element, ki omogoča shrambo varnostnih ključev, avtentikacijo ter kodiranje podatkov.

### 3.3. KOMUNIKACIJA Z UKAZI

Komunikacija med kartico in čitalnikom poteka v obliki vnaprej dogovorjenih ukazov, ki so del dogovorjenih protokolov. Posamezni ukazi so definirani v tretjem in četrtem delu ISO/IEC 14443 [1] ter ISO/IEC 7816 [3]. Vsi ukazi, ki jih podpira MIFARE DESFire so zbrani v [5].

#### 3.3.1. KOMUNIKACIJSKI UKAZI ISO/IEC 14443-3

Tretji del ISO/IEC 14443 definira postopek inicializacije in antikolizijski (anti-collision) algoritem. Postopek inicializacije posamezne kartice pomeni izmenjavo ustreznih ukazov, ki omogočajo izmenjavo UID med kartico in čitalnikom. Antikolizijski postopek pa definira izmenjavo podatkov o UID in izbiro ustrezne kartice v primeru, da je v magnetnem polju sočasno prisotnih več kartic. Za uspešno izvedbo postopka inicializacije se uporabljajo naslednji ukazi:

- **REQA in ATQA**

Na ukaz REQA, ki ga pošlje čitalnik, kartica odgovori z ukazom ATQA v katerem sporoči velikost UID v bajtih in število kaskadnih operacij komunikacije za prenos celotnega UID-eja. V primeru velikosti UID-eja 7 B se UID prenese v dveh delih z dvema kaskadnima operacijama ANTICOLLISION/SELECT, SAK.

- **WUPA**

Ukaz WUPA se uporabi, če želimo s kartico, ki je v stanju mirovanja, ponovno komunicirati.

- **ANTICOLLISION in SELECT**

Po poslanem ukazu ANTICOLLISION kartica odgovori z UID-ejem. V primeru, da je prišlo do kolizije čitalnik ponovno pošlje ukaz ANTICOLLISION, ki vsebuje prejeti del UID-eja. Kartica nato pošlje še preostali del UID-eja. Ko čitalnik prejme celoten UID kartici pošlje ukaz SELECT, ki ga kartica potrdi s SAK. V primeru velikosti UID-eja 7 B se UID prenese v dveh delih. Posledično se naslednji del UID-eja ponovi postopek pošiljanja ukazov ANTICOLLISION, SELECT in SAK. 7 B UID se prejme v dveh delih po 4 B, kjer ima v prvem delu prvi bajt vrednost 0x88, sledijo 3 B UID. V drugem delu sledijo preostali 4 B UID številke.

- **HALT**

Z uporabo HALT ukaza kartica preide v stanje, kjer se ne odziva na ukaze. Če želimo ponovno komunicirati s kartico moramo uporabiti ukaz WUPA.

### 3.3.2. KOMUNIKACIJSKI UKAZI ISO/IEC 14443-4

Četrty del ISO/IEC 14443 obravnava izmenjavo informacij povezanih z najrazličnejšimi parametri, ki definirajo lastnosti komunikacije med kartico in čitalnikom. Postopek izmenjave se prične po uspešni izbiri kartice s pošiljanjem ukaza RATS s strani čitalnika. Za nastavitve posameznih parametrov se uporabljajo naslednji ukazi:

- **RATS**

Omogoča pridobitev podatkov o trenutnih parametrih v paketu ATS s kartice.

- **PPS**

Omogoča nastavitve posameznih parametrov hitrosti komunikacije s strani čitalnika.

- **WTX**

Se pošlje v primeru, da kartica potrebuje več časa za odgovor kot je definirano v parametrih.

- **DESELECT**

Se pošlje z namenom prekinitve komunikacije s kartico, ki preide v stanje HALT.

### 3.3.2.1. Ukaz RATS in ATS

Z ukazom RATS čitalnik zahteva odgovor od kartice v zaporedju več bajtov ATS. V ukazu RATS čitalnik pošlje informacijo o najdaljšem možnem okvirju, ki ga lahko prejme. V odgovor na RATS kartica posreduje ATS paket, ki ga sestavljajo naslednji bajti:

- **TL** - predstavlja dolžino paketa
- **T0** - se nahajajo informacije o prisotnosti bajtov ki sledijo TA, TB in TC ter informacijo o maksimalni dolžini okvirja, ki ga kartica lahko sprejme.
- **TA** - v primeru ustrezne informacije v T0 je TA prisoten. Bajt TA vsebuje informacije o hitrosti komunikacije v obeh smereh. Ter informacijo o možnosti, da informacija poteka v različnih smereh z različnima hitrostma.
- **TB** - v primeru ustrezne informacije o pristnosti v T0 je TB prisoten. TB vsebuje vrednost FWT, ki predstavlja največji dovoljen čas po prejetem okvirju od čitalnika v katerem mora kartica začeti komunikacijo. TB vsebuje še vrednost SFGT, ki predstavlja čas, ki ga kartica potrebuje po poslanem ATS, da se pripravi na komunikacijo.
- **TC** - vsebuje še dodatne parametre.
- **T1-Tk** - predstavljajo zgodovinske bajte, ki so predstavljeni v ISO/IEC 7816-4.

### 3.3.2.2. Ukaz PPS zahteva in PPS odgovor

Ukaz PPS se uporablja za nastavitev hitrosti komunikacije med čitalnikom in kartico. Kartica MIFARE DESFire EV1 omogoča, da komunikacija v obeh smereh poteka pri različnih hitrostih. Paket PPS sestavljajo:

- **PPSS** - start bajt,
- **PPS0** - ki podaja informacijo o prisotnosti naslednjega bajta PPS1,
- **PPS1** - vsebuje informacije o hitrostih komunikacije v posameznih smereh.

### 3.3.3. VARNOSTNI UKAZI, UKAZI ZA UPRAVLJANJE S KARTICO, APLIKACIJAMI IN PODATKI

Pri komunikaciji se uporabljajo naslednji varnostni ukazi:

- **Authenticate**  
Pri postopku avtentikacije kartica in čitalnik izmenjata kodirane podatke, s pomočjo katerih verificirata uporabo istega ključa. Po kartica in čitalnik generirata nov ključ – t.i ključ seje (ang. Session key), ki se nato uporablja pri nadaljnji komunikaciji.
- **Change Key Settings**

## DS 4: Integracija IS sistemov prevoznika

---

Change Key Seettings se uporablja za spremembo nastavitev »master« ključa na kartici in na nivoju aplikacije.

- **Set Configuration**

Set Configuration se uporablja za konfiguracijo kartice in sicer ali se pošilja UID ali naključen ID v začetni fazi komunikacije ter konfigurira vsebino ATS paketa.

- **Change Key**

Change Key zamenja ključ na kartici.

- **Get Key Versio**

Get Key Version prebere verzijo poljubnega ključa shranjenega na kartici.

Naslednji ukazi se uporabljajo za upravljanje s kartico:

- **Create Application**

Create Application ustvari novo aplikacijo na kartici.

- **Delete Application**

Delete Application trajno deaktivira aplikacijo na kartici.

- **Get Applications IDs**

Get Application IDs vrne vse IDje vseh aplikacij, ki so na kartici.

- **Free Memory**

Free Memory vrne podatek o prostem pomnilniku na kartici.

- **GetDFNames**

GetDFNames vrne imena datotek, ki vsebujejo aplikacije.

- **Get KeySettings**

Get KeySettings posreduje informacije o nastavitvah ključa na kartici in master aplikacije. Vrne tudi maksimalno število ključev, ki so nastavljeni za izbrano aplikacijo.

- **Select Application**

Select Application izbere specifično aplikacijo za nadaljnji dostop do podatkov v aplikaciji.

- **Format MF3ICD81**

Format MF3ICD81 sprosti uporabniški spomin na MF3ICD81 .

- **Get Version**

Get Version vrne podatke, ki so vezani na proizvodnjo kartice.

- **GetCardUID**

GetCardUID vrne UID kartice.

Nadaljnji ukazi se uporabljajo pri komunikaciji s posamezno aplikacijo:

- **Get FileIDs**

Get FileIDs vrne vse IDje datotek, ki so aktivne znotraj izbrane aplikacije.

## DS 4: Integracija IS sistemov prevoznika

---

- **Get FileSettings**  
Get FileSettings vrne informacije o določeni datoteki.
- **Change FileSettings**  
Change FileSettings spremeni parametre za dostop obstoječe datoteke.
- **Create StdDataFile**  
Create StdDataFile ustvari standardne podatkovne datoteke za shranjevanje uporabniških podatkov znotraj obstoječe aplikacije.
- **Create BackupDataFile**  
Create BackupDataFile ustvari podatkovne datoteke za shranjevanje uporabniških podatkov znotraj obstoječe aplikacije, ki omogočajo mehanizme za varnostno kopijo.
- **Create ValueFile**  
Create ValueFile ustvari datoteke numeričnih vrednosti (32 bitne predznačena cela števila) za shranjevanje uporabniških podatkov znotraj obstoječe aplikacije.
- **Create LinearRecordFile**  
Create LinearRecordFile ustvari linearne datoteke vnosov za večkratno shranjevanje strukturiranih uporabniških podatkov znotraj obstoječe aplikacije. Ko je datoteka polna, jo je za vnovično zapisovanje potrebno zbrisati.
- **Create CyclicRecordFile**  
Create CyclicRecordFile ustvari ciklične datoteke vnosov za večkratno shranjevanje strukturiranih uporabniških podatkov znotraj obstoječe aplikacije. Ko je datoteka polna, kartica začne vpisovati na mesto najstarejšega vnosa.
- **DeleteFile**  
DeleteFile trajno deaktivira datoteko znotraj direktorija izbrane aplikacije.

Naslednji ukazi za uporabljajo za manipulacijo s podatki znotraj posamezne datoteke.

- **Read Data**  
Read Data bere podatke iz standardne podatkovne datoteke ali iz podatkovne datoteke z varnostno kopijo.
- **Write Data**  
Write Data zapisuje podatke v standardne podatkovne datoteke ali v podatkovne datoteke z varnostno kopijo.
- **Get Value**  
Get Value prebere trenutno shranjeno vrednost iz datoteke numerične vrednosti.
- **Credit**  
Credit poveča vrednost shranjeno v datoteki numerične vrednosti.
- **Debit**  
Debit zmanjša vrednost shranjeno v datoteki numerične vrednosti.
- **Limited Credit**  
Limited Credit omogoča omejeno povečanje vrednosti v datoteki numerične vrednosti brez polnih pooblastil.
- **Write Record**

## DS 4: Integracija IS sistemov prevoznika

---

Write Record zapisuje podatke v datoteke linearnih ali cikličnih vnosov.

- **Read Records**

Read Records bere podatke iz datotek linearnih ali cikličnih vnosov.

- **Clear RecordFile**

Clear RecordFile ponastavi datoteko linearnih ali cikličnih vnosov na prazno.

- **Commit Transaction**

Commit Transaction potrdi vse predhodne operacije zapisovanj v datoteke znotraj aplikacije.

- **Abort Transaction**

Abort Transaction prekliče vse predhodne operacije zapisovanja v datoteke znotraj aplikacije.

Vsi do sedaj obravnavani ukazi v tem podpoglavju se pošiljajo v obliki bločnega formata, ki je definiran v četrtem delu ISO/IEC 14443. Sledijo pa ukazi, ki so definirani v četrtem delu ISO/IEC 7816:

- **INS code 'A4' SELECT**

se pošlje z namenom izbire specifične datoteke znotraj izbrane aplikacije. Čitalnik znotraj ukaza posreduje ustrezne informacije (ID datoteke, ime datoteka ali pot do datoteke) o datoteki, do katere želi dostopati.

- **INS code 'B0' READ BINARY**

v ukazu čitalnik posreduje informacije o natančnem mestu in dolžini podatkov, ki jih zahteva od kartice.

- **INS code 'D6' UPDATE BINARY**

z ukazom se posodobijo podatki na določeni lokaciji točno določene dolžine s tem, da se primerjajo z obstoječimi podatki in se spremenijo na zahtevano vrednost s strani čitalnika.

- **INS code 'B2' READ RECORDS**

se uporablja za branje linearnih ali cikličnih datotek vnosov.

- **INS code 'E2' APPEND RECORD**

se uporablja za dodajanje vnosov v linearne ali ciklične datoteke vnosov.

- **INS code '84' GET CHALLENGE**

se uporablja za pridobitev naključnega števila, ki se uporablja za postopek kriptografske avtentikacije.

- **INS code '88' INTERNAL AUTHENTICATE**

se uporablja za avtentikacijo.

- **INS code '82' EXTERNAL AUTHENTICATE**

se uporablja za avtentikacijo.

### 3.4. POSTOPKI KOMUNIKACIJE MED ČITALNIKOM IN KARTICO

## DS 4: Integracija IS sistemov prevoznika

---

Postopki komunikacije med čitalnikom in kartico se sprožijo kadar prislonimo kartico v neposredno bližino čitalnika. Posamezni postopki branja in pisanja se razlikujejo od tipa operacije, ki jo želimo izvesti. V nadaljevanju bomo predstavili naslednje kartične operacije:

- Protokol komunikacije pri prijavi
- Protokol komunikacije pri branju informacij o obstoječih vozovnicah na kartici
- Protokol komunikacije pri nakupu vozovnice z gotovino
- Protokol komunikacije pri validaciji/aktivaciji obstoječe vozovnice
- Protokol komunikacije pri preverjanju vozovnice
- Protokol komunikacije pri stornaciji vozovnice
- Protokol komunikacije ob popolnem vračilu – izplačilu vozovnice
- Protokol komunikacije ob delnem vračilu – delnem izplačilu vozovnice
- Protokol komunikacije ob blokadi aplikacije IJPP
- Protokol komunikacije ob blokadi vozovnice
- Protokol komunikacije ob deblokadi aplikacije IJPP
- Protokol komunikacije ob deblokadi vozovnice

### *3.4.1. PROTOKOL KOMUNIKACIJE PRI PRIJAVI*

Po vklopu terminala je za nadaljnjo uporabo potrebno opraviti prijavo z ustrezno identifikacijsko kartico. Ko identifikacijsko kartico približamo k terminalu, se podatki o kartici preberejo in posledično shranijo v podatkovno strukturo. Posledično lahko preverimo ustreznost prislonjene identifikacijske kartice in izvršimo prijavo. Odjavo pa lahko v terminalu izvedemo bodisi z izbiro ustrezne funkcije na terminalu bodisi s ponovno prislonitvijo identifikacijske kartice in s postopkom, ki je identičen postopku prijave.

Cilj postopka komunikacije pri prijavi je ustrezen protokol komunikacije med čitalnikom in kartico. Ustrezen protokol omogoča, da se iz ustrezne kartice preberejo in zapišejo v ustrezno podatkovno strukturo na terminalu ustrezne informacije o identifikacijski kartici. Postopek branja informacij o kartici je razdeljen v različna zaporedna stanja. V posameznih stanjih se izvajajo določeni postopki branja posameznih datotek na kartici.

#### **3.4.1.1. Stanje pripravljenosti**



## DS 4: Integracija IS sistemov prevoznika

---

- **Akcije**

1. V stanju pripravljenosti se opravijo različni inicializacijski postopki podatkovne strukture, ki se uporablja pri procesu.
2. V strukturo se zapiše **Čas transakcije**
3. V strukturo se zapiše **UID kartice**  
UID se prebere iz kartice s postopkom izbire oz. antikolizije. Tako je UID pripravljen takoj, ko je kartica pripravljena za nadaljnja branja in pisanja podatkov.

Sledi operacija izbire ustrezne aplikacije (AID) s katero želi čitalnik komunicirati čitalnik pošlje ukaz:

- **Ukaz:** Select Application
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešna izbira aplikacije.

### 3.4.1.2. Stanje izberi aplikacijo

- **Akcije**

1. V stanju izberi aplikacijo čitalnik pošlje ukaz za avtentikacijo.
2. Postopek avtentikacije.

- **Ukaz:** Authenticate

Aktivira se postopek avtentikacije med čitalnikom in kartico.

- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešna avtentikacija med čitalnikom in kartico.

### 3.4.1.3. Stanje avtentikacija

V stanju avtentikacija se po uspešno zaključeni avtentikaciji lahko prične postopek branja in pisanja v zelenih datotekah.

- **Akcije**

Branje verzije kartice oz. datoteke INFO.

- **Ukaz:** Read Data

Z ukazom aktiviramo postopek branja datoteke INFO na kartici, ki je namenjena informacijam o verziji kartice.

- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje datoteke INFO.

#### 3.4.1.4. Stanje branje verzije (datoteka INFO)

- **Akcije:**
  1. V stanju branje verzije shranimo podatke o verziji kartice v ustrezno podatkovno strukturo.
  2. Branje osebnih podatkov oz. datoteke CARD INFO.
- **Pogoji:**

Primerjamo prebrane podatke o zapisani verziji na kartici z želeno verzijo kartice za nadaljnjo komunikacijo.

  1. Če verzija trenutne kartice ni ustrezna, prekinemo operacijo nakupa vozovnice in obvestimo uporabnika o nastali napaki.
  2. Če je verzija ustrezna nadaljujemo nakup vozovnice.
- **Ukaz:** Read Data  
Z ukazom aktiviramo postopek branja datoteke CARD INFO, ki vsebuje splošne informacije o kartici in osebne podatke o imetniku kartice.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje datoteke CARD INFO.

#### 3.4.1.5. Stanje branje info (datoteka CARD INFO)

- **Akcije:**
  1. V tem stanju shranimo splošne informacije o kartici in imetniku kartice v ustrezno podatkovno strukturo.
  2. Branje datoteke DATA INDEX.
- **Pogoji:**

Preverimo veljavnost kartice. Če ni veljavna prekinemo operacijo nakupa in uporabnika obvestimo o poteku veljavnosti kartice.
- **Ukaz:** Read Data  
Z ukazom aktiviramo postopek branja indeksa posameznih statusov oz datoteke DATA INDEX.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje datoteke DATA INDEX.

#### 3.4.1.6. Stanje branje indeksov (datoteka DATA INDEX)

## DS 4: Integracija IS sistemov prevoznika

---

- **Akcije:**
  1. V stanju shranimo informacije o mestu in dolžini zapisa podatkov, ki predstavljajo posamezni status, ki je zapisan v datoteki DATA INDEX na kartici.
  2. Branje datoteke DATA FILE.
- **Ukaz: Read Data**

Z ukazom aktiviramo branje posameznega statusa na kartici oz. datoteka DATA FILE.  
V primeru, da je na kartici več statusov, preberemo vse statute.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešno branje datoteke DATA FILE.

### 3.4.1.7. Stanje branje statusa (datoteka DATA FILE)

- **Akcije:**

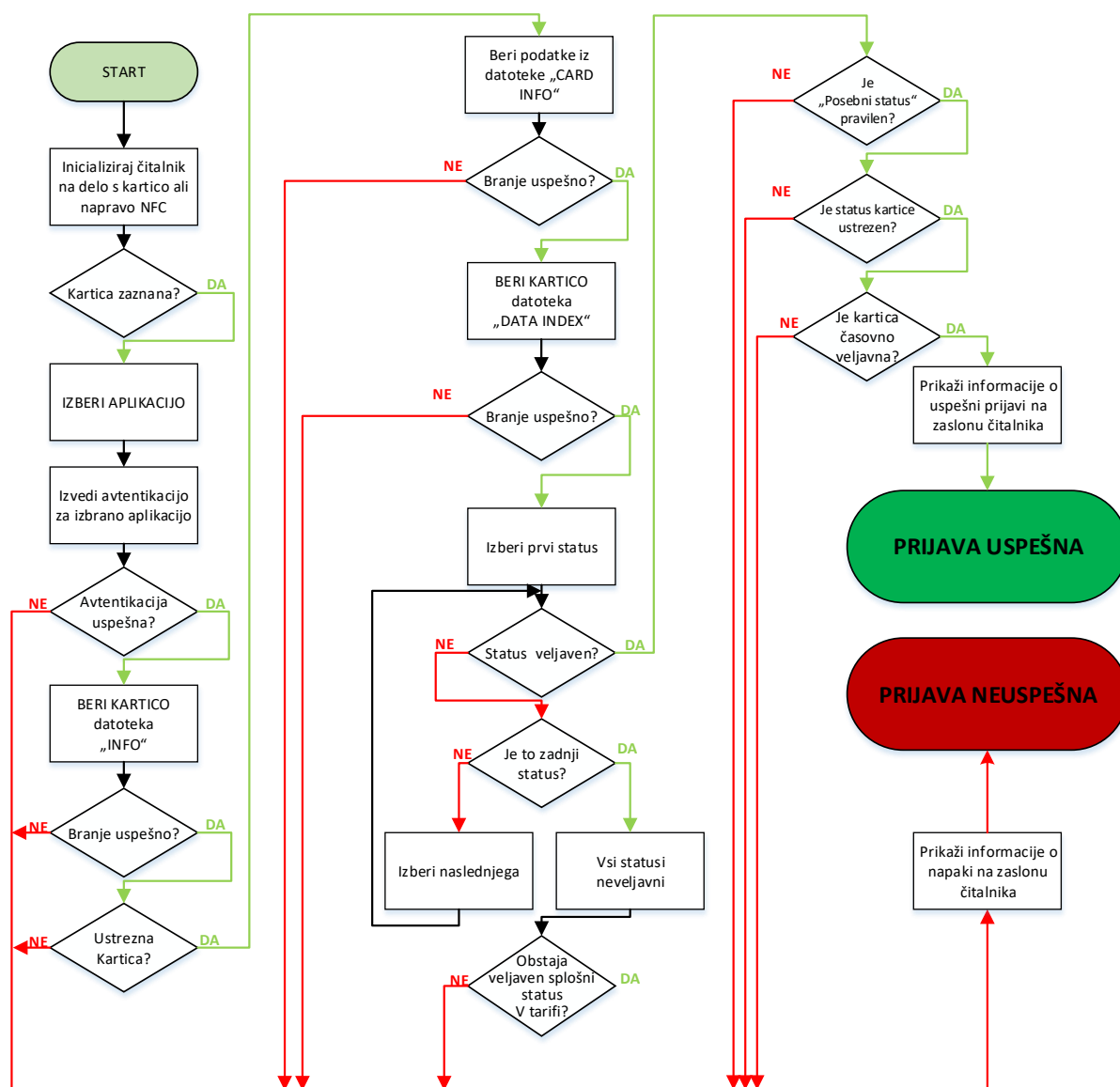
V stanju shranimo ustrezni status v podatkovno strukturo. V primeru, da je na kartici več statusov, je potrebno pregledati vse statute na kartici in izbrati ustreznega.
- **Pogoji:**

V primeru, da so na kartici neveljavni statusi, nastavimo splošni status, ki ga preberemo iz tarife na čitalniku. Ob branju splošnega statusa iz tarife preverimo ustreznost tarife.

  1. Če tarife ni ustrezen prekinemo operacijo nakupa vozovnice in uporabnika obvestimo o napaki.
  2. Preverimo ustreznost statusa kartice. Če je neustrezen prekinemo postopek uporabnika obvestimo o napaki.
  3. Preverimo ustreznost parametra posebnega statusa kartice (vrsta kartice) v datoteki CARD INFO. Če je neustrezen prekinemo postopek in uporabnika obvestimo o napaki.

S stanjem **Stanje branje statusa (datoteka DATA FILE)** se postopek prijave konča, če kartica ustreza pogojem identifikacijske kartice, postopek končamo. Na zaslonu terminala prikažemo sporočilo, da je prijava uspešna.

## DS 4: Integracija IS sistemov prevoznika



SLIKA 3.3 POSTOPEK PRIJAVE

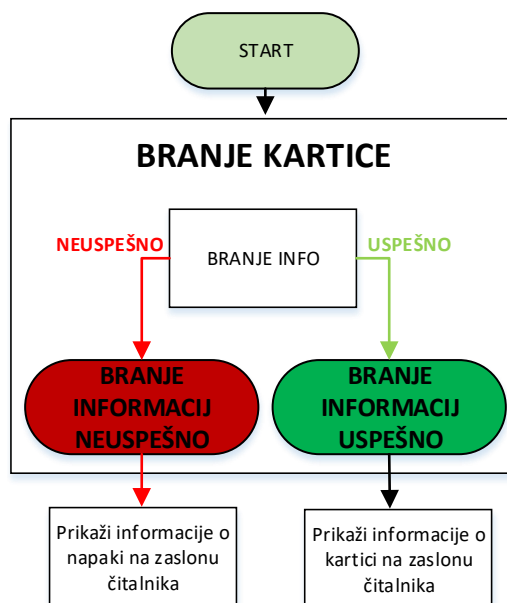
### 3.4.2. PROTOKOL KOMUNIKACIJE MED ČITALNIKOM IN KARTICO PRI BRANJU INFORMACIJ O OBSTOJEČIH VOZOVNICAH NA KARTICI

Na terminalu aktiviramo aplikacijo branje Info in približamo kartico za katero želimo prikazati informacije k terminalu. Po uspešnem postopku se informacije o kartici in o vozovnicah na kartici shranijo v ustrezno podatkovno strukturo, ki jih nato lahko prikažemo na zaslonu. V nadaljevanju je natančneje predstavljen protokol komunikacije.

Cilj postopka komunikacije pri pregledovanju informacij o kartici in o obstoječih vozovnicah je ustrezen protokol komunikacije med čitalnikom in kartico. Ustrezen protokol omogoča, da se iz ustrezne kartice

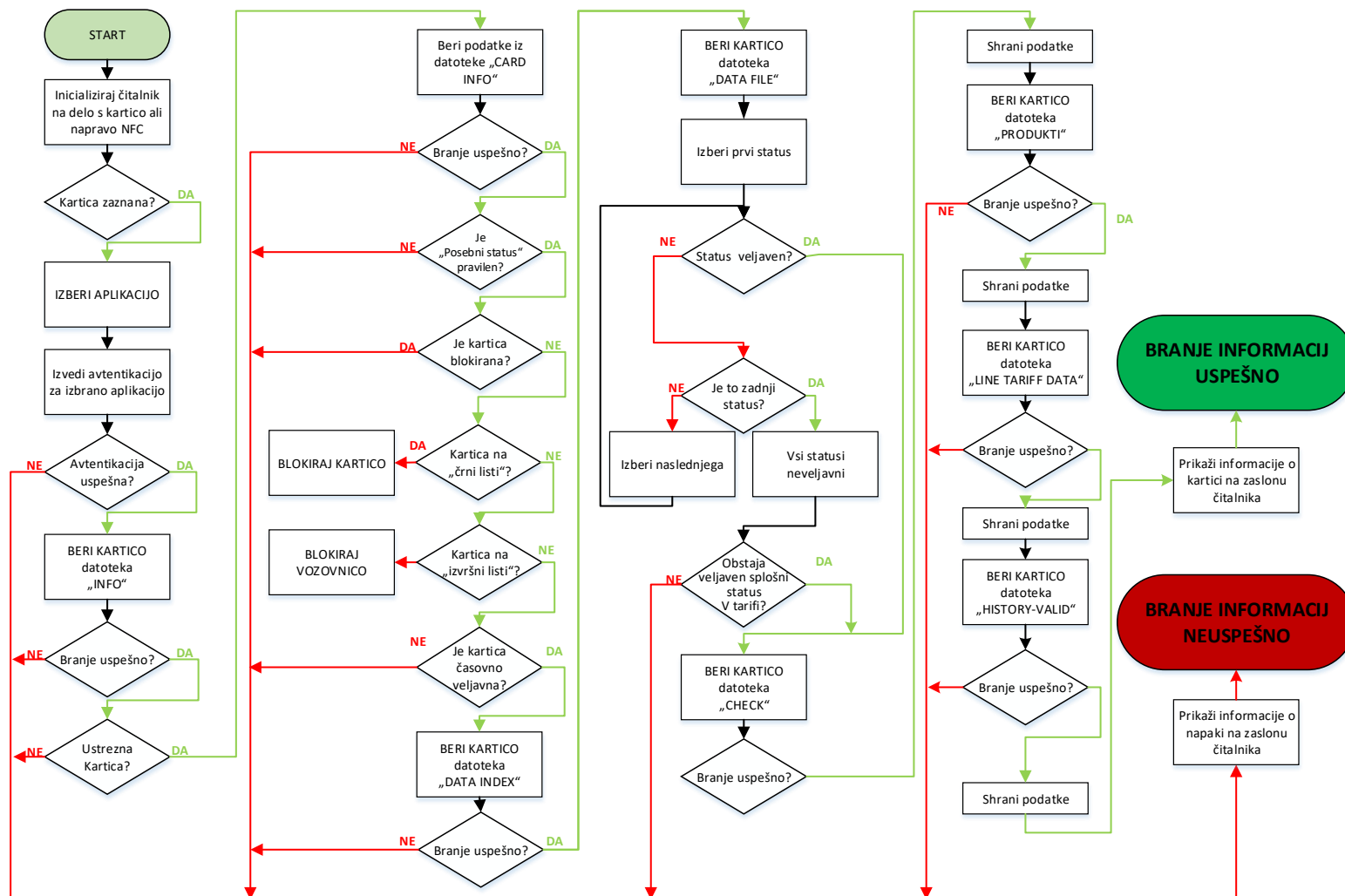
#### DS 4: Integracija IS sistemov prevoznika

preberejo in zapišejo v ustrezno podatkovno strukturo na terminalu informacije o vseh obstoječih vozovnicah na kartici. Postopek branja informacij o kartici in o obstoječih vozovnicah je razdeljen v različna zaporedna stanja. V posameznih stanjih se izvajajo določeni postopki branja posameznih datotek na kartici. Ko se podatki uspešno prebrani in posledično shranjeni v podatkovni strukturi, lahko informacije prikažemo na zaslonu terminala.



SLIKA 3.4 BRANJE INFO KARTICE – SPLOŠEN PREGLED

#### DS 4: Integracija IS sistemov prevoznika



SLIKA 3.5: BRANJE INFO KARTICE

### 3.4.2.1. Stanje pripravljenosti

- **Akcije**

1. V stanju pripravljenosti se opravijo različni inicializacijski postopki podatkovne strukture, ki se uporablja pri procesu.
2. V strukturo se zapiše **Čas transakcije**
3. V strukturo se zapiše **UID kartice**  
UID se prebere iz kartice s postopkom izbire oz. antikolizije. Tako je UID pripravljen takoj, ko je kartica pripravljena za nadaljnja branja in pisanja podatkov.

Sledi operacija izbire ustrezne aplikacije (AID) s katero želi čitalnik komunicirati čitalnik pošlje ukaz:

- **Ukaz:** Select Application
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešna izbira aplikacije.

### 3.4.2.2. Stanje izbiri aplikacijo

- **Akcije**

1. V stanju izberi aplikacijo čitalnik pošlje ukaz za avtentikacijo.
2. Postopek avtentikacije.

- **Ukaz:** Authenticate

Aktivira se postopek avtentikacije med čitalnikom in kartico.

- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešna avtentikacija med čitalnikom in kartico.

### 3.4.2.3. Stanje avtentikacija

V stanju avtentikacija se po uspešno zaključeni avtentikaciji lahko prične postopek branja in pisanja v želenih datotekah.

- **Akcije**

Branje verzije kartice oz. datoteke INFO.

- **Ukaz:** Read Data

Z ukazom aktiviramo postopek branja datoteke INFO na kartici, ki je namenjena informacijam o verziji kartice.

- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje datoteke INFO.

#### 3.4.2.4. Stanje branje verzije (datoteka INFO)

- **Akcije:**

1. V stanju branje verzije shranimo podatke o verziji kartice v ustrezno podatkovno strukturo.
2. Branje osebnih podatkov oz. datoteke CARD INFO.

- **Pogoji:**

Primerjamo prebrane podatke o zapisani verziji na kartici z želeno verzijo kartice za nadaljnjo komunikacijo.

1. Če verzija trenutne kartice ni ustrezna, prekinemo operacijo nakupa vozovnice in obvestimo uporabnika o nastali napaki.
2. Če je verzija ustrezna nadaljujemo nakup vozovnice.

- **Ukaz:** Read Data

Z ukazom aktiviramo postopek branja datoteke CARD INFO, ki vsebuje splošne informacije o kartici in osebne podatke o imetniku kartice.

- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešno branje datoteke CARD INFO.

#### 3.4.2.5. Stanje branje info (datoteka CARD INFO)

- **Akcije:**

1. V tem stanju shranimo splošne informacije o kartici in imetniku kartice v ustrezno podatkovno strukturo.
2. Branje datoteke DATA INDEX.

- **Pogoji:**

1. Če je STATUS kartice blokirana ali ni aktivirana vrnemo napako in prekinemo operacijo nakupa vozovnice.
2. Preverimo, če se kartica nahaja na črni listi. Če je kartica na črni listi sprožimo postopek blokade kartice.
3. Preverimo, če se kartica nahaja na izvršni listi. Če je kartica na črni listi, sprožimo postopek blokade vozovnice.
4. Preverimo ustreznost parametra POSEBNI STATUS. Če ni ustrezen prekinemo operacijo nakupa vozovnice in uporabnika obvestimo o napaki.
5. Preverimo veljavnost kartice. Če ni veljavna prekinemo operacijo nakupa in uporabnika obvestimo o poteku veljavnosti kartice.



## DS 4: Integracija IS sistemov prevoznika

---

Če so zgornji pogoji ustrezni in se postopek nakupa vozovnice lahko nadaljuje, operacijo nadaljujemo z branjem indeksa statusov na kartici.

- **Ukaz:** Read Data  
Z ukazom aktiviramo postopek branja indeksa posameznih statusov oz datoteke DATA INDEX.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje datoteke DATA INDEX.

### 3.4.2.6. Stanje branje indeksov (datoteka DATA INDEX)

- **Akcije:**
  1. V stanju shranimo informacije o mestu in dolžini zapisa podatkov, ki predstavljajo posamezni status, ki je zapisan v datoteki DATA INDEX na kartici.
  2. Branje datoteke DATA FILE.
- **Ukaz:** Read Data  
Z ukazom aktiviramo branje posameznega statusa na kartici oz. datoteka DATA FILE.  
V primeru, da je na kartici več statusov, preberemo vse statute.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje datoteke DATA FILE.

### 3.4.2.7. Stanje branje statusa (datoteka DATA FILE)

- **Akcije:**
  1. V stanju shranimo ustrezni status v podatkovno strukturo. V primeru, da je na kartici več statusov, je potrebno pregledati vse statute na kartici in izbrati ustreznega.
  2. Branje datoteke CHECK.
- **Pogoji:**

V primeru, da so na kartici neveljavni statusi, nastavimo splošni status, ki ga preberemo iz tarife na čitalniku. Ob branju splošnega statusa iz tarife preverimo ustreznost tarife.

  - Če tarife ni ustrezen prekinemo operacijo nakupa vozovnice in uporabnika obvestimo o napaki.
- **Ukaz:** Read Data  
Z ukazom aktiviramo branje kontrolnih podatkov oz datoteke CHECK.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**

## DS 4: Integracija IS sistemov prevoznika

---

Uspešno branje datoteke CHECK.

### 3.4.2.8. Stanje branje kontrolnih podatkov (datoteka CHECK)

- **Akcije:**
  1. Shranimo kontrolne podatke v strukturo.
  2. Branje dela datoteke PRODUCTS, kjer so zapisani statusi posameznih vozovnic.
- **Ukaz:** Read Value  
Z ukazom aktiviramo branje statusov obstoječih vozovnic na kartici.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje podatkov o statusih posameznih vozovnic.

### 3.4.2.9. Stanje branje statusov vozovnic (datoteka PRODUCTS)

- **Akcije:**
  1. Shranimo statuse posameznih vozovnic, ki so prisotne na kartici.
  2. Branje informacij o vozovnicah oz. preostanek datoteke PRODUCTS. Beremo le tisti del datoteke, ki ustreza veljavnim vozovnicam.
- **Ukaz:** Read Data  
Z ukazom aktiviramo branje vozovnic prisotnih na kartici.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje podatkov o vozovnicah v datoteki PRODUCTS.

### 3.4.2.10. Stanje branje vozovnic (datoteka PRODUCTS)

- **Akcije:**  
Shranimo posamezne prebrane informacije o vozovnicah v podatkovno strukturo.
- **Pogoji:**
  1. Če se na kartici nahajajo relacijske vozovnice, nadaljujemo z branjem informacij o relacijah oz. datoteke LINE TARIFF DATA.
  2. Če se na kartici ne nahajajo relacijske vozovnice nadaljujemo z branjem zgodovine validacij oz. datoteke HISTORY-VALID.
- **Ukaz:** Read Data

#### DS 4: Integracija IS sistemov prevoznika

---

Z ukazom aktiviramo branje datoteke LINE TARIFF DATA (aktiviramo branje statusov relacij) ali datoteke HISTORY-VALID

- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešno branje podatkov.

##### 3.4.2.11. Stanje branje statusov relacij (datoteka LINE TARIFF DATA)

- **Akcije:**

1. Shranimo statuse posameznih relacij, ki so prisotne na kartici.
2. Branje informacij o relacijah oz. preostanek datoteke LINE TARIFF DATA. Beremo le tisti del datoteke, ki ustreza veljavnim relacijam.

- **Ukaz:** Read Data

Z ukazom aktiviramo branje ustreznih relacij prisotnih na kartici.

- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešno branje podatkov o relacijah v datoteki LINE TARIFF DATA.

##### 3.4.2.12. Stanje branje relacij (datoteka LINE TARIFF DATA)

- **Akcije:**

Shranimo posamezne prebrane informacije o relacijah v podatkovno strukturo.

- **Ukaz:** Read Data

Z ukazom aktiviramo branje zgodovine validacij oz. datoteke HISTORY-VALID.

- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešno branje podatkov o zgodovini validacij oz. datoteke HISTORY-VALID.

##### 3.4.2.13. Stanje branje zgodovine validacij (datoteka HISTORY-VALID)

- **Akcije:**

Shranimo informacije o zgodovini validacij v podatkovno strukturo.

S stanjem **Stanje branje zgodovine validacij (datoteka HISTORY-VALID)** se protokol BRANJE INFO KARTICE konča. Po uspešno izvedenem postopku branja prikažemo informacije o kartici in o vozovnicah na zaslonu.

### 3.4.3. PROTOKOL KOMUNIKACIJE MED ČITALNIKOM IN KARTICO PRI NAKUPU VOZOVNICE Z GOTOVINO

Cilj postopka nakupa vozovnice je ustrezen protokol komunikacije med čitalnikom in kartico, ki omogoči, da se na ustrezno kartico zapiše izbrana vozovnica. Postopek nakupa je razdeljen v različna zaporedna stanja. V posameznih stanjih se izvajajo določeni postopki branja ali pisanja v določene datoteke na kartici. Pred nadaljnjimi operacijami nakupa vozovnice se izvede postopek branje info (Slika 3.4). S tem s kartice preberemo vse potrebne informacije o kartici in o obstoječih vozovnicah na kartici. Po izbiri zelenega produkta, postopek nakupa vozovnice nadaljujemo po sledečem protokolu.

#### 3.4.3.1. Stanje branje zgodovine validacij (datoteka HISTORY-VALID)

- **Akcije:**
  1. Shranimo zgodovino validacij v podatkovno strukturo.
  2. Aktiviramo zapis statusa nove vozovnice.
- **Ukaz: Write Data**

Z ukazom aktiviramo zapis statusa nove vozovnice, ki jo želimo zapisati datoteko PRODUCTS.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešen zapis podatkov na kartico.

#### 3.4.3.2. Stanje zapis statusa (datoteka PRODUCTS)

V stanju aktiviramo ukaz za zapis podatkov vozovnice.

- **Ukaz: Write Data**

Z ukazom aktiviramo zapis preostalih informacij o produktu oz. vozovnici, ki jo želimo zapisati v datoteko PRODUCTS.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešen zapis podatkov na kartico.

#### 3.4.3.3. Stanje zapis podatkov vozovnice (datoteka PRODUCTS)

Če želimo zapisati na kartico še kakšno vozovnico nadaljujemo z zapisovanjem statusa naslednje vozovnice in nadaljujemo v stanju zapis statusa. V primeru, da smo na kartico uspešno zapisali vse vozovnice pa nadaljujemo z zapisovanjem naključnega števila.

- **Ukaz: Write Data**

## DS 4: Integracija IS sistemov prevoznika

---

Z ukazom aktiviramo zapis naključnega števila v podatkovno datoteko CHECK.

- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.

### 3.4.3.4. Stanje zapis naključnega števila (datoteka CHECK)

Po uspešnem zapisu naključnega števila na kartico zapišemo še čas transakcije.

- **Ukaz:** Write Data  
Z ukazom aktiviramo postopek zapisa časa transakcije v datoteko CHECK.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.

### 3.4.3.5. Stanje zapis časa zadnje transakcije (datoteka CHECK)

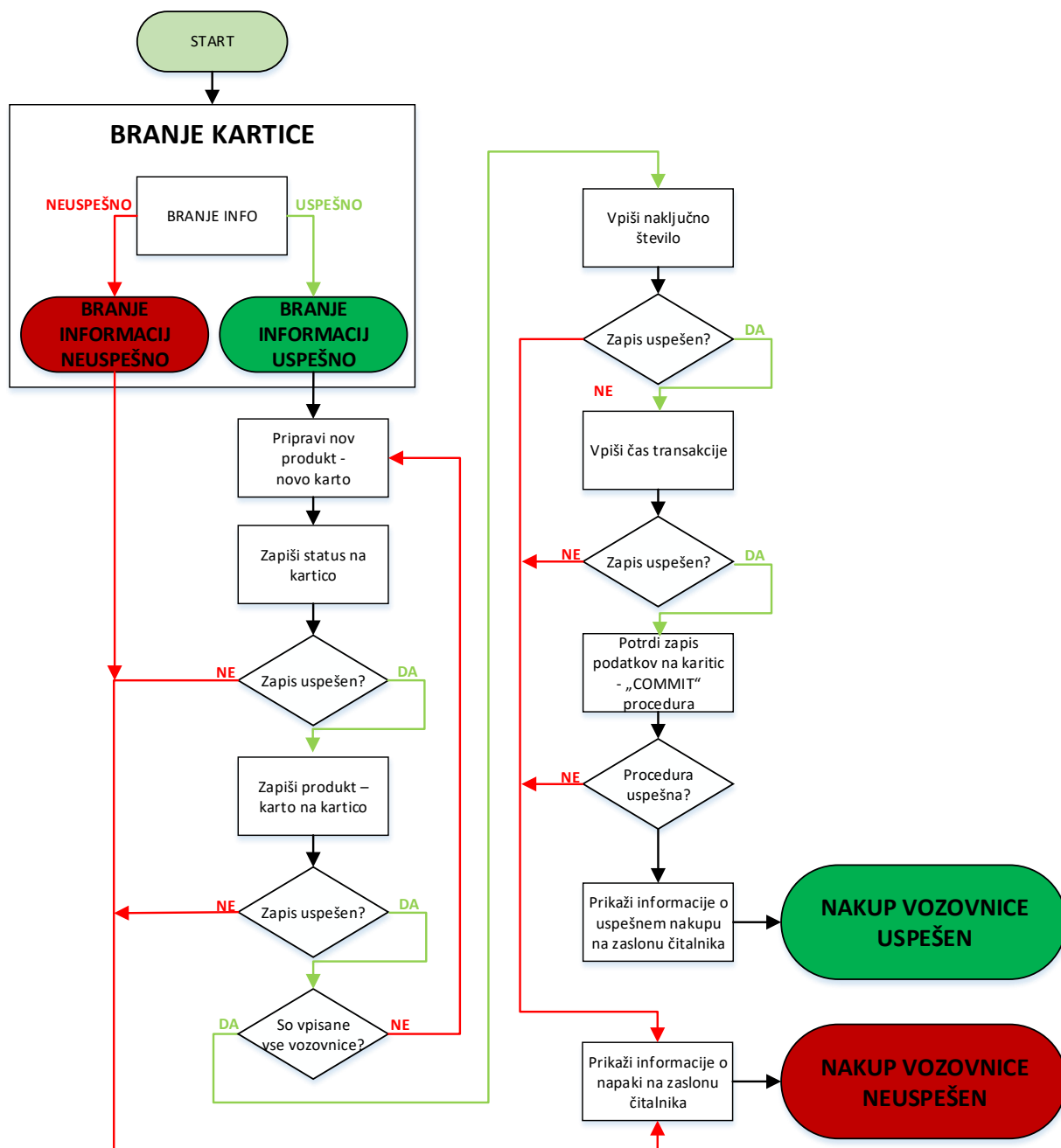
V stanju izvedemo ukaz za potrditev zapisanih podatkov na kartico.

- **Ukaz:** Commit Transaction  
Z ukazom aktiviramo postopek potrjevanja zapisanih podatkov.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno potrjevanje podatkov.

### 3.4.3.6. Stanje potrdi zapis podatkov

Po uspešnem potrjevanju podatkov na kartici uporabnika obvestimo o uspešni transakciji.

#### DS 4: Integracija IS sistemov prevoznika



SLIKA 3.6: NAKUP VOZOVNICE

#### 3.4.4. PROTOKOL KOMUNIKACIJE MED ČITALNIKOM IN KARTICO PRI VALIDACIJI / AKTIVACIJI OBSTOJEČE VOZOVNICE

Ko je na terminalu izbrana in se izvaja ustrezna aplikacija za validacijo oz. aktivacijo vozovnic, lahko kadarkoli k terminalu prislonimo kartico, ki jo želimo validirati oz. aktivirati, da se izvede postopek validacije oz. aktivacije. Po uspešno izvedenem postopku se informacije o validaciji izpišejo na zaslonu.

## DS 4: Integracija IS sistemov prevoznika

Cilj postopka validacije/aktivacije vozovnice je ustrezen protokol komunikacije med čitalnikom in kartico, ki omogoča, da se na kartici validira oz. aktivira obstoječa veljavna vozovnica. Postopek validacije/aktivacije vozovnice je razdeljen v različna zaporedna stanja. V posameznih stanjih se izvajajo določeni postopki branja ali pisanja v določene datoteke na kartici. Pred nadaljnjimi operacijami validacije ali aktivacije vozovnic se izvede postopek branje info (Slika 3.4). S tem s kartice preberemo vse potrebne informacije o kartici in o obstoječih vozovnicah na kartici. Postopek validacije ali aktivacije nadaljujemo v zadnjem stanju postopka branje INFO po sledečem protokolu.

#### 3.4.4.1. Stanje branje zgodovine validacij (datoteka HISTORY-VALID)

- **Akcije:**
  1. Shranimo zgodovino validacij v podatkovno strukturo.
  2. Poiščemo vozovnice, katerih veljavnost se je iztekla in sprožimo postopek brisanja oz. spremembe statusa teh vozovnic.
  3. Sprožimo postopek VALIDACIJA ali AKTIVACIJE.
- **Pogoji:**
  - Če so na kartici neveljavne conske vozovnice sprožimo postopek spremembe STATUS-a (brisanje) vozovnic v neveljavne po naslednjem postopku:
    1. **Stanje zapis statusov vozovnic (datoteka PRODUCTS)**  
Spremenimo status neveljavnih vozovnic v NEVELJAVNA.
    2. **Stanje potrdi zapis podatkov**  
Potrdimo zapisane podatke ne vozovnico.
    3. **Stanje branje vozovnic (datoteka PRODUCTS)**  
Ponovno preberemo vozovnice na kartici ter nadaljujemo postopek VALIDACIJE ali AKTIVACIJE.
  - Če je na kartici neveljavna relacijska vozovnica pri kateri je potrebno pobrisati relacijo sprožimo postopek spremembe STATUS-a (brisanje) relacij v neveljavne po naslednjem postopku:
    1. **Stanje zapis statusov relacij (datoteka PRODUCTS)**  
Spremenimo status neveljavnih relacijskih vozovnic v NEVELJAVNA.
    2. **Stanje potrdi zapis podatkov**  
Potrdimo zapisane podatke ne vozovnico.
    3. **Stanje branje vozovnic (datoteka PRODUCTS)**  
Ponovno preberemo vozovnice na kartici ter nadaljujemo postopek VALIDACIJE ali AKTIVACIJE.
  - Če je na kartici ustrezna veljavna vozovnica gre za postopek VALIDACIJE. Postopek nadaljujemo v stanju:
    - **Stanje zapis validacije v zgodovino validacij (datoteka HISTORY-VALID).**

## DS 4: Integracija IS sistemov prevoznika

---

- Če je na kartici ustrezna veljavna vozovnica vendar še ni aktivirana gre za postopek AKTIVACIJE. Postopek nadaljujemo v stanju
    - **Stanje zapis statusov vozovnic (datoteka PRODUCTS).**
  - Če na kartici ni ustrezne vozovnice, uporabnika obvestimo, da na kartici ni ustrezne vozovnice in prekinemo operacijo.
- **Ukaz v primeru VALIDACIJE:** Write Record  
Z ukazom aktiviramo postopek zapisovanja podatkov validacije v ciklično datoteko vnosov HISTORY-VALID.
  - **Ukaz v primeru AKTIVACIJE:** Write Data  
Z ukazom aktiviramo postopek zapisa statusa vozovnice v datoteki PRODUCTS.
  - **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.

### 3.4.4.2. Stanje zapis statusov vozovnic (datoteka PRODUCTS)

V stanju aktiviramo ukaz za zapis podatkov vozovnice.

- **Ukaz:** Write Data  
Z ukazom aktiviramo zapis preostalih informacij o produktu oz. vozovnici, ki jo želimo aktivirati ali zapisati na kartico.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.

### 3.4.4.3. Stanje zapis vozovnic (datoteka PRODUCTS)

Če želimo aktivirati še kakšno vozovnico nadaljujemo z zapisovanjem statusa naslednje vozovnice v **stanju zapis statusov vozovnic (datoteka PRODUCTS)**. Ko smo aktivirali vse vozovnice nadaljujemo v **stanju zapis validacije v zgodovino validacij (datoteka HISTORY-VALID)**

- **Ukaz:** Write Record  
Z ukazom aktiviramo zapis podatkov validacije v zgodovino validacij v ciklično datoteko vnosov HISTORY-VALID.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.

### 3.4.4.4. Stanje zapis validacije v zgodovino validacij (datoteka HISTORY-VALID)



## DS 4: Integracija IS sistemov prevoznika

---

V stanju zapišemo informacije o validaciji v zgodovino validacij.

- **Ukaz:** Write Data  
Z ukazom aktiviramo zapis naključnega števila v datoteko CHECK.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.

### 3.4.4.5. Stanje zapis naključnega števila (datoteka CHECK)

Po uspešnem zapisu naključnega števila na kartico zapišemo še čas transakcije.

- **Ukaz:** Write Data  
Z ukazom aktiviramo postopek zapisa časa transakcije na kartico v datoteko CHECK.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.

### 3.4.4.6. Stanje zapis časa zadnje transakcije (datoteka CHECK)

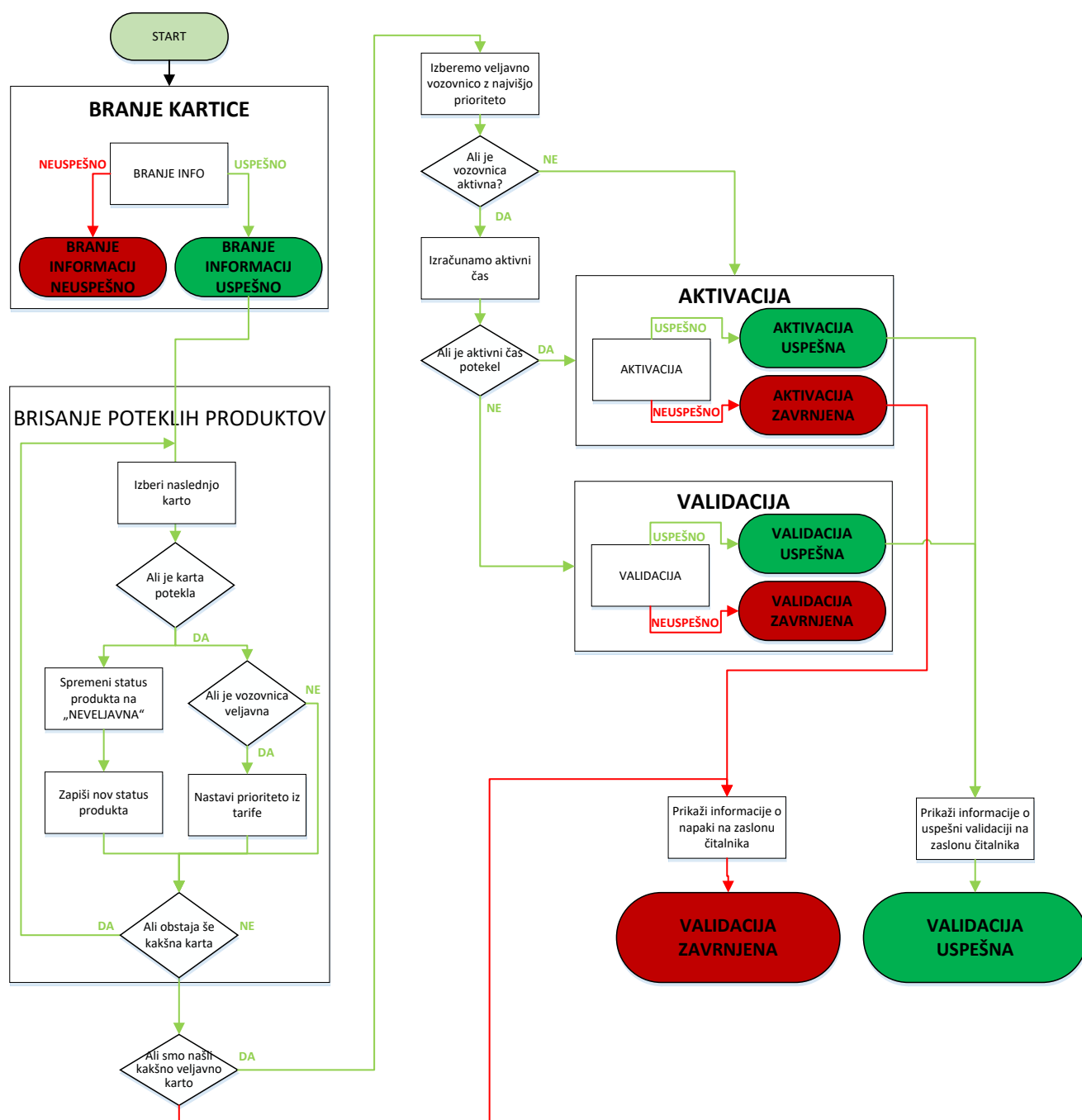
V stanju izvedemo ukaz za potrditev zapisanih podatkov na kartico.

- **Ukaz:** Commit Transaction  
Z ukazom aktiviramo postopek potrjevanja zapisanih podatkov na kartico.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno potrjevanje podatkov.

### 3.4.4.7. Stanje potrdi zapis podatkov

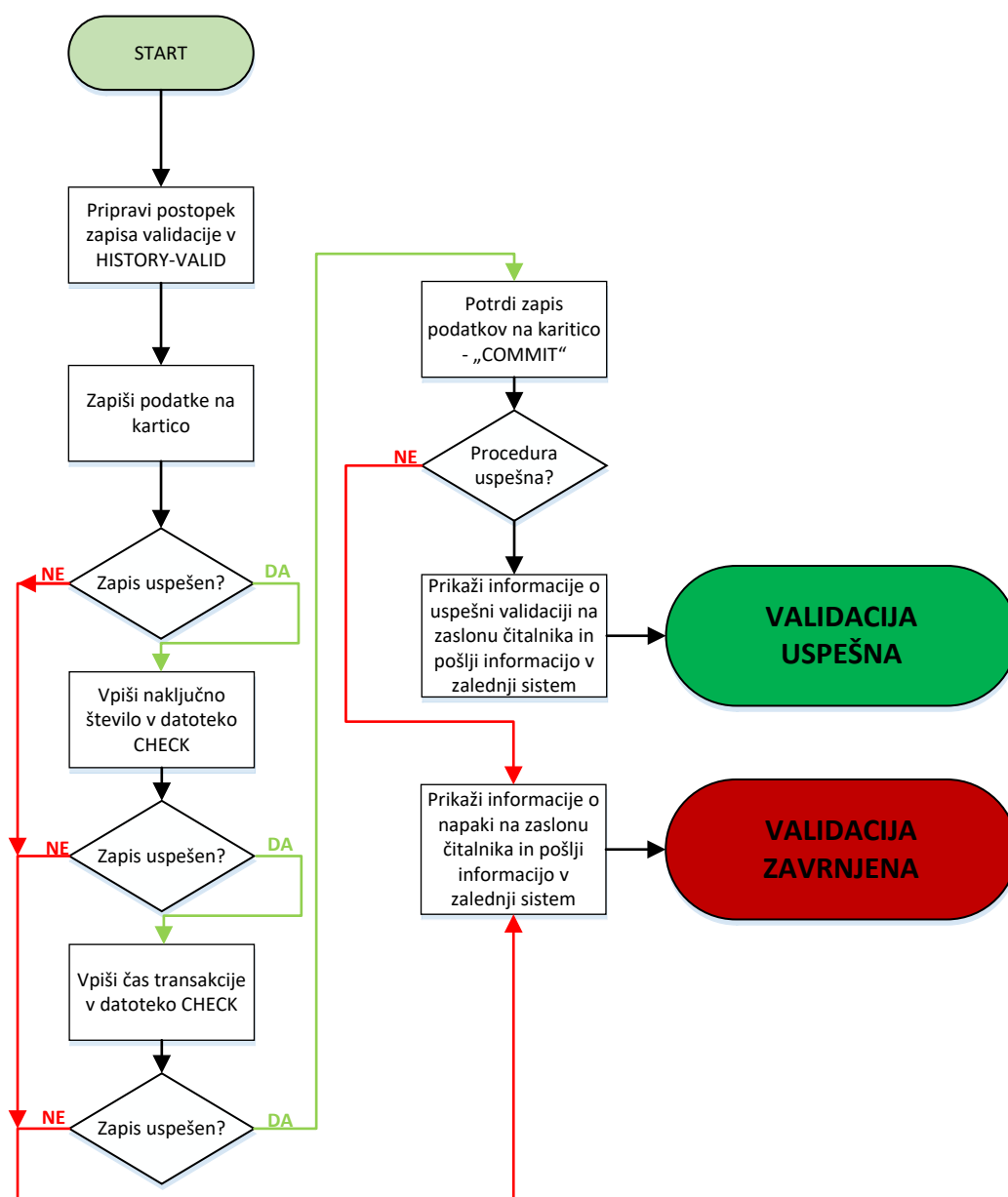
Po uspešnem potrjevanju podatkov na kartici uporabnika obvestimo o uspešni transakciji.

DS 4: Integracija IS sistemov prevoznika



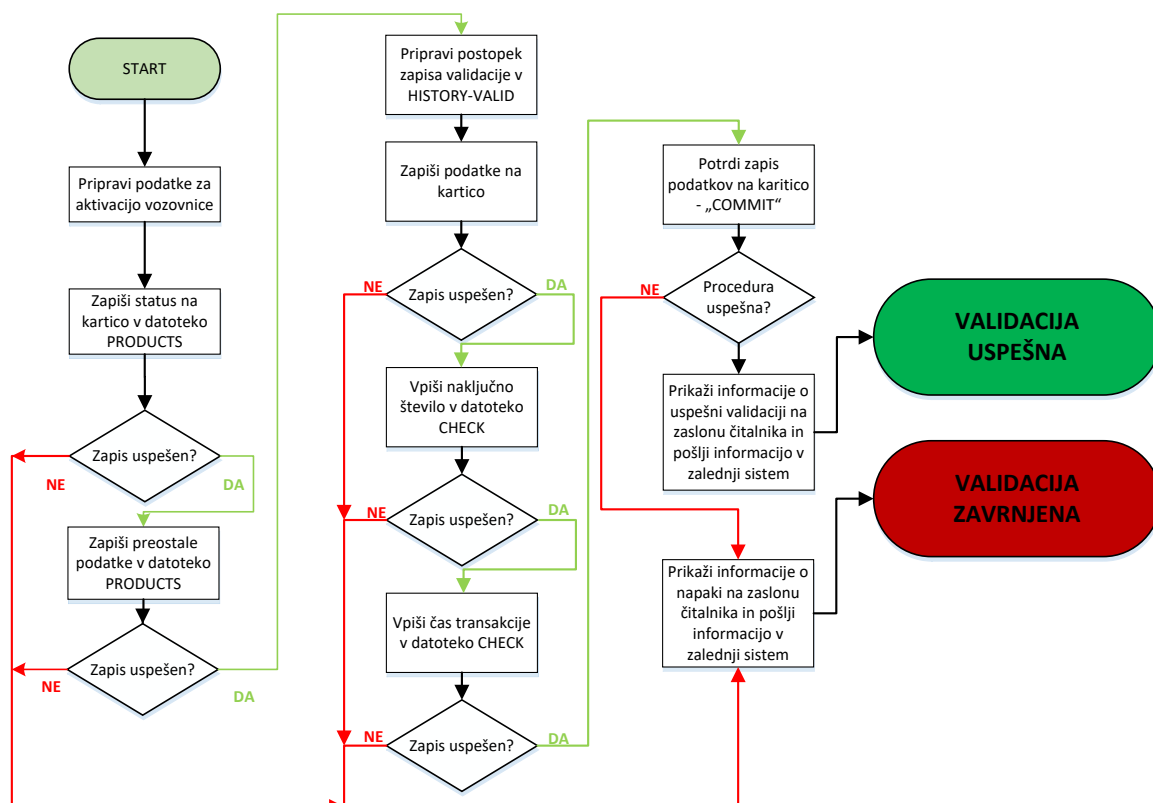
SLIKA 3.7: POTEK POSTOPKA VALIDACIJE IN AKTIVACIJE

DS 4: Integracija IS sistemov prevoznika



SLIKA 3.8: POSTOPEK VALIDACIJE

## DS 4: Integracija IS sistemov prevoznika



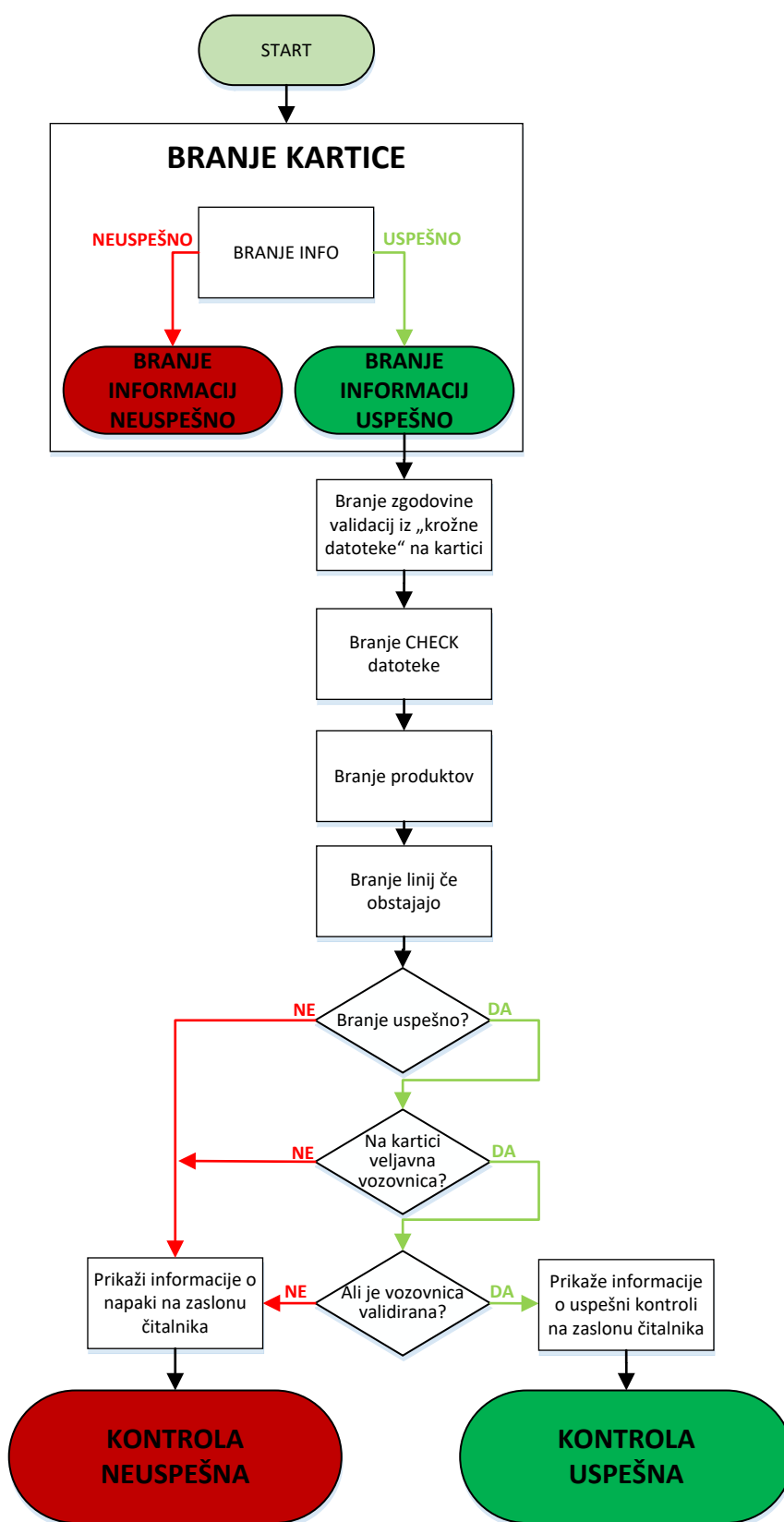
SLIKA 3.9: POSTOPEK AKTIVACIJE VOZOVNICE

### 3.4.5. PROTOKOL KOMUNIKACIJE MED ČITALNIKOM IN KARTICO PRI PREVERJANJU VOZOVNICE

Cilj postopka preverjanja vozovnice je ustrezen protokol komunikacije med čitalnikom in kartico, ki omogoči preverjanje veljavne vozovnice na kartici. Proces preverjanja veljavne vozovnice je namenjen napravam, ki so namenjene kontrolorjem. Za postopek preverjanja je potrebno najprej prebrati vsebino kartice s pomočjo postopka branja info (Slika 3.4). Postopek preverjanja vozovnice je razdeljen v različna zaporedna stanja. V posameznih stanjih se izvajajo določeni postopki branja določenih datotek na kartici.

Informacije o veljavnosti oz. neveljavnosti posamezne vozovnice prikažemo na zaslonu čitalnika. V primeru, da uporabnik nima veljavne vozovnice na kartici, se lahko uporabniku izda opozorilo. Najstrožji ukrep je odvzem kartice ali dodaja UID kartice na črno listo kartic. Če je UID kartice na črni listi, se kartica zaklene ob naslednji aktivaciji na poljubnem terminalu v sistemu.

DS 4: Integracija IS sistemov prevoznika



SLIKA 3.10: POSTOPEK PREVERJANJA VOZOVNICE

### 3.4.6. PROTOKOL KOMUNIKACIJE MED ČITALNIKOM IN KARTICO PRI STORNACIJI VOZOVNICE

Ko je na terminalu izbrana in se izvaja ustrezna aplikacija za storno nakup vozovnice, lahko k terminalu prislonimo kartico, za katero želimo opraviti storno nakup vozovnice. Najprej se izvede branje info (Slika 3.4) oz. ustreznih informacij o kartici na podlagi katerih se lahko na zaslonu prikažejo vsi ustrezni produkti, ki jih za to kartico lahko storniramo. Po branju informacij terminal preveri naključno število zapisano na kartici in ga primerja z naključnim številom, ki se je ustvarilo pri zadnji transakciji na terminalu. S tem lahko določimo, če se vozovnice na kartici lahko stornirajo. Storniramo namreč lahko vozovnice, ki so bile predhodno kupljene na istem terminalu. Po izbiri ustreznega produkta, kartico ponovno prislonimo k terminalu. Po uspešno izvedenem nakupu se na zaslonu izpišejo informacije o storno postopku. Imetniku kartice pa se povrnejo sredstva v obliki gotovine. Predpogoj za storno nakupa vozovnice je uspešen predhoden nakup vozovnice na istem čitalniku.

Cilj postopka storno nakupa vozovnice je ustrezen protokol komunikacije med čitalnikom in kartico, ki omogoči, da se zadnji nakup na določenem čitalniku stornira. Postopek storno je razdeljen v različna zaporedna stanja. V posameznih stanjih se izvajajo določeni postopki branja in pisanja določenih datotek na kartici. Pred nadaljnjimi operacijami storno nakupa vozovnice se izvede postopek branje info. S tem s kartice preberemo vse potrebne informacije o kartici in o obstoječih vozovnicah na kartici. Postopek storno nakupa vozovnice nadaljujemo po izbiri produkta, ki ga želimo stornirati, po sledečem protokolu.

#### 3.4.6.1. Stanje branje zgodovine validacij (datoteka HISTORY-VALID)

- **Akcije:**
  1. Shranimo zgodovino validacij v podatkovno strukturo.
  2. Aktiviramo pisanje statusa NEVELJAVNA izbrane vozovnice.
- **Ukaz: Write Data**

Z ukazom aktiviramo pisanje statusa vozovnice v datoteko PRODUCTS.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešno pisanje podatkov.

#### 3.4.6.2. Stanje zapis statusa (datoteka PRODUCTS)

V stanju aktiviramo ukaz za zapis naključnega števila.

- **Ukaz: Write Data**

## DS 4: Integracija IS sistemov prevoznika

---

Z ukazom aktiviramo zapis naključnega števila v datoteko CHECK.

- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.

### 3.4.6.3. Stanje zapis naključnega števila (datoteka CHECK)

Po uspešnem zapisu naključnega števila na kartico zapišemo še čas transakcije.

- **Ukaz:** Write Data  
Z ukazom aktiviramo postopek zapisa časa transakcije v datoteko CHECK.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.

### 3.4.6.4. Stanje zapis časa zadnje transakcije (datoteka CHECK)

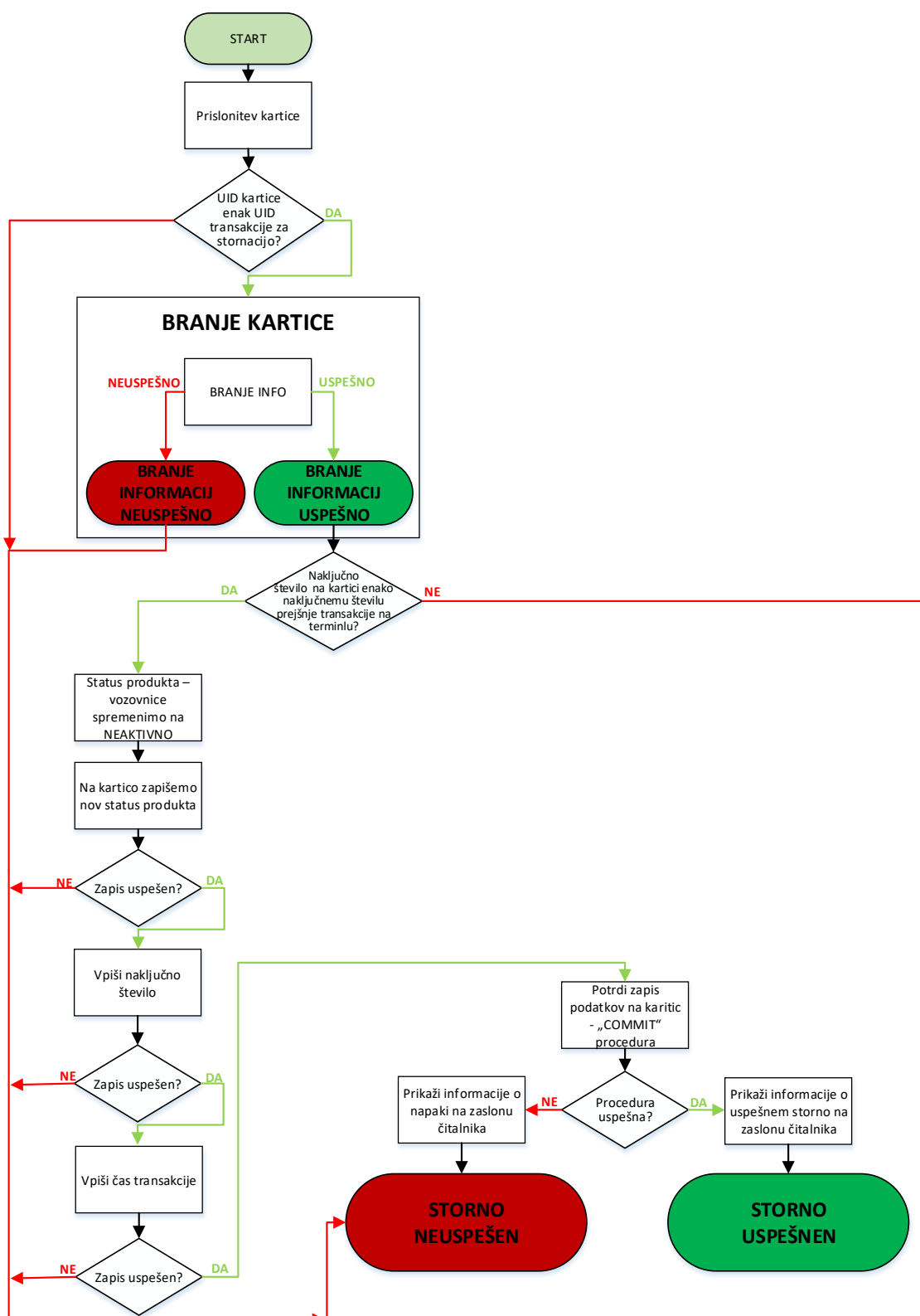
V stanju izvedemo ukaz za potrditev zapisanih podatkov na kartico.

- **Ukaz:** Commit Transaction  
Z ukazom aktiviramo postopek potrjevanja zapisanih podatkov na kartico.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno potrjevanje podatkov.

### 3.4.6.5. Stanje potrdi zapis podatkov

Po uspešnem potrjevanju podatkov na kartici uporabnika obvestimo o uspešni storno transakciji.

DS 4: Integracija IS sistemov prevoznika



SLIKA 3.11: STORNO NAKUPA VOZOVNICE



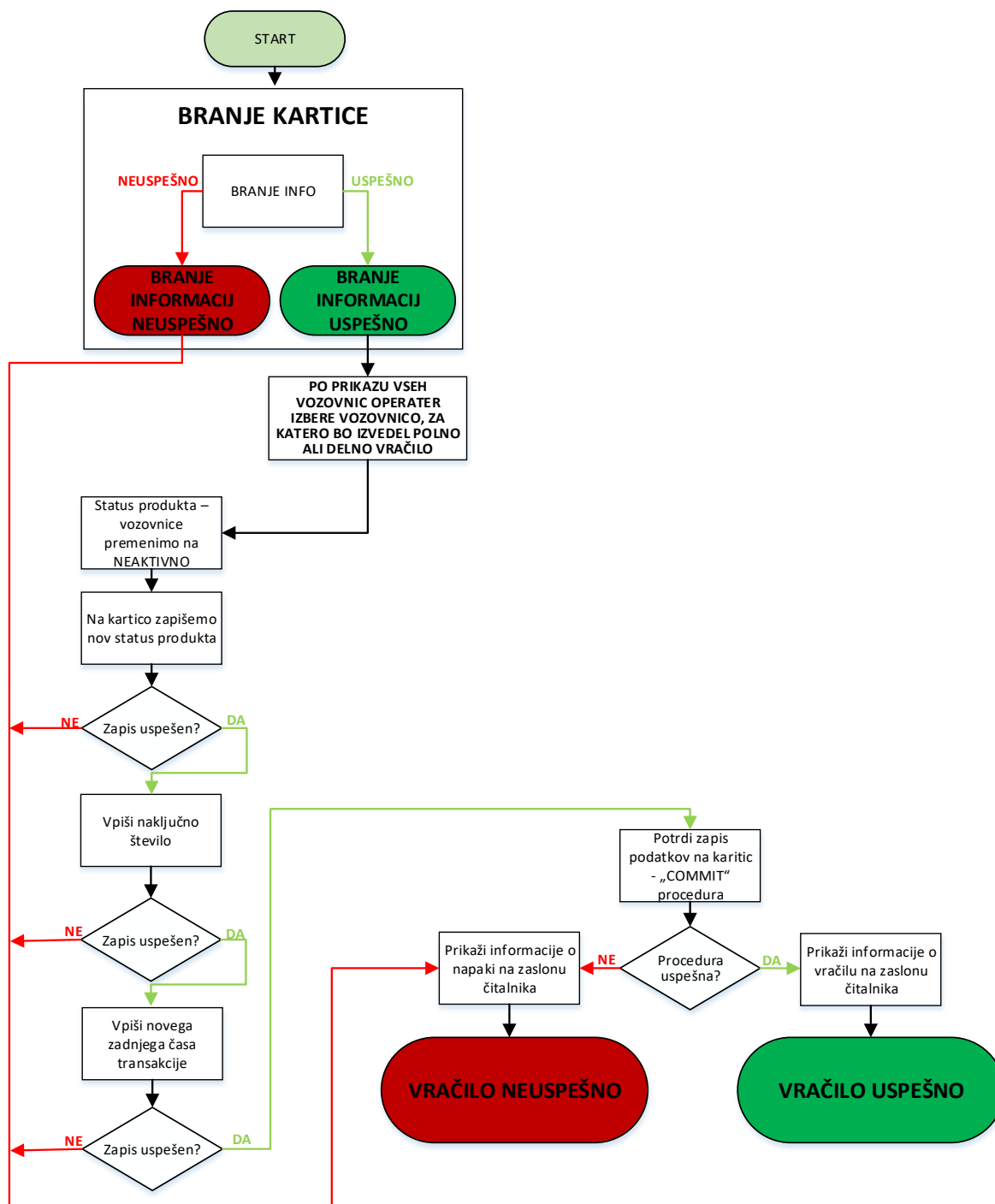
### *3.4.7. PROTOKOL KOMUNIKACIJE MED ČITALNIKOM IN KARTICO OB POPOLNEM VRAČILU - IZPLAČILU VOZOVNICE*

Cilj postopka popolnega vračila oz. izplačila vozovnice je ustrezen protokol komunikacije med čitalnikom in kartico, ki omogoči, da se na ustrezni kartici izbriše ustrezna vozovnica. Sredstva nakupa vozovnice pa se povrnejo imetniku kartice. Postopek nakupa je razdeljen v različna zaporedna stanja. V posameznih stanjih se izvajajo določeni postopki branja ali pisanja v določene datoteke na kartici.

Operacijo popolnega vračila in izplačila vozovnice lahko izvedemo s pomočjo Protokola komunikacije med čitalnikom in kartico pri branju informacij (Slika 3.4) o obstoječih vozovnicah na kartici po naslednjem postopku:

- Na zaslonu čitalnika prikažemo vse vozovnice na kartici.
- Izberemo ustrezno vozovnico za katero želimo opraviti postopek vračila – izplačila vozovnice.
- Ponovno prislavimo kartico k čitalniku, kjer se izvede postopek brisanja vozovnice na kartici.
- Sredstva se uporabniku vrnejo gotovini.

#### DS 4: Integracija IS sistemov prevoznika



SLIKA 3.12: POSTOPEK PRI POLNEM VRAČILU - IZPLAČILU VOZOVNICE

#### 3.4.7.1. Stanje pripravljenosti

V stanju pripravljenosti se opravijo različni inicializacijski postopki različnih podatkovnih struktur, ki se uporabljajo pri procesu. V ustrezno podatkovno strukturo, ki služi kot zbiralnik podatkov, ki se uporabljajo med postopkom se zapišejo naslednji podatki:

## DS 4: Integracija IS sistemov prevoznika

---

- **Čas transakcije**
- **UID kartice**  
UID se prebere iz kartice s postopkom izbire oz. antikolizije. Tako je UID pripravljen takoj, ko je kartica pripravljena za nadaljnja branja in pisanja podatkov.
- **Ukaz: Select Application**  
Ko kartica odgovori se komunikacija nadaljuje v naslednjem stanju.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešna izbira aplikacije.

### 3.4.7.2. Stanje izberi aplikacijo

V stanju izberi aplikacijo čitalnik pošlje ukaz za avtentikacijo:

- **Ukaz: Authenticate**  
Aktivira se postopek avtentikacije med čitalnikom in kartico.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešna avtentikacija med čitalnikom in kartico.

### 3.4.7.3. Stanje avtentikacija

V stanju avtentikacija se po uspešno zaključeni avtentikaciji lahko prične postopek branja in pisanja v zelenih datotekah. Nadaljujemo z branjem verzije kartice z ukazom:

- **Ukaz: Read Data**  
Z ukazom aktiviramo postopek branja časa zadnje transakcije in naključnega števila.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje podatkov.

### 3.4.7.4. Stanje branje časa zadnje transakcije in naključnega števila

V tem stanju preberemo čas zadnje transakcije in naključno število na kartici.

- **Akcije:**  
Shranimo čas zadnje uspešne transakcije in naključno število.
- **Pogoji:**  
V primeru gotovinskega vračila vozovnice nadaljujemo v stanju **Stanje zapis statusa**.
- **Ukaz v primeru gotovinskega vračila: Write Data**  
Z ukazom aktiviramo spremembo STATUSA vozovnice iz aktivne v neaktivno.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno pisanje podatkov.

### 3.4.7.5. Stanje zapis statusa

V stanju aktiviramo ukaz za zapis naključnega števila.

## DS 4: Integracija IS sistemov prevoznika

---

- **Ukaz: Write Data**  
Z ukazom aktiviramo zapis naključnega števila.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.

### 3.4.7.6. Stanje zapis naključnega števila

Po uspešnem zapisu naključnega števila na kartico zapišemo še čas transakcije.

- **Ukaz: Write Data**  
Z ukazom aktiviramo postopek zapisa časa transakcije na kartico.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.

### 3.4.7.7. Stanje zapis časa zadnje transakcije

V stanju izvedemo ukaz za potrditev zapisanih podatkov na kartico.

- **Ukaz: Commit Transaction**  
Z ukazom aktiviramo postopek potrjevanja zapisanih podatkov na kartico.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno potrjevanje podatkov.

### 3.4.7.8. Stanje potrdi zapis podatkov

Po uspešnem potrjevanju podatkov na kartici uporabnika obvestimo o uspešni storno transakciji.

## 3.4.8. PROTOKOL KOMUNIKACIJE MED ČITALNIKOM IN KARTICO OB DELNEM VRAČILU – DELNEM IZPLAČILU VOZOVNICE

Cilj postopka delnega vračila oz. izplačila vozovnice je ustrezen protokol komunikacije med čitalnikom in kartico, ki omogoči, da se na ustrezni kartici izbriše ustrezna vozovnica. Sredstva nakupa vozovnice pa se delno povrnejo imetniku kartice. Postopek nakupa je razdeljen v različna zaporedna stanja. V posameznih stanjih se izvajajo določeni postopki branja ali pisanja v določene datoteke na kartici.

Operacijo delnega vračila in izplačila vozovnice lahko izvedemo s pomočjo Protokola komunikacije med čitalnikom in kartico pri branju informacij (Slika 3.4) o obstoječih vozovnicah na kartici po naslednjem postopku:

- Na zaslonu čitalnika prikažemo vse vozovnice na kartici.
- Izberemo ustrezno vozovnico za katero želimo opraviti postopek delnega vračila – izplačila vozovnice.

## DS 4: Integracija IS sistemov prevoznika

---

- Ponovno prislono kartico k čitalniku, kjer se izvede postopek brisanja vozovnice na kartici.
- Sredstva se uporabniku vrnejo bodisi v gotovini.

Postopek brisanja kartica ter povračila sredstev je identičen kot v primeru celotnega vračila – izplačila vozovnice.

### *3.4.9. PROTOKOL KOMUNIKACIJE MED ČITALNIKOM IN KARTICO OB BLOKADI APLIKACIJE IJPP*

Blokada aplikacije se izvede samodejno pri postopku branja info (Slika 3.4) kartice v primeru, da se UID kartice nahaja na črni listi.

Cilj postopka blokade aplikacije IJPP je ustrezen protokol komunikacije med čitalnikom in kartico, ki omogoči, da se STATUS aplikacije IJPP nastavi na vrednost BLOKIRANA. Postopek blokade aplikacije IJPP je razdeljen v različna zaporedna stanja. V posameznih stanjih se izvajajo določeni postopki branja ali pisanja v določene datoteke na kartici

#### **3.4.9.1. Stanje pripravljenosti**

##### ○ **Akcije**

1. V stanju pripravljenosti se opravijo različni inicializacijski postopki podatkovne strukture, ki se uporablja pri procesu.
2. V strukturo se zapiše **Čas transakcije**
3. V strukturo se zapiše **UID kartice**  
UID se prebere iz kartice s postopkom izbire oz. antikalizije. Tako je UID pripravljen takoj, ko je kartica pripravljena za nadaljnja branja in pisanja podatkov.

Sledi operacija izbire ustrezne aplikacije (AID) s katero želi čitalnik komunicirati čitalnik pošlje ukaz:

- **Ukaz:** Select Application
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešna izbira aplikacije.

#### **3.4.9.2. Stanje izbiri aplikacijo**

##### ○ **Akcije**

1. V stanju izberi aplikacijo čitalnik pošlje ukaz za avtentikacijo.

## DS 4: Integracija IS sistemov prevoznika

---

### 2. Postopek avtentikacije.

- **Ukaz:** Authenticate  
Aktivira se postopek avtentikacije med čitalnikom in kartico.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešna avtentikacija med čitalnikom in kartico.

#### 3.4.9.3. Stanje avtentikacija

V stanju avtentikacija se po uspešno zaključeni avtentikaciji lahko prične postopek branja in pisanja v zelenih datotekah.

- **Akcije**  
Branje verzije kartice oz. datoteke INFO.
- **Ukaz:** Read Data  
Z ukazom aktiviramo postopek branja datoteke INFO na kartici, ki je namenjena informacijam o verziji kartice.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje datoteke INFO.

#### 3.4.9.4. Stanje branje verzije (datoteka INFO)

- **Akcije:**
  1. V stanju branje verzije shranimo podatke o verziji kartice v ustrezno podatkovno strukturo.
  2. Branje osebnih podatkov oz. datoteke CARD INFO.
- **Pogoji:**

Primerjamo prebrane podatke o zapisani verziji na kartici z želeno verzijo kartice za nadaljnjo komunikacijo.

  1. Če verzija trenutne kartice ni ustrezna, prekinemo operacijo nakupa vozovnice in obvestimo uporabnika o nastali napaki.
  2. Če je verzija ustrezna nadaljujemo nakup vozovnice.
- **Ukaz:** Read Data  
Z ukazom aktiviramo postopek branja datoteke CARD INFO, ki vsebuje splošne informacije o kartici in osebne podatke o imetniku kartice.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje datoteke CARD INFO.

#### 3.4.9.5. Stanje branje info (datoteka CARD INFO)

- **Akcije:**

V tem stanju shranimo splošne informacije o kartici in imetniku kartice v ustrezno podatkovno strukturo.

- **Pogoji:**

1. Če je STATUS kartice blokirana ali ni aktivirana vrnemo napako in prekinemo operacijo nakupa vozovnice.
2. Preverimo, če se kartica nahaja na črni listi sprožimo postopek blokade kartice.
3. Preverimo ustreznost parametra POSEBNI STATUS. Če ni ustrezen prekinemo operacijo nakupa vozovnice in uporabnika obvestimo o napaki.
4. Preverimo veljavnost kartice. Če ni veljavna prekinemo operacijo nakupa in uporabnika obvestimo o poteku veljavnosti kartice.

Če je kartica na črni listi se nadaljuje postopek blokade kartice.

- **Ukaz: Write Data**

Z ukazom aktiviramo postopek pisanja STATUSA kartice, kjer zapišemo vrednost BLOKIRANA.

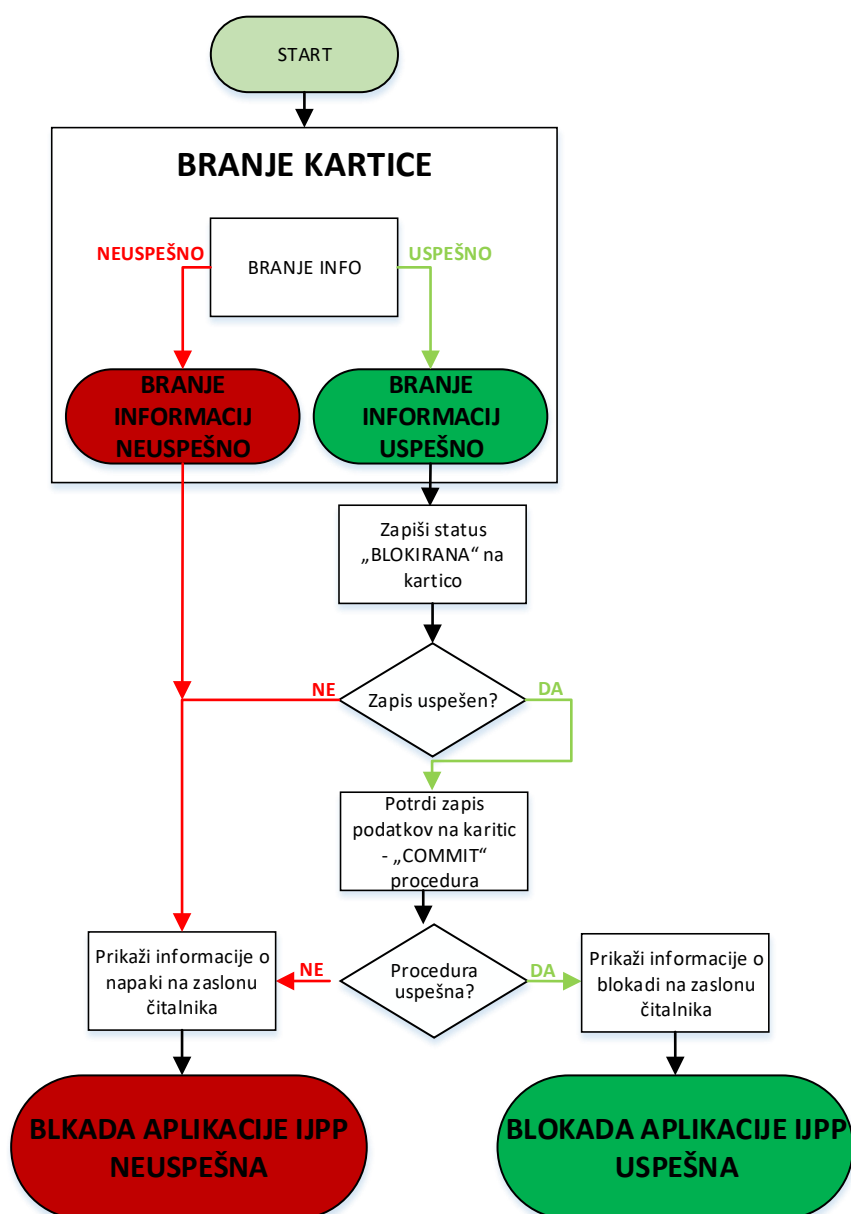
- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešno pisanje podatkov.

#### 3.4.9.6. Stanje potrdi zapis podatkov

Po uspešnem potrjevanju podatkov na kartici uporabnika obvestimo o uspešni blokadi kartice.

#### DS 4: Integracija IS sistemov prevoznika



SLIKA 3.13: BLOKADA APLIKACIJE IJPP

#### 3.4.10. PROTOKOL KOMUNIKACIJE MED ČITALNIKOM IN KARTICO OB BLOKADI VOZOVNICE

Blokada vozovnice se izvede samodejno pri postopku branja info (Slika 3.4) kartice v primeru, da se UID kartice nahaja na izvršni listi. Najprej se izvede branje INFO oz. ustreznih informacij o kartici na podlagi katerih se lahko locira ustrezna vozovnica, ki jo potrebno na podlagi informacij izvršne liste blokirati. Nato se izvrši zapis statusa ustrezne vozovnice na enak način kot pri postopku storno nakupa vozovnice.

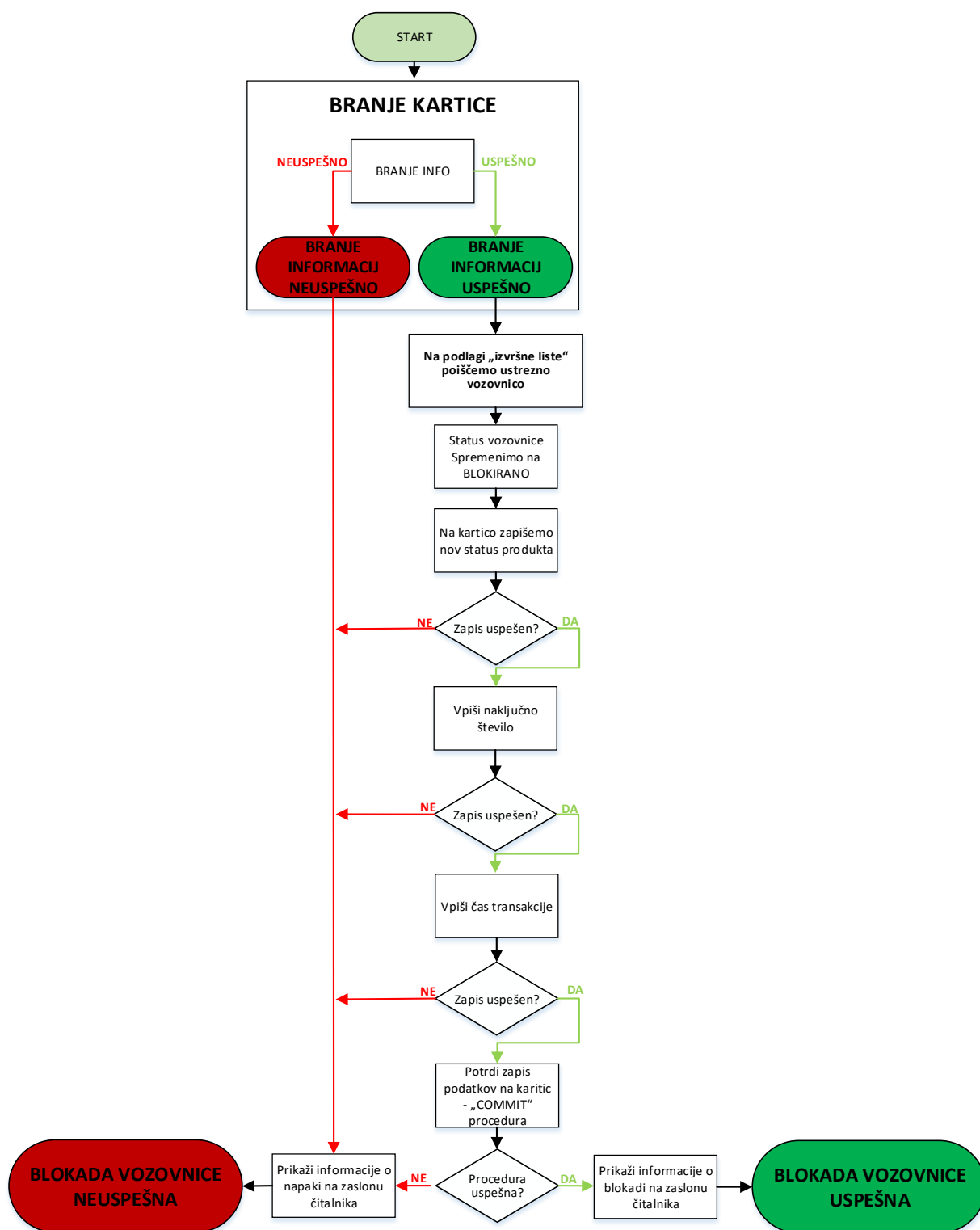


#### DS 4: Integracija IS sistemov prevoznika

---

Cilj postopka blokade vozovnice je ustrezen protokol komunikacije med čitalnikom in kartico, ki omogoči, da se na ustrezni kartici blokira ustrezna vozovnica. Postopek blokade vozovnice je razdeljen v različna zaporedna stanja. V posameznih stanjih se izvajajo določeni postopki branja ali pisanja v določene datoteke na kartici. Operacijo blokade vozovnice izvedemo s pomočjo postopka branje INFO in postopka storno nakupa vozovnice.

#### DS 4: Integracija IS sistemov prevoznika



SLIKA 3.14: POSTOPEK BLOKAD VOZOVNICE

#### 3.4.10.1. Stanje pripravljenosti

V stanju pripravljenosti se opravijo različni inicializacijski postopki različnih podatkovnih struktur, ki se uporabljajo pri procesu. V ustrezno podatkovno strukturo, ki služi kot zbirnik podatkov, ki se uporabljajo med postopkom se zapišejo naslednji podatki:

- **Čas transakcije**
- **UID kartice**  
UID se prebere iz kartice s postopkom izbire oz. antikalizije. Tako je UID pripravljen takoj, ko je kartica pripravljena za nadaljnja branja in pisanja podatkov.
- **Ukaz: Select Application**  
Ko kartica odgovori se komunikacija nadaljuje v naslednjem stanju.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešna izbira aplikacije.

#### 3.4.10.2. Stanje izberi aplikacijo

V stanju izberi aplikacijo čitalnik pošlje ukaz za avtentikacijo:

- **Ukaz: Authenticate**  
Aktivira se postopek avtentikacije med čitalnikom in kartico.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešna avtentikacija med čitalnikom in kartico.

#### 3.4.10.3. Stanje avtentikacija

V stanju avtentikacija se po uspešno zaključeni avtentikaciji lahko prične postopek branja in pisanja v zelenih datotekah. Nadaljujemo z branjem verzije kartice z ukazom:

- **Ukaz: Read Data**  
Z ukazom aktiviramo postopek branja časa zadnje transakcije in naključnega števila.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje podatkov.

#### 3.4.10.4. Stanje branje časa zadnje transakcije in naključnega števila

V tem stanju preberemo čas zadnje transakcije in naključno število na kartici.

- **Akcije:**  
Shranimo čas zadnje uspešne transakcije in naključno število.
- **Ukaz: Write Data**  
Z ukazom aktiviramo pisanje statusa vozovnice.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje podatkov.

#### 3.4.10.5. Stanje zapis statusa

V stanju zapišemo STATUS posamezne vozovnice z vrednostjo BLOKIRANA. V nadaljevanju aktiviramo ukaz za zapis naključnega števila.

- **Ukaz:** Write Data  
Z ukazom aktiviramo zapis naključnega števila.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.

#### 3.4.10.6. Stanje zapis naključnega števila

Po uspešnem zapisu naključnega števila na kartico zapišemo še čas transakcije.

- **Ukaz:** Write Data  
Z ukazom aktiviramo postopek zapisa časa transakcije na kartico.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.

#### 3.4.10.7. Stanje zapis časa zadnje transakcije

V stanju izvedemo ukaz za potrditev zapisanih podatkov na kartico.

- **Ukaz:** Commit Transaction  
Z ukazom aktiviramo postopek potrjevanja zapisanih podatkov na kartico.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno potrjevanje podatkov.

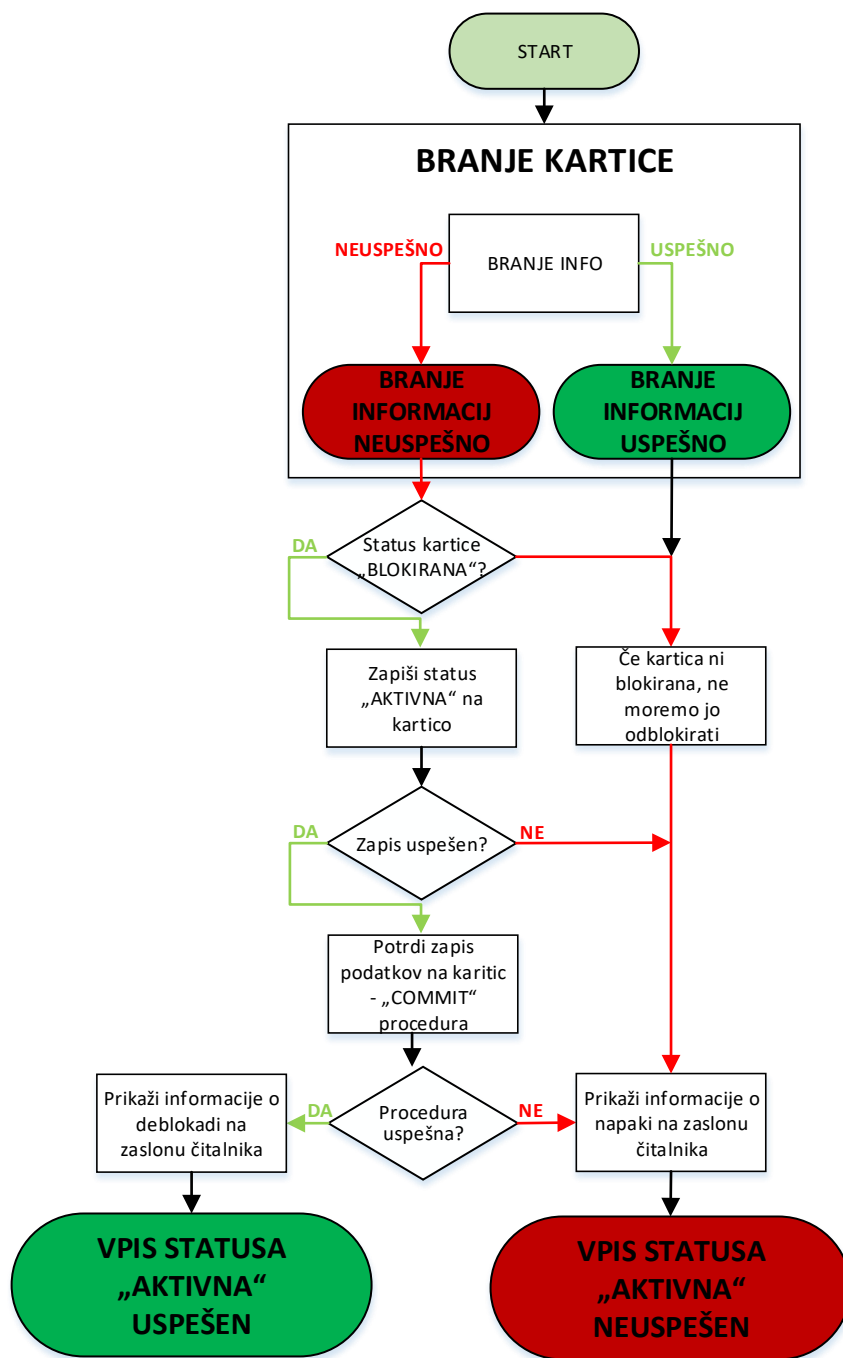
#### 3.4.10.8. Stanje potrdi zapis podatkov

Po uspešnem potrjevanju podatkov na kartici uporabnika obvestimo o uspešni storno transakciji.

### 3.4.11. PROTOKOL KOMUNIKACIJE MED ČITALNIKOM IN KARTICO OB DEBLOKADI APLIKACIJE IJPP

Cilj postopka deblokade aplikacije IJPP je ustrezen protokol komunikacije med čitalnikom in kartico, ki omogoči, da se STATUS aplikacije IJPP nastavi na vrednost AKTIVNA. Postopek deblokade je razdeljen v različna zaporedna stanja. V posameznih stanjih se izvajajo določeni postopki branja ali pisanja v določene datoteke na kartici.

#### DS 4: Integracija IS sistemov prevoznika



SLIKA 3.15: POSTOPEK DEBLOKADE APLIKACIJE IJPP

##### 3.4.11.1. Stanje pripravljenosti

V stanju pripravljenosti se opravijo različni inicializacijski postopki različnih podatkovnih struktur, ki se uporabljajo pri procesu. V ustrezno podatkovno strukturo, ki služi kot zbiralnik podatkov, ki se uporabljajo med postopkom se zapišejo naslednji podatki:

- Čas transakcije
- UID kartice

## DS 4: Integracija IS sistemov prevoznika

---

UID se prebere iz kartice s postopkom izbire oz. antikolizije. Tako je UID pripravljen takoj, ko je kartica pripravljena za nadaljnja branja in pisanja podatkov.

Sledi operacija izbire ustrezne aplikacije (AID) s katero želi čitalnik komunicirati čitalnik pošlje ukaz:

- **Ukaz:** Select Application  
Ko kartica odgovori se komunikacija nadaljuje v naslednjem stanju.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešna izbira aplikacije.

### 3.4.11.2. Stanje izberi aplikacijo

V stanju izberi aplikacijo čitalnik pošlje ukaz za avtentikacijo:

- **Ukaz:** Authenticate  
Aktivira se postopek avtentikacije med čitalnikom in kartico.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešna avtentikacija med čitalnikom in kartico.

### 3.4.11.3. Stanje avtentikacija

V stanju avtentikacija se po uspešno zaključeni avtentikaciji lahko prične postopek branja in pisanja v zelenih datotekah. Nadaljujemo z branjem verzije kartice z ukazom:

- **Ukaz:** Read Data  
Z ukazom aktiviramo postopek branja datoteke na kartici, ki je namenjena informacijam o verziji kartice. Za datoteko, ki vsebuje podatke o verziji kartice ni potrebno, da ima mehanizem varnostne kopije.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje podatkov.

### 3.4.11.4. Stanje branje verzije kartice

- **Akcije:**  
V stanju branje verzije shranimo podatke o verziji kartice v ustrezno podatkovno strukturo.
- **Pogoji:**  
Primerjamo prebrane podatke o zapisani verziji na kartici z želeno verzijo kartice za nadaljnjo komunikacijo. Če verzija trenutne kartice ni ustrezna, prekinemo operacijo nakupa vozovnice in obvestimo uporabnika o nastali napaki. Če je verzija ustrezna nadaljujemo nakup vozovnice.
- **Ukaz:** Write Data  
Z ukazom aktiviramo postopek zapisa dela datoteke, ki vsebuje STATUS kartice.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno pisanje podatkov.

#### 3.4.11.5. Stanje zapis info

Po uspešnem zapisu STATUSA AKTIVNA na kartico potrdimo zapisane podatke.

- **Ukaz:** Commit Transaction  
Z ukazom aktiviramo postopek potrditve zapisa.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno potrjevanje podatkov.

#### 3.4.11.6. Stanje potrdi zapis podatkov

Po uspešnem potrjevanju podatkov na kartici uporabnika obvestimo o uspešni deblokadi kartice.

### 3.4.12. PROTOKOL KOMUNIKACIJE MED ČITALNIKOM IN KARTICO OB DEBLOKADI VOZOVNICE

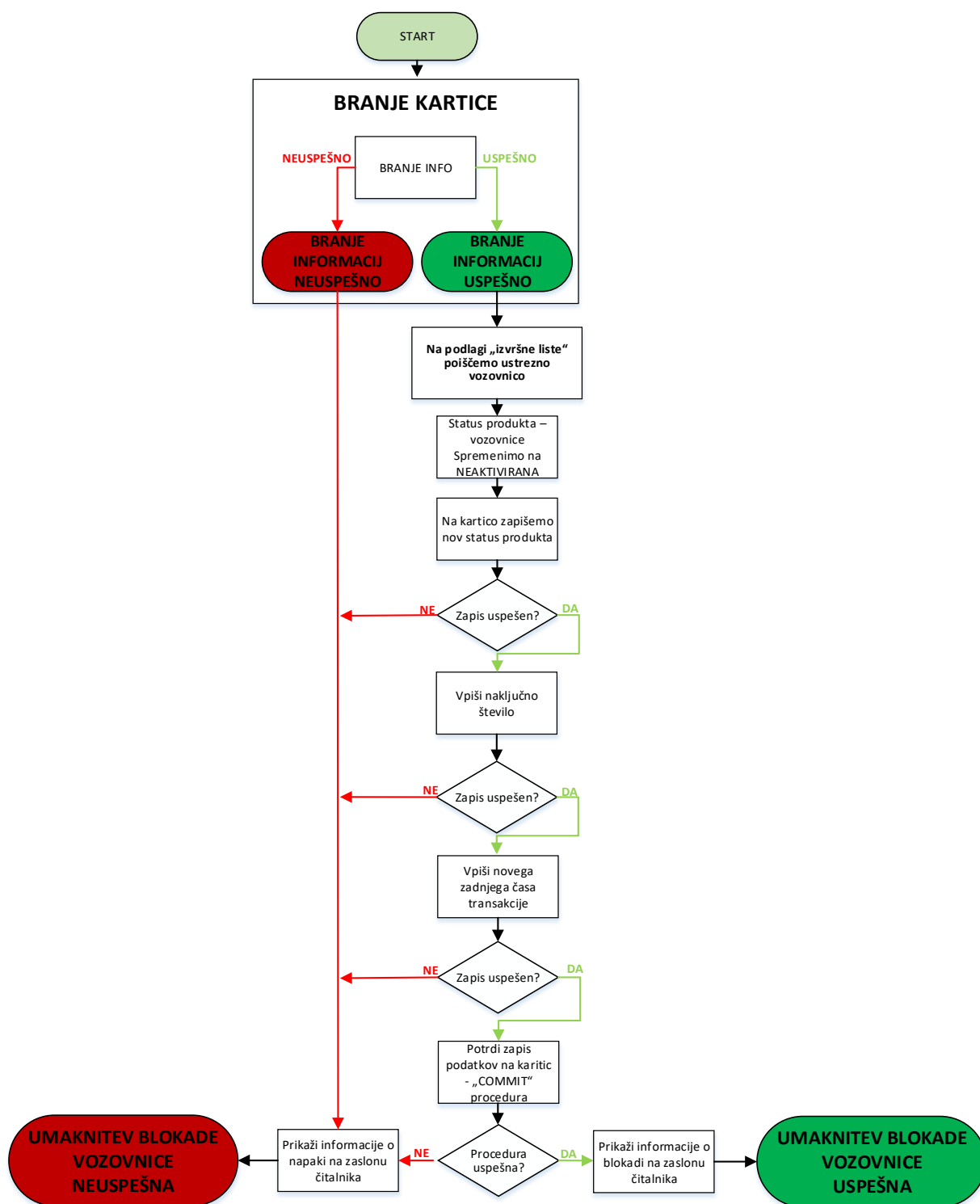
Cilj postopka deblokade vozovnice je ustrezen protokol komunikacije med čitalnikom in kartico, ki omogoči, da se na ustrezni kartici deblokira ustrezna vozovnica. Postopek nakupa je razdeljen v različna zaporedna stanja. V posameznih stanjih se izvajajo določeni postopki branja ali pisanja v določene datoteke na kartici.

Operacijo deblokade vozovnice lahko izvedemo s pomočjo Protokola komunikacije med čitalnikom in kartico pri branju informacij (Slika 3.4) o obstoječih vozovnicah na kartici po naslednjem postopku:

- Na zaslonu čitalnika prikažemo vse vozovnice na kartici.
- Izberemo ustrezno vozovnico, ki jo želimo deblokirati.
- Ponovno prislonimo kartico k čitalniku, kjer se izvede postopek deblokade vozovnice na kartici.

Postopek deblokade vozovnice je identičen postopku blokade vozovnice, kjer spremenimo obstoječ status vozovnice iz vrednosti BLOKIRANA na vrednost NEAKTIVIRANA.

#### DS 4: Integracija IS sistemov prevoznika



SLIKA 3.16: POSTOPEK DEBLOKADE VOZOVNICE

#### 3.4.13. PROTOKOL KOMUNIKACIJE MED ČITALNIKOM IN KARTICO OB PREVERJANJU USPEŠNOSTI ZAPISA VOZOVNICE



## DS 4: Integracija IS sistemov prevoznika

Poseben mehanizem preverjanja uspešnosti zapisa vozovnice se aktivira v primeru, da je bila kartica prehitro odstranjena od terminala. V redkih primerih se zgodi, da se podatki ustrezno zapišejo na kartico medtem ko, se operacija aktivacije ali nakupa vozovnice prekine zaradi prehitre odstranitve kartice.

Cilj postopka preverjanja uspešnosti zapisa vozovnice je ustrezen protokol komunikacije med čitalnikom in kartico, ki omogoči, da se na ustrezni kartici preveri uspešnost zapisa vozovnice. Postopek preverjanja uspešnosti zapisa vozovnice je razdeljen v različna zaporedna stanja. V posameznih stanjih se izvajajo določeni postopki branja ali pisanja v določene datoteke na kartici. Operacija preverjanja uspešnosti zapisa se izvede samodejno po postopku aktivacije ali nakupa vozovnice v primeru, da se kartica predčasno odmakne in se s tem postopek prekine.

Če NFC na terminalu zazna, da je bila kartica odstranjena v zadnjem stanju (**Stanje potrdi zapis podatkov**) postopka aktivacije ali nakupa vozovnice, se na terminalu shranijo UID kartice in naključno število zadnje transakcije, ki se je zapisalo v stanju (**Stanje zapis naključnega števila (datoteka CHECK)**). Mehanizem preverjanja uspešnosti zapisa vozovnice se nadaljuje, ko uporabnik ponovno prisloni kartico k terminalu. Postopek preverjanja uspešnosti zapisa se nadaljuje po naslednjih stanjih:

### 3.4.13.1. Stanje pripravljenosti

- **Akcije**

1. V stanju pripravljenosti se opravijo različni inicializacijski postopki podatkovne strukture, ki se uporablja pri procesu.
2. V strukturo se zapiše **Čas transakcije**
3. V strukturo se zapiše **UID kartice**  
UID se prebere iz kartice s postopkom izbire oz. antikolizije. Tako je UID pripravljen takoj, ko je kartica pripravljena za nadaljnja branja in pisanja podatkov.

- **Pogoji:**

Preveri se UID trenutne kartice in UID, ki se je shranil ob prekinitvi predhodnega postopka.

1. Če je trenutni UID enak UID-eju predhodnega postopka. Nadaljujemo postopek preverjanja uspešnosti zapisa vozovnice.
2. Če UID-eja nista enaka nadaljujemo postopek bodisi aktivacije vozovnice bodisi nakupa vozovnice.

Sledi operacija izbire ustrezne aplikacije (AID) s katero želi čitalnik komunicirati čitalnik pošlje ukaz:

- **Ukaz:** Select Application
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešna izbira aplikacije.

### 3.4.13.2. Stanje izbiri aplikacijo

- **Akcije**
  1. V stanju izberi aplikacijo čitalnik pošlje ukaz za avtentikacijo.
  2. Postopek avtentikacije.
- **Ukaz: Authenticate**

Aktivira se postopek avtentikacije med čitalnikom in kartico.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešna avtentikacija med čitalnikom in kartico.

### 3.4.13.3. Stanje avtentikacija

V stanju avtentikacija se po uspešno zaključeni avtentikaciji lahko prične postopek branja in pisanja v zelenih datotekah.

- **Akcije**

Branje verzije kartice oz. datoteke INFO.
- **Ukaz: Read Data**

Z ukazom aktiviramo postopek branja datoteke CHECK na kartici.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešno branje datoteke CHECK.

### 3.4.13.4. Stanje branje kontrolnih podatkov (datoteka CHECK)

- **Akcije:**
  1. Shranimo kontrolne podatke v strukturo.
  2. Branje dela datoteke PRODUCTS, kjer so zapisani statusi posameznih vozovnic.
- **Pogoji:**

Preverimo, če se naključno število na trenutni kartici ujema z naključnim številom predhodnega postopka aktivacije ali nakupa vozovnice. Če se trenutno naključno število ne ujema s predhodnim naključnim številom, lahko ugotovimo, da se vozovnica pri predhodnem postopku ni zapisala na kartico. Če se naključni števili ujemata, ugotovimo, da se je vozovnica pri predhodnem postopku uspešno zapisala na kartico.

  1. Če se naključni števili ne ujemata, nadaljujemo s postopkom aktivacije ali nakupa vozovnice.

## DS 4: Integracija IS sistemov prevoznika

---

2. Če se naključni števili ujemata, nadaljujemo s postopkom preverjanja uspešnega zapisa vozovnice.

- **Ukaz:** Read Value  
Z ukazom aktiviramo branje statusov obstoječih vozovnic na kartici.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje podatkov o statusih posameznih vozovnic.

### 3.4.13.5. Stanje branje statusov vozovnic (datoteka PRODUCTS)

- **Akcije:**
  1. Shranimo statute posameznih vozovnic, ki so prisotne na kartici.
  2. Branje informacij o vozovnicah oz. preostanek datoteke PRODUCTS. Beremo le tisti del datoteke, ki ustreza veljavnim vozovnicam.
- **Ukaz:** Read Data  
Z ukazom aktiviramo branje vozovnic prisotnih na kartici.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje podatkov o vozovnicah v datoteki PRODUCTS.

### 3.4.13.6. Stanje branje vozovnic (datoteka PRODUCTS)

- **Akcije:**  
Shranimo posamezne prebrane informacije o vozovnicah v podatkovno strukturo.
- **Pogoji:**
  1. Če se na kartici nahajajo relacijske vozovnice, nadaljujemo z branjem informacij o relacijah oz. datoteke LINE TARIFF DATA.
  2. Če se na kartici ne nahajajo relacijske vozovnice nadaljujemo z branjem zgodovine validacij oz. datoteke HISTORY-VALID.
- **Ukaz:** Read Data  
Z ukazom aktiviramo branje datoteke LINE TARIFF DATA (aktiviramo branje statusov relacij) ali datoteke HISTORY-VALID
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje podatkov.

### 3.4.13.7. Stanje branje statusov relacij (datoteka LINE TARIFF DATA)

- **Akcije:**
  1. Shranimo status posameznih relacij, ki so prisotne na kartici.
  2. Branje informacij o relacijah oz. preostanek datoteke LINE TARIFF DATA.  
Beremo le tisti del datoteke, ki ustreza veljavnim relacijam.
- **Ukaz:** Read Data  
Z ukazom aktiviramo branje ustreznih relacij prisotnih na kartici.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje podatkov o relacijah v datoteki LINE TARIFF DATA.

### 3.4.13.8. Stanje branje relacij (datoteka LINE TARIFF DATA)

- **Akcije:**  
Shranimo posamezne prebrane informacije o relacijah v podatkovno strukturo.
- **Ukaz:** Read Data  
Z ukazom aktiviramo branje zgodovine validacij oz. datoteke HISTORY-VALID.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje podatkov o zgodovini validacij oz. datoteke HISTORY-VALID.

### 3.4.13.9. Stanje branje zgodovine validacij (datoteka HISTORY-VALID)

- **Akcije:**
  1. Shranimo informacije o zgodovini validacij v podatkovno strukturo.
  2. Sprožimo postopek spremembe STATUS-a (brisanje) vozovnic, ki so bile v predhodnem postopku zapisane na kartico.
- **Ukaz:** Write Data  
Z ukazom aktiviramo postopek zapisa statusa vozovnice v datoteki PRODUCTS.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.

### 3.4.13.10. Stanje zapis statusov vozovnic (datoteka PRODUCTS)

V stanju izvedemo ukaz za potrditev zapisanih podatkov na kartico.

## DS 4: Integracija IS sistemov prevoznika

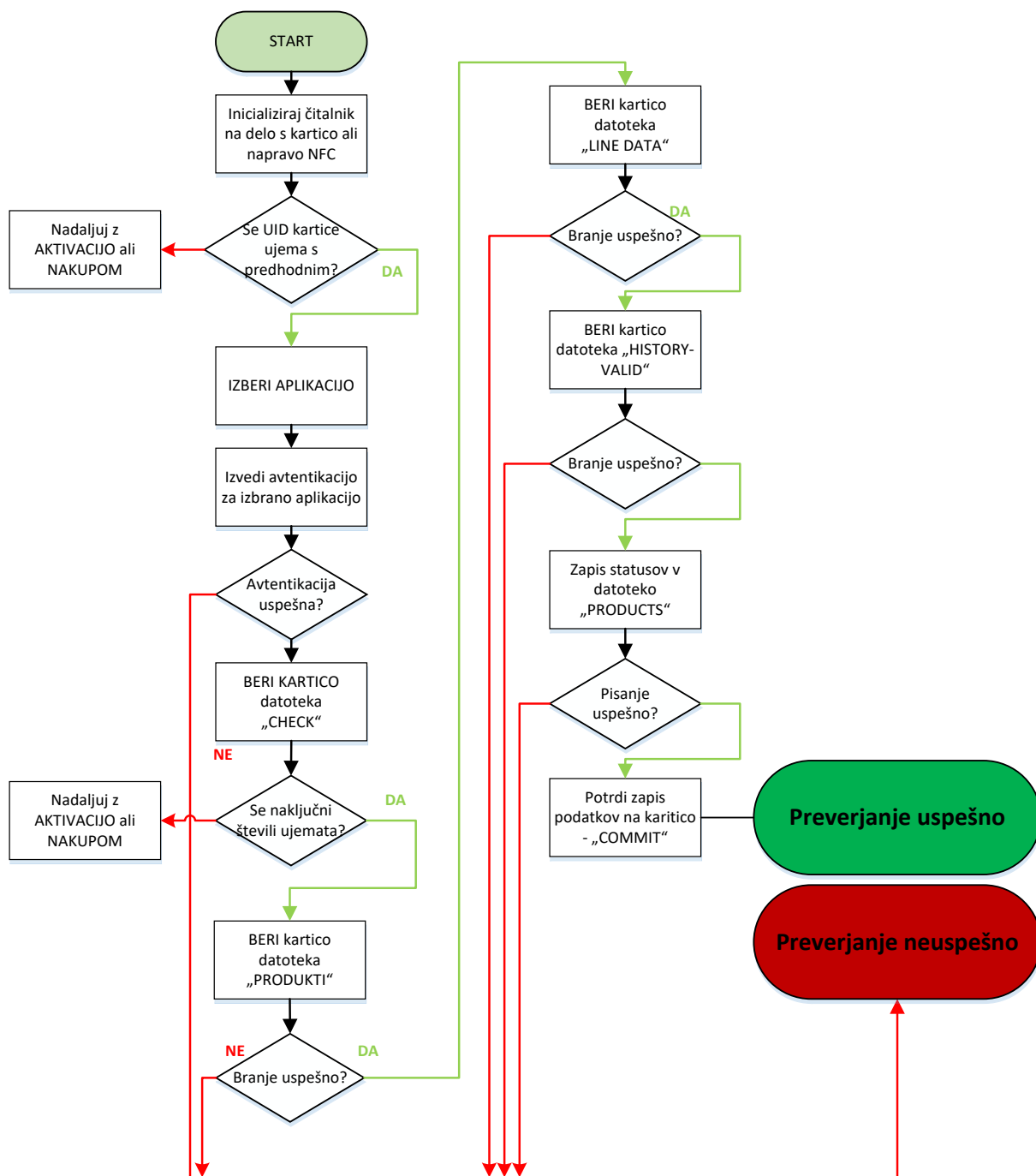
---

- **Ukaz:** Commit Transaction  
Z ukazom aktiviramo postopek potrjevanja zapisanih podatkov na kartico.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno potrjevanje podatkov.

### 3.4.13.11. Stanje potrdi zapis podatkov

Po uspešnem potrjevanju podatkov na kartici ponovimo prvotni postopek aktivacija ali nakupa vozovnice.

#### DS 4: Integracija IS sistemov prevoznika



SLIKA 3.17: PREVERJANJE USPEŠNOSTI ZAPISA VOZOVNICE

### 3.5. ČASOVNE OMEJITVE PRI POSTOPKIH KOMUNIKACIJE MED ČITALNIKOM IN KARTICO

## DS 4: Integracija IS sistemov prevoznika

Časovne omejitve pri postopkih komunikacije med čitalnikom in kartico lahko določimo na podlagi hitrosti komunikacije med čitalnikom in kartico, količine podatkov ter ocene trajanja preostalih postopkov, ki se izvajajo na čitalniku (npr. izvajanje programske kode, procesiranje algoritmov, preverjanje, če se kartica nahaja na črni ali izvršni listi). Pri določitvi časovnih omejitev si lahko pomagamo z izračunom časa komunikacije ob upoštevanju najnižje možne hitrosti komunikacije (105 kb/s) ter največje možne prenesene količine podatkov na kartici.

Za oceno potrebnega časa se predvideva, da ima terminal naložene vse potrebne podatke in lahko transakcijo zaključi avtonomno, brez interakcije z zalednim sistemom.

Na podlagi izračuna lahko določimo čas potreben izključno za prenos podatkov med kartico in čitalnikom. V primeru maksimalnega števila podatkov na kartici za validacijo ali nakup skupaj z operacijo branje info znaša čas za prenos podatkov okvirno 0.3 s. Če k temu prištejemo še oceno za procesiranje podatkov na čitalniku, lahko določimo maksimalni časovni okvir za izvajanje offline kartičnih transakcij, ki znaša 0.7 s.

Čas validacije ali nakupa se lahko nekoliko spremeni v primeru, če se kartica nahaja na izvršni listi in je za to kartico potrebno opraviti določeno opravilo, saj se s tem prvotna operacija za nekaj časa prekine. Izmed obstoječih opravil izvršne liste (prenos vozovnice na kartico, brisanje vozovnice na kartici, nastavitve novega statusa na kartici, začasno onemogočanje uporabe kartice, online preverjanje kartice, začasno onemogočanje uporabe vozovnice) je časovno najbolj potratno opravilo online preverjanje, saj je potrebno trenutno kartično operacijo začasno prekiniti in izvesti ustrezno transakcijo na zaledni sistem. Po posredovanju ustreznih podatkov, se lahko začasno prekinjena transakcija nadaljuje.

### 3.6. PODATKOVNA STRUKTURA

#### 3.6.1. POZICIONIRANJE APLIKACIJE IJPP

Za aplikacijo IJPP je na kartici namenjen **AID (ApplicationID) = 0x05**

Vsi znakovni nizi (stringi) na kartici so opisani z windows-1250 CP.

**OPOMBA: AID za aplikacijo IJPP se po usklajevanju z obstoječimi tehnološkimi ponudniki lahko še spremeni.**

#### 3.6.2. STRUKTURA POSAMEZNIH DATOTEK

DS 4: Integracija IS sistemov prevoznika

Aplikacija: IJPP - AID: 0x000005 v1.0			
Ime datoteke	Vsebina		
INFO	FID: 0x00		Tip: DATA FILE
	Vsebina:		Size [B]: 32
	VERSION_INFO		1
	VERSION_CARD_INFO		1
	VERSION_DATA_INDEX		1
	VERSION_DATA_FILE		1
	RFU		1
	RFU		1
	RFU		1
	VERSION_CHECK		1
	VERSION_PRODUCTS		1
	VERSION_HISTORY_VALIDATION		1
	VERSION_RELATIONS		1
RFU		Rezervirano za kasnejšo uporabo	21

CARD INFO	FID: 0x01		Tip: BACKUP DATA FILE
	Vsebina:		Size [B]: 256
	STATUS	Neaktivna 01 Aktivna 02 Blokirana 04 Začasno onemogočena 08	1
	DATUM PRIČETKA VELJAVE KARTICE	Veljavnost kartice	4
	DATUM KONCA VELJAVE KARTICE		4
	TIP PERSONIFIKACIJE	Ni za person. 01 Neperson. Namenjena za person. 02 Neperson. Personificirana 04 Personif.	1
	POSEBNI STATUS *	0x00 – IJPP	1
	IME, PRIIMEK, NASLOV	ime:priimek:naslov:poštna št.:pošta	200
	EMŠO	13-mestna EMŠO	13
	DAVČNA ŠTEVILKA	8-mestna DŠ	8
	DATUM INICIAL. KARTICE		4
	DATUM PERSONIFIKACIJE	Če kartica ni person: 0xFFFFFFFF	4
	RFU	Rezervirano za kasnejšo uporabo	16

DATA INDEX	FID: 0x 02		Tip: BACKUP DATA FILE
------------	------------	--	-----------------------



#### DS 4: Integracija IS sistemov prevoznika

Vsebina:		Size [B]:	48
SP_ID1, SP ID2... SP ID8	Število ponudnikov storitev: 8, vsak 2B	16	
FORMAT** (Glej spodnjo tabelo!!)	0x00, 0x01, 0x02, 0x03	1	
OFFSET	Način, verzija, format zapisa		
LENGTH	Pričetek zapisa od danega SP	2	
	Dolžina zapisa od danega SP	1	

FORMAT 0x00	DATA FILE struktura
Definicija potnikovega statusa	<b>2B</b> ID Statusa <b>4B</b> Datum pričetka veljave statusa (nekateri statusi so časovno neomeni *) <b>4B</b> Datum konca veljave statusa (nekateri statusi so časovno neomejeni *)

\*časovno omejene/neomejene statuse določa vrsta statusa v tarifi (glej strukturo TariffStatus)

FORMAT 0x01	DATA FILE struktura
Podatki o vlogi za subvencioniranje	<b>20B</b> Šifra vloge za subvencioniranje <b>4B</b> Datum pričetka veljave vloge <b>4B</b> Datum konca veljave vloge <b>1B</b> Vrsta vloge (MESEČNA=1 M10=2) <b>1B</b> Status (Dijak, Študent, Udeleženec izobraževanja odraslih) <b>1B</b> TariffLocationID za prvi mestni promet na vlogi <b>1B</b> Število področij za prvi mestni promet na vlogi <b>1B</b> TariffLocationID za drugi mestni promet na vlogi <b>1B</b> Število področij za drugi mestni promet na vlogi <b>1B</b> TariffLocationID za tretji mestni promet na vlogi <b>1B</b> Število področij za tretji mestni promet na vlogi <b>1B</b> Index vloge – zaporedna številka vloge na kartici (01, 02, ...)

FORMAT 0x02	DATA FILE struktura
Definicija službenega statusa	<b>2B</b> ID Statusa (1=voznik, 2=sprevodnik, 3=kontrolor) <b>4B</b> ID Prevoznika <b>4B</b> Datum pričetka veljave statusa <b>4B</b> Datum konca veljave statusa

DATA FILE	FID: 0x 03	Tip:BACKUP DATA FILE
Vsebina:		Size [B]: 128
DATA	Podatki, ki pripadajo posameznemu SP iz datoteke DATA INDEX	

#### DS 4: Integracija IS sistemov prevoznika

RFU	FID: 0x 04		Tip: VALUE FILE
	Vsebina:		Size [B]: 64
	RFU	Rezervirano za kasnejšo uporabo	

RFU	FID: 0x 05		Tip: CYCLIC RECORD FILE
	St. record.: 3		
	Vsebina zapisa:		Size [B]: 26
	RFU	Rezervirano za kasnejšo uporabo	

RFU	FID: 0x 06		Tip: CYCLIC RECORD FILE
	St. record.: 3		
	Vsebina zapisa:		Size [B]: 26
	RFU	Rezervirano za kasnejšo uporabo	

<b>CHECK</b> Kontrola stanja	FID: 0x 07		Tip: BACKUP DATA FILE
	Vsebina:		Size [B]: 32
	RANDOM-Transaction ID	4B dolg posebni transaction ID, vpiše terminal	4
	LastTransaction Time	čas zadnje uspešne transakcije	4
	TransactionExecutionID		6
	Number of warnings	število opozoril s strani kontrolorske službe	1
	RFU		17

## DS 4: Integracija IS sistemov prevoznika

PRODUCTS	FID: 0x 08			Tip: BACKUP DATA FILE
	Vsebina:		Size [B]:	440
	STATUS1, STATUS2,...STATUS10	Vsak bajt predstavlja status posameznega zapisa znotraj datoteke. Zapisi so fiksne dolžine, možne vrednosti posameznega statusa so: 0x01: Neaktivna 0x02: Aktivna 0x04: Neveljavna (porabljena) 0x08: Začasno onemogočena	8	
	TARIFF_LOCATION_ID		2	
	UNIQUE TICKET ID	ID vozovnice - mora biti unikaten 6B : - 3B: TerminalID - 3B: Številka vozovnica	6	
	STATUS POTNIKA	ID potnikovega statusa (npr. ŠOLAR)	2	
	PRODUKT	ID produkta iz tarife	2	
	ŠT. PREHODOV	Število prehodov med conami	2	
	ŠT. KART	Število produktov	1	
	DATUM ZAČETKA VELJAVNOSTI		4	
	DATUM KONCA VELJAVNOSTI.		4	
	DATUM KONCA VELJAVNOSTI AKTIVACIJE		4	
	VREDNOST OB NAKUPU	Vrednost produkta ob nakupu	4	
	ID TARIFNEGA SISTEMA		2	
	NAČIN TARIFNEGA SISTEMA	0x00 - conski, 0x01 - linijski, 0x02 - kombinirani	1	
	LINIJA	ID linije	4	
	VSTOPNA CONA - Conski	ID vstopne cone	2	
	INDEX RELACIJE	Index za relacije v datoteki FID=0x0A – v primeru relacijske vozovnice	1	
	TRENTNI MESEC	Trenutni mesec koriščenja kuponov	1	
	ŠTEVILO KUPONOV	število preostali kuponov za določen mesec	1	

## DS 4: Integracija IS sistemov prevoznika

	ID SKUPINE PRODUKTOV	Če je več produktov prodanih kot paket, se sem notri vpiše ID (zaporedna številka skupine). 0x00=ni skupine	1
	RFU	Rezervirano za kasnejšo uporabo	10

<b>HISTORY-VALID</b> Zgodovina validacij	Tip: CYCLIC RECORD FILE:		
	FID: 0x 09 St. record.: 3		
	Vsebina:		Size [B]: 32
	TID	Terminal ID	4
	ČAS		4
	POSTAJA	ID Postaje, na kateri se je validiral potnik	4
	LINIJA	ID Linije, na kateri se je validiral potnik	4
	INDEKS produkta	Indeks pripadajočega zapisa v datoteki "PRODUKTI"	1
	UNIQUE TICKET ID	Številka vozovnice iz datoteke PRODUKTI	6
	BUS NUMBER	Številka avtobusa	3
	RFU	Rezervirano za kasnejšo uporabo	6

<b>RELATIONS</b> Podatki o relacijah	Tip: BACKUP DATA FILE		
	FID: 0x 0A		
	Vsebina:		Size [B]: 640
	STATUS1, STATUS2,...	Valid-Permanent: 0x01, Valid-Temporary: 0x02 Invalid: 0x04 Začasno onemogočena: 0x08	5
	FORMAT IDENTIFIER		1
	LINE DEFINITION		1
	LINE ID		2
	VALID FROM	Začetek veljavnosti relacije	4
	VALID TO	Konec veljavnosti relacije	4
	DISTANCE	Razdalja v 0.1km (15,5km = 155)	2
	START STATION NAME	Ime začetne postaje (10 znakov)	10
	STOP STATION NAME	Ime končne postaje (10 znakov)	10
	NUMBER OF STATIONS IN RECORD	Število postaj, ki so uporabljene	1
	BUS STATION ID1		4
	BUS STATION ID2		4
	BUS STATION ID3		4
	BUS STATION ID4		4
	BUS STATION ID5		4
	BUS STATION ID6		4

#### DS 4: Integracija IS sistemov prevoznika

BUS STATION ID7		4
BUS STATION ID8		4
BUS STATION ID9		4
BUS STATION ID10		4
BUS STATION ID11		4
BUS STATION ID12		4
BUS STATION ID13		4
BUS STATION ID14		4
BUS STATION ID15		4
BUS STATION ID16		4
BUS STATION ID17		4
BUS STATION ID18		4
BUS STATION ID19		4
BUS STATION ID20		4
ONE-WAY	0x01 = relacija je enosmerna	1
SUBSIDIZED	0x00 = relacija ni subvencionirana 0x01, 0x02... = relacija je subvencionirana, vrednost pomeni index vloge v FID=0x03 (format 0x01)	1
START STATION LOCATION_ID	LocationID, če je v mestnem prometu	1
STOP STATION LOCATION_ID	LocationID, če je v mestnem prometu	1
RFU		8

### 3.6.3. OPIS POSAMEZNIH DATOTEK

#### 3.6.3.1. FID 0x00 (INFO)

Ta datoteka vsebuje informacije o različnih posameznih datotekah na kartici. Aplikacija za IJPP se lahko s časom zaradi potreb ali zahtev spreminja. Vsaka sprememba bo verzionirana in podatki o teh verzijah sprememb se nahajajo v datoteki FID 0x00.

VERSION_INFO	Informacije o strukturi FID0x00
VERSION_CARD_INFO	Informacije o strukturi FID0x01
VERSION_DATA_INDEX	Informacije o strukturi FID0x02
VERSION_DATA_FILE	Informacije o strukturi FID0x03
VERSION_BALANCE	Informacije o strukturi FID0x04
VERSION_HISTORY_CREDIT	Informacije o strukturi FID0x05
VERSION_HISTORY_DEBIT	Informacije o strukturi FID0x06
VERSION_CHECK	Informacije o strukturi FID0x07
VERSION_PRODUCTS	Informacije o strukturi FID0x08
VERSION_HISTORY_VALIDATION	Informacije o strukturi FID0x09
VERSION_RELATIONS	Informacije o strukturi FID0x0A

## DS 4: Integracija IS sistemov prevoznika

### 3.6.3.2. FID 0x01 (CARD INFO)

Ta datoteka vsebuje osnovne informacije o kartici in podatke o imetniku kartice

STATUS	Ena od vrednosti 0x01 = Neaktivna 0x02 = Aktivna 0x04 = Blokirana 0x08 = Začasno onemogočena
DATUM PRIČETKA VELJAVE KARTICE	Datum kdaj prične kartica veljati. Glej postopek »Postopek za zapis datuma na kartico« s katerim datum zapišemo s 4 zlogi.
DATUM KONCA VELJAVE KARTICE	Datum kdaj kartica preneha veljati. Glej postopek »Postopek za zapis datuma na kartico« s katerim datum zapišemo s 4 zlogi.
TIP PERSONIFIKACIJE	Identifikator za kakšno uporabo je kartica namenjena. Običajno gre za imensko ali neimensko. Ena od vrednosti: 0x01 = neimenska (anonimna kartica) 0x04 = imenska (kartica namenjena točno določenemu uporabniku).
POSEBNI STATUS	Status s katerim določamo namembnost kartice. Kartice so lahko namenjene potnikom, lahko pa so namenjene drugemu osebju za identifikacijo ali prijavo v razne storitve v sistemu. Sem spadajo recimo tudi vozniki, vzdrževalci, kontrolorji, testerji itd. Kartica namenjena potnikom ima to vrednost postavljeno na 0x00.
IME, PRIIMEK, NASLOV	Podatki uporabnika. Zapis mora na kartico biti zapisan v naslednji obliki: Ime:priimek:naslov:poštna_št:Pošta\0 Dvopičje služi kot separator med posameznimi sklopi. Na koncu mora biti niz zaključen z '\0' (null terminated).
EMŠO	13 mestna EMŠO številka uporabnika kartice (opcijsko).
DAVČNA ŠTEVILKA	8 mestna davčna številka uporabnika (opcijsko).
DATUM INICIAL. KARTICE	Datum kdaj je bila kartica inicializirana. Inicializacija kartice pomeni pripravljena struktura kartice in zapisani podatki v sistem. Glej postopek »Postopek za zapis datuma na kartico« s katerim datum zapišemo s 4 zlogi.
DATUM PERSONIFIKACIJE	Datum kdaj je bila kartica dodeljena potniku. Običajno sta datuma inicializacije in personifikacije enaka. Glej postopek »Postopek za zapis datuma na kartico« s katerim datum zapišemo s 4 zlogi.
RFU	Neizkoriščeni zlogi, ki so rezervirani za kasnejšo uporabo. Vsebina neizkoriščenih zlogov ni pomembna, vendar je iz dobre prakse postavljena na vrednost 0x00.

### 3.6.3.3. FID 0x02 (DATA INDEX)

Ta datoteka vsebuje informacije o zapisu statusov uporabnika.

SP_ID1, SP_ID2...	Identifikatorji ponudnikov storitev. Vsak ponudnik storitve se lahko zapiše z 2 zlogoma.
FORMAT	Identifikator, ki določa strukturo s katero je opisana pravica uporabnika.
OFFSET	Odmik na katerem se prične vsebina opisa statusa potnika v FID0x03
LENGTH	Dolžina v zlogih, ki jo zavzame zapis statusa v FID 0x03.

### 3.6.3.4. FID 0x03 (DATA FILE)

Ta datoteka služi kot vsebnik za podatke o statusih potnika. Lahko si jo predstavljamo kot polje 128 zlogov, ki so na voljo za zapis statusov.

#### DS 4: Integracija IS sistemov prevoznika

DATA	Vsebina zlogov, ki določajo statute uporabnika.
------	---

##### 3.6.3.5. FID 0x07 (CHECK)

Ta datoteka vsebuje podatke s pomočjo katerih se na napravah in v sistemu izvajajo razne kontrole stanja kartice.

RANDOM-Transaction ID	Posebni identifikator transakcije, ki ga zapiše terminal
Last Transaction Time	Čas zadnje transakcije, ki je bila urejana na kartici. Glej postopek »Postopek za zapis datuma na kartico« s katerim datum zapišemo s 4 zlogi.
TransactionExecutionID	Identifikator zadnje transakcije
Number of warnings	število opozoril s strani kontrolorske službe
RFU	Neizkoriščeni zlogi, ki so rezervirani za kasnejšo uporabo. Vsebina neizkoriščenih zlogov ni pomembna, vendar je iz dobre prakse postavljena na vrednost 0x00.

##### 3.6.3.6. FID 0x08 (PRODUCTS)

Ta datoteka vsebuje podatke o produktih, ki se nahajajo na kartici.

STATUS1, STATUS2...STATUS8	Statusi produktov, ki so na kartici. Možne so naslednje vrednosti:  Neaktiven = 0x01 [označuje veljaven produkt, ki še ni bil aktiviran na vozilu. Takšni so vsi produkti takoj po nakupu.]  Aktiven = 0x02 [označuje veljaven produkt, ki je že bil aktiviran na vozilu. Takšni so vsi produkti, takoj ko uporabnik prisloni kartico k napravi za validacijo vozovnic v vozilu].  Neveljaven = 0x04 [Označuje prazno mesto kamor se lahko zapiše nov produkt]
TARIFF_LOCATION_ID	Identifikator lokacije tarife, definira za lokacijsko katero tarifo gre (vezano na lokacijo mestnega prometa trenutno).
PRODUCT_ID	Unikaten identifikator, s pomočjo katerega vemo, kdo je produkt prodal. Sestavljen je na naslednji način: [3B: Identifikator naprave s katero je bil produkt zapisan na kartico] [3B: Enoličen identifikator vozovnice v sistemu prodajalca]
STATUS POTNIKA	Identifikator, ki določa status potnika
PRODUKT	Identifikator, ki določa produkt (MESEČNA, LETNA, POLLETNA...)+
ŠT. PREHODOV	Identifikator, ki določa št. prehodov do katerih je potnik upravičen v mestnem prometu.
ŠT. KART	Koliko vozovnic predstavlja produkt. Koristimo v primeru žetonov ali kuponov.
DATUM ZAČ. VELJ.	Datum od kdaj produkt velja. Glej postopek »Postopek za zapis datuma na kartico« s katerim datum zapišemo s 4 zlogi.
DATUM KONCA VELJ.	Datum do kdaj produkt velja. Glej postopek »Postopek za zapis datuma na kartico« s katerim datum zapišemo s 4 zlogi.
DATUM KONCA VELJAVNOSTI AKTIVACIJE	Datum do kdaj velja aktiviran produkt. Glej postopek »Postopek za zapis datuma na kartico« s katerim datum zapišemo s 4 zlogi.

#### DS 4: Integracija IS sistemov prevoznika

VREDNOST OB NAKUPU	Vrednost produkta ob nakupu, katero je plačal potnik. $1200 = 12.00\text{€} \cdot 500 = 5.00\text{€}$ . Vrednost je namreč predstavljena v obliki XYY (XX = celi del, YY = del za decimalno piko).
ID TARIFNEGA SISTEMA	Identifikator tarifnega sistema po katerem je bil produkt kupljen.
NAČIN TARIFNEGA SISTEMA	Identifikator ali gre za medkrajevni produkt ali gre za mestni produkt. Možne so naslednje vrednosti:  0x00 = conski (mestna vozovnica) 0x01 = linijski (medkrajevna vozovnica)
LINIJA	Identifikator linije. Zapiše naprava na vozilu v kolikor je produkt kupljen na vozilu.
VSTOPNA CONA	Identifikator vstopne cone v kolikor je produkt kupljen na vozilu.
INDEX RELACIJE	Index, ki kaže na zapis v datoteki FID 0x0A in določa za katero relacijo / linijo je medkrajevni produkt namenjen.
TRENTNI MESEC	Uporablja se v primeru koriščenja kuponov in pove trenutni mesec za katerega so kuponi namenjeni. 1 = JANUAR 2 = FEBRUAR ... 12 = DECEMBER
ŠTEVILO KUPONOV	V Primeru da je produkt kuponski, ta vrednost predstavlja količino kuponov, ki jo potnik lahko koristi.
ID SKUPINE PRODUKTOV	Identifikator, ki se dodeli skupini produktov – recimo ,da se kupi medkrajevna vozovnica skupaj z mestno – kot paket. Na ta način se označi, da nekaj vozovnic spada v isti paket.
RFU	Neizkoriščeni zlogi, ki so rezervirani za kasnejšo uporabo. Vsebina neizkoriščenih zlogov ni pomembna, vendar je iz dobre prakse postavljena na vrednost 0x00.

#### 3.6.3.7. FID 0x09 (HISTORY-VALID)

Ta datoteka služi kot dnevnik validacij. Te podatke zapisujejo naprave na vozilih.

TID	Identifikator naprave na kateri je bila izvedena validacija.
ČAS	Čas kdaj je bila validacija izvedena. Glej postopek »Postopek za zapis datuma na kartico« s katerim datum zapišemo s 4 zlogi.
POSTAJA	Identifikator postaje na kateri je bila validacija izvedena.
LINIJA	Identifikator linije na kateri je bila validacija izvedena.
INDEKS produkta	Indeks produkta, s katerim se je potnik validiral. Indeks kaže na FID0x08.
UNIQUE TICKET ID	Serijska številka vozovnice
BUS NUMBER	Identifikator vozila na katerem je bila izvedena validacija.
RFU	Neizkoriščeni zlogi, ki so rezervirani za kasnejšo uporabo. Vsebina neizkoriščenih zlogov ni pomembna, vendar je iz dobre prakse postavljena na vrednost 0x00.



## DS 4: Integracija IS sistemov prevoznika

### 3.6.3.8. FID 0x0A (RELATIONS)

Ta datoteka vsebuje podatke o relacijah na katerih lahko potnik potuje v medkrajevnem prometu.

STATUS1, STATUS2,...	Statusi relacij zapisani na kartici. Lahko imajo eno od naslednjih vrednosti.  0x01 = trajno veljavna relacija 0x02 = začasno veljavna relacija (zapiše naprava na vozilu, v primeru, da uporabnik kupi medkrajevno vozovnico na vozilu).  0x04 = neveljavna relacija (označuje prazno mesto kamor lahko zapišemo novo relacijo).
FORMAT IDENTIFIER	Identifikator zapisa relacije: 0x01 – privzet identifikator relacije. Trenutno edina vrednost, ki se lahko zapiše na kartico.
LINE DEFINITION	Edina trenutno veljavna vrednost je 0x01.
LINE ID	Identifikator linije – vožnje iz voznega reda, zapiše se samo na vozilu, če je uporabnik kupil medkrajevno vozovnico na vozilu.
VALID FROM	Veljavnost od kdaj velja relacija. Glej postopek »Postopek za zapis datuma na kartico« s katerim datum zapišemo s 4 zlogi.
VALID TO	Veljavnost do kdaj velja linija. Glej postopek »Postopek za zapis datuma na kartico« s katerim datum zapišemo s 4 zlogi.
DISTANCE	Razdalja med vstopno in izstopno postajo.
START STATION NAME	Ime vstopne postaje.
STOP STATION NAME	Ime izstopne postaje.
NUMBER OF STATIONS IN RECORD	Število postaj v zapisu. Privzeta vrednost je MAX (trenutno je to 20).
BUS STATION ID1 ... BUS STATION ID20	Identifikatorji vstopne, izstopne in vseh prestopnih postaj s katerimi smiselno definiramo odseke, ki določajo pot med vstopno in izstopno postajo potnika. Ti odseki hkrati določajo segmente na katerih lahko potnik vstopa in izstopa na vozilo.
ONE WAY	Identifikator enosmerne vozovnice. 0 = dvosmerna 1 = enosmerna
SUBVENCIONIRANA	Identifikator subvencionirane vozovnice na tej relaciji. 0 = NI SUBVENCIONIRANA >0 = SUBVENCIONIRANA, vrednost pomeni index vloge, torej polje Index Vloge v opisu vloge (format=0x01) v DATA FILE (FID=0x03)
START STATION LOCATION_ID	TariffLocationID za začetno postajo – če je le ta v mestnem prometu. Drugače je postavljen na 0.
STOP STATION LOCATION_ID	TariffLocationID za končno postajo – če je le ta v mestnem prometu. Drugače je postavljen na 0.
RFU	Neizkoriščeni zlogi, ki so rezervirani za kasnejšo uporabo. Vsebina neizkoriščenih zlogov ni pomembna, vendar je iz dobre prakse postavljena na vrednost 0x00.

### 3.7. VARNOSTNA SPECIFIKACIJA

#### 3.7.1. OPISI VARNOSTNIH / KRIPTOGRAFSKIH PROTOKOLOV

##### 3.7.1.1. Izbor kriptografskih algoritmov

V sistemu IJPP so uporabljeni kriptografski algoritmi AES z dolžino ključa 128 bitov in 192 bitov, DES, trojni DES z uporabo 3 ključev (v nadaljevanju 3KDES) dolžine 168 bitov in algoritmi za SSL/TLS.

Kriptografski algoritem AES je skladen s standardom FIPS PUB 197.

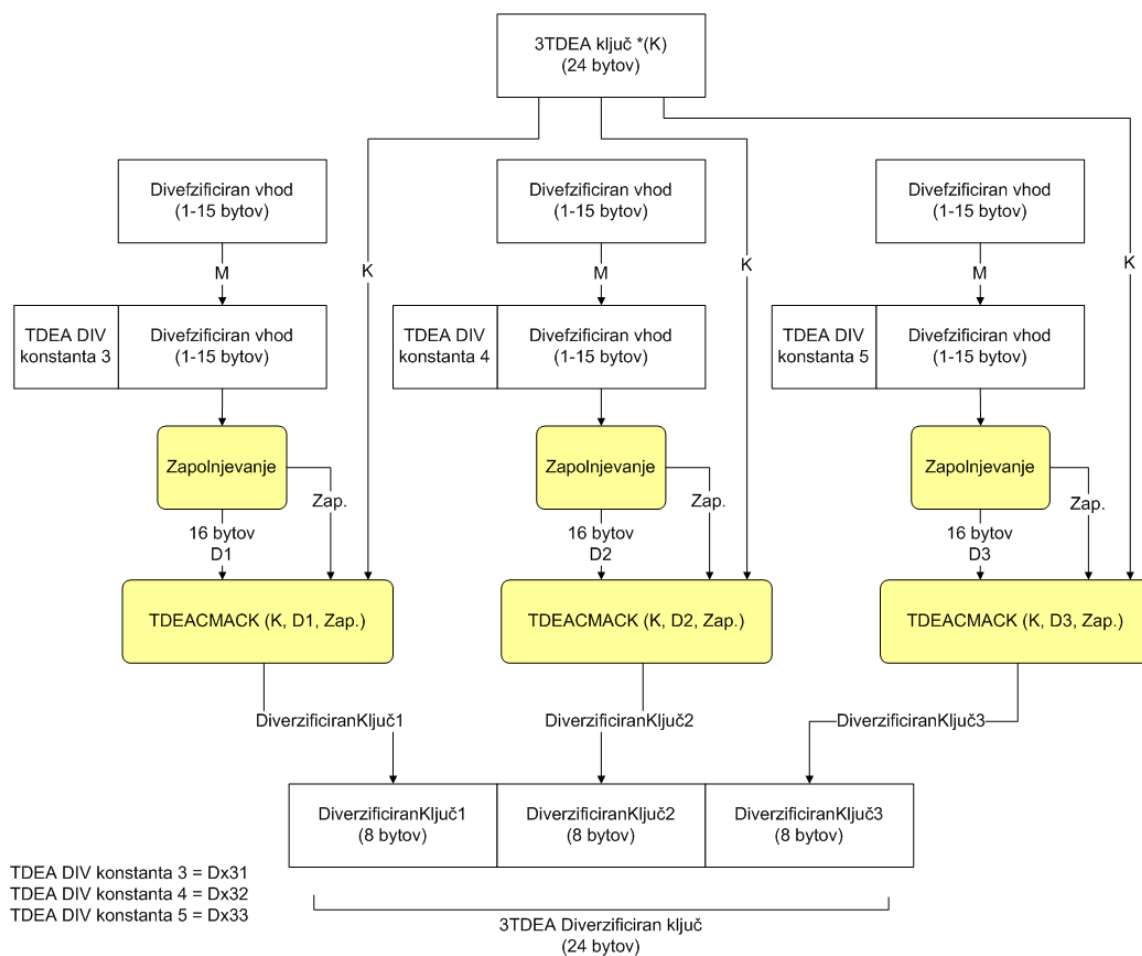
Kriptografski algoritem 3KDES definira prva možnost standarda FIPS PUB 46-3 in ANSI x9.52-1998. Definirana dolžina posameznega ključa je 8 bytov, pri katerem je zadnjih 8 bitov uporabljenih za opis verzije ključa in ne za paritetne bite.

##### 3.7.1.2. Diverzifikacija ključev

Diverzifikacija je proces, s katerim se iz enega glavnega oz. baznega ključa, generira unikatne ključke, na podlagi vhodnih unikatnih podatkov. Z diverzificiranimi ključki so opremljene brezstične kartice, terminalska oprema, pa ima v varnostnem elementu nameščen bazni ključ. V primeru vdora v eno kartico, tako ni ogroženo delovanje celotnega sistema.

V sistemu IJPP je uporabljena metoda CMAC (definirana s standardom NIST 800-38B) za generiranje 3KDES avtentikacijskih ključev. Slika 3.18 prikazuje blokovni diagram algoritma za diverzifikacijo 3KDES ključa.

## DS 4: Integracija IS sistemov prevoznika



SLIKA 3.18: BLOKOVNI DIAGRAM ALGORITMA DIVERZIFIKACIJE ZA KLJUČE 3KTDES

Ključni za brezstične kartice se generirajo ob zapisu nove IJPP aplikacije. Podatki za diverzifikacijo so sestavljeni iz:

- CUID številke (8 bajtov),
- AID IJPP aplikacije (3 bajti)
- Številka ključa za avtentikacijo (1 bajt)

Med uporabo brezstične IJPP kartice, terminalska oprema, iz podatkov brezstične kartice in baznega ključa, generira diverzificiran ključ in opravi postopek avtentikacije.

### 3.7.1.3. Avtentikacija

Uporablja se tri-stopenjska avtentikacija, ki omogoča varno sporočanje s pomočjo kriptirane komunikacije.

## DS 4: Integracija IS sistemov prevoznika

---

V postopku avtentikacije se uporabi varnostni element, ki ima nameščene ustrezne ključe. Avtentikacija med terminalsko opremo in IJPP kartico je tipa 3KTDDES, ki uporablja 3 ključe, velikosti 168 bitov.

### *3.7.2. UPORABA VARNOSTNEGA ELEMENTA PRI DELU S IJPP KARTICO (OZ. APLIKACIJO NA KARTICI)*

Vsak terminal, ki je vključen v sistem IJPP, mora imeti varnostni element in pripadajočo programsko opremo, ki omogoča njegovo uporabo. Varnostni element mora omogočati varno shranjevanje kriptografskih ključev in certifikatov ter izvajati kriptografske algoritme, opisane v poglavju 3.7.1.1.

Ena od možnosti pri izbiri varnostnega elementa, je SAM modul, ki je tej specifikaciji opisan kot primer. Izbira vrste varnostnega elementa ni omejena, vendar mora opravljati enakovredne operacije in funkcije kot SAM modul.

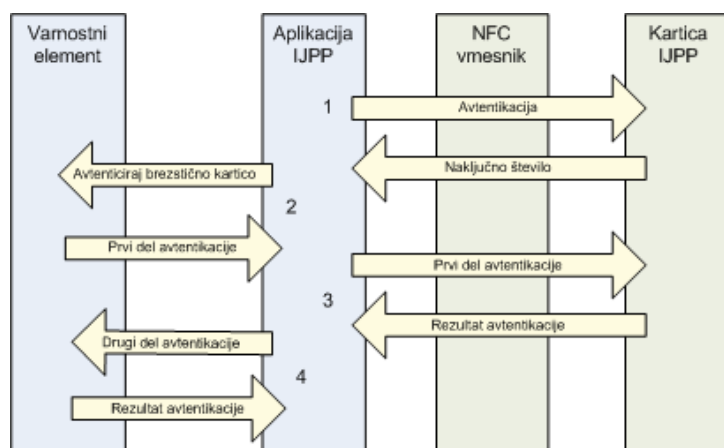
Pri komunikaciji z IJPP kartico se uporablja avtentikacija, kriptiranje in dekriptiranje podatkov, ki se izmenjujejo med izvajanjem transakcije.

Postopek avtentikacije je naslednji:

1. Aplikacija na terminalski opremi (IJPP aplikacija) prične z avtentikacijo IJPP kartice.
2. IJPP aplikacija posreduje odgovore med IJPP kartico in varnostnim elementom.
3. Varnostni element sporoči IJPP aplikacija izid avtentikacije.
4. Nadaljnja komunikacija je kriptirana s ključem seje.

Po uspešni izvedbi avtentikacije imata varnostni element in kartica IJPP ključ seje, s katerim izvajata varno komunikacijo. Slika 3.19 prikazuje potek komunikacije med postopkom avtentikacije.

#### DS 4: Integracija IS sistemov prevoznika

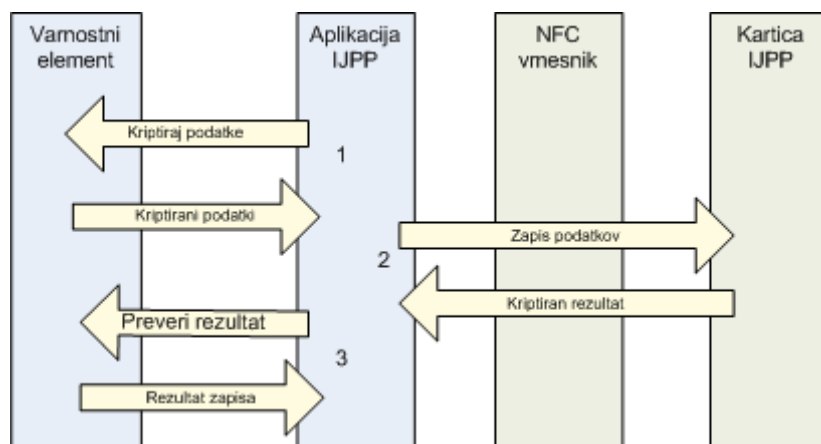


SLIKA 3.19: POTEK KOMUNIKACIJE MED AVTENTIKACIJO IJPP KARTICE

Postopek zapisa podatkov na IJPP kartico je naslednji:

1. IJPP aplikacija pošlje podatke za kriptiranje varnostnemu elementu, ki vrne kriptirane podatke.
2. IJPP aplikacija pošlje ukaz za zapis podatkov s kriptiranimi podatki kartici IJPP, ki vrne kriptiran rezultat.
3. IJPP aplikacija preko varnostnega elementa preveri rezultat zapisa.

Slika 3.20 prikazuje potek zapisa podatkov na IJPP kartico.



SLIKA 3.20: POTEK ZAPISA NA IJPP KARTICO

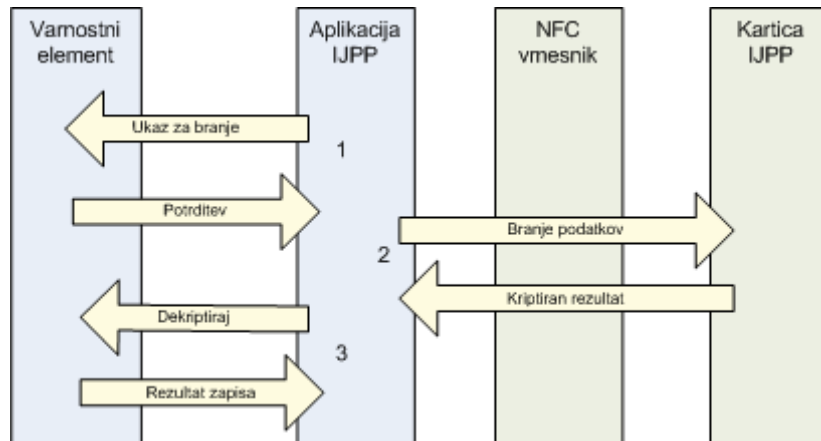
Postopek branja podatkov iz IJPP kartice je naslednji:

1. IJPP aplikacija pošlje ukaz za branje varnostnemu elementu, da osveži kriptografski števec.

#### DS 4: Integracija IS sistemov prevoznika

2. IJPP aplikacija pošlje nekriptiran ukaz za branje podatkov na kartico IJPP, ki vrne kriptirane podatke.
3. IJPP aplikacija preko varnostnega elementa dekriptira podatke.

Slika 3.21 prikazuje potek zapisa podatkov na IJPP kartico.



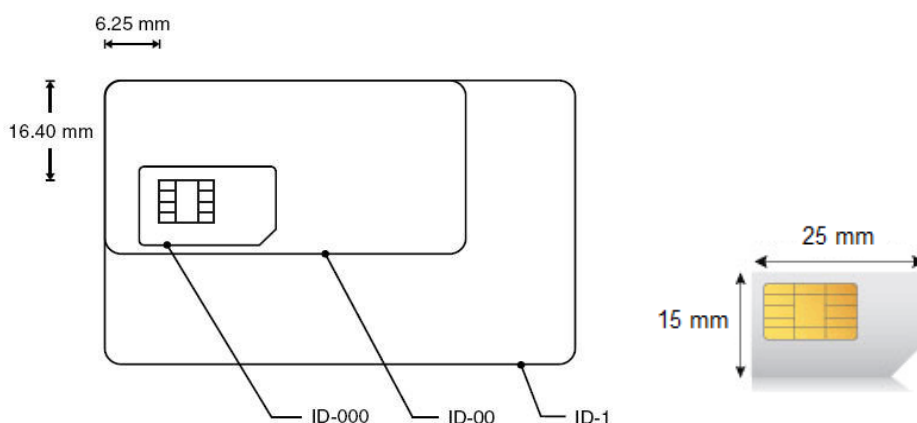
SLIKA 3.21: POTEK BRANJA IZ IJPP KARTICE

#### 3.7.2.1. Opis SAM modula

SAM modul (angl. Secure access module) je posebna izvedba pametne kartice, ki je namenjena izboljšanju varnosti in kriptografskih sposobnosti naprave, v kateri se uporablja. Uporablja se za izvajanje kriptografskih algoritmov, avtentikacije zunanjih naprav in varni shrambi podatkov (to so certifikati, varnostni ključi, ...).

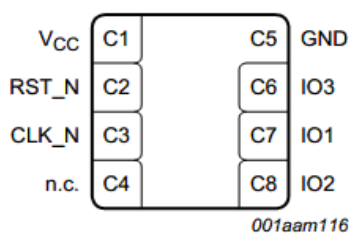
Velikost SAM modula določa standard ISO 7810. Uporabljen velikost je ID-000, ki jo poznamo kot velikost mini SIM kartice (Slika 3.22: Velikosti, ki jih definira standard ISO 7810 in velikost ID-000 (SIM))

#### DS 4: Integracija IS sistemov prevoznika



SLIKA 3.22: VELIKOSTI, KI JIH DEFINIRA STANDARD ISO 7810 IN VELIKOST ID-000 (SIM)

Za komunikacijo z zunanjimi napravami in zagotovitev napajanja so uporabljeni kontakti, definirani s standardom ISO 7816. Slika 3.23 prikazuje razporeditev kontaktov.



SLIKA 3.23: POZICIJA KONTAKTOV, KI JIH DEFINIRA STANDARD ISO 7810

Pozicija kontaktov, ki jih definira standard ISO 7810 (Slika 3.23):

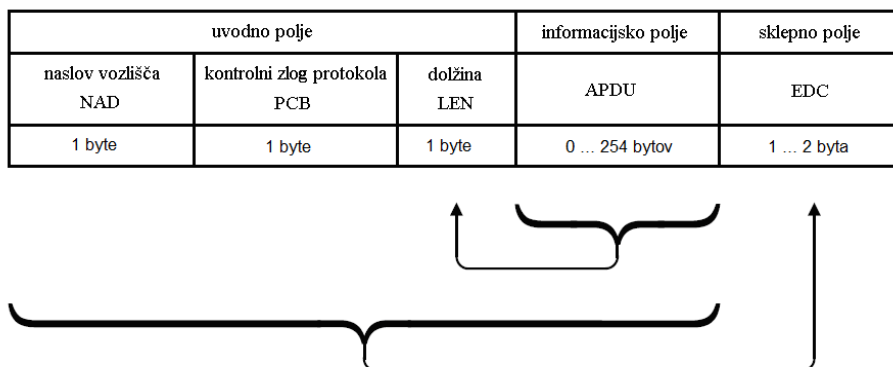
- C1 – napajalna napetost,
- C2 – reset,
- C3 – ura,
- C4 – rezervirano,
- C5 - masa,
- C6 – Pomožna vhodno/izhodna linija,
- C7 – vhodno/izhodna linija,
- C8 – Pomožna vhodno/izhodna linija

S standardom ISO 7816 so definirane še preostale fizične, električne in logične karakteristike pametnih kartic. Prav tako je definiran komunikacijski protokol med čitalnikom in SAM modulom, s katerim mora biti skladen čitalnik SAM modula.

Uporabljen prenosni protokol, T=1, je asinhroni, izmenično enosmerni, blokovno naravnani, kar pomeni, da je najmanjša prenesena podatkovna enota en blok. Struktura blokovne enote je sestavljena iz uvodnega, informacijskega in končnega sklepne polja. Uvodno polje je obvezno, prav tako sklepno,

#### DS 4: Integracija IS sistemov prevoznika

medtem ko je informacijsko polje neobvezno. Takšna struktura omogoča zanesljivo in varno sporočane. Slika 3.24 prikazuje strukturo prenosnega bloka.



SLIKA 3.24: STRUKTURA PRENOSNEGA BLOKA PROTOKOLA T=1

TABELA 3.1: OPIS SESTAVNIH DELOV PROTOKOLA T=1

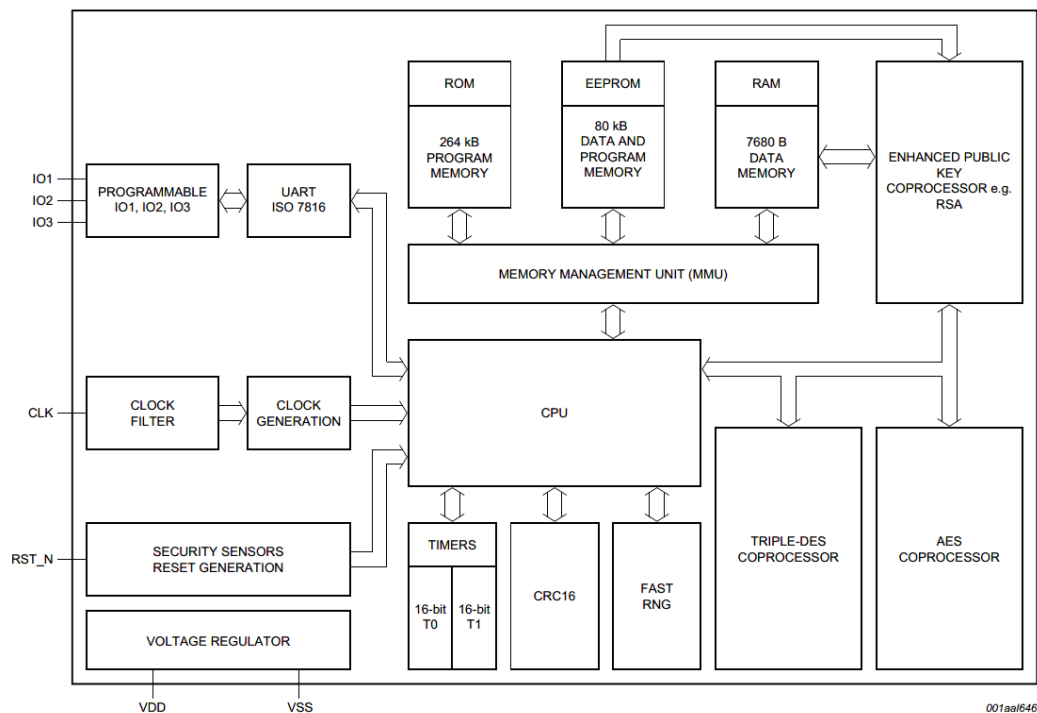
Oznaka	Velikost	Opis
NAD	1 byte	Naslov vozlišča se uporablja za shranjevanje naslova vira in cilja
PCB	1 byte	Kontrolni zlog protokola je namenjen kontroli in nadzoru prenosnega protokola. Prvotno je namenjen kodiranju tipa bloka in pripadajočih pomožnih podatkov
LEN	1 byte	Polje dolžine podaja dolžino informacijskega polja v heksadecimalni vrednosti.
APDU	0 - 254 bytov	Podatki oz. aplikacijski ukaz
EDC	1 – 2 byta	Sklepno polje vsebuje kodo za odkrivanje napak, ki je izračunana iz predhodnih podatkov

Notranja zgradba SAM modula bazirana na uporabi preizkušenega in prilagojenega mikroprocesorskega jedra. V pomoč izvajanju kriptografskih algoritmov so dodani pomožni procesorji za posamezni tip algoritma, generator naključnih števil in pomožna logika.

Slika 3.25 prikazuje običajno zgradbo SAM modula.



## DS 4: Integracija IS sistemov prevoznika



SLIKA 3.25: BLOKOVNA SHEMA SAM MODULA

Za varno hranjenje podatkov so uporabljene različne metode:

- avtentikacija za dostop do podatkov,
- logike za upravljanje dostopa do pomnilnika (MMU),
- logike za detekcijo vnosa napak,
- enkripcija shranjenih podatkov,
- vnos naključnih zakasnitev,
- zapletena strojna arhitektura,
- senzorji za detekcijo vdora, ...

Funkcije, ki jih lahko opravlja SAM modul:

- generiranje ključev za aplikacije iz glavnega ključa,
- varno hranjenje ključev,
- izvajanje kriptografskih algoritmov,
- varno izvajanje kriptografije,
- izvajanje avtentikacije,
- generiranje ključev seje,
- izvajanje varnega sporočanja, ...

### 3.7.3. SPECIFIKACIJA VARNOSTNEGA ELEMENTA

V nadaljevanju sledijo lastnosti varnostnega elementa za uporabo v sistemu IJPP:

- procesor: 16-bitni ali več,
- pomnilnik: 40 kB ali več,
- komunikacijski protokol T=1,
- hitrost komunikacije 115200 b/s,
- podpora kriptografije DES, 3DES in 3TDES, AES,
- diverzifikacija ključev 3TDES,
- podpora HASH funkcij (SHA-1, SHA-224 in SHA 256),
- računanje CRC32,
- shranjevanje vsaj 16 simetričnih ključev in 2 asimetričnih parov,
- pravi generator naključnega števila,
- generiranje ključev seje,
- podpora vsaj 2 logičnih kanalov,
- avtentikacijski postopek Desfire 3TDES.

#### 3.7.3.1. Ključi na varnostnem elementu

Ključi na Sam modulu:

- $K_{SAK\_M}$  - master ključ varnostnega elementa
- $K_{SAK}$  - ključ, ki omogoča avtentikacijo med terminalsko opremo in varnostnim elementom (nameščen ob namestitvi in aktivaciji varnostnega elementa).
- $K_{NFC\_R}$  - Izvorni 3TDEA ključ za Mifare DesFire kartice IJPP z opcijo pisanja ključev na DesFire kartico. To je ključ, ki se uporabi za avtentikacijo z brezstično kartico kot tudi za zaklepanje aplikacije na brezstični kartici z njenim diverzificiranim ključem.
- $K_{NFC\_I}$  - Privzeti ključ nove IJPP brezstične aplikacije, ki mora biti zapisan v aplikaciji pred menjavo z diverzificiranim ključem izvedenim iz  $K_{NFC\_R}$

Oznaka	Tip	Pozicija SAM	Velikost [bit]	Veljavnost	Opomba
$K_{SAM\_M}$	AES	00	192	Do preklica	Master ključ
$K_{SAM\_A}$	AES	01	128	Do preklica	Ključ za avtentikacijo
$K_{NFC\_R}$	3TDES	02	3 x 56	Do preklica	Ključ je uporabljen za dodajanje novega ključa na brezstično kartico
$K_{NFC\_I}$	3TDES	03	3 x 56	Do preklica	Privzeti ključ

### 3.7.4. ŽIVLJENJSKI CIKEL VARNOSTNEGA ELEMENTA

Upravljaivec sistema IJPP določi vrsto varnostnega elementa, za katerega mora skrbeti skozi celotni življenjski cikel.

Glede na izbiro varnostnega elementa, se lahko življenjski cikel, v manjši meri razlikuje od opisanega, ki kot varnostni element uporablja SAM modul.

#### 3.7.4.1. Dobava

Proizvajalec oz. lastnik varnostnega elementa (v kolikor ni to upravljaivec) dostavi upravljavcu ključke za inicializacijo. S tem pridobi upravljaivec dostop do varnostnega elementa.

#### 3.7.4.2. Personalizacija

Personalizacijo izvede upravljaivec. Namesti potrebne ključke in certifikate v varnostni element in ga pripravi za distribucijo. Po končani personalizaciji je varnostni element pripravljen za uporabo v validirani terminalski opremi IJPP.

Postopek:

- Avtentikacija s ključem za inicializacijo.
- Vpis ključev in certifikatov IJPP.
- Vpis varnostnega elementa v bazo zalednega sistema.

#### 3.7.4.3. Distribucija

Postopek distribucije varnostnega elementa mora opraviti pooblašena oseba. Vsak varnostni element ima lasten ključ, ki ga, validirana terminalska oprema IJPP pridobi v postopku registracije terminala.

Postopek:

- Lastniku terminalske opreme se dodeli varnostni element.
- Pooblašena oseba dostavi varnostni element.

#### 3.7.4.4. Namestitev in aktivacija

Fizična namestitev varnostnega elementa (v kolikor je potrebna) zahteva posredovanje pooblaščenega uporabnika in lastnika terminalske opreme. Po namestitvi se izvede postopek aktivacije, ki omogoči uporabo varnostnega elementa in terminalske opreme IJPP.

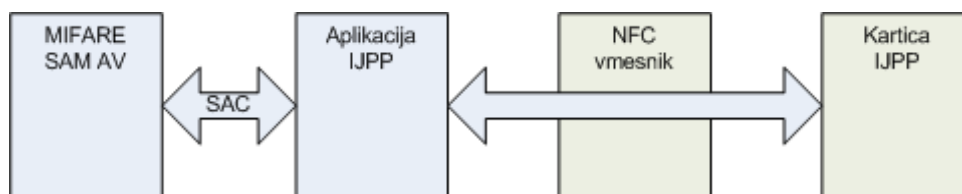
Postopek:

## DS 4: Integracija IS sistemov prevoznika

- Fizična namestitev varnostnega elementa v terminal.
- Terminal, ki je predhodno že registriran v zalednem sistemu IJPP (glej poglavje »Registracija terminala«), sproži postopek za pridobivanje avtentikacijskega ključa za SAM modul.
- Vpis varnostnega elementa in terminalske opreme v zaledni sistem.
- Zaledni sistem dodeli terminalske opremi avtentikacijski ključ za pripradajoč SAM modul.

### 3.7.4.5. Delovanje

Terminalska oprema se ob zagonu avtentificira z varnostnim elementom, kar omogoča uporabo kriptografskih funkcij in delovanje IJPP aplikacije. Hkrati se vzpostavi varni komunikacijski kanal (SAC), ki omogoča varno sporočanje med varnostnim elementom in IJPP aplikacijo (na terminalske opremi).



SLIKA 3.26: VARNI KANAL SAC MED VARNOSTNIM ELEMENTOM IN APLIKACIJO IJPP

Postopek:

1. Terminalska oprema se uspešno avtentificira z varnostnim elementom.
2. Ob uporabi IJPP kartice se v avtentikacijski postopek vključi SAM modul.

Postopek delovanja je opisan v prejšnjih poglavjih.

Med delovanjem lahko pride do zahteve za zamenjavo terminalske opreme, ki je posledica okvare ali odtujitve.

**Ob okvari je potrebno:**

- Začasno deaktivirati staro terminalske opreme in varnostni element v zalednem sistemu.
- Namestiti novo terminalske opreme in varnostni element.
- Izvesti postopek dodelitve in namestitve in aktivacije.

**Ob odtujitvi je potrebno:**

- Trajno blokirati odtujeno terminalske opreme in varnostni element v zalednem sistemu.
- Namestiti novo terminalske opreme in varnostni element.
- Izvesti postopek dodelitve in namestitve in aktivacije.

#### 3.7.4.6. Deaktivacija / blokada

Deaktivacija terminalske opreme in varnostnega elementa omogoča začasno izključitev iz sistema IJPP. Izvede se v zalednem sistemu IJPP.

Za blokadado varnostnega elementa se v zalednem sistemu izvede vpis v listo blokiranih varnostnih elementov.

#### 3.7.4.7. Uničenje

Varnostni element je potrebno po prenehanju uporabe vrniti izdajatelju, ki ga uniči ali pripravi za ponovno uporabo. Hkrati se izvedejo ustrezni postopki v zalednih sistemih.

### 3.8. PROTOKOL DELA

#### 3.8.1. *PODATKI, KI SE PROCESIRAJO PRI BRANJU INFO KARTICE*

Pri postopku branje INFO kartice se preberejo vse informacije o kartici in o vozovnicah na kartici. Te informacije so ključne v primeru, ko želimo s kartico opravljati nadaljnje postopke, kot so validacija ali nakup vozovnic. Pri postopku branje INFO se s kartice preberejo naslednje datoteke INFO, CARD INFO, DATA INDEX, DATA FILE, CHECK, PRODUCTS, RELATIONS in HISTORY-VALID.

##### 3.8.1.1. Podatki, ki se preverjajo na kartici ob branju INFO kartice

Pri posameznih prebranih datotekah, čitalnik preveri ustreznost podatkov. Pri postopku branje INFO kartice se preberejo naslednje datoteke:

- standardna podatkovna datoteka INFO,

Posamezni parametri v datoteki INFO predstavljajo numerične identifikatorje verzij datotek na kartic, velikosti enega bajta.

- Čitalnik mora preveriti parametre VERSION\_INFO, VERSION\_CARD\_INFO, VERSION\_DATA\_INDEX, VERSION\_DATA\_FILE, VERSION\_CHECK, VERSION\_PRODUCTS, VERSION\_HISTORY\_VALIDATION, VERSION\_RELATIONS, če ustrezajo določenim verzijam. Vrednosti vseh verzij datotek morajo biti 0x01. Vrednosti RFU (rezervirano za morebitno prihodnjo uporabo) pa so 0x00.
- podatkovna datoteka z varnostno kopijo CARD INFO,

#### DS 4: Integracija IS sistemov prevoznika

V datoteki CARD INFO se nahajajo naslednji parametri: STATUS, DATUM PRIČETKA VELJAVE KARTICE, DATUM KONCA VELJAVE KARTICE, TIP PERSONIFIKACIJE, POSEBNI STATUS, IME ter NASLOV, EMŠO, DAVČNA ŠTEVILKA, DATUM INICIALIZACIJE KARTICE, DATUM PERSONIFIKACIJE KARTICE.

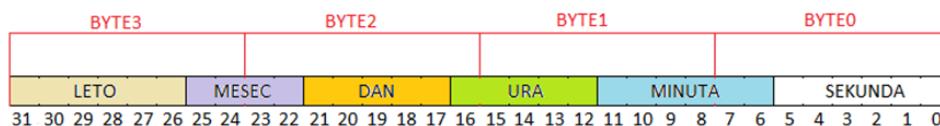
- Če je STATUS Blokirana, Neaktivna ali Onemogočena se postopek nadaljnega branja prekine. Terminal javi ustrezno napako, da je kartica Blokirana, Neaktivna ali Onemogočena. Veljaven STATUS za nadaljevanje postopka je Aktivna 0x02.
- Preveri se, če se kartica nahaja na črni ali izvršni listi.

Če se **kartica** nahaja na črni listi in jo je potrebno blokirati, je potrebno v STATUS **kartice** (datoteka CARD INFO) zapisati Blokirana (0x04). Tukaj se postopek konča, akcija se poroča v zaledni sistem.

Če se **voznica** nahaja na izvršni listi in jo je potrebno začasno blokirati, je potrebno v STATUS **voznice** (datoteka PRODUCTS) zapisati Začasno blokirana (0x08).

Če se **voznica** nahaja na izvršni listi in jo je potrebno izbrisati, je potrebno v STATUS **voznice** zapisati Neveljavna (0x04).

- Preveri se veljavnost kartice s preverjanjem ustreznosti vrednosti parametrov DATUM PRIČETKA in DATUM KONCA VELJAVNOSTI KARTICE. Vsi datumi se na kartico zapišejo v obliki 4 bajtnega števila, ki sestavlja informacije o datumu kot prikazuje spodnja slika.



- Preveri se, če POSEBNI STATUS ustreza vrednosti IJPP (0x00).
- podatkovna datoteka z varnostno kopijo DATA INDEX

Iz datoteke DATA INDEX se preberejo indeksi oz. mesta v datoteki DATA FILE, kjer se nahajajo zapisi posameznih statusov na kartici. Če na kartici ni zapisanega nobenega indeksa, se iz tarife prebere in nastavi privzeti status. S tem se preskoči branje DATA FILE, saj v datoteki ni podatkov. Če pri branju tarife pride do napake, prekinemo postopek branje INFO in uporabnika obvestimo o napaki.

- podatkovna datoteka z varnostno kopijo DATA FILE

Iz določenih mest datoteke DATA FILE, ki jih definira datoteka DATA INDEX, se preberejo statusi, ki so zapisani na kartici. Pri tem je pomembno, da izberemo status, ki je veljaven in ima morebiti prednost pred ostalimi. Posamezen status je zapisan z določenim formatom (FORMAT 0x0, FORMAT 0x1, ..). Če nobeden status ni veljaven, se iz tarife prebere in nastavi

## DS 4: Integracija IS sistemov prevoznika

---

privzeti status. Če pri branju tarife pride do napake, prekinemo postopek branje INFO in uporabnika obvestimo o napaki.

- podatkovna datoteka z varnostno kopijo CHECK

Iz datoteke se preberejo informacije povezane z zadnjo opravljeno transakcijo.

- podatkovna datoteka z varnostno kopijo PRODUCTS

Iz datoteke se preberejo informacije o vozovnicah na kartici. Branje datoteke PRODUCTS je lahko izvedeno postopoma. Najprej se lahko preberejo statusi vseh potencialnih vozovnic, nato se lahko preberejo še vozovnice. Vozovnice so v datoteki zapisane za statusi druga za drugo (posamezna vozovnica je zapisana na mestu  $10+(n-1)*54$ , kjer je n zaporedna številka statusa).

- podatkovna datoteka z varnostno kopijo RELATIONS.

Iz datoteke se preberejo informacije o relacijah, ki so zapisane na kartico.

- Ciklična datoteka HISTORY-VALID

Iz datoteke se preberejo informacije o zadnji validaciji vozovnice.

Branje posameznih datotek poteka v zaporedju. Ko se posamezna datoteka prebere in se informacije shranijo v ustrezno podatkovno strukturo na čitalniku, se na čitalniku preveri ustreznost posameznih podatkov.

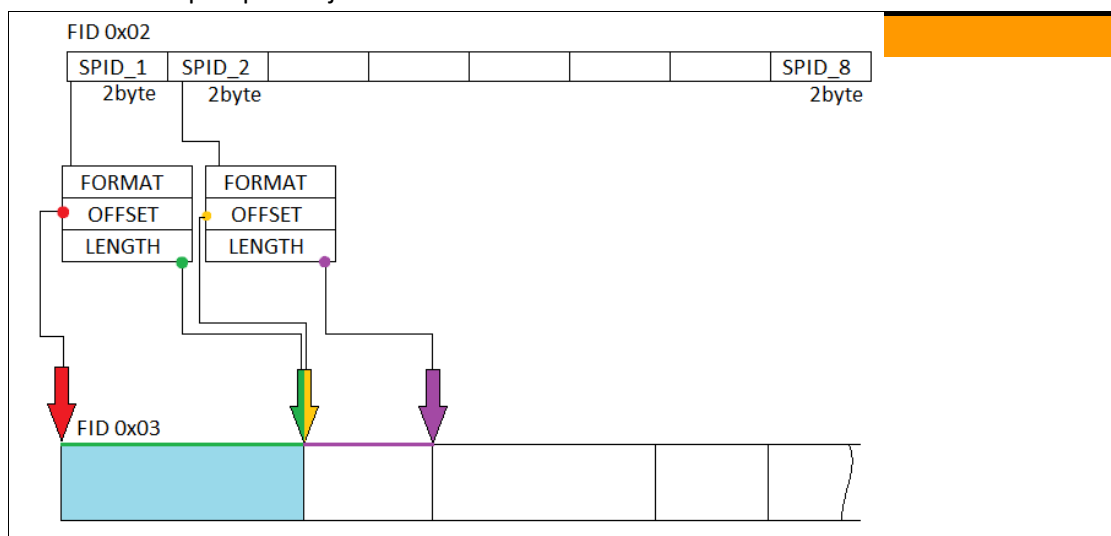
### 3.8.2. PODATKI, KI SE PROCESIRAJO PRI NAKUPU VOZOVNICE

Pred vsakim nakupom vozovnice je potrebno zagotoviti, da se na prodajnem mestu nahaja pravilno inicializirana kartica katero določa struktura kartice IJPP. Predpostavimo, da takšno kartico imamo. Med prvim in vsakim naslednjim nakupom vozovnice je razlika v tem, da pred prvim nakupom vozovnice moramo ustrezno zapisati vsebino podatkov v FID 0x02, FID 0x03 in FID 0x0A (FID 0x0A v primeru medkrajevnih vozovnic). Pri prvem nakupu v FID 0x02 zapišemo podatke o ponudniku storitve in obliki formata vsebine. V odvisnosti od formata vsebine, določimo tudi odmik in dolžino zapisa. V FID 0x03 v nadaljevanju zapišemo vsebino, ki je definirana s formatom v FID 0x02. Postopek zapisa zajema:

1. iskanje prostega mesta za zapis SPID. Prosto mesto je označeno z vsebino 0x00 »nič«
2. izračun odmik in dolžine (določimo na katero mesto se podatki zapišejo, pazimo, da ne prepišemo obstoječih)

## DS 4: Integracija IS sistemov prevoznika

Primer zapisa prikazuje slika:



SLIKA 3.27: PRIMER POVEZAVE MED FID 0x02 IN FID 0x03

### Primer zapisa FID 0x02 in FID 0x03:

SPID\_1 = 12

FORMAT = 0

OFFSET = 0

LENGTH = 10

Format0:

ID Statusa = 1

Datum pričetka veljave = 1.9.2016

Datum konca veljave = 30.6.2017

SPID\_2 = 12

FORMAT = 1

OFFSET = 10

LENGTH = 30

Format1:

Šifra vloge = 2015A0067254

Datum pričetka veljave = 1.9.2016

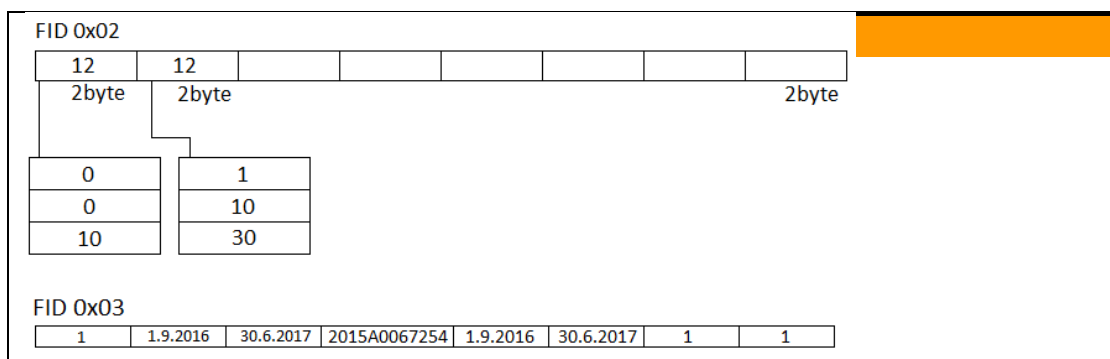
Datum konca veljave = 30.6.2017

Vrsta medkrajevne vozovnice = 1 (M)

Status potnika = 1



#### DS 4: Integracija IS sistemov prevoznika



Če želimo ob prvem nakupu prodati medkrajevno vozovnico moramo najprej zapisati podatke o relaciji za katero želimo vozovnico prodati. Na kartico lahko zapišemo 5 različnih relacij. Postopek zajema iskanje prostega mesta in zapis podatkov. Prosto mesto je označeno s statusom 0x00 »nič« ali s statusom 0x04 »neveljavna«.

	status	status			status	
	Podatki relacije	Podatki relacije			Podatki relacije	

#### Primer zapisa relacije:

STATUS = 1

FORMAT IDENTIFIER = 1

LINE DEFINITION = 1

LINE ID = 1234

VALID FROM = 1.9.2016

VALID TO = 30.6.2017

DISTANCE = 12000 = 120km (!!glej dokument strukture kartice!!)

START STATION NAME = Maribor (max 10 znakov, če je več se porežejo)

STOP STATION NAME = Ljubljana (max 10 znakov, če je več se porežejo)

NUMBER OF STATIONS IN RECORD = 5 (število postaj s katerimi je linija definirana)

BUS STATION ID1 = 1253

BUS STATION ID2 = 1277

BUS STATION ID3 = 1346

BUS STATION ID4 = 1378

BUS STATION ID5 = 1409

BUS STATION ID6... (vsi nadaljnji Idji so postavljeni na 0x00 »nič«)

ONE WAY = 0

SUBVENCIONIRANA = 1

RFU = 0 (vse neuporabljene byte postavimo na 0x00 »nič«.)

1	0x04	0	0	0
1	Podatki relacije			
1				
1				
1234				
1.9.2016				
30.6.2017				
12000				
Maribor				
Ljubljana				
5				
1253				
1277				
1346				
1378				
1409				
0				
0				
0				
0				
0				
0				
0				
0				
0				
0				
0				
0				
0				
0				
0				
1				
5617				
0				

ŠT. PREHODOV = 1

#### DS 4: Integracija IS sistemov prevoznika

ŠT. KART = 1

DATUM ZAČ. VELJ. = 1.9.2016 00:00:00 (pomembno, da je dodana ura!!)

DATUM KONCA VELJ. = 30.9.2016 23:59:59 (pomembno, da je dodana ura!!)

DATUM NAKUPA = 25.8.2016 12:34

VREDNOST OB NAKUPU = 2000 (= 20.00, 1750 = 17.50)

ID TARIFNEGA SISTEMA = 2367

NAČIN TARIFNEGA SISTEMA = 0x01

LINIJA = 123

VSTOPNA CONA = 0

FID0x0A Index = 0 (na tem mestu se v FID 0x0A nahajajo podatki relacije)

TRENTNI MESEC = 11 (glej strukturo kartice)

ŠTEVILO KUPONOV = 0

ID SKUPINE PRODUKTOV = 0

RFU = 0

1	0x04		0
1	podatki		
12	produkta		
2334123			
14			
930		.	.
1		.	.
1		.	.
1.9.2016 00:00:00		.	.
30.9.2016 23:59:59		.	.
25.8.2016 12:34		.	.
2000			
2367			
1			
123			
0			
0			
11			
0			
0			
0			

Pri vsakem naslednjem nakupu ni potrebno urejati FID 0x02, FID 0x03 in FID 0x0A, razen v primerih, kjer je to potrebno. Ti primeri so lahko sprememba statusa potnika, sprememba veljave statusa, sprememba relacije, ali sprememba formata. V kolikor teh sprememb ni se ob vsakem naslednjem nakupu zapiše le FID 0x08. Postopek je tak da se najprej poišče prosto mesto in nato se na to prosto mesto zapišejo podatki.

#### 3.8.2.1. Nakup enkratne vozovnice

Nakup enkratne vozovnice je tehnično popolnoma enak nakupu terminskih vozovnic. Najprej se zapiše FID0x02, FID 0x03 in FID 0x0A. Razlika je v vsebini zapisa, FID 0x0A, kadar gre za nakup enkratne, medkrajevne vozovnice. V tem primeru v FID 0x0A zapišemo začasno linijo. STATUS v FID 0x0A postavimo na vrednost 0x02 »Valid-Temporary«

### 3.8.2.2. Nakup enosmerne vozovnice

Nakup enosmerne vozovnice je tehnično popolnoma enak nakupu povratnih vozovnic.

Najprej se zapiše FID0x02, FID 0x03 in FID 0x0A. Razlika je v vsebini zapisa, FID 0x0A, kadar gre za nakup enosmerne, medkrajeve vozovnice. V tem primeru v FID 0x0A postavimo parameter ONE-WAY = 1, kar označuje, da gre za enosmerno vozovnico.

### 3.8.2.3. Prodajna mesta IJPP

Vozovnice IJPP bo možno kupiti na vseh IJPP prodajnih mestih. Prodajno mesto IJPP je definirano kot točka na kateri je nameščena strojna in programska oprema, s pomočjo katere je možno urejati brezstične kartice. Z urejanjem brezstične kartice so označeni vsi postopki ki omogočajo zapis in manipulacijo podatkov, določenih s strukturo kartice IJPP.

### 3.8.2.4. Inicializacija kartice

Ob inicializaciji kartice se na brezstično kartico zapiše struktura IJPP podatkovnega modela. Ta struktura je predstavljena v podnaslovu **3.5.2**. Na kartici se kreira aplikacija in posamezni vsebinski odseki označeni od FID 0x00 do FID 0x0A.

#### FID 0x00

Vsebina namenjena osnovnim informacijam. Podatki od VERSION\_INFO pa vse do VERSION\_RELATIONS se postavijo na vrednost 1. Vse vrednosti RFU se postavijo na 0.

#### FID 0x01

Vsebina namenjena informacijam o kartici. Podatki se postavijo na naslednje vrednosti:  
STATUS = 2 (aktivna)

DATUM\_PRIČETKA\_VELJAVE\_KARTICE = trenutek od kdaj je kartica veljavna (1.1.2016 00:00:00) !pomembno da se zraven datuma zapiše še ura!

DATUM\_KONCA\_VELJAVE\_KARTICE = trenutek do kdaj je kartica veljavna (31.12.2030 23:59:59) !pomembno, da se zraven datuma zapiše še ura!

TIP = 0x04 (v primeru, da gre za imensko kartico, za kartico, ki je namenjena točno določenemu potniku)

TIP = 0x01 (v primeru, da gre za neimensko kartico. Ni vezana na točno določenega potnika)

POSEBNI STATUS = 0x00

IME,PRIIMEK,NASLOV = »ime:priimek:naslov:postna\_st:kraj« !(če ni imenska kartica se zapiše »::::« dvopičje služi kot separator med posameznimi deli v imenskem sklopu)!

## DS 4: Integracija IS sistemov prevoznika

---

EMŠO = 1905982500385 (EMŠO potnika. Tukaj je treba paziti kako je številka predstavljena na kartici!! Vsak byte predstavlja eno od števk v EMŠO)

DAVČNA ŠTEVILKA = 0x0000 (inicializacija na 0)

DATUM INICIALIZACIJE KARTICE = 1.9.2016 12:14:52

DATUM PERSONIFIKACIJE KARTICE = 1.9.2016 12:16:37

RFU = 0 (postavimo vse byte RFU na 0)

### **FID 0x02**

Vsi SPID = 0

Vsi podatki inicializirani z 0

### **FID 0x03**

Vsi podatki znotraj FID0x03 inicializirani z 0

### **FID 0x04**

BALANCE inicializiran na 0

### **FID 0x05**

Vsi podatki inicializirani z 0

### **FID 0x06**

Vsi podatki inicializirani z 0

### **FID 0x07**

Vsi podatki inicializirani z 0

### **FID 0x08**

Vsi statusi inicializirani z 0x04 (neveljavna)

Vsi podatki inicializirani z 0

### **FID 0x09**

Vsi podatki inicializirani z 0

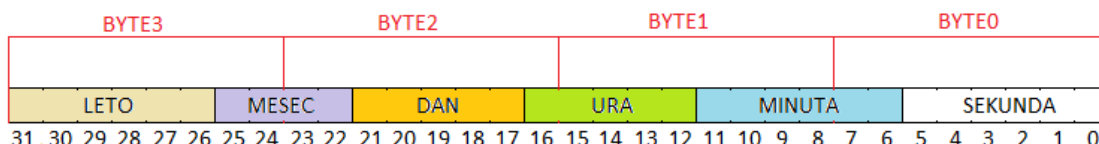
### **FID 0x0A**

Vsi podatki inicializirani z 0

### **3.8.2.5. Postopek za zapis datuma**

**!POMEMBNO!** Postopek za zapis datuma na kartico! Datumi na kartici so predstavljeni s 4 zlogi na sledeč način:

## DS 4: Integracija IS sistemov prevoznika

$$\begin{aligned} \text{DATUM} = & ((\text{LETO\_ZDAJ} - 2002) \ll 26) + \\ & (\text{MESEC\_ZDAJ} \ll 22) + \\ & (\text{DAN\_ZDAJ} \ll 17) + \\ & (\text{URA\_ZDAJ} \ll 12) + \\ & (\text{MINUTA\_ZDAJ} \ll 6) + \\ & (\text{SEKUNDA\_ZDAJ}) \end{aligned}$$


P

**Primer zapisa datuma 13.5.2014 6:52:46**  
 = 00110001010110100110110100101110 (bin)  
 = 315A6D2E (hex)  
 = 828009774 (dec)

### 3.8.2.6. Podatki na kartici, ki so potrebni za nakup vozovnic

Pred nakupom vozovnic morajo biti na kartici in v centru zapisani ustrezni podatki o potniku, statusih, vlogah za subvencije in relacijah. Podatki o potniku so potrebni za nakup vozovnic, pri katerih je zahtevana identifikacija potnika. Statusi potnika uvrščajo v določene skupine, kot so šolarji, upokojenci, delavci, splošni potniki, ipd. Podatki o vlogah za subvencije omogočajo potnikom nakup subvencioniranih vozovnic. Relacijski podatki določajo stalne relacije na katerih se potniki vozijo v šolo, službo, ipd. Ti podatki se na kartici zapišejo pod: FID 01x01, FID 0x02, FID 0x03, FID 0x0A in FID 0x08.

**Na center se pošljejo naslednji podatki:**

1. Podatki o potniku
  - struktura: Passenger
  - metoda: SetPassenger
2. Podatki o kartici
  - struktura: Card
  - metoda: SetCard
3. Podatki o statusih potnika
  - struktura: CardPassengerStatus
  - metoda: UpdateCardPassengerStatus
4. Podatki o subvencijah potnika
  - struktura: SubsidyApplication

DS 4: Integracija IS sistemov prevoznika

---

- metoda: UpdateCardPassengerStatus

4. Podatki o relacijah

- struktura: IJPPTtransaction

- metoda: ProcessTransactions

**Ko imamo na kartici zapisane vse potrebne podatke o potniku, statusih, relacijah in vlogah, lahko izvedemo nakupe vozovnic, kjer se na center pošljejo naslednji podatki:**

5. Podatki o produktih

- struktura: IJPPTtransaction

- metoda: ProcessTransactions

### 3.8.3. PODATKI, KI SE PROCESIRAJO PRI VALIDACIJI/AKTIVACIJI VOZOVNICE

Podatki, ki se procesirajo pri validaciji oz. aktivaciji se nekoliko razlikujejo. V primeru validacije se na kartici nahaja vozovnica, ki je časovno veljavna in ustreza trenutni coni oz. liniji. V takšnem primeru je potrebno vnesti nove podatke o validaciji v datoteko HISTORY-VALID (FID 9) ter informacije o transakciji v datoteko CHECK (FID 7). V primeru aktivacije pa je zraven posodobitve datotek HISTORY-VALID (FID 9) in CHECK (FID 7) potrebno posodobiti še datoteko PRODUCTS (FID 8) in s tem informacije o vozovnici, ki jo je potrebno aktivirati.

#### 3.8.3.1. Podatki, ki se preverjajo na kartici ob validaciji/aktivaciji

Pri postopku validacija/aktivacija se najprej izvede postopek branje info. S tem se informacije o kartici in vozovnicah na njej shranijo v ustrezne podatkovne strukture. Pred nadaljevanjem postopka validacije ali aktivacije se pri branju info preveri ustreznost podatkov zapisanih v datotekah INFO, CARD INFO, DATA INDEX in DATA FILE.

Po postopku branje info se morebitne pretekle vozovnice na kartici pobrišejo, kar pomeni, da se posameznim pretečenim vozovnicam v datoteki PRODUCTS spremeni STATUS v Neveljavna (0x04). Tako ni potrebno brisati celotnega posameznega dela vsebine datoteke, ki definira informacije o pretekli vozovnici. Podobno se ravna v primeru začasnih (v datoteki RELATIONS začasne relacije označuje STATUS z vrednostjo 0x02), pretečenih relacij, kjer se relacije pobrišejo tako, da se STATUS posamezne relacije nastavi na Neveljavna (0x04)

Če se na kartici nahaja veljavna vozovnica, se nadaljuje postopek validacije. Če se na kartici nahaja veljavna vozovnica, ki jo je potrebno aktivirati, se nadaljuje postopek aktivacije. Veljavnost vozovnice se določi na podlagi časovne veljavnosti vozovnice ter na podlagi pravil conskega oz. linijskega prometa.

### 3.8.3.2. Podatki, ki se zapišejo ob validaciji

V nadaljevanju validacije se zapišejo informacije o validaciji v datoteko HISTORY-VALID. Informacije o kartični transakciji pa se zapišejo v datoteko CHECK. V datoteko HISTORY-VALID se zapišejo:

- ID TERMINALA,
- ČAS VALIDACIJE,
- ID POSTAJE,
- ID LINIJE,
- INDEKS VALIDIRANEGA PRODUKTA,
- SERIJSKA ŠTEVILKA VOZOVNICE ter
- ID VOZILA.

Primer zapisa podatkov v datoteko HISTORY-VALID:

- ID TERMINALA 12345  
dolžina 4 B: 0x00003039
- ČAS VALIDACIJE 17.11.2015 12:05:45  
dolžina 4 B: 0x36E2C16D
- ID POSTAJE 1234  
dolžina 4 B: 0x000004D2
- ID LINIJE 4321  
dolžina 4 B: 0x000010E1
- INDEKS VALIDIRANEGA PRODUKTA 9  
dolžina 2 B: 0x09
- SERIJSKA ŠTEVILKA VOZOVNICE 12345678910  
dolžina 6 B: 0x0002DFDC1C3E
- ID VOZILA 555  
dolžina 2 B: 0x022B



## DS 4: Integracija IS sistemov prevoznika

---

V datoteko CHECK pa se zapišejo:

- NAKLJUČNO ŠTEVILO TRANSAKCIJE,
- ČAS TRANSAKCIJE,
- ID TERMINALA,
- ŠTEVILO OPOMINOV.

Primer zapisa podatkov v datoteko CHECK:

- NAKLJUČNO ŠTEVILO TRANSAKCIJE 123456789  
dolžina 4 B: 0x075BCD15
- ČAS TRANSAKCIJE 17.11.2015 12:05:45  
dolžina 4 B: 0x36E2C16D
- ID TERMINALA 12345  
dolžina 4 B: 0x00003039
- ŠTEVILO OPOMINOV 99  
dolžina 1 B: 0x63

### 3.8.3.3. Podatki, ki se zapišejo ob aktivaciji

Če je vozovnico pred validacijo potrebno predhodno aktivirati, je pred nadaljevanjem postopka validacije potrebno zapisati ustrezne informacije v datoteko PRODUCTS. V datoteko PRODUCTS je potrebno vpisati:

- status vozovnice Aktivna (0x02),
- v primeru conskih vozovnic je potrebno vpisati VSTOPNO CONO (ID vstopne cone),
- v primeru linijskih vozovnic je potrebno vpisati LINIJO (ID linije),
- v primeru dnevne, večdnevne ali tedenske terminske vozovnice, ki začnejo veljati s časom aktivacije, pa je potrebno v datoteko PRODUCTS vpisati čas veljavnosti.

DATUM ZAČETKA VELJAVNOSTI se določi s trenutnim časom aktivacije. DATUM KONCA VELJAVNOSTI pa se izračuna na podlagi začetka veljavnosti in veljavnosti vozovnice.

#### DS 4: Integracija IS sistemov prevoznika

---

Zraven prilagoditve podatkov v datoteki PRODUCTS se po aktivaciji vpišejo še podatki o validaciji v datoteko HISTORY-VALID ter podatki o transakciji v datoteko CHECK.

Primer zapisa podatkov v datoteko PRODUCTS ob aktivaciji tri dnevne conske vozovnice, dne 17.11.2015 ob 12:05:45:

- STATUS3 2  
dolžina 1 B: 0x02
- DATUM ZAČETKA VELJAVNOSTI 17.11.2015 12:05:45  
dolžina 4 B: 0x36E2C16D
- DATUM KONCA VELJAVNOSTI 20.11.2015 12:05:45  
dolžina 4 B: 0x36E8C16D
- VSTOPNA CONA 33  
dolžina 2 B: 0x0021

#### **3.8.3.4. Podatki, ki se pošljejo proti centru ob validaciji/aktivaciji**

Ob uspešni validaciji/aktivaciji se na zaledni center pošljejo podatki o validaciji/aktivaciji z metodo spletnih storitev ProcessTransactions.

#### **3.8.4. PODATKI, KI SE PROCESIRAJO PRI STORNACIJI VOZOVNICE**

Storniramo lahko zadnji nakup in to na istem terminalu, kjer smo opravili nakup. Kot je razvidno iz postopka stornacije je bistveno, da se naključni števili shranjeni na terminalu in na kartici v datoteki CHECK ujemata. Na terminalu se vozovnica stornira tako, da se v datoteki PRODUCTS za posamezno vozovnico ustrezno spremeni status na Neveljavna (0x04).

Po uspešni spremembi statusa vozovnice, se v datoteko CHECK shranijo informacije o transakciji.

Ob uspešni stornaciji se na zaledni center pošljejo podatki z metodo spletnih storitev ProcessTransactions.

### *3.8.5. PODATKI, KI SE PROCESIRAJO PRI BLOKADI APLIKACIJE/VOZOVNICE*

Če se kartica nahaja na črni listi se pri postopku branje info, natančneje pri branju datoteke CARD INFO aktivira postopek blokade kartice. Pri blokadi kartice gre za spremembo parametra STATUS v datoteki CARD INFO.

Če se kartica nahaja na izvršni listi in dodatni parameter definira, da je potrebno blokirati določeno vozovnico na kartici, se pri postopku branje info oz. branju datoteke CARD INFO aktivira postopek blokade vozovnice. Bistvo postopka blokade vozovnice je sprememba statusa vozovnice v datoteki PRODUCTS v blokirana, če jo je možno kasneje deblokirati ali v neaktivna, če jo je potrebno trajno odstraniti.

Po uspešni spremembi statusa vozovnice ali kartice, se v datoteko CHECK shranijo informacije o transakciji.

Ob uspešni blokadi aplikacija ali vozovnice se na zaledni center pošljejo podatki z metodo spletnih storitev ProcessTransactions.

### *3.8.6. PODATKI, KI SE PROCESIRAJO PRI DEBLOKADI APLIKACIJE/VOZOVNICE*

Kartica se lahko deblokira le na ustreznem prodajnem mestu po plačilu ustrezne sankcije.

Če se kartica nahaja na izvršni listi in dodatni parameter definira, da je potrebno določeno vozovnico na kartici deblokirati, se pri postopku branje info oz. branju datoteke CARD INFO aktivira postopek deblokade vozovnice. Bistvo postopka deblokade vozovnice je sprememba statusa vozovnice v datoteki PRODUCTS iz blokirana v aktivna.

Po uspešni spremembi statusa vozovnice ali kartice, se v datoteko CHECK shranijo informacije o transakciji.

Ob uspešni deblokadi aplikacije ali vozovnice se na zaledni center pošljejo podatki z metodo spletnih storitev ProcessTransactions.

## **3.9. LISTNA VOZOVNICA KOT ALTERNATIVA BREZSTIČNI PAMETNI KARTICI**

V primeru, da potnik ne želi uporabljati brezstične kartice, lahko kot alternativo uporabi listno vozovnico. Lista vozovnica se izdaja le na vozilih in je opsijsko opremljena z AZTEC kodo. Trenutno se

## DS 4: Integracija IS sistemov prevoznika

listna vozovnica ne uporablja za prestopanje med vozili in je možno nabaviti le za relacijo, ki jo izvaja dano prevozno sredstvo. Implementacija AZTEC kode, pa omogoča tudi kasnejšo nadgradnjo sistema, ki bo na podlagi branja dane kode omogočala tudi validacijo, kar pa posledično omogoča prestopanje z listno vozovnico.

V primeru uporabe listnate vozovnice, na kateri je natisnjena AZTEC koda, mora biti ta zaščitena z AZTEC barkodo[6].

### 3.9.1. OPIS BARKODE

Bar koda se sestoji iz treh delov:

- Glava (Header)
- Odprti podatki (Open Data)
- Podpis (Seal)

Glava	Verzija
	Koda izdajatelja
	ID koda
	Koda tipa vozovnice
Odprti podatki	Število odraslih
	Število otrok
	Prvi dan veljavnosti
	Pod-Id koda
	...
	Informacijsko sporočilo
Kriptiran podpis	Kontrolna vsota

### 3.9.2. GENERIRANJE HASH KODE IN KRIPTIRANEGA PODPISA

Proces enkripcije:

- Nad vsemi podatki v polju glave in odprtih podatkov se izračuna hash po alhoritmu SHA-2 224.
- Rezultat iz prejšnje točke se kriptira z DSA 2048 (uporabljen privatni ključ).

Ta proces je definiran kot SHA-224, DSA, 2048. Podpis, ki ga dobimo je velik 56 bajtov. V primeru, da dobimo manjši podpis, so preostali biti postavljeni na 0.

Končni podpis je potrebno razdeliti v dve številki, ki jima sledijo ničle.

## DS 4: Integracija IS sistemov prevoznika

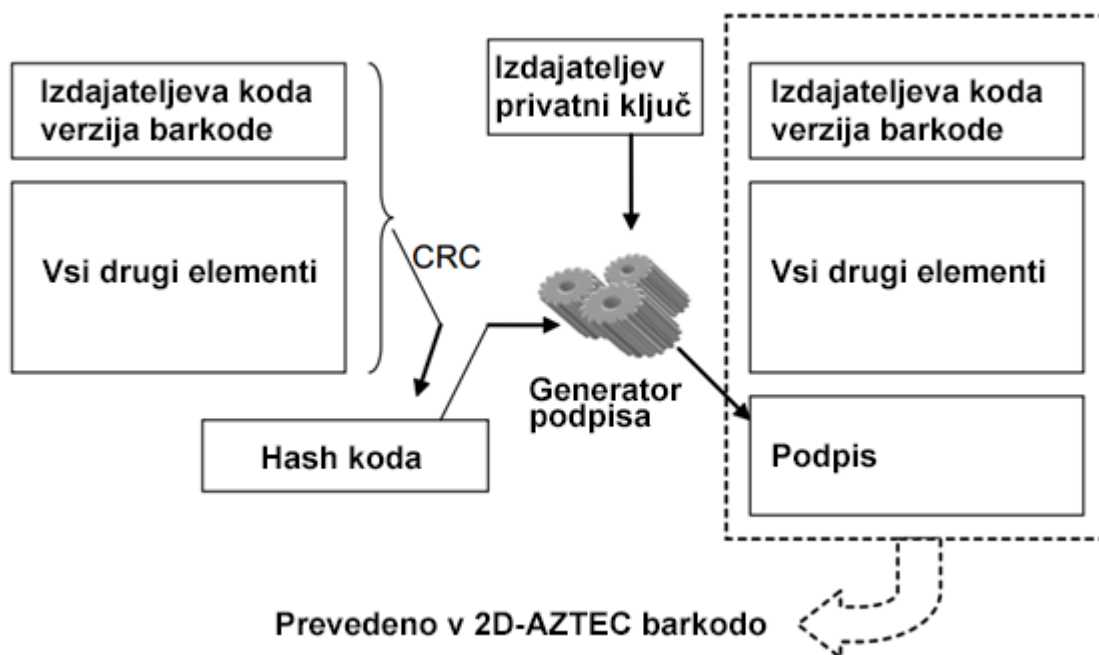
### 3.9.2.1. SHA 224

Funkcija SHA-224, ki je definirana v standardu NSA iz leta 2001, transformira vhodne podatke v podatek z določeno dolžino, ki ga imenujemo hash. Ta podatek je unikaten in se uporablja za preverjanje integritete podatkov nad katerimi je izračunan.

### 3.9.2.2. DSA 2048

Algoritem digitalnega podpisa (DSA) je definiran z ameriškim standardom FIPS za digitalne podpise.

Uporablja se par ključev: privatni in javni. Velikost ključev je 2048 bitov



SLIKA 3.28: SHEMATIČNI PRIKAZ GENERIRANJA PODPISA

### 3.9.3. KLJUČI DSA

Generiranje para ključev (privatni in javni ključ) DSA se mora opraviti vsakih 18 mesecev.

Za varno distribucijo in shranjevanje poskrbi upravljavec. Terminalska oprema prejme ključe preko zalednega sistema IJPP in jih mora varno shraniti.

### 3.9.4. AZTEC BARKODA

## DS 4: Integracija IS sistemov prevoznika

---

Podatki vozovnice so prevedeni v 6 slojno AZTEK kodo (simbol velikosti 41 x 41), ki teoretično nudi maksimalno 114 bajtov podatkov.

Barkoda meri 24 mm x 24 mm, kar predstavlja 41 x 41 pik.

Vsaka pika meri 0,59 mm x 0,59 mm.

Resolucija tiskanja je minimalno 150 dpi.

Stopnja popravljanja napak mora biti vsaj 23%, pri uporabi standardnega AZTEC mehanizma za popravilo napak.

### 3.10. REFERENCE

[1] – ISO/IEC Standard – ISO/IEC 14443 Identification cards – Contactless integrated circuit cards – Proximity cards.

[2] – ISO/IEC Standard – ISO/IEC 7810 Identification cards – Physical characteristics.

[3] – ISO/IEC Standard – ISO/IEC 7816 Identification cards – Integrated circuit.

[4] – GlobalPlatform – Card Specification, version 2.2.1

[5] – MF3ICDx21\_41\_81 MIFARE DESFire EV1 contactless multi-application IC Rev. 3.1 — 21 December 2010, 145631.

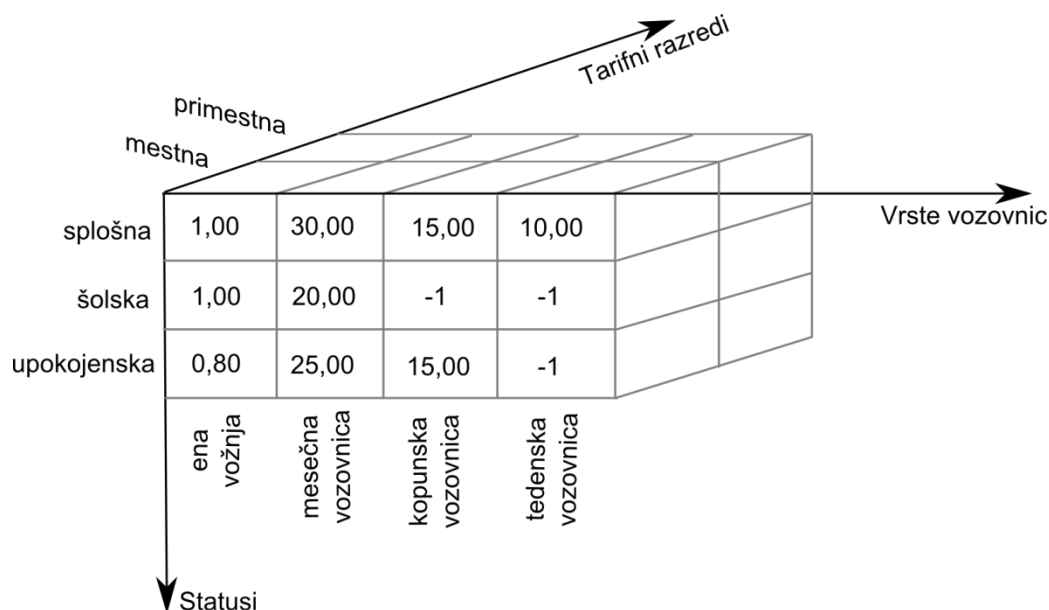
[6] – UICC CODE 918-2 – Layout for electronically issued transport documents on secured paper – Rail Combined Ticket 2 (RCT2) and Rail Credit Card Sized Ticket (RCCST), 7<sup>th</sup> edition, Marec 2014

## 4 ZGRADBA TARIFE

### 4.1 KONCEPTUALNI OPIS TARIFE

Tarifa je podatkovna struktura, ki določa pravila za prodajo in validacijo vozovnic v transakcijskem sistemu IJPP. Pravila za prodajo in validacijo vozovnic se nanašajo na vse igralce, ki sodelujejo v sistemu IJPP. To so končni uporabniki in podjetja, ki tržijo vozovnice IJPP. Podjetja lahko prevzamejo dve različni vlogi v sistemu IJPP to sta vlogi prevoznika in trgovca.

Za končne uporabnike je cena vozovnic najpomembnejši podatek iz tarife. V osnovi lahko tarifo opišemo s tridimenzionalnim podatkovnim poljem, ki določa cene. Prvo dimenzijo predstavlja vrsta vozovnic, drugo dimenzijo predstavljajo statusi in tretjo dimenzijo predstavljajo tarifni razredi oziroma cone. Na sliki (Slika 4.1) je prikazan primer takšnega podatkovnega polja s cenami vozovnic.



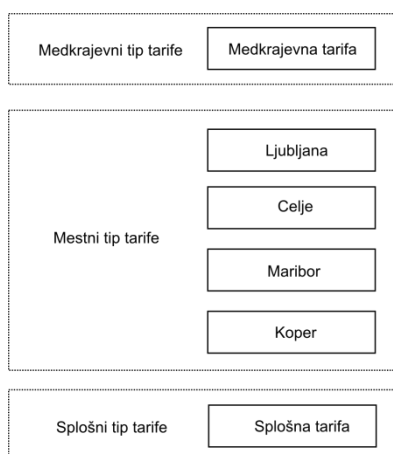
SLIKA 4.1: PRIMER PODATKOVNEGA POLJA S CENAMI

Iz primera je razvidno, da je na prodajnih mestih možno kupiti, uporabiti ali preveriti različne vrste predhodno definiranih vozovnic. Te vrste vozovnic so na primer ena vožnja, mesečna vozovnica, kuponska vozovnica, ipd. Vrste vozovnic so kombinirane s statusi oziroma pravicami uporabnikov. Statuse uporabljamo za kategorizacijo vozovnic glede na uporabnike z določenimi pravicami. Primeri statusov so: splošna, šolska, upokojenska, ipd., ki se nanašajo na kategorije uporabnikov, kot so splošni uporabniki, šolarji in upokojenci, ki imajo pravice do nakupov cenejših vozovnic. Vrste vozovnic in statuse kombiniramo s tarifnimi razredi oziroma območji vožnje, ki predstavljajo tretjo dimenzijo

#### DS 4: Integracija IS sistemov prevoznika

tarife. Tarifne razrede uporabimo za kategorizacijo vrste vozovnic in statusov glede na geografska področja, kjer se le ti lahko uporabljajo. Na primer vozovnice v mestnem tarifnem razredu imajo različne cene, kot vozovnice v medkrajevnem tarifnem razredu. Vsaka kombinacija vrste vozovnice, statusa in tarifnega razreda, ki je na voljo za prodajo končnemu uporabniku ima zabeleženo ceno v evrih. Kombinacije, ki nimajo zabeležene cene (oziroma imajo vrednost cene nastavljeno na -1), niso na voljo za prodajo. Na primer: iz tarife na sliki (Slika 4.1) lahko vidimo, da je mesečna vozovnica za upokoјence v mestnem tarifnem razredu na voljo za 25,00 evrov, medtem ko tedenske vozovnice za upokoјence ni možno kupiti. Podrobnejši opis za vrste vozovnic, statusov in tarifnih razredov sledi v poglavju o cenovnem modelu tarife.

Obstajajo trije tipi tarif: mestni tip, medkrajevni tip in splošni tip, kot je prikazano na sliki (Slika 4.2).



SLIKA 4.2: PRIKAZ RAZLIČNIH TIPOV TARIF

V sistemu IJPP imamo eno tarifo medkrajevnega tipa v kateri so definirani statusi, tarifni razredi, vrste vozovnic in cene, ki se uporabljajo v medkrajevnem prometu. Prav tako imamo eno tarifo splošnega tipa, ki vsebuje vse pripadajoče podatke za dodatne produkte IJPP. Za conske sisteme uporabljamo mestni tip tarife, kjer je za vsako lokacijo uporabe definirana ločena tarifa. V Poglavju Distribucija tarif je podrobneje opisana vloga in distribucija tarif v sistemu IJPP.

Tarife predstavljajo kombinacije cen, vrste vozovnic, statusov in tarifnih razredov, ki se s časom spreminjajo. Zato so tarife definirane le za določena časovna obdobja. Več aktivnih tarif lahko imamo znotraj različnih časovnih obdobj, vendar je v določenem trenutku lahko aktivna le ena tarifa iz katere določamo cene produktov. Zaradi tega moramo tarife unikatno identificirati. Tarifa, ki je trenutno aktivna se pošlje na terminale v naprej predpisani obliki. Podrobnejši opis strukture tarife, ki se pošlje na terminale je definiran v poglavju: Struktura tarife na terminalih.

Na izbranih prodajnih mestih lahko onemogočimo prodajo določenih vozovnic, medtem ko na preostalih prodajnih mestih, ki niso izbrana, prodajo dovolimo. To storimo z definiranjem pravil za tarife. S pravili za tarife eksplicitno določimo katere vrste vozovnic, statusi in tarifni razredi v tarifi so



## DS 4: Integracija IS sistemov prevoznika

na voljo za prodajo na določenih lokacijah. Podrobnejši opis delovanja pravil za tarife je predstavljen v poglavju: Pravila za tarife.

V poglavju Upravljanje s tarifami, so opisani posopki kreiranja, spreminjanja in posodobitev tarif.

### 4.2 CENOVNI MODEL TARIFE

To poglavje opisuje cenovni model tarife, ki ga predstavljajo različni parametri tarife, kot so: vrste vozovnic, statusi in tarifni razredi, ipd..., ki so podrobneje opisani v naslednjih podpoglavjih.

Osnovni podatki, ki določajo tarifo so:

- **TariffFormat** (int) – identifikator verzije tarife.
- **ServiceProviderId** (int) – unikatni identifikator ponudnika storitev.
- **ServiceProviderName** (string) – ime ponudnika storitev IJPP.
- **ServiceProviderTariff** (**ServiceProviderTariffData**) – podatki, ki definirajo tarife na lokacijah uporabe.

**TariffFormat** (int) – določa verzijo formata tarife. Trenutna verzija tarife je 1.

**ServiceProviderId** (int) – unikatni identifikator ponudnika storitev IJPP.

**ServiceProviderName** (string) – ime ponudnika storitev IJPP.

**ServiceProviderTariff** (**ServiceProviderTariffData**) – podatkovna struktura, ki določa pravila za prodajo vozovnic. **ServiceProviderTariff** se definira posebej za vsako lokacijo uporabe.

**TariffLocationZone** (**TariffLocationZoneData**) – podatkovna struktura, ki določa katerim conam pripada lokacija uporabe.

#### 4.2.1 OSNOVNI PODATKI TARIFE – STRUKTURA *TARIFFLOCATIONSUBLOCATION*

Lokacije uporabe v tarifi se lahko navezujejo na eno ali več con. Struktura **TariffLocationSubLocation** določa seznam preslikav lokacija uporabe – podrejene lokacije uporabe. Podatki ki določajo preslikave so:

- **LocationIdId** (int) – unikatni identifikator lokacije uporabe
- **SubLocationId** (int) – unikatni identifikator podrejene lokacije uporabe

**LocationId** unikatni identifikator lokacije uporabe

**SubLocationId** unikatni identifikator podrejene lokacije uporabe

#### 4.2.2 OSNOVNI PODATKI TARIFE – STRUKTURA SERVICEPROVIDERTARIFF

Tarifa za vsako lokacijo uporabe je določena s strukturo ServiceProviderTariff, katero opisujejo naslednji osnovni podatki:

- **TariffId** (int) – unikatni identifikator tarife
- **TariffLocationId** (int) – unikatni identifikator lokacije uporabe
- **TariffLocationName** (string) – ime lokacije uporabe
- **TariffType** (int) – tip tarife
- **StartTariffTime** (datetime) – začetni čas veljavnosti tarife
- **EndTariffTime** (datetime) – končni čas veljavnosti tarife
- **CurrencyCode** (integer) – valuta za plačilo (978 = EUR)
- **TariffProduct** (TariffProductData) – struktura opisuje lastnosti vrste vozovnic
- **TariffStatus** (TariffStatusData) – struktura opisuje lastnosti statusov vozovnic
- **TariffClass** (TariffClassData) – struktura opisuje lastnosti tarifnih razredov vozovnic
- **TariffPrice** (TariffPriceData) – struktura opisuje cene vozovnic

**TariffId** določa unikatni identifikator tarife.

**TariffLocationId** unikatni identifikator lokacije uporabe.

**TariffLocationName** določa ime lokacije uporabe tarife. Na primer: Ljubljana, Maribor, Celje...

**TariffType** predstavlja tip tarife (0=mestna, 1=medkrajevna, 2=splošna,...).

**StartTariffTime** določa točen čas pričetka veljave tarife. Čas je določen do sekunde natančno. Tarifa se lahko uporablja le v času, ki je večji ali enak StartTariffTime.

**EndTariffTime** določa točen čas konca veljave tarife. Čas je določen do sekunde natančno. Tarifa se lahko uporablja le v času, ki je manjši od EndTariffTime.

**CurrencyCode** določa valuto v kateri so navedeni zneski za plačilo (978 = EUR).

**TariffProduct** je podatkovna struktura, ki določa lastnosti vrste vozovnic.

**TariffStatus** je podatkovna struktura, ki določa lastnosti statusov vozovnic.

**TariffClass** je podatkovna struktura, ki določa lastnosti tarifnih razredov.

**TariffPrice** je podatkovna struktura, ki določa cene vozovnic.

#### 4.2.3 VRSTE VOZOVNIC – STRUKTURA TARIFFPRODUCT

Vrsta vozovnice je določena s podatkovno strukturo, katero predstavljajo naslednji parametri:

#### DS 4: Integracija IS sistemov prevoznika

---

- `TariffProductId` (int) – unikatni identifikator vrste vozovnice
- `TariffProductName` (string) – ime vrste vozovnice
- `ProductType` (int) – tip vrste vozovnice
- `IsConditional` (bool) – identifikator pogojne vozovnice
- `InactiveStartPeriod` (int) – izbira začetnega neaktivnega časa veljavnosti vozovnice
- `InactiveTimeUnit` (int) – časovna enota neaktivnega časa veljavnosti vozovnice
- `InactiveTimeQuantity` (int) – število časovnih enot neaktivnega časa veljavnosti vozovnice
- `InactiveOffsetTimeUnit` (int) – časovna enota zamika začetnega in končnega neaktivnega časa veljavnosti vozovnice
- `InActiveOffsetStartTime` (int) – zamik začetnega neaktivnega časa veljavnosti vozovnice
- `InActiveOffsetEndTime` (int) – zamik končnega neaktivnega časa veljavnosti vozovnice
- `ActiveStartPeriod` (int) – izbira začetnega aktivnega časa veljavnosti vozovnice
- `ActiveTimeUnit` (int) – časovna enota aktivnega časa veljavnosti vozovnice
- `ActiveTimeQuantity` (int) – število časovnih enot aktivnega časa veljavnosti vozovnice
- `ActiveOffsetTimeUnit` (int) – časovna enota zamika začetnega in končnega aktivnega časa veljavnosti vozovnice
- `ActiveOffsetStartTime` (int) – zamik začetnega aktivnega časa veljavnosti vozovnice
- `ActiveOffsetEndTime` (int) – zamik končnega aktivnega časa veljavnosti vozovnice
- `NumberOfProducts` (int) – število aktivnih vozovnic znotraj neaktivnega časa vozovnice
- `NumberOfProductsSetPeriod` (int) – perioda ponastavitve števila neaktivnih vozovnic
- `NumberOfProductsSetExtensionUnit` (int) – časovna enota podaljšanja ponastavitve vozovnic
- `NumberOfProductsSetExtensionQuantity` (int) – število enot podaljšanja ponastavitve vozovnic
- `DisablePurchaseTerminalType` (int) – omejitev nakupa glede na tip terminala
- `ProductCardLimit` (int) – največje dovoljeno število vozovnic zapisanih na kartico
- `VAT` (float) – Davčna stopnja davka na dodano vrednost ddv v odstotkih
- `IsSwitching` (bool) – identifikator prestopne vozovnice
- `InactiveValidationPriority` (int) – prioriteta aktivacije/validacije vozovnice
- `ActiveValidationPriority` (int) – prioriteta aktivacije/validacije vozovnice
- `ProductTimePriority` (int) – prioriteta aktivacije/validacije vozovnice
- `AbsoluteValidationTimeValidity` (`AbsoluteTicketTimeDefinition`) – absolutni dovoljeni čas uporabe vozovnice
- `PurchaseStartOffsetTimeUnit` (int) – časovna enota zamika, ki določa začetni čas za nakup vozovnice
- `PurchaseStartOffset` (int) – časovni zamik začetnega časa za nakup vozovnice
- `PurchaseEndOffsetTimeUnit` (int) – časovna enota zamika, ki določa končni čas za nakup vozovnice
- `PurchaseEndOffset` (int) – časovni zamik končnega časa za nakup vozovnice
- `Direction` (int) – smer vozovnice
- `AbsolutePurchaseTimeValidity` (`AbsoluteTicketTimeDefinition`) – absolutni dovoljeni čas nakupa vozovnice

## DS 4: Integracija IS sistemov prevoznika

- ConditionalProduct (ConditionalProductData) – parametri, ki določajo pogoje nakupa pogojnih vozovnic.

**TariffProductId** določa unikatni identifikator vrste vozovnice.

**TariffProductName** je ime vrste vozovnice, ki je lahko dolžine največ 25 znakov.

**ProductType** ali tip vrste vozovnice, določa kako se naj vozovnica obravnava na terminalu. V tabeli (Tabela 4.1) so definirane naslednje vrednosti vrst vozovnic.

TABELA 4.1: DEFINICIJA VREDNOSTI VRST VOZOVNIC

Vrednost ProductType	Opis
0	Privzeta vozovnica
1	Prtljaga
2	Odškodnina
3	Listnata vozovnica
4	Vozovnica z ločenim aktivnim časom trajanja
5	Listnata vozovnica z ločenim neaktivnim/aktivnim časom trajanja
6	Vozovnica veljavna na vseh relacijah
10	Medkrajevna vozovnica za skupino odraslih (posebni časovni režim – ločen aktivni čas trajanja)
11	Medkrajevna vozovnica za skupino otrok (posebni časovni režim – ločen aktivni čas trajanja)
12	Vozovnica za skupino odraslih veljavna na vseh relacijah (posebni časovni režim – ločen aktivni čas trajanja)
13	Vozovnica za skupino otrok veljavna na vseh relacijah (posebni časovni režim – ločen aktivni čas trajanja)
14	Medkrajevna vozovnica za skupino odraslih
15	Medkrajevna vozovnica za skupino otrok
16	Vozovnica za skupino odraslih veljavna na vseh relacijah
17	Vozovnica za skupino otrok veljavna na vseh relacijah

V tabeli (Tabela 4.1) imamo določene vrste vozovnic.

Privzeti tip vozovnice določa vozovnico, ki je namenjena potniku za lastni prevoz. Prevoz prtljage je predstavljen kot posebna vrsta vozovnice. Odškodnina je prav tako določena kot posebna vrsta vozovnice. Listnata vozovnica se natisne, to vrsto vozovnice je mogoče kupiti brez kartice. Vozovnica z ločenim aktivnim časom trajanja je vozovnica katere aktivni čas trajanja je določen v strukturi TariffClass z vrednostjo ValidityMinutes. Vozovnica s tem tipom ima vrednost ActiveTimeUnit privzeto

## DS 4: Integracija IS sistemov prevoznika

nastavljeno na minute (0) pri tem je vrednost `ActiveTimeQuantity` nadomeščena z vrednostjo `ValidityMinutes`. Ločen aktivni čas trajanja se uporablja predvsem pri medkrajevnih vozovnicah, kjer je čas trajanja aktivacije odvisen od prevožene razdalje.

**IsConditional** parameter določa ali je vrsto vozovnice mogoče kupiti le pod pogojem, da je na kartici predhodno kupljena neka druga vozovnica (`true`=vozovnico je mogoče kupiti le pod pogojem, `false`=vozovnico je mogoče kupiti brezpogojno).

**InactiveStartPeriod** določa začetni neaktivni čas veljavnosti vozovnice. Neaktivni čas veljavnosti vozovnice je čas znotraj katerega je zapisana vozovnica na kartici pripravljena na uporabo. Neaktivni čas veljavnosti se določi pri nakupu vozovnice, kjer se vozovnica na kartico zapiše kot neaktivna. Neaktivna vozovnica ima zapisan začetni in končni čas uporabe, ki določa časovno območje znotraj katerega je možno vozovnico aktivirati in uporabiti. Vozovnica se aktivira ob prvi uporabi (aktivaciji), le če je trenutni čas aktivacije znotraj neaktivnega časa vozovnice. Ob aktivaciji ali validaciji mora terminal preveriti ali obstaja absolutni čas za validacijo vozovnice (`AbsoluteValidationTimeValidity`). Če absolutni čas ne obstaja, terminal preverja le relativni neaktivni čas. V primeru ko absolutni čas obstaja, je dovoljena aktivacija ali validacija le, če tudi absolutni čas v danem trenutku omogoča aktivacijo ali validacijo. V tabeli (Tabela 4.2) so definirane časovne vrednosti `InactiveStartPeriod`. Začetni čas neaktivne vozovnice se določi na terminalu relativno glede na željeni čas uporabe vozovnice.

TABELA 4.2: DEFINICIJA ČASOVNIH VREDNOSTI

Časovna vrednost	Opis
0	sedaj (vrednost označuje trenutek takoj - sedaj)
1	dan (vrednost označuje vsak dan ob 00:00:00 uri)
2	vikend (vrednost označuje vsako soboto ob 00:00:00 uri)
3	teden (vrednost označuje vsak ponedeljek ob 00:00:00 uri)
4	mesec (vrednost označuje vsak mesec ob 00:00:00 uri)
5	1. januar (vrednost označuje vsak 1. januar ob 0:00:00)
6	1. februar (vrednost označuje vsak 1. februar ob 0:00:00)
7	1. marec (vrednost označuje vsak 1. marec ob 0:00:00)
8	1. april (vrednost označuje vsak 1. april ob 0:00:00)
9	1. maj (vrednost označuje vsak 1. maj ob 0:00:00)
10	1. junij (vrednost označuje vsak 1. junij ob 0:00:00)
11	1. julij (vrednost označuje vsak 1. julij ob 0:00:00)
12	1. avgust (vrednost označuje vsak 1. avgust ob 0:00:00)
13	1. september (vrednost označuje vsak 1. september ob 0:00:00)
14	1. oktober (vrednost označuje vsak 1. oktober ob 0:00:00)
15	1. november (vrednost označuje vsak 1. november ob 0:00:00)
16	1. december (vrednost označuje vsak 1. december ob 0:00:00)

V tabeli (Tabela 4.2) imamo definirane čase za pričetek veljavnosti neaktivne vozovnice.

## DS 4: Integracija IS sistemov prevoznika

**InactiveTimeUnit** določa časovno enoto za čas trajanja neaktivne vozovnice, ki se prične z InactiveStartPeriod. Vrednosti, ki predstavljajo časovne enote so opisane v tabeli (Tabela 4.3).

TABELA 4.3: DEDINICIJA VREDNOSTI ČASOVNOIH ENOT

Vrednost TimeUnit	Opis
0	Minuta
1	Dan
2	Teden
3	Mesec
4	Leto
5	Delovni dan (ponedeljek, torek, sredo, četrtek, petek)
6	Dan, ki se vedno prične v začetku prejšnjega meseca
7	Delovni dan (ponedeljek, torek, sredo, četrtek, petek, sobota)

**InactiveTimeQuantity** določa količino časovnih enot InactiveTimeUnit za določanje časa neaktivne vozovnice. Čas neaktivne vozovnice se ob nakupu določi tako: InactiveStartPeriod določa začetni čas. Začetnemu času se prišteje število InactiveTimeQuantity časovnih enot definiranih s parametrom InactiveTimeUnit.

**InactiveOffsetTimeUnit** določa časovno enoto zamika začetnega in končnega časa neaktivne vozovnice. V primeru, ko se čas veljavnosti neaktivne vozovnice prične izven časa definirane s parametrom InactiveStartPeriod, lahko začetni in končni čas neaktivne vozovnice zamaknemo za časovno enoto InactiveOffsetTimeUnit.

**InactiveOffsetStartTime** določa število enot InactiveOffsetTimeUnit za katere se zamkne začetni čas neaktivne vozovnice. Število enot zamika začetnega časa neaktivne vozovnice je lahko pozitivno ali negativno. Pri pozitivni vrednosti se zamik prišteva. Pri negativni vrednosti se zamik odšteva.

**InactiveOffsetEndTime** določa število enot InactiveOffsetTimeUnit za katere se zamkne končni čas neaktivne vozovnice. Število enot zamika končnega časa neaktivne vozovnice je lahko pozitivno ali negativno. Pri pozitivni vrednosti se zamik prišteva. Pri negativni vrednosti se zamik odšteva.

**ActiveStartPeriod** določa začetni aktivni čas vozovnice. Pri nakupu se vozovnica na kartico zapiše kot neaktivna. Pri prvi uporabi se vozovnica aktivira le, če je trenutni čas znotraj neaktivnega časa vozovnice. Aktivno vozovnico lahko potnik uporablja znotraj aktivnega časa veljavnosti. Aktivni čas vozovnice je torej čas, ko se potnik nahaja na vozilu. Aktivni čas veljavnosti vozovnice ne sme presegati neaktivnega časa veljavnosti vozovnice. V primeru, ko se aktivni čas prične znotraj in konča zunaj neaktivnega časa veljavnosti vozovnice, mora terminal skrajšati aktivni čas tako, da ne presega neaktivnega časa veljavnosti. V tabeli (Tabela 4.2) so definirane časovne vrednosti ActiveStartPeriod. Začetni čas aktivne vozovnice se določi relativno glede na trenutni čas prve validacije. V tabeli (Tabela 4.2) imamo definirane čase za pričetek veljavnosti aktivne vozovnice.

**ActiveTimeUnit** določa časovno enoto za čas trajanja aktivne vozovnice, ki se prične z ActiveStartPeriod. Vrednosti, ki predstavljajo časovne enote so opisane v tabeli (Tabela 4.1).

#### DS 4: Integracija IS sistemov prevoznika

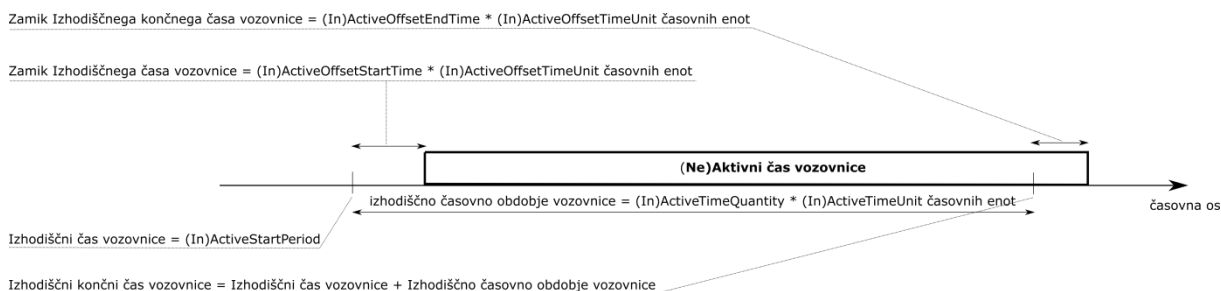
**ActiveTimeQuantity** določa količino časovnih enot ActiveTimeUnit za določanje časa aktivne vozovnice. Čas aktivne vozovnice se ob nakupu določi tako: ActiveStartPeriod določa začetni čas. Začetnemu času se prišteje število ActiveTimeQuantity časovnih enot definiranih s parametrom ActiveTimeUnit.

**ActiveOffsetTimeUnit** določa časovno enoto zamika začetnega in končnega časa neaktivne vozovnice. V primeru, ko se čas veljavnosti neaktivne vozovnice prične izven časa definiranega s parametrom InactiveStartPeriod, lahko začetni in končni čas neaktivne vozovnice zamaknemo za časovno enoto InactiveOffsetTimeUnit.

**ActiveOffsetStartTime** določa število enot ActiveOffsetTimeUnit za katere se zamkne začetni čas aktivne vozovnice. Število enot zamika začetnega časa aktivne vozovnice je lahko pozitivno ali negativno. Pri pozitivni vrednosti se zamik prišteva. Pri negativni vrednosti se zamik odšteva.

**ActiveOffsetEndTime** določa število enot ActiveOffsetTimeUnit za katere se zamkne končni čas aktivne vozovnice. Število enot zamika končnega časa aktivne vozovnice je lahko pozitivno ali negativno. Pri pozitivni vrednosti se zamik prišteva. Pri negativni vrednosti se zamik odšteva.

Parametri (InactiveStartPeriod, InactiveTimeUnit, InactiveTimeQuantity, InactiveOffsetTimeUnit, InactiveOffsetStartTime, InactiveOffsetEndTime) in (ActiveStartPeriod, ActiveTimeUnit, ActiveTimeQuantity, ActiveOffsetTimeUnit, ActiveOffsetStartTime, ActiveOffsetEndTime) določajo neaktivni in aktivni čas vozovnice. Določitev obeh časov (neaktivnega in aktivnega) prikazuje slika spodaj, kjer je neaktivni oziroma aktivni čas prikazan poudarjeno.



SLIKA 4.3: DOLOČANJE NEAKTIVNEGA ALI AKTIVNEGA ČASA VOZOVNICE

Pri določanju aktivnega časa vozovnice velja, da aktivni čas vozovnice ne sme presegati neaktivnega časa vozovnice. V primeru kadar so parametri obeh časov nastavljeni tako, da je aktivni čas v določenem časovnem segmentu zunaj neaktivnega časa, je potrebno aktivni čas korigirati tako, da ne bo presegal neaktivnega časa. Pričetek aktivnega ali neaktivnega časa je všteti v časovno periodo. Konec aktivnega ali neaktivnega časa ni všteti v časovno periodo.

Ob nakupu vozovnice se določita in na kartico zapišeta začetni in končni neaktivni čas vozovnice. Neaktivni čas vozovnice se prične z začetkom prve sekunde začetnega časa zapisanega na kartici in se konča s koncem zadnje sekunde končnega časa zapisanega na kartici.



## DS 4: Integracija IS sistemov prevoznika

Ob aktivaciji vozovnice se na kartico zapiše le končni aktivni čas vozovnice. Začetni aktivni čas vozovnice se ne zapisuje na kartico. Aktivni čas vozovnice se določi ob aktivaciji vozovnice in se konča s koncem zadnje sekunde končnega časa zapisanega na kartici.

**NumberOfProducts** določa število koriščenj vozovnice. Za kuponske vozovnice ta vrednost določa število voženj (kuponov). Pri ostalih vozovnicah je ta vrednost enaka 1.

**NumberOfProductsSetPeriod** določa periodo ponastavitve števila neaktivnih vozovnic. Vrednosti so opisane v tabeli (Tabela 4.2). S to vrednostjo nastavimo periodično ponastavitev števila koriščenj vozovnic na vrednost definirano s parametrom **NumberOfProducts**. Periodična ponastavitev števila koriščenj vozovnic poteka tako, da se ob prvi validaciji vozovnice na kartici, v času po definirani periodi **NumberOfProductsSetPeriod**, ponastavi število koriščenj vozovnice na vrednost **NumberOfProducts**. Pri kuponskih vozovnicah lahko s tem parametrom nastavimo časovno omejitev porabe kuponov. Na primer: kadar imamo neaktivni čas vozovnice nastavljen na večmesečno obdobje in periodo ponastavitve števila aktivnih vozovnic nastavljeno na 4 (začetek meseca), bo validator pri prvi validaciji po začetku meseca na kartici nastavljal prvotno število koriščenj vozovnice. S tem ima uporabnik mesečno omejeno porabo števila vozovnic definirano z **NumberOfProducts**. Pri ponastavitvi vozovnice se upošteva preostali neaktivni čas vozovnice. Kadar je preostali neaktivni čas ob ponastavitvi vozovnice večji ali enak eni časovni enoti neaktivnega časa, se ponastavitev števila aktivnih vozovnic izvede. Sicer se dovoli le aktivacija ene preostale neaktivirane vozovnice, pri tem se število preostalih neaktiviranih vozovnic postavi na 0.

**NumberOfProductsSetExtensionUnit** predstavlja časovno enoto za zamik podaljšanja ponastavitve števila vozovnic, ki ga določa parameter **NumberOfProductsSetPeriod**. Ponastavitev vozovnic lahko časovno zamaknemo le v primeru, ko pred ponastavitvijo niso porabljene vse vozovnice. Če imamo prisotna parametra **NumberOfProductsSetExtensionUnit** in **NumberOfProductsSetExtensionQuantity**, se ponastavitev vozovnic lahko zamakne največ za čas, ki ga določata ta parametra. Časovni zamik ponastavitve se določi tako, da se absolutnemu času, ki je določen s parametrom **NumberOfProductsSetPeriod** prišteje število časovnih enot določeno s parametrom **NumberOfProductsSetExtensionQuantity**. S tem se potniku omogoči, da znotraj podaljšanega časa aktivira (uporabi) preostale neaktivirane vozovnice. Šele po aktivaciji vseh preostalih vozovnic se lahko znotraj podaljšanega časa ponastavi število neaktivnih vozovnic. Po poteku podaljšanega časa se preostale neaktivirane vozovnice odstranijo in število neaktivnih vozovnic ponastavi. Časovne enote so opisane v tabeli (Tabela 4.1).

**NumberOfProductsSetExtensionQuantity** določa količino časovnih enot **NumberOfProductsSetExtensionUnit** za določanje podaljšanega časa za ponastavitev števila vozovnic.

**DisablePurchaseTerminalType** določa tip terminala na katerem je nakup vozovnice onemogočen. Nakup na posameznem tipu terminala se onemogoči s postavitvijo določenega bita. (bit0=1 onemogočen nakup na validatorjih, bit1=1 onemogočen nakup na mobilnih terminalih, bit2=1 onemogočen nakup na aplikacijah). Bit, ki predstavlja tip terminala določa parameter **iTerminalTypeMask** iz strukture **IdentifyData** (parameter **iTerminalTypeMask** ima postavljen en bit na zaporednem mestu, ki predstavlja tip terminala).



DS 4: Integracija IS sistemov prevoznika

---

**ProductCardLimit** določa omejitev števila kupljenih vozovnic istega tipa. Omejitev se nanaša le na število neaktivnih vozovnic zapisanih na kartico.

**VAT** določa davčno stopnjo davka v odstotkih, ki se upošteva pri prodaji vozovnice.

**IsSwitching** določa ali je vozovnica prestopna (true=prestopna vozovnica, false=neprestopna vozovnica). Pri prestopni vozovnici je dovoljen prestop med različnimi prevoznimi sredstvi v aktivnem času vozovnice. Sicer prestop ni dovoljen.

**InactiveValidationPriority** določa prioriteto vozovnice za aktivacijo. Kadar je na kartici več različnih vozovnic, terminal na podlagi prioritete izvede aktivacijo ali validacijo vozovnice. Posotopsek aktivacije in validacije je opisan spodaj.

**ActiveValidationPriority** določa prioriteto vozovnice za validacijo. Kadar je na kartici več različnih vozovnic, terminal na podlagi prioritete izvede aktivacijo ali validacijo vozovnice. Posotopsek aktivacije in validacije je opisan spodaj.

**ProductTimePriority** določa prioriteto vozovnice za validacijo glede na trenutni čas. Kadar je na kartici več različnih vozovnic, terminal na podlagi prioritete izvede aktivacijo ali validacijo vozovnice. Posotopsek aktivacije in validacije je opisan spodaj.

Postopek aktivacije ali validacije vozovnice je sledeč: Potnik kupi vozovnico, ki jo pri koriščenjih vožnje aktivira in validira. Pri prvem koriščenju vožnje se vozovnica aktivira. Le aktivirana vozovnica se lahko validira. Torej imamo najprej aktivacijo s katero določimo aktivni čas trajanja vozovnice. Znotraj aktivnega časa lahko vozovnico večkrat validiramo. Hkrati lahko aktiviramo ali validiramo le eno vozovnico. Kadar je na kartici več vozovnic določimo vozovnico za aktivacijo ali validacijo po sledečem postopku:

Vsaki vozovnici na kartici (ki ima ustrezen neaktivni ali aktivni čas veljave) določimo prioriteto tako, da upoštevamo vrednosti parametrov: **InactiveValidationPriority** in **ActiveValidationPriority**. Vrednost parametra **InactiveValidationPriority** upoštevamo le pri neaktivnih vozovnicah. Vrednost parametra **ActiveValidationPriority** upoštevamo le pri aktivnih vozovnicah. Tako določeno prioriteto vozovnic primerjamo. Vozovnico z najvišjo prioriteto aktiviramo oziroma validiramo. Kadar imamo več vozovnic z enako prioriteto, potem upoštevamo vrednost parametra **ProductTimePriority**. S pomočjo parametra **ProductTimePriority** določimo časovno prioriteto aktivacije ali validacije, ki ustreza absolutnemu času trajanja vozovnice v sekundah, določenemu po sledečem postopku:

V primeru, ko je **ProductTimePriority** enak 0:

*(časovna prioriteta) = število sekund od začetka neaktivnega časa vozovnice do trenutnega časa. (1)*

V primeru, ko je **ProductTimePriority** enak 1:

*(časovna prioriteta) = število sekund od trenutnega časa do konca neaktivnega časa vozovnice. (2)*

V primeru vozovnic z ločenim aktivnim časom trajanja (**ProductType=4**) namesto neaktivnega časa v enačbah (1) in (2) upoštevamo čas validiranega obdobja.

#### DS 4: Integracija IS sistemov prevoznika

**AbsoluteValidationTimeValidity** določa absolutni čas veljavnosti, s katerim natančno določimo kdaj se lahko vozovnica uporabi (aktivira/validira). **AbsoluteValidationTimeValidity** je določen s časom **AbsoluteTicketTimeDefinition**.

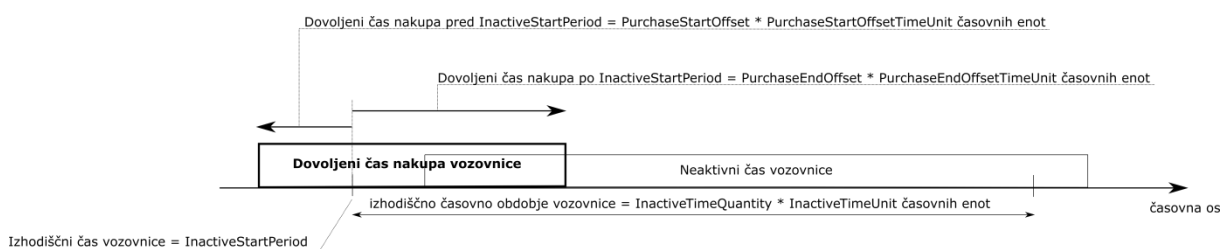
**PurchaseStartOffsetTimeUnit** določa časovne enote začetnega časa nakupa vozovnice relativno na parameter **InactiveStartPeriod**. Vrednosti, ki predstavljajo časovne enote so opisane v tabeli (Tabela 4.3).

**PurchaseStartOffset** določa število enot, ki predstavljajo relativni začetni čas v katerem je dovoljen nakup vozovnice. Relativni začetni čas nakupa vozovnice je določen relativno glede na parameter **InactiveStartPeriod**. Začetni čas nakupa je določen tako, da se parametru **InactiveStartPeriod** prišteje ali odšteje število **PurchaseStartOffset** časovnih enot v obliki **PurchaseStartOffsetTimeUnit**.

**PurchaseEndOffsetTimeUnit** določa časovne enote končnega časa nakupa vozovnice relativno na parameter **InactiveStartPeriod**. Vrednosti, ki predstavljajo časovne enote so opisane v tabeli (Tabela 4.3).

**PurchaseEndOffset** določa število enot, ki predstavljajo relativni končni čas v katerem je dovoljen nakup vozovnice. Relativni končni čas nakupa vozovnice je določen relativno glede na parameter **InactiveStartPeriod**. Končni čas nakupa je določen tako, da se parametru **InactiveStartPeriod** prišteje ali odšteje število **PurchaseStartOffset** časovnih enot v obliki **PurchaseStartOffsetTimeUnit**.

Parametri (**PurchaseStartOffsetTimeUnit**, **PurchaseStartOffset**, **PurchaseEndOffsetTimeUnit**, **PurchaseEndOffset**) predstavljajo relativni čas znotraj katerega je možno vozovnico kupiti glede na izhodiščni parameter **InactiveStartPeriod**, kot je poudarjeno prikazano na sliki spodaj.



SLIKA 4.4: DOVOLJENI RELATIVNI ČAS NAKUPA VOZOVNICE

Pred nakupom vozovnice mora terminal preveriti ali se nahaja znotraj dovoljenega relativnega časa za nakup vozovnice. Prav tako mora terminal preveriti ali obstaja absolutni čas za nakup vozovnice (**AbsolutePurchaseTimeValidity**). Če absolutni čas ne obstaja, terminal preverja le relativni čas. V primeru ko absolutni čas obstaja, je dovoljen nakup le, če se absolutni čas popolnoma prekriva z relativnim časom znotraj katerega je dovoljen nakup.

**Direction** – smer vozovnice (NULL ali 0- dvosmerna, 1- Enosmerna v smeri, ki je definirana z relacijo)

**AbsolutePurchaseTimeValidity** določa absolutni čas veljavnosti, s katerim natančno določimo za kateri neaktivni čas veljavnosti vozovnice se lahko vozovnica kupi, pri tem mora biti celoten neaktivni čas

#### DS 4: Integracija IS sistemov prevoznika

vozovnice znotraj absolutnega časa veljavnosti `AbsolutePurchaseTimeValidity`. `AbsolutePurchaseTimeValidity` je določen s časom `AbsoluteTicketTimeDefinition`.

`AbsoluteTicketTimeDefinition` predstavlja bitno polje s katerim natančno določimo absolutni čas znotraj katerega je dovoljen nakup ali validacija vozovnice. Dovoljen čas nakupa ali validacije imenujemo čas veljavnosti. Določen je s postavitvijo posameznih bitov v bitnem polju `AbsoluteTicketTimeDefinition` z naslednjimi pravili:

- Prvi trije bajti (**byte [0 - 2]**) določajo uro dneva, ko je vozovnica veljavna. Pri tem uporabimo bite od 0 do 23 (po vrstnem redu mali endian). Postavitev posameznega bita na 1 pomeni, da je vozovnica veljavna le v izbrani uri dneva. Postavitev posameznega bita na 0 pomeni, da vozovnica v tem času ni veljavna. Če so vsi biti postavljeni na 0, je vozovnica veljavna v vseh urah (enako kot če bi bili vsi biti postavljeni na 1).
- Naslednji bajt (**byte [3]**) določa dan v tednu, ko je vozovnica veljavna. Dneve v tednu določajo biti 0 do 6 (po vrstnem redu mali endian). Postavitev posameznega bita na 1 pomeni, da je vozovnica veljavna le v izbranem dnevu tedna. Če so vsi biti postavljeni na 0, je vozovnica veljavna vse dni v tednu (enako kot če bi bili vsi biti postavljeni na 1).
- Sledijo štirje bajti (**byte [4 - 7]**), ki določajo dan v mesecu, ko je vozovnica veljavna. Za to uporabimo bite 0 do 30 (po vrstnem redu mali endian). Postavitev bita na 1 pomeni, da je vozovnica veljavna le izbrani dan v mesecu. Če so vsi biti postavljeni na 0, je vozovnica veljavna vse dni v mesecu (enako kot če bi bili vsi biti postavljeni na 1).
- Sledita dva bajta (**byte [8 - 9]**), ki določata veljavnost vozovnice za mesec v letu. Mesec v letu določa prvih 12 bitov – biti 0 do 11 (po vrstnem redu mali endian). Postavitev posameznega bita na 1 pomeni, da je vozovnica veljavna le v izbranem mesecu leta. Če so vsi biti postavljeni na 0, je vozovnica veljavna vse mesece v letu (enako kot če bi bili vsi biti postavljeni na 1).

Čas veljavnosti lahko nastavljamo v poljubnih časovnih periodah - sistem nastavljanja časa veljavnosti vozovnic omogoča fleksibilno določanje časovnih period. Pravila za časovne periode so sledeča:

- Če želimo omejiti absolutni čas veljavnosti vozovnice mu dodamo eno ali več vrednosti `AbsoluteTicketTimeDefinition`.
- Če vrednost `AbsoluteTicketTimeDefinition` nivpisana, čas veljavnosti ni neomejen.
- Posamezna vrednost `AbsoluteTicketTimeDefinition` omogoča nastavitev naslednjih kategorij za časovno veljavnost: ure v dnevu, dnevi v tednu, dnevi v mesecu in meseci v letu.
- Če v posamezni kategoriji ni izbrana nobena vrednost (ni postavljen noben bit), se to upošteva, enako kot v primeru, če bi bile izbrane vse vrednosti (postavljeni vsi biti).
- Posamezna vrsta vozovnice ima lahko vpisanih največ 10 vrednosti `AbsoluteTicketTimeDefinition`.

Strukturo `AbsolutePurchaseTimeValidity` uporabljamo za absolutno določanje časa validacije in nakupa vozovnic.

**ConditionalProduct** predstavlja podatkovno strukturo, ki določa vozovnice katere je mogoče pogojno kupiti (pogojne vozovnice), če je na kartici predhodno kupljena vozovnica, ki vsebuje s to strukturo.

## DS 4: Integracija IS sistemov prevoznika

Pogojne vozovnice so označene s parametrom `IsConditional=true`. Pogojne vozovnice v strukturi `ConditionalProduct` definiramo s tremi parametri:

- **TariffLocationId** identifikator lokacije uporabe pogojne vozovnice
- **TariffProductId** identifikator vrste pogojne vozovnice
- **TariffClassId** identifikator tarifnega razreda pogojne vozovnice

Pri vseh vrstah vozovnic so časovni parametri definirani tako, da se merijo do sekunde natančno. Pri tem imajo vsi časovni intervali spodnjo mejo vključeno v interval in posledično zgornjo mejo izključeno.

### 4.2.4 STATUSI – STRUKTURA *TARIFFSTATUS*

Status je podatkovna struktura definirana z naslednjimi parametri:

- **Id** (int) – unikatni identifikator statusa
- **StatusType** – (int) – unikatni identifikator tipa statusa.
- **Name** (string) – ime statusa

**Id** določa unikatni identifikator statusa.

**StatusType** določa tipa statusa. Tipi statusov so predstavljeni v tabeli (Tabela 4.4).

TABELA 4.4: DEFINICIJA STATUSOV

Vrednost StatusType	Opis
0	Privzet status za nepersonificirane kartice
1	Subvencioniran status dijak (za potnike s pravico do nakupa subvencioniranih vozovnic)(**)
2	Subvencioniran status študent (za potnike s pravico do nakupa subvencioniranih vozovnic)(**)
3	Subvencioniran status udeleženec izobraževanja odraslih (za potnike s pravico do nakupa subvencioniranih vozovnic)(**)
4	Status za potnike brez kartice IJPP (možnost nakupa le listnatih vozovnic)
5	Invalid s spremljevalcem
6	Zunanji status
7	Splošni status za personificirane kartice
8	Subvencioniran status dijak športnik (**)
9	Subvencioniran status študent športnik (**)
10	Subvencioniran status udeleženec izobraževanja odraslih športnik (**)
11	Subvencioniran upokojenec. Samodejno podaljševanje na terminalu za eno leto(*)(**).
12	Subvencioniran vojni veteran. Samodejno podaljševanje na terminalu za eno leto(*)(**).

## DS 4: Integracija IS sistemov prevoznika

13	Subvencioniran invalid. Samodejno podaljševanje na terminalu za eno leto(*)(**).
14	Subvencioniran starejši od 65 let. Samodejno podaljševanje na terminalu za eno leto(*)(**).
14	Otroški imenski status (**)
15	Otroški prenosni status (**)

(\*)Nekatere vrste statusov lahko podaljšujemo samodejno na terminalu za eno leto. Podaljšanje se lahko izvede eno leto po poteku statusa.

(\*\*) Čas veljavnosti statusa je omejen. Po poteku časa veljavnosti, ni več možen nakup produktov za ta status. Statusi brez (\*\*) so časovno neomejeni.

**Name** je ime statusa, ki je lahko dolžine največ 25 znakov.

### 4.2.5 TARIFNI RAZREDI – STRUKTURA TARIFFCLASS

Tarifni razred je podatkovna struktura definirana z naslednjimi parametri:

- Id (int) – unikatni identifikator tarifnega razreda
- Name (string) – ime tarifnega razreda
- Priority (int) – Prioriteta tarifnega razreda
- KilometersFrom (int) – kilometri od. Spodnja meja je vključena v interval (od – do)
- KilometersTo (int) – kilometri do. Zgornja meja ni vključena v interval (od – do)
- ValidityMinutes (int) – čas veljavnosti v minutah

**Id** določa unikatni identifikator tarifnega razreda.

**Name** je ime tarifnega razreda, ki je lahko dolžine največ 25 znakov.

**Priority** predstavlja prioriteto tarifnega razreda, ki nam pove znotraj koliko tarifnih razredov velja vozovnica. Za podrobnejši opis glej poglavje.

**KilometersForm** in **KilometersTo** – se uporablja le za tarifne razrede v linijski tarifi. Vrednosti predstavljata spodnjo in zgornjo mejo v prevoženih kilometrih na linijskem prometu. Vrednosti sta predstavljeni s celim številom, kjer je osnovna enota 1 km.

**ValidityMinutes** – čas veljavnosti tarifnega razreda (v minutah), za vozovnice kjer je veljavnost časovno omejena glede na tarifni razred.

### 4.3 PODATKOVNA STRUKTURA TARIF NA TERMINALIH

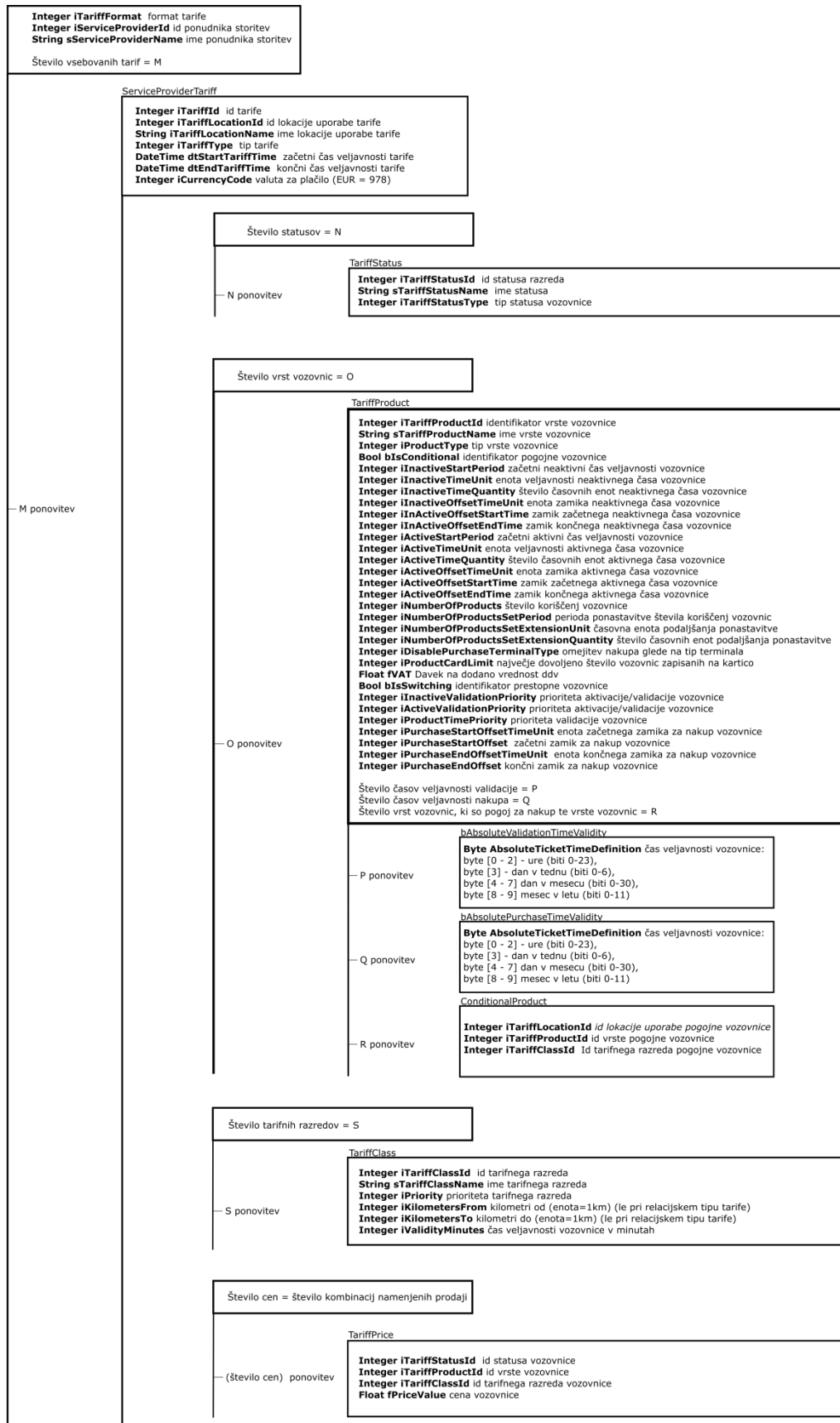
#### DS 4: Integracija IS sistemov prevoznika

---

Terminali na katerih se lahko vozovnice kupijo, uporabijo ali preverijo, potrebujejo vse podatke iz tarife. Trenutno aktualno tarifo nudi zaledna stran preko web servisa z metodo GetTariff, kjer je vhodni podatek identifikator terminala za katerega želimo tarifo. Izhodni podatek je seznam tarif, ki so na voljo za prodajo na tem terminalu. Zgradba podatkovne strukture tarife (izhodni podatek web servis metode GetTariff) je prikazana na sliki (Slika 4.5).

Podatkovna struktura tarife vsebuje podatke za vse lokacije uporabe, kjer se lahko prodajajo in validirajo vozovnice na določenem terminalu.

## DS 4: Integracija IS sistemov prevoznika



SLIKA 4.5: ZGRADBA PODATKOVNE STRUKTURE TARIFE

## DS 4: Integracija IS sistemov prevoznika

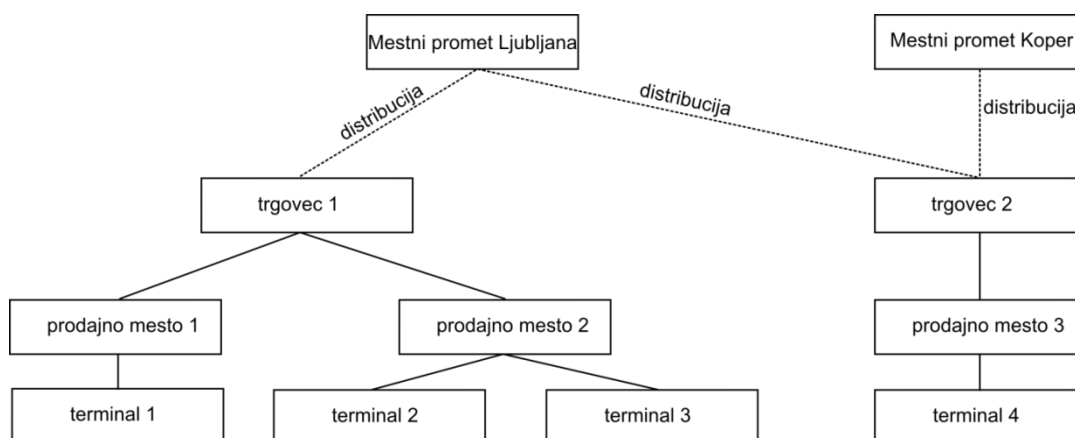
## 4.4 DISTRIBUCIJA TARIF

Vozovnice je možno kupiti, uporabiti ali preveriti le na vnaprej določenih terminalih prevoznikov, ali podjetij katerim transakcijski sistem pošlje izbrane tarife na podlagi unikatnega identifikatorja terminala. Katere tarife bo prejel terminal je odvisno nastavljene distribucijske verige v zalednem sistemu IJPP. Pri nastavitvah distribucijskih verige se upoštevajo naslednja dejstva:

- Posamezna lokacija uporabe ima lahko hkrati aktivno le eno tarifo.
- Trgovec je podjetje, ki trži vozovnice za izbrane lokacije uporabe, zato prejme preddefiniran seznam aktivnih tarif lokacij uporabe, katerih vozovnice lahko trži.
- Trgovec ima eno ali več prodajnih mest na katerih prodaja vozovnice iz tarife končnim uporabnikom. Trgovec lahko opsijsko pošilja podatke o prodajnih mestih v sistem IJPP.
- Na prodajnih mestih so nameščeni terminali, s katerimi se izvedejo elektronske transakcije o nakupih, uporabah ali preverjanjih vozovnic iz tarife. Te transakcije se morajo pošiljati v zaledni sistem IJPP.

Primer distribucijske verige je prikazan na sliki (Slika 4.6), kjer imamo dve lokaciji uporabe (Mestni promet Ljubljana in Mestni promet Koper), dva trgovca, tri prodajna mesta in štiri terminale. Trgovec 1 je lastnik dveh prodajnih mest »prodajno mesto 1« in »prodajno mesto 2«. Na prodajnem mestu 1 je nameščen terminal 1, na prodajnem mestu 2 sta nameščena terminala 2 in 3. Trgovec 1 ima nastavljeno distribucijo le za tarifo Mestnega prometa Ljubljana, torej lahko na terminalih 1, 2 in 3 prodaja le vozovnice za Mestni promet Ljubljana.

Trgovec 2 je lastnik enega prodajnega mesta »prodajno mesto 3«, na katerem je nameščen terminal 4. Trgovec 2 ima nastavljeno distribucijo obeh lokacij uporabe, zato lahko na terminalu 4 prodaja vozovnice iz tarif Mestnega prometa Ljubljana in Mestnega prometa Koper.

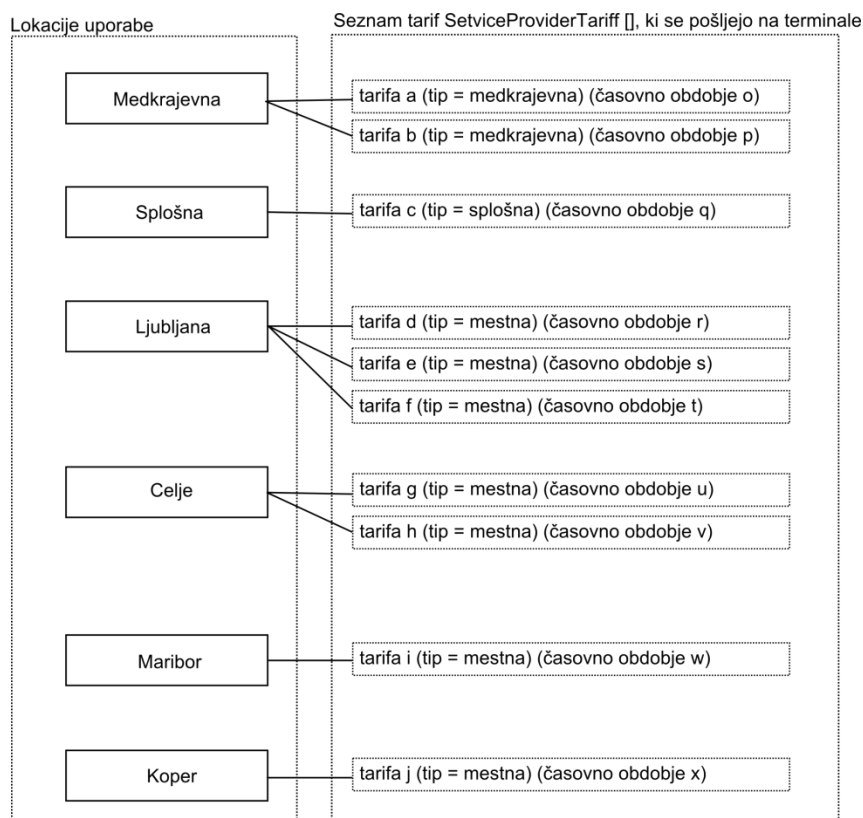


SLIKA 4.6: HIERARHIJA PRI PRODAJI VOZOVNIC

Tarife ločujemo glede na lokacijo uporabe kot prikazuje slika (Slika 4.6).



## DS 4: Integracija IS sistemov prevoznika



SLIKA 4.7: PRAKAZ LOKACIJ UPORABE IN SEZNAMA TARIF, KI SE POŠLJEJO NA TERMINALE

Posamezna lokacija uporabe lahko vsebuje eno ali več aktivnih tarif. Primer je prikazan na sliki (Slika 4.7). Na posamezni lokaciji uporabe je v določenem časovnem obdobju lahko aktivna le ena tarifa. Pri tem imamo lahko več tarif aktivnih v več različnih neprekrivajočih se časovnih območjih. Več tarif uporabimo kadar imajo na primer produkti v časovnem območju (o) različne cene kot v časovnem območju (p) pri medkrajevni lokaciji uporabe. Pri tem se časovna območja (o) in (p) ne prekrivata. Prav tako se med seboj ne prekrivajo časovna območja (r), (s) in (t) v lokaciji uporabe Ljubljana. Enako velja za časovna območja (u) in (v) v lokaciji uporabe Celje. Časovna območja se na različnih lokacijah uporabe med seboj lahko prekrivajo.

## 4.5 VEČRAZREDNE MESTNE TARIFE

V primeru, ko imamo mestni tip tarife, le ta lahko vsebuje več razredov (območij). V definirani strukturi tarife nimamo podatkov o prehodih med območji, zato te podatke pridobimo z WS metodo GetTariffClassTransitionMatrix, ki nam vrne tabelo kot je prikazana spodaj.

TABELA 4.5: TABELA S PREHODI MED OBMOČJI

Območje iz/v	V Ljubljano	V Škofljico	V Grosuplje
Iz Ljubljane	1	2	3
Iz Škofljice	2	1	2
Iz Grosupelj	3	2	1

V tabeli je predstavljeno število območij (tarifnih razredov), ki jih mora vsebovati vozovnica, kadar potnik potuje med ali znotraj območji.

Tarifni razredi so znotraj tarife rastoče razvrščeni po prioriteti s parametrom (iPriority). Na prvem mestu nahaja tarifni razred z najnižjo prioriteto. Na drugem mestu se nahaja tarifni razred z drugo najnižjo prioriteto itd.. Prioriteta v tarifi nam pove za koliko tarifnih razredov (območij) iz zgornje tabele potrebujemo vozovnico, da lahko potujemo iz območja A v območje B.

Na primeru zgornje tabele vidimo, da za vozovnico znotraj Ljubljane, Škofljice ali Grosupelj potrebujemo vozovnico katere tarifni razred ima prioriteto enako 1. Kadar pa na primer potujemo iz Škofljice v Ljubljano, potrebujemo vozovnico katere tarifni razred ima prioriteto 2.

## 4.6 PRAVILA ZA TARIFE

Lokacija uporabe ima v tarifi definirane vozovnice, ki se lahko pri trgovcih na terminalih kupijo, uporabijo in preverjajo. Pogosto želimo onemogočiti prodajo določenih vrst vozovnic glede na segment v prodajni verigi, kot so terminali, prodajna mesta, podjetja, ipd. Takšno onemogočanje prodaje dosežemo s pravili za tarife. Osnovna ideja pravil za tarife je v maskiranju cen za posamezne vrste vozovnic, statusa in tarifne razrede. In sicer: s pravili določimo katere vrste vozovnic, statusa in tarifne razrede želimo omogočiti na posameznem segmentu.

Omejitev prodaje vozovnic na terminalih izvedemo s pravili za tarife. S pravili omejimo prodajo vozovnic tako, da cene v tarifi nastavimo na -1. Pravila za tarifo delujejo kot maska s katero izbrane cene vozovnic v tarifi postavimo na -1 preden se tarifa pošlje na terminal. Vrednost cene -1 v tarifi pomeni, da kombinacije vrste vozovnic, statusa in tarifnega razreda ni mogoče prodati.

## DS 4: Integracija IS sistemov prevoznika

---

V prodajni verigi je lahko hkrati aktivnih več pravil, vendar se za maskiranje cen upošteva le eno pravilo glede na prioritete:

1. Najprej se v transakcijskem sistemu preveri ali obstaja aktivno pravilo za terminal. Če obstaja, se upošteva to pravilo.
2. Če aktivno pravilo za terminal ne obstaja, se preveri ali obstaja aktivno pravilo za podjetje. Če obstaja, se upošteva to pravilo.

Če nobeno od zgoraj naštetih pravil ne obstaja, se na terminal pošlje nespremenjena tarifa brez omejitev prodaje. V primeru, ko imamo aktivni dve ali več pravil enake prioritete, terminal ne bo prejel tarife.

Pravila lahko vpišemo s pomočjo spletnega vmesnika na dva načina:

- ločeno za vrste vozovnic, statuse in tarifne razrede
- posebej za vsako kombinacijo vrste vozovnice, statusa in tarifnega razreda

V prvem primeru vpisa pravil posebej izberemo tarifne razrede vrste vozovnic in statuse, ki jih želimo omogočiti za prodajo na terminalih.

Če na terminalih želimo natančno določiti kombinacije tarifnih razredov, vrst vozovnic in statusov, ki so na voljo za prodajo, izberemo še drugi način vpisa pravila za tarife na spletni vmesnik.

Pri vpisu pravila izberemo (s kljukico) kaj želimo na terminalu prodajati. Neobkljukane kombinacije na terminalu ne bodo na voljo za prodajo.

Ko je pravilo določeno ga shranimo in aktiviramo. Če imamo predhodno aktivno pravilo enake prioritete, moramo po aktivaciji novega pravila staro pravilo arhivirati.

### 4.7 UPRAVLJANJE S TARIFAMI - ADMINISTRACIJA

V tem poglavju bomo predstavili postopke upravljanja s tarifami kot so: kreiranje nove tarife, kreiranje pravil in napotke pri upravljanju s tarifami

#### 4.7.1 KREIRANJE NOVE TARIFE

Pri kreiranju nove tarife moramo imeti pripravljeno naslednje:

- Pred kreiranjem nove tarife morajo biti vpisani vse vrste vozovnic, statusi in tarifni razredi, ki bodo vsebovani v tarifi. Če katera vrsta vozovnice, status ali tarifni razred manjka ga vpišemo na novo. Že obstoječih vozovnic, statusov in tarifnih razredov ne spreminjamo, če so že vsebovani v aktivni tarifi.

## DS 4: Integracija IS sistemov prevoznika

---

- Pripravljen moramo imeti veljavni cenik za vrste vozovnic, statute in tarifne razrede, ki bodo vsebovani v novi tarifi.

Novo tarifo kreiramo po naslednjem vrstnem redu:

1. Na spletnem vmesniku izberemo novo tarifo, prikažejo se prazna polja za vnos podatkov.
2. Vpišemo lokacijo uporabe.
3. Vpišemo začetni datum veljavnosti tarife.
4. Vpišemo končni datum veljavnosti tarife.
5. Vpišemo prioriteto. Za posamezno lokacijo uporabe sta relevantni le dve tarifi z najvišjo prioriteto. Zato izberemo ustrezno največjo vrednost prioritete, glede na že obstoječe tarife lokacije uporabe.
6. Status pustimo na *Neaktivna*.
7. Izberemo privzeti status (če obstaja).
8. Kliknemo na gumb *Shrani*.

Sedaj imamo shranjeno neaktivno tarifo v katero dodamo vrste vozovnic, statute in tarifne razrede, ki jih želimo vključiti v novo tarifo. V naslednjem koraku vpišemo cene za vse kombinacije vrst vozovnic, statusov in tarifnih razredov. Če posamezna kombinacija vrste vozovnice, statusa in tarifnega razreda ni voljo za nakup vpišemo ceno -1.

Ko imamo vpisane vse podatke, tarifo shranimo. V zadnjem koraku tarifo aktiviramo in shranimo. Aktivirana tarifa je na voljo za prenos na terminale.

### 4.7.2 KREIRANJE PRAVIL

Aktivna tarifa z veljavnim datumom in z najvišjo prioriteto se pošlje terminalom za ustrezno lokacijo uporabe. To tarifo lahko omejimo s pravili za tarife, ki jih prav tako konfiguriramo v s pomočjo spletnega vmesnika.

Glede na prioriteto ločimo več tipov pravil. Tip pravila nam podaja množico terminalov za katere velja pravilo. Obstaja pet različnih tipov pravil, ki so predstavljena po prioritetni lestvici od največje do najmanjše:

- pravilo za terminal
- pravilo za tip terminala
- pravilo za podjetje.

Najvišjo prioriteto ima pravilo za terminal, sledi mu pravilo za tip terminala. Najnižjo prioriteto ima pravilo za podjetje. Prioriteta nam pove, katero pravilo bo uporabljeno, če jih je aktivnih več. Namreč uporabljeno je lahko vedno le eno pravilo. Na primer: če imamo aktivno pravilo za tip terminala in pravilo za podjetje, se bo upoštevalo le pravilo za tip terminala, ker ima višjo prioriteto. Več identičnih aktivnih pravil ne sme obstajati hkrati, sicer sistem ne more določiti pravila z najvišjo prioriteto.

## DS 4: Integracija IS sistemov prevoznika

---

Pravilo kreiramo na sledeč način:

1. V spletnem vmesniku za upravljanje s pravili za tarife izberemo novo pravilo.
2. Vpišemo lokacijo uporabe.
3. Izberemo želen tip pravila.
4. Vrednost *Status* pustimo na *Neaktivno* in pravilo shranimo.
5. Izberemo vrste vozovnic, statusa in tarifne razrede, ki jih želimo omogočiti za prodajo.
6. Pravilo ponovno shranimo.

Pravilo je pripravljeno na aktivacijo. Aktiviramo ga s spremembo stanja iz *Neaktivno* v *Aktivno* in s shranjevanjem spremenjenega stanja.

Če želimo določiti natančnejšo kombinacijo vrste vozovnic, statusov in tarifnih razredov, ki se lahko prodajajo na terminalih, izberemo natančnejši vmesnik za določitev pravil, kjer imamo možnost določitve poljubne kombinacije za prodajo.

### 4.7.3 POMEMBNI NAPOTKI PRI UPRAVLJANJU S TARIFAMI

Spremembe v tarifah pomenijo spremembo delovanja v celotnem transakcijskem sistemu. Za zmanjšanje možnosti napak, se moramo držati napotkov in protokola pri vsaki posodobitvi tarife ali pravila za tarife. Osnovni napotki pri upravljanju s tarifami so:

- Vsako tarifo ali pravilo je potrebno pred posodobitvijo na produkcijskem sistemu testirati na ločenem testnem sistemu.
- Ko se spreminjajo tarife in pravila za tarife se obvesti vse odgovorne osebe, ki skrbijo za terminale, ki se jim je spremenila tarifa ali pravilo.
- Po uspešnem testiranju in potrditvi iz strani testerjev, se pripravi dokument za spremembo tarife v produkciji kjer se navedejo spremembe, ki so bile izvedene v tarifi ali pravilih (Zabeleži se staro in novo stanje).
- Dokument se pošlje vsem vpletenim osebam, ki morajo potrditi posodobitev.
- Nato se lahko prične z deli za posodobitev tarife ali pravil na produkcijskem sistemu.

Posodobitev tarif je potrebno izvajati po dogovorjenem protokolu. Primer protokola pri različnih postopkih upravljanja s tarifami:

1. Ko smo testirali spremembe tarife na testnem sistemu lahko posodobimo produkcijski sistem.
2. Pred posodobitvijo tarife pregledamo trenutno stanje že aktivnih tarif in pravil. S tem ugotovimo spremembe, ki jih bomo vnesli z novimi tarifami ali pravili.
3. Pri posodobitvah tarif in pravil moramo voditi evidenco, kjer zapišemo spremembe, ki smo jih vnesli v sistem.
4. Pred aktivacijo tarif ali pravil še enkrat preverimo vpisane vrednosti. Napačno vpisane vrednosti pomenijo napačne cene ali onemogočen nakup vozovnic na terminalih.

#### DS 4: Integracija IS sistemov prevoznika

---

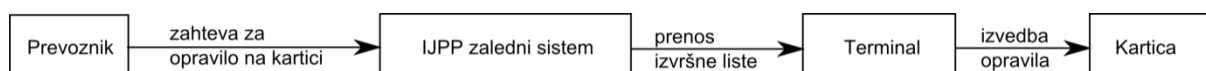
5. Po aktivaciji, tarife ali pravila ni več možno spreminjati. Če želimo vnesti spremembe je potrebno kreirati novo tarifo ali pravilo.
6. Aktivacijo pravila izvedemo tako, da najprej aktiviramo novo pravilo, nato v čim krajšem času arhiviramo staro pravilo. Starega pravila ne smemo arhivirati pred aktivacijo novega pravila.
7. Dva enaka pravila z enako prioriteto ne smeta biti aktivna istočasno (razen kratek čas med aktivacijo novega in arhiviranjem starega pravila).
8. Po posodobitvi tarif ali pravil, spremljamo delovanje sistema in se prepričamo, da smo posodobitev izvedli uspešno.

## 5 ZGRADBA IZVRŠNE LISTE

### 5.1 KONCEPTUALNI OPIS IZVRŠNE LISTE

Izvršna lista je seznam kartic (CUID) znotraj sistema IJPP, ki se prenaša iz zalednega sistema na terminal. Z razliko od črne liste izvršna lista vsebuje dodatne podatke za posamezno kartico (CUID). Ti dodatni podatki za posamezno kartico (CIUD) določajo posamezna opravila (npr. prenos vozovnice na kartico), ki se izvedejo na terminalu ob prislonitvi kartice k terminalu.

Funkcija izvršne liste je avtomatizirana izvedba opravil na IJPP karticah, katerih zahteve se nastavijo v zalednem sistemu in izvršijo na terminalih, kot je prikazano na sliki (Slika 5.1). Izvršno listo uporabljamo v primeru, kadar nimamo fizično prisotne kartice, da bi lahko opravilo takoj izvedli na kartici. V tem primeru z izvršno listo nastavimo oddaljeno opravilo, ki se izvede v trenutku, ko je kartica naslednjič uporabljena na terminalu sistema IJPP.



SLIKA 5.1: OSNOVNI KONCEPT IZVEDBE ODDALJENIH OPRAVIL Z IZVRŠNO LISTO

Izvršna lista je podatkovna struktura, ki omogoča oddaljeno izvedbo različnih opravil na karticah s pomočjo terminalov v transakcijskem sistemu IJPP. Oddaljeno se na terminalih lahko izvajajo sledeča opravila:

- Prenos vozovnice na kartico,
- Brisanje vozovnice na kartici,
- Nastavitev novega statusa na kartici,
- Začasno onemogočanje uporabe kartice,
- Onemogočanje uporabe vozovnice,
- Omogočanje uporabe vozovnice.
- Posodobitev relacije na kartici.
- Posodobitev vloge za subvencionirane vozovnice na kartici.

V naslednjih poglavjih bomo natančneje opisali princip delovanja izvršne liste.

## 5.2 DELOVANJE IZVRŠNE LISTE

IJPP funkcionalnost izvršne liste podpira dva načina delovanja: offline in online način delovanja.

V offline načinu delovanja se izvršna lista prenese na terminale z vsemi podatki. V tem primeru lahko terminali izvajajo opravila tudi, ko nimajo povezave z zalednim sistemom IJPP, saj izvršna lista vsebuje vse potrebne podatke o opravilih. Ko uporabnik prisloni kartico k terminalu, lahko terminal takoj izvede opravilo, ne da bi se pri tem povezal na zaledni sistem, saj so v izvršni listi prisotni vsi podatki o opravilih.

V online načinu delovanja se v izvršni listi na terminale prenesjo le podatki o seznamu kartic in vrsti opravila. Ko uporabnik prisloni kartico, terminal preveri ali je kartica v seznamu izvršne liste. Če je kartica v seznamu, se terminal po potrebi sproti poveže na zaledni sistem in prenese morebitne dodatne podatke o opravi, ki ga je potrebno izvesti na kartici.

Prednost offline načina je hitra izvedba opravil brez sprotne povezave terminala z zalednim sistemom. Slabost je potencialno velika količina podatkov izvršne liste, ki jih mora terminal prenesti iz taledenga sistema in hraniti v svojem pomnilniku. Za lažji prenos velike količine podatkov izvršne liste iz zalednega sistema, lahko terminali izvedejo prenos izvršne liste po delih.

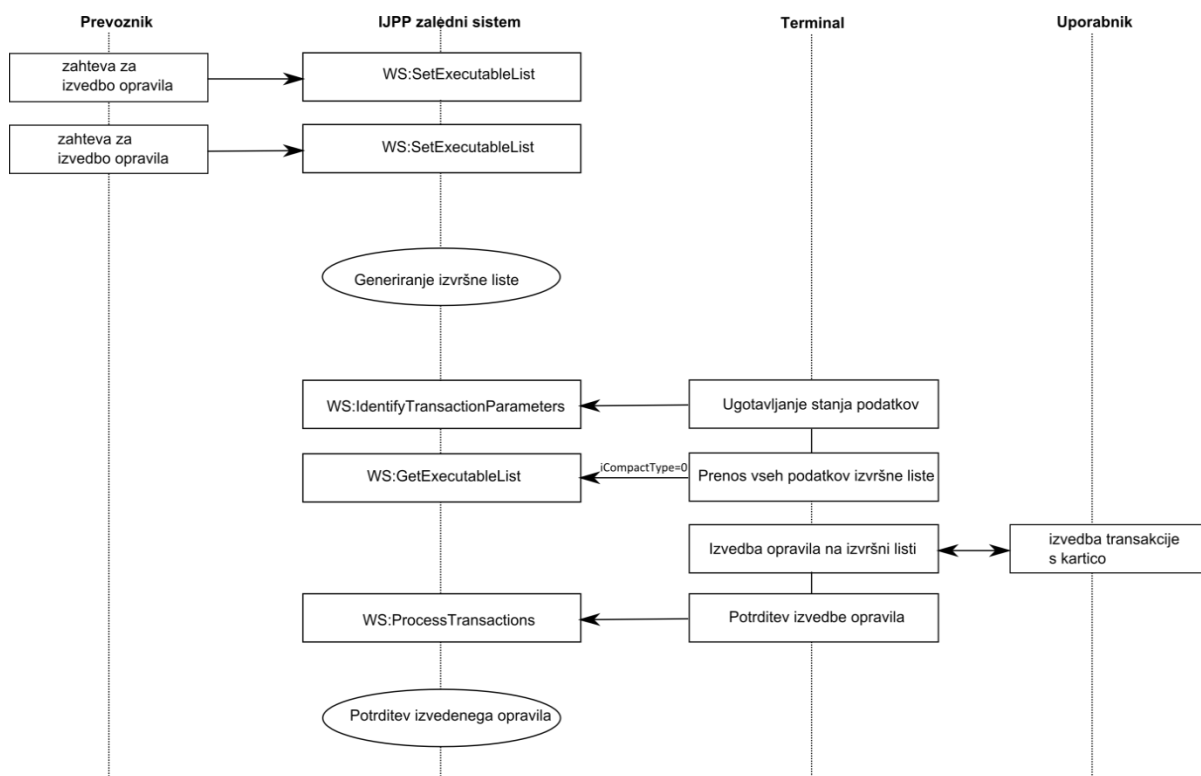
Prednost online načina je majhna količina podatkov izvršne liste na terminalu, med tem ko je slabost počasnejše delovanje zaradi sprotne povezovanja terminala na zaledni sistem in nedelovanje izvršne liste v primeru, ko terminal nima povezave z zalednim sistemom.

V nadaljevanju bomo natančneje opisali delovanje offline in online načina izvršne liste



### 5.2.1 OFFLINE NAČIN DELOVANJA IZVRŠNE LISTE

V praksi offline način delovanja izvršne liste poteka po sledečem postopku, kot je prikazan na sliki (Slika 5.2):



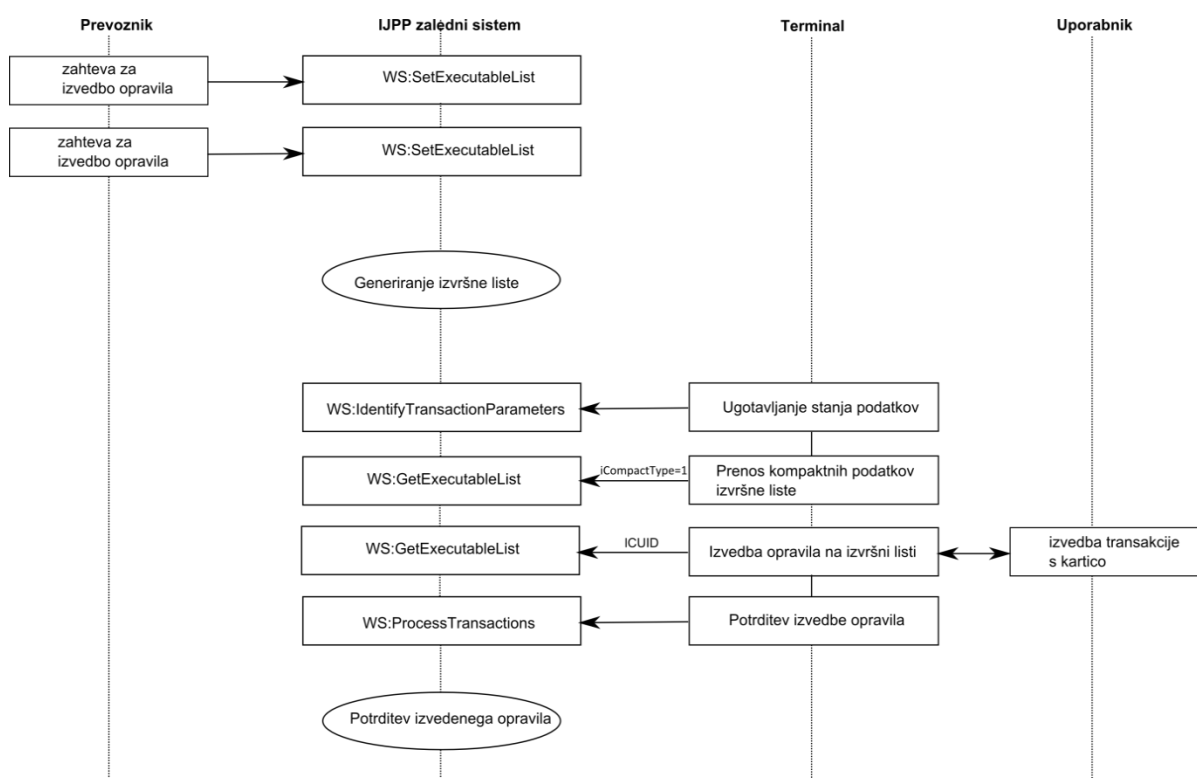
SLIKA 5.2: OFFLINE NAČIN DELOVANJA IZVRŠNE LISTE

Slika (Slika 5.2) prikazuje primer offline delovanja izvršne liste. Offline delovanje izvršne liste lahko opišemo z naslednjimi koraki:

- Ponudnik storitve ali prevoznik odda zahteve na zaledni sistem IJPPP za izvedbo določenih opravil na izbrani kartici (WS metoda SetExecutableList).
- Zaledni sistem enkrat na dan (običajno zgodaj zjutraj) generira izvršno listo, v katero doda zahteve za oddaljena opravila na karticah iz prejšnjega koraka.
- Izvršna lista se prenese na terminale na podlagi preverjanja verzije izvršne liste (WS metoda IdentifyTransactionParameters). Izvršna lista vsebuje vse podatke za izvedbo navedenih opravil. Terminal pridobi vse podatke izvršne liste tako, da postavi parameter iCompactType=0 v WS metodi GetExecutableList.
- Pri vsaki prislomitvi kartice k terminalu, terminal preveri ali je prislonjena kartica na izvršni listi. Če je kartica na izvršni listi, terminal izvede opravilo na kartici, ki je opisano v izvršni listi in v zaledni sistem pošlje IJPP transakcijo o uspešno izvedenem opravilu.
- Po prejeti IJPP transakciji o uspešno izvedenem opravilu, zaledni sistem potrdi izvedbo opravila na kartici, ki se odstrani v naslednji verziji izvršne liste.

### 5.2.2 ONLINE NAČIN DELOVANJA IZVRŠNE LISTE

Online način delovanja izvršne liste je podoben offline načinu le, da se pri online načinu terminal ob izvedbi transakcije s kartico poveže na zaledni sistem, kjer dobi vse podatke o opravi za trenutno prislono kartico. Online način je praviloma počasnejši kot offline način, ker se terminal ob prislono kartici sproti povezuje na zaledni sistem. Online načina izvedbe opravi iz izvršne liste ni možno izvesti, če terminal nima povezave na zaledni sistem. Online izvedba opravi poteka po postopku, kot je prikazan na sliki (Slika 5.3):



SLIKA 5.3: ONLINE NAČIN DELOVANJA IZVRŠNE LISTE

Slika (Slika 5.3) prikazuje primer online delovanja izvršne liste. Online delovanje izvršne liste lahko opišemo z naslednjimi koraki:

- Ponudnik storitve ali prevoznik odda zahteve na zaledni sistem IJPP za izvedbo določenih opravi na izbrani kartici (WS metoda SetExecutableList).
- Zaledni sistem enkrat na dan (običajno zgodaj zjutraj) generira izvršno listo, v katero doda zahteve za oddaljena opravi na karticah iz prejšnjega koraka.
- Izvršna lista se prenese na terminale na podlagi preverjanja verzije izvršne liste (WS metoda IdentifyTransactionParameters). Izvršna lista vsebuje kompaktne podatke za izvedbo navedenih opravi. Terminal pridobi kompaktne podatke izvršne liste tako, da postavi parameter iCompactType=1 v WS metodi GetExecutableList. Kompaktni podatki vsebujejo le podatkovne pare CUID in Action.

- Pri vsaki prislonitvi kartice k terminalu, terminal preveri ali je prislonjena kartica na izvršni listi. Če je kartica na izvršni listi, se terminal poveže na zaledni sistem, kjer pridobi manjkajoče podatke za izvedbo opravila na kartici (z WS metodo `GetExecutableList` kjer je prisoten vhodni parameter `ICUID`). Ko prejeme podatke terminal izvede opravilo na kartici, ki je opisano v izvršni listi in v zaledni sistem pošlje IJPP transakcijo o uspešno izvedenem opravilu.
- Po prejemu IJPP transakcije o uspešno izvedenem opravilu, zaledni sistem potrdi izvedbo opravila na kartici, ki se odstrani v naslednji verziji izvršne liste.

### 5.3 UPRAVLJANJE Z IZVRŠNO LISTO - ADMINISTRACIJA

Upravljanje z izvršno listo izvajamo z administratorskima WS metodama `SetExecutableList` in `DelExecutableList`, s katero dodajamo in odvezujemo opravila izvršne liste. Dodajanje opravil poteka v dveh korakih:

- Zaledni sistem redno sprejema zahteve za dodajanje opravil na izvršno listo z WS metodo `SetExecutableList`.
- V preddefiniranih časovnih periodah zaledni sistem sprejete zahteve doda na izvršno listo. Pri dodajanju sprejetih opravil se generira nova verzija izvršne liste, kjer se zraven novih zahtev v izvršno listo dodajo tudi neizvedena stara opravila iz prejšnje verzije izvršne liste.

Dodana opravila na izvršno listo lahko prekličemo z administratorsko WS metodo `DelExecutableList`, kar prav tako poteka v dveh korakih:

- Zaledni sistem redno sprejema zahteve za odvzem (preklic) neizvršenih opravil iz izvršne liste z WS metodo `DelExecutableList`. Zahteva za preklic se nanaša na vsa neizvedena opravila za določeno kartico od naslednjega generiranja izvršne liste naprej.
- V preddefiniranih časovnih periodah zaledni sistem sprejete zahteve doda na izvršno listo. Starejša neizvršena opravila na karticah, ki so bila preklicana, se pri tem ne dodajo na novo verzijo izvršne liste.

### 5.4 PODATKOVNI MODEL IZVRŠNE LISTE

To poglavje opisuje podatkovni model izvršne liste. Podatke izvršne liste terminal prejme z WS metodo `GetExecutableList`.

Vhodni podatki metode `GetExecutableList`:

- `TerminalId (int)` - Identifikator terminala
- `iCompactType(int)` – Vrsta podatkov, ki jih želimo na terminalu (0-vsi podatki, 1-kompaktni podatki)

- CUID(long) - Identifikator kartice je neobvezen podatek. Če je ta podatek prisoten, potem dobimo v izvršni listi podatke le za kartico, ki ustreza vrednosti CUID. To možnost uporabimo le pri online preverjanju kartice.

Izhodni podatki metode GetExecutableList so vsebovani v strukturi mExecutableData. Struktura mExecutableData vsebuje naslednje podatke:

- CUID(long) – identifikator kartice na kateri je potrebno izvesti opravilo
- Action(int) – vrsta opravlila. (0-online preverjanje kartice, 1-prenos vozovnice na kartico, 2-brisanje vozovnice na kartici, 3-nastavitev novega statusa na kartici, 4-začasno onemogočanje uporabe kartice, 5-onemogočanje uporabe vozovnice, 6-omogočanje uporabe vozovnice, 7-posodobitev relacije na kartici, 8-posodobitev vloge o subvencioniranih vozovnicah na kartici .
- AdditionalCardObject – je abstraktni tip, ki lahko vsebuje podatke o vrsti vozovnice, statusu ali identifikatorju transakcije.

Abstraktni tip AdditionalCardObject je povezan z vrednostjo Action, kjer predstavlja tip podatkov odvisno od vrednosti Action, kot je opisano v tabeli (Tabela 5.1):

TABELA 5.1: TIP PODATKOV ADDITIONALCARDOBJECT GLEDE NA VREDNOST ACTION

Action	Vrsta opravlila	AdditionalCardObject
0	online preverjanje kartice	Brez (le pri kompaktnem tipu izvršne liste)
1	prenos vozovnice na kartico	CardProduct
2	brisanje vozovnice na kartici	CardProductShort
3	nastavitev novega statusa na kartici	CardPassengerStatus
4	začasno onemogočanje uporabe kartice	TemporaryCUID
5	onemogočanje uporabe vozovnice	CardProductShort
6	omogočanje uporabe vozovnice	CardProductShort
7	posodobitev relacije	CardRelation
8	Posodobitev vloge subvencioniranih voz.	SubsidyApplication

## 5.1 INKREMENTALNI PRENOS IZVRŠNE LISTE

Izvršna lista lahko potencialno vsebuje zelo veliko količino podatkov. Da bi preprečili vsakokratni prenos istih podatkov iz izvršne liste, lahko Izvršno listo nalagamo inkrementalno.

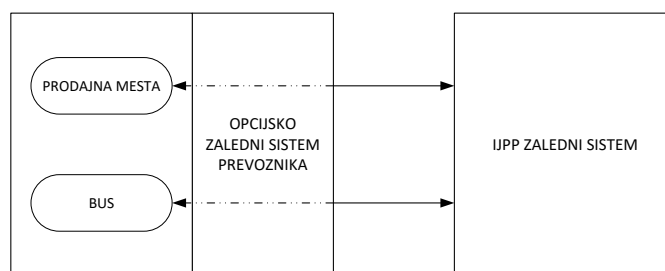
Inkrementalni prenos izvršne liste sprožimo tako, da pri klicu WS metode GetExecutableList pošljemo identifikator zadnje izvršne liste na terminalu. V tem primeru nam WS metoda vrne le razliko podatkov med zadnjo izvršno listo na terminalu in trenutno izvršno listo.

## 6 INTEGRACIJA Z ZALEDNIM SISTEMOM

Zaledni sistem IJPP omogoča komunikacijo med naslednjimi komponentami:

1. Terminalska opremo prevoznikov,
2. Opcijsko z zalednim sistemom prevoznikov.

Na spodnji sliki (Slika 6.1) sta prikazani dve opciji za integracijo z zalednim sistemom IJPP.



SLIKA 6.1: SHEMA INTEGRACIJE Z ZALEDNIM SISTEMOM

Z IJPP zalednim sistemom se prevozniki lahko integrirajo neposredno s svojo terminalske opremo ali posredno preko svojega zalednega sistema.

### 6.1 PODPRTE FUNKCIONALNOSTI

IJPP zaledni sistem omogoča napreden seznam podprtih funkcionalnosti. Sistemi integrirani v IJPP zaledni sistem imajo velik izbor funkcionalnosti ki najbolj ustrezajo njihovim procesnim zahtevam. To poglavje navaja seznam teh funkcionalnosti ki so dostopne preko IJPP komunikacijskega vmesnika. Integracija se lahko izvede delno ali v celoti, glede na zahteve integriranja.

Zaledni sistem tako omogoča komunikacijo z različnimi podsklopi:

1. Vozni redi (postajališča, linije, cone, delovni nalogi,...),
2. Terminali,
3. Tarifa (cene, vozovnice, produkti,...),
4. Črne liste,
5. Izvršne liste,
6. Šifranti (napake, prodajna mesta, pogodbe,... )

Komunikacija je dvosmerna. Tako lahko oprema ki je integrirana v zaledni sistem prejema podatke kot so črne liste, tarife, vozni redi in tudi pošilja podatke v zaledni sistem. Sporoča lahko na primer dodajanje nove opreme (recimo terminal, prodajno mesto,..) ali blokado kartice.

Funkcionalnosti transakcijskega sistema delimo na dva dela: transakcijske in administrativne funkcionalnosti. Naslednje funkcionalnosti so podprte s strani IJPP zalednega sistema:

#### **Transakcijske Funkcionalnosti:**

- Identifikacija stanja transakcijskih podatkov v zalednem sistemu (6.1.1)
- Distribucija črne liste (6.1.3)
- Ugotavljanje statusa kartice (6.1.4)
- Nastavitev statusa kartice (6.1.5)
- Distribucija izvršne liste (6.1.6)
- Distribucija tarif (6.1.7)
- Distribucija podatkov o prehodih med področji (6.1.8)
- Prenos uspešno izvedenih transakcij za procesiranje na zaledni sistem (6.1.9)
  - Prodaja vozovnice (6.1.9.1),
  - Validacija vozovnice (6.1.9.2),
  - Permanentna blokada kartice (6.1.9.4)
  - Začasna blokada kartice (**Napaka! Vira sklicevanja ni bilo mogoče najti.**)
  - Blokada vozovnice (**Napaka! Vira sklicevanja ni bilo mogoče najti.**)

#### **Administracijske funkcionalnosti:**

- Vnos, spreminjanje in pridobivanje podatkov o karticah.
- Vnos, spreminjanje in pridobivanje podatkov o uporabnikih
- Vnos, spreminjanje in pridobivanje podatkov o podjetjih.
- Vnos, spreminjanje in pridobivanje podatkov o uporabnikih podjetja.
- Vnos, spreminjanje in pridobivanje podatkov o prodajnih mestih.
- Vnos, spreminjanje in pridobivanje podatkov o terminalih.
- Vnos, nastavitev in pridobivanje parametrov vrste vozovnic.
- Vnos, nastavitev in pridobivanje parametrov statusov vozovnic.
- Vnos, nastavitev in pridobivanje parametrov tarifnih razredov.
- Vnos, nastavitev in pridobivanje podatkov o tarifah.
- Vnos, nastavitev in pridobivanje podatkov o distribucijah tarif
- Vnos, nastavitev in pridobivanje podatkov o pravilih za tarife.
- Vnos, nastavitev in pridobivanje podatkov o izvršni listi
- Pridobivanje podatkov o poročilih izvedenih transakcij

V naslednjih poglavjih bomo na kratko opisali namen in delovanje naštetih funkcionalnosti.

#### 6.1.1 IDENTIFIKACIJA STANJA TRANSAKCIJSKIH PODATKOV V ZALEDNEM SISTEMU

Terminali potrebujejo ažurne transakcijske podatke za sprotno delovanje. Ti podatki predstavljajo točen čas v podatkovni bazi, identifikacijo aktualne črne liste, izvršne liste, tarif in prehodih med področji.

#### 6.1.2 ČASOVNI INTERVALI AŽURIRANJA

Časovni intervali ažuriranja podatkov naj ne presegajo 24 ur. To pomeni da terminali morajo v manj kot 24 urah vsaj enkrat preveriti v IJPP zalednem sistemu ali je mogoče prišlo do sprememb v:

- Tarifi,
- Črni listi,
- Izvršni listi,
- Seznamu voznih redov.

Terminali pridobijo ta podatek z metodo za ugotavljanje trenutnega stanja podatkov zalednega sistema (metoda `IdentifyTransactionParameters`).

Prav tako morajo proti IJPP zalednemu sistemu pošiljati podatke o transakcijah v časovnem intervalu ki naj bo čimkrajši in ne presega 24 ur.

Za prenos IJPP transakcij se uporablja metoda `ProcessTransaction`.

V primeru da je terminal ugotovil da je prišlo do spremembe ta v naslednjem koraku pridobi nove podatke.

#### 6.1.3 DISTRIBUCIJA ČRNE LISTE

Črna lista vsebuje seznam kartic katerih veljavnost je bila nepreklicno prekinjena zaradi izgube kartice, tehnične neizpravnosti, nespoštovanja pravil s strani uporabnika kartice,... Prezem črne liste in uporaba logik povezanih z njo je ključnega pomena za pravilno delovanje sistema. Komunikacijski vmesnik omogoča prevzem črne liste. Prav tako omogoča dodajanje novih kartic na listo preko zahteve za blokado kartice. Osnovni podatek o kartici na črni listi je enolična 7 bajtna identifikacijska številka. Vsaka črna lista ima tudi unikatni podatek o verziji črne liste. Kasneje se ta verzija pošilja proti zalednemu sistemu v primeru da je kartica bila blokirana. S tem se vodi podatek o tem katera verzija črne liste je omogočila blokado kartice. Preverjanje črne liste se izvaja intervalno (interval je definiran v poglavju 6.1.2). Prenos pa po potrebi če je prišlo do spremembe. Terminal takoj prične s pridobivanjem črne liste, ko ugotovi, da je prišlo do sprememb v črni listi. Črna lista se lahko prenese v celoti ali inkrementalno. V celoti se črna lista prenese na primer ob prvem zagonu. Da se privarčuje na prenesenih podatkih in poveča odzivnost sistema, ta omogoča inkrementalni prenos črne liste, kjer se prenese samo kartice ki so bile dodane ali odstranjene iz črne liste.

#### **6.1.4 UGOTAVLJANJE STATUSA KARTICE**

Kartice lahko imajo enega izmed naslednjih statusov: (0 – neaktivna, 1 – aktivna, 3 – blokirana, 4 – uničena, 5 – odvzeta, 8 – najdena, 10 – odvzeta-uničena, 11– zastarana, 12 – zastarana blokirana. Pri ugotavljanju statusa kartice se iz zalednega sistema pošlje podatek o trenutnem statusu kartice.

#### **6.1.5 NASTAVITEV STATUSA KARTICE**

Statuse kartic opisane v prejšnjem poglavju želimo v določenih primerih spremeniti. Za to uporabimo transakcijsko funkcionalnost za spremembo statusa kartice.

#### **6.1.6 DISTRIBUCIJA IZVRŠNE LISTE**

Izvršna lista vsebuje seznam kartic za katere je potrebno izvesti eno izmed naslednjih opravil: Prenos vozovnice na kartico, brisanje vozovnice na kartici, nastavitev novega statusa na kartici, začasno onemogočanje uporabe kartice, online preverjanje kartice, začasno onemogočanje uporabe vozovnice Komunikacijski vmesnik omogoča prevzem izvršne liste. Preverjanje izvršne liste se izvaja intervalno (interval je definiran v poglavju 6.1.2). Prenos pa po potrebi če je prišlo do spremembe. Terminal takoj prične s pridobivanjem nove izvršne liste, ko ugotovi, da je prišlo do sprememb.

#### **6.1.7 DISTRIBUCIJA TARIF**

Tarifa je podatkovna struktura, ki določa pravila za prodajo in validacijo vozovnic v transakcijskem sistemu IJPP. Komunikacijski vmesnik omogoča prevzem aktualnih tarif za nakup in validacijo vozovnic. Preverjanje tarife se izvaja intervalno (interval je definiran v poglavju 6.1.2). Prenos pa po potrebi če je prišlo do spremembe. Terminal takoj prične s pridobivanjem nove tarife, ko ugotovi, da je prišlo do sprememb.

#### **6.1.8 DISTRIBUCIJA PODATKOV O PREHODIH MED PODROČJI**

Pri prehodu potnika med različnimi področji potrebujemo pravila o prehodih med področji. Za to uporabimo transakcijsko funkcionalnost za prevzem podatkov o prehodih med področji.

#### **6.1.9 PRENOS USPEŠNO IZVEDENIH TRANSAKCIJ ZA PROCESIRANJE NA ZALEDNI SISTEM**

Vse transakcije izvedene s karticami na terminalih je potrebno prenesti v zaledni sistem, kjer se procesirajo, saj na podlagi procesiranih podatkov lahko izdelamo poročila za poravnavo in statistična poročila. Prenos uspešno izvedenih transakcij se izvaja intervalno (interval je definiran v poglavju 6.1.2). V naslednjih podpoglavjih so opisani različni tipi transakcij.

##### **6.1.9.1 Prodaja vozovnice**

Podatke o prodanih vozovnicah je potrebno posredovati v zaledni sistem. Ti podatki se procesirajo in pripravijo za izdelavo poročil za poravnavo.

##### **6.1.9.2 Validacija vozovnice**

Potniki se morajo ob vstopu na vozilo validirati. Validacija mora biti opravljena tudi v primeru da ima potnik veljavno aktivno vozovnico na kartici. To omogoča vodenje statističnih podatkov za kasnejšo morebitno procesiranje. Te validacije je potrebno pošiljati v zaledni sistem. Validator je lahko avtonomen oziroma validacijo opravlja sprevodnik ali voznik.



Ob validaciji se v navedenem vrstnem redu preverja:

- Veljavnost kartice,
- Veljavnost vozovnice na vstopnem mestu.

#### **6.1.9.3 Stornacija nakupa vozovnice**

Nekoriščno (neaktivirano) vozovnico je možno stornirati. Podatke o storniranih vozovnicah je potrebno posredovati v zaledni sistem.

#### **6.1.9.4 Permanentna blokada kartice**

Vse blokade kartice ki se na podlagi črne liste permanentno blokirajo je potrebno sporočiti v zaledni sistem. Ta mehanizem omogoča reduciranje črnih list. Ponovna aktivacija kartice po permanentni blokadi ni mogoča.

#### **6.1.9.5 Online prenos vozovnice na kartico**

Vozovnico je mogoče prenesti na kartico daljinsko (online) z izvršno listo. Podatke o prenosu vozovnice na kartico iz izvršne liste je potrebno sporočiti v zaledni sistem.

#### **6.1.9.6 Online brisanje vozovnice na kartici**

Vozovnico je mogoče daljinsko (online) izbrisati iz kartice z izvršno listo. Podatke o izbrisu vozovnice iz kartice je potrebno sporočiti v zaledni sistem.

#### **6.1.9.7 Online nastavitev novega statusa na kartici**

Status potnika, ki je zapisan na kartici je mogoče daljinsko (online) prenesti na kartico z izvršno listo. Podatke o prenosu statusa na kartico je potrebno sporočiti v zaledni sistem.

#### **6.1.9.8 Zavrnitev transakcije**

V primeru, ko potnik nima možnosti nakupa, aktivacije ali validacije vozovnice, je v zaledni sistem potrebno posredovati podatke o zavrnitvi transakcije. Vzroki za zavrnitev transakcije so lahko sledeči:

- Potnik nima veljavne vozovnice
- Kartica je začasno blokirana
- Vozovnica je začasno blokirana

#### **6.1.9.9 Prijava vožnje**

Vsaka prijava vožnje se posreduje na zaledni sistem.

#### **6.1.9.10 Prijava blagajnika**

Vsaka prijava blagajnika se posreduje na zaledni sistem.

#### **6.1.9.11 Odjava blagajnika**

Vsaka odjava blagajnika se posreduje na zaledni sistem.

#### **6.1.9.12 Aktivacija vozovnice**

Potniki se morajo ob vstopu na vozilo validirati. Prvo validacijo po nakupu vozovnice imenujemo aktivacija. Vse aktivacije je potrebno posredovati na zaledni sistem. Ob aktivaciji se v navedenem vrstnem redu preverja:

- Veljavnost kartice,
- Veljavnost vozovnice na vstopnem mestu.

#### **6.1.9.13 Inicializacija kartice**

Postopek vpisa, priprave in formatiranja kartice za uporabo v transakcijskem sistemu IJPP imenujemo inicializacija kartice. Vsaka inicializacija kartice se pošlje v zaledni sistem.

#### **6.1.9.14 Duplikat kartice**

Pri vsaki izdelavi duplikata kartice moramo podatke o duplikatu posredovati na zaledni sistem.

#### **6.1.9.15 Kontrolorjeva transakcija**

Pri vsaki kontroli vozovnice iz strani kontrolorjev moramo podatke o kontroli posredovati na zaledni sistem.

#### **6.1.9.16 Prijava Kontrolorja**

Pred vsako kontrolo se mora kontrolor prijaviti na validator vozila. Pri tem moramo podatke o prijavi kontrolorja posredovati na zaledni sistem.

#### **6.1.9.17 Dodajanje medkrajevne relacije na kartico**

Na kartico lahko dodamo medkrajevno relacijo brez vozovnice. Pri tem moramo podatke o zapisani medkrajevni relaciji posredovati na zaledni sistem.

#### **6.1.9.18 Brisanje medkrajevne relacije na kartici**

Iz kartice lahko izbrišemo medkrajevno relacijo. Pri tem moramo na zaledni sistem posredovati podatek (ideks) izbrisane relacije.

#### **6.1.9.19 Vračilo vozovnice**

V določenih primerih lahko potnik vrne vozovnico. Pri tem moramo na zaledni sistem posredovati podatke o vrnjeni vozovnici.

#### **6.1.9.20 Zamenjava vozovnice - stornacija**

V določenih primerih lahko potnik zamenja vozovnico. Pri tem moramo na zaledni sistem posredovati podatke o zamenjani in novi vozovnici.

#### **6.1.9.21 Ročna validacija vozovnice**

Pri ročni validaciji moramo na zaledni sistem posredovati enake podatke kot pri validaciji.

#### **6.1.9.22 Blokiranje vozovnice**

Na kartici lahko blokiramo vozovnico preko izvršne liste. Pri tem moramo na zaledni sistem posredovati podatke o blokiran vozovnici.

#### **6.1.9.23 Odblokiranje vozovnice**

Na kartici lahko odblokiramo blokirano vozovnico preko izvršne liste. Pri tem moramo na zaledni sistem posredovati podatke o odblokirani vozovnici.

#### **6.1.9.24 Neuspelo branje kartice**

Ko kartice ne moremo v celoti prebrati je na zaledni sistem potrebno posredovati prebrane podatke o kartici.

#### **6.1.9.25 Zamenjava vozovnice - nakup**

V določenih primerih lahko potnik zamenja vozovnico. Pri tem moramo na zaledni sistem posredovati podatke o zamenjani in novi vozovnici.

#### **6.1.9.26 Posodobitev relacije na izvršni listi**

Relacije na kartici lahko posodobaljšamo preko izvršne liste. Pri takšni posodobitvi relacije moramo na zaledni sistem posredovati podatke o posodobitvi.

#### **6.1.9.27 Transakcija za razhroščevanje**

Terminal lahko pošlje transakcijo, ki v zalednem sistemu nima učinka in je namenjena le za preverjanje podatkov, testiranje in razhroščevanje.



## 7 METODE SPLETNE STORITVE IJPP - TRANSAKCIJSKI SISTEM

Spletne storitve IJPP temeljijo na konceptih storitveno usmerjene arhitekture (SOA) omogočajo nam dinamično gradnjo postopkov za pridobivanje logično povezanih podatkov iz sistema IJPP. Prav tako nam spletne storitve IJPP omogočajo varno in zanesljivo hranjenje podatkov v transakcijski ali arhivski zbirki. Spletne storitve IJPP uporabljajo SOAP protokol in XML/XSD sheme za zagotavljanje povezljivosti z različnimi odjemalci.

Komunikacija med odjemalcem in spletnimi storitvami IJPP poteka preko varne (SSL) seje. Pri varni komunikaciji med strežnikom spletnih storitev IJPP in odjemalcem se izvede avtentikacija odjemalca. Avtentikacija odjemalca se zagotovi z odjemalčevim certifikatom, ki ga pri komunikaciji zahteva strežnik spletnih storitev IJPP. Certifikat za dostop do spletnih strani je izdan s strani IJPP institucije.

Vsaka metoda spletne storitve je sestavljena iz dveh vhodnih parametrov in izhodnega parametra (odgovora) preko katerega se vrača rezultat. Vhodna parametra sta tipa **wsMessageContext** in tipa **ime metode +Request**. Vhodni parameter tipa **ime metode +Request** vsebuje parametre, ki so deklarirani pri metodah kot vhodni parametri. Izhodni parameter je tipa **ime metode +Response** in vsebuje parametre, ki so deklarirani pri metodah kot izhodni parametri. Privzeta kodna tabela metod spletne storitve IJPP je UTF-8.

## 7.1 IDENTIFYTRANSACTIONPARAMETERS

Metodo `IdentifyTransactionParameters` uporabljamo za ugotavljanje trenutnega stanja podatkov zalednega sistema, potrebnih za pravilno delovanje terminalov v transakcijskem sistemu IJPP. V zalednem sistemu so verzije podatkov ločene z identifikatorji, kjer vsako verzijo podatkovne strukture ki jo potrebuje terminal določa unikatni identifikator.

Terminali s periodičnim klicanjem metode `IdentifyTransactionParameters` pridobivajo identifikatorje zadnjih verzij tarif, črnih list, izvršnih list, voznih redov in linij ter seznama tarifnih razredov. Ti identifikatorji se hranijo na terminalih, zato lahko terminali le ob spremembi posameznega indikatorja, izvedejo prenos nove verzije podatkov iz zalednega sistema.

Deklaracija metode	<b>Int</b> <code>IdentifyTransactionParameters</code> ( <b>int</b> iTerminalId [IN], <b>int</b> iResultCode [OUT], <b>string</b> sResultDescription [OUT], <b>IdentifyData</b> mIdentifyData [OUT])		
Parameter	Tip parametra	Obveznost parametra / Vhodni -Izhodni	Opis parametra
iTerminalId	<b>Integer</b>	Yes / Input	Identifikator terminal.
iResultCode	<b>Integer</b>	- / Output	Tip kode odgovora.
sResultDescription	<b>String</b>	- / Output	Opis tipa kode odgovora.
mIdentifyData	<b>IdentifyData*</b> , <b>NULL</b>	- / Output	Struktura s podatki identifikatorjev.

\* Struktura **IdentifyData** je definirana v poglavju definicije podatkovnih struktur.

Možne kode rezultata za metodo **IdentifyTransactionParameters**.

iResultCode	sResultDescription	Extra description
-501	Terminal not found	Terminal ni bil najden v sistemu.
0	OK	

## 7.2 GETBLACKLIST

Metodo `GetBlacklist` uporabljamo za prenos seznama črne liste iz zalednega sistema na terminale. V primeru, da potnik želi izvesti transakcijo s kartico na črni listi, jo mora terminal blokirati. Metoda deluje v dveh načinih: običajni način in inkrementalni način. Pri običajnem načinu izhodni podatki vsebujejo vse kartice na črni listi. Pri inkrementalnem načinu so v izhodnih podatkih kartice, ki so spremenile status, oziroma so bile na novo dodane ali odvzete iz črne liste. Inkrementalni način uporabljamo, kadar želimo prihraniti pri velikosti seznama kartic, ki se prenaša na terminale. V inkrementalnem načinu mora terminal sam ugotoviti katere kartice so bile dodane oziroma odvzete iz črne liste s primerjavo posamezne kartice z seznamom zadnje verzije črne liste, ki jo vsebuje terminal. Če posamezna kartica iz inkrementalnega seznama obstaja na zadnji verziji črne liste, jo je potrebno odvzeti. V kolikor posamezne kartice iz inkrementalnega seznama ni na črni listi, jo je potrebno dodati.

Pri inkrementalnem načinu je pomemben vhodni parameter LastBlackListId, ki pove od katere verzije črne liste naprej (zadnja verzija črne liste na terminalu) se upoštevajo dodane ali odvzete kartice. Za preverjanje pravilnega delovanja se na terminale pošilja izhodni parameter iBlackListCRC. Terminal po dodajanju in odstranitvi kartic na inkrementalni listi izračuna CRC na končni sestavljeni črni listi, kjer so kartice razvrščene po velikosti vrednosti ICuid od najmanjše do največje. Če je izračunani CRC enak poslanemu (iBlackListCRC), pomeni, da ima terminal v novem sestavljenem seznamu identične vrednosti ICuid, kot na črni listi v zalednem sistemu. Če parameter LastBlackListId ni prisoten (NULL), metoda deluje v običajnem načinu, sicer v inkrementalnem načinu prenosa črne liste.

Zarven naštetih parametrov lahko uporabimo še opcijske vhodne parametre iReceiveBlockSize, iReceiveBlockNumber in iReceiveBlockListId, kadar želimo omejiti velikost prejetih podatkov in podatke črne liste prejeti v več delih (blokih) z večkratnimi klici metode GetBlackist. In sicer s parametrom iReceiveBlockSize določimo največjo dovoljeno velikost paketa prejetih podatkov. Parameter iReceiveBlockNumber določa zaporedno številko paketa prejetih podatkov. iReceiveBlockListId določa identifikator trenutne črne liste, ki se prejema po delih. Kadar so prisotni vhodni parametri iReceiveBlockSize, iReceiveBlockNumber in iReceiveBlockListId, mora terminal potencialno večkrat poklicati metodo GetBlackist za prenos vseh podatkov iz zalednega sistema. Pri prvem klicu terminal pošlje vrednost parametra iReceiveBlockNumber enako 1 in iReceiveBlockListId enako 0. Pri tem prejme število paketov s parametrom iReceiveBlocks. Terminal nato izvede nadaljnje klice metode GetBlackist (če je parameter iReceiveBlocks večji od 1) tako, da nastavi naslednjo vrednost parametra iReceiveBlockNumber in vrednost parametra iReceiveBlockListId enako prejeti vrednosti iz strukture mBlackListDat.iBlacklistId iz prvega klica. Število vseh klicev metode GetBlackist, za prenos celotne izvršne liste, je enako vrednosti parametra iReceiveBlocks.

Deklaracija metode		Int GetBlackList(int iTerminalId [IN], int iLastBlackListId [IN], int iResultCode [OUT], string sResultDescription [OUT], BlackList mBlackListData [OUT])	
Parameter	Tip parametra	Obveznost parametra / Vhodni -Izhodni	Opis parametra
iTerminalId	Integer	Yes / Input	Identifikator terminal.
iLastBlackListId	Integer	No / Input	Identifikator zadnje verzije črne liste na terminalu.
iReceiveBlockSize	Integer	No / Input	Velikost podatkovnega bloka, kadar želi terminal omejiti število prejetih zapisov v polju elementov črne liste.
iReceiveBlockNumber	Integer	No / Input	Zaporedna številka podatkovnega bloka (pri velikosti bloka iReceiveBlockSize), kadar želi terminal omejiti število prejetih zapisov v polju elementov črne liste.
iReceiveBlockListId	Integer	No / Input	Identifikator črne liste, katere podatki se prejemajo po delih.

iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.
iReceiveBlocks	Integer	- / Output	Število podatkovnih blokov potrebnih za prenos celotnega polja črne liste.
mBlackListData	BlackList*, NULL	- / Output	Struktura s podatki črne liste.

\* Struktura **BlackList** je definirana v poglavju definicije podatkovnih struktur.

Možne kode rezultata za metodo **GetBlackList**.

iResultCode	sResultDescription	Extra description
-501	Terminal not found	Terminal ni bil najden v sistemu
0	OK	

### 7.3 GETCARDSTATUS

Metodo GetCardStatus uporabljamo za ugotavljanje trenutnega statusa kartice. Kartica ima lahko enega izmed naslednjih statusov: (0 – neaktivna, 1 – aktivna, 3 – blokirana, 4 – uničena, 5 – odvzeta, 8 – najdena, 10 – odvzeta-uničena, 11– zastarana, 12 – zastarana blokirana).

Deklaracija metode		Int GetCardStatus(int iTerminalId [IN], long ICuid [IN], int iResultCode [OUT], string sResultDescription [OUT], int iCardStatus [OUT])	
Parameter	Tip parametra	Obveznost parametra / Vhodni -Izhodni	Opis parametra
iTerminalId	Integer	Yes / Input	Identifikator terminal.
ICuid	Long	yes / Input	Identifikator kartice.
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.



iCardStatus	Integer, NULL	- / Output	status kartice
-------------	---------------	------------	----------------

Možne kode rezultata za metodo **GetCardStatus**.

iResultCode	sResultDescription	Extra description
-713	Card not found	Kartica ni bila najdena v sistemu.
-501	Terminal not found	Terminal ni bil najden v sistemu.
0	OK	

## 7.4 SETNEWCARDSTATUS

Metodo SetNewCardStatus uporabljamo za spremembo statusa kartice. Kartica ima lahko enega izmed naslednjih statusov: (0 – neaktivna, 1 – aktivna, 3 – blokirana, 4 – uničena, 5 – odvzeta, 8 – najdena, 10 – odvzeta-uničena, 11– zastarana, 12 – zastarana blokirana).

<b>Deklaracija metode</b>	<b>Int SetNewCardStatus(int iTerminalId [IN], long ICuid [IN], int iCardStatus [IN], int iResultCode [OUT], string sResultDescription [OUT], int iOldCardStatus [OUT])</b>		
Parameter	Tip parametra	Obveznost parametra / Vhodni -Izhodni	Opis parametra
iTerminalId	Integer	Yes / Input	Identifikator terminal.
ICuid	Long	yes / Input	Identifikator kartice.
iCardStatus	Integer	yes / Input	Nov status kartice, ki ga dodamo odvisno od namena:  0=neaktivna kartica, 1=aktivna kartica, 3=blokirana kartica, 4=uničena kartica, 5=odvzeta kartica, 8=najdena kartica, 10=odvzeta in uničena kartica, 11=zastarana kartica, 12=zastarana in blokirana kartica.
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.
iOldCardStatus	Integer,NULL	- / Output	Prejšnji status kartice.

Možne kode rezultata za metodo **SetNewCardStatus**.

iResultCode	sResultDescription	Extra description
-714	Invalid card status	Napačen status kartice.
-713	Card not found	Kartica ni bila najdena v sistemu.
-501	Terminal not found	Terminal ni bil najden v sistemu.
0	OK	

## 7.5 GETEXECUTABLELIST

Metodo `GetExecutableList` uporabljamo za prenos seznama kartic na izvršni listi, na katerih je potrebno izvesti eno izmed naslednjih opravil: Prenos vozovnice na kartico, brisanje vozovnice na kartici, nastavitve novega statusa na kartici, začasno onemogočanje uporabe kartice, online preverjanje kartice, začasno onemogočanje uporabe vozovnice. Metoda ima opsijski vhodni parameter `ICuid`. Če je parameter `ICuid` prisoten, bo metoda vrnila strukturo s podatki `mExecutableData`, v kateri bodo podatki le za eno kartico (`ICuid`). Če parameter `ICuid` ni prisoten, bo metoda vrnila strukturo s podatki `mExecutableData`, v kateri bodo podatki za vse kartice. Z vhodnim parametrom `iCompactType` določimo ali želimo kompakten tip strukture `mExecutableData` ali nekompaten. (`iCompactType=0` vrne nekompatno strukturo `mExecutableData`, `iCompactType=1` vrne kompaktno strukturo `mExecutableData`). Zarven naštetih parametrov lahko uporabimo še opsijske vhodne parametre `iReceiveBlockSize`, `iReceiveBlockNumber` in `iReceiveBlockListId`, kadar želimo omejiti velikost prejetih podatkov in podatke izvršne liste prejeti v več delih (blokih) z večkratnimi klici metode `GetExecutableList`. In sicer s parametrom `iReceiveBlockSize` določimo največjo dovoljeno velikost paketa prejetih podatkov. Parameter `iReceiveBlockNumber` določa zaporedno številko paketa prejetih podatkov. `iReceiveBlockListId` določa identifikator trenutne izvršne liste, ki se prejema po delih. Kadar so prisotni vhodni parametri `iReceiveBlockSize`, `iReceiveBlockNumber` in `iReceiveBlockListId`, mora terminal potencialno večkrat poklicati metodo `GetExecutableList` za prenos vseh podatkov iz zalednega sistema. Pri prvem klicu terminal pošlje vrednost parametra `iReceiveBlockNumber` enako 1 in `iReceiveBlockListId` enako 0. Pri tem prejme število paketov s parametrom `iReceiveBlocks`. Terminal nato izvede nadaljnje klice metode `GetExecutableList` (če je parameter `iReceiveBlocks` večji od 1) tako, da nastavi naslednjo vrednost parametra `iReceiveBlockNumber` in vrednost parametra `iReceiveBlockListId` enako prejeti vrednosti `iCurrentExeListId` iz prvega klica. Število vseh klicev metode `GetExecutableList`, za prenos celotne izvršne liste, je enako vrednosti parametra `iReceiveBlocks`.

<b>Deklaracija metode</b>	<b><code>Int GetExecutableList(int iTerminalId [IN], int iCompactType [IN], long ICuid [IN], int iResultCode [OUT], string sResultDescription [OUT], ExecutableData[] mExecutableData [OUT])</code></b>		
<b>Parameter</b>	<b>Tip parametra</b>	<b>Obveznost parametra / Vhodni -Izhodni</b>	<b>Opis parametra</b>
<code>iTerminalId</code>	<code>Integer</code>	Yes / Input	Identifikator terminal.
<code>iCompactType</code>	<code>Integer</code>	Yes / Input	Tip seznama, ki ga želimo na terminalu
<code>ICuid</code>	<code>Long</code>	No/ Input	Identifikator kartice
<code>iExeListId</code>	<code>Integer</code>	Yes/Input	Identifikator zadnje prejete izvršne liste (potrebno za inkrementalni prenos izvršne liste)
<code>iReceiveBlockSize</code>	<code>Integer</code>	Yes/Input	Velikost podatkovnega bloka, kadar želi terminal omejiti število prejetih zapisov v polju struktur <code>mExecutableData</code> .

iReceiveBlockNumber	Integer	Yes/Input	Zaporedna številka podatkovnega bloka (pri velikosti bloka iReceiveBlockSize), kadar želi terminal omejiti število prejetih zapisov v polju struktur mExecutableData.
iReceiveBlockListId	Integer	Yes/Input	Identifikator izvršne liste, katere podatki se prejemajo po delih.
iReceiveBlocks	Integer	- / Output	Število podatkovnih blokov potrebnih za prenos celotnega polja struktur mExecutableData
iCurrentExeListId	Integer	- / Output	Identifikator prejete izvršne liste
iExeListCRC	Integer	- / Output	CRC prejete izvršne liste.
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.
mExecutableData	Executable Data[]*	- / Output	Polje struktur s podatki in sezname kartic.

\* Struktura **ExecutableData** je definirana v poglavju definicije podatkovnih struktur.

Možne kode rezultata za metodo **GetExecutableList**.

iResultCode	sResultDescription	Extra description
-501	Terminal not found	Terminal ni bil najden v sistemu.
0	OK	

## 7.6 GETTARIFF

Metodo GetTariff uporabljamo za prenos seznama tarif. Vhodni podatek iTerminalId nam pove za kateri terminal bo zaledni sistem generiral tarifo. Opcijsko lahko kot vhodni podatek podamo seznam želenih identifikatorjev tarif (iTariffIds), ki jih želimo prenesti na terminal. Če seznam ni podan, se na terminal pošljejo vse trenutno aktivne tarife. Seznam (iTariffIds) uporabljamo kadar se je spremenilo le

določeno število aktivnih tarif, ki jih želimo prenesti na terminal. S tem se izognemo odvečnemu prenosu nespremenjenih tarif, ki jih že imamo shranjene na terminalu.

<b>Deklaracija metode</b>	<b>Int GetTariff (int iTerminalId [IN], int[] iTariffIds[IN], int iResultCode [OUT], string sResultDescription [OUT], Tariff[] mTariffData [OUT])</b>		
<b>Parameter</b>	<b>Tip parametra</b>	<b>Obveznost parametra / Vhodni -Izhodni</b>	<b>Opis parametra</b>
iTerminalId	Integer	Yes / Input	Identifikator terminal.
iTariffIds	Integer[]	No / Input	Seznam identifikatorjev tarif
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.
mTariffData	Tariff*	- / Output	Seznam tarif, ki se uporabljajo na terminalu.

\* Struktura **Tariff** je definirana v poglavju definicije podatkovnih struktur.

Možne kode rezultata za metodo **GetTariff**.

iResultCode	sResultDescription	Extra description
-501	Terminal not found	Terminal ni bil najden v sistemu.
0	OK	

## 7.7 GETTARIFFCLASSTRANSITIONMATRIX

Metodo GetTariffClassTransitionMatrix uporabljamo za prenos seznama tarifnih razredov in pravil za prehod med tarifnimi razredi le v večconskih mestnih tarifah. Pri prehodu potnika med tarifnimi razredi moramo vedeti koliko tarifnih razredov je potnik prestopil, če se pot prične v področju A in konča v področju B.

Deklaracija metode	<b>Int GetTariffClassTransitionMatrix</b> ( <b>int</b> iTerminalId [IN], <b>int</b> iResultCode [OUT], <b>string</b> sResultDescription [OUT], <b>TariffClassTransitions</b> TariffClassTransitionData [OUT])		
Parameter	Tip parametra	Obveznost parametra / Vhodni -Izhodni	Opis parametra
iTerminalId	Integer	Yes / Input	Identifikator terminal.
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.
mTariffClassTransitionData	TariffClassTransitions*	- / Output	Struktura s podatki o prehodih med tarifnimi razredi.

\* Struktura **TariffClassTransitions** je definirana v poglavju definicije podatkovnih struktur.

Možne kode rezultata za metodo **GetTariffClassTransitionMatrix**.

iResultCode	sResultDescription	Extra description
-501	Terminal not found	Terminal ni bil najden v sistemu.
0	OK	

## 7.8 PROCESSTRANSACTIONS

Metodo ProcessTransactions uporabljamo za prenos uspešno izvedenih transakcij o nakupih, preverjanjih in uporabi vozovnic na terminalih do procesnega centra zalednega sistema. Te transakcije imenujemo tudi IJPP Transakcije. Služijo nam za generiranje poročil za poravnavo. Zaradi tega je pomembno, da so te transakcije:

- Atomične – vsaka uspešno izvedena transakcija na terminalu mora biti zavedena kot uspešna tudi v zalednem sistemu. Transakcije izvedene na terminalih morajo biti prenesene v čim krajšem času na zaledni sistem, kjer so procesirane za izdelavo poročil na podlagi katerih izvajamo poravnavo.
- Konsistentne – transakcijski podatki morajo ustrezati pravilom za izvedbo transakcij tako, da vsaka transakcija izvedena na terminalih pravilno spremeni stanje podatkovne baze v zalednem sistemu.
- Izolirane – rezultati procesiranja transakcij morajo biti enaki, če transakcije procesiramo zaporedno ali hkrati v paketu po več skupaj.
- Trajne – rezultati procesiranih transakcij v podatkovni bazi so trajni – po končanem procesiranju se transakcije trajno hranijo v podatkovni bazi.

Deklaracija metode		<b>Int ProcessTransactions (IJPPTransaction[] mIJPPTransactionData [IN], int iResultCode [OUT], string sResultDescription [OUT])</b>	
Parameter	Tip parametra	Obveznost parametra / Vhodni -Izhodni	Opis parametra
iTerminalId	Integer	Yes / Input	Identifikator terminal.
mIJPPTransactionData	IJPPTransaction*[]	Yes / Input	Seznam izvedenih transakcij na terminalih. V seznamu je lahko največ 20 IJPP transakcij.
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.
mTransactionExecutionStatus	TransactionExecutionStatus*[]	- / Output	Seznam vseh prejetih transakcij z statusi potrditve ali zavrnitve.

\* Struktura **IJPPTransaction** in **TransactionExecutionStatus** je definirana v poglavju definicije podatkovnih struktur.

Možne kode rezultata za metodo **ProcessTransactions**.

iResultCode	sResultDescription	Extra description
0	OK	transakcije so bile sprejete uspešno
-501	Invalid date parameter	transakcije niso bile sprejete zaradi nepravilnega datuma
-719	Invalid transaction type parameter	transakcije niso bile sprejete zaradi nepravilnega tipa transakcije
-732	Missing input data	transakcije niso bile sprejete zaradi manjkajočih podatkov
-733	Not all transactions were successfull	vse transakcije niso bile sprejete
-721	Invalid input data	transakcije niso bile sprejete zaradi nepravilnih parametrov

## 7.9 GETIJPPDATEIME

Metodo GetIJPPDateTime uporabljamo za sinhronizacijo časa med sistemom IJPP in odjemalci.

Deklaracija metode		<b>Int</b> GetIJPPDateTime( <b>int</b> iTerminalId [IN], <b>int</b> iResultCode [OUT], <b>string</b> sResultDescription [OUT], <b>DateTime</b> dtIJPPDateTime [OUT])	
Parameter	Tip parametra	Obveznost parametra / Vhodni -Izhodni	Opis parametra
iTerminalId	<b>Integer</b>	Yes / Input	Identifikator terminal.
iResultCode	<b>Integer</b>	- / Output	Tip kode odgovora.
sResultDescription	<b>String</b>	- / Output	Opis tipa kode odgovora.
dtIJPPDateTime	<b>DateTime, NULL</b>	- / Output	Čas sistema IJPP

Možne kode rezultata za metodo **GetIJPPDateTime**.

iResultCode	sResultDescription	Extra description
-501	Terminal not found	Terminal ni bil najden v sistemu.
0	OK	



## 7.10 GETIJPDATETIME\_PLADENJ

Metodo GetIJPDateime\_Pladenj je identična metodi GetIJPDateime, sicer se uporablja za potrebe **Pladenj**.

<b>Deklaracija metode</b>	<b>Int GetIJPDateime_Pladenj (int iTerminalId [IN], int iResultCode [OUT], string sResultDescription [OUT], DateTime dtIJPDateime [OUT])</b>		
Parameter	Tip parametra	Obveznost parametra / Vhodni -Izhodni	Opis parametra
iTerminalId	Integer	Yes / Input	Identifikator terminal.
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.
dtIJPDateime	DateTime, NULL	- / Output	Čas sistema IJP

Možne kode rezultata za metodo **GetIJPDateime\_Pladenj**.

iResultCode	sResultDescription	Extra description
-501	Terminal not found	Terminal ni bil najden v sistemu.
0	OK	

## 7.11 DEFINICIJE PODATKOVNIH STRUKTUR

**wsMessageContext** struktura vsebuje naslednje parametre:

struktura wsMessageContext		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iRequestId	long	Identifikator poslani zahteve, mora biti edinstven (obvezen parameter).
sSenderId	String	Identifikator pošiljatelja zahteve (Id terminala iz katerega je zahteva poslana - obvezen parameter).
sSenderPass	String	Geslo pošiljatelja zahteve. (opcijski parameter)

**IdentifyData** struktura vsebuje naslednje parametre:

struktura IdentifyData		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
mTariffdef	TariffDef[]	Seznam identifikatorjev zadnjih verzij tarif.
iBlacklistId	Integer	Identifikator zadnje verzije črne liste.
iExeListId	Integer	Identifikator zadnje verzije izvršne liste.
iLineListId	Integer	Identifikator zadnje verzije seznama voznih redov in linij.
iBusOperatorId	Integer	Identifikator prevoznika.
iServiceCallInterva	Integer	Časovni interval v sekundah med servisnima klicema.
iExeListInterval	Integer	Časovni interval v sekundah med klicema izvršne liste.
mTransitionMatrix	TaransitionMatrix[]	Identifikator verzije podatkov o prehodih med področji.
bTerminalEnabled	Bool	Stanje terminala true=omogočen, false je onemogočen.
iTerminalTypeMask	Integer	Bitna maska, ki pripada (pod)tipu terminala
dtCurrentTime	datetime	Točen čas iz podatkovne baze.
iSamplingPeriod	Integer	V zalednem sistemu nastavljen podatek o periodi vzorčenja, privzeto vsakih 20 s.
iSyncPeriod	Integer	V zalednem sistemu nastavljen podatek o periodi pošiljanja podatkov v zaledni sistem, privzeto vsakih 20 s.

**TariffDef** struktura vsebuje naslednje parametre:

struktura Tariff		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iTariffId	Integer	Identifikator tarife
iTariffLocationId	Integer	Identifikator lokacije uporabe tarife

**TransitionMatrix** struktura vsebuje naslednje parametre:

struktura Tariff		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iTariffClassTransitionsId	Integer	Identifikator verzije podatkov o prehodih med področji.
iTariffLocationId	Integer	Identifikator lokacije uporabe tarife

**BlackList** struktura vsebuje naslednje parametre:

struktura BlackList		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
ICuid	Long[]	CUID seznam kartic.
iBlacklistId	Integer	Identifikator zadnje verzije črne liste na zalednem sistemu.
iBlackListCRC	Integer	CRC izračunan nad seznamom črne liste.

**ExecutableData** struktura vsebuje naslednje parametre:

struktura ExecutableData		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iAction	Integer	Oznaka operacije, ki se izvede na kartici (0=online preverjanje kartice, 1=Prenos vozovnice na kartico, 2=brisanje vozovnice na kartici, 3=nastavitev novega statusa na kartici, 4=začasno onemogočanje uporabe kartice, 5=onemogočanje uporabe vozovnice, 6=omogočanje uporabe vozovnice, 7=posodobitev relacije, 8=posodobitev vloge.
mAdditionalCardObject	ExeCardObject*[], NULL	Dodaten objekt ali seznam objektov, ki vsebuje parametre potrebne za izvedbo operacije.

\*ExeCardObject je abstractni razred, ki lahko vsebuje naslednje podrazrede(strukture ali osnovne tipe):

- CUID(Long) – Identifikator kartice na kateri je potrebno izvesti operacijo
- CardProduct – struktura s podatki o vozovnici poslani na izvršno listo(podrobno)
- CardProductShort – struktura s podatki o vozovnici poslani na izvršno listo
- CardPassengerStatus – struktura s podatki o statusu potnika

- TemporaryCUID – struktura s podatki o začasno blokirani kartici
- CardRelation – struktura s podatki o relaciji
- SubsidyApplication – struktura s podatki o vlogi za subvencionirane vozovnice

Tipi uporabljenih razredov oz osnovnih tipov v razredu ExeCardObject glede na tip akcije:

- 0 - online preverjanje kartice (na terminalu imamo kompaktno izvršno listo) -> CUID
- 1 - prenos vozovnice na kartico -> CardProduct
- 2 - brisanje vozovnice na kartici -> CardProductShort
- 3 - nastavitev novega statusa na kartici -> CardPassengerStatus
- 4 - začasno onemogočanje uporabe kartice -> TemporaryCUID
- 5 – blokiranje (onemogočanje) uporabe vozovnice -> CardProductShort
- 6 – odblokiranje (omogočanje) uporabe vozovnice ->CardProductShort
- 7 – posodobitev relacije na kartici -> CardRelation.
- 8 – posodobitev vloge za subvencionirane vozovnice na kartici ->SubsidyApplication

Struktura SubsidyApplication je definirana v poglavju administracijskih metod spletnih storitev IJPP.

**CardProduct** struktura vsebuje naslednje parametre:

struktura CardProduct		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
ICUID	Long	Identifikator kartice.
bAction	Bool	true = dodajanje vozovnice na kartico, false = odvzem vozovnice iz kartice
lTransactionExecutionId	Long	Identifikator transakcije določen v zalednem sistemu.
iTariffProductId	Integer	Identifikator vrste vozovnice.
iTariffStatusId	Integer	Identifikator statusa vozovnice.
iTariffClassId	Integer	Identifikator tarifnega razreda vozovnice.
iTariffLocationID	Integer	Identifikator lokacije tarife.
dtStartDate	DateTime	Začetni čas veljavnosti vozovnice.
dtEndDate	DateTime	Končni čas veljavnosti vozovnice.
iNoOfTickets	Integer	Število vozovnic.
iNoOfTicketsInterval	Integer, Null	Časovni interval nastavitve števila vozovnic

fPriceValue	Float	Znesek za plačilo vozovnice.
iTariffId	Integer	Identifikator tarife vozovnice.
iTariffType	Integer	tarifni način (mestna, medkrajevna,..)
iLineId	Integer, Null	ID linije.
dtActivationEnd	DateTime, Null	Končni čas aktivacije vozovnice. Če je vozovnica aktivirana, je vrednost dtActivationEnd večja od vrednosti dtStartDate. Če vozovnica ni aktivirana je vrednost dtActivationEnd enaka vrednosti dtStartDate.
mRelationData	CardRelation, Null	Struktura s podatki o relaciji na kartici (v primeru če je medkrajevni).
iCardRelationIndex	Integer, Null	Lokacija na kartici, kjer je zapisana relacija vozovnice (lokacije označujemo od indeksa 0 naprej: pri index=0 je relacija zapisana na prvem mestu, pri index=1 je relacija zapisana na drugem mestu, itd...).

**CardRelation** struktura vsebuje naslednje parametre:

struktura CardRelation		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
ICUID	Long	Identifikator kartice.
iStartStationId	Integer	Identifikator vstopne postaje.
iEndStationId	Integer	Identifikator izstopne postaje.
sStartStationName	String	Ime vstopne postaje
sEndStationName	String	Ime izstopne postaje
iStartStationLocationId	Integer	Identifikator lokacije vstopne postaje
iEndStationLocationId	Integer	Identifikator lokacije izstopne postaje
dtValidFrom	DateTime	Od kdaj relacija velja.
dtValidTo	DateTime	Do kdaj relacija velja.
iDistance	Integer	Razdalja relacije v metrih.
iNumberOfStations	Integer	Število vseh postaj ki definirajo linijo.
iStID	Integer[]	Identifikatorji postaj na katerih lahko potnik vstopa / izstopa/prestopa.
iOneWay	Integer	Identifikator ali gre za enosmerno ali dvosmerno relacijo.
iSubsidy	Integer	Identifikator ali gre za subvencionirano relacijo.
iValidStatus	Integer	Status veljavnosti vozovnice(1 = velja za vedno, 2 = velja začasno, 4 = neveljavno)
iFormatIdentifier	Integer	Identifikacija strukture zapisa na kartici.  0x01: relacija zapisana na kartici  0x02: virtualna relacija, ki je ne zapišemo na kartico in lahko pripada le medkrajevnomu produktu, ki je neomejeno veljaven na vseh relacijah.
iLineDefinition	Integer	Definicija načina zapisa posamezne linije na kartici.
iLineId	Integer	ID Linije.
iCardRelationIndex	Integer	Lokacija na kartici, kjer je zapisana relacija vozovnice (lokacije označujemo od indeksa 0 naprej: pri index=0 je relacija zapisana na prvem mestu, pri index=1 je relacija zapisana na drugem mestu, itd...).

ITransactionExecutionId	Long	Identifikator transakcije določen v zalednem sistemu
-------------------------	------	--

**CardProductShort** struktura vsebuje naslednje parametre:

struktura CardProductShort		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
ICUID	Long	Identifikator kartice.
ITransactionExecutionId	Long	Identifikator transakcije določen v zalednem sistemu
dtStartDate	DateTime	Začetni čas onemogočene vozovnice
dtEndDate	DateTime	Končni čas onemogočene vozovnice

**TemporaryCUID** struktura vsebuje naslednje parametre:

struktura TemporaryCUID		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
ICUID	Long	Identifikator kartice.
ITransactionExecutionId	Long	Identifikator transakcije določen v zalednem sistemu
dtStartDate	DateTime	Začetni čas onemogočene kartice
dtEndDate	DateTime	Končni čas onemogočene kartice

**CardPassengerStatus** struktura vsebuje naslednje parametre:

struktura CardPassengerStatus		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
ICUID	Long	Identifikator kartice.
ITransactionExecutionId	Long, NULL	Identifikator transakcije določen v zalednem sistemu
iTicketStatusId	Integer	identifikator statusa vozovnice
dtTicketStatusStartDate	DateTime	začetni čas veljavnosti vozovnice.
dtTicketStatusEndDate	DateTime	končni čas veljavnosti vozovnice.
iTicketStatusFormat	Integer	format statusa vozovnice

**Tariff** struktura vsebuje naslednje parametre:

struktura Tariff		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iTariffFormat	Integer	format tarife
iServiceProviderId	Integer	identifikator ponudnika storitev
sServiceProviderName	String	Ime ponudnika storitev
mServiceProviderTariffData	ServiceProviderTariff[]	seznam podatkov o tarifah
mTariffLocationZones	TariffLocationZones[]	Seznam povezav med lokacijami uporabe in conami

**TariffLocationSubLocations** struktura vsebuje naslednje parametre:

struktura TariffLocationSubLocations		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iLocationId	Integer	Identifikator lokacije uporabe
iSubLocationId	Integer, NULL	identifikator podrejene lokacije uporabe



**ServiceProviderTariff** struktura vsebuje naslednje parametre:

struktura ServiceProviderTariff		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iTariffId	Integer	identifikator tarife
iTariffLocationId	Integer	Identifikator lokacije uporabe
sTariffLocationName	String	Ime lokacije uporabe
iTariffType	Integer	tip tarife
dtStartTariffTime	DateTime	začetni čas veljavnosti tarife
dtEndTariffTime	DateTime	končni čas veljavnosti tarife
iCurrencyCode	Integer	Valuta za plačilo (EUR = 978)
mTariffStatusData	TariffStatus[]	seznam statusov vozovnic v tarifi
mTariffProductData	TariffProduct[]	seznam vrste vozovnic
mTariffClassData	TariffClass[]	seznam tarifnih razredov vozovnic
mTariffPriceData	TariffPrice[]	cenik vozovnic

**TariffStatus** struktura vsebuje naslednje parametre:

struktura TariffStatus		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iTariffStatusId	Integer	identifikator statusa potnika
sTariffStatusName	String	ime statusa potnika
iTariffStatusType	Integer	tip statusa potnika

**TariffProduct** struktura vsebuje naslednje parametre:

struktura TariffProduct		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iTariffProductId	Integer	identifikator vrste vozovnice
sTariffProductName	String	ime vrste vozovnice
iProductType	Integer	tip vrste vozovnice

blsConditional	bool	Določa ali je vozovnico mogoče kupiti le pod pogojem, da je bila predhodno kupljena neka druga vozovnica, ki je definirana v strukturi ConditionalProduct (nakup vozovnice je brezpogojen = false; nakup vozovnice je pogojen = true)
ilInactiveStartPeriod	Integer	pričetek neaktivnega časa veljavnosti vozovnice
ilInactiveTimeUnit	Integer	Časovna enota neaktivnega časa veljavnosti vozovnice
ilInactiveTimeQuantity	Integer	število časovnih enot neaktivnega časa veljavnosti vozovnice
ilInactiveOffsetTimeUnit	Integer	časovna enota zamika začetnega in končnega neaktivnega časa veljavnosti vozovnice
ilInactiveOffsetStartTime	Integer	zamik začetnega neaktivnega časa veljavnosti vozovnice
ilInactiveOffsetEndTime	Integer	zamik končnega neaktivnega časa veljavnosti vozovnice
iActiveStartPeriod	Integer	pričetek aktivnega časa veljavnosti vozovnice
iActiveTimeUnit	Integer	Časovna enota aktivnega časa veljavnosti vozovnice
iActiveTimeQuantity	Integer	število časovnih enot aktivnega časa veljavnosti vozovnice
iActiveTimeOffsetTimeUnit	Integer	časovna enota zamika začetnega in končnega aktivnega časa veljavnosti vozovnice
iActiveTimeOffsetStartTime	Integer	zamik začetnega aktivnega časa veljavnosti vozovnice
iActiveTimeOffsetEndTime	Integer	zamik končnega aktivnega časa veljavnosti vozovnice
iNumberOfProducts	Integer	število koriščenj vozovnice je za kuponske vozovnice večje od 1. Za ostale vozovnice je ta vrednost privzeto enaka 1.

iNumberOfProductsSetPeriod	Integer, NULL	Perioda ponastavitve števila koriščenj vozovnic
iNumberOfProductsSetExtension Unit	Integer, Null	časovna enota podaljšanja ponastavitve vozovnic
iNumberOfProductsSetExtension Quantity	Integer, Null	število časovnih enot podaljšanja ponastavitve vozovnic
iDisablePurchaseTerminalType	Integer	Omejitev nakupa glede na tip terminala
iProdcutCardLimit	Integer	največje dovoljeno število vozovnic zapisanih na kartico
fVAT	Float	Davek na dodano vrednost
blsSwitching	Bool	Identifikator prestopne vozovnice
ilInactiveValidationPriority	Integer	Prioriteta validacije neaktivne vozovnice
iActiveValidationPriority	Integer	Prioriteta validacije aktivne vozovnice
iProductTimePriority	Integer	Prioriteta validacije vozovnice
bAbsoluteValidationTimeValidity	Byte[], NULL	absolutni dovoljeni čas uporabe vozovnice
iPurchaseStartOffsetTimeUnit	Integer	časovna enota zamika, ki določa začetni čas za nakup vozovnice
iPurchaseStartOffset	Integer	časovni zamik začetnega časa za nakup vozovnice
iPurchaseEndOffsetTimeUnit	Integer	časovna enota zamika, ki določa končni čas za nakup vozovnice
iPurchaseEndOffset	Integer	časovni zamik končnega časa za nakup vozovnice
iDirection	Integer, NULL	Smer vozovnice (NULL, 0=dvosmerna. 1=enosmerna v smeri relacije)
bAbsolutePurchaseTimeValidity	Byte[], NULL	absolutni dovoljeni čas nakupa vozovnice
mConditionalProductData	ConditionalProduct [], NULL	Struktura za določanje pogojnih vozovnic v primeru, ko imamo blsConditional=true.

**ConditionalProduct** struktura vsebuje naslednje parametre:

struktura ConditionalProduct		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iTariffLocationId	Integer	Identifikator lokacije uporabe
iTariffProductId	Integer	identifikator vrste vozovnice
iTariffClassId	Integer	identifikator tarifnega razreda

**TariffClass** struktura vsebuje naslednje parametre:

struktura TariffClass		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iTariffClassId	Integer	identifikator tarifnega razreda
sTariffClassName	String	ime tarifnega razreda
iPriority	Integer	prioriteta (vrstni red) tarifnega razreda
iKilometersFrom	Integer, NULL	kilometri od (spodnja meja v kilometrih je vključena v tarifni razred – v tarifni razred spadajo kilometri večji ali enaki iKilometersFrom)
iKilometersTo	Integer, NULL	kilometri do (zgornja meja v kilometrih ni vključena v tarifni razred – v tarifni razred spadajo kilometri manjši od iKilometersTo)
iValidityMinutes	Integer, NULL	Čas veljavnosti vozovnice v minutah, ki je odvisna od tarifnega razreda

**TariffPrice** struktura vsebuje naslednje parametre:

struktura TariffPrice		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iTariffStatusId	Integer	identifikator statusa vozovnice
iTariffProductId	Integer	identifikator vrste vozovnice
iTariffClassId	Integer	identifikator tarifnega razreda
fPriceValue	Float	cena vozovnice

**TariffClassTransitions** struktura vsebuje naslednje parametre:

struktura TariffClassTransitions		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
mLocationData	Location[]	seznam podatkov o področjih
mLocationTransitionsData	LocationTransitions[]	seznam podatkov o prehodih med področji.

**Location** struktura vsebuje naslednje parametre:

struktura Location		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iLocationId	Integer	identifikator področja
sLocationName	String	ime področja

**LocationTransition** struktura vsebuje naslednje parametre:

struktura LocationTransitionmatrix		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iTariffLocationId	Integer	Identifikator lokacije uporabe tarife
iTariffClassTransitionsId	Integer	Identifikator verzije podatkov o prehodih med področji.
mLocationTransitionMatrixData	LocationTransitionsMatrix	seznam podatkov o prehodih med področji.

**LocationTransitionMatrix** struktura vsebuje naslednje parametre:

struktura LocationTransitionmatrix		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iLocationIdFrom	Integer	identifikator vstopnega področja;
iLocationIdTo	Integer	identifikator izstopnega področja;
iNumberOfTransitions	Integer	število prehodov tarifnih razredov med vstopnim in izstopnim področjem.

**IJPPTransaction** struktura vsebuje naslednje parametre:

struktura IJPPTransaction		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
ITransactionExecutionId	Long	Identifikator transakcije
iTransactionType	Integer	tip transakcije (natančneje opisano v poglavju: Tipi IJPP transakcij)
dtStartTime	DateTime	čas izvedbe transakcije na terminalu
ICUID	Long, NULL	identifikator kartice
dtLastSuccessfulTime	DateTime, NULL	Čas zadnje uspešne transakcije na kartici.
ILastSuccessfulTransactionExecutionId	Long, NULL	Identifikator zadnje uspešne transakcije na kartici.
iExecutionStatus	Integer, NULL	Status izvedbe transakcije na kartici. (0- uspešno izvedena transakcija, 1- nepreverjena transakcija)
sSubsidyId	String, NULL	Identifikator subvencije
iTariffProductId	Integer, NULL	identifikator vrste vozovnice
iTicketStatus	Integer, NULL	Identifikator statusa (pravica)
iTariffClassId	Integer, NULL	identifikator tarifnega razreda
fPriceValue	Float, NULL	cena vozovnice
iTariffId	Integer, NULL	identifikator tarife
iTariffLocationId	Integer, NULL	Identifikator lokacije uporabe
iTariffClassTransitionsId	Integer, NULL	Identifikator verzije podatkov o prehodih med področji
iTerminalId	Integer, NULL	identifikator terminala
iPosId	Integer, NULL	identifikator prodajnega mesta
iCurrency	Integer, NULL	valuta (978=EUR)
iStationId	Integer, NULL	Identifikator vstopne postaje
iExitStationId	Integer, NULL	Identifikator izstopne postaje
iTripId	Integer, NULL	identifikator vožnje
iPaymentType	Integer, NULL	vrsta plačila
iBlacklistId	Integer, NULL	id aktivne črne liste v času izvedbe transakcije

iExeListId	Integer, NULL	id aktivne izvršne liste v času izvedbe transakcije
iBusId	Integer, NULL	identifikator vozila
iLineId	Integer, NULL	Identifikator linije
iTripId	Integer, NULL	Identifikator vožnje
ICashierId	Long, NULL	identifikator blagajnika
sCashierName	String, NULL	Ime blagajnika
dtValidFrom	DateTime, NULL	začetni čas veljavnosti vozovnice
dtValidTo	DateTime, NULL	končni čas veljavnosti vozovnice
iNumberOfProducts	Integer, NULL	število produktov (kuponov)
iNumberOfProductsSetInterval	Integer, NULL	časovni interval nastavitve števila produktov
IRceiptNumber	Long, NULL	številka računa
iTicketStatusFormat	Integer, NULL	format statusa vozovnice
iZoneLocation	Integer, NULL	identifikator področja postaje
dGPSLongitude	Decimal, NULL	geografski položaj transakcije
dGPSLatitude	Decimal, NULL	geografski položaj transakcije
IRefTransactionExecutionId	Long, NULL	referenčni identifikator transakcije
iErrorCode	Integer, NULL	Koda napake (neustrezen format kartice=1, kartica nima ustrezne vozovnice=2, kartica je začasno onemogočena na izvršni listi=3, vozovnica je začasno onemogočena na izvršni listi = 4)
IRefCUID	Long, NULL	identifikator referenčne (druge) kartice
iOfflineStatus	Integer, NULL	Status, ki ga določi zaledni sistem pri procesiranju IJPP transakcije. To je izhodni podatek. Kot vhodni podatek se ta vrednost pobriše.
dtExecutionTime	DateTime, NULL	Čas procesiranja transakcije na zalednem sistemu. Ta čas služi za razvrščanje transakcij v časovna obdobje pri poročilih za poravnavo.
mCardRelationData	CardRelation, NULL	Podatki o relaciji vozovnice

iOffenseType	Integer, NULL	Tip prekrška (0-veljavna kartica, 1-kartica je blokirana ali na črni listi, 2-vozovnica ni aktivirana ali validirana, 3-na kartici ni veljavne vozovnice, 4-poneverjanje identitete, 5-potnik nima kartice/vozovnice)
iSanctionType	Integer, NULL	Tip sankcije (0-ni sankcije vozovnica je veljavna, 1-izrečena sankcija, kjer je bil potnik je bil sankcioniran z globo, 2-kartica je bila blokirana, 3-kartica je bila dodana na črno listo, 4-izrečeno opozorilo, 5-kartica je bila blokirana zaradi preseženega maksimalnega števila opozoril)
iOffenseCount	Integer, NULL	Skupno število prekrškov, ki so bili izrečeni lastniku kartice
iWorkOrderId	Integer, NULL	Identifikator delovne naloge
sPurchaseOrder	String, NULL	Oznaka naročilnice
fSubFullPrice	Float, NULL	Polna cena subvencioniranih vozovnic
fSubFullPriceReported	Float, NULL	Poročana cena subvencioniranih vozovnic
fSubPayment	Float, NULL	Plačilo subvencioniranih vozovnic
sNotes	String, NULL	Opombe pri IJPP transakciji

**TransactionExecutionStatus** struktura vsebuje naslednje parametre:

struktura TransactionExecutionStatus		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
TransactionExecutionID	Long	Identifikator transakcije.
Status	Integer	Status transakcije pri obdelavi . 0=OK -1=transakcija zavrnjena (Potrebno <b>ponovno</b> pošiljanje ) -2=TransactionExecutionID že obstaja ( <b>Ni</b> potrebno ponovno pošiljanje ) -3= Transakcija zavrnjena zaradi nepravilnih podatkov. ( <b>Ni</b> potrebno ponovno pošiljanje – te vrste napak se odpravijo v času razvoja)



Description	String	Opis statusa transakcije pri obdelavi.
-------------	--------	--

## 7.12 TIPI IJPP TRANSAKCIJ

Zaledni sistem IJPP preko web servisa (WS) z metodo ProcessTransactions sprejema podatke o izvedenih transakcijah potnikov na terminalih, ki jih imenujemo IJPP transakcije (oziroma samo transakcije), ki jih ločujemo po tipu. Tipi transakcij, ki se izvajajo na terminalih so: nakup vozovnice, validacija vozovnice, stornacija nakupa vozovnice, blokiranje kartice, online prenos vozovnice na kartico, online brisanje vozovnice na kartici, online nastavitev novega statusa na kartici in zavrnitev transakcije s kartico. Posamezna transakcija je definirana s podatki v strukturi IJPPTransaction, kjer je tip transakcije določen z unikatno vrednostjo **iTransactionType**.

Vsaka transakcija vsebuje 8 bajtni unikatni identifikator **ITransactionExecutionId**, ki ga terminal generira pri izvedbi transakcije po sledečem principu:

- Vrednost ITransactionExecutionId sestavlja 8 bajtov:  
byte7:byte6:byte5:byte4:byte3:byte2:byte1:byte0, kjer je byte7 najbolj uteženi bajt in byte0 najmanj uteženi bajt.
- Bajta byte7 in byte6 sta vedno enaka 0.
- Naslednji trije bajti byte5:byte4:byte3 predstavljajo identifikator terminala **iTerminalId** na katerem je bila izvedena transakcija.
- Spodnji trije bajti byte2:byte1:byte0 predstavljajo zaporedno številko transakcije izvedene na terminalu. Terminal si sproti beleži število izvedenih transakcij, kjer se prva transakcija prične s številko 0.

Podatki potrebni za posamezen tip IJPP transakcije so podrobneje opisani v naslednjih podpoglavjih.

#### *7.12.1 NAKUP VOZOVNICE*

Pri vsakem nakupu vozovnice se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri nakupu vozovnice so v strukturi IJPPTransaction obvezni naslednji podatki:**

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 1.

**dtStartTime** = čas nakupa vozovnice (čas zapisa vozovnice na kartico).

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**fPriceValue** = cena vozovnice.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iCurrency** = valuta zneska (978=EUR).

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**dtValidFrom** = začetni neaktivni čas veljavnosti vozovnice.

**dtValidTo** = končni neaktivni čas veljavnosti vozovnice.

**iNumberOfProducts** = število produktov (kuponov), določa koliko krat lahko potnik aktivira vozovnico.

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**sCashierName** = Ime blagajnika.

**Pri nakupu vozovnice so v strukturi IJPPTtransaction pogojno obvezni naslednji podatki:**

**ICUID** = identifikator kartice je obvezen podatek le, če je vozovnica zapisana na kartici.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici je obvezen podatek le, če je vozovnica zapisana na kartici.

**lLastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici je obvezen podatek le, če je vozovnica zapisana na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija) je obvezen podatek le, če je vozovnica zapisana na kartici.

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik kupi subvencionirano vozovnico.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica je obvezen podatek v primeru, ko potnik kupi vozovnico, ki velja v več območjih.

**iStationId** = Identifikator postaje je obvezen podatek v primeru, ko ima terminal informacijo o vstopni postaji.

**iTripId** = identifikator vožnje je obvezen podatek le v primeru, ko ima terminal informacijo o vožnji.

**iExitStationId** = Identifikator izstopne postaje, je obvezen podatek le pri nakupih medkrajevnih vozovnic, kjer je na voljo informacija o izstopni postaji.

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji je obvezen podatek le pri večconskih vozovnicah.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice. Obvezen podatek pri nakupu medkrajevne vozovnice

**Pri nakupu vozovnice so v strukturi IJPPTtransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**IRefTransactionExecutionId** = referenčni identifikator transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iLineId** = identifikator linije.

**IRefCUID** = identifikator referenčne kartice

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

### *7.12.2 VALIDACIJA VOZOVNICE*

Validacije vozovnic pošiljajo le terminali validatorji. Validacija vozovnice se pošiljajo le pri prestopnih vozovnicah. Po aktivaciji vozovnice, lahko le to uporabljamo dokler je aktivna. Ob vsaki uporabi vozovnice po aktivaciji se izvede validacija vozovnice, le kadar gre za prestopno vozovnico. Podatki

validacije se prenesejo iz validatorjev v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

***Pri validaciji vozovnice so v strukturi IJPPTtransaction obvezni naslednji podatki:***

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 2.

**dtStartTime** = čas validacije vozovnice.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**fPriceValue** = cena vozovnice.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice pri nakupu.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**iPosId** = identifikator prodajnega mesta (vozila).

**lRefTransactionExecutionId** = referenčni identifikator transakcije s katero je bil izveden nakup vozovnice, ki jo validiramo.

**iNumberOfProducts** = število produktov (kuponov), določa koliko preostalih aktivacij vozovnice ima potnik še na voljo.

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**iTripId** = identifikator vožnje.

***Pri validaciji vozovnice so v strukturi IJPPTtransaction pogojno obvezni naslednji podatki:***

**ICUID** = identifikator kartice je obvezen podatek le, če je bila validirana vozovnica na kartici.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici je obvezen podatek le, če je bila validirana vozovnica na kartici.

**lLastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici je obvezen podatek le, če je bila validirana vozovnica na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija) je obvezen podatek le, če je bila validirana vozovnica na kartici.

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik kupi subvencionirano vozovnico.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica je obvezen podatek v primeru, ko potnik kupi vozovnico, ki velja v več območjih.

**iStationId** = Identifikator postaje je obvezen podatek v primeru, ko ima terminal informacijo o vstopni postaji.

**iExitStationId** = Identifikator izstopne postaje, je obvezen podatek le pri validacijah medkrajevnih vozovnic, kjer je na voljo informacija o izstopni postaji.

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji je obvezen podatek le pri večconskih vozovnicah.

**iOffenseType** = tip prekrška se pošlje kadar je validacija izvedena na postaji izven voznega reda.

**iWorkOrderId** = identifikator delovne naloge.

**Pri validaciji vozovnice so v strukturi IJPPTransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**iCurrency** = valuta zneska (978=EUR).

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iLineId** = identifikator linije.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**sCashierName** = Ime blagajnika.

**lRefCUID** = identifikator referenčne kartice.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

### *7.12.3 STORNACIJA NAKUPA VOZOVNICE*

Pri vsaki stornaciji nakupa vozovnice se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri stornaciji nakupa vozovnice so v strukturi IJPPTtransaction obvezni naslednji podatki:**

**lTransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 3.

**dtStartTime** = čas stornacije nakupa vozovnice.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**fPriceValue** = cena vozovnice. Pri stornaciji posredujemo negativno vrednost zneska, ki smo ga imeli pri nakupu .

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iCurrency** = valuta zneska (978=EUR).

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**lRefTransactionExecutionId** = identifikator transakcije s katero je bil izveden nakup vozovnice, ki ga storniramo.

**iNumberOfProducts** = število produktov (kuponov). Pri stornaciji posredujemo negativno vrednost števila kuponov, ki smo ga imeli pri nakupu.

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**sCashierName** = Ime blagajnika.

**Pri stornaciji nakupa vozovnice so v strukturi IJPPTtransaction pogojno obvezni naslednji podatki:**

**ICUID** = identifikator kartice je obvezen podatek le, če je bila stornacija izvedena na kartici.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici je obvezen podatek le, če je bila stornacija izvedena na kartici.

**lLastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici je obvezen podatek le, če je bila stornacija izvedena na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija) je obvezen podatek le, če je bila stornacija izvedena na kartici.

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik kupi subvencionirano vozovnico.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica je obvezen podatek v primeru, ko potnik kupi vozovnico, ki velja v več območjih.

**iStationId** = Identifikator postaje je obvezen podatek v primeru, ko ima terminal informacijo o vstopni postaji.

**iTripId** = identifikator vožnje je obvezen podatek le v primeru, ko ima terminal informacijo o vožnji.

**Pri stornaciji nakupa vozovnice so v strukturi IJPPTtransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**lCashierId** = identifikator blagajnika.

**lReceiptNumber** = številka računa.



**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iExitStationId** = Identifikator izstopne postaje.

**iLineId** = identifikator linije.

**lRefCUID** = identifikator referenčne kartice

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

#### *7.12.4 BLOKIRANA KARTICA*

Če je transakcija na terminalu izvedena s kartico, ki se nahaja na črni listi, mora terminal to kartico blokirati. Pri vsaki blokadi kartice se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri blokadi kartice so v strukturi IJPPTransaction obvezni naslednji podatki:**

**lTransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 4.

**dtStartTime** = čas blokade kartice.

**ICUID** = identifikator kartice.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici.

**lLastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija).

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**Pri blokadi kartice so v strukturi IJPPTtransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**fPriceValue** = cena vozovnice.

**sSubsidyId** = Identifikator subvencije.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica.

**iStationId** = Identifikator postaje.

**iTripId** = identifikator vožnje.

**iCurrency** = valuta zneska (978=EUR).

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**iNumberOfProducts** = število produktov (kuponov).

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**lRefTransactionExecutionId** = referenčni identifikator transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iExitStationId** = Identifikator izstopne postaje.

**iLineId** = identifikator linije.

**sCashierName** = Ime blagajnika.

**lRefCUID** = identifikator referenčne kartice.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

### *7.12.5 PRENOS VOZOVNICE NA KARTICO IZ IZVRŠNE LISTE*

Pri vsakem zapisu vozovnice na kartico iz izvršne liste se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Vozovnica se na kartico iz izvršne liste prenese le enkrat. Večkratni prenos vozovnice iz izvršne liste terminal prepreči tako, da pred vsakim prenosom vozovnice preveri vrednost ITransactionExecutionId zapisano v strukturi produkta na kartici. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

***Pri prenosu vozovnicena kartico so v strukturi IJPPTtransaction obvezni naslednji podatki:***

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 5.

**dtStartTime** = čas zapisa vozovnice na kartico.

**ICUID** = identifikator kartice.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici.

**lLastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcija na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija).

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**fPriceValue** = cena vozovnice.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iCurrency** = valuta zneska (978=EUR).

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**lRefTransactionExecutionId** = referenčni identifikator transakcije je določen predhodno na zalednem sistemu in je prisoten kot parameter v izvršni listi.

**iNumberOfProducts** = število produktov (kuponov), določa koliko krat lahko potnik aktivira vozovnico.

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**Pri online prenosu vozovnice so v strukturi IJPPTransaction pogojno obvezni naslednji podatki:**

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik dobi subvencionirano vozovnico.

**Pri prenosu vozovnice so v strukturi IJPPTransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica.

**iStationId** = Identifikator postaje.

**iTripId** = identifikator vožnje.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iExitStationId** = Identifikator izstopne postaje.

**iLineId** = identifikator linije.

**sCashierName** = Ime blagajnika.

**IRefCUID** = identifikator referenčne kartice.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

#### *7.12.6 BRISANJE VOZOVNICE NA KARTICI IZ IZVRŠNE LISTE*

Pri vsakem brisanju vozovnice, ki je na izvršni listi se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri brisanju vozovnice so v strukturi IJPPTtransaction obvezni naslednji podatki:**

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 6.

**dtStartTime** = čas izbrisa vozovnice na kartici.

**ICUID** = identifikator kartice.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici.

**ILastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija).

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**lRefTransactionExecutionId** = referenčni identifikator transakcije s katero je bila naložena vozovnica, ki jo brišemo iz kartice.

**Pri brisanju vozovnice so v strukturi IJPPTtransaction obvezni naslednji podatki, če so prisotni na terminalu::**

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**fPriceValue** = cena vozovnice.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iCurrency** = valuta zneska (978=EUR).

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik dobi subvencionirano vozovnico.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica.

**iStationId** = Identifikator postaje.

**iTripId** = identifikator vožnje.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**iNumberOfProducts** = število produktov (kuponov).

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iExitStationId** = Identifikator izstopne postaje.

**iLineId** = identifikator linije.

**sCashierName** = Ime blagajnika.

**lRefCUID** = identifikator referenčne kartice.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

### *7.12.7 NASTAVITEV NOVEGA STATUSA NA KARTICI IZ IZVRŠNE LISTE*

Pri vsaki nastavitvi novega statusa, ki je na izvršni listi se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri nastavitvi novega statusa so v strukturi IJPPTransaction obvezni naslednji podatki:**

**lTransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 7.



**dtStartTime** = čas nastavitve novega statusa na kartici.

**ICUID** = identifikator kartice.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici.

**lLastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija).

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iTicketStatusFormat** = format statusa vozovnice.

**lRefTransactionExecutionId** = referenčni identifikator transakcije je določen predhodno na zalednem sistemu in je prisoten kot parameter v izvršni listi.

**Pri nastavitvi novega statusa so v strukturi IJPPTtransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**fPriceValue** = cena vozovnice.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iCurrency** = valuta zneska (978=EUR).

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik dobi subvencionirano vozovnico.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica.

**iStationId** = Identifikator postaje.

**iTripId** = identifikator vožnje.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**iNumberOfProducts** = število produktov (kuponov).

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**IRceiptNumber** = številka računa.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iExitStationId** = Identifikator izstopne postaje.

**iLineId** = identifikator linije.

**sCashierName** = Ime blagajnika.

**IRefCUID** = identifikator referenčne kartice.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

### 7.12.8 ZAVRNITEV TRANSAKCIJE

Pri vsaki zavrnitvi izvedbe transakcije na kartici, se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri zavrnitvi transakcije na terminalu so v strukturi IJPPTtransaction obvezni naslednji podatki:**

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** =8.

**dtStartTime** = čas zavrnitve transakcije.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iErrorCode** = koda napake, zaradi katere je bila transakcija zavrnjena (kode napak so opisane v tabeli 7.1).

**Pri zavrnitvi transakcije na terminalu so v strukturi IJPPTtransaction pogojno obvezni naslednji podatki:**

**ICUID** = identifikator kartice je obvezen podatek, kadar je bila zavrnjena transakcija na kartici.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici je obvezen podatek, kadar je bila zavrnjena transakcija na kartici.

**ILastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici je obvezen podatek, kadar je bila zavrnjena transakcija na kartici.

**Pri zavrnitvi transakcije na terminalu so v strukturi IJPPTtransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija).

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**fPriceValue** = cena vozovnice.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iCurrency** = valuta zneska (978=EUR).

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik dobi subvencionirano vozovnico.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica.

**iStationId** = Identifikator postaje.

**iTripId** = identifikator vožnje.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**iNumberOfProducts** = število produktov (kuponov).

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**IRefTransactionExecutionId** = referenčni identifikator transakcije.

**iExitStationId** = Identifikator izstopne postaje.

**iLineId** = identifikator linije.

**sCashierName** = Ime blagajnika.

**IRefCUID** = identifikator referenčne kartice.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

TABELA 7.1: KODE NAPAK PRI ZAVRNITVI TRANSAKCIJE

Koda napake	Opis kode napake
1	Nepravilna/napačna kartica (napačen format, napačni ključi, crypto,...).
2	Kartica je potekla.
3	Kartica še ni veljavna.
4	Kartica je zaklenjena. Kartica se nahaja na črni listi.
5	Željen tarifni razred na kartici ni najden.
6	Nemogoč nakup produktov, ker je kartica polna.
7	Nemogoč nakup medkrajevnih produktov, ker je kartica polna.
8	Zahtevana UID in dejanska UID se razlikujeta.
9	Število dovoljenih kart željenega tipa je doseženo, nedevojen nakup.
10	Pri zahtevi SESSION CHECK je bila prislonjena druga kartica, kot v prvotni kartični transakciji.
11	Pri zahtevi SESSION CHECK je bilo ugotovljeno, da se podatki na kartico niso pravilno vpisali.
12	Pri zahtevi SESSION CHECK je bilo ugotovljeno, da so podatki na kartico bili pravilno vpisani.
13	S službeno kartico ni možna transakcija.
14	Drug uporabnik je še vedno prijavljen. Potrebno je izvesti odjavo.
15	Pri kartični transakciji je prišlo do napake.
16	Napačen tip službene kartice.
17	Neupešna odjava.
18	Kartica je blokirana.
19	Napaka v verziji kartice.
20	Napačen poseben status.

21	Na kartici ni statusov.
22	Napaka v tarifi.
23	Vozovnica je že validirana.
24	Napaka pri shranjevanju IJPP transakcije.
25	Tiskalnik je brez papirja.
26	Tiskalnik se pregreva.
27	Napaka na tiskalniku.
28	Ni transakcij za stornacijo.
29	Napaka na RFID.
30	Kartica ni validirana v pravilni coni ali relaciji.
31	Kartica ne vsebuje veljavne vozovnice glede na trenutni čas.
32	Kartica ni validirana na določenem vozilu ali je potekel čas.
512	RFID - komunikacijska napaka.
768	RFID - Kartica odstranjena. Ni odziva v želenem času.
1024	RFID - Napačen ukaz v tej fazi.
1280	RFID - Kartica ignorirana.
1536	RFID - Nedelaven RFID.
1792	RFID - Napaka v primeru, ko je prisotnih več kartic.
2048	RFID - RFID modul ni inicializiran, mora opraviti INIT RFID proceduro.
2304	RFID - RFID modul je trenutno zaseden, poskusi kasneje.
2560	RFID - Nekdo je poskušal preklicati kartico, vendar nobena kartica ni izbrana.
2816	RFID - Napačen tip kartice.
2828	RFID - Ni sprememb v kopiji fila
2830	RFID - Ukaz se ne more zaključiti, ker primanjkuje NV-pomnilnika.
2844	RFID - Koda ukaza ni podprta.
2846	RFID - CRC ali MAC podatki se ne ujemajo. ( Padding bytes not valid)
2880	RFID - Neveljaven ključ.
2942	RFID - Dolžina ukaza je neveljavna.
2973	RFID - Trenutna konfiguracija / status ne dovoljuje zahtevenega ukaza.
2974	RFID - Vrednost parametra je neveljavna.
2976	RFID - Zahtevan AID ni prisoten na PICC.
2977	RFID - Nepopravljiva napaka v aplikaciji. Aplikacija bo onemogočena.
2990	RFID - Trenutni status pristnosti ne dovoljuje zahtevan ukaz.
3006	RFID - Poskus branja/pisanja podatkov od/do preko datotek/meje zapisov.
3009	RFID - Nepopravljiva napaka v PICC, PICC bo onemogočen.
3018	RFID - Prejšnji ukaz ni bil zaključen.
3021	RFID - PICC je bil onemogočen z nepopravljivo napako.
3022	RFID - Število aplikacij omejeno na 28. Dodajanje novih aplikacij (» CreateApplication«) ni mogoče.
3038	RFID - Ustvarjanje datoteke/aplikacije ni mogoča, ker datoteka/aplikacija z enakim številko že obstaja.
3054	RFID - NV-Write operacija se ni zaključila zaradi izgube elektrike, notranje kopije/aktiviranega rollback mehanizma.

3056	RFID - Številka datoteke ne obstaja.
3057	RFID - Nepopravljiva napaka v file, datoteka bo onemogočena.
3072	RFID - 14443-4 in 14443-3 in DESFire napaka protokola.
3328	RFID - Vrednost parametra je neveljavna.
3584	RFID - Napačen tip kartice.
4096	RFID - Napaka SAM.

### 7.12.9 PRIJAVA VOŽNJE

Vsako izvedeno vožnjo na določeni liniji mora terminal, ki se uporablja za validiranje vozovnic na vozilu, poročati na zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri prijavi vožnje so v strukturi IJPPTtransaction obvezni naslednji podatki:**

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 9.

**dtStartTime** = čas vožnje

**iTripId** = identifikator vožnje.

**iTerminalId** = identifikator terminala, ki je nameščen v vozilu.

**iStationId** = Identifikator vstopne postaje.

**iExitStationId** = Identifikator izstopne postaje.

**Pri prijavi vožnje so v strukturi IJPPTtransaction obvezni naslednji podatki le, če so prisotni na terminalu::**

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice pri nakupu.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**ICUID** = identifikator kartice.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici.

**lLastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici.

**sSubsidyId** = Identifikator subvencije.

**iZoneLocation** = identifikator področja.

**lRefTransactionExecutionId** = referenčni identifikator transakcije.

**fPriceValue** = cena vozovnice.

**iCurrency** = valuta zneska (978=EUR).

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**lCashierId** = identifikator blagajnika.

**iNumberOfProducts** = število produktov (kuponov).

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**lReceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iLineId** = identifikator linije.

**sCashierName** = Ime blagajnika.

**lRefCUID** = identifikator referenčne kartice

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice



**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

#### *7.12.10 PRIJAVA BLAGAJNIKA*

Terminal lahko opcijsko izvede IJPP transakcijo za prijavo blagajnika ob pričetku prodaje vozovnic. Prijava se izvede le za blagajnike registrirane v IJPP sistemu. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri prijavi blagajnika so v strukturi IJPPTransaction obvezni naslednji podatki:**

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 10.

**dtStartTime** = čas prijave.

**iTerminalId** = identifikator terminala.

**Pri prijavi blagajnika so v strukturi IJPPTransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**iStationId** = Identifikator vstopne postaje.

**iExitStationId** = Identifikator izstopne postaje.

**iTripId** = identifikator vožnje.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice pri nakupu.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**ICUID** = identifikator kartice.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici.

**lLastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici.

**sSubsidyId** = Identifikator subvencije.

**iZoneLocation** = identifikator področja.

**lRefTransactionExecutionId** = referenčni identifikator transakcije.

**fPriceValue** = cena vozovnice.

**iCurrency** = valuta zneska (978=EUR).

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**iNumberOfProducts** = število produktov (kuponov).

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**lReceiptNumber** = številka računa.

**lCashierId** = identifikator blagajnika.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iLineId** = identifikator linije.

**sCashierName** = Ime blagajnika.

**IRefCUID** = identifikator referenčne kartice.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

#### **7.12.11 ODJAVA BLAGAJNIKA**

Terminal lahko opsijsko izvede IJPP transakcijo za odjavo blagajnika ob koncu prodaje vozovnic. Odjava se izvede le za blagajnike registrirane v IJPP sistemu. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri odjavi blagajnika so v strukturi IJPPTtransaction obvezni naslednji podatki:**

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 11.

**dtStartTime** = čas odjave.

**iTerminalId** = identifikator terminala.

**Pri odjavi blagajnika so v strukturi IJPPTtransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**iStationId** = Identifikator vstopne postaje.

**iExitStationId** = Identifikator izstopne postaje.

**iTripId**= identifikator vožnje.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice pri nakupu.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**ICUID** = identifikator kartice.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici.

**lLastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici.

**sSubsidyId** = Identifikator subvencije.

**iZoneLocation** = identifikator področja.

**lRefTransactionExecutionId** = referenčni identifikator transakcije.

**fPriceValue** = cena vozovnice.

**iCurrency** = valuta zneska (978=EUR).

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**iNumberOfProducts** = število produktov (kuponov).

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**lReceiptNumber** = številka računa.

**lCashierId** = identifikator blagajnika.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iLineId** = identifikator linije.

**sCashierName** = Ime blagajnika.

**lRefCUID** = identifikator referenčne kartice.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevnih vozovnic.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

### **7.12.12      AKTIVACIJA VOZOVNICE**

Po nakupu vozovnice je potrebno pri prvi uporabi vozovnico aktivirati. Pri vsaki aktivaciji vozovnice se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri aktivaciji vozovnice so v strukturi IJPPTransaction obvezni naslednji podatki:**

**lTransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 12.

**dtStartTime** = čas aktivacije vozovnice.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**fPriceValue** = cena vozovnice.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice pri nakupu.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**iPosId** = identifikator prodajnega mesta (vozila).

**dtValidFrom** = začetni čas veljavnosti aktivirane vozovnice.

**dtValidTo** = končni čas veljavnosti aktivirane vozovnice.

**lRefTransactionExecutionId** = referenčni identifikator transakcije s katero je bil izveden nakup vozovnice, ki jo aktiviramo.

**iNumberOfProducts** = število produktov (kuponov), določa koliko preostalih aktivacij vozovnice ima potnik še na voljo.

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**iTripId** = identifikator vožnje.

**Pri aktivaciji vozovnice so v strukturi IJPPTTransaction pogojno obvezni naslednji podatki:**

**ICUID** = identifikator kartice je obvezen podatek le, če je bila aktivirana vozovnica na kartici.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici je obvezen podatek le, če je bila aktivirana vozovnica na kartici.

**lLastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici je obvezen podatek le, če je bila aktivirana vozovnica na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija) je obvezen podatek le, če je bila aktivirana vozovnica na kartici.

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik kupi subvencionirano vozovnico.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica je obvezen podatek v primeru, ko potnik kupi vozovnico, ki velja v več območjih.

**iStationId** = Identifikator postaje je obvezen podatek v primeru, ko ima terminal informacijo o vstopni postaji.

**iExitStationId** = Identifikator izstopne postaje, je obvezen podatek le pri aktivacijah medkrajevnih vozovnic, kjer je na voljo informacija o izstopni postaji.

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji je obvezen podatek le pri večconskih vozovnicah.

**iOffenseType** = tip prekrška se pošlje kadar je validacija izvedena na postaji izven voznega reda.

**iWorkOrderId** = identifikator delovne naloge.

**Pri aktivaciji vozovnice so v strukturi IJPPTransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**iCurrency** = valuta zneska (978=EUR).

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iLineId** = identifikator linije.

**sCashierName** = Ime blagajnika.

**IRefCUID** = identifikator referenčne kartice.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

### **7.12.13      INICIALIZACIJA KARTICE**

Ob inicializaciji IJPP kartice se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri inicializaciji kartice so v strukturi IJPPTtransaction obvezni naslednji podatki:**

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 13.

**dtStartTime** = čas inicializacije kartice.

**ICUID** = identifikator kartice.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija).

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**Pri inicializaciji kartice so v strukturi IJPPTtransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici.

**ILastSuccessfulTransactionExecutionId** = identifikator terminala na katerem je bila izvedena zadnja uspešna transakcija na kartici.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.



**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**fPriceValue** = cena vozovnice.

**sSubsidyId** = Identifikator subvencije.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica.

**iStationId** = Identifikator postaje.

**iTripId** = identifikator vožnje.

**iCurrency** = valuta zneska (978=EUR).

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**iNumberOfProducts** = število produktov (kuponov).

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**ISRefTransactionExecutionId** = referenčni identifikator transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iExitStationId** = Identifikator izstopne postaje.

**iLineId** = identifikator linije.

**sCashierName** = Ime blagajnika.

**ISRefCUID** = identifikator referenčne kartice.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

#### *7.12.14 DUPLIKAT KARTICE*

Pri izdelavi duplikata IJPP kartice se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri inicializaciji kartice so v strukturi IJPPTransaction obvezni naslednji podatki:**

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 14.

**dtStartTime** = čas izdelave duplikata kartice.

**ICUID** = identifikator kartice.

**ISRefCUID** = identifikator originalne kartice iz katere se je izdelal duplikat.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija).

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**Pri inicializaciji kartice so v strukturi IJPPTransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici.

**ILastSuccessfulTransactionExecutionId** = identifikator terminala na katerem je bila izvedena zadnja uspešna transakcija na kartici.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**fPriceValue** = cena vozovnice.

**sSubsidyId** = Identifikator subvencije.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica.

**iStationId** = Identifikator postaje.

**iTripId** = identifikator vožnje.

**iCurrency** = valuta zneska (978=EUR).

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**iNumberOfProducts** = število produktov (kuponov).

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**IRefTransactionExecutionId** = referenčni identifikator transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iExitStationId** = Identifikator izstopne postaje.

**iLineId** = identifikator linije.

**sCashierName** = Ime blagajnika.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

#### **7.12.15**      *KONTROLORJEVA TRANSAKCIJA*

Ob kontroli vozovnice na kontrolorjevem terminalu se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri kontrolorjevi transakciji so v strukturi IJPPTransaction obvezni naslednji podatki:**

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 15.

**dtStartTime** = čas kontrole vozovnice.

**iTerminalId** = identifikator kontrolorjevega terminala na katerem je izvedena transakcija.

**iOffenseType** = tip prekrška, ki ga je ugotovil kontrolor (0-veljavna kartica, 1-kartica je blokirana ali na črni listi, 2-vozovnica ni aktivirana ali validirana, 3-na kartici ni veljavne vozovnice, 4-poneverjanje identitete, 5-potnik nima kartice/vozovnice, 6-vozovnica ni veljavna v trenutni coni).

**lRefCUID** = identifikator kontrolorjeve kartice.

**Pri kontrolorjevi transakciji so v strukturi IJPPTtransaction pogojno obvezni naslednji podatki:**

**ICUID** = identifikator potnikove kartice kartice (parameter pošljemo, kadar se kontrolira kartica).

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na potnikovi kartici (parameter pošljemo, kadar se kontrolira kartica).

**lLastSuccessfulTransactionExecutionId** = identifikator terminala na katerem je bila izvedena zadnja uspešna transakcija na potnikovi kartici (parameter pošljemo, kadar se kontrolira kartica).

**iExecutionStatus** = Status izvedbe transakcije na potnikovi kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija) (parameter pošljemo, kadar se kontrolira kartica).

**iTariffProductId** = identifikator globe iz tarife.

**iTicketStatus** = Identifikator statusa globe.

**iTariffClassId** = identifikator tarifnega razreda globe.

**iTariffId** = identifikator tarife iz katere je bila določena globa.

**iTariffLocationId** = identifikator lokacije uporabe tarife iz katere je bila določena globa.

**fPriceValue** = vrednost globe.

**iTripId** = identifikator vožnje.

**iCurrency** = valuta globe(978=EUR).

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**iPaymentType** = vrsta plačila.

**iNumberOfProducts** = število produktov (kuponov).

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji je obvezen podatek le pri kontroli večconskih vozovnic.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica je obvezen podatek le pri kontroli večconskih vozovnic.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iSanctionType** = tip sankcije, ki jo je izvedel kontrolor (1-izrečena sankcija, kjer je bil potnik je bil sankcioniran z globo, 2-kartica je bila blokirana, 3-kartica je bila dodana na črno listo, 4-izrečeno opozorilo, 5-kartica je bila blokirana zaradi preseženega maksimalnega števila opozoril).

**iWorkOrderId** = identifikator delovne naloge.

**Pri kontrolorjevi transakciji so v strukturi IJPPTtransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**sSubsidyId** = Identifikator subvencije.

**iStationId** = Identifikator postaje.

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**iPosId** = identifikator prodajnega mesta.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**IRefTransactionExecutionId** = referenčni identifikator transakcije.

**iExitStationId** = Identifikator izstopne postaje.

**iLineId** = identifikator linije.

**sCashierName** = Ime blagajnika.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

#### **7.12.16      PRIJAVA KONTROLORJA**

Kontrolor se ob pričetku kotrole prijavi na terminal-validator. Pri tem se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri prijavi kontrolorja so v strukturi IJPPTtransaction obvezni naslednji podatki:**

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 16.

**dtStartTime** = čas prijave

**ICUID** = identifikator kontrolorjeve kartice.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici.

**ILastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici.

**iTripId**= identifikator vožnje.

**iTerminalId** = identifikator terminala, na katerem je izvedena prijava kontrolorja.

**iStationId** = Identifikator vstopne postaje.

**Pri prijavi kontrolorja so v strukturi IJPPTtransaction obvezni naslednji podatki le, če so prisotni na terminalu::**

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice pri nakupu.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**sSubsidyId** = Identifikator subvencije.

**iZoneLocation** = identifikator področja.

**lRefTransactionExecutionId** = referenčni identifikator transakcije.

**iExitStationId** = Identifikator izstopne postaje.

**fPriceValue** = cena vozovnice.

**iCurrency** = valuta zneska (978=EUR).

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**lCashierId** = identifikator blagajnika.

**iNumberOfProducts** = število produktov (kuponov).

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**lReceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iLineId** = identifikator linije.

**sCashierName** = Ime blagajnika.

**lRefCUID** = identifikator referenčne kartice

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice



**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

#### **7.12.17 DODAJANJE RELACIJE NA KARTICO**

Na kartico lahko dodamo medkrajevno relacijo, kjer se zraven ne zapiše medkrajevna vozovnica za to relacijo. V tem primeru se izvede ta transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

***Pri dodajanju relacije na kartico brez vozovnice so v strukturi IJPPTtransaction obvezni naslednji podatki:***

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** =17.

**dtStartTime** = čas transakcije.

**ICUID** = identifikator kartice.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici.

**ILastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija).

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice, ki je bila dodana.

***Pri dodajanju relacije na kartico brez vozovnice so v strukturi IJPPTtransaction obvezni naslednji podatki le, če so prisotni na terminalu:***

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**fPriceValue** = cena vozovnice.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iCurrency** = valuta zneska (978=EUR).

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik dobi subvencionirano vozovnico.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica.

**iStationId** = Identifikator postaje.

**iTripId** = identifikator vožnje.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**iNumberOfProducts** = število produktov (kuponov).

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**IRefTransactionExecutionId** = referenčni identifikator transakcije.

**iExitStationId** = Identifikator izstopne postaje.

**iLineId** = identifikator linije.

**sCashierName** = Ime blagajnika.

**IRefCUID** = identifikator referenčne kartice.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

#### **7.12.18**      *BRISANJE RELACIJE IZ KARTICE*

Na kartici lahko brišemo posamezno medkrajevno relacijo. Pri tem se izvede ta transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri brisanju relacije iz kartice so v strukturi IJPPTransaction obvezni naslednji podatki:**

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 18.

**dtStartTime** = čas transakcije.

**ICUID** = identifikator kartice.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici.

**ILastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija).

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice, ki je bila izbrisana.

**Pri brisanju relacije iz kartice so v strukturi IJPPTransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**fPriceValue** = cena vozovnice.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iCurrency** = valuta zneska (978=EUR).

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik dobi subvencionirano vozovnico.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica.

**iStationId** = Identifikator postaje.

**iTripId** = identifikator vožnje.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**iNumberOfProducts** = število produktov (kuponov).

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**IRefTransactionExecutionId** = referenčni identifikator transakcije.

**iExitStationId** = Identifikator izstopne postaje.

**iLineId** = identifikator linije.

**sCashierName** = Ime blagajnika.

**IRefCUID** = identifikator referenčne kartice.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

#### *7.12.19 VRAČILO VOZOVNICE*

Vračilo vozovnice uporabljamo v primeru, ko potnik vrne vozovnico, ki jo je že uporabil za vožnjo. Pri vsakem vračilu vozovnice se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri vračilu vozovnice so v strukturi IJPPTtransaction obvezni naslednji podatki:**

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 19.

**dtStartTime** = čas vračila vozovnice.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**fPriceValue** = cena vozovnice. Pri vračilu posredujemo negativno vrednost zneska, ki je vrnjen potniku. Znesek ima lahko poljuben delež negativne vrednosti cene vozovnice iz tarife.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iCurrency** = valuta zneska (978=EUR).

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**lRefTransactionExecutionId** = identifikator transakcije s katero je bil izveden nakup vozovnice, ki ga vračamo.

**iNumberOfProducts** = število produktov (kuponov). Pri vračilu posredujemo negativno vrednost števila kuponov, ki so ostali neaktivirani.

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**sCashierName** = Ime blagajnika.

**Pri vračilu vozovnice so v strukturi IJPPTransaction pogojno obvezni naslednji podatki:**

**ICUID** = identifikator kartice je obvezen podatek le, če je bil nakup vrnjene vozovnice izveden na kartici.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici je obvezen podatek le, če je bil nakup vrnjene vozovnice izveden na kartici.

**lLastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici je obvezen podatek le, če je bil nakup vrnjene vozovnice izveden na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija) je obvezen podatek le, če je bil nakup vrnjene vozovnice izveden na kartici.

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik kupi subvencionirano vozovnico.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica je obvezen podatek v primeru, ko potnik kupi vozovnico, ki velja v več območjih.

**iStationId** = Identifikator postaje je obvezen podatek v primeru, ko ima terminal informacijo o vstopni postaji.

**iTripId** = identifikator vožnje je obvezen podatek le v primeru, ko ima terminal informacijo o vožnji.

**Pri vračilu vozovnice so v strukturi IJPPTransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iExitStationId** = Identifikator izstopne postaje.

**iLineId** = identifikator linije.

**IRefCUID** = identifikator referenčne kartice

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

#### **7.12.20 ZAMENJAVA VOZOVNICE – STORNACIJA**

Zamenjavo vozovnice izvedemo v primeru, ko ima potnik že koriščeno vozovnico, katero zamenjamo z drugo vrsto vozovnice. Zamenjavo opravimo z dvema IJPP transakcijama. Najprej izvedemo transakcijo »Zamenjava vozovnice - stornacija«, kjer storniramo zamenjano vozovnico. Nato izvedemo transakcijo »Zamenjava vozovnice - nakup«, kjer na kartico naložimo novo vozovnico. Pri zamenjavi vozovnice – stornaciji se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.



***Pri zamenjavi vozovnice- stornaciji so v strukturi IJPPTtransaction obvezni naslednji podatki:***

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 20.

**dtStartTime** = čas zamenjave – stornacije vozovnice.

**iTariffProductId** = identifikator vrste zamenjane vozovnice.

**iTicketStatus** = Identifikator statusa (pravica) zamenjane vozovnice.

**iTariffClassId** = identifikator tarifnega razreda zamenjane vozovnice.

**fPriceValue** = negativna vrednost cene zamenjane vozovnice.

**iTariffId** = identifikator tarife iz katere je bila določena cena zamenjane vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iCurrency** = valuta zneska (978=EUR).

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**dtValidFrom** = začetni neaktivni čas veljavnosti zamenjane vozovnice.

**dtValidTo** = končni neaktivni čas veljavnosti zamenjane vozovnice.

**iNumberOfProducts** = negativna vrednost števila produktov (kuponov), določa koliko krat lahko potnik aktivira vozovnico.

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**sCashierName** = Ime blagajnika.

**lRefTransactionExecutionId** = referenčni identifikator transakcije s katero je bila kupljena zamenjana vozovnica.

***Pri zamenjavi vozovnice- stornaciji so v strukturi IJPPTtransaction pogojno obvezni naslednji podatki:***

**ICUID** = identifikator kartice je obvezen podatek le, če je zamenjana vozovnica zapisana na kartici.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici je obvezen podatek le, če je zamenjana vozovnica zapisana na kartici.

**lLastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici je obvezen podatek le, če je zamenjana vozovnica zapisana na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija) je obvezen podatek le, če je zamenjana vozovnica zapisana na kartici.

**Pri zamenjavi vozovnice - stronaciji so v strukturi IJPPTransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**sSubsidyId** = Identifikator subvencije .

**iZoneLocation** = identifikator področja postaje.

**iStationId** = Identifikator postaje.

**iTripId** = identifikator vožnje .

**iExitStationId** = Identifikator izstopne postaje.

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iLineId** = identifikator linije.

**IRefCUID** = identifikator referenčne kartice

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

### **7.12.21      Ročna validacija vozovnice**

V posebnem primeru, ko potnik nima vozovnice z ustrezno relacijo, je v sistemu IJPP dovoljena ročna validacija vozovnice. Pri tej validaciji upravljalet terminala ročno potrdi (validira) vozovnico, ki bi bila sicer zavrnjena zaradi neustrezne relacije. Vozovnica na kartici v tem primeru ostane nespremenjena (neaktivirana). Ročne validacije vozovnic pošiljajo le terminali validatorji. Podatki validacije se prenesejo iz validatorjev v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri ročni validaciji vozovnice so v strukturi IJPPTransaction obvezni naslednji podatki:**

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 21.

**dtStartTime** = čas ročne validacije vozovnice.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice pri nakupu.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**iPosId** = identifikator prodajnega mesta (vozila).

**lRefTransactionExecutionId** = referenčni identifikator transakcije s katero je bil izveden nakup vozovnice, ki jo ročno validiramo.

**iNumberOfProducts** = število produktov (kuponov), določa koliko aktivacij vozovnice ima potnik na voljo.

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**iTripId** = identifikator vožnje.

**sCashierName** = Ime blagajnika.

**Pri ročni validaciji vozovnice so v strukturi IJPPTtransaction pogojno obvezni naslednji podatki:**

**ICUID** = identifikator kartice je obvezen podatek le, če je bila validirana vozovnica na kartici.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici je obvezen podatek le, če je bila validirana vozovnica na kartici.

**lLastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici je obvezen podatek le, če je bila validirana vozovnica na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija) je obvezen podatek le, če je bila validirana vozovnica na kartici.

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik kupi subvencionirano vozovnico.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica je obvezen podatek v primeru, ko potnik kupi vozovnico, ki velja v več območjih.

**iStationId** = Identifikator postaje je obvezen podatek v primeru, ko ima terminal informacijo o vstopni postaji.

**iExitStationId** = Identifikator izstopne postaje, je obvezen podatek le pri validacijah medkrajevnih vozovnic, kjer je na voljo informacija o izstopni postaji.

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji je obvezen podatek le pri večconskih vozovnicah.

**iOffenseType** = tip prekrška se pošlje kadar je validacija izvedena na postaji izven voznega reda.

**iWorkOrderId** = identifikator delovne naloge.

**Pri ročni validaciji vozovnice so v strukturi IJPPTtransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**fPriceValue** = cena vozovnice.

**iCurrency** = valuta zneska (978=EUR).

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iLineId** = identifikator linije.

**IRefCUID** = identifikator referenčne kartice.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

#### **7.12.22**      *BLOKIRANJE VOZOVNICE NA KARTICI IZ IZVRŠNE LISTE*

Pri vsaki blokirani vozovnici, ki je na izvršni listi se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri blokiranju vozovnice na kartici so v strukturi IJPPTransaction obvezni naslednji podatki:**

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 22.

**dtStartTime** = čas blokade vozovnice na kartici.

**ICUID** = identifikator kartice.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici.

**lLastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija).

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**lRefTransactionExecutionId** = identifikator transakcije s katero je bil izveden nakup vozovnice, ki jo blokiramo.

**Pri blokiranju vozovnice na kartici so v strukturi IJPPTtransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**fPriceValue** = cena vozovnice.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iCurrency** = valuta zneska (978=EUR).

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik dobi subvencionirano vozovnico.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica.

**iStationId** = Identifikator postaje.

**iTripId** = identifikator vožnje.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**iNumberOfProducts** = število produktov (kuponov).

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iExitStationId** = Identifikator izstopne postaje.

**iLineId** = identifikator linije.

**sCashierName** = Ime blagajnika.

**ISRefCUID** = identifikator referenčne kartice.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

### **7.12.23 ODBLOKIRANJE VOZOVNICE NA KARTICI IZ IZVRŠNE LISTE**

Pri vsaki odblokirani vozovnici, ki je na izvršni listi se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri odblokiranju vozovnice na kartici so v strukturi IJPPTtransaction obvezni naslednji podatki:**

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** =23.

**dtStartTime** = čas deblokade vozovnice na kartici.

**ICUID** = identifikator kartice.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici.

**ILastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija).

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**IRefTransactionExecutionId** = identifikator transakcije s katero je bil izveden nakup vozovnice, ki jo odblokiramo.

**Pri blokiranju vozovnice na kartici so v strukturi IJPPTtransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**fPriceValue** = cena vozovnice.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iCurrency** = valuta zneska (978=EUR).

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.



**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik dobi subvencionirano vozovnico.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica.

**iStationId** = Identifikator postaje.

**iTripId** = identifikator vožnje.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**iNumberOfProducts** = število produktov (kuponov).

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iExitStationId** = Identifikator izstopne postaje.

**iLineId** = identifikator linije.

**sCashierName** = Ime blagajnika.

**IRefCUID** = identifikator referenčne kartice.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

#### **7.12.24 NEUSPELO BRANJE KARTICE**

V določenih primerih terminal ne uspe v celoti prebrati kartice. V takšnih primerih se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri neuspelem branju kartice so v strukturi IJPPTtransaction obvezni naslednji podatki:**

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** =24.

**dtStartTime** = čas branja kartice.

**ICUID** = identifikator kartice.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**Pri neuspelem branju kartice so v strukturi IJPPTtransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici.

**lLastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija).

**lRefTransactionExecutionId** = identifikator transakcije s katero je bil izveden nakup vozovnice, ki jo odblokiramo.

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**fPriceValue** = cena vozovnice.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iCurrency** = valuta zneska (978=EUR).

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik dobi subvencionirano vozovnico.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica.

**iStationId** = Identifikator postaje.

**iTripId** = identifikator vožnje.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**iNumberOfProducts** = število produktov (kuponov).

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**IRefTransactionExecutionId** = referenčni identifikator transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iExitStationId** = Identifikator izstopne postaje.

**iLineId** = identifikator linije.

**sCashierName** = Ime blagajnika.

**lRefCUID** = identifikator referenčne kartice.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

#### **7.12.25 ZAMENJAVA VOZOVNICE – NAKUP**

Zamenjavo vozovnice izvedemo v primeru, ko ima potnik že korišćeno vozovnico, katero zamenjamo z drugo vrsto vozovnice. Zamenjavo opravimo z dvema IJPP transakcijama. Najprej izvedemo transakcijo »Zamenjava vozovnice - stornacija«, kjer storniramo zamenjano vozovnico. Nato izvedemo transakcijo »Zamenjava vozovnice - nakup«, kjer na kartico naložimo novo vozovnico. Pri zamenjavi vozovnice – stornaciji se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

***Pri zamenjavi vozovnice- nakupu so v strukturi IJPPTtransaction obvezni naslednji podatki:***

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 25.

**dtStartTime** = čas zamenjave – nakupa vozovnice.

**iTariffProductId** = identifikator vrste nove vozovnice.

**iTicketStatus** = Identifikator statusa (pravica) nove vozovnice.

**iTariffClassId** = identifikator tarifnega razreda nove vozovnice.

**fPriceValue** = cena nove vozovnice.

**iTariffId** = identifikator tarife iz katere je bila določena cena nove vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iCurrency** = valuta zneska (978=EUR).

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**dtValidFrom** = začetni neaktivni čas veljavnosti nove vozovnice.

**dtValidTo** = končni neaktivni čas veljavnosti nove vozovnice.

**iNumberOfProducts** = število produktov (kuponov), določa koliko krat lahko potnik aktivira vozovnico.

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**sCashierName** = Ime blagajnika.

**lRefTransactionExecutionId** = referenčni identifikator transakcije s katero je bila kupljena vozovnica, ki jo želimo zamenjati.

***Pri zamenjavi vozovnice- nakupu so v strukturi IJPPTtransaction pogojno obvezni naslednji podatki:***

**ICUID** = identifikator kartice je obvezen podatek le, če je vozovnica zapisana na kartici.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici je obvezen podatek le, če je vozovnica zapisana na kartici.

**lLastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici je obvezen podatek le, če je vozovnica zapisana na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija) je obvezen podatek le, če je vozovnica zapisana na kartici.

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik kupi subvencionirano vozovnico.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica je obvezen podatek v primeru, ko potnik kupi vozovnico, ki velja v več območjih.

**iStationId** = Identifikator postaje je obvezen podatek v primeru, ko ima terminal informacijo o vstopni postaji.

**iTripId** = identifikator vožnje je obvezen podatek le v primeru, ko ima terminal informacijo o vožnji.

**iExitStationId** = Identifikator izstopne postaje, je obvezen podatek le pri nakupih medkrajevnih vozovnic, kjer je na voljo informacija o izstopni postaji.

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji je obvezen podatek le pri večconskih vozovnicah.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice. Obvezen podatek pri nakupu medkrajevne vozovnice

***Pri zamenjavi vozovnice- nakupu so v strukturi IJPPTransaction obvezni naslednji podatki le, če so prisotni na terminalu:***

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iLineId** = identifikator linije.

**IRefCUID** = identifikator referenčne kartice

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

#### *7.12.26 POSODOBITEV RELACIJE NA KARTICI IZ IZVRŠNE LISTE*

Pri vsakem prenosu relacije iz izvršne liste na kartico se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri posodobitvi relacije iz izvršne liste so v strukturi IJPPTtransaction obvezni naslednji podatki:**

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** =26.

**dtStartTime** = čas izvedbe transakcije na kartici.

**ICUID** = identifikator kartice.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici.

**ILastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija).

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**lRefTransactionExecutionId** = referenčni identifikator transakcije je določen predhodno na zalednem sistemu in je prisoten kot parameter v izvršni listi.

**mCardRelationData** = podatki o relaciji medkrajevnih vozovnic.

**Pri posodobitvi relacije iz izvršne liste so v strukturi IJPPTtransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**fPriceValue** = cena vozovnice.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iCurrency** = valuta zneska (978=EUR).

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik dobi subvencionirano vozovnico.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica.

**iStationId** = Identifikator postaje.

**iTripId** = identifikator vožnje.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**iNumberOfProducts** = število produktov (kuponov).

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.



**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iExitStationId** = Identifikator izstopne postaje.

**iLineId** = identifikator linije.

**sCashierName** = Ime blagajnika.

**lRefCUID** = identifikator referenčne kartice.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

#### *7.12.27      TRANSAKCIJA ZA RAZHROŠČEVANJE*

Terminal lahko na zaledni sistem pošlje IJPP transakcijo, namenjeno za preverjanje podatkov, testiranje in razhroščevanje. Transakcija v zalednem sistemu nima učinka. Zaledni sistem le hrani podatke te transakcije. Transakcija je vedno zavedena kot neuspešna in se ne beleži v poročilih. Podatki transakcije brez učinka se prenesejo v zaledni sistem z WS metodo ProcessTransactions.

**Pri transakciji zarazhroščevanje so v strukturi IJPPTransaction obvezni naslednji podatki:**

**lTransactionExecutionId** = identifikator transakcije.

**iTransactionType** =27.

**dtStartTime** = čas izvedbe transakcije.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**Pri transakciji za razhroščevanje so v strukturi IJPPTransaction neobvezni naslednji podatki:**

**ICUID** = identifikator kartice.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici.

**lLastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija).

**lRefTransactionExecutionId** = referenčni identifikator transakcije je določen predhodno na zalednem sistemu in je prisoten kot parameter v izvršni listi.

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**fPriceValue** = cena vozovnice.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iCurrency** = valuta zneska (978=EUR).

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik dobi subvencionirano vozovnico.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica.

**iStationId** = Identifikator postaje.

**iTripId** = identifikator vožnje.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**iNumberOfProducts** = število produktov (kuponov).

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake.

**iExitStationId** = Identifikator izstopne postaje.

**iLineId** = identifikator linije.

**sCashierName** = Ime blagajnika.

**IRefCUID** = identifikator referenčne kartice.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

### **7.12.31**      *POGOJNA VALIDACIJA VOZOVNICE*

Pogojne validacije vozovnic pošiljajo le terminali validatorji na vožnjah, ki so sestavljene iz »sestavljenih linij«, kjer ena validacija ni dovolj za opis potnikove vožnje. Pogojne validacije vozovnic se pošiljajo le pri prestopnih vozovnicah. Kadar imamo vožnjo sestavljeno iz več linij, terminal pošlje validacijo za vsako linijo posebej, če je potnik upravičen do vožnje po sestavljenih linijah. Podatki pogojne validacije se prenesejo iz validatorjev v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

***Pri pogojni validaciji vozovnice so v strukturi IJPPTtransaction obvezni naslednji podatki:***

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 31.

**dtStartTime** = čas pogojne validacije vozovnice.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**fPriceValue** = cena vozovnice.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice pri nakupu.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**iPosId** = identifikator prodajnega mesta (vozila).

**lRefTransactionExecutionId** = referenčni identifikator transakcije s katero je bil izveden nakup vozovnice, ki jo validiramo.

**iNumberOfProducts** = število produktov (kuponov), določa koliko preostalih aktivacij vozovnice ima potnik še na voljo.

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**iTripId** = identifikator vožnje.

***Pri pogojni validaciji vozovnice so v strukturi IJPPTtransaction pogojno obvezni naslednji podatki:***

**ICUID** = identifikator kartice je obvezen podatek le, če je bila validirana vozovnica na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija) je obvezen podatek le, če je bila validirana vozovnica na kartici.

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik kupi subvencionirano vozovnico.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica je obvezen podatek v primeru, ko potnik kupi vozovnico, ki velja v več območjih.

**iStationId** = Identifikator postaje je obvezen podatek v primeru, ko ima terminal informacijo o vstopni postaji.

**iExitStationId** = Identifikator izstopne postaje, je obvezen podatek le pri validacijah medkrajevnih vozovnic, kjer je na voljo informacija o izstopni postaji.

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji je obvezen podatek le pri večconskih vozovnicah.

**iOffenseType** = tip prekrška se pošlje kadar je validacija izvedena na postaji izven voznega reda.

**iWorkOrderId** = identifikator delovne naloge.

**Pri pogojni validaciji vozovnice so v strukturi IJPPTransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici.

**lLastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici.

**iCurrency** = valuta zneska (978=EUR).

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iLineId** = identifikator linije.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**sCashierName** = Ime blagajnika.

**lRefCUID** = identifikator referenčne kartice.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

### **7.12.36 POSODOBITEV VLOGE NA KARTICI IZ IZVRŠNE LISTE**

Pri vsaki posodobitvi vloge na kartici se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri posodobitvi vloge iz izvršne liste so v strukturi IJPPTtransaction obvezni naslednji podatki:**

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** =36.

**dtStartTime** = čas izvedbe transakcije na kartici.

**ICUID** = identifikator kartice.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici.

**ILastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija).

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**IRefTransactionExecutionId** = referenčni identifikator transakcije je določen predhodno na zalednem sistemu in je prisoten kot parameter v izvršni listi.

**sSubsidyId** = Identifikator subvencije.

**Pri posodobitvi vloge iz izvršne liste so v strukturi IJPPTtransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**fPriceValue** = cena vozovnice.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iCurrency** = valuta zneska (978=EUR).

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica.

**iStationId** = Identifikator postaje.

**iTripId** = identifikator vožnje.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**iNumberOfProducts** = število produktov (kuponov).

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iExitStationId** = Identifikator izstopne postaje.

**iLineId** = identifikator linije.

**sCashierName** = Ime blagajnika.

**IRefCUID** = identifikator referenčne kartice.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

#### *7.12.37 POSODOBITEV VLOGE NA KARTICI*

Pri vsaki posodobitvi vloge na kartici se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri posodobitvi vloge so v strukturi IJPPTransaction obvezni naslednji podatki:**

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** =36.

**dtStartTime** = čas izvedbe transakcije na kartici.

**ICUID** = identifikator kartice.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici.

**ILastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija).

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**sSubsidyId** = Identifikator subvencije.



***Pri posodobitvi vloge so v strukturi IJPPTtransaction obvezni naslednji podatki le, če so prisotni na terminalu:***

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**fPriceValue** = cena vozovnice.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iCurrency** = valuta zneska (978=EUR).

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica.

**lRefTransactionExecutionId** = referenčni identifikator transakcije.

**iStationId** = Identifikator postaje.

**iTripId** = identifikator vožnje.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**lCashierId** = identifikator blagajnika.

**iNumberOfProducts** = število produktov (kuponov).

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**lReceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iExitStationId** = Identifikator izstopne postaje.

**iLineId** = identifikator linije.

**sCashierName** = Ime blagajnika.

**IRefCUID** = identifikator referenčne kartice.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

### **7.12.38**      *VRAČILO NEPREVZETE VOZOVNICE*

Vračilo neprevzete vozovnice uporabljamo v primeru, ko potnik preko spleta kupi vozovnico, vendar je ne prevzame oziroma naloži na kartico. Pri vsakem vračilu neprevzete vozovnice se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri vračilu neprevzete vozovnice so v strukturi IJPPTransaction obvezni naslednji podatki:**

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 38.

**dtStartTime** = čas vračila vozovnice.

**ICUID** = identifikator kartice.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**fPriceValue** = cena vozovnice. Pri vračilu posredujemo negativno vrednost zneska, ki je vrnjen potniku. Znesek ima lahko poljuben delež negativne vrednosti cene vozovnice iz tarife.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iCurrency** = valuta zneska (978=EUR).

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**lRefTransactionExecutionId** = identifikator transakcije s katero je bil izveden nakup vozovnice, ki ga vračamo.

**iNumberOfProducts** = število produktov (kuponov). Pri vračilu posredujemo negativno vrednost števila kuponov, ki so ostali neaktivirani.

**sCashierName** = Ime blagajnika.

**Pri vračilu vozovnice so v strukturi IJPPTtransaction pogojno obvezni naslednji podatki:**

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik kupi subvencionirano vozovnico.

**Pri vračilu vozovnice so v strukturi IJPPTtransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici je obvezen podatek le, če je bil nakup vrnjene vozovnice izveden na kartici.

**ILastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici je obvezen podatek le, če je bil nakup vrnjene vozovnice izveden na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija) je obvezen podatek le, če je bil nakup vrnjene vozovnice izveden na kartici.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica je obvezen podatek v primeru, ko potnik kupi vozovnico, ki velja v več območjih.

**iStationId** = Identifikator postaje je obvezen podatek v primeru, ko ima terminal informacijo o vstopni postaji.

**iTripId** = identifikator vožnje je obvezen podatek le v primeru, ko ima terminal informacijo o vožnji.

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iExitStationId** = Identifikator izstopne postaje.

**iLineId** = identifikator linije.

**IRefCUID** = identifikator referenčne kartice

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

### **7.12.39      *PRENOS KOORDINAT IZ POSTAJNE TOČKE***

Terminal lahko v zaledni sistem pošlje GPS koordinate postajne točke. Pri tem se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

***Pri prenosu koordinat iz postajne točke so v strukturi IJPPTtransaction obvezni naslednji podatki:***

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** =39.

**dtStartTime** = čas izvedbe transakcije.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

***Pri prenosu koordinat iz postajne točke so v strukturi IJPPTtransaction obvezni naslednji podatki le, če so prisotni na terminalu:***

**ICUID** = identifikator kartice.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici.

**ILastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija).

**ISRefTransactionExecutionId** = identifikator transakcije s katero je bil izveden nakup vozovnice, ki jo odblokiramo.

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**fPriceValue** = cena vozovnice.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iCurrency** = valuta zneska (978=EUR).

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik dobi subvencionirano vozovnico.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica.

**iStationId** = Identifikator postaje.

**iTripId** = identifikator vožnje.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**iNumberOfProducts** = število produktov (kuponov).

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**IRefTransactionExecutionId** = referenčni identifikator transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iExitStationId** = Identifikator izstopne postaje.

**iLineId** = identifikator linije.

**sCashierName** = Ime blagajnika.

**lRefCUID** = identifikator referenčne kartice.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

#### *7.12.40 REKLAMACIJA*

Pri vsaki reklamaciji se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri reklamaciji so v strukturi IJPPTransaction obvezni naslednji podatki:**

**lTransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 40.

**dtStartTime** = čas reklamacije vozovnice (čas zapisa vozovnice na kartico).

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**fPriceValue** = cena vozovnice.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iCurrency** = valuta zneska (978=EUR).

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**dtValidFrom** = začetni neaktivni čas veljavnosti vozovnice.

**dtValidTo** = končni neaktivni čas veljavnosti vozovnice.

**iNumberOfProducts** = število produktov (kuponov), določa koliko krat lahko potnik aktivira vozovnico.

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**sCashierName** = Ime blagajnika.

**Pri reklamaciji so v strukturi IJPPTtransaction pogojno obvezni naslednji podatki:**

**ICUID** = identifikator kartice je obvezen podatek le, če je vozovnica zapisana na kartici.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici je obvezen podatek le, če je vozovnica zapisana na kartici.

**lLastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici je obvezen podatek le, če je vozovnica zapisana na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija) je obvezen podatek le, če je vozovnica zapisana na kartici.

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik kupi subvencionirano vozovnico.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica je obvezen podatek v primeru, ko potnik kupi vozovnico, ki velja v več območjih.

**iStationId** = Identifikator postaje je obvezen podatek v primeru, ko ima terminal informacijo o vstopni postaji.

**iTripId** = identifikator vožnje je obvezen podatek le v primeru, ko ima terminal informacijo o vožnji.

**iExitStationId** = Identifikator izstopne postaje, je obvezen podatek le pri nakupih medkrajevnih vozovnic, kjer je na voljo informacija o izstopni postaji.

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji je obvezen podatek le pri večconskih vozovnicah.



**mCardRelationData** = podatki o relaciji medkrajevne vozovnice. Obvezen podatek pri nakupu medkrajevne vozovnice

**Pri reklamaciji so v strukturi IJPPTransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**IRefTransactionExecutionId** = referenčni identifikator transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iLineId** = identifikator linije.

**IRefCUID** = identifikator referenčne kartice

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

#### 7.12.41 VRAČILO DOBROPISA

Vračilo dobropisa uporabljamo v primeru, ko potnik vrne vozovnico zaradi omejitev javnega potniškega prometa. Pri vsakem vračilu vozovnice se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri vračilu dobropisa so v strukturi IJPPTtransaction obvezni naslednji podatki:**

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 41.

**dtStartTime** = čas vračila vozovnice.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**ICUID** = identifikator karice.

**iCurrency** = valuta zneska (978=EUR).

**fPriceValue** = znesek dobropisa. Pri vračilu posredujemo negativno vrednost zneska.

**sCashierName** = Ime blagajnika.

**Pri vračilu dobropisa so v strukturi IJPPTtransaction pogojno obvezni naslednji podatki:**

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici je obvezen podatek le, če je bil nakup vrnjene vozovnice izveden na kartici.

**lLastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici je obvezen podatek le, če je bil nakup vrnjene vozovnice izveden na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija) je obvezen podatek le, če je bil nakup vrnjene vozovnice izveden na kartici.

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik kupi subvencionirano vozovnico.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica je obvezen podatek v primeru, ko potnik kupi vozovnico, ki velja v več območjih.

**iStationId** = Identifikator postaje je obvezen podatek v primeru, ko ima terminal informacijo o vstopni postaji.

**iTripId** = identifikator vožnje je obvezen podatek le v primeru, ko ima terminal informacijo o vožnji.

**iNumberOfProducts** = število produktov (kuponov). Pri vračilu posredujemo negativno vrednost števila kuponov, ki so ostali neaktivirani.

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**lRefTransactionExecutionId** = identifikator transakcije s katero je bil izveden nakup vozovnice, ki ga vračamo.

***Pri koriščenju dobropisa so v strukturi IJPPTransaction obvezni naslednji podatki le, če so prisotni na terminalu:***

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**lCashierId** = identifikator blagajnika.

**lReceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**lErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iExitStationId** = Identifikator izstopne postaje.

**lLineId** = identifikator linije.

**lRefCUID** = identifikator referenčne kartice

**lOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije. Če je izbrano plačilno sredstvo 30 – nakazilo, potem se v polju Notes zapiše string v obliki : #TRRSI 1234 134134 4134 14123;ime;priimek;tukaj 12;maribor;2000TRR# kjer so med znakoma #TRR in TRR# sledeči podatki: številka trr računa; ime; priimek; naslov; mesto; zip. Razločevalnik naštetih podatkov predstavlja podpičje. Ti podatki so na voljo 10 dni od izvedbe transakcije. Po tem času se izbrišejo.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

#### **7.12.42 KORIŠČENJE DOBROPISA**

Koriščenje dobropisa uporabljamo v primeru, ko potnik koristi dobropis. Pri vsakem koriščenju dobropisa se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

**Pri koriščenju dobropisa so v strukturi IJPPTtransaction obvezni naslednji podatki:**

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 42.

**dtStartTime** = čas koriščenja dobropisa.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**ICUID** = identifikator karice.

**iCurrency** = valuta zneska (978=EUR).

**fPriceValue** = znesek dobropisa. Pri vračilu posredujemo pozitivno vrednost zneska.

**sCashierName** = Ime blagajnika.

**Pri koriščenju dobropisa so v strukturi IJPPTtransaction pogojno obvezni naslednji podatki:**

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici je obvezen podatek le, če je bil nakup vrnjene vozovnice izveden na kartici.

**lLastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici je obvezen podatek le, če je bil nakup vrnjene vozovnice izveden na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija) je obvezen podatek le, če je bil nakup vrnjene vozovnice izveden na kartici.

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik kupi subvencionirano vozovnico.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica je obvezen podatek v primeru, ko potnik kupi vozovnico, ki velja v več območjih.

**iStationId** = Identifikator postaje je obvezen podatek v primeru, ko ima terminal informacijo o vstopni postaji.

**iTripId** = identifikator vožnje je obvezen podatek le v primeru, ko ima terminal informacijo o vožnji.

**iNumberOfProducts** = število produktov (kuponov). Pri vračilu posredujemo negativno vrednost števila kuponov, ki so ostali neaktivirani.

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**lRefTransactionExecutionId** = identifikator transakcije s katero je bil izveden nakup vozovnice, ki ga vračamo.

**Pri koriščenju dobropisa so v strukturi IJPPTtransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**IRceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iExitStationId** = Identifikator izstopne postaje.

**iLineId** = identifikator linije.

**IRefCUID** = identifikator referenčne kartice

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

#### **7.12.43**      *PRIPIS DOBROPISA*

Pripis dobropisa uporabljamo v primeru, ko potnik prejme dobropis. Pri vsakem prejemu dobropisa se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

***Pri pripisu dobropisa so v strukturi IJPPTtransaction obvezni naslednji podatki:***

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** = 43.

**dtStartTime** = čas pripisa dobropisa.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**ICUID** = identifikator karice.

**iCurrency** = valuta zneska (978=EUR).

**fPriceValue** = znesek dobropisa.

**sCashierName** = Ime blagajnika.

***Pri pripisu dobropisa so v strukturi IJPPTtransaction pogojno obvezni naslednji podatki:***

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**dtValidFrom** = začetni čas veljavnosti vozovnice.

**dtValidTo** = končni čas veljavnosti vozovnice.

**iTariffProductId** = identifikator vrste vozovnice.

**iTicketStatus** = Identifikator statusa (pravica).

**iTariffClassId** = identifikator tarifnega razreda.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici je obvezen podatek le, če je bil nakup vrnjene vozovnice izveden na kartici.

**lLastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici je obvezen podatek le, če je bil nakup vrnjene vozovnice izveden na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija) je obvezen podatek le, če je bil nakup vrnjene vozovnice izveden na kartici.

**sSubsidyId** = Identifikator subvencije je obvezen podatek v primeru, ko potnik kupi subvencionirano vozovnico.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica je obvezen podatek v primeru, ko potnik kupi vozovnico, ki velja v več območjih.

**iStationId** = Identifikator postaje je obvezen podatek v primeru, ko ima terminal informacijo o vstopni postaji.

**iTripId** = identifikator vožnje je obvezen podatek le v primeru, ko ima terminal informacijo o vožnji.

**iNumberOfProducts** = število produktov (kuponov). Pri vračilu posredujemo negativno vrednost števila kuponov, ki so ostali neaktivirani.

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**IRefTransactionExecutionId** = identifikator transakcije s katero je bil izveden nakup vozovnice, ki ga vračamo.

**Pri koriščenju dobropisa so v strukturi IJPPTransaction obvezni naslednji podatki le, če so prisotni na terminalu:**

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**ICashierId** = identifikator blagajnika.

**IReceiptNumber** = številka računa.

**iTicketStatusFormat** = format statusa vozovnice.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iExitStationId** = Identifikator izstopne postaje.

**iLineId** = identifikator linije.

**IRefCUID** = identifikator referenčne kartice

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.



**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

#### **7.12.47      *PODALJŠANJE SUBVENCIONIRANEGA STATUSA NA KARTICI***

Pri podaljšanju subvencioniranega statusa na kartici, se izvede transakcija, katere podatki se prenesejo v zaledni sistem z WS metodo ProcessTransactions. Podaljšanje se izvede samodejno na terminalu. Samodejno se podaljšajo naslednje vrste statusov (glej poglavje 4.2.4 Statusi): 11, 12, 13, 14: Zaledni sistem transakcijo procesira, preveri pravilnost podatkov in pripravi podatke za poročila.

***Pri podaljšanju statusa so v strukturi IJPPTtransaction obvezni naslednji podatki:***

**ITransactionExecutionId** = identifikator transakcije.

**iTransactionType** =47.

**dtStartTime** = čas podaljšanja statusa.

**ICUID** = identifikator kartice.

**iTicketStatus** = Identifikator statusa (pravica).

**dtValidFrom** = začetni čas veljavnosti podaljšanega statusa.

**dtValidTo** = končni čas veljavnosti podaljšanega statusa.

**sSubsidyId** = Identifikator subvencije.

**dtLastSuccessfulTime** = čas zadnje uspešne transakcije na kartici.

**ILastSuccessfulTransactionExecutionId** = identifikator zadnje uspešne transakcije na kartici.

**iExecutionStatus** = Status izvedbe transakcije na kartici (0-uspešno izvedena transakcija, 1-nepreverjena transakcija).

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iTicketStatusFormat** = format statusa vozovnice.

***Pri podaljšanju statusa so v strukturi IJPPTtransaction obvezni naslednji podatki le, če so prisotni na terminalu:***

**lRefTransactionExecutionId** = referenčni identifikator transakcije je določen predhodno na zalednem sistemu in je prisoten kot parameter v izvršni listi.

**iTariffClassTransitionsId** = Identifikator verzije podatkov o prehodih med področji.

**iTariffProductId** = identifikator vrste vozovnice.

**iTariffClassId** = identifikator tarifnega razreda.

**fPriceValue** = cena vozovnice.

**iTariffId** = identifikator tarife iz katere je bila določena cena vozovnice.

**iTariffLocationId** = identifikator lokacije uporabe tarife.

**iTerminalId** = identifikator terminala na katerem je izvedena transakcija.

**iCurrency** = valuta zneska (978=EUR).

**iBlacklistId** = identifikator aktivne črne liste v času izvedbe transakcije.

**iExeListId** = identifikator aktivne izvršne liste v času izvedbe transakcije.

**iZoneLocation** = identifikator področja postaje v katerem je bila kupljena vozovnica.

**iStationId** = Identifikator postaje.

**iTripId** = identifikator vožnje.

**iPosId** = identifikator prodajnega mesta.

**iPaymentType** = vrsta plačila.

**iBusId** = identifikator vozila.

**lCashierId** = identifikator blagajnika.

**iNumberOfProducts** = število produktov (kuponov).

**iNumberOfProductsSetInterval** = časovni interval nastavitve števila produktov.

**lReceiptNumber** = številka računa.

**dGPSLongitude** = geografski položaj izvedbe transakcije.

**dGPSLatitude** = geografski položaj izvedbe transakcije.

**iErrorCode** = koda napake. Prisotna koda napake z vrednostjo različno od 0 označuje neuspešno ijpp transakcijo.

**iExitStationId** = Identifikator izstopne postaje.

**iLineId** = identifikator linije.

**sCashierName** = Ime blagajnika.

**lRefCUID** = identifikator referenčne kartice.

**iOfflineStatus** = Status kartice, ki ga nastavi zaledni sistem pri procesiranju IJPP transakcije.

**dtExecutionTime** = Čas procesiranja transakcije na zalednem sistemu.

**mCardRelationData** = podatki o relaciji medkrajevne vozovnice.

**iOffenseType** = tip prekrška.

**iSanctionType** = tip sankcije.

**iOffenseCount** = Skupno število prekrškov, ki so bili izrečeni lastniku kartice.

**iWorkOrderId** = identifikator delovne naloge.

**sNotes** = zaznamki transakcije.

**sPurchaseOrder** = oznaka naročilnice.

**fSubFullPrice**= polna cena subvencioniranih vozovnic.

**fSubFullPriceReported**= poročana cena subvencioniranih vozovnic.

**fSubPayment**= plačilo subvencioniranih vozovnic.

## 8 METODE SPLETNE STORITVE IJPP - ADMINISTRACIJA

V tem poglavju so opisane administracijske metode spletne storitve IJPP. Privzeta kodna tabela metod spletne storitve IJPP je UTF-8.

## 8.1 SETPASSENGER

Metodo SetPassenger uporabljamo za upravljanje podatkov o potnikih. Metoda omogoča vnos novih potnikov v zaledni sistem. Prav tako lahko s pomočjo metode podatke obstoječega potnika spreminjamo oziroma posodabljam.

Deklaracija metode			
<b>Int SetPassenger (Passenger mPassengerData [IN], int iResultCode [OUT], string sResultDescription [OUT], int iPid [OUT])</b>			
Parameter	Tip parametra	Obveznost parametra / Vhodni -Izhodni	Opis parametra
mPassengerData	Passenger*	Yes / Input	Struktura s podatki potnika.
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.
iPid	Integer	- / Output	Identifikator potnika.

\* Struktura **Passenger** je definirana v poglavju definicije podatkovnih struktur.

Možne kode rezultata za metodo **SetPassenger**.

iResultCode	sResultDescription	Extra description
-721	Invalid input data.	Vnos potnika neuspešen zaradi nepravilnih podatkov.
-732	Missing input data.	Vnos potnika neuspešen zaradi manjkajočih podatkov.
0	OK	

## 8.2 GETPASSENGER

Metodo GetPassenger uporabljamo za pridobivanje podatkov o potnikih iz zalednega sistema.

<b>Deklaracija metode</b>	<b>Int GetPassenger (int iPid [IN], int iResultCode [OUT], string sResultDescription [OUT], Passenger mPassengerData [OUT] )</b>		
<b>Parameter</b>	<b>Tip parametra</b>	<b>Obveznost parametra / Vhodni -Izhodni</b>	<b>Opis parametra</b>
iPid	Integer	No / Input	Identifikator potnika.
sIdentityNumber	String	No/ Input	Emšo potnika.
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.
mPassengerData	Passenger *	- / Output	Struktura s podatki potnika.

\* Struktura **Passenger** je definirana v poglavju definicije podatkovnih struktur.

Možne kode rezultata za metodo **GetPassenger**.

iResultCode	sResultDescription	Extra description
-716	Passenger not found.	Potnik ni bil najden v sistemu.
0	OK	

### 8.3 SETCARD

Metodo SetCard uporabljamo za upravljanje podatkov na karticah. Metoda nam omogoča vnos nove kartice v zaledni sistem.

<b>Deklaracija metode</b>	<b>Int SetCard (Card mCardData [IN], int iResultCode [OUT], string sResultDescription [OUT] )</b>		
<b>Parameter</b>	<b>Tip parametra</b>	<b>Obveznost parametra / Vhodni -Izhodni</b>	<b>Opis parametra</b>
mCardData	Card*	Yes / Input	Struktura s podatki kartice.
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.

\* Struktura **Card** je definirana v poglavju definicije podatkovnih struktur.

Možne kode rezultata za metodo **SetCard**.

iResultCode	sResultDescription	Extra description
-721	Invalid input data.	Vnos kartice neuspešen zaradi nepravilnih podatkov.
-732	Missing input data.	Vnos kartice neuspešen zaradi manjkajočih podatkov.
0	OK	Uspešen vnos kartice.

## 8.4 UPDATECARD

Metodo UpdateCard uporabljamo za upravljanje podatkov na karticah. Metoda nam omogoča urejanje podatkov na kartici.

Deklaracija metode	<b>Int UpdateCard (Card mCardData [IN], int iResultCode [OUT], string sResultDescription [OUT] )</b>		
Parameter	Tip parametra	Obveznost parametra / Vhodni -Izhodni	Opis parametra
mCardData	Card*	Yes / Input	Struktura s podatki kartice.
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.

\* Struktura **Card** je definirana v poglavju definicije podatkovnih struktur.

Možne kode rezultata za metodo **UpdateCard**.

iResultCode	sResultDescription	Extra description
-721	Invalid input data.	Posodobitev kartice neuspešna zaradi nepravilnih podatkov.
-732	Missing input data.	Posodobitev kartice neuspešna zaradi manjkajočih podatkov.
0	OK	Uspešna posodobitev kartice.

## 8.5 GETCARD

Metodo GetCard uporabljamo za pridobivanje podatkov o karticah. Metoda nam vrne podatke kartice na osnovi CUID številke, ki je identifikator kartice.

Deklaracija metode	<b>Int GetCard (Long ICUID [IN], int iResultCode [OUT], string sResultDescription [OUT], Card mCardData [OUT] )</b>		
Parameter	Tip parametra	Obveznost parametra / Vhodni -Izhodni	Opis parametra
ICUID	Long	Yes / Input	Identifikator kartice.
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.
mCardData	Card*	- / Output	Struktura s podatki kartice.

\* Struktura **Card** je definirana v poglavju definicije podatkovnih struktur.

Možne kode rezultata za metodo **GetCard**.

iResultCode	sResultDescription	Extra description
-713	Card not found.	Kartica ni bila najdena v sistemu.
0	OK	



## 8.6 UPDATECARDPASSENGERSTATUS

Metodo UpdateCardStatus uporabljamo za upravljanje podatkov o pravicah potnika. Metoda nam omogoča vnos in urejanje pravic potnika v zalednem sistemu.

<b>Deklaracija metode</b>	<b>Int UpdateCardPassengerStatus</b> (Long ICUID [IN], <b>CardStatus[]</b> mCardStatusData [IN], <b>int</b> iResultCode [OUT], <b>string</b> sResultDescription [OUT] )		
<b>Parameter</b>	<b>Tip parametra</b>	<b>Obveznost parametra / Vhodni -Izhodni</b>	<b>Opis parametra</b>
ICUID	Long	Yes / Input	Identifikator potnikove kartice
mCardStatusData	CardPassengerStatus[]*	No / Input	Polje struktur s podatki o pravicah potnika.
mSubsidyApplication	SubsidyApplication[]*	No / Input	Polje struktur s podatki o subvencijah potnika
mWorkStatus	Workstatus[]*	No / Input	Polje struktur s podatki o službenih statusih potnika
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.

\* **CardPassengerStatus** struktura je opisana pri definiciji podatkovnih struktur pri transakcijskih metodah spletne storitve. **SubsidyApplication** struktura je definirana v poglavju definicije podatkovnih struktur. **WorkSatus** struktura je definirana v poglavju definicije podatkovnih struktur.

Možne kode rezultata za metodo **UpdateCardPassengerStatus**.

iResultCode	sResultDescription	Extra description
-721	Invalid input data.	Vnos pravice neuspešen zaradi nepravilnih podatkov.
-732	Missing input data.	Vnos pravice neuspešen zaradi manjkajočih podatkov.
0	OK	Uspešen vnos pravice v zaledni sistem.

## 8.7 GETCARDPASSENGERSTATUS

Metodo GetCardStatus uporabljamo za pridobivanje podatkov o pravicah potnikov. Metoda nam vrne podatke o pravicah potnika na kartici na osnovi CUID številke, ki je identifikator potnikove kartice.

Deklaracija metode		<b>Int GetCardPassengerStatus (Long lCUID[IN], int iResultCode[OUT], string sResultDescription [OUT], CardStatus[] mCardStatusData[OUT] )</b>	
Parameter	Tip parametra	Obveznost parametra / Vhodni -Izhodni	Opis parametra
lCUID	Long	Yes / Input	Identifikator potnikove kartice.
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.
mCardStatusData	CardPassengerStatus[]*	- / Output	Polje struktur s podatki o pravicah potnika.
mSubsidyApplication	SubsidyApplication[]*	- / Output	Polje struktur s podatki o subvencijah potnika
mWorkStatus	Workstatus[]*	- / Output	Polje struktur s podatki o službenih statusih potnika

\* **CardPassengerStatus** struktura je opisana pri definiciji podatkovnih struktur pri transakcijskih metodah spletne storitve. **SubsidyApplication** struktura je definirana v poglavju definicije podatkovnih struktur. **WorkSatus** struktura je definirana v poglavju definicije podatkovnih struktur.

Možne kode rezultata za metodo **GetCardPassengerStatus**.

iResultCode	sResultDescription	Extra description
-713	Card not found.	Kartica ni bila najdena v sistemu.
0	OK	

## 8.8 GETRELATION

Metodo GetRelation uporabljamo za pridobivanje podatkov o relacijah med dvema krajema. Metoda ima obvezna oba vhodna parametra **iStartStationId** in **iEndStationId**. Preko izhodnih parametrov metoda vrne podatke o relaciji in postajah ki so na njej.

<b>Deklaracija metode</b>	<b>Int GetRelation (int sStartStationId[IN], int sEndStationId[IN], int[] iIntermediateStations[IN], int iResultCode[OUT], string sResultDescription [OUT], Relation mRelationData [OUT] )</b>		
Parameter	Tip parametra	Obveznost parametra / Vhodni -Izhodni	Opis parametra
iStartStationId	Integer	yes / Input	Identifikator začetne postaje.
iEndStationId	Integer	yes / Input	Identifikator končne postaje.
iIntermediateStations	Integer[]	no / Input	Polje identifikatorjev vmesnih postaj.
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.
mRelationData	Relation*	- / Output	Struktur s podatki o relaciji.

\* Struktura **Relation** je definirana v poglavju definicije podatkovnih struktur.

Možne kode rezultata za metodo **GetRelation**.

iResultCode	sResultDescription	Extra description
-721	Missing input data.	Manjkajo vhodni podatki.
-736	Relation not found.	Relacija ni bila najdena v sistemu.
0	OK	

## 8.9 GETCARDPRODUCTS

Metodo GetCardProducts uporabljamo za pridobivanje podatkov o produktih potnika iz zalednega sistema. Metoda nam vrne podatke o produktih potnika na osnovi CUID številke, ki je identifikator potnikove kartice.

<b>Deklaracija metode</b>	<b>Int GetCardProducts (Long ICUID [IN], int iResultCode [OUT], string sResultDescription [OUT], CardProduct[] mCardProductData[OUT] )</b>		
<b>Parameter</b>	<b>Tip parametra</b>	<b>Obveznost parametra / Vhodni -Izhodni</b>	<b>Opis parametra</b>
ICUID	Long	Yes / Input	Identifikator potnikove kartice.
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.
mCardProductData	CardProduct[] *	- / Output	Polje struktur o produktih potnika.

\* Struktura **CardProduct** je opisana pri definiciji podatkovnih struktur pri transakcijskih metodah spletne storitve.

Možne kode rezultata za metodo **GetCardProducts**.

iResultCode	sResultDescription	Extra description
-713	Card not found.	Kartica ni bila najdena v sistemu.
0	OK	

## 8.10 GETPRODUCTPRICE

Metodo GetProductPrice uporabljamo za pridobivanje podatkov o cenah produktov iz zalednega sistema.

<b>Deklaracija metode</b>	<b>Int GetProductPrice (ProductPriceIn mProductPriceIn [IN], int iResultCode [OUT], string sResultDescription [OUT], ProductPriceOut mProductPriceOut [OUT] )</b>		
<b>Parameter</b>	<b>Tip parametra</b>	<b>Obveznost parametra / Vhodni -Izhodni</b>	<b>Opis parametra</b>
mProductPriceIn	ProductPriceIn*	Yes / Input	Struktura z vhodnimi podatki o produktu.
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.
mProductPriceOut	ProductPriceOut*	- / Output	Struktura z podatki o ceni produkta.

\*Strukturi **ProductPriceIn** in **ProductPriceOut** sta definirani v poglavju definicije podatkovnih struktur.

Možne kode rezultata za metodo **GetProductPrice**.

iResultCode	sResultDescription	Extra description
-721	Invalid input data.	Nepravilni vhodni podatki.
-732	Missing input data.	Manjkajo vhodni podatki.
0	OK	Uspešna pridobitev cene produkta.

## 8.11 SETPOS

Metodo SetPos uporabljamo za upravljanje podatkov o prodajnih mestih. Metoda omogoča vnos novih prodajnih mest v zaledni sistem. Prav tako lahko s pomočjo metode podatke obstoječega prodajnega mesta spreminjamo oziroma posodabljam.

Deklaracija metode	<b>Int SetPos (Pos mPosData [IN], int iResultCode [OUT], string sResultDescription [OUT], int iPosId [OUT])</b>		
Parameter	Tip parametra	Obveznost parametra / Vhodni -Izhodni	Opis parametra
mPosData	Pos*	Yes / Input	Struktura s podatki prodajnega mesta.
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.
iPosId	Integer	- / Output	Identifikator prodajnega mesta.

\* Struktura **Pos** je definirana v poglavju definicije podatkovnih struktur.

Možne kode rezultata za metodo **SetPos**.

iResultCode	sResultDescription	Extra description
-721	Invalid input data.	Nepravilni vhodni podatki.
-732	Missing input data.	Manjkajo vhodni podatki.
0	OK	

## 8.12 GETPOS

Metodo GetPos uporabljamo za pridobivanje podatkov o prodajnih mestih iz zalednega sistema.

<b>Deklaracija metode</b>	<b>Int GetPos (int iPosId [IN], int iResultCode [OUT], string sResultDescription [OUT], Pos mPosData [OUT] )</b>		
<b>Parameter</b>	<b>Tip parametra</b>	<b>Obveznost parametra / Vhodni -Izhodni</b>	<b>Opis parametra</b>
iPosId	Integer	Yes / Input	Identifikator prodajnega mesta.
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.
mPosData	Pos *	- / Output	Struktura s podatki prodajnega mesta.

\* Struktura **Pos** je definirana v poglavju definicije podatkovnih struktur.

Možne kode rezultata za metodo **GetPos**.

iResultCode	sResultDescription	Extra description
-717	Pos not found.	Prodajno mesto ni bilo najden v zalednem sistemu.
0	OK	

### 8.13 SETTERMINAL

Metodo SetTerminal uporabljamo za upravljanje podatkov o terminalih. Metoda omogoča vnos novih terminalov v zaledni sistem. Prav tako lahko s pomočjo metode podatke obstoječega terminala spreminjamo oziroma posodabljam.

<b>Deklaracija metode</b>	<b>Int SetTerminal (Terminal mTerminalData [IN], int iResultCode [OUT], string sResultDescription [OUT], int iTid [OUT])</b>		
<b>Parameter</b>	<b>Tip parametra</b>	<b>Obveznost parametra / Vhodni -Izhodni</b>	<b>Opis parametra</b>
mTerminalData	Terminal*	Yes / Input	Struktura s podatki terminala.
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.
iTid	Integer	- / Output	Identifikator terminala.

\* Struktura **Terminal** je definirana v poglavju definicije podatkovnih struktur.

Možne kode rezultata za metodo **SetTerminal**.

iResultCode	sResultDescription	Extra description
-721	Invalid input data.	Vnos terminala neuspešen zaradi nepravilnih vhodnih podatkov.
-732	Missing input data.	Vnos terminala neuspešen zaradi manjkajočih vhodnih podatkov.
0	OK	



## 8.14 GETTERMINAL

Metodo GetTerminal uporabljamo za pridobivanje podatkov o terminalih.

<b>Deklaracija metode</b>	<b>Int GetTerminal (int iTId [IN], int iResultCode [OUT], string sResultDescription [OUT], Terminal mTerminalData [OUT] )</b>		
<b>Parameter</b>	<b>Tip parametra</b>	<b>Obveznost parametra / Vhodni -Izhodni</b>	<b>Opis parametra</b>
iTId	Integer	Yes / Input	Identifikator terminala.
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.
mTerminalData	Terminal*	- / Output	Struktura s podatki terminala.

\* Struktura **Terminal** je definirana v poglavju definicije podatkovnih struktur.

Možne kode rezultata za metodo **GetTerminal**.

iResultCode	sResultDescription	Extra description
-501	Terminal not found.	Terminal ni bil najden v sistemu.
0	OK	

## 8.15 SETBLACKLIST

Metodo SetBlackList uporabljamo za dodajanje kartice na črno listo zalednega sistema.

Deklaracija metode		<b>Int SetBlackList</b> ( <b>long</b> ICUID [IN], <b>int</b> iResultCode [OUT], <b>string</b> sResultDescription [OUT], <b>int</b> iBlackListId [OUT])	
Parameter	Tip parametra	Obveznost parametra / Vhodni -Izhodni	Opis parametra
ICUID	Long	Yes / Input	CUID kartice dodane na črno listo.
iBLCause	Integer	Yes / Input	Vzrok dodajanja kartice na črno listo: 1=duplikat, 2=izgubljena, 3=kontrolor, 4=zastaranje, 5=uničena.
iUserId	Integer	Yes / Input	Identifikator uporabnika
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.
iBlackListId	Integer	- / Output	Identifikator zadnje aktivne verzije črne liste.

Možne kode rezultata za metodo **SetBlackList**.

iResultCode	sResultDescription	Extra description
-713	Card not found.	Kartica ni bil najdena v sistemu.
0	OK	

## 8.16 SETEXECUTABLELIST

Metodo SetExecutableList uporabljamo za določitev seznama kartic izvršne liste, na katerih je potrebno izvesti eno izmed naslednjih opravil: Prenos vozovnice na kartico, brisanje vozovnice na kartici, nastavitev novega statusa na kartici, nastavitev novega produkta na kartici, začasno onemogočanje uporabe kartice.

<b>Deklaracija metode</b>	<b>Int</b> SetExecutableList( <b>ExecutableData</b> []) mExecutableData [IN], <b>int</b> iResultCode [OUT], <b>string</b> sResultDescription [OUT])		
<b>Parameter</b>	<b>Tip parametra</b>	<b>Obveznost parametra / Vhodni -Izhodni</b>	<b>Opis parametra</b>
mExecutableData	<b>ExecutableData</b> [] *	Yes / Input	Polje struktur s podatki o karticah.
iResultCode	<b>Integer</b>	- / Output	Tip kode odgovora.
sResultDescription	<b>String</b>	- / Output	Opis tipa kode odgovora.

\* Struktura **ExecutableData** je definirana v poglavju definicije podatkovnih struktur.

Možne kode rezultata za metodo **SetExecutableList**.

iResultCode	sResultDescription	Extra description
-721	Invalid input data.	Vnos izvršne liste neuspešen zaradi nepravilnih vhodnih podatkov.
-732	Missing input data.	Vnos izvršne liste neuspešen zaradi manjkajočih vhodnih podatkov.
0	OK	

## 8.17 DELEXECUTABLELIST

Metodo DelExecutableList uporabljamo odstranitev posameznega neizvršenega opravila iz izvršne liste.

<b>Deklaracija metode</b>	<b>Int</b> SetExecutableList( <b>DelExecutableData</b> [] mDelExecutableData [IN], <b>int</b> iResultCode [OUT], <b>string</b> sResultDescription [OUT])		
<b>Parameter</b>	<b>Tip parametra</b>	<b>Obveznost parametra</b> <b>Vhodni -Izhodni</b>	<b>Opis parametra</b>
mDelExecutableData	<b>DelExecutableData</b> []*	Yes / Input	Polje struktur s podatki o odstranjenih opravilih.
iResultCode	<b>Integer</b>	- / Output	Tip kode odgovora.
sResultDescription	<b>String</b>	- / Output	Opis tipa kode odgovora.

\* Struktura **DelExecutableData** je definirana v poglavju definicije podatkovnih struktur.

Možne kode rezultata za metodo **DelExecutableList**.

iResultCode	sResultDescription	Extra description
-721	Invalid input data.	Vnos zahteve neuspešen zaradi nepravilnih vhodnih podatkov.
-732	Missing input data.	Vnos zahteve neuspešen zaradi manjkajočih vhodnih podatkov.
0	OK	

## 8.18 REGISTRATIONTERMINAL

Metodo RegistrationTerminal uporabljamo za registracijo terminala v IJPP sistemu. Metoda se poklice z uporabo registracijskega certifikata, ki je namenjen samo za to metodo oziroma registracijo terminala. Metoda registrira terminal s poslano serijsko številko in mu pošlje nazaj njegov TID ter dodeli njegov SSL certifikat.

<b>Deklaracija metode</b>	<b>Int RegistrationTerminal(string sSerialNumber [IN], int iResultCode [OUT], string sResultDescription [OUT], int iTid[OUT], SSLCertificate*[] sSSLCertificate[OUT])</b>		
<b>Parameter</b>	<b>Tip parametra</b>	<b>Obveznost parametra / Vhodni -Izhodni</b>	<b>Opis parametra</b>
sSerialNumber	String	Yes / Input	Serijska številka terminala
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.
iTid	Integer	- / Output	Identifikator terminala.
mSSLCertificate	SSLCertificate*[]	- / Output	Polje struktur SSL certifikatov terminala.

\* Struktura **SSLCertificate** je definirana v poglavju definicije podatkovnih struktur.

Možne kode rezultata za metodo **RegistrationTerminal**.

iResultCode	sResultDescription	Extra description
-501	Terminal not found.	Terminal ni bil najden v sistemu.
-721	Invalid input data.	Napačni vhodni podatki.
-732	Missing input data.	Manjkajo vhodni podatki.
0	OK	

## 8.19 ACTIVATIONSAM

Metodo ActivationSAM uporabljamo za aktivacijo SAM modula v IJPP sistemu. Metoda se pokliče z uporabo lastnega unikatnega SSL certifikata, ki se pridobi z registracijo terminala. Metoda omogoča aktivacijo SAM modula tako da na osnovi vhodnih podatkov o terminalu in serijske številke SAM modula vrne avtentikacijski ključ za aktivacijo SAM modula. Avtentikacijski ključ za aktivacijo SAM modula je šifriran z RSA algoritmom, kjer se uporabi ključ lastnega certifikata. Postopek dešifriranja avtentikacijskega ključa SAM modula se na klient strani izvede z RSA algoritmom in privatnim ključem lastnega certifikata. Privatni ključ lastnega certifikata je zapisan v obliki base64.

<b>Deklaracija metode</b>	<b>Int ActivationSAM( int iTId [IN], string sTerminalSerialNumber [IN], string sSAMSerialNumber [IN], int iResultCode [OUT], string sResultDescription [OUT], string sSAMActivateKey [OUT]))</b>		
<b>Parameter</b>	<b>Tip parametra</b>	<b>Obveznost parametra / Vhodni -Izhodni</b>	<b>Opis parametra</b>
iTId	Integer	Yes / Input	Identifikator terminala.
sTerminalSerialNumber	String	Yes / Input	Serijska številka terminala
sSAMSerialNumber	String	Yes / Input	Serijska številka SAM modula
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.
sSAMAuthetnticationKey	String	- / Output	Avtentikacijski Ključ SAM Modula.

Možne kode rezultata za metodo **ActivationSAM**.

iResultCode	sResultDescription	Extra description
-501	Terminal not found.	Terminal ni bil najden v sistemu.
-721	Invalid input data.	Napačni vhodni podatki.
-732	Missing input data.	Manjkajo vhodni podatki.
0	OK	

## 8.20 GETDUPLICATEDATA

Metoda GetDuplicateData uporabljamo pri izdelavi duplikata kartice. Metoda se pokliče z uporabo lastnega unikatnega SSL certifikata, ki se pridobi z registracijo terminala. Metoda omogoča pridobivanje vseh podatkov kartice, ki jih potrebujemo za izdelavo duplikata.

<b>Deklaracija metode</b>	<b>Int GetDuplicateData( int iTid [IN], Long ICUID [IN], DuplicateData mDuplicateData, int iResultCode [OUT], string sResultDescription [OUT])</b>		
<b>Parameter</b>	<b>Tip parametra</b>	<b>Obveznost parametra / Vhodni -Izhodni</b>	<b>Opis parametra</b>
ICUID	Long	Yes / Input	Identifikator kartice
mDuplicateData	DuplicateData	- / Output	Podatki za izdelavo duplikata
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.

Možne kode rezultata za metodo **GetDuplicateData**.

iResultCode	sResultDescription	Extra description
-713	Card not found.	Kartica ni bil najdena v sistemu.
-721	Invalid input data.	Napačni vhodni podatki.
-732	Missing input data.	Manjkajo vhodni podatki.
0	OK	

## 8.21 GETTRANSACTIONS

Metodo GetTransactions uporabljamo za pridobivanje podatkov o IJPP transakcijah, ki so bile izvedene pri določenem prevozniku, ki ga določa vhodni podatek iCompanyId. Terminal sproži poizvedbo o IJPP transakcijah, izvedenih v določenem časovnem obdobju, definiranim s parametroma dtDateFrom in dtDateUntil. Metoda vrne seznam IJPP transakcij, ki so jih terminali prevoznika v definiranem časovnem območju poslali na zaledni sistem z metodo ProcessTransactions. Metoda GetTransactions lahko v enem klicu vrne omejeno število IJPP transakcij, ki so razvrščene naraščajoče glede na vrednost ITransactionExecutionId. V primeru, ko imamo več IJPP transakcij, kot jih metoda lahko vrne, moramo pridobiti transakcijske podatke v več korakih. Če niso bile vrnjene vse transakcije, metoda vrne ustrezen iResultCode. V tem primeru moramo v naslednjem koraku metodi posredovati TransactionExecutionId zadnje transakcije, ki je bila v prejšnjem seznamu (ILastTransactionExecutionId) in dobimo naslednji seznam transakcij, ki so bile izvedene po transakciji ILastTransactionExecutionId. Korake ponavljamo tako dolgo dokler ne prenesemo vseh transakcij.

<b>Deklaracija metode</b>	<b>Int GetTransactions (int iTid [IN], DateTime dtDateFrom [IN], DateTime dtDateUntil [IN], Long ILastTransactionExecutionId [IN], int iCompanyId[IN], int iResultCode[OUT], string sResultDescription [OUT], IJPPTransaction mIJPPTransactionData [OUT] )</b>		
Parameter	Tip parametra	Obveznost parametra / Vhodni -Izhodni	Opis parametra
dtDateFrom	DateTime	Yes / Input	Začetni čas časovnega obdobja izvedenih transakcij na terminalu (začetni čas je vključen v časovno obdobje).
dtDateUntil	DateTime	Yes / Input	Končni čas časovnega obdobja izvedenih transakcij na terminalu (končni čas ni vključen v časovno obdobje).
ILastTransactionExecutionId	Long	- / Input	Zadnji TransactionExecutionId prejšnjega seznama.
iCompanyId	Integer	- / Input	Identifikator prevoznika
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.
mIJPPTransactionData	IJPPTransaction*[]	- / Output	Seznam izvedenih IJPP transakcij

\* Struktura **IJPPTransaction** je definirana v poglavju definicije podatkovnih struktur transakcijskih metod spletne storitve IJPP.



Možne kode rezultata za metodo **GetTransactions**.

iResultCode	sResultDescription	Extra description
-501	Terminal not found.	Terminal ni bil najden v sistemu.
0	OK	
1	OK – Seznam ni prenešen v celoti	

## 8.22 GETSUBTRANSACTIONSBySUBSIDY

Metodo GetSubTransactionsBySubsidy uporabljamo za pridobivanje podatkov o subvencioniranih transakcijah, ki so bile izvedene pri določenem prevozniku, ki ga določa vhodni podatek iCompanyId. Terminal sproži poizvedbo o transakcijah, izvedenih v določenem časovnem obdobju (največ 2 uri), definiranim s parametroma dtDateFrom in dtDateUntil. Metoda vrne seznam transakcij, ki so jih terminali prevoznika v definiranem časovnem območju poslali na zaledni sistem z metodo ProcessTransactions.

<b>Deklaracija metode</b>	<b>Int GetSubTransactionsBySubsidy (, DateTime dtDateFrom [IN], DateTime dtDateUntil [IN], string sSubsidyId [IN], int iCompanyId[IN], int iResultCode[OUT], string sResultDescription [OUT], SubTransaction mSubTransactionData [OUT] )</b>		
<b>Parameter</b>	<b>Tip parametra</b>	<b>Obveznost parametra / Vhodni -Izhodni</b>	<b>Opis parametra</b>
dtDateFrom	DateTime	Yes / Input	Začetni čas časovnega obdobja izvedenih transakcij na terminalu (začetni čas je vključen v časovno obdobje).
dtDateUntil	DateTime	Yes / Input	Končni čas časovnega obdobja izvedenih transakcij na terminalu (končni čas ni vključen v časovno obdobje).
sSubsidyId	String	Yes / Input	Identifikator vloge
iCompanyId	Integer	Yes / Input	Identifikator prevoznika
iResultCode	Integer	- / Output	Tip kode odgovora.
sResultDescription	String	- / Output	Opis tipa kode odgovora.
mSubTransactionData	SubTransaction*[]	- / Output	Seznam izvedenih Sub transakcij

\* Struktura **SubTransaction** je definirana v poglavju definicije podatkovnih struktur transakcijskih metod spletne storitve IJPP.

Možne kode rezultata za metodo **GetSubTransactionsBySubsidy**.

iResultCode	sResultDescription	Extra description
-501	Terminal not found.	Terminal ni bil najden v sistemu.
0	OK	

## 8.23 GETCUIDSBYIDENTITYNUMBER

Metoda GetCUIDsByIdentityNumber vrne seznam CUID za potnika glede na vpisani EMŠO.

Deklaracija metode	int GetCUIDsByIdentityNumber ( <b>string</b> EMSO [IN], <b>long[]</b> CUID [OUT], <b>int</b> rezultat [OUT], <b>string</b> rezultatOpis [OUT])		
Parameter	Tip parametra	Obveznost parametra / vhodni - izhodni	Opis parametra
EMSO	string	DA / vhodni	EMŠO potnika, za katerega se pridobiva kartice
CUID[]	long	- / izhodni	Seznam CUID-ov za potnika
rezultat	integer	- / izhodni	Tip kode odgovora
rezultatOpis	string	- / izhodni	Opis tipa odgovora

iResultCode	sResultDescription	Extra description
-721	Invalid input data	Napačni vhodni podatek – potnik ne obstaja.
0	V redu	

GetSubTransactionsBySubsidy

## 8.24 GETIJPPCATALOGS

Metoda GetIJPPCatalogs vrne vrednosti šifranta za zahtevan tip šifranta.

Deklaracija metode	int GetIJPPCatalogs (int iCatalogType [IN], CatalogData[] mCatalogData [OUT], int rezultat [OUT], string rezultatOpis [OUT])		
Parameter	Tip parametra	Obveznost parametra / vhodni - izhodni	Opis parametra
iCatalogType	integer	DA / vhodni	Tip kataloga
mCatalogData	CatalogData[]*	- / izhodni	Seznam zahtevanih šifrantov
rezultat	integer	- / izhodni	Tip kode odgovora
rezultatOpis	string	- / izhodni	Opis tipa odgovora

\* Struktura **CatalogData** je definirana v poglavju definicije podatkovnih struktur.

iResultCode	sResultDescription	Extra description
-721	Invalid input data	Napačni vhodni podatek - neustrezni tip kataloga
0	V redu	

\*CatalogData je abstractni razred, ki lahko vsebuje naslednje podrazrede(strukture):

- BlacklistCatalog – struktura šifranta z razlogi za dodajanje na črno listo (iCatalogType -1)
- CardStatusCatalog – struktura šifranta s statusi kartice (iCatalogType -2)
- WorkStatusCatalog – struktura šifranta z delovnimi statusi kartice (iCatalogType -3)
- StatusData – struktura s podatki statusa kartice (iCatalogType -4)
- IJPPTransactionTypeCatalog – struktura s podatki tipov IJPP transakcij (iCatalogType -5)
- OfflineStatusCatalog – Struktura z opisi statusov IJPP transakcij (OfflineStatus) (iCatalogType -6)
- SifrantObcin - Struktura šifranta z podatki občin (iCatalogType -7)
- SifrantNaselij - Struktura šifranta z podatki naselij (iCatalogType -8)
- SifrantUpravljalcev - Struktura šifranta z podatki upravljalcev (iCatalogType -9)
- TerminalTypeCatalog- Struktura šifranta z tipi terminalov (iCatalogType -10)
- TerminalSubTypeCatalog- Struktura šifranta z podtipi terminalov (iCatalogType -11)

## 8.25 GETIJPPCATALOGS\_PLADENJ

Metoda GetIJPPCatalogs\_Pladenj je identična metodi GetIJPPCatalogs, sicer se uporablja za potrebe **Pladenj** sistema.

Deklaracija metode	int GetIJPPCatalogs_Pladenj (int iCatalogType [IN], CatalogData_Pladenj []mCatalogData [OUT], int rezultat [OUT], string rezultatOpis [OUT])		
Parameter	Tip parametra	Obveznost parametra / vhodni - izhodni	Opis parametra
iCatalogType	integer	DA / vhodni	Tip kataloga
mCatalogData	CatalogData_Pladenj []*	- / izhodni	Seznam zahtevanih šifrantov
rezultat	integer	- / izhodni	Tip kode odgovora
rezultatOpis	string	- / izhodni	Opis tipa odgovora

\* Struktura **CatalogData\_Pladenj** je definirana v poglavju definicije podatkovnih struktur.

iResultCode	sResultDescription	Extra description
-721	Invalid input data	Napačni vhodni podatek - neustrezni tip kataloga
0	V redu	

\*CatalogData\_Pladenj je abstractni razred, ki lahko vsebuje naslednje podrazrede(strukture):

- BlacklistCatalog\_Pladenj – struktura šifranta z razlogi za dodajanje na črno listo (iCatalogType -1)
- CardStatusCatalog\_Pladenj – struktura šifranta s statusi kartice (iCatalogType -2)
- WorkStatusCatalog\_Pladenj – struktura šifranta z delovnimi statusi kartice (iCatalogType -3)
- StatusData\_Pladenj – struktura s podatki statusa kartice (iCatalogType -4)
- IJPPTransactionTypeCatalog\_Pladenj – struktura s podatki tipov IJPP transakcij (iCatalogType -5)
- OfflineStatusCatalog\_Pladenj – Struktura z opisi statusov IJPP transakcij (OfflineStatus) (iCatalogType -6)
- SifrantObcin\_Pladenj - Struktura šifranta z podatki občin (iCatalogType -7)
- SifrantNaselij\_Pladenj - Struktura šifranta z podatki naselij (iCatalogType -8)
- SifrantUpravljalcev\_Pladenj - Struktura šifranta z podatki upravljalcev (iCatalogType -9)

- TerminalTypeCatalog\_Pladenj - Struktura šifranta z tipi terminalov (iCatalogType -10)
- TerminalSubTypeCatalog\_Pladenj - Struktura šifranta z podtipi terminalov (iCatalogType -11)

## 8.26 DEFINICIJE PODATKOVNIH STRUKTUR

**Passenger** struktura vsebuje naslednje parametre:

struktura Passenger		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
sFirstName	String	Ime potnika.
sLastName	String	Priimek potnika.
sAddress	String	Naslov potnika.
sCity	String	Mesto potnika.
iZip	Integer	Pošta potnika.
sCountry	String	Država potnika.
sEmail	String	Elektronska pošta potnika.
sMSISDN	String	Telefonska številka potnika.
sIdentityNumber	String	EMŠO potnika.
iStatus	Integer	Status potnika.
dtBirthDate	DateTime	Rojstni datum potnika.

**Card** struktura vsebuje naslednje parametre:

struktura Card		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
ICUID	Long	Identifikator kartice.
dtInitDate	DateTime	Datum inicializacije kartice.
dtValidToDate	DateTime	Datum veljavnosti kartice.
iCardType	Integer	Tip kartice.
dtPersonificationDate	DateTime	Datum personifikacije kartice.

iPid	Integer, NULL	Identifikator potnika.
iStatus	Integer	Status kartice.
dtLastStatusChangeDate	DateTime	Datum zadnje spremembe.
iBlacklisted	Integer, NULL	Parameter določa ali je kartica na črni listi. (NULL, 0)- kartica ni na črni listi 1 – kartica je na črni listi
iBlacklistedChangeReasonId	Int, NULL	Vzrok dodajanja kartice na črno listo: 1=duplikat, 2=izgubljena, 3=kontrolor, 4=zastaranje, 5=uničena.
dtBlacklistedTime	DateTime, NULL	Čas postavitve kartice na črno listo

**Relation** struktura vsebuje naslednje parametre:

struktura Relation		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iStartStationId	Integer	Identifikator začetne postaje.
iEndStationId	Integer	Identifikator končne postaje.
dDistance	Decimal	Dolžina linije v kilometrih.
iNumberOfStations	Integer	Število postaj na liniji.
iStID	Integer[]	Polje Id-jev vmesnih postaj na liniji.

**ProductPriceln** struktura vsebuje naslednje parametre:

struktura ProductPriceln		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iProductId	Integer	identifikator produkta
iTariffClassId	Integer	identifikator tarifnega razreda
iTicketStatus	Integer	Identifikator statusa (pravica)
iTariffLocationId	Integer	Identifikator lokacije tarife

**ProductPriceOut** struktura vsebuje naslednje parametre:

struktura ProductPriceOut		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iTariffId	Integer	identifikator tarife
fPriceValue	Float	cena produkta

**Pos** struktura vsebuje naslednje parametre:

struktura Pos		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iPosId	Integer	Identifikator prodajnega mesta.
iCompanyId	Integer	Identifikator podjetja, ki mu pripada prodajno mesto.
sName	String	Ime prodajnega mesta.
sAddress	String	Naslov prodajnega mesta.
sCity	String	Mesto prodajnega mesta.
sZIP	String	Poštna številka prodajnega mesta.
sCountry	String	Država prodajnega mesta.
sPhone	String	Telefonska številka prodajnega mesta.
sEmail	String	Elektronska pošta prodajnega mesta.
Longitude	Decimal, NULL	Geografski položaj prodajnega mesta.
Latitude	Decimal, NULL	Geografski položaj prodajnega mesta.
sSubBusOperatorCode	String	Šifra prevoznika za subvencije
sSubUserName	String	Uporabniško ime prevoznika za subvencije
sSubPassword	String	Geslo prevoznika za subvencije



**Terminal** struktura vsebuje naslednje parametre:

struktura Terminal		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iTId	Integer	Identifikator terminala.
sSerialNumber	String	Serijska številka terminala.
iTerminalTypeld	Integer	Identifikator tipa terminala.
iTerminalSubTypeld	Integer, NULL	Identifikator podtipa terminala.
iStatusId	Integer	Identifikator statusa terminala.
dtProductionDate	DateTime	Datum produkcije terminala.
sSoftwareVersion	String	Verzija produkcije terminala.
dtSoftwareRevisionDate	DateTime, NULL	Datum posodobitve terminala.
sSoftwareRevision	String	Verzija posodobitve terminala.
sNotes	String	Opombe
iOwnerId	Integer	Identifikator lastnika terminala.
iMerchantId	Integer, NULL	Identifikator trgovca ki mu terminal pripada.
iPosId	Integer, NULL	Identifikator prodajnega mesta kjer je terminal nameščen.

**SSLCertificate** struktura vsebuje naslednje parametre:

struktura SSLCertificate		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
sSSLCertFileName	String	Ime certifikata
bSSLCertFileContent	byte[ ]	Vsebina certifikata

**ExecutableData** struktura je opisana pri definiciji podatkovnih struktur pri transakcijskih metodah spletne storitve.

**DelExecutableData** struktura vsebuje naslednje parametre:

struktura DelExecutableData		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
ICUID	Long	Identifikator potnikove kartice
iAction	Integer	Identifikator opravila na izvršni listi, ki ga želimo odstraniti
ITransactionExecutionId	Long, NULL	Referenca vozovnice ali statusa

**SubTransaction** struktura vsebuje naslednje parametre:

struktura SubTransaction		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
ITransactionExecutionId	Long	Identifikator transakcije
iTransactionType	Integer	tip transakcije (natančneje opisano v poglavju: Tipi IJPP transakcij)
lRefTransactionExecutionId	Long, NULL	referenčni identifikator transakcije
dtStartTime	DateTime	čas izvedbe transakcije na terminalu
sSubsidyId	String, NULL	Identifikator subvencije
iTariffProductId	Integer, NULL	identifikator vrste vozovnice
iTicketStatus	Integer, NULL	Identifikator statusa (pravica)
iTariffClassId	Integer, NULL	identifikator tarifnega razreda
fPriceValue	Float, NULL	cena vozovnice
fSubFullPrice	Float, NULL	Polna cena subvencioniranih vozovnic
fSubFullPriceReported	Float, NULL	Poročana cena subvencioniranih vozovnic
fSubPayment	Float, NULL	Plačilo subvencioniranih vozovnic
iTariffLocationId	Integer, NULL	Identifikator lokacije uporabe
sCompanyDomain	String, NULL	Domena podjetja
sCompanyCode	String, NULL	Šifra podjetja
iCompanyId	Integer, NULL	Identifikator podjetja
sPosName	String, NULL	Ime prodajnega mesta
iPosId	Integer, NULL	Identifikator prodajnega mesta

iStartStationId	Integer, NULL	Identifikator začetnega postajališča.
iEndtStationId	Integer, NULL	Identifikator končnega postajališča.
iDistance	Integer, NULL	Razdalja med postajališčema.
dtValidFrom	DateTime, NULL	začetni čas veljavnosti vozovnice
dtValidTo	DateTime, NULL	končni čas veljavnosti vozovnice
iOfflineStatus	Integer, NULL	Status, ki ga določi zaledni sistem pri procesiranju IJPP transakcije. To je izhodni podatek. Kot vhodni podatek se ta vrednost pobriše.

**DuplicateData** struktura vsebuje naslednje parametre:

struktura DuplicateData		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
mCardProducts	CardProduct[]	Podatki o vozovnicah na originalni kartici. Struktura CardProduct v tem primeru ne vsebuje podatkov o medkrajevnih relacijah, ker se ti podatki prenašajo posebej.
mCardRelations	CardRelation[]	Podatki o medkrajevnih relacijah na originalni kartici.
mCardPassengerstatuses	CardPassengerStatus[]	Podatki o statusih potnika na originalni kartici
mSubsidyApplication	SubsidyApplication[]	Podatki o vlogah za subvencije
mWorkStatus	WorkStatus[]	Podatki o službenih statusih
mCard	Card	Podatki o originalni kartici
mPassenger	Passenger, NULL	Podatki o potniku

**CardProduct** struktura je opisana pri definiciji podatkovnih struktur pri transakcijskih metodah spletne storitve.

**CardPassengerStatus** struktura je opisana pri definiciji podatkovnih struktur pri transakcijskih metodah spletne storitve.

**CardRelation** struktura je opisana pri definiciji podatkovnih struktur pri transakcijskih metodah spletne storitve.

**SubsidyApplication** struktura vsebuje naslednje parametre:

struktura SubsidyApplication		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
sSubsidyCode	String	Identifikator vloge v sistemu
lTransactionExecutionId	Long, NULL	TransactionexecutionId se uporablja le v primeru posodobitve vloge z izvršno listo
iSubsidyStatus	Int, NULL	iSubsidyStatus se uporablja le v primeru posodobitve vloge z izvršno listo. iSubsidyStatus=1 (vpis/posodobitev vloge) iSubsidyStatus=2 (neveljavna vloga - izbris)
iServiceProviderId	Int, NUL	Identifikator ponudnika storitev
ICUID	Long	Identifikator potnikove kartice
dtStartTime	DateTime	Pričetek veljavnosti vloge
dtEndTime	DateTime	Konec veljavnosti vloge
iApplicationType	Integer	Vrsta vloge (MESEČNA=1 M10=2)
iApplicationStatus	Integer	Status (Dijak, Študent, Udeleženec izobraževanja odraslih)
iIndexRelation	Integer	Indeks za povezavo vloge z relacijo (identifikator vloge na kartici)
TariffLocationIdNo1	Integer	TariffLocationID za prvi mestni promet na vlogi
NumberOfLocationsNo1	Integer	Število področij za prvi mestni promet na vlogi
TariffLocationIdNo2	Integer	TariffLocationID za druge mestni promet na vlogi
NumberOfLocationsNo2	Integer	Število področij za drugi mestni promet na vlogi
TariffLocationIdNo3	Integer	TariffLocationID za tretji mestni promet na vlogi

NumberOfLocationsNo3	Integer	Število področij za tretji mestni promet na vlogi
----------------------	---------	---

**StatusData** struktura vsebuje naslednje parametre:

struktura TariffStatus		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iTariffStatusId	Integer	identifikator statusa
sTariffStatusName	String	ime statusa
iTariffStatusType	Integer	tip statusa
dtDefaultStartTime	DateTime, NULL	privzeti začetni čas statusa
dtDefaultEndTime	Datetime, NULL	privzeti končni čas statusa

**WorkStatus** struktura vsebuje naslednje parametre:

struktura WorkStatus		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
ICUID	Long	Identifikator potnikove kartice
iWorkStatusId	Integer	identifikator službenega statusa
sWorkStatusName	String	ime službenega statusa
iCompanyId	Integer	Identifikator prevoznika
dtStartTime	DateTime	začetni čas službenega statusa
dtEndTime	Datetime	končni čas službenega statusa

**BlacklistCatalog** struktura vsebuje naslednje parametre:

struktura BlacklistCatalog		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iType	Integer	Tip razloga kartice.
sName	String	Ime razloga za dodajanje na črno listo.

**BlacklistCatalog\_Pladenj** struktura vsebuje naslednje parametre:

struktura BlacklistCatalog_Pladenj		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iType	Integer	Tip razloga kartice.
sName	String	Ime razloga za dodajanje na črno listo.

**CardStatusCatalog** struktura vsebuje naslednje parametre:

struktura CardStatusCatalog		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iType	Integer	Tip statusa kartice.
sName	String	Ime statusa kartice.
sDescription	String	Opis statusa kartice.

**CardStatusCatalog\_Pladenj** struktura vsebuje naslednje parametre:

struktura CardStatusCatalog_Pladenj		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iType	Integer	Tip statusa kartice.
sName	String	Ime statusa kartice.
sDescription	String	Opis statusa kartice.

**WorkStatusCatalog** struktura vsebuje naslednje parametre:

struktura WorkStatusCatalog		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke

iType	Integer	Tip službenega statusa kartice.
sName	String	Ime službenega statusa kartice.
sDescription	String	Opis statusa kartice.

**WorkStatusCatalog\_Pladenj** struktura vsebuje naslednje parametre:

struktura WorkStatusCatalog_Pladenj		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iType	Integer	Tip službenega statusa kartice.
sName	String	Ime službenega statusa kartice.
sDescription	String	Opis statusa kartice.

**IJPPTransactionTypeCatalog** struktura vsebuje naslednje parametre:

struktura IJPPTransactionTypeCatalog		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iTransactionType	Integer	Tip IJPP transakcije.
sDescription	String	Opis IJPP transakcije.

**IJPPTransactionTypeCatalog\_Pladenj** struktura vsebuje naslednje parametre:

struktura IJPPTransactionTypeCatalog_Pladenj		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iTransactionType	Integer	Tip IJPP transakcije.
sDescription	String	Opis IJPP transakcije.

**OfflineStatusCatalog** struktura vsebuje naslednje parametre:

struktura OfflineStatusCatalog		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke

iOfflineStatus	Integer	OfflineStatus dodeljen IJPP transakciji.
sDescription	String	Opis OfflineStatus-a.

**SifrantObcin** struktura vsebuje naslednje parametre:

struktura SifrantObcin		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iIdObcina	Integer	Id občine.
sImeObcine	String	Ime občine.

**OfflineStatusCatalog\_Pladenj** struktura vsebuje naslednje parametre:

struktura OfflineStatusCatalog_Pladenj		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iOfflineStatus	Integer	OfflineStatus dodeljen IJPP transakciji.
sDescription	String	Opis OfflineStatus-a.

**SifrantObcin** struktura vsebuje naslednje parametre:

struktura SifrantObcin		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iIdObcina	Integer	Id občine.
sImeObcine	String	Ime občine.

**SifrantNaselij** struktura vsebuje naslednje parametre:

struktura SifrantNaselij		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iIdNaselje	Integer	Id naselja.
iIdObcina	Integer	Id občine.
sImeNaselja	String	Ime naselja.



iIdGURS	Integer	Id Gurs.
---------	---------	----------

**SifrantUpravljalcev** struktura vsebuje naslednje parametre:

struktura SifrantUpravljalcev		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iIdUpravljalca	Integer	Id upravljalca.
sSifraUpravljalca	String	Šifra upravljalca
sImeUpravljalca	String	Ime upravljalca.

**TerminalTypeCatalog** struktura vsebuje naslednje parametre:

struktura TerminalTypeCatalog		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iTerminalTypeId	Integer	Tip terminala.
sTerminalTypeName	String	Ime tipa terminala.

**TerminalTypeCatalog\_Pladenj** struktura vsebuje naslednje parametre:

struktura TerminalTypeCatalog_Pladenj		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iTerminalTypeId	Integer	Tip terminala.
sTerminalTypeName	String	Ime tipa terminala.

**TerminalSubTypeCatalog** struktura vsebuje naslednje parametre:

struktura TerminalSubTypeCatalog		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iTerminalSubTypeId	Integer	Tip podterminala.
iTerminalTypeId	Integer	Tip terminala.
sTerminalSubTypeName	String	Ime tipa podterminala.

**TerminalSubTypeCatalog\_Pladenj** struktura vsebuje naslednje parametre:

struktura TerminalSubTypeCatalog_Pladenj		
Ime spremenljivke	Tip spremenljivke	Opis spremenljivke
iTerminalSubTypeId	Integer	Tip podterminala.
iTerminalTypeId	Integer	Tip terminala.
sTerminalSubTypeName	String	Ime tipa podterminala.

## 9 POSTOPKI PRI UPORABI SPLETNIH STORITEV IJPP

Zaledni sistem IJPP nudi spletne storitve s pomočjo katerih lahko inicializiramo in urejamo različne entitete v transakcijskem sistemu, kot so terminali, prodajna mesta, kartice, potniki, ipd. V tem poglavju so opisani splošni postopki uporabe spletne storitve IJPP. Spletno storitev IJPP uporabljamo pri naslednjih postopkih:

- Registracija terminala
- Sprememba podatkov o terminalu
- Aktivacija SAM
- Registracija prodajnega mesta
- Sprememba podatkov o prodajnem mestu
- Registracija validatorja
- Registracija uporabnika
- Registracija nepersonificirane kartice
- Registracija personificirane kartice
- Nastavitev prenosa vozovnice na kartico z izvršno listo
- Nastavitev brisanja vozovnice na kartici z izvršno listo
- Nastavitev novega statusa uporabnika z izvršno listo
- Nastavitev začasno onemogočene kartice z izvršno listo
- Nastavitev začasno onemogočene vozovnice z izvršno listo

### 9.1 REGISTRACIJA TERMINALA

Vsi terminali, ki jih uporabljamo v transakcijskem sistemu IJPP morajo biti registrirani. Pri registraciji vsak terminal prejme unikatni identifikator, ki ga zaledni sistem IJPP uporablja kot referenco pri izvedbi transakcij. Prav tako terminal prejme lasten certifikat s katerim se lahko poveže na zaledni sistem. Registracija terminala poteka po sledečem postopku:

1. Nov terminal vpišemo v IJPP zaledni sistem z WS metodo SetTerminal, kjer pošljemo podatke o terminalu v strukturi mTerminalData. Pri tem Struktura mTerminalData ne vsebuje parametra iTId (identifikator terminala). WS metodo SetTerminal lahko kličemo le, z registriranim terminalom, ki ima nameščen SSL certifikat za dostop do WS zalednega sistema. V primeru, ko nimamo registriranega terminala, s katerim bi lahko izvedli vpis novega terminala, lahko vpis opravimo preko administracijskega spletnega vmesnika, oziroma vpis opravi upravitelj sistema namesto nas.

2. Po vpisu novega terminala prejmemo identifikator terminala iTId, ki se uporablja kot referenca pri transakcijah izvedenih z registriranim terminalom. Na nov terminal namestimo privzeti (registracijski) certifikat, ki ga prejmemo od upravljalca sistema.
3. Z registracijskim certifikatom registriramo terminal z WS metodo `RegistrationTerminal`. Metoda nam vrne SSL certifikat novega terminala za dostop do preostalih WS metod zalednega sistema.
4. Za delo s karticami mora terminal v zadnjem koraku aktivirati še SAM.

## 9.2 SPREMEMBA PODATKOV O TERMINALU

Registriranem terminalu lahko spremenimo parametre. Sprememba parametrov terminala poteka po sledečem postopku:

1. Najprej z WS metodo `GetTerminal` pridobimo trenutne podatke o terminalu. Pridobljene podatke ustrezno spremenimo.
2. Z WS metodo `SetTerminal` pošljemo na IJPP zaledni sistem nove podatke o terminalu v strukturi `mTerminalData`. Struktura `mTerminalData` vsebuje parameter iTId, s katerim identificiramo terminal, ki mu želimo spremeniti podatke.
3. WS metoda ob uspešni posodobitvi parametrov prodajnega mesta vrne iTId. Brez certifikata

## 9.3 AKTIVACIJA SAM

Terminali, ki so del IJPP transakcijskega sistema za delovanje potrebujejo SAM. Pred uporabo je potrebno SAM aktivirati po sledečem postopku:

1. Z WS metodo `ActivationSAM` pošljemo na IJPP zaledni sistem TID, serijsko številko terminala in serijsko številko SAM kartice. Struktura `mSAMData` vsebuje parameter iTId, sSerialNumber in sSAMNumber.
2. WS metoda ob uspešni aktivaciji vrne aktivacijski ključ za SAM (SAK)

## 9.4 REGISTRACIJA PRODAJNEGA MESTA

Prodajno mesto je lokacija, kjer se nahaja eden ali več terminalov za prodajo ali validacijo vozovnic. Podjetje z vlogo trgovca mora za vsak terminal registrirati prodajno mesto. Registracija prodajnega mesta poteka po sledečem postopku:

1. Z WS metodo `SetPos` pošljemo na IJPP zaledni sistem podatke o prodajnem mestu v strukturi `mPosData`. Struktura `mPosData` vsebuje vse parametre razen parametra iPosId.
2. Ob uspešni registraciji prodajnega mesta WS metoda `SetPos` vrne unikatni identifikator iPosId, ki ga je zaledni sistem določil registriranemu prodajnemu mestu.

## 9.5 SPREMEMBA PODATKOV O PRODAJNEM MESTU

Sprememba podatkov o prodajnem mestu poteka po sledečem postopku:

1. Najprej z WS metodo `GetPos` pridobimo trenutne podatke o prodajnem mestu. Pridobljene podatke ustrezno spremenimo.

2. Z WS metodo SetPos pošljemo na IJPP zaledni sistem nove podatke o prodajnem mestu v strukturi mPosData. Struktura mPosData vsebuje parameter iPosId, s katerim identificiramo prodajno mesto, ki mu želimo spremeniti podatke.
3. WS metoda ob uspešni posodobitvi parametrov prodajnega mesta vrne iPosId.

## 9.6 REGISTRACIJA VALIDATORJA

Validatorji so terminali s katerimi lahko validiramo in kupujemo vozovnice na vozilih. Vsak validator v sistemu IJPP mora vsebovati podatke o vozilu na katerem je nameščen. Pri tem se vsako vozilo obravnava kot ločeno prodajno mesto. Registracija validatorja poteka po sledečem postopku:

1. S postopkom registracije terminala (validatorja) pridobimo unikatni identifikator terminala iTid.
2. S postopkom registracije prodajnega mesta pridobimo unikatni identifikator prodajnega mesta iPosId, ki predstavlja vozilo na katerem je nameščen validator.
3. S postopkom spremembe podatkov o terminalu (validatorju), validatorju nastavimo pridobljeni identifikator prodajnega mesta iPosId, ki predstavlja vozilo na katerem je nameščen validator.

## 9.7 REGISTRACIJA UPORABNIKA

Registracija uporabnika poteka po sledečem postopku:

1. Z WS metodo SetPassenger pošljemo na IJPP zaledni sistem podatke o uporabniku v strukturi mPassengerData.
2. Ob uspešni registraciji WS metoda SetPassenger vrne parameter iPIId, ki predstavlja unikatni identifikator potnika.

## 9.8 REGISTRACIJA NEPERSONIFICIRANE KARTICE

Nepersonificirane kartice uporabljajo anonimni uporabniki. Registracija nepersonificirane kartice poteka po sledečem postopku:

1. Z WS metodo SetCard pošljemo na IJPP zaledni sistem podatke o kartici v strukturi mCardData. Znotraj strukture mCardData nastavimo tip nepersonificirane kartice: CardType=1. Identifikatorja uporabnika ne vnašamo: iPIId=NULL.
2. WS metoda SetCard vrne rezultat izvedbe.

## 9.9 REGISTRACIJA PERSONIFICIRANE KARTICE

Personificirane kartice uporabljajo registrirani uporabniki. Registracija personificirane kartice poteka po sledečem postopku:

1. Izvedemo registracijo uporabnika, kjer od zalednega sistema prejmemo unikatni identifikator potnika iPIId.
2. Z WS metodo SetCard pošljemo na IJPP zaledni sistem podatke o kartici v strukturi mCardData. Znotraj strukture mCardData nastavimo tip nepersonificirane kartice: CardType=2 in identifikator potnika (iPIId), ki bo uporabljal registrirano personificirano kartico.
3. WS metoda SetCard vrne rezultat izvedbe.

## 9.10 NASTAVITEV PRENOSA VOZOVNICE NA KARTICO Z IZVRŠNO LISTO

Izvršna lista omogoča oddaljen prenos vozovnice iz zalednega sistema na kartico. Zahtevo za prenos vozovnice (ki se bo prenesla na kartico) z izvršno listo vpišemo v zaledni sistem po sledečem postopku:

1. Z WS metodo SetExecutableList pošljemo na IJPPzaledni sistem podatke o kartici in vozovnici. Podatke o kartici in vozovnici zapakiramo v podatkovno strukturo mExecutableData. V tej strukturi nastavimo identifikator kartice ICUID na katero želimo prenesti vozovnico. Parameteru iAction nastavimo vrednost 1. Abstraktni razred mAdditionalCardObject predstavlja struktura CardProduct, ki vsebuje parametre vozovnice, ki se bo prenesla na kartico.
2. WS metoda vrne rezultat izvedbe. Po uspešni izvedbi se zahteva za online prenos vozovnice na kartico vpiše v naslednjo aktivno izvršno listo.

## 9.11 NASTAVITEV BRISANJA VOZOVNICE NA KARTICI Z IZVRŠNO LISTO

Izvršna lista omogoča oddaljeno brisanje vozovnice na kartici. Zahtevo za online brisanje vozovnice na kartici z izvršno listo vpišemo v zaledni sistem po sledečem postopku:

1. Z WS metodo SetExecutableList pošljemo na IJPPzaledni sistem podatke o kartici in vozovnici, ki jo želimo odstraniti iz kartice. Podatke o kartici in vozovnici zapakiramo v podatkovno strukturo mExecutableData. V tej strukturi nastavimo identifikator kartice ICUID, ki vsebuje vozovnico za brisanje. Parameteru iAction nastavimo vrednost 2. Abstraktni razred mAdditionalCardObject predstavlja struktura CardProductShort, ki vsebuje parametre vozovnice, ki se bo brisala iz kartice.
2. WS metoda vrne rezultat izvedbe. Po uspešni izvedbi se zahteva za online brisanje vozovnice na kartici vpiše v naslednjo aktivno izvršno listo.

## 9.12 NASTAVITEV NOVEGA STATUSA UPORABNIKA Z IZVRŠNO LISTO

Izvršna lista omogoča oddaljeno nastavitev statusa uporabnika na kartici. Zahtevo za nastavitev novega statusa na kartici z izvršno listo vpišemo v zaledni sistem po sledečem postopku:

1. WS metodo SetExecutableList pošljemo na IJPPzaledni sistem podatke o kartici in statusu, ki ga želimo prenesti na kartico. Podatke o kartici in statusu zapakiramo v podatkovno strukturo mExecutableData. V tej strukturi nastavimo identifikator kartice ICUID na katero želimo prenesti nov status. Parameteru iAction nastavimo vrednost 3. Abstraktni razred mAdditionalCardObject predstavlja struktura CardPassengerStatus, ki vsebuje parametre

novega statusa, ki se bo prenesel na kartico. Po prenosu novega statusa, se stari status na kartici ne spremeni le, če se veljavnost novega statusa časovno ne prekriva s starim. V primeru, da se novi status časovno prekriva s starim, se veljavnost starega statusa skrajša tako, da velja do začetka novega statusa.

2. WS metoda vrne rezultat izvedbe. Po uspešni izvedbi se zahteva za online nastavitev statusa vpiše v naslednjo aktivno izvršno listo.

### 9.13 NASTAVITEV ZAČASNO ONEMOGOČENE KARTICE Z IZVRŠNO LISTO

Izvršna lista omogoča začasno oddaljeno onemogočanje uporabe kartice na terminalih. Zahtevo za onemogočanje kartice z izvršno listo vpišemo v zaledni sistem po sledečem postopku:

1. Z WS metodo SetExecutableList pošljemo na IJPP zaledni sistem identifikator kartice, ki jo želimo onemogočiti za uporabo. Podatke o kartici zapakiramo v podatkovno strukturo mExecutableData. V tej strukturi nastavimo identifikator kartice ICUID. Parameteru iAction nastavimo vrednost 4. Prav tako nastavimo začetni in končni čas obdobja v katerem želimo onemogočiti kartico.
2. WS metoda vrne rezultat izvedbe. Po uspešni izvedbi se zahteva za online onemogočanje kartice vpiše v naslednjo aktivno izvršno listo.

### 9.14 NASTAVITEV ZAČASNO ONEMOGOČENE VOZOVNICE Z IZVRŠNO LISTO

Izvršna lista omogoča začasno oddaljeno onemogočanje uporabe vozovnice. Zahtevo za onemogočanje vozovnice izvedemo po sledečem postopku:

1. Z WS metodo SetExecutableList pošljemo na IJPP zaledni sistem podatke o kartici in vozovnici, ki jo želimo onemogočiti za uporabo. Podatke o kartici in vozovnici zapakiramo v podatkovno strukturo mExecutableData. V tej strukturi nastavimo identifikator kartice CUID, ki vsebuje vozovnico za brisanje. Parameteru iAction nastavimo vrednost 5. Abstraktni razred mAdditionalCardObject predstavlja struktura CardProductShort, ki vsebuje parametre vozovnice, ki jo želimo onemogočiti. Prav tako nastavimo začetni in končni čas obdobja v katerem želimo onemogočiti vozovnico.
2. WS metoda vrne rezultat izvedbe. Po uspešni izvedbi se zahteva za online onemogočanje vozovnice vpiše v naslednjo aktivno izvršno listo.

### 9.15 ODSTRANITEV NEIZVRŠENEGA OPRAVILA IZ IZVRŠNE LISTE.

Če želimo preklicati določeno opravilo ki je že na izvršni listi, ali je v postopku dodajanja na izvršno listo, lahko to izvedemo po sledečem postopku:

3. Z WS metodo DelExecutableList pošljemo na IJPP zaledni sistem podatke o kartici, opravi in referenci vozovnice ali statusa na izvršni listi.

4. WS metoda vrne rezultat izvedbe. Po uspešni izvedbi metode, preklicano opravilo ni več prisotno v naslednji aktivni izvršni listi.



## 10 POSTOPKI ZA VERIFIKACIJO STROJNE IN PROGRAMSKE OPREME TER BREZSTIČNE KARTICE

Dandanes se programska in strojna oprema pojavlja povsod. Oba segmenta, programska in strojna opreme delujeta vzajemno ena z drugo. Za brezhibno delovanje naprave v celoti (mobilni telefon, POS terminal, parkomat, računalnik, TV,... ) morata brezhibno delovati tudi strojna in programska oprema. Zaradi pomembnosti obeh segmentov tako iz socialnega (nezaupanje v implementirano rešitev – slaba reklama) kot ekonomskega vidika (naraščajoči stroški pri odpravah napak operativnega sistema) morata biti oba segmenta preizkušena na vseh nivojih. Oprema (strojna, programska), ki lahko povzroči izgubo podatkov zaradi specifične pomanjkljivosti v takšnem sistemu, kot je IJPP ni več sprejemljiva. Iz tega razloga v tem segmentu podajamo priporočila in testne postopke, ki jih mora implementirana programska in strojna oprema prestati kot pogoj kasnejše uspešne implementacije v celovit sistem IJPP. Pri zagotavljanju kakovosti obeh omenjenih segmentov (programski, strojni) ima ključno vlogo postopek testiranja. Ker so programske, kot tudi rešitve strojne opreme vedno bolj kompleksne, postaja s tem tudi postopek testiranja kompleksnejši. Trend v zadnjem času stremi k popolni avtomatizaciji izvedbe testnih scenarijev oziroma vsaj k delni avtomatizaciji le teh. Cilj je v čim krajšem času in na učinkovit način zagotoviti zanesljivo in učinkovito delovanje opreme (strojne, programske). Z vnaprej pripravljenimi testi programske in strojne opreme lahko njihovo delovanje preverimo hitro in učinkovito.

### 10.1 SPECIFIKACIJA PROCEDURE ZA PRISTOP K VERIFIKACIJSKIM POSTOPKOM

V sklopu tega poglavja je definirana procedura za pristop k verifikacijskim postopkom, in koraki postopanja od pridobitve dokumentacije standarda IJPP do priglasitve testnih postopkov in opreme v za to kvalificiranih testnih laboratorijih.

Zaporedje posameznik korakov procedure za pristop k verifikacijskim postopkom:

- Pridobitev dokumentacije standarda IJPP,
- prilagoditev strojne in programske opreme po standardu IJPP,
- spletni obrazec za priglasitev opravljanja verifikacijskih postopkov v IJPP laboratoriju,
- odgovor laboratorija IJPP,
- potrditev termina izvajanja testov v laboratoriju IJPP s strani odgovorne (registrirane) osebe tehnološkega ponudnika,
- postavitve opreme tehnološkega ponudnika v okolje laboratorija,
- izvedba specificiranih testov za priglašeno opremo tehnološkega ponudnika.

#### ***10.1.1 PRIDOBITEV DOKUMENTACIJE STANDARDA IJPP***

Dokumentacija standarda IJPP je na razpolago na spletnem portalu IJPP ([www.ijpp-standard.si](http://www.ijpp-standard.si)). Standard je na razpolago registriranim tehnološkim ponudnikom, ki želijo sodelovati znotraj projekta IJPP. Za to je potrebna predhodna registracija (ime, priimek, rojstni datum, tehnološki ponudnik – polno ime podjetja, sedež podjetja, kontaktna telefonska številka in naslov elektronske pošte) odgovorne osebe tehnološkega ponudnika na portalu IJPP ([www.ijpp-standard.si](http://www.ijpp-standard.si)), ki je zadolžena za koordinacijo priprave testne programske in strojne opreme s testnim laboratorijem IJPP, oziroma enim izmed potrjenih laboratorijev iz strani IJPP. Preden se registriranemu tehnološkemu ponudniku omogoči prenos standarda IJPP se mora izrecno strinjati z EULA (ang. End User License Agreement) razpisanimi pogoji, ki so uporabniku podani pred prenosom dokumentacije. Če registrirana oseba tehnološkega ponudnika ne sprejme pogojev, se mu hkrati onemogoči prenos dokumenta. Poslovno-občutljive vsebine dokumenta, ki se ne tičejo tehnološkega ponudnika se varujejo pod NDA.

#### ***10.1.2 PRILAGODITEV STROJNE IN PROGRAMSKE OPREME STANDARDU IJPP***

Ponudnik strojne in pripadajoče programske opreme mora v skladu s standardom IJPP le to prilagoditi, da izpolnjuje vse pogoje IJPP aplikacije, kot jo definira standard IJPP. V pomoč prilagajanju strojne in programske opreme so lahko tehnološkemu ponudniku tudi definirani testni postopki s katerimi lahko pri sebi preveri pravilnost delovanja posameznih sklopov, komponent, naprav. Če s testnimi postopki potrdi skladno delovanje mu je na razpolago naslednji korak v katerem zaprosi za opravljanje verifikacijskih postopkov (korak 3) v IJPP laboratoriju oz. enem izmed laboratorijev, ki jih odobri IJPP.

#### ***10.1.3 SPLETNI OBRAZEC ZA PRIGLASITEV OPRAVLJANJA VERIFIKACIJSKIH POSTOPKOV V IJPP LABORATORIJU***

Registrirani uporabnik (oseba) tehnološkega ponudnika (iz koraka 1) ima v spletnem portalu IJPP ([www.ijpp-priglasitev.si](http://www.ijpp-priglasitev.si)) na razpolago spletni obrazec za priglasitev opravljanja verifikacijskih postopkov. Obrazec je na razpolago samo registriranim osebam tehnoloških ponudnikov. V obrazcu je potrebno navesti element ali skupino elementov, ki bodo predmet testiranja, specificirati, kaj točno se bo testiralo, okviren čas testiranja (zaradi rezervacije terminov v laboratoriju), obseg testov in specifikacije v kateri navezi se bo testiranec uporabljal z drugimi elementi sistema IJPP (npr. komponente drugih tehnoloških ponudnikov). V obrazcu se v obliki priponke laboratoriju posreduje popolna dokumentacija naprave ali sklopa naprav, ki so predmet testiranja in vsi certifikati, če jih oprema premore. Ker obrazec izpolnjuje že prijavljena in registrirana oseba tehnološkega ponudnika, podatkov te osebe ni potrebno ponovno vnašati.

#### ***10.1.4 ODGOVOR LABORATORIJA IJPP***

Laboratorij IJPP zadolžen za verifikacijo programske in strojne opreme bo v najkrajšem možnem času (v roku 30 dni od prejete priglasitve) prijavitelju sporočil na elektronski naslov možne termine za izvedbo verifikacijskih postopkov, oz. v primeru nepopolne vloge sporočil pomanjkljivosti, ki jih mora prijavitelj v priglasitvi dopolniti. Za primer dopolnitve manjkajočih podatkov je poskrbljeno znotraj prijavnega obrazca.

#### ***10.1.5 POTRDITEV TERMINA IZVAJANJA TESTOV V LABORATORIJU IJPP S STRANI ODGOVORNE (REGISTRIRANE) OSEBE TEHNOLOŠKEGA PONUDNIKA***

Tehnološki ponudnik je v primeru oddaje ustrezne dokumentacije pri priglasitvi verifikacijskih postopkov obveščen preko elektronske pošte, da ga na prijavnem obrazcu čakajo možni termini, v katerih lahko opravi verifikacijske postopke. Odgovorna oseba tehnološkega ponudnika mora s klikom na odgovarjajoč termin potrditi tistega, ki mu ustreza (možna samo ena izbira). S potrditvijo termina v spletnem obrazcu, se ta podatek posreduje do IJPP laboratorija, ki poskrbi za rezervacijo prostora v tem terminu.

#### ***10.1.6 POSTAVITEV OPREME TEHNOLOŠKEGA PONUDNIKA V OKOLJE LABORATORIJA***

V rezerviranem terminu mora tehnološki ponudnik v laboratorij dostavit kompletno opremo, ki je bila priglašena v spletnem obrazcu, in poskrbeti za morebitne medsebojne povezave, ki se uporabljajo pri medsebojni komunikaciji naprav tehnološkega ponudnika.

#### ***10.1.7 IZVEDBA SPECIFICIRANIH TESTOV ZA PRIGLAŠENO OPREMO TEHNOLOŠKEGA PONUDNIKA***

Teste, ki se tičejo priglašene opreme (testiranci) izvaja kvalificirano osebje laboratorija IJPP v navezi z odgovornimi osebami tehnološkega ponudnika, ki prisostvujejo testnim proceduram. Pred izvedbo testnih postopkov se izvede zapisnik prisotnih oseb iz strani tehnološkega ponudnika in kvalificiranih oseb, ki izvajajo teste v IJPP laboratoriju. Skladno z definicijo testov se izpolnjuje testni obrazec za posamezen sklop, v katerega se vnašajo informacije o opravitih/neopravitvi testa, ter opombe in morebitne opažene pomanjkljivosti. Za testne obrazce poskrbi testni laboratorij IJPP oz. laboratorij, ki ga pooblasti IJPP. Po opravljenih testih se testne obrazce ožigosa z uradnim žigom testnega laboratorija in podpisom uradne osebe laboratorija in tehnološkega ponudnika. V kolikor je naprava opravila vse deklarirane teste uspešno, se ji izda potrdilo z odobritvijo integracije v IJPP sistem. V primeru neopravljenih testnih procedur testirana oprema ne prejme potrdila za integracijo v IJPP sistem. Testna dokumentacija se vodi v treh izvodih. En izvod pripada tehnološkemu ponudniku, en laboratoriju, preostali izvod testne dokumentacije testiranja, izdan iz strani laboratorija pa se obvezno hrani pri upravitelju IJPP za obdobje najmanj deset let.

Pomoč uporabnikom v zvezi preglašanja verifikacije programske in strojne opreme je na razpolago preko elektronske pošte [verifikacije@ijpp.si](mailto:verifikacije@ijpp.si) ali preko telefonske številke 01-xxx xx xx.

Opomba: dokumentacija standarda IJPP je tehnološkim ponudnikom na razpolago brezplačno! Izvajanje verifikacijskih postopkov v IJPP laboratoriju in preostalih koncesijskih laboratorijih se plača po definiranem ceniku opravljanja laboratorijskih storitev po urni postavki.

#### **10.1.8 POGOJI, KDAJ MORA PERIFERIJA TEHNOLOŠKEGA PONUDNIKA OPRAVITI TEST**

Testi v IJPP laboratoriju so obvezni za vsakega tehnološkega ponudnika, katerega strojna oz. programska oprema še nima pridobljenega soglasja za integracijo v IJPP sistem. Za opremo, ki je že pridobila soglasje integracije v IJPP, in je na njej prišlo do spremembe, se:

- Testov ne izvaja, če gre za minimalno spremembno, in ne vpliva na funkcionalnost
- Izvede predpisan nabor obstoječih testov, ki potrdijo pravilnost delovanja implementirane spremembe (v kolikor obstoječi testi ne zajemajo področje sprememb, nadgrajenj, jih IJPP laboratorij definira v skladu s spremembami)
- Izvede celoten nabor obstoječih in novo definiranih testov, kadar gre za večje posege v periferijo in delovanje same aplikacije IJPP.

Npr. testov na opremi, ki že ima soglasje se ne izvaja (prva točka) v primeru, kadar gre za minimalistične spremembe v aplikaciji IJPP (npr. sprememba določenega izpisa na GUI, majhne vizuelne spremembe na GUI-ju brez poseganja v funkcionalnost ipd.). Določen nabor obstoječih testov, se izvede v primerih, kadar gre za majhne spremembe že funkcionalnega in predhodno odobrenega segmenta (npr. sprememba vrstnega reda v meniju). Nabor novo specificiranih testov se izvede za segment, ki je bil naknadno dodan IJPP aplikaciji in nima vpliva na obstoječo funkcionalnost. V primerih kadar nova dodana funkcionalnost vpliva na obstoječo (npr. spreminjanje varnostnih mehanizmov, spreminjanje datotečnega zapisa IJPP kartice, ipd.) znotraj IJPP aplikacije, pa se lahko izvede tudi kompleten nabor obstoječih in novo definiranih testov, o čimer presoja laboratorij IJPP.

Vsak ponudnik tehnološke in programske opreme mora poskrbeti še za računanje CRC-ja, ki zajema tako periferijo kot programsko opremo, s čimer se omogoča nadzor nad celoto, ki se nahaja na terenu, skladno z evidencami opravljenih/neopravljenih testov posamezne različice. Oprema tehnološkega ponudnika mora v zaledni sistem posredovati predhodno izračunan CRC in verzijo programske opreme s katero razpolaga na testu in kasneje na terenu.

V primeru ugotovitve neskladnja že odobrene različice programske opreme (različica na terenu) si IJPP laboratorij pridružuje pravico do ponovne izvedbe testov. V tem primeru IJPP laboratorij nemudoma obvesti tehnološkega ponudnika.

#### **10.2 DEFINICIJA IN OSNOVNI POJMI UPORABLJENI V POSTOPKU VERIFICIRANJA**

Definicij, ki zastopajo pojem testiranja je veliko, ki pa so si v osnovi, ki se nanaša na proceduro testiranja enotne. Testiranje je torej proces, s katerim skušamo dokazati da v programski opremi, ki jo gradimo, oziroma smo jo zgradili ni napak, pomanjkljivosti ipd. Hkrati s testnim procesom dokazujemo, da programska rešitev deluje v pričakovanih okvirjih in v skladu s specifikacijami in deklaracijami, ki spremljajo nek končen produkt. Namen testiranja je odkrivanje morebitnih napak. Šele po uspešno opravljenem postopku testiranja lahko dokažemo in dosežemo zaupanje, da testirane rešitve delujejo v skladu s podanimi deklaracijami oz. zahtevami, ki jih specificira naročnik. Iz tega vidika ni postopek testiranja nič kaj drugega kot postopek vrednotenja posamezne komponente, ki razkrije odstopanje med dejanskimi izhodnimi podatki in pričakovanimi, specificiranimi podatki. Pomembno je tudi dejstvo, da testiranje lahko prikaže samo prisotnost napak, ne more pa dokazati odsotnosti le teh. Ker bo v nadaljevanju segmenta govora o postopkih testiranja, specificiranih testnih scenarijev, bomo najprej podali par osnovnih pojmov, za kasnejše lažje razumevanje terminologije pravilne rabe terminov.

**Verifikacija** – je proces pri katerem se preverja ali produkt tekoče faze ustreza zahtevam, ki so bile postavljene v predhodni fazi.

**Validacija** - predstavlja proces vrednotenja programske opreme, modula ob koncu razvojnega cikla z namenom ugotavljanja skladnosti z zahtevami.

**Modul** - je najmanjša enota, ki jo v postopku testiranja lahko urejamo, preverjamo, spreminjamo in testiramo.

**Testni primer** – gre za podrobno in natančno specificiran testni postopek v katerem so jasno opisani vsi koraki, potek, omejitve, navodila ipd. za uspešno izvedbo specifičnega testnega scenarija.

**Produkt** – je končni izdelek razvojnega procesa, ki mora prestat testno proceduro za ugotavljanje skladnosti s podanimi naročnikovimi zahtevami.

**Tester** - je oseba, ki po specificiranih testnih scenarijih preizkuša skladnost razvite opreme. Ta oseba je usmerjena na kakovost produkta in je do njega maksimalno kritična.

### 10.3 PROCES TESTIRANJA IN NJEGOVI SKLOPI

Proces testiranja razdelimo v tri sklope, ki zavesijo od tega kaj testiramo. V IJPP primeru bomo v tem dokumentu specificirali vse tri načine. Prvi se nanaša na testiranje posameznih modulov, kjer se preverja ustreznost delovanja modula. Drugi način zajema integracijsko testiranje pri katerem se testirajo moduli združeni v celoto oziroma posamezne korelacije med moduli. Tretji način zajema sistemsko testiranje, kjer se preverja ustreznost razvitega produkta kot celote skladno z naročnikovo zahtevo.

Glede na način testiranja se bodo testi izvajali na dveh osnovnih načelih. Uporaba načela t.i. bele skrinjice in načela t.i. črne skrinjice. V primeru testnega postopka uporabe bele skrinjice bo tester potreboval dostop do notranje strukture programa oz. algoritmov. Pri takšni vrsti testa se izberejo vhodni podatki in določijo ustrezni izhodni podatki. Ker se v sistemu IJPP uporabljajo sistemi različnih ponudnikov, spadajo takšni testi v domeno ponudnika opreme. S takšnim testom razvijalec rešitve zagotovi in potrdi pravilno delovanje, oziroma ugotovi posamezne pomanjkljivosti.

Drugi način izvajanja testov, ki se bo v večini uporabljal na ravni IJPP kot skupen ocenjevalni test pravilnosti delovanja rešitev različnih proizvajalcev integriranih v celoto, je test na osnovi črne skrinjice. Test na osnovi črne skrinjice oz. funkcionalno testiranje je najpogostejša oblika testiranja, in mora biti izvedeno v vsakem primeru. V tem primeru se rešitev obravnava takšna kot je, pri čemer nas notranja arhitektura strojne oz. programske opreme ne zanima. Pri tem testu preverjamo samo delovanje rešitve, pri čemer spremljamo odziv rešitve na poznan vhod oz. poznan nabor vhodnih podatkov. Če se rezultat testa (odziv izhod) ujema s pričakovanim, lahko test potrdimo kot uspešen.

Ker so testni scenariji v nadaljevanju dokumenta zasnovani za različna področja in različne sklope opreme, bomo za vsak specificiran testni scenarij podali pogoj, kdaj se mora takšen scenarij izvesti. To pomeni naslednje. Če kot primer vzamemo test ustrezne jakosti polja NFC čitalnika. Slednjega se izvede na izbranem številu testiranih naprav. Če vzorec nakazuje na pozitivno sprejetje testa, potem slednjega ni potrebno izvajati vse do npr. menjave NFC modula, njegove fizične komponente ipd. S specificiranimi pogoji testa (kdaj in kako) bo nabor testnih scenarijev pri specifičnih spremembah posameznih ponudnikov enostaven.

Vrste testov morajo biti izvedene na način, kot navaja Tabela 10.1:

TABELA 10.1: POSTOPEK VERIFIKACIJE

	Razvojno okolje	Testno okolje	Produksijsko okolje
<b>Testiranje modulov</b>	✓		
<b>Integracijski testi</b>	✓	✓	
<b>Varnostni testi</b>		✓	✓
<b>Sistemske testi</b>		✓	

**Razvojno okolje:** integracijske teste v sklopu razvojnega okolja izvede ponudnik opreme sam.

**Testno okolje:** integracijske teste, varnostne teste in sistemske teste v testnem okolju izvede za to zadolžena institucija v skladu s tem dokumentom.

**Produksijsko okolje:** varnostne teste v produkcijskem okolju izvede upravljavec in za to zadolžena institucija skladno s podanim testnim načrtom v tem dokumentu.

Testno okolje je pod nadzorom testne ekipe in mora biti ločeno od razvojnega okolja (razvojne ekipe). Bolj kot testno okolje in njegove lastnosti približamo produkcijskemu okolju (iste naprave, isti strežniki, isti operacijski sistemi, povezave sličnih prenosnih hitrosti, ipd.), tem boljši bodo testni rezultati. Produksijsko okolje predstavlja pogoje v katerem bo kasneje sistem zaživel in obratoval, ko bo predan naročniku. Kot je definirano v zgornji tabeli se bodo v testnem okolju izvajali večinoma varnostni in sistemske testi, ter pretežno tudi integracijski testi, kadar bo to potrebno. V produkcijskem okolju bodo po uspešnih testih v testnem okolju izvedeni še varnostni testi. V sklopu testov v razvojnem okolju morajo ponudniki svojih rešitev poskrbeti tako za testne procedure modulov kot tudi za same integracijske teste.

## 10.4 SPLOŠNI TESTNI NAČRT

Osnove oziroma osnutek osnovnega testnega načrta priporoča standard ANSI/IEEE 829-1983 (ang. Software Test Documentation), ki zajema naslednje točke:

- Identifikator testnega načrta
- Opis produkta
- Karakteristike, ki bodo testirane
- Karakteristike, ki ne bodo testirane

- Uporabljene metode testiranja in terminalna kriterijska funkcija
- Prekinitev in nadaljevanje testiranja
- Identifikacija izročljivih dokumentov
- Potrebne aktivnosti za izvedbo testiranja
- Opis testnega okolja
- Identifikacija odgovornosti
- Človeški viri in potrebno šolanje
- Časovni roki
- Tveganja in posledice
- Potrditev testnega načrta
- Dodatki

Načrtovanje testnega načrta zajema obravnavo vseh aktivnosti, ki so potrebne za izvajanje testiranja in verifikacijo. Testni načrt obravnava tudi organiziranost in učinkovito uporabo procesov, osebja, orodja in opreme. Testni načrt zajema:

- Točno definirane vloge in odgovornost udeležencev v projektu
- Specifikacije testnega procesa
- Specifikacije testnega okolja in zahteve
- Specifikacijo uporabljenih metod testiranja, in
- Časovni načrt za razvoj in izvajanje testnih primerov.

## 10.5 ZASNOVA TESTNIH SCENARIJEV PO SKLOPIH TESTIRANIH SEGMENTOV


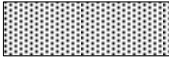
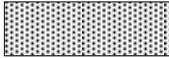

Glede na uporabo različnih komponent in naprav v sistemu IJPP so testni scenariji razdeljeni v posamezne sklope glede na uporabljeno terminalsko opremo. Testni scenariji pokrivajo ključne komponente vključene v IJPP sistem, kar pomeni, da so v tem dokumentu specificirani vsi ključni testni scenariji od preizkušanja pravilnosti delovanja NFC modula, brezstičnih kartic, validatorjev, povezav, komunikacijskih protokolov, formata zapisa itd. Kot omenjeno so testi, ki se tičejo delovanja NFC sistema in brezstičnih kartic zajeti v enem podpoglavju, testi npr. validacijske enote v drugem podpoglavju, testi komunikacijskih protokolov v tretjem podpoglavju itn. To omogoča enostaven pregled skupne testov, ki se tičejo specifičnega področja.

### *10.5.1 TEST POENOTENEGA DELOVANJA VALIDIRNIH ENOT RAZLIČNIH PONUDNIKOV*

Ker so v sistem IJPP vključene različne validirne enote različnih ponudnikov, ki se nadgrajujejo za potrebe IJPP je iz stališča uporabnika potrebno zagotoviti enotno obnašanje. Test zajema naprave brez zaslona s prisotno eno svetlečo diodo, naprave brez zaslona in štirimi svetlečimi diodami in naprave brez zaslona in brez svetlečih diod opremljene z zvočno napravo. Ta specifikacija je povzeta po dokumentu »EMV Contactless Specifications for Payment Systems«, ki se uporablja v sistemih brezstičnega plačevanja.

### 10.5.1.1 Test indikacije nepripravljenosti validirne enote






Identifikacijska oznaka testa: IND001

Status	Opis	Samo en LED indikator	Samo zaslon	Samo avdio	Zaslon ali 4 LED indikatorji
Nepripravljenost validatorja	Kadar čitalnik validirne enote ni vključen, ali če ne komunicira z nobeno terminalsko opremo ni prikazano nobeno tekstovno ali signalno sporočilo			Brez zvočnih signalov	 

Test:	IND001	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

### 10.5.1.2 Test indikacije pri vklopu validirne enote

Identifikacijska oznaka testa: IND002



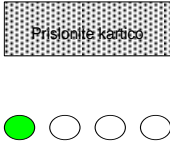
Status	Opis	Samo en LED indikator	Samo zaslon	Samo avdio	Zaslon ali 4 LED indikatorji
Vklop validatorja	Validator je vključen in ustrezno povezan, vendar še ni v pripravljenosti čitanja kartice. V primeru enega oz. štirih svetlečih indikatorjev prvi utripa z dolžino 200ms na periodo 5s.			Brez zvočnih signalov	  

Test:	IND002	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		



### 10.5.1.3 Test indikacije pripravljenosti validirne enote na sprejem kartice



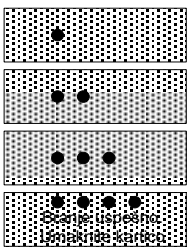
Identifikacijska oznaka testa: IND003

Status	Opis	Samo en LED indikator	Samo zaslon	Samo avdio	Zaslon ali 4 LED indikatorji
Pripravljenost na branje brezstične kartice	Validator ima izvršene vse procedure, ki so potrebne pred branjem brezstične kartice. S konstantno vključeno svetlečo diodo (vmesnik s štirimi svetlečimi diodami), oziroma utripajočo diodo (vmesnik z eno svetlečo diodo) periode 1000ms in časovno dolžino vklopa 200ms signalizira pripravljenost na branje. V primeru validirne enote z zaslonom, izpiše sporočilo za branje brezstične kartice.			Brez zvočnih signalov	

Test:	IND003	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

### 10.5.1.4 Test indikacije procedure branja kartice na validirni enoti

Identifikacijska oznaka testa: IND004

Status	Opis	Samo en LED indikator	Samo zaslon	Samo avdio	Zaslon ali 4 LED indikatorji
Pripravljenost na branje brezstične kartice	V primeru uporabniškega vmesnika z eno svetlečo diodo, slednja po uspešnem branju kartice ostane vklopljena za periodo 2 sekundi (preide iz utripanja v konstanten vklop za dolžino dveh sekund). Po dveh sekundah se vrne v stanje pripravljenosti za branje. V primeru vmesnika štirih svetlečih diod se diode od prve do zadnje v proceduri branja vključijo v intervalu 250ms in ostanejo vključene za obdobje			Po uspešnem branju signal za uspeh (glej definicijo signala za uspeh)	

	750ms. V primeru zaslona, se na zaslon prikaže indikator napredka. Po končanem uspešnem branju pa še sporočilo o uspešnem branju. V primeru samo zvočnega vmesnika, se po uspešnem branju predvaja ton za uspeh. Tej indikaciji sledi nadaljevanje transakcije, ki pa je v teh testih ne obravnavamo.				
--	---	--	--	--	--

Test:	IND004	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

### 10.5.1.5 Test indikacije procedure neuspešnega branja kartice na validirni enoti

Identifikacijska oznaka testa: IND005

Status	Opis	Samo en LED indikator	Samo zaslon	Samo avdio	Zaslon ali 4 LED indikatorji
Napaka pri branju brezstične kartice	<p>Kadar pride do nepopravljive napake med bralnikom in brezstično kartico se slednji glede na uporabljen vmesnik inducira na sledeč način. Vizualni indikatorji (štiri svetleče diode) so v takšnem primeru izključene in ne oddajajo nikakršnih svetlobnih signalov. V primeru vmesnika z eno svetlečo diodo slednja signalizira na napako s periodo utripa na vsakih 500ms.</p> <p>V primeru napake pri branju kartice se mora validirna enota po preteku dveh sekund prestaviti v stanje pripravljenosti branja kartice: »stanje – prislomite kartico«. Identična signalizacija je uporabljena za testi primer, kadar sta bralniku istočasno približani dve brezstični kartici!</p>			<p>Predvajanje zvoka za napako (glej definicijo zvoka v nadaljevanju)</p>	

Test:	IND005	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

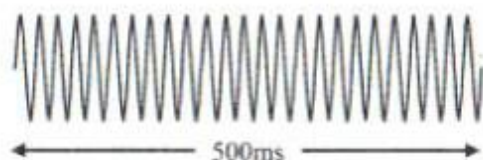
#### 10.5.1.6 Test avdio indikacije na validirni enoti

Validirne enote, ki podpirajo zvočno indikacijo, morajo vsebovati dva različna tona, izmed katerih eden signalizira uspešno izvedeno proceduro in drugi neuspešno. Validirna naprava mora biti sposobna predvajati oba zvoka, ki se jasno razlikujeta med seboj, in sta slišna v operativnem okolju. Pri implementaciji jakosti zvočnih signalov se upošteva šum okolice.

*Identifikacijska oznaka testa: IND006*

##### *Zvočni signal za uspešno izvedeno proceduro*

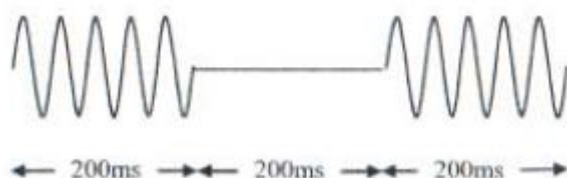
Zvočni signal za uspešno izvedeno proceduro na validirni napravi mora biti sinusni val, frekvenca 1500Hz in periode 500ms.



Zvočni signal za uspešno izvedeno proceduro.

##### *Zvočni signal za neuspešno izvedeno proceduro*

Zvočni signal za neuspešno izvedeno proceduro na validirni enoti ima nižjo frekvenco sinusnega vala v primerjavi z uspešnim zvočnim signalom, in sicer 750Hz, ki predstavlja dvojni pisk z periodo 200ms predvajanja zvoka nato sledi perioda 200ms tišine, in nato ponovno perioda 200ms predvajanja zvoka za neuspešno proceduro (glej spodnjo sliko).



Zvočni signal neuspešno izvedene procedure.

V kolikor so indikacijski testi validirnih enot različnih ponudnikov v skladu z opisanim delovanjem posameznega indikacijskega testa v tem poglavju, validator opravi test poenotenega delovanja, in

lahko nadaljuje z naslednjo serijo testov. V primeru neuspešnih indikacijskih testov, se napravo zavrne in od ponudnika zahteva, da zagotovi indikacijo, kot jo definirajo indikacijski testi. V primeru neuspelega testa indikacije se lahko izvedejo preostali testi, vendar naprava ne dobi odobritve za integracijo v IJPP sistem.

Test:	IND006	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

### 10.5.2 PRAVILA IN PROCEDURE ZA VALIDACIJO TERMINALSKE OPREME

V nadaljevanju bodo opisana pravila, postopki za validacijo terminalske opreme (validator), ki bo vključena v sistem IJPP. Testi zajemajo kombinacijo strojne in programske opreme.

Testiranje terminalske opreme NI zahtevano v primerih:

- Spremembe komunikacijskega kanala, v kolikor že v osnovi podpira različne tipe (GPRS, Bluetooth, TCP/IP)
- Spremembe na programski opremi, ki ne vpliva na delovanje IJPP aplikacije
- Spremembe na strojni opremi, ki ne vplivajo na brezstično komunikacijo in komunikacijo z varnostnim elementom.

V zgoraj opisanih primerih, mora ponudnik terminalske opreme narejene spremembe ustrezno zabeležiti in po potrebi izvesti interne teste.

Testiranje terminalske opreme je zahtevano v primerih:

- Ob spremembi strojne opreme, ki se nanaša na brezstično komunikacijo ali komunikacijo z varnostnim elementom.
- Ob spremembi načina komunikacije zalednega sistema s terminalsko opremo
- Ob spremembi programske opreme, ki lahko vpliva na delovanje IJPP aplikacije (gonilniki, operacijski sistem)

Testni scenariji zajemajo kombinacijo strojne in programske opreme.

Za uporabo varnostnega elementa je potreben čitalnik varnostne kartice, ki je skladen s standardom ISO 7816. Le- ta mora poleg električnih in komunikacijskih zahtev zadoščati še pogojem, ki jih specificirajo testi v nadaljevanju.

Za izvedbo testov je potreben testni varnostni element in nameščena testna programska oprema »VEL\_test\_v.xxx«.

#### 10.5.2.1 Fizična namestitev varnostnega elementa

Za namestitev modula mora biti na voljo ustrezna reža. V tem testu je uporabljen testni varnostni element.

*Identifikacijska oznaka testa: VEL001*

#### *Scenarij testa*

Na testiranem validatorju je potrebno identificirati in omogočiti dostop do reže za vstavitve varnostnega elementa. Testni varnostni element se mora prilegati v režo validatorja; pri tem se morajo kontakti validatorja in varnostnega elementa ustrezno skleniti.

#### *Kdaj izvesti test?*

Test se mora izvesti vedno v primeru, kadar gre za nov tip validatorja.

Test:	VEL001	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.5.2.2 Logični test preizkusa protokola T=1**

Če je predhodni tip testa uspel, nadaljujemo s tem. Tudi ta tip testa izvedemo s pomočjo testnega varnostnega elementa. V tem testu preizkusimo komunikacijski protokol med čitalcem in varnostnim elementom.

*Identifikacijska oznaka testa: VEL002*

#### *Scenarij testa*

V testiranem validatorju mora biti pravilno vstavljen testni varnostni element. Ročno se požene testna aplikacija na validatorju, ki mora izpisati tekst: »MIFARE Plus SAM« in UID: »04XXXXXXXXXXXX«.

V kolikor test ni uspešen, ponovno vstavimo varnostni element. Če test v večkratnih poizkusih ni uspešen, mora ponudnik produkta poiskati ustrezne vzroke.

#### *Kdaj izvesti test?*

Test se mora izvesti vedno v primeru, kadar gre za nov tip validatorja in pri zamenjavi gonilnikov oz. bazne programske opreme validatorja. S testom potrdimo delovanje komunikacijskega protokola T=1.

Test:	VEL002
Tester: (Ime, Priimek)	
Datum testa (DD.MM.YYYY)	

Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

### 10.5.2.3 Validacija avtentikacijskega postopka brezstične kartice

Pogoj za izvedbo testa so opravljeni testi VEL001, VEL002, NFCXXX, ...V testu je uporabljena testna IJPP kartica in testni varnostni element.

Test služi preverjanju ustrezne komunikacije IJPP aplikacije z varnostnim elementom in brezstično kartico.

*Identifikacijska oznaka testa: AUTH001*

#### Scenarij testa

V testiranem validatorju mora biti pravilno vstavljen testni varnostni element in nameščena IJPP aplikacija.

Če test v večkratnih poizkusih ni uspešen, mora ponudnik produkta poiskati ustrezne vzroke.

#### Kdaj izvesti test?

Test se mora izvesti vedno v primeru, kadar gre za nov tip validatorja, pri zamenjavi gonilnikov oz. bazne programske opreme validatorja in zamenjavi IJPP aplikacije. S testom potrdimo delovanje avtentikacije testne IJPP kartice na validatorju.

Test:	AUTH001	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

### 10.5.3 TEST NFC MODULA

V tem sklopu testov so specificirani scenariji, s katerimi je moč potrditi ustrezno delovanje integriranega NFC modula. Operativnost NFC-ja predstavlja že sam standard ISO/IEC14443 in njegove izpeljanke (nadgradnje) A, B ,... in zadnja izpeljanka FeliCa JIS, ki se ga morajo proizvajalci opreme že tako ali tako držati. Kadar pa gre za lastne produkte z integriranimi NFC rešitvami, pa je smiselno izvesti posamezne teste, s katerimi se ugotavlja izpravnost NFC modula, zadostna jakost NFC bližnjega polja, oddaljenost delovanja, lokacija delovanja (ustrezno označevanje področja NFC komunikacije) itn. S samo nekaj osnovnimi testi, ki jih podajamo v nadaljevanju, se lahko hitro in učinkovito ugotovi ustreznost delovanja NFC komponente (modula). Glede na to da IJPP temelji na brezstični kartici, gre

za prvo skupino testov v seriji, ki morajo biti izvedeni, preden preidemo na višje nivojske teste, ki so pogojeni z omenjeno funkcionalnostjo.

### 10.5.3.1 Test prisotnosti NFC polja naprave z NFC modulom

Ta tip testa se izvede s pomočjo NFC Go detekcijske kartice. Slednja je identična klasični NFC kartici, le da je opremljena z LED svetilom. S približevanjem takšne kartice v magnetno polje kratkega dosega aktivne naprave (npr. validatorja) se kot posledica indukcije v tuljavi NFC Go kartice prižge LED dioda. Njena jakost svetilnosti se spreminja s približevanjem in oddaljevanjem NFC go kartice. Bližje kot je kartica NFC sprejemno/oddajni enoti aktivne naprave, večja je svetilnost LED. Z oddaljevanjem se jakost svetilnosti manjša, v primeru nahajanja izven magnetnega polja kratkega dosega aktivne naprave, pa LED dioda povsem ugasne.

*Identifikacijska oznaka testa: NFC001*

#### *Scenarij testa*

Testirana naprava z NFC modulom (testiranec) mora imeti NFC omogočen. Za test niso zahtevani nobeni posebni pogoji, temveč slični kot je realno delovno okolje, v katerem bodo integrirane rešitve delovale (avtobusi, prodajna mesta, trafike). V kolikor ima testiranec označeno mesto, kje se nahaja NFC čitalnik, se v naslednjem koraku NFC Go kartico približa označenemu mestu, ter spremlja odziv LED diode. V kolikor LED dioda zasveti, je jakost magnetnega polja kratkega dosega ustrezna, s tem pa tudi test uspešen. V kolikor mesto NFC čitalnika testiranca ni označeno, se s kartico po pasovih in v neposredni bližini (kolikor je mogoče) sprehajamo po frontalnem delu. Ko NFC Go testna kartica signalizira s prižgano LED diodo detekcijo NFC polja je test prav tako uspešen. V tem primeru priporočamo 'markiranje' položaja NFC čitalnika testirane naprave s standardno NFC nalepko (uporabniku prijazen vmesnik). V kolikor s prečesavanjem frontalne površine testiranca magnetnega polja kratkega dosega (NFC) ne zaznamo, se ponovno preverijo nastavitve NFC modula in testiranec se vrne ponovno v identično testno proceduro. V kolikor slednja ponovno ni uspešna se test smatra kot neuspešen, kar pomeni, da ne izpolnjuje pogojev za integracijo v IJPP sistem.

#### *Kdaj izvesti test?*

Test se mora izvesti vedno v primeru, kadar gre za zamenjavo NFC modula, oziroma njegove posamezne komponente, ali celotne tiskanine v primeru integrirane izvedbe. S testom potrdimo aktivnost NFC modula, ki predstavlja temelj za izvedbo brezstične funkcionalnosti.

Test:	NFC001	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

### 10.5.3.2 Logični test preizkusa protokola NFC

Če je predhodni tip testa uspel, nadaljujemo s tem. Tudi ta tip testa izvedemo s pomočjo NFC Go kartice, katera emulira tako imenovani NFC 'tag'. Ker smo s predhodnim testom potrdili ustrezno ustvarjanje magnetnega polja kratkega dosega, v tem testu preizkusimo še komunikacijski protokol med NFC čitalnikom in NFC brezstično kartico. V tem testu mora aktivna naprava (npr. validator) podpirati branje in v ustrezni obliki prikaz prebranega sporočila »IJPP Project«, ki je shranjen na NFC Go brezstični kartici.

*Identifikacijska oznaka testa: NFC002*

#### *Scenarij testa*

Tudi v tem testu mora testiranec imeti omogočen NFC modul (enako kot v predhodnem koraku). Za potrebe izvedbe testa testna oseba najprej poskrbi za zapis vsebine »IJPP Project« na NFC 'tag', ki ga emulira NFC Go kartica. V naslednjem koraku pripravljeno NFC Go kartico približamo področju NFC oddajnika aktivne naprave (npr. validator). V kolikor prebrana vsebina na testirancu ustreza tisti, ki smo jo zapisali »IJPP Project« na NFC 'tag' je test uspešen. Pomeni, da smo s tem testom potrdili tudi pravilno delovanje komunikacijskega protokola med testirancem in brezstično kartico. V kolikor test ni uspešen, ga ponovimo. Če test v večkratnih poizkusih ni uspešen, mora ponudnik produkta poiskati ustrezne vzroke (napačna raba modulacije, demodulacije, nosilne frekvence, okvara modula ipd.).

#### *Kdaj izvesti test?*

Test se mora izvesti vedno v primeru, kadar gre za zamenjavo NFC modula, oziroma njegove posamezne komponente, ali celotne tiskanine v primeru integrirane izvedbe. S testom potrdimo delovanje komunikacijskega protokola, ki prav tako predstavlja temelj za izvedbo brezstične funkcionalnosti.

Test:	<b>NFC002</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.5.3.3 Test razdalje (dometa) delovanja kartice NFC**

Kot velewa standard ISO/IEC 14443, je domet magnetnega polja aktivne NFC naprave do 10cm. V praksi velikokrat manj (realno do 5cm). Kratka razdalja komunikacije neposredno povečuje varnost prenosa in zmanjšuje potencialno nevarnost napada tretje osebe. Z obzirom maksimalne pozitivne uporabniške izkušnje je pomembno izvesti test, s katerim potrdimo ustreznost delovanja integriranih enot v IJPP sistem. Za test postavljamo sprejemljivo mejo oddaljenosti vsaj 3cm ob pogoju poznane lokacije NFC oddajno/sprejemne enote testiranca.

*Identifikacijska oznaka testa: NFC003*

#### *Scenarij testa*



Test pričnemo 15 cm nad lokacijo NFC sprejemno/oddajne enote tako, da NFC Go tesno kartico počasi vzporedno s frontalnim delom testiranca približujemo sprejemniku. V kolikor sta oba predhodna testa bila uspešna, mora testiranec na določeni razdalji (običajno med 0cm – 5cm) vzpostaviti komunikacijo z brezžično kartico, kar pomeni prenos podatkov (v tem primeru zgolj branje podatkov iz NFC 'taga' – NFC Go testne kartice). Razdalja, na kateri se je vzpostavila komunikacija, predstavlja realno operativno razdaljo testiranca. Za določitev srednje vrednosti se uporabi vzorec petih naključno izbranih testirancev, kjer se s pomočjo dobljenih meritev izračuna srednja vrednost. Test se smatra kot uspešen, če se komunikacija med testirancem in testno kartico vzpostavi na razdalji 0cm – 10cm. V kolikor več takšnih testov ni uspešnih, se je potrebno vrniti na predhodne testne scenarije.

#### *Kdaj izvesti test?*

Test se mora izvesti vedno v primeru, kadar gre za zamenjavo NFC modula, oziroma njegove posamezne komponente, ali celotne tiskanine v primeru integrirane izvedbe. S testom potrdimo ustrezno delovanje in izmenjavo informacij v predpisanih oddaljenostih, kot jih navaja standard ISO IEC 14443 in njegove izpeljanke.

Test:	<b>NFC003</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.5.3.4 Test prisotnosti dveh kartic NFC**

Zaradi zagotavljanja maksimalne varnosti in preprečevanja zlorab, in kot velewa standard ISO/IEC 14443 lahko ena aktivna NFC naprava komunicira zgolj s samo eno entiteto. Temu preizkusu je namenjen naslednji test.

#### *Identifikacijska oznaka testa: NFC004*

##### *Scenarij testa*

Test pričnemo z dvema NFC karticama sistema IJPP. Kartici, vzporedno ena z drugo, približamo magnetnemu polju testiranca (validator, terminal). Približujemo ju proti čelni površini aktivne NFC naprave. Test mora vrniti rezultat zavrnitve obeh, da se smatra kot uspešen. Le na ta način se zagotovi ustrezno varnost in prepreči morebitne zlorabe, kljub uporabi komunikacije na kratke razdalje. V kolikor test ni uspešen, se lahko izvedejo še drugi testi, vendar naprava ne dobi soglasja za integracijo v sistem IJPP.

#### *Kdaj izvesti test?*

Test te funkcionalnosti se izvede samo ob prvi integraciji končnega produkta s čimer se potrdi enako delovanje opreme različnih ponudnikov.

Test:	IND004	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

## 10.6 TESTI BREZSTIČNE IJPP KARTICE

Testi kartice IJPP so namenjeni preverjanju ustreznega delovanja kodirne enote in programske opreme validatorja, prodajnega mesta, ki operirajo s takšnimi podatki. Spodaj specifikirani testi so tipa zapiši in preberi. Primerjava poznane zapisane vsebine na kartico in prebrane vsebine iz kartice je ocenjevalni element, ki definira uspešnost oz. neuspešnost testa. Testi se izvedejo na osnovi specifikirane strukture posameznih datotek (v1.0) IJPP kartice.

### 10.6.1 TESTI ZNOTRAJ DATOTEKE INFO

Testi znotraj info datoteke vključujejo teste vsebin podatkov samo znotraj tega segmenta, glede na definirano in rezervirano velikost prostora za posamezni podatek, kot je definirano v poglavju 3 o strukturi brezstične kartice. V testno proceduro je vključena periferija kodirne enote, ki skrbi za zapis podatkov po bytih v definirane strukture posameznih datotek. Za branje specifičnega podatka znotraj info datoteke se uporabi še bralnik brezstičnih kartic s sposobnostjo prikazovanja vsebine posameznega byta v testirani datoteki. Vsebina INFO datoteke je definirana na sledeč način:

#### 10.6.1.1 Test zapisane verzije podatkovne datoteke INFO (VERSION\_INFO)

S tem testom se preverja pravilnost zapisa verzije podatkovne datoteke INFO na brezstično kartico. Če se v prihodnosti struktura kartice spremeni, se lahko skladno s tem spremeni tudi verzija podatkovne datoteke. Na ta način je mogoče na terminalskih enotah zagotoviti delovanje npr. po starem in novem režimu (s staro in novo podatkovno strukturo npr.).

*Identifikacijska oznaka testa: BKR001*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni skladno s specifikacijo. V testu se uporabi še kodirna enota in testni validator s prilagojeno aplikacijo in zmožnostjo prikazovanja prebranih podatkov posamezne podatkovne datoteke (npr. pošiljanje na serijsko povezavo, prikazovanje na zaslon). S pomočjo kodirne enote se naj prvi bajt podatkovne datoteke INFO postavi na vrednost 0 (0x00). Ker smo spremenili samo prvi byte podatkovne INFO datoteke, vsi ostali podatki podatkovne datoteke ostanejo na privzetih vrednostih (običajno 0xff). V naslednjem koraku s testnim

validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega podatka `VERSION_INFO`, jo lahko na tem mestu primerjamo s prebrano vsebino tega byta. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v info podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za vsebino prvega byta obravnavane podatkovne datoteke ne doseže uspeha (dokler zapisana vsebina ni enaka prebrani vsebini).

Test:	<b>BKR001</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.1.2 Test zapisane verzije podatkovne datoteke CARD INFO (VERSION\_CARD\_INFO)**

S tem testom se preverja pravilnost zapisa verzije podatkovne datoteke CARD INFO na brezstično kartico. Če se v prihodnosti struktura kartice spremeni, se skladno s tem lahko spremeni tudi verzija podatkovne datoteke in s tem celotna podatkovna struktura. Na ta način je mogoče na terminalskih enotah (validirne enote) zagotoviti delovanje npr. po starem in novem režimu (s staro in novo podatkovno strukturo npr.).

*Identifikacijska oznaka testa: BKR002*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni skladno s specifikacijo. V testu se uporabi še kodirna enota in testni validator s prilagojeno aplikacijo in zmožnostjo prikazovanja prebranih podatkov posamezne podatkovne datoteke (npr. pošiljanje na serijsko povezavo, prikazovanje na zaslon). S pomočjo kodirne enote se naj drugi byte obravnavane podatkovne datoteke postavi na vrednost 0x01. Ker smo spremenili samo drugi byte podatkovne INFO datoteke, vsi ostali podatki podatkovne datoteke ostanejo na privzetih vrednostih (običajno 0xff). V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega podatka `VERSION_CARD_INFO`, jo lahko na tem mestu primerjamo s prebrano vsebino tega byta. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v obravnavani podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za vsebino drugega byta obravnavane podatkovne datoteke ne doseže uspeha (dokler zapisana vsebina ni enaka prebrani vsebini).

Test:	<b>BKR002</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

### 10.6.1.3 Test zapisane verzije podatkovne datoteke DATA INDEX (VERSION\_DATA\_INDEX)

S tem testom se preverja pravilnost zapisa verzije podatkovne datoteke DATA INDEX na brezstično kartico. Če se v prihodnosti struktura kartice spremeni, se skladno s tem lahko spremeni tudi verzija podatkovne datoteke in s tem celotna podatkovna struktura. Na ta način je mogoče na terminalskih enotah (validirne enote) zagotoviti delovanje npr. po starem in novem režimu (s staro in novo podatkovno strukturo npr.).

*Identifikacijska oznaka testa: BKR003*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni skladno s specifikacijo. V testu se uporabi še kodirna enota in testni validator s prilagojeno aplikacijo in zmožnostjo prikazovanja prebranih podatkov posamezne podatkovne datoteke (npr. pošiljanje na serijsko povezavo, prikazovanje na zaslon). S pomočjo kodirne enote se naj tretji byte obravnavane podatkovne datoteke postavi na vrednost 0x02. Ker smo spremenili samo tretji byte podatkovne INFO datoteke, vsi ostali podatki podatkovne datoteke ostanejo na privzetih vrednostih (običajno 0xff). V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega podatka VERSION\_DATA\_INDEX, jo lahko na tem mestu primerjamo s prebrano vsebino tega byta. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v obravnavani podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za obravnavano vsebino tretjega byta obravnavane podatkovne datoteke ne doseže uspeha (dokler zapisana vsebina ni enaka prebrani vsebini).

Test:	<b>BKR003</b>
-------	---------------

Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### 10.6.1.4 Test zapisane verzije podatkovne datoteke DATA FILE (VERSION\_DATA\_FILE)

S tem testom se preverja pravilnost zapisa verzije podatkovne datoteke DATA FILE na brezstično kartico. Če se v prihodnosti struktura kartice spremeni, se skladno s tem lahko spremeni tudi verzija podatkovne datoteke in s tem celotna podatkovna struktura. Na ta način je mogoče na terminalskih enotah (validirne enote) zagotoviti delovanje npr. po starem in novem režimu (s staro in novo podatkovno strukturo npr.).

*Identifikacijska oznaka testa: BKR004*

##### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni skladno s specifikacijo. V testu se uporabi še kodirna enota in testni validator s prilagojeno aplikacijo in zmožnostjo prikazovanja prebranih podatkov posamezne podatkovne datoteke (npr. pošiljanje na serijsko povezavo, prikazovanje na zaslon). S pomočjo kodirne enote se naj četrti byte obravnavane podatkovne datoteke postavi na vrednost 0x03. Ker smo spremenili samo četrti byte podatkovne INFO datoteke, vsi ostali podatki podatkovne datoteke ostanejo na privzetih vrednostih (običajno 0xff). V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega podatka VERSION\_DATA\_FILE, jo lahko na tem mestu primerjamo s prebrano vsebino tega byta. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v obravnavani podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

##### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za obravnavano vsebino četrtega byta obravnavane podatkovne datoteke ne doseže uspeha (dokler zapisana vsebina ni enaka prebrani vsebini).

Test:	<b>BKR004</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>

Ugotovitve:	
-------------	--

#### 10.6.1.5 Test zapisane verzije podatkovne datoteke CHECK (VERSION\_CHECK)

S tem testom se preverja pravilnost zapisa verzije podatkovne datoteke CHECK na brezstično kartico. Če se v prihodnosti struktura kartice spremeni, se skladno s tem lahko spremeni tudi verzija podatkovne datoteke in s tem celotna podatkovna struktura. Na ta način je mogoče na terminalskih enotah (validirne enote) zagotoviti delovanje npr. po starem in novem režimu (s staro in novo podatkovno strukturo npr.).

*Identifikacijska oznaka testa: BKR008*

##### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni skladno s specifikacijo. V testu se uporabi še kodirna enota in testni validator s prilagojeno aplikacijo in zmožnostjo prikazovanja prebranih podatkov posamezne podatkovne datoteke (npr. pošiljanje na serijsko povezavo, prikazovanje na zaslon). S pomočjo kodirne enote se naj osmi byte podatkovne datoteke postavi na vrednost 0x07. Ker smo spremenili samo osmi byte podatkovne INFO datoteke, vsi ostali podatki podatkovne datoteke ostanejo na privzetih vrednostih (običajno 0xff). V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega podatka CHECK, jo lahko na tem mestu primerjamo s prebrano vsebino tega byta. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v obravnavani podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

##### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za obravnavano vsebino osmega byta podatkovne datoteke ne doseže uspeha (dokler zapisana vsebina ni enaka prebrani vsebini).

Test:	<b>BKR008</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### 10.6.1.6 Test zapisane verzije podatkovne datoteke PRODUCTS (VERSION\_PRODUCTS)

S tem testom se preverja pravilnost zapisa verzije podatkovne datoteke PRODUCTS na brezstično kartico. Če se v prihodnosti struktura kartice spremeni, se skladno s tem lahko spremeni tudi verzija

podatkovne datoteke in s tem celotna podatkovna struktura. Na ta način je mogoče na terminalskih enotah (validirne enote) zagotoviti delovanje npr. po starem in novem režimu (s staro in novo podatkovno strukturo npr.).

*Identifikacijska oznaka testa: BKR009*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni skladno s specifikacijo. V testu se uporabi še kodirna enota in testni validator s prilagojeno aplikacijo in zmožnostjo prikazovanja prebranih podatkov posamezne podatkovne datoteke (npr. pošiljanje na serijsko povezavo, prikazovanje na zaslon). S pomočjo kodirne enote se naj deveti byte obravnavane podatkovne datoteke postavi na vrednost 0x08. Ker smo spremenili samo deveti byte podatkovne INFO datoteke, vsi ostali podatki podatkovne datoteke ostanejo na privzetih vrednostih (običajno 0xff). V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega podatka PRODUCTS, jo lahko na tem mestu primerjamo s prebrano vsebino tega byta. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v obravnavani podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za vsebino devetega byta obravnavane podatkovne datoteke ne doseže uspeha (dokler zapisana vsebina ni enaka prebrani vsebini).

Test:	<b>BKR009</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

### **10.6.1.7 Test zapisane verzije podatkovne datoteke HISTORY (VERSION\_HISTORY\_VALIDATION)**

S tem testom se preverja pravilnost zapisa verzije podatkovne datoteke HISTORY\_VALIDATION na brezstično kartico. Če se v prihodnosti struktura kartice spremeni, se skladno s tem lahko spremeni tudi verzija podatkovne datoteke in s tem celotna podatkovna struktura. Na ta način je mogoče na terminalskih enotah (validirne enote) zagotoviti delovanje npr. po starem in novem režimu (s staro in novo podatkovno strukturo npr.).

*Identifikacijska oznaka testa: BKR010*

### Scenarij testa

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni skladno s specifikacijo. V testu se uporabi še kodirna enota in testni validator s prilagojeno aplikacijo in zmožnostjo prikazovanja prebranih podatkov posamezne podatkovne datoteke (npr. pošiljanje na serijsko povezavo, prikazovanje na zaslon). S pomočjo kodirne enote se naj deseti byte te podatkovne datoteke postavi na vrednost 0x09. Ker smo spremenili samo deseti byte podatkovne INFO datoteke, morajo vsi ostali podatki podatkovne datoteke ostati na privzetih vrednostih (običajno 0xff). V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega podatka HISTORY\_VALIDATION, jo lahko na tem mestu primerjamo s prebrano vsebino tega byta. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v obravnavani podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

### Kdaj izvesti test

Specificiran test kartice je potrebno izvajati dokler se za vsebino desetega byta obravnavane podatkovne datoteke 0x00 ne doseže uspeha (dokler zapisana vsebina ni enaka prebrani vsebini).

Test:	<b>BKR010</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

### 10.6.1.8 Test zapisane verzije podatkovne datoteke RELATIONS (VERSION\_RELATIONS)

S tem testom se preverja pravilnost zapisa verzije podatkovne datoteke RELATIONS na brezstično kartico. Če se v prihodnosti struktura kartice spremeni, se skladno s tem lahko spremeni tudi verzija podatkovne datoteke in s tem celotna podatkovna struktura. Na ta način je mogoče na terminalskih enotah (validirne enote) zagotoviti delovanje npr. po starem in novem režimu (s staro in novo podatkovno strukturo npr.).

Identifikacijska oznaka testa: BKR011

### Scenarij testa

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni skladno s specifikacijo. V testu se uporabi še kodirna enota in testni validator s prilagojeno aplikacijo in zmožnostjo prikazovanja prebranih podatkov posamezne podatkovne datoteke (npr. pošiljanje na serijsko povezavo, prikazovanje na zaslon). S pomočjo kodirne enote se naj enajsti byte te podatkovne datoteke postavi na vrednost 0x0A. Ker smo spremenili samo enajsti byte podatkovne INFO datoteke, vsi ostali podatki



podatkovne datoteke ostanejo na privzetih vrednostih (običajno 0xff). V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega podatka RELATIONS, jo lahko na tem mestu primerjamo s prebrano vsebino tega byta. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v tej podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za obravnavano vsebino tega byta podatkovne datoteke ne doseže uspeha (dokler zapisana vsebina ni enaka prebrani vsebini).

Test:	<b>BKR011</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

OPOMBA: V INFO podatkovni datoteki je preostalih 21 byte-ov rezerviranih za kasnejšo uporabo, in v trenutni podatkovni strukturi niso uporabljeni. Iz tega razloga morajo pri izvajanju testov INFO datoteke ostati na privzetih vrednostih (običajno 0xff).

### **10.6.2 TESTI ZNOTRAJ DATOTEKE CARD INFO**

Testi znotraj Card Info datoteke vključujejo teste vsebin podatkov samo znotraj tega segmenta, glede na definirano in rezervirano velikost prostora za posamezni podatek, kot je definirano v poglavju o strukturi brezstične kartice. Za posamezno velikost definiranega podatka je obvezna uporaba podatkovne strukture. V testno proceduro je vključena periferija kodirne enote, ki skrbi za zapis podatkov po bytih v definirano strukturo datoteke CARD INFO. Za branje specifičnega podatka znotraj info datoteke se uporabi še čitalnik brezstičnih kartic (validator, mobilni validator) s sposobnostjo prikazovanja vsebine posameznega byta v testirani datoteki.

#### **10.6.2.1 Test zapisanega statusa kartice v podatkovni datoteki CARD INFO (STATUS)**

S tem testom se preverja pravilnost zapisa posameznih podatkov po sklopih v CARD INFO podatkovno datoteko brezstične kartice. Če se v prihodnosti struktura kartice spremeni, se lahko skladno s tem spremeni tudi verzija podatkovne datoteke. Na ta način je mogoče na terminalskih enotah zagotoviti delovanj npr. po starem in novem režimu (s staro in novo podatkovno strukturo npr.).

*Identifikacijska oznaka testa: BKR012*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni skladno s specifikacijo. V testu se uporabi še kodirna enota in testni validator s prilagojeno aplikacijo in zmožnostjo prikazovanja

prebranih podatkov posamezne podatkovne datoteke (npr. pošiljanje na serijsko povezavo, prikazovanje na zaslon).

#### *Prvi testni scenarij*

S počjo kodirne enote se naj prvi bajt te podatkovne datoteke postavi na vrednost 0x01, kar predstavlja status neaktivne kartice. Ker smo spremenili samo prvi byte podatkovne CARD INFO datoteke, vsi ostali podatki podatkovne datoteke ostanejo na privzetih vrednostih (običajno 0xff). V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega podatka STATUS, jo lahko na tem mestu primerjamo s prebrano vsebino tega byta. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v tej podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### *Drugi testni scenarij*

S počjo kodirne enote se naj prvi bajt te podatkovne datoteke postavi na vrednost 0x02, kar predstavlja status aktivne kartice. Ker smo spremenili samo prvi byte podatkovne CARD INFO datoteke, vsi ostali podatki podatkovne datoteke ostanejo na privzetih vrednostih (običajno 0xff). V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega podatka STATUS, jo lahko na tem mestu primerjamo s prebrano vsebino tega byta. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v tej podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### *Tretji testni scenarij*

S počjo kodirne enote se naj prvi bajt te podatkovne datoteke postavi na vrednost 0x04, kar predstavlja status blokirane kartice. Ker smo spremenili samo prvi byte podatkovne CARD INFO datoteke, vsi ostali podatki podatkovne datoteke ostanejo na privzetih vrednostih (običajno 0xff). V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega podatka STATUS, jo lahko na tem mestu primerjamo s prebrano vsebino tega byta. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v tej podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### *Četrti testni scenarij*

S počjo kodirne enote se naj prvi bajt obravnavane podatkovne datoteke postavi na vrednost 0x08, kar predstavlja status začasno onemogočene kartice. Ker smo spremenili samo prvi byte podatkovne CARD INFO datoteke, vsi ostali podatki podatkovne datoteke ostanejo na privzetih vrednostih (običajno 0xff). V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega podatka STATUS, jo lahko na tem mestu primerjamo s prebrano vsebino tega byta. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v tej podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za vsebino prvega byta obravnavane podatkovne datoteke ne doseže uspeha (dokler zapisana vsebina ni enaka prebrani vsebini).

Test:	<b>BKR012</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

### 10.6.2.2 Test zapisanega datuma pričetka veljave kartice v podatkovni datoteki CARD INFO (DATUM PREČETKA VELJAVE KARTICE)

S tem testom se preverja pravilnost zapisa podatka datuma pričetka veljave kartice v CARD INFO podatkovni datoteki brezstične kartice. Če se v prihodnosti struktura kartice spremeni, se lahko skladno s tem spremeni tudi verzija podatkovne datoteke. Na ta način je mogoče na terminalskih enotah zagotoviti delovanje npr. po starem in novem režimu (s staro in novo podatkovno strukturo npr.).

*Identifikacijska oznaka testa: BKR013*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni skladno s specifikacijo. V testu se uporabi še kodirna enota in testni validator s prilagojeno aplikacijo in zmožnostjo prikazovanja prebranih podatkov podatkovne datoteke CARD INFO. S pomočjo kodirne enote se naj na naslednje štiri byte te podatkovne datoteke zapiše aktualni datum izvedbe testa, ki bo hkrati predstavljal datum pričetka veljave kartice. Ker smo spremenili samo naslednje 4 byte podatkovne CARD INFO datoteke, ostanejo vsi preostali byte podatkovne datoteke na privzetih vrednostih (običajno 0xff). V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega podatka DATUM PRIČETKA VELJAVE KARTICE, jo lahko na tem mestu primerjamo s prebrano vsebino štirih bytov, ki sledijo statusu. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v tej podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za obravnavano vsebino štirih bytov, ki sledijo statusu ne doseže uspeha (dokler zapisana vsebina ni enaka prebrani vsebini).

Test:	<b>BKR013</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>

Ugotovitve:	
-------------	--

### 10.6.2.3 Test zapisanega datuma konca veljave kartice v podatkovni datoteki CARD INFO (DATUM KONCA VELJAVE KARTICE)

S tem testom se preverja pravilnost zapisa podatka datuma konca veljave kartice v CARD INFO podatkovni datoteki brezstične kartice. Če se v prihodnosti struktura kartice spremeni, se lahko skladno s tem spremeni tudi verzija podatkovne datoteke. Na ta način je mogoče na terminalskih enotah zagotoviti delovanj npr. po starem in novem režimu (s staro in novo podatkovno strukturo npr.).

*Identifikacijska oznaka testa: BKR014*

#### Scenarij testa

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni na inicialno vrednost skladno s specifikacijo. V testu se uporabi še kodirna enota in testni validator s prilagojeno aplikacijo in zmožnostjo prikazovanja prebranih podatkov podatkovne datoteke CARD INFO. S pomočjo kodirne enote se naj na naslednje štiri byte podatkovne datoteke 0x01 (glej tabelo), ki sledijo štirim bytom z datumom pričetka veljavnosti kartice, zapiše datum, ki je definiran kot trenutni datum, ki mu prištejemo eno leto (trenutni datum + 1 leto). Zapisan datum predstavlja termin konca veljave kartice IJPP. Ker smo spremenili samo naslednje 4 byte podatkovne CARD INFO datoteke, ostanejo vsi preostali byti te podatkovne datoteke na privzetih vrednostih (običajno 0xff). V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega podatka DATUM KONCA VELJAVE KARTICE, jo lahko na tem mestu primerjamo s prebrano vsebino štirih bytov, ki sledijo po datumu pričetka veljave. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### Kdaj izvesti test

Specificiran test kartice je potrebno izvajati dokler se za obravnavano vsebino štirih bytov, ki sledijo datumu pričetka veljave kartice ne ujemajo (dokler zapisana vsebina ni enaka prebrani vsebini).

Test:	<b>BKR014</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

### 10.6.2.4 Test zapisanega tipa personifikacije v podatkovni datoteki CARD INFO (TIP PERSONIFIKACIJE) brezstične IJPP kartice

S tem testom se preverja pravilnost zapisa podatka tipa personifikacije kartice v CARD INFO podatkovni datoteki brezstične kartice. Če se v prihodnosti struktura kartice spremeni, se lahko skladno s tem spremeni tudi verzija podatkovne datoteke. Na ta način je mogoče na terminalskih enotah zagotoviti delovanje npr. po starem in novem režimu (s staro in novo podatkovno strukturo npr.).

*Identifikacijska oznaka testa: BKR015*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni na inicialno vrednost skladno s specifikacijo. V testu se uporabi še kodirna enota in testni validator s prilagojeno aplikacijo in zmožnostjo prikazovanja prebranih podatkov podatkovne datoteke CARD INFO.

#### *Prvi testni scenarij*

S pomočjo kodirne enote se naj na naslednji byte te podatkovne datoteke (glej tabelo), ki sledi štirim bytom z datumom konca veljavnosti kartice, zapiše tip personifikacije 0x01 – kartica ni namenjena personificiranju. Ker smo spremenili samo naslednji byte podatkovne CARD INFO datoteke, ostanejo vsi preostali byti te podatkovne datoteke na privzetih vrednostih (običajno 0xff). V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega podatka TIP PERSONIFIKACIJE, jo lahko na tem mestu primerjamo s prebrano vsebino byta, ki sledi po vsebini datuma konca veljave. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v tej podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### *Drugi testni scenarij*

S pomočjo kodirne enote se naj na naslednji byte te podatkovne datoteke (glej tabelo), ki sledi štirim bytom z datumom konca veljavnosti kartice, zapiše tip personifikacije 0x02 – nepersonificirana kartica ki je namenjena personificiranju. Ker smo spremenili samo naslednji byte podatkovne CARD INFO datoteke, ostanejo vsi preostali byti te podatkovne datoteke na privzetih vrednostih (običajno 0xff). V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega podatka TIP PERSONIFIKACIJE, jo lahko na tem mestu primerjamo s prebrano vsebino byta, ki sledi po vsebini datuma konca veljave. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v tej podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### *Tretji testni scenarij*

S pomočjo kodirne enote se naj na naslednji byte te podatkovne datoteke (glej tabelo), ki sledi štirim bytom z datumom konca veljavnosti kartice, zapiše tip personifikacije 0x04 – personificirana kartica. Ker smo spremenili samo naslednji byte podatkovne CARD INFO datoteke, ostanejo vsi preostali byti podatkovne datoteke 0x01 na privzetih vrednostih (običajno 0xff). V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega podatka TIP PERSONIFIKACIJE, jo lahko na tem mestu primerjamo s prebrano vsebino byta, ki sledi po vsebini datuma konca veljave. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v tej podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za obravnavano vsebino enega byta, ki sledi bytom z datumu konca veljave kartice, ne doseže ujemanja (dokler zapisana vsebina ni enaka prebrani vsebini).

Test:	BKR015	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.2.5 Test zapisanega posebnega statusa v podatkovni datoteki CARD INFO (POSEBNI STATUS) brezstične IJPP kartice**

S tem testom se preverja pravilnost zapisa podatka posebnega statusa v CARD INFO (0x01) podatkovni datoteki brezstične kartice. Če se v prihodnosti struktura kartice spremeni, se lahko skladno s tem spremeni tudi verzija podatkovne datoteke. Na ta način je mogoče na terminalskih enotah zagotoviti delovanje npr. po starem in novem režimu (s staro in novo podatkovno strukturo npr.).

*Identifikacijska oznaka testa: BKR016*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni na inicialno vrednost skladno s specifikacijo. V testu se uporabi še kodirna enota in testni validator s prilagojeno aplikacijo in zmožnostjo prikazovanja prebranih podatkov podatkovne datoteke CARD INFO.

#### *Testni scenarij*

S pomočjo kodirne enote se naj na naslednji byte te podatkovne datoteke (glej tabelo), ki sledi byte-u personifikacije, zapiše tip kartice 0x00 (tip IJPP). Ker smo spremenili samo naslednji byte podatkovne CARD INFO datoteke, ostanejo vsi preostali byti te podatkovne datoteke na privzetih vrednostih (običajno 0xff), in vsi predhodni byti na že postavljenih vrednostih. V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega podatka POSEBNI STATUS, jo lahko na tem mestu primerjamo s prebrano vsebino byta, ki sledi po vsebini tipa personifikacije. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v tej podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

*OPOMBA: Za POSEBNI STATUS je samo vrednost 0x00-> SAMO IJPP zaradi česar spodnja testna scenarija (drugi in tretji) odpadeta*

#### *Drugi testni scenarij*

S pomočjo kodirne enote se naj na naslednji byte te podatkovne datoteke (glej tabelo), ki sledi byte-u personifikacije (enak byte kot v prvem testnem scenariju), zapiše tip kartice 0x01 (tip Identifikacijska TDPM IJPP kartica). Ker smo spremenili samo naslednji byte podatkovne CARD INFO datoteke, ostanejo vsi preostali byti te podatkovne datoteke na privzetih vrednostih (običajno 0xff), in vsi predhodni byti na že postavljenih vrednostih. V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega podatka POSEBNI STATUS, jo lahko na tem mestu primerjamo s prebrano vsebino byta, ki sledi po vsebini tipa personifikacije. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v tej podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### *Tretji testni scenarij*

S pomočjo kodirne enote se naj na naslednji byte te podatkovne datoteke (glej tabelo), ki sledi byte-u personifikacije (enak byte kot v prvem in drugem testnem scenariju), zapiše tip kartice 0x02 (tip Trenutno nedefinirani tip kartice). Ker smo spremenili samo naslednji byte podatkovne CARD INFO datoteke, ostanejo vsi preostali byti te podatkovne datoteke na privzetih vrednostih (običajno 0xff), in vsi predhodni byti na že postavljenih vrednostih. V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega podatka POSEBNI STATUS, jo lahko na tem mestu primerjamo s prebrano vsebino byta, ki sledi po vsebini tipa personifikacije. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v tej podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za obravnavano vsebino enega byta, ki sledi byte-u tipa personifikacije, ne doseže ujemanja (dokler zapisana vsebina ni enaka prebrani vsebini).

Test:	<b>BKR016</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.2.6 Test zapisanih osebnih podatkov v podatkovno datoteko CARD INFO (IME, PRIIMEK, NASLOV) brezstične IJPP kartice**

S tem testom se preverja pravilnost zapisa osebnih podatkov v CARD INFO podatkovno datoteko brezstične kartice. Če se v prihodnosti struktura kartice spremeni, se lahko skladno s tem spremeni tudi verzija podatkovne datoteke. Na ta način je mogoče na terminalskih enotah zagotoviti delovanje npr. po starem in novem režimu (s staro in novo podatkovno strukturo npr.).

Identifikacijska oznaka testa: BKR017

#### Scenarij testa

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni na inicialno vrednost skladno s specifikacijo. V testu se uporabi še kodirna enota in testni validator s prilagojeno aplikacijo in zmožnostjo prikazovanja prebranih podatkov podatkovne datoteke CARD INFO.

S pomočjo kodirne enote se naj na naslednjih 200 byte-ov podatkovne datoteke (glej tabelo), ki sledi byte-u posebnega statusa, zapišejo osebni podatki v naslednjem vrstnem redu; ime, priimek, naslov, poštna številka, pošta. Podatki ki se naj zapišejo so sledeči: (»Testni Uporabnik, Ulica testnega uporabnika 77, 1000 Ljubljana«). Ker smo spremenili vsebino 200-tih byte-ov podatkovne CARD INFO datoteke, ki sledijo byte-u posebnega statusa, ostanejo vsi preostali byti te podatkovne datoteke na privzetih vrednostih (običajno 0xff), in vsi predhodni byti na že postavljenih vrednostih. V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanih osebnih podatkov, jo lahko na tem mestu primerjamo s prebrano vsebino 200-tih byte-ov, ki sledijo po vsebini posebnega statusa. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v tej podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### Kdaj izvesti test

Specificiran test kartice je potrebno izvajati dokler se za obravnavano vsebino 200-tih byte-ov, ki sledijo byte-u posebnega statusa, ne doseže ujemanja (dokler zapisana vsebina ni enaka prebrani vsebini), in ob pogoju, da preostali byte-i pred in po obravnavanem podatku ostanejo nespremenjeni .

Test:	BKR017	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### 10.6.2.7 Test zapisane EMŠO številke v podatkovno datoteko CARD INFO brezstične IJPP kartice

S tem testom se preverja pravilnost zapisa 13-mestne EMŠO številke v CARD INFO podatkovno datoteko brezstične kartice. Če se v prihodnosti struktura kartice spremeni, se lahko skladno s tem spremeni tudi verzija podatkovne datoteke. Na ta način je mogoče na terminalskih enotah zagotoviti delovanje npr. po starem in novem režimu (s staro in novo podatkovno strukturo npr.).

Identifikacijska oznaka testa: BKR018

#### Scenarij testa



Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni na inicialno vrednost skladno s specifikacijo. V testu se uporabi še kodirna enota in testni validator s prilagojeno aplikacijo in zmožnostjo prikazovanja prebranih podatkov podatkovne datoteke CARD INFO.

S pomočjo kodirne enote se naj na naslednjih 13 byte-ov te datoteke, ki sledijo byte-om osebnih podatkov, zapiše naslednja 13-mestna EMŠO številka: 3003971500800. Ker smo v tem testu spremenili samo vsebino 13-tih byte-ov podatkovne CARD INFO datoteke, ki sledijo 200-tim byte-om osebnih podatkov, ostanejo vsi preostali byti te podatkovne datoteke na privzetih vrednostih (običajno 0xff), in vsi predhodni byti na že postavljenih vrednostih. V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega EMŠO podatka, jo lahko na tem mestu primerjamo s prebrano vsebino 13-tih byte-ov, ki sledijo po vsebini osebnih podatkov. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v tej podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za obravnavano vsebino 13-tih byte-ov, ki sledijo byte-om osebnih podatkov, ne doseže ujemanja (dokler zapisana vsebina ni enaka prebrani vsebini), in ob pogoju, da preostali byte-i pred in po obravnavanem podatku ostanejo nespremenjeni .

Test:	<b>BKR018</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.2.8 Test zapisane DAVČNE številke v podatkovno datoteko CARD INFO brezstične IJPP kartice**

S tem testom se preverja pravilnost zapisa 8-mestne davčne številke v CARD INFO podatkovno datoteko brezstične kartice. Če se v prihodnosti struktura kartice spremeni, se lahko skladno s tem spremeni tudi verzija podatkovne datoteke. Na ta način je mogoče na terminalskih enotah zagotoviti delovanje npr. po starem in novem režimu (s staro in novo podatkovno strukturo npr.).

*Identifikacijska oznaka testa: BKR019*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni na inicialno vrednost skladno

s specifikacijo. V testu se uporabi še kodirna enota in testni validator s prilagojeno aplikacijo in zmožnostjo prikazovanja prebranih podatkov podatkovne datoteke CARD INFO.

S pomočjo kodirne enote se naj na naslednjih 8 byte-ov te podatkovne datoteke (glej tabelo), ki sledijo byte-om EMŠO številke, zapiše naslednja 8-mestna davčna številka: 43381177. Ker smo v tem testu spremenili samo vsebino 8-ih byte-ov podatkovne CARD INFO datoteke, ki sledijo 13-tim byte-om EMŠO številke, ostanejo vsi preostali byti podatkovne datoteke na privzetih vrednostih (običajno 0xff), in vsi predhodni byti na že postavljenih vrednostih. V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisane davčne številke, jo lahko na tem mestu primerjamo s prebrano vsebino 8-ih byte-ov, ki sledijo po vsebini EMŠO številke. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v obravnavani podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za obravnavano vsebino 8-ih byte-ov, ki sledijo byte-om emšo številke, ne doseže ujemanja (dokler zapisana vsebina ni enaka prebrani vsebini), in ob pogoju, da preostali byte-i pred in po obravnavanem podatku ostanejo nespremenjeni.

Test:	<b>BKR019</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.2.9 Test zapisanega DATUMA INICIALIZACIJE kartice v podatkovno datoteko CARD INFO brezstične IJPP kartice**

S tem testom se preverja pravilnost zapisa datuma inicializacije IJPP kartice v CARD INFO podatkovno datoteko brezstične kartice. Če se v prihodnosti struktura kartice spremeni, se lahko skladno s tem spremeni tudi verzija podatkovne datoteke. Na ta način je mogoče na terminalskih enotah zagotoviti delovanje npr. po starem in novem režimu (s staro in novo podatkovno strukturo npr.).

*Identifikacijska oznaka testa: BKR020*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni na inicialno vrednost skladno s specifikacijo. V testu se uporabi še kodirna enota in testni validator s prilagojeno aplikacijo in zmožnostjo prikazovanja prebranih podatkov podatkovne datoteke CARD INFO.

S pomočjo kodirne enote se naj na naslednje 4 byte te podatkovne datoteke, ki sledijo byte-om davčne številke, zapiše kar trenutni datum opravljanja testa v naslednji obliki: dd.mm.yyyy. Ker smo v tem

testu spremenili samo vsebino 4-ih byte-ov podatkovne CARD INFO datoteke, ki sledijo 8-im byte-om davčne številke, ostanejo vsi preostali byti te podatkovne datoteke na privzetih vrednostih (običajno 0xff), in vsi predhodni byti na že postavljenih vrednostih. V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega datuma inicializacije IJPP kartice, jo lahko na tem mestu primerjamo s prebrano vsebino 4-ih byte-ov, ki sledijo po vsebini davčne številke. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za obravnavano vsebino 4-ih byte-ov, ki sledijo byte-om davčne številke, ne doseže ujemanja (dokler zapisana vsebina ni enaka prebrani vsebini), in ob pogoju, da preostali byte-i pred in po obravnavanem podatku ostanejo nespremenjeni.

Test:	<b>BKR020</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.2.10 Test zapisanega datuma personifikacije kartice v podatkovno datoteko CARD INFO brezstične IJPP kartice**

S tem testom se preverja pravilnost zapisa datuma personifikacije IJPP kartice v CARD INFO podatkovno datoteko brezstične kartice, in sicer samo v primeru, ko gre za personificirano kartico. Če se v prihodnosti struktura kartice spremeni, se lahko skladno s tem spremeni tudi verzija podatkovne datoteke. Na ta način je mogoče na terminalskih enotah zagotoviti delovanje npr. po starem in novem režimu (s staro in novo podatkovno strukturo npr.).

**Opomba:** Test velja samo za personificirane IJPP kartice! Če je kartica nepersonificirana je nastavljena vrednost teh štirih byte-ov enaka 0xFFFFFFFF!

*Identifikacijska oznaka testa: BKR021*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni na inicialno vrednost skladno s specifikacijo. V testu se uporabi še kodirna enota in testni validator s prilagojeno aplikacijo in zmožnostjo prikazovanja prebranih podatkov podatkovne datoteke CARD INFO.

S pomočjo kodirne enote se naj na naslednje 4 byte podatkovne datoteke, ki sledijo byte-om inicializacije kartice, zapiše kar trenutni datum opravljanja testa v naslednji obliki: dd.mm.yyyy. Ker

smo v tem testu spremenili samo vsebino 4-ih byte-ov podatkovne CARD INFO datoteke, ki sledijo 4-im byte-om datuma inicializacije, ostanejo vsi preostali byti te podatkovne datoteke na privzetih vrednostih (običajno 0xff), in vsi predhodni byti na že postavljenih vrednostih. V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega datuma personifikacije IJPP kartice, jo lahko na tem mestu primerjamo s prebrano vsebino 4-ih byte-ov, ki sledijo byte-om z informacijo datuma inicializacije kartice. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v tej podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za obravnavano vsebino 4-ih byte-ov, ki sledijo byte-om datuma inicializacije kartice, ne doseže ujemanja (dokler zapisana vsebina ni enaka prebrani vsebini), in ob pogoju, da preostali byte-i pred in po obravnavanem podatku ostanejo nespremenjeni.

Test:	<b>BKR021</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

**Opomba:** preostalih 16 byte-ov CARD INFO datoteke je rezerviranih za morebitne kasnejše nadgradnje. Slednji morajo biti od samega začetka izvajanja kartičnih testov na vrednosti 0xff, in se v sklopu izvajanja testnih procedur ne smejo spreminjati! Če se rezervirana vsebina spreminja, se test avtomatično ovrednoti kot neuspešen!

### **10.6.3 TESTI ZNOTRAJ DATOTEKE DATA INDEX**

Testi znotraj DATA INDEX datoteke vključujejo teste vsebin podatkov samo znotraj tega segmenta, glede na definirano in rezervirano velikost prostora za posamezni podatek, kot je definirano v poglavju o strukturi brezstične kartice. V testno proceduro je vključena periferija kodirne enote, ki skrbi za zapis podatkov po bytih v definirano strukturo datoteke DATA INDEX. Za branje specifičnega podatka znotraj DATA INDEX datoteke se uporabi še čitalnik brezstičnih kartic (validator, mobilni validator) s sposobnostjo prikazovanja vsebine posameznega byta v testirani datoteki.

#### **10.6.3.1 Test zapisanega števila ponudnikov storitev (SP\_ID) v podatkovni datoteki DATA INDEX**

S tem testom se preverja pravilnost zapisa števila posameznih ponudnikov storitev v DATA INDEX podatkovno datoteko brezstične kartice. Če se v prihodnosti struktura kartice spremeni, se lahko skladno s tem spremeni tudi verzija podatkovne datoteke. Na ta način je mogoče na terminalskih

enotah zagotoviti delovanje npr. po starem in novem režimu (s staro in novo podatkovno strukturo npr.).

*Identifikacijska oznaka testa: BKR022*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni skladno s specifikacijo. V testu se uporabi še kodirna enota in testni validator s prilagojeno aplikacijo in zmožnostjo prikazovanja prebranih podatkov posamezne podatkovne datoteke (npr. pošiljanje na serijsko povezavo, prikazovanje na zaslon). S pomočjo kodirne enote se naj v prih 16 Byte-ov zapiše število ponudnikov storitev (npr. 8). Ker smo število ponudnikov storitev zapisali v prvih 16 Byte-ov, morajo vsi preostali podatki te podatkovne datoteke ostati na privzetih vrednostih (običajno 0xff). V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega podatka števila ponudnikov storitev, jo lahko na tem mestu primerjamo s prebrano vsebino prvih 32 Byte-ov DATA INDEX datoteke. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v tej podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za obravnavano vsebino prvih 16 Byte-ov te podatkovne datoteke ne doseže uspeha (dokler zapisana vsebina ni enaka prebrani vsebini).

Test:	<b>BKR022</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

### **10.6.3.2 Test zapisanega formata podatkovne datoteke ponudnika v datoteki DATA INDEX**

S tem testom se preverja pravilnost zapisa podatkovnega formata ponudnika storitev v DATA INDEX datoteko brezstične kartice. Če se v prihodnosti struktura kartice spremeni, se lahko skladno s tem spremeni tudi verzija podatkovne datoteke. Na ta način je mogoče na terminalskih enotah zagotoviti delovanje npr. po starem in novem režimu (s staro in novo podatkovno strukturo npr.).

*Identifikacijska oznaka testa: BKR023*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni skladno s specifikacijo. V testu

se uporabi še kodirna enota in testni validator s prilagojeno aplikacijo in zmožnostjo prikazovanja prebranih podatkov posamezne podatkovne datoteke (npr. pošiljanje na serijsko povezavo, prikazovanje na zaslon). S pomočjo kodirne enote se v naj v naslednji Byte, ki sledi 16 Byte-om števila ponudnikov, zapiše format prvega ponudnika, in sicer 0x00. Preostali formati zajeti v testu so 0x01, 0x02 in 0x03. Analiza vsebine podatkovne datoteke, ki jo definira format tega testa bo izvedena v poglavju 10.6.10. Ker smo format zapisali v naslednji Byte, ki sledi 16-tim Byte-om s številom ponudnikov, morajo vsi naslednji Byte-i te podatkovne datoteke DATA INDEX ostati nespremenjeni (0xff). V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega podatka formata, jo lahko na tem mestu primerjamo s prebrano vsebino Byte-a, ki sledi 16-tim Bytom, ki zajemajo število ponudnikov storitev v DATA INDEX datoteki. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v tej podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za obravnavano vsebino naslednjega Byte-a, ki sledi predhodnim 32-tim v tej podatkovni datoteki ne dosežemo uspeha (dokler zapisana vsebina ni enaka prebrani vsebini).

Test:	<b>BKR023</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.3.3 Test zapisane informacije o zamiku (OFFSET) podatkov ponudnika storitev v datoteki DATA INDEX**

S tem testom se preverja pravilnost zapisa zamika podatkov ponudnika storitev v DATA INDEX datoteki brezstične kartice. Če se v prihodnosti struktura kartice spremeni, se lahko skladno s tem spremeni tudi verzija podatkovne datoteke. Na ta način je mogoče na terminalskih enotah zagotoviti delovanje npr. po starem in novem režimu (s staro in novo podatkovno strukturo npr.).

*Identifikacijska oznaka testa: BKR024*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni skladno s specifikacijo. V testu se uporabi še kodirna enota in testni validator s prilagojeno aplikacijo in zmožnostjo prikazovanja prebranih podatkov posamezne podatkovne datoteke (npr. pošiljanje na serijsko povezavo, prikazovanje na zaslon). S pomočjo kodirne enote se naj v naslednja 2 Byte-a, ki sledita Byte-u formata, zapiše informacija o zamiku podatkov ponudnika storitev. Analiza vsebine podatkovne datoteke, bo

izvedena v poglavju 10.6.10. Ker smo informacijo o zamiku podatkov zapisali v 2 Byte-a, ki sledita Byte-u formata, morajo vsi naslednji Byte-i te podatkovne datoteke DATA INDEX ostati nespremenjeni (0xff). V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega podatka zamika, jo lahko na tem mestu primerjamo s prebrano vsebino dveh Byte-ov, ki zajemata informacijo o zamiku podatkov ponudnika storitev v DATA INDEX datoteki. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v tej podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za obravnavano vsebino dveh Byte-ov, ki sledita predhodnemu (FORMAT) v tej podatkovni datoteki ne doseže uspeha (dokler zapisana vsebina ni enaka prebrani vsebini).

Test:	<b>BKR024</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.3.4 Test zapisane informacije o dolžini (LENGTH) podatkov ponudnika storitev v datoteki DATA INDEX**

S tem testom se preverja pravilnost zapisa informacije o dolžini podatkov ponudnika storitev v DATA INDEX datoteki brezstične kartice. Če se v prihodnosti struktura kartice spremeni, se lahko skladno s tem spremeni tudi verzija podatkovne datoteke. Na ta način je mogoče na terminalskih enotah zagotoviti delovanje npr. po starem in novem režimu (s staro in novo podatkovno strukturo npr.).

*Identifikacijska oznaka testa: BKR025*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni skladno s specifikacijo. V testu se uporabi še kodirna enota in testni validator s prilagojeno aplikacijo in zmožnostjo prikazovanja prebranih podatkov posamezne podatkovne datoteke (npr. pošiljanje na serijsko povezavo, prikazovanje na zaslon). S pomočjo kodirne enote se naj v naslednji Byte, ki sledi dvema Byte-oma informacije o zamiku, zapiše informacija o dolžini podatkov ponudnika storitev. Analiza vsebine podatkovne datoteke bo izvedena v poglavju 10.6.10. Ker smo informacijo o dolžini podatkov zapisali v en Byte, ki sledi Byte-oma z informacijo zamika, morajo vsi naslednji Byte-i te podatkovne datoteke (DATA INDEX) ostati nespremenjeni (0xff). V naslednjem koraku s testnim validatorjem preberemo zapisano vsebino kartice, in podatke izpišemo ali na zaslon ali jih posredujemo na serijsko povezavo. Ker poznamo vsebino zapisanega podatka dolžine, jo lahko na tem mestu primerjamo s prebrano

vsebino Byte-a, ki definira dolžino podatkov v DATA INDEX datoteki. Če sta oba podatka (zapisani in prebrani) enaka, in ostala vsebina v podatkovni datoteki nespremenjena, se test vrednoti kot uspešen.

#### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za vsebino obravnavanega Byte-a v tej podatkovni datoteki ne doseže uspeha (dokler zapisana vsebina ni enaka prebrani vsebini).

Test:	<b>BKR025</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.4 TESTI ZNOTRAJ DATOTEKE DATA FILE**

Testi znotraj DATA FILE datoteke vključujejo teste vsebin podatkov ponudnikov storitev samo znotraj tega segmenta, glede na definirano in rezervirano velikost prostora za posamezni podatek, kot je definirano v poglavju o strukturi brezstične kartice. V testno proceduro je vključena periferija kodirne enote, ki skrbi za zapis podatkov po bytih v definirano strukturo datoteke DATA FILE. Za branje specifičnega podatka znotraj DATA FILE datoteke se uporabi še čitalnik brezstičnih kartic (validator, mobilni validator) s sposobnostjo prikazovanja vsebine posameznega byta v testirani datoteki.

##### **10.6.4.1 Test zapisanih podatkov ponudnika v podatkovni datoteki DATA FILE**

S tem testom se preverja zapis podatkov ponudnika storitve v DATA FILE podatkovni datoteki brezstične kartice. Če se v prihodnosti struktura kartice spremeni, se lahko skladno s tem spremeni tudi verzija podatkovne datoteke. Na ta način je mogoče na terminalskih enotah zagotoviti delovanje npr. po starem in novem režimu (s staro in novo podatkovno strukturo npr.).

*Identifikacijska oznaka testa: BKR026A*

#### *Scenarij testa za format 0x00 podatkovne datoteke*

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere vsi predhodni in naknadni podatki morajo biti postavljeni skladno s specifikacijo. Za potrebe testa se uporabi še periferija kodirne enote, ki služi pisanju podatkov na brezstično kartico v formatu 0x00 (ti podatki definirajo potnikov status) podatkovne datoteke in periferija čitalca (validator, parser, RS232 vmesnik z algoritmom zajemanja podatkov, USB vmesnik, ...). Pred izvedbo testa morajo biti vsi podatki rezervirani za format 0x00 postavljeni na privzeto vrednost (0xff). V naslednjem koraku s kodirno enoto na kartico zapišemo naslednje podatke v sledečem vrstnem redu:

- ID statusa (2 Byte-a) -> v testu se naj zapiše vrednost 7
- Datum pričetka veljave statusa (4 Byte-i) -> v testu se vpiše trenutni datum opravljanja testa



- Datum konca veljave statusa (4 Byti) -> v testu se vpiše trenutni datum opravljanja testa + 1 dan

Podatki se zapišejo v seriji, in se v naslednji proceduri branja (preverjanje zapisa)čitajo in prikazujejo tudi v seriji - za razliko od predhodnih podatkov katere smo v predhodnih testih preverjali korak za korakom. Če je pogoj, da so zapisani podatki enaki prebranim, in vsi predhodno testirani podatki v prehodnih podatkovnih datotekah nespremenjeni, in vsi naslednji podatki na privzetih vrednostih (običajno 0xff), se test vrednoti kot uspešen. V kolikor test ni uspešen je potrebno ugotavljanje vzroka, ali je napaka na strani kodirne enote ali na strani čitalca, validatorja, namenske periferije s programsko opremo namenjeno analizi branih podatkov. Skladno s tem se ukrepa na posameznem segmentu za odpravo napake. V primeru neuspešnega testa je potrebno test ponavljati, prehod na naslednje teste, ki se tičejo brezstične kartice pa ni mogoč! Pogoj za napredovanje na naslednji kartični test je 100 odstotna uspešnost tega testa.

#### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za vsebino obravnavanih Byte-ov (glej zgornje alineje) v tej podatkovni datoteki ne doseže uspeha (dokler zapisana vsebina ni enaka prebrani vsebini).

Test:	<b>BKR026A</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### *Identifikacijska oznaka testa: BKR026B*

#### *Scenarij testa za format 0x01 podatkovne datoteke*

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere vsi predhodni in naknadni podatki morajo biti postavljeni skladno s specifikacijo. Za potrebe testa se uporabi še periferija kodirne enote, ki služi pisanju podatkov v formatu 0x01 (ti podatki definirajo vlogo za subvencioniranje)) podatkovne datoteke na brezstično kartico in periferija čitalca (validator, parser, RS232 vmesnik z algoritmom zajemanja podatkov, USB vmesnik, ...). Pred izvedbo testa morajo biti vsi podatki rezervirani za format 0x01 postavljeni na privzeto vrednost (0xff). V naslednjem koraku s kodirno enoto na kartico zapišemo naslednje podatke v sledečem vrstnem redu:

- Šifra vloge za subvencioniranje (20 Byte-ov) -> v testu se naj zapiše vrednost 17
- Datum pričetka veljave vloge (4 Byte-i) -> v testu se vpiše trenutni datum opravljanja testa
- Datum konca veljave vloge (4 Byti) -> v testu se vpiše trenutni datum opravljanja testa + 1 dan
- Vrsta vloge (1 Byte) -> v testu se naj zapiše vrednost 1 (mesečna)
- Status (1 Byte) -> v testu se naj zapiše vrednost 1 (študent)
- TariffLocationID za prvi mestni promet na vlogi (1 Byte) -> v testu se naj zapiše vrednost 8
- Število področij za prvi mestni promet na vlogi (1 Byte) -> v testu se naj zapiše vrednost 1

- TariffLocationID za drugi mestni promet na vlogi (1 Byte) -> v testu se naj zapiše vrednost 9
- Število področij za drugi mestni promet na vlogi (1 Byte) -> v testu se naj zapiše vrednost 2
- TariffLocationID za tretji mestni promet na vlogi (1 Byte) -> v testu se naj zapiše vrednost 10
- Število področij za tretji mestni promet na vlogi (1 Byte) -> v testu se naj zapiše vrednost 3

Podatki se zapišejo v seriji, in se v naslednji proceduri branja (preverjanje zapisa)čitajo in prikazujejo prav tako v seriji - za razliko od predhodnih testov v katerih smo preverjali korak za korakom. Če je pogoj, da so zapisani podatki enaki prebranim, in vsi predhodno testirani podatki v prehodnih podatkovnih datotekah nespremenjeni, in vsi naslednji podatki na privzetih vrednostih (običajno 0xff), se test vrednoti kot uspešen. V kolikor test ni uspešen je potrebno ugotavljanje vzroka, ali je napaka na strani kodirne enote ali na strani čitalca, validatorja, namenske periferije s programsko opremo namenjeno analizi branih podatkov. Skladno s tem se ukrepa na posameznem segmentu za odpravo napake. V primeru neuspešnega testa je potrebno test ponavljati, prehod na naslednje teste, ki se tičejo brezstične kartice pa ni mogoč! Pogoj za napredovanje na naslednji kartični test je 100 odstotna uspešnost tega testa.

#### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za vsebino obravnavanih Byte-ov (glej zgornje alineje) v tej podatkovni datoteki ne doseže uspeha (dokler zapisana vsebina ni enaka prebrani vsebini).

Test:	<b>BKR026B</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

OPOMBA: Datoteke z indeksi FID 0x04, FID 0x05 in FID 0x06 so rezervirane za kasnejšo uporabo, in se jih v sklopu IJPP ne uporablja. Iz tega razloga niso vključene v testno proceduro. Njihove velikosti 64 Byte-ov, 26 Byte-ov in 26 Byte-ov so rezervirane, primarne vrednosti pa so nastavljene na privzeto vrednost (0xff...)!

#### **10.6.5 TESTI ZNOTRAJ DATOTEKE CHECK – KONTROLA STANJA**

Test znotraj CHECK datoteke vključuje poizvedbo o zapisu vsebine po končani transakciji samo znotraj tega segmenta, glede na definirano in rezervirano velikost prostora za posamezni podatek, kot je definirano v poglavju o strukturi brezstične kartice. Sklop testov je v osnovi namenjen višje nivojskim aplikacijam, ki se tičejo terminalske opreme, ki upravlja s podatki datoteke CHECK. Pri transakciji prodajno mesto zapiše podatke transakcije v CHECK datoteko, ki vsebuje: naključni TRID, ki ga generira terminalska enota (prodajno mesto, ipd.), čas zadnje transakcije, identiteto zadnje transakcije in žeton produkta. Ta datoteka služi poizvedbi, ali je zapis na IJPP kartico dejansko uspel ali ne, za kar skrbi

logika prodajnega mesta. V spodnji tabeli je definirana in rezervirana velikost prostora za zapis podatkov v datoteko CHECK. V testno proceduro je vključena periferija prodajnega mesta in IJPP kartica. Za branje specifičnega podatka znotraj CHECK datoteke se prav tako uporabi periferija prodajnega mesta, kartomata, s sposobnostjo pošiljanja prebranih podatkov na serijsko povezavo.

#### **10.6.5.1 Test zapisanih podatkov v CHECK datoteki**

Test se izvaja v celoti, in se ga ne obravnava po posameznih Byte-ih iz razloga, ker se morebitna nepravilnost zapisanih podatkov te datoteke odraža kasneje na višjem nivoju terminalske opreme (prodajnega mesta), kjer se preverja uspešnost zapisa na kartico. Ker terminalska oprema generira naključno identiteto transakcije, je kot tester črne škatle ne poznamo. Ker pa se pri transakciji 'Session check' najprej preveri naključna identiteta kartice, ko uporabnik ponovno prisloni kartico (izvajanje poizvedbe ali je polnitev uspela), in si je terminal le to zapomnil, je na tem mestu nesmiselno ugotavljati ali je vsebina posameznih Byte-ov ustrezna. V CHECK datoteko se zapiše naslednja vsebina: naključni id transakcije (4 Byte-i), čas zadnje uspešne transakcije (4 Byte-i), zadnji ID uspešne transakcije (4 Byte-i), nezaseden podatek RFU (4 Byte-i), žeton produkta (4 Byte-i), število opozoril s strani kontrolske službe (1 Byte) in nezaseden podatek RFU v dolžini 11 Byte-ov.

*Identifikacijska oznaka testa: BKR027*

##### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni skladno s specifikacijo. Za potrebe testa se uporabi še periferija prodajnega mesta. Za potrebe izvedbe tega testa je potrebno upoštevati, da s kartico IJPP še ni bila izvedena nobena transakcija. Če je ta pogoj izpolnjen preberemo vsebino kartice, natančneje vsebino CHECK datoteke, z orodjem COLLIS in pripadajočo programsko opremo. V tem primeru morajo biti podatki CHECK datoteke postavljeni na privzeto vrednost 0xff. Če to drži lahko nadaljujemo proceduro testa. Naslednji korak je izvedba transakcije, ki se lahko za potrebe testa izbere poljubno. Ker po transakciji uporabnik ni 100-odstotno prepričan ali je nalaganje uspelo, se na terminalski enoti prodajnega mesta izvede CHECK procedura, ki od uporabnika zahteva ponovno prislonitev kartice. Pred izvedbo check procedure se lahko z orodjem COLLIS ponovno prebere vsebina CHECK datoteke, in se jo primerja z vsebino pred izvedbo transakcije. V kolikor so podatki različni od 0xff, se smatra da se je določena vsebina vpisala v CHECK datoteko. Če pa je vsebina dejansko prava, pa pove nadaljevanje testa, ki se že tiče algoritma aplikacije prodajnega mesta. Ko uporabnik prisloni ponovno IJPP kartico, se primerjajo podatki CHECK datoteke s tistimi, ki jih terminal hrani pri sebi ob pogoju, da v tem času ni bila prislonjena nobena druga kartica in se ni izvedla nobena druga transakcija. Pri tem mora algoritem prodajnega mesta iz zapisanih podatkov CHECK datoteke in podatkov hranjenih pri sebi pravilno ugotoviti zapis transakcije, če se podatki seveda ujemajo. Če se podatki ne ujemajo in preverjanje CHECK datoteke ne uspe v večkratnih zaporednih poizkusih se test ovrednoti kot neuspešen. V primeru neuspeha je potrebno napako iskati na relaciji terminalskega algoritma in pravilnosti zapisa CHECK podatkov na kartico.

##### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za obravnavano vsebino CHECK datoteke ne doseže uspeha. To pomeni, dokler terminalski algoritem in algoritem zapisa kontrolnih podatkov CHECK datoteke nista usklajena (pravilno prepoznavanje zapisa/nezapisa).

Test:	BKR027	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

### 10.6.6 TEST DATOTEKE PRODUCTS

Test znotraj PRODUCTS datoteke vključuje preverjanje pravilnosti zapisanega produkta in njegove vsebine na brezstično kartico IJPP, kot je definirano v poglavju o strukturi brezstične kartice.

PRODUCTS podatkovna datoteka vključena v strukturo brezstične IJPP kartice je namenjena hranjenju produktov. V testno proceduro je vključena periferija prodajnega mesta in IJPP kartica. Za branje vsebine PRODUCTS datoteke po naloženem produktu se uporabi prilagojen testni validator, ki prebrane podatke posreduje na serijsko povezavo, kar omogoča detajlno analizo zapisanih podatkov. Kot redundantno orodje se lahko uporabi paket COLLIS.

#### 10.6.6.1 Test zapisanih podatkov v PRODUCTS datoteki

Test se izvaja v celoti, in se ga ne obravnava po posameznih Byte-ih iz razloga, ker se morebitna nepravilnost zapisanih podatkov te datoteke odraža kasneje na višjem nivoju terminalske opreme (prodajnega mesta, validatorja, kontrolorja, kartomata), kjer se preverja ustreznost naloženega produkta.

*Identifikacijska oznaka testa: BKR028*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni skladno s specifikacijo. Za potrebe testa se uporabi še periferija prodajnega mesta. Za potrebe izvedbe tega testa je potrebno upoštevati, da IJPP kartica v začetni fazi nima naloženega nobenega produkta. Če je ta pogoj izpolnjen preberemo vsebino kartice, natančneje vsebino PRODUCTS datoteke, z orodjem COLLIS in pripadajočo programsko opremo. V tem primeru morajo biti podatki PRODUCTS datoteke postavljeni na privzeto vrednost 0xff. Za dodatni kriterij pri izvajanju testa se lahko uporabi še validator oz. kartomat, ki mora v tem primeru ob postavitvi IJPP kartice v NFC polje, javiti neobstoje produkta. Če to drži lahko nadaljujemo proceduro testa. Naslednji korak je izvedba transakcije z nalaganjem produkta (enkratna vozovnica za relacijo A-B). Na kartomatu oz. prodajnem mestu se izbere produkt enkratne vozovnice za relacijo A-B, s prislunitvijo kartice v NFC polje pa se v datoteko 0x08 naloži veljaven produkt (Status = 1 Byte (rezervirano 10 Byte-ov za status), Tariff\_Location = 2 Byte-a, Unique\_Ticket\_ID = 6 Byte-ov, Status\_Potnika = 2 Byte-a, Produkt = 2 Byte-a, Št\_Prehodov = 2 Byte-a, Št\_Kart = 1 Byte, Datum\_Začetka\_Veljave = 4 Byte-i, Datum\_Konca\_Veljave = 4 Byte-i, Datum\_Nakupa = 4 Byte-i, Vrednost\_Ob\_Nakupu = 4 Byte-i, ID\_Tarifnega\_Sistema = 2 Byte-a, Način\_Tarifnega\_Sistema = 1 Byte,

Linija = 4 Byte-i, Vstopna\_Conn = 2 Byte-a, FID\_Index = 1 Byte, Trenutni\_Mesec = 1 Byte, Število\_Kuponov = 1 Byte, ID\_Skupine\_Produktov = 1 Byte, RFU = 10 Byte-ov), s čimer se napolni struktura datoteke v skladu z definirano tabelo podatkov PRODUCTS in sočasno izvede plačilo produkta. S pomočjo orodja COLLIS se ponovno prebere vsebino IJPP kartice, natančneje datoteke PRODUCTS, ki ima sedaj že naložen veljaven produkt. Iz prebrane vsebine se po posameznih Byte-ih zgornje datoteke izlušči uporabna vsebina. Zapisana vsebina produkta se prebere prav tako na testnem validatorju, ki na ekran oz. serijsko povezavo izpiše podatke o produktu. Če je prepoznava produkta s strani validirne enote uspešna, in so vpisani podatki skladni z naloženim produktom, in ni nikakršnega odstopanja v produktni shemi, se test ovrednoti kot uspešen. V nasprotnem primeru, da prepoznava produkta ni pravilna, ali so zapisani podatki napačni se test kartice ovrednoti kot neuspešen, s čimer kartica ne dobi odobritve za integracijo v IJPP sistem.

#### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za obravnavano vsebino PRODUCTS datoteke ne doseže uspeha. To pomeni, dokler terminalski algoritem in algoritem zapisa podatkov produktov PRODUCTS datoteke nista usklajena (pravilno pisanje produktov v definirano podatkovno strukturo PRODUCTS datoteke in pravilno branje).

Test:	<b>BKR028</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.7 TEST DATOTEKE HISTORY VALID**

Test znotraj HISTORY VALID datoteke vključuje preverjanje pravilnosti zapisane validacije produkta in njegove vsebine na brezstično kartico IJPP, kot je definirano v poglavju o strukturi brezstične kartice. HISTORY VALID je podatkovna datoteka vključena v strukturo brezstične IJPP kartice, namenjena hranjenju zgodovine validacije produktov. V poglavju 3 je definirana in rezervirana velikost prostora za zapis podatkov zgodovine validacije produkta v datoteko HISTORY-VALID. V testno proceduro je vključena periferija validatorja, IJPP kartica z naloženim produktom (predhodni test), orodje COLLIS in validator-kontrolor za kontrolo validacij. Za branje vsebine HISTORY-VALID datoteke po naloženem in validiranem produktu se uporabi validator-kontrolor, ki prebrane podatke zadnje validacije (zgodovine) posreduje na serijsko povezavo oziroma jih izpiše direktno na zaslon.

##### **10.6.7.1 Test zapisanih podatkov v HISTORY-VALID datoteki**

Test se izvaja v celoti, in se ga ne obravnava po posameznih Byte-ih iz razloga, ker se morebitna nepravilnost zapisanih podatkov te datoteke odraža kasneje na višjem nivoju terminalske opreme

(prodajnega mesta, validatorja, kontrolorja), kjer se preverja ustreznost zapisane vsebine opravljene validacije (kontrolor).

#### *Identifikacijska oznaka testa: BKR029*

##### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni skladno s specifikacijo. Za potrebe izvedbe tega testa je potrebno upoštevati, da mora imeti IJPP kartica naložen aktiven produkt enkratne vozovnice za relacijo A-B. Pred nadaljevanjem testa preberemo vsebino kartice, natančneje vsebino HISTORY-VALID datoteke, z orodjem COLLIS in pripadajočo programsko opremo. Ker še produkt na kartici in kartica sama nista udeležena v nobeni proceduri validacije mora biti vsebina HISTORY-VALID datoteke enaka 0xff glede na specificirane podatke. Če je ta pogoj izpolnjen lahko test nadaljujemo s prislunitvijo kartice v NFC polje testnega validatorja, kateri mora upravljati z validacijami na relaciji A-B. Hkrati z izvedeno validacijo mora validator v HISTORY-VALID datoteko zapisati podatke validacije, in sicer: (ID validatorja (TID) = 4 Byte-i, čas validacije = 4 Byte-i, postaja = 4 Byte-i, linijo = 4 Byte-i, indeks produkta = 1 Byte, Unique\_Ticket\_ID = 6 Byte-ov, številka\_avtobusa = 2 Byte-a in RFU = 7 Byte-ov. Vsi naštetih podatki so znani (TID, čas validacije, postaja, linija, index produkta, , unique ticket id, št. avtobusa) in služijo neposredni primerjavi. Vsako validacijo zabeleži validirna enota, in jo posreduje v zaledni sistem, tako, da je na višjem nivoju (test sistema kot celote) možno primerjati zapisane podatke validacije na kartico s podatki zalednega sistema. Ker so ti testi namenjeni preizkušanju posameznih komponent (v tem primeru brezstične kartice), mora validirna enota ob izvedeni validaciji na serijsko povezavo vrniti podatke (TID, čas validacije, postaja, linija, index produkta, unique ticket id, št. avtobusa) , ki bi jih naj zapisala na IJPP kartico v datoteko HISTORY-VALID. Podatki iz serijske povezave v tem testu predstavljajo osnovo primerjalnega testa, v katerem se le ti primerjajo s podatki zapisanimi na kartici. Da lahko izvedemo neposredno primerjavo, se v testu uporabi mobilni validator (kontrolor), ki omogoča izpis podatkov o opravljeni predhodni validaciji. V tem primeru je potrebno IJPP kartico z predhodno validiranim produktom enkratne vozovnice postaviti v NFC polje kontrolorja (mobilni validator), ki mora prebrati podatke iz HISTORY-VALID datoteke, in jih prikazati na zaslon, ali prav tako posredovati na serijsko povezavo. Če so podatki validacije identični prebranim podatkom iz HISTORY-VALID datoteke se test vrednoti kot uspešen. V primeru neuspeha, neujemajočih podatkov se mora iz strani ponudnika testirane opreme ugotoviti, kateri segment ne deluje pravilno (validator, algoritem zapisa podatkov validatorja, mobilni validator-kontrolor, algoritem branja podatkov mobilnega validatorja itd.). Dokler sklop uporabljenih elementov (naprav) v testu (validator, mobilni validator-kontrolor in IJPP KARTICA) ne opravijo testa tudi kot celota ne dobijo soglasja za integracijo v IJPP sistem.

##### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za obravnavano vsebino HISTORY-VALID datoteke ne doseže uspeha. To pomeni, dokler pisanje in branje vsebine na/iz HISTORY-VALID ni usklajeno (zapisana vsebina v HISTORY-VALID datoteko iz strani validirne enote, mora biti enaka prebrani vsebini iz HISTORY-VALID datoteke iz strani mobilne validirne enote-kontrolorja).

Test:	<b>BKR029</b>
-------	---------------

Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

### 10.6.8 TEST DATOTEKE RELATIONS

Test znotraj RELATIONS datoteke vključuje preverjanje pravilnosti zapisane tarife linije na brezstično kartico IJPP, kot je definirano v poglavju 3 o strukturi brezstične kartice. RELATIONS podatkovna datoteka vključena v strukturo brezstične IJPP kartice je namenjena hranjenju podatkov tarife, ki se zabeleži ob nakupu produkta. V testno proceduro je vključena periferija prodajnega mesta (kartomat), IJPP kartica in validator z zmožnostjo pošiljanja prebranih podatkov datoteke RELATIONS na serijsko povezavo.

#### 10.6.8.1 Test zapisanih podatkov v RELATIONS datoteki

Test se izvaja v celoti, in se ga ne obravnava po posameznih Byte-ih iz razloga, ker se morebitna nepravilnost zapisanih podatkov te datoteke odraža kasneje na višjem nivoju terminalske opreme (prodajnega mesta, validatorja, kontrolorja), kjer se preverja ustreznost zapisane vsebine tarife.

*Identifikacijska oznaka testa: BKR030*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje brezstična IJPP kartica z naloženimi diverzificiranimi ključi in definirano podatkovno strukturo, katere podatki morajo biti postavljeni skladno s specifikacijo. Za potrebe izvedbe tega testa je potrebno upoštevati, da mora imeti IJPP kartica naložen aktiven produkt enkratne vozovnice za relacijo A-B (nadaljevanje predhodnih testov). Ker je produkt že naložen na IJPP kartici, se je skladno z nakupom produkta v RELATIONS zapisala tudi tarifa pod katerimi pogoji je bil produkt kupljen. Ob prodaji produkta je na prodajnem mestu (kartomatu) zabeležena transakcija in tarifa s katero je terminal kartomata takrat razpolagal. Ker poznamo vsebino tarife (Status = 5 Byte-ov, Format\_Identifier = 1 Byte, Line\_Definition = 1 Byte, Line\_ID = 2 Byte-a, Valid\_From = 4 Byte-i, Valid\_To = 4 Byte-i, Distance = 2 Byte-a, Start\_Station\_Name = 10 Byte-ov, Stop\_Station\_Name = 10 Byte-ov, Bus\_Station\_Id1... Bus\_Station\_Id20 -> vsak ID = 4 Byte, ONE\_WAY = 1 Byte, Subsidized = 1 Byte, IJPP\_CardRelation\_ID = 4 Byte in RFU = 6 Byte-ov), ob času naložitve produkta na IJPP kartico, se te podatke shrani in uporabi za neposredno primerjavo s podatki, ki so zapisani na kartici. Za dostop do zapisanih podatkov datoteke RELATIONS na IJPP kartici se uporabi validator za aktivno linijo A-B aktivnega produkta. S postavitvijo kartice v polje NFC validatorja, se produkt validira, s tem pa tudi prebere vsebina RELATIONS datoteke, katere vsebino mora testni validator posredovati na serijsko povezavo. Iz posredovanih podatkov se izlušči zapisana vsebina RELATIONS podatkov skladno z definirano strukturo v poglavju 3, in se primerja le te s poznano vsebino tarife ob nakupu produkta. Če se vsebina ujema, in v primerjani vsebini ne prihaja do odstopanj, se test ovrednoti kot uspešen, s čimer se tudi potrdi pravilno zapisovanje v RELATIONS datoteko in pravilno branje iz iste datoteke. V nasprotnem primeru, da primerjana vsebina ni identična je od tehnološkega ponudnika odvisna

lokalizacija problema. V kolikor test ni uspešen celoten sklop testirane opreme ne dobi soglasja za integracijo v IJPP sistem.

#### *Kdaj izvesti test*

Specificiran test kartice je potrebno izvajati dokler se za obravnavano vsebino RELATIONS datoteke ne doseže uspeha. To pomeni, dokler pisanje in branje vsebine na/iz RELATIONS ni usklajeno (zapisana vsebina v RELATIONS datoteko iz strani kartomata (nakup), mora biti enaka prebrani vsebini iz RELATIONS datoteke iz strani validirne enote).

Test:	<b>BKR030</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

### **10.6.9 TEST VOZOVNICE**

Test vozovnice zajema tri testne scenarije, ki se tičejo uspešnega nakupa, neuspešnega nakupa in nakupa s kartico, ki se nahaja na črni listi. Vsak testni scenarij je v nadaljevanju obravnavan ločeno, in se ga izvaja ločeno.

#### **10.6.9.1 Test uspešne prodaje vozovnice za relacijo A-B**

Pri tem testu izvedemo scenarij prodaje oz. nakupa enkratne vozovnice za relacijo A-B. V testu preizkusimo nakup in shranjevanje enkratne vozovnice na kartico.

*Identifikacijska oznaka testa: VOZ001*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje kartica z že prednastavljenimi podatki o imetniku, zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice. Pred izvršbo testa se vsebina zapisa na kartici preveri z orodjem COLLIS, s čimer se pridobi vsebina 'batch' kartice pred nakupom. V testnem scenariju se uporabi še kartomat za prodajo vozovnic. Na kartomatu se za testni namen izbere v naprej pripravljeno testno vozovnico za relacijo A-B. Skladno z navodilom kartomata se v testu na označeno mesto prisloni brezstična kartica s čimer se izvrši nakup. Glede na poznano podatkovno strukturo brezstične kartice IJPP se s programskim paketom COLLIS preveri zapisani podatek na za to definirano lokacijo (glej poglavje 3 o specifikirani strukturi kartice). V kolikor se z orodjem COLLIS ugotovi, da je vozovnica z vsemi podatki na kartici uspešno zapisana in je pred tem kartomat uporabniku sporočil uspešen nakup, in se je transakcija zaključila v predpisanem časovnem okvirju in brez napak, se test smatra kot uspešen. V kolikor kateri izmed naštetih kriterijev ni izpolnjen (zapis na napačno mesto, nepravilen zapis, ipd.) se test ovrednoti kot neuspešen.



Ker obstaja v sistemu IJPP več različnih ponudnikov kartomato ipd. je test potrebno izvesti na vsaj eni napravi (kartomatu) vsakega ponudnika, s čimer se hkrati preverja unificirano delovanje kartomato različnih ponudnikov.

#### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske opreme kartomata testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico. Test je obvezen in je predpogoj za izvajanje naslednjih kartičnih testov. V kolikor je test neuspešen, preostalih dveh testov glede nakupa enkratne vozovnice ni smiselno izvajati.

Test:	<b>VOZ001</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.9.2 Test neuspešne prodaje vozovnice za relacijo A-B**

Pri tem testu izvedemo scenarij prodaje oz. nakupa enkratne vozovnice za relacijo A-B s premajhnim vplačilom. V testu preizkusimo zavrnitev nakupa enkratne vozovnice in hkrati beleženje nespremenjenih podatkov na kartici. Podatki pred neuspešnim nakupom morajo ostati identični podatkom po neuspešnem nakupu.

*Identifikacijska oznaka testa: VOZ002*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje kartica z že prednastavljenimi podatki o imetniku, zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice. V testnem scenariju se uporabi še kartomat za prodajo vozovnic. Pred izvršbo testa se vsebina zapisa na kartici preveri z orodjem COLLIS, s čimer se pridobi vsebina 'batch' kartice pred nakupom. Na kartomatu se v testni namen izbere vnaprej pripravljeno testno vozovnico za relacijo A-B. Skladno z navodilom kartomata se v testu na označeno mesto prisloni brezstična kartica in nezadostno plačilo, s čimer se prične postopek nakupa. Ker vplačilo ne dosega vrednosti, ki je potrebna za nakup enkratne vozovnice, se transakcija prekine, kartomat pa mora uporabniku ali s signalnim (svetlobnim, zvočnim – glej poglavje o poenoteni signalizaciji) zapisom sporočiti, da nakup ni uspel. V primeru zaslona mora kartomat uporabniku izpisati sporočilo »Nezadostno vplačilo. Nakup neuspešen«. Glede na poznano podatkovno strukturo brezstične kartice IJPP se s programskim paketom COLLIS preveri za morebitnimi spremembami podatkov na kartici. Ker nakup v takšnem testnem primeru ni uspešen, se na kartici podatki o naloženih produktih ne smejo spremeniti. V kolikor se z orodjem COLLIS ugotovi, da so podatki na kartici pri neuspešnem nakupu ostali nespremenjeni, se test smatra kot uspešen. V kolikor pa je pri takšnem testu prišlo do spremembe podatkov (vozovnic), se test ovrednoti kot neuspešen.

Ker obstaja v sistemu IJPP več različnih ponudnikov kartomato, ipd. je test potrebno izvesti na vsaj eni napravi (kartomatu) vsakega ponudnika, s čimer se hkrati preverja unificirano delovanje kartomato različnih ponudnikov.

#### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske opreme kartomata testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico. Test je obvezen in je predpogoj za izvajanje preostalih kartičnih testov. V kolikor je test neuspešen se preostali test lahko opravi, vendar naprava ne dobi oddobritve za integracijo v IJPP. Test je obvezen in mora biti uspešen.

Test:	VOZ002	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.9.3 Test prodaje vozovnice za relacijo A-B s kartico, ki je na črni listi**

Test je namenjen preverjanju, ali so mehanizmi za upravljanje s karticami na črni listi ustrezno konfigurirani in opravljajo svojo nalogo. Rezultat testa mora odražati preprečevanje akcij s kartico iz črne liste.

*Identifikacijska oznaka testa: VOZ003*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje kartica z že prednastavljenimi podatki o imetniku, zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice. Serijska številka kartice mora biti pred testom vstavljena v skupno bazo črnih list, prav tako pa mora biti kartomat posodobljen s svežnjem kartic na črni listi. Pred izvršbo testa se vsebina zapisa na kartici preveri z orodjem COLISS, s čimer se pridobi vsebina 'batch' kartice pred nakupom. V testnem scenariju se uporabi še kartomat za prodajo vozovnic. Na kartomatu se v testni namen izbere vnaprej pripravljeno testno vozovnico za relacijo A-B. Skladno z navodilom kartomata se v testu na označeno mesto prisloni brezstična kartica, ki je na črni listi, s čimer se prične postopek nakupa. Ker kartomat že v prvem koraku preverja serijsko številko kartice s serijskimi številkami kartic iz črne liste s katerimi razpolaga, mora kartico nemudoma zavrniti brez nadaljnjih procedur. Pri tem mora kartomat na lokacijo za aktivnost, neaktivnost, zaklenitev zapisati informacijo o zaklenitvi kartice. Kartomat mora uporabniku ali s signalnim (svetlobnim, zvočnim – glej poglavje o poenoteni signalizaciji 10.5.1) zapisom sporočiti, da nakup ni uspel. V primeru zaslona mora kartomat uporabniku izpisati sporočilo »Blokirana kartica. Nakup neuspešen«. Glede na poznano podatkovno strukturo brezstične kartice IJPP pred postopkom zapisa blokade se s programskim paketom COLLIS ponovno preveri spremembe

podatkov na kartici. Primerjava podatkov kartice mora v tem primeru odražati samo spremenjen podatek o blokadi (sprememba iz aktivnega stanja v stanje blokade). Vsi ostali podatki morajo ostati nespremenjeni saj nakup v takšnem primeru ni uspešen. V kolikor se z orodjem COLLIS ugotovi, da so podatki na kartici, ki je na črni listi ustrezno korigirani samo v segmentu statusa aktivnosti kartice (prehod iz aktivne kartice v stanje blokirane kartice) se test vrednoti kot uspešen. V kolikor je pri takšnem testu prišlo do napačne spremembe podatkov (vozovnic), ali se celo dovoli nakup, kljub temu da je kartica na črni listi, se test ovrednoti kot neuspešen.

Ker obstaja v sistemu IJPP več različnih ponudnikov kartomatov, ipd. je test potrebno izvesti na vsaj eni napravi (kartomatu) vsakega ponudnika, s čimer se hkrati preverja unificirano delovanje kartomatov različnih ponudnikov.

#### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske opreme kartomata testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, oziroma spremembe logike in algoritmov za upravljanje s črnimi listami. Test je obvezen in je predpogoj za izvajanje preostalih kartičnih testov. V kolikor je test neuspešen naprava ne dobi odobritve za integracijo v IJPP. Test je obvezen in mora biti uspešen.

Test:	<b>VOZ003</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.10 TEST VALIDATORJA**

Ker gre za unificiran sistem integriranega javnega potniškega prometa, v katerem sodelujejo različni ponudniki programske in strojne opreme, mora biti delovanje iste naprave (validatorja) različnih ponudnikov identično. Identično pomeni, da pravila, ki so postavljena glede zaklepa validatorja, obratovanja v 'offline' načinu, pravila za pošiljanje batchev, blokade validatorja ipd. za vse ponudnike enaka. Takšna hipoteza se potrjuje z naslednjim naborom testnih scenarijev, ki se tičejo validacijske enote (validator).

##### **10.6.10.1 Test ujemanja diverzificiranih ključev na kartici s ključi validirne enote**

Gre za enega izmed inicialnih testov v katerem se preveri ujemanje ključev, oziroma preverjanje delovanja algoritma na validatorju, ki preračunava prebran diverzificiran ključ iz brezstične kartice v obliko, kot je definirana za validirno enoto.

*Identifikacijska oznaka testa: VAL001*

### Scenarij testa

V prvem koraku se na kartico s pomočjo kodirne enote zapiše tako imenovani diverzificiran ključ. Za test ni potrebna posebna priprava kartice, temveč klasična, ki se uporabi za pripravo uradne kartice (vsi podatki, vključno s produkti in diverzificiranim ključem). V naslednjem koraku postavimo kartico v polje NFC čitalnika validirne enote in jo v polju pustimo zadosten čas, da se validirna naprava odzove. Validator v tem postopku prebere diverzificiran ključ, in ga z integriranim in specificiranim algoritmom pretvori (preračuna v klasično obliko ključa, ki je identična obliki ključa shranjenem na validirni enoti. Validator mora v primeru neujemanja ključev signalizirati avtentifikacijsko napako (validacija ni mogoča). Signalizacija mora biti za vse ponudnike validirnih enot identična v odvisnosti od uporabljenega uporabniškega vmesnika in specificiranih lastnosti v predhodnem poglavju. Neuspeh avtentifikacije odraža nepravilno delovanje algoritma za preračun ključa iz diverzificirane oblike v klasično obliko. V primeru izvedbe validacije na validirni enoti pomeni, da je algoritem za preračunavanje diverzificirane oblike ključa v klasično obliko ustrezen, s čimer je tudi test potrjen.

### Kdaj izvesti test?

Glede same narave testa, se bo slednji vede oz. nevede izvajal ob vsaki postavitvi kartice v NFC polje validirne enote. Branje diverzificiranega ključa in pretvorba v ustrezno obliko, je eden izmed prvih korakov oz. pogojev za izvedbo transakcije. Kot test je naveden ločeno, čeprav je sestavni del vsake validacije, in zato nima posebej definiranega pogoja, kdaj in v katerih okoliščinah ga je potrebno izvesti. V primeru neuspešnega testa se testni scenariji, ki se tičejo branja in pisanja posameznih podatkov na kartico s strani validirne enote prekinajo.

Test:	VAL001	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

### 10.6.10.2 Test preverjanja seje v proceduri pisanja in spremembe podatkov na kartici s strani validirne enote

V tem testu preverjamo pravilno delovanje algoritma preverjanja stanja seje pisanja podatkov oz. spreminjanja podatkov na brezstični kartici (ang. t.i. 'session check').

Identifikacijska oznaka testa: VAL002

### Scenarij testa

V prvem koraku se v testnem laboratoriju pripravi identična brezstična kartica, kot bo uporabljena v sistemu IJPP. Testni laboratorij pa mora za te potrebe testa pripraviti dva testna scenarija:

- Scenarij z brezstično kartico v kateri se blokira tako imenovana povratna komanda (informacija) o uspešnosti spremembe podatkov oz. zapisa podatkov na kartico. To pomeni da

sporočilo kartice, da je bila uspešno zapisana, oz. vsebina modificirana, ne doseže čitalnika validirne enote. Tak testni scenarij posnema obnašanje uporabnikov, ki prehitro umaknejo kartico iz polja. Na testu bo kartica dovolj dolgo v polju, vendar se bo pri validaciji s tem tipom kartice emuliralo nepošiljanje OK komande. Pri tem mora biti tako na validatorju kot kartici poskrbljeno preračunano naključno število (ang. random number). V tem testu, kljub temu, da validirna enota ne prejme OK komande, izvede transakcijo v ozadju, pri čemer mora zaslon oz. indikacija s svetlobnimi signali ostati identična (nespremenjena). Ker simulirani uporabnik ne zazna spremembe, je prisiljen kartico ponovno prisloniti. Zato se v drugem koraku tega testa najprej preveri izračunano naključno število. Če je slednje enako se uporabniku ne odšteje produkt ipd., temveč se ga spusti dalje in to ustrezno signalizira na uporabniškem vmesniku. V kolikor pa je uporabnik prehitro umaknil kartico, da sprememba zapisa na kartici ni uspela se s tem ne ujema random število, kar pomeni, da bo validator v naslednjem uspešnem poizkusu validiral produkt. V tem drugem primeru mora validator javiti oz. signalizirati neuspešno validacijo. Na enak način morajo delovati vsi validatorji različnih ponudnikov brez izjeme.

- V drugem testu se izvede procedura pisanja na kartico s strani validirne enote, kjer je uporabljena kartica brez blokirane povratne informacije o uspešnosti zapisa oz. spremembe podatkov na brezstični kartici. V testu gre enako kot pri predhodno opisanem za validacijo, oz. koriščenje specifičnega produkta. V tem testu se kartico postavi v NFC polje validirne naprave in se jo namerno pusti dovolj časa v polju, da kartica vrne OK proceduro pisanja, ter da se izvede 'transakcija' katere potek se sočasno odrazi na uporabniškem vmesniku validirne enote.

#### *Kdaj izvesti test?*

Test je potrebno izvajati dokler naprava ne prestane takšnega testa. Ko validirna enota uspešno prestane takšen testni scenarij, ga v naslednjih izpeljankah nadgrajene programske opreme ni potrebno izvajati, razen, če je prišlo do sprememb logike in algoritmov v segmentu, ki nadzira t.i. 'session check'.

Test:	VAL002	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.10.3 Test odziva validatorja v primeru manjkajočega podatka na brezstični kartici**

V tem testu obravnavamo scenarij, kadar se na validirnem mestu pojavi brezstična kartica, kateri manjka ključni podatek (ime in priimek). Takšen test obravnava scenarij kadar je pri distribuciji kartice in vnosu podatkov prišlo do napake referenta, ki je pozabil vnesti ključen podatek. Preverjanje s takšnim testnim scenarijem zmanjša možnost nepričakovanih napak in nepredvidljivega obnašanja opreme različnih ponudnikov v sistemu IJPP.

Identifikacijska oznaka testa: VAL003

#### Scenarij testa

V prvem koraku se za testne namene ustvari brezstična kartica s podatkovno strukturo, kot jo definira standard tega dokumenta (poglavje 3). Pri kreaciji kartice se vnesejo vsi podatki skladno s specifikacijo, razen imena in priimka. Ko je kartica pripravljena se test nadaljuje s postavitvijo le te v magnetno polje kratkega dosega, ki ga ustvarja čitalnik NFC integriran v validirno enoto. Ker kartica ne vsebuje ključnega podatka, mora validirna enota kartico zavrniti s tekstovnim sporočilom »Neveljavna kartica« v primeru zaslona, oziroma signalnim svetlobnim ali zvočnim opozorilom (glej teste poenotene signalizacije). Test je opisan na primeru manjkajočega podatka lastnika in velja tudi za vse ostale podatke, ki so v specifikaciji definirani kot ključni. V vsakem primeru manjka ključnega podatka se mora validirna enota odzvati na enak način (zavrnitev s tekstovnim, signalnim ali zvočnim opozorilom).

#### Kdaj izvesti test?

Test je potrebno izvajati dokler naprava ne prestane takšnega testa. Ko validirna enota uspešno prestane takšen testni scenarij, ga v naslednjih izpeljankah nadgrajene programske opreme ni potrebno izvajati, razen, če je prišlo do sprememb logike in algoritmov v segmentu obdelave prebranih podatkov iz kartice na validirni enoti.

Test:	VAL003	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### 10.6.10.4 Test zaklenitve validatorja

Validator je neposredno povezan s centrom preko GPRS omrežja. To pomeni, da deluje v tako imenovanem 'online' režimu. Le ta mu omogoča, da v skupno bazo posreduje podatke o vsaki posamezni validaciji. Zaradi geografske razgibanosti terena Republike Slovenije obstajajo področja, ki niso pokrita z GPRS signalom. V takšnem primeru, ko validator izgubi povezavo do GPRS omrežja preide v tako imenovani 'offline' režim obratovanja. Odvisno od končnega dogovora lahko v 'offline' režimu delovanja ostane zgolj dogovorjenih 'x' ur. V tem času omogoča nemoteno koriščenje produktov, in kreira tako imenovane 'batche', ki jih upravlja varnostni element. Če validator v predpisanem obdobju 'x' ur offline režima ne uspe s povratkom in registracijo na GPRS omrežje, se mora slednji zakleniti.

Identifikacijska oznaka testa: VAL004

#### Scenarij testa

Za potrebe testa se validatorju definira dovoljeno obdobje, ki ga lahko preživi v 'offline' režimu. Za nastavitve in podprto funkcionalnost poskrbi vsak ponudnik produkta sam. Za verodostojnost izvedenega testa se pri prvem testu uporabi 'x-minutno' obdobje, v drugem 'x-urno' obdobje in v tretjem primeru 'x-dnevno' obdobje. Offline obdobje bo definirano kasneje. Ko je validator primerno

konfiguriran, se mu izključi povezavo na GPRS omrežje (izključi se GPRS modem, izvzame se SIM kartica ipd.). Sočasno z izključitvijo iz GPRS omrežja se vzporedno meri čas do trenutka, ko se validator zaklene. Izmerjen čas se primerja s prednastavljeno periodo 'offline' režima. V kolikor se validator po preteku definiranega 'offline' obdobja ustrezno zaklene je test uspešen. V obratnem primeru validator ne prestane testa in se mu ne odobri integracije v IJPP sistem. Ves čas testa je povezava na GPRS omrežje izključena.

#### *Kdaj izvesti test?*

Izvedba testa je predvidena v primerih, kadar ponudnik terminalske opreme na kakršen koli način posega v ta segment, spreminja algoritme, ki upravljajo ta segment, oz. kadar zamenja posamezne komponente RTC modula. Ko programska oprema definirane verzije prestane takšen test, ga v naslednjih verzijah ni potrebno izvajati, če nič od naštetega v prvem stavku ne drži. V kolikor obstaja kakršnakoli sprememba v segmentih navedenih v prvem stavku, je test potrebno izvesti.

Test:	<b>VAL004</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.10.5 Test zavrnitve koriščenja produktov v režimu popolne zaklenitve validatorja**

Ko terminal preseže dovoljeno obdobje 'x' ki mu omogoča 'offline' delovanje preide v režim popolne zaklenitve. V takšnem primeru je koriščenje produktov onemogočeno. To pravilo velja za vse ponudnike validacijske opreme.

*Identifikacijska oznaka testa: VAL005*

#### *Scenarij testa*

Test se navezuje na predhodnega. Od trenutka, ko je validirna enota blokirana zaradi preseženega dovoljenega časovnega okvirja v offline načinu, ne sme sprejeti več nobene validacije. Ko tester približa brezstično kartico k validirni enoti, slednja prepreči validacijo. Koriščenje produktov iz brezstične kartice v tem režimu validirne enote ni dovoljeno. Pri testu se mora v prvem pogoju validirna enota nahajati v blokiranem načinu (preseže se časovni okvir, ki dovoljuje delovanje validirne naprave v offline načinu), ob tem pa se mora brezstična kartica nahajati v magnetnem polju NFC modula validirne naprave. Če validator koriščenje produkta zavrne je test uspešen, v nasprotnem primeru naprava ne izpolnjuje pogojev za integracijo v IJPP.

#### *Kdaj izvesti test?*

Izvedba testa je predvidena v primerih, kadar ponudnik produkta na kakršen koli način posega v ta segment, spreminja algoritme, ki upravljajo ta segment, oz. kadar zamenja posamezne komponente

RTC modula. Ko programska oprema definirane verzije preneha takšen test, ga v naslednjih verzijah ni potrebno izvajati, če nič od naštetega v prvem stavku ne drži. V kolikor obstaja kakršnakoli sprememba v segmentih navedenih v prvem stavku, je test potrebno izvesti.

Test:	<b>VAL005</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### 10.6.10.6 Test koriščenja specifičnih produktov v offline režimu delovanja validatorja

Kadar terminal izgubi aktivno povezavo do GPRS omrežja, preide v tako imenovan 'offline' način delovanja v katerem je dovoljeno koriščenje specifičnih produktov (glej poglavje definiranih produktov in pravil uporabe). Funkcionalnost mora biti brezhibna, saj omogoča nemoteno uporabo sistema IJPP in koriščenja javnega prevoza.

*Identifikacijska oznaka testa: VAL006*

##### *Scenarij testa*

V testu poskrbimo, da validacijska enota preide v režim offline delovanja. Za takšen režim je potrebno prekiniti GPRS povezavo. Odvisno od ponudnika validirne enote se slednjo prekine s pomočjo izklopa GPRS modema, odstavitve SIM kartice ipd. V kolikor na podoben način GPRS povezave ni mogoče prekiniti, mora ponudnik validirne enote s pomočjo programske posodobitve omogočiti izvedbo testa. V drugem koraku testni laboratorij pripravi NFC Missfire-Desfire 8K testno kartico z vsemi produkti, izmed katerih je zgolj specifičnim produktom dovoljeno koriščenje v offline režimu (glej poglavje o specifičnih produktih sistema IJPP). Ko validacijska enota preide v offline režim delovanja, se testno NFC kartico postavi v neposredno bližino validirne naprave (v neposredno bližino NFC čitalnika) in znotraj dosega magnetnega polja kratkega dosega validirne naprave. V testu offline obratovanja se preizkusijo vsi produkti. Če produkte iz seznama, ki jim ni dovoljeno koriščenje v offline režimu, validirna enota zavrne, in tiste iz seznama, ki jim je koriščenje v offline režimu dovoljeno sprejme, in je test 100 odstotno uspešen, in njegova večkratna ponovljivost prav tako, potem se test smatra kot uspešen. V kolikor validirna enota sprejme produkt, ki ga v offline režimu ne bi smela (glej seznam produktov), oziroma sprejme takšnega ki ga ne bi smela (pa čeprav samo enega) je test neuspešen. V primeru neuspešnega testa, lahko naprava opravi še preostale teste na seznamu v tem dokumentu, vendar ne dobi soglasja za integracijo v IJPP sistem (pogoji za integracijo niso izpolnjeni).

##### *Kdaj izvesti test?*

Izvedba testa je predvidena v primeru prve integracije validacijske enote v sistem IJPP. Test je potrebno izvesti tudi, ob spremembah politike pravil koriščenja produktov v offline načinu delovanja validirne enote. Test je smiselno izvesti tudi v primerih, ko se dodajajo novi produkti. V kolikor je validirna enota



test uspešno prestala, in v segmentu politike koriščenja produktov ni bilo nobenih sprememb, testa ni potrebno ponovno izvajati.

**Opomba:** Shema koriščenja dovoljenih in nedovoljenih produktov v offline režimu bo dodana naknadno, ko bodo znani vsi produkti in znana politika koriščenja!

Test:	VAL006	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### 10.6.10.7 Test pošiljanja in upravljanja batchev pri offline koriščenju produktov

Kadar terminal izgubi aktivno povezavo do GPRS omrežja, preide v tako imenovan 'offline' način delovanja v katerem je dovoljeno koriščenje specifičnih produktov (glej poglavje definiranih produktov in pravil uporabe). Vsak uporabljen produkt v offline načinu se mora zabeležiti za potrebe kasnejših bilanc in evidenc. Drugi pomemben vidik beleženja pa je preprečevanje zlorab. Za beleženje izvršenih transakcij (validacij) v offline načinu specifikacija predvideva uporabo varnostnega elementa. Slednji skrbi za beleženje izvedenih offline validacij, ki jih v obliki batcha hrani v kriptirani obliki. Z uporabo kriptografije so podatki na varnostnem elementu zaščiteni. Funkcionalnost mora biti brezhibna, saj omogoča nemoteno uporabo sistema IJPP in koriščenja javnega prevoza.

*Identifikacijska oznaka testa: VAL007*

##### Scenarij testa 1

Batch je lahko uporaben na dva načina. V prvem načinu (ta tip testa) se batch mora avtomatično poslati k strežnemu sistemu, ko se validirna enota ponovno prijavi na GPRS omrežje po GPRS podatkovnem kanalu. Na ta način se dopolni evidenca o offline izvedenih validacijah. Potek testa je definiran na sledeč način: Validirna enota se mora nahajati v offline režimu. Za potrebe testa terminal ne sme vsebovati kriptiranih batchev iz morebitnih predhodnih scenarijev. Na testni NFC brezstični kartici morajo biti tri pred definirani testni produkti, ki jim je dovoljeno koriščenje v offline načinu delovanja validirne enote. V testu morajo biti vsi tri pred definirani aktivni produkti uporabljeni. Koristijo se eden za drugim na način postavitve NFC kartice v doseg NFC magnetnega polja validirne enote. Ker so vhodni podatki znani (pred naloženi produkti), je poznan tudi izhod (batch- uporabljeni produkti). V naslednjem koraku testa se validatorju omogoči vzpostavitev povezave z GPRS omrežjem (vključitev modema, vstavitve SIM kartice, programska omogočitev ipd. – odvisno od ponudnika validirne enote). V trenutku vzpostavitve povezave z GPRS omrežjem in strežnim sistemom validirna enota v testu pošlje batch na strežni sistem. Ker so uporabljeni podatki znani, se vsebina batcha primerja z vhodnimi podatki. Primerjava je analitična. V analitični primerjavi se primerja datum, čas, ura, produkti, id validirne naprave, id kartice s katero je bila validacija izvedena in id produkta. Če vsi podatki ustrezajo, in batch zajema vse uporabljene produkte, se test smatra kot uspešen, in iz tega vidika validirna enota

izpolnjuje pogoje za integracijo v IJPP sistem. V kolikor validirna enota batcha ob aktivni GPRS povezavi v predpisanem časovnem okviru (glej aplikacijski protokol) ne posreduje na strežni sistem je test iz tega vidika neuspešen.

#### Scenarij testa 2

Drugi način uporabe batcha je fizičen prenos podatkov iz terminalske opreme. Ta testni scenarij se izvede na sledeč način: validirno enoto postavimo v offline način obratovanja, pri čemer dovoljeno časovno periodo 'x' ki dovoljuje offline operativni režim postavimo na minimalno raven petih minut. Znotraj teh petih minut pokoristimo tri preddefinirane produkte na enak način kot pri scenariju testa 1. Ko so produkti pokoriščeni so vhodni podatki znani. Pri testu počakamo na pretek petih minut, da se validirna enota zaklene (uporaba produktov ni več mogoča). Pri tem testu, ne vzpostavljamo povezave z GPRS omrežjem, saj se simulira zaklenitev validirne naprave in kasnejši fizičen prenos podatkov iz 'batcha' iz terminalske opreme. Ko je validirna naprava zaklenjena se iz nje prenese kriptirana batch datoteka, ki vsebuje opravljene validacije. Kriptirani podatki se nato ročno prenesejo (importirajo) in vstavijo v strežni sistem, na katerem se dekriptirajo in zabeležijo v sistem. Preverjanje značilk batcha je identično kot pri predhodnem testu (scenarij tipa 1). Primerjava je analitična. V analitični primerjavi se primerja datum, čas, ura, produkti, id validirne naprave, id kartice s katero je bila validacija izvedena in id produkta. Če vsi podatki ustrezajo, in batch zajema vse uporabljene produkte, se test smatra kot uspešen, in iz tega vidika validirna enota izpolnjuje pogoje v primeru popolne zaklenitve glede hranjenja opravljenih validacij v batchu za integracijo v IJPP sistem.

Pogoj za uspešno opravljen test sta uspešno opravljena oba zgoraj definirana podtesta. Če kateri izmed njiju spodleti, kriterij za integracijo v IJPP sistem ni izpolnjen, saj v takšnem primeru obstaja možnost manipulacije in mahinacije v finančnem toku. V kolikor test ni uspešen se naslednji testi lahko kljub temu izvedejo vendar enota ne dobi odobritve za integracijo v IJPP sistem.

#### Kdaj izvesti test?

Izvedba testa je predvidena v primeru prve integracije validacijske enote v sistem IJPP.

**Opomba:** Shema koriščenja treh testnih produktov za potrebe tega testa bo dodana naknadno, ko bodo znani produkti.

Test:	VAL007	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### 10.6.10.8 Test uspešne validacije vozovnice

S tem testom se ugotavlja ustreznost procedure, ki skrbi za koriščenje enkratne vozovnice v sistemu IJPP. Preverja se, ali se na validirni enoti vozovnica uspešno validira, in ali so podatki validacije ustrezno zapisani na samo kartico, ter ali je interakcija (signalizacija, sporočila) validirne enote ustrezna.

*Identifikacijska oznaka testa: VAL008*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje kartica z že prednastavljenimi podatki o imetniku, zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice. Na kartici skladno s specifikacijami le te (glej poglavje specifikacije IJPP kartice) mora biti shranjena tudi datumsko veljavna enkratna vozovnica za relacijo A-B (glej test nakupa enkratne vozovnice – poglavje 10.6.9.1). Pred izvršbo testa se vsebina zapisa na kartici preveri z orodjem COLISS, s čimer se pridobi vsebina 'batch' kartice pred validacijo. V testnem scenariju se uporabi še validator za beleženje uporabe vozovnic. Za pričetek testnega scenarija, se kartico s produktom enkratne vozovnice prisloni na označeno mesto validirne enote. Slednja v medsebojni interakciji prebere podatke, preveri ustreznost (datum veljavnosti vozovnice mora v tem testu biti aktiven), ter na kartico zapiše podatek o uspešni validaciji. Glede na poznano podatkovno strukturo brezstične kartice IJPP pred validacijo se s programskim paketom COLLIS in pripadajočim bralnikom kartic COLLIS preveri zapisani podatek na za to definirano lokacijo (glej dokument o specifikaciji strukturi kartice – poglavje 3). V kolikor se z orodjem COLLIS ugotovi, da je vozovnica uspešno validirana, in je zapis validacije na kartico uspešno izveden, in se je transakcija zaključila v predpisanem časovnem okvirju in brez napak, se test obravnava kot uspešen. Pogoji za uspešen test je tudi ustrezna zvočna, svetlobna tekstovna signalizacije validirne enote. To pomeni, da mora biti signalizacija za uspešno validacijo skladna s testom 10.5.1.4. V primeru da validirna enota podpira zaslon, mora slednja prav tako izpisati sporočilo »Validacija uspešna«. V kolikor kateri izmed naštetih kriterijev ni izpolnjen (zapis na napačno mesto, nepravilen zapis, napačna signalizacija, oz. tekstovno sporočanje se test ovrednoti kot neuspešen. V kolikor je kateri izmed predhodnih testov 10.5.1.4 ali 10.6.9.1 neuspešen se tega testa ne izvaja.

Ker obstaja v sistemu IJPP več različnih ponudnikov validirnih enot je test potrebno izvesti na vsaj eni napravi vsakega ponudnika, s čimer se hkrati preverja unificirano delovanje validatorjev različnih ponudnikov. V kolikor ima en ponudnik validirnih enot na terenu več različnih modelov, se mora test izvesti na vseh modelih in na vsaj eni napravi.

#### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske opreme validatorja testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je prišlo do sprememb programske opreme validirne enote, ki se tičejo procedur branja/zapisa podatkov na brezstično kartico. Test je obvezen in je predpogoj za izvajanje nadaljnjih validacijskih testov.

Test:	<b>VAL008</b>
Tester: (Ime, Priimek)	
Datum testa (DD.MM.YYYY)	

Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### 10.6.10.9 Test validacije vozovnice, ki je na črni list

S tem testom se ugotavlja ustreznost procedure, ki skrbi za preprečevanje koriščenja enkratne vozovnice iz črne liste v sistemu IJPP. Preverja se, ali se na validirni enoti prepreči validacija enkratne vozovnice iz črne liste, in ali so podatki blokirane vozovnice po validaciji ustrezno zapisani na samo kartico (glej specifikacijo kartice), ter ali je interakcija (signalizacija, sporočila) validirne enote ustrezna.

*Identifikacijska oznaka testa: VAL009*

##### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje kartica z že prednastavljenimi podatki o imetniku, zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice (poglavje 3). Na kartici skladno s specifikacijami le te (poglavje 3) mora biti shranjena tudi enkratna vozovnica za relacijo A-B (glej test nakupa enkratne vozovnice – poglavje 10.6.9.1). Ista serijska številka naložene enkratne vozovnice za relacijo A-B mora biti pred testom vstavljena na črno listo vozovnic, validator pa mora pred samo izvedbo testa biti posodobljen s tako imenovanim paketom vozovnic na črni listi, vključno s testno vozovnico. Pred izvršbo testa se vsebina zapisa na kartici preveri z bralnikom COLISS in pripadajočo programsko opremo COLLIS za upravljanje s prebrano strukturo podatkov iz kartice, s čimer se pridobi vsebina 'batch' kartice pred validacijo. V testnem scenariju se uporabi še validator za beleženje uporabe vozovnic. Za pričetek testnega scenarija, se kartico s produktom enkratne vozovnice, ki je na črni listi prisloni na označeno mesto validirne enote. Slednja v medsebojni interakciji prebere podatke, in preveri ali je enkratna vozovnica na črni listi. Validator mora potrditi, da je vozovnica na črni listi, ter vozovnico zakleniti (zapisati blokado vozovnice na kartico – glej poglavje o strukturi kartice – poglavje 3). Glede na poznano podatkovno strukturo brezstične kartice IJPP pred validacijo se s programskim paketom COLLIS in pripadajočim bralnikom kartic COLLIS preveri zapisani podatek na za to definirano lokacijo. V kolikor se z orodjem COLLIS ugotovi, da je vozovnica blokirana, in je zapis blokade vozovnice na kartico uspešno izveden, in se je transakcija zaključila v predpisanem časovnem okvirju in brez napak, se test obravnava kot uspešen. Pogoji za uspešen test je tudi ustrezna zvočna, svetlobna, tekstovna signalizacija validirne enote. To pomeni, da mora biti signalizacija za neuspešno validacijo skladna s testom 10.5.1.5. V primeru da validirna enota podpira zaslon, mora slednja prav tako izpisati sporočilo »Validacija neuspešna«. V kolikor kateri izmed naštetih kriterijev ni izpolnjen (zapis na napačno mesto, nepravilen zapis, napačna signalizacija, oz. napačno tekstovno sporočanje se test ovrednoti kot neuspešen.

Ker obstaja v sistemu IJPP več različnih ponudnikov validirnih enot je test potrebno izvesti na vsaj eni napravi vsakega ponudnika, s čimer se hkrati preverja unificirano delovanje validatorjev različnih ponudnikov. V kolikor ima en ponudnik validirnih enot na terenu več različnih modelov, se mora test izvesti na vseh modelih in na vsaj eni napravi dotičnega modela.

##### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske opreme validatorja testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je prišlo do sprememb programske opreme validirne enote, ki se tičejo procedur branja/zapisa podatkov na brezstično kartico. Test je obvezen in je predpogoj za izvajanje nadaljnjih validacijskih testov.

Test:	VAL009	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### 10.6.10.10 Test validacije vozovnice s kartico, ki je na črni listi

S tem testom se ugotavlja ustreznost procedure, ki skrbi za preprečevanje akcij z brezstično kartico IJPP, ki se nahaja na seznamu črne liste. Preverja se, ali se na validirni enoti preprečijo validacije s kartico iz črne liste, in ali so podatki blokade kartice ustrezno zapisani na samo kartico (glej poglavje 3), ter ali je interakcija (signalizacija, sporočila) validirne enote ustrezna.

*Identifikacijska oznaka testa: VAL010*

##### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje kartica z že prednastavljenimi podatki o imetniku, zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice. Serijska številka kartice mora biti pred testom vstavljena na črno listo kartic, hkrati pa mora validator pred samo izvedbo testa razpolagati s tako imenovanim seznamom brezstičnih kartic na črni listi. Pred izvršbo testa se vsebina zapisa na kartici preveri z bralnikom COLISS in pripadajočo programsko opremo COLLIS za upravljanje s prebrano strukturo podatkov iz kartice, s čimer se pridobi vsebina 'batch' kartice pred validacijo. V testnem scenariju se uporabi še validator. Za pričetek testnega scenarija, se kartico, katere serijska številka se nahaja na črni listi, prisloni na označeno mesto validirne enote. Slednja v medsebojni interakciji prebere podatke, in preveri ali je brezstična kartica na črni listi. Validator mora potrditi, da je kartica na črni listi in blokirati validacijo. Pri blokadi validacije mora validirna enota zapisati na kartico njeno blokado (glej poglavje o strukturi kartice). Glede na poznano podatkovno strukturo brezstične kartice IJPP pred validacijo se s programskim paketom COLLIS in pripadajočim bralnikom kartic COLLIS preveri zapisani podatek na za to definirano lokacijo (glej dokument o specifikirani strukturi kartice). V kolikor se z orodjem COLLIS ugotovi, da je kartica pravilno blokirana, in je zapis blokade kartice na le to uspešno izveden, in se je transakcija zaključila v predpisanem časovnem okvirju in brez napak, se test obravnava kot uspešen. Pogoj za uspešen test je tudi ustrezna zvočna, svetlobna, tekstovna signalizacije validirne enote. To pomeni, da mora biti signalizacija za neuspešno validacijo skladna s testom 10.5.1.5. V primeru da validirna enota podpira zaslon, mora slednja prav tako izpisati sporočilo »Validacija neuspešna. Blokirana kartica«. V kolikor kateri izmed naštetih kriterijev ni izpolnjen (zapis

na napačno mesto, nepravilen zapis, napačna signalizacija, oz. napačno tekstovno sporočanje se test ovrednoti kot neuspešen.

Ker obstaja v sistemu IJPP več različnih ponudnikov validirnih enot je test potrebno izvesti na vsaj eni napravi vsakega ponudnika, s čimer se hkrati preverja unificirano delovanje validatorjev različnih ponudnikov. V kolikor ima en ponudnik validirnih enot na terenu več različnih modelov, se mora test izvesti na vseh modelih in na vsaj eni napravi dotičnega modela.

#### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske opreme validatorja testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je prišlo do sprememb programske opreme validirne enote, ki se tičejo procedur branja/zapisa podatkov na brezstično kartico, oz. pri vsakršnih spremembah glede upravljanja črnih list. Test je obvezen in je predpogoj za izvajanje nadaljnjih validacijskih testov.

Test:	<b>VAL010</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.10.11 Test validacije vozovnice zunaj časa njene veljavnosti**

S tem testom se ugotavlja ustreznost procedure, ki skrbi za preprečevanje akcij z vozovnicami, ki so izven časa veljavnosti (potekla veljavnost). Preverja se, ali se na validirni enoti preprečijo validacije časovno neveljavnih vozovnic, ter ali je interakcija (signalizacija, sporočila) validirne enote ustrezna.

*Identifikacijska oznaka testa: VAL011*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje kartica z že prednastavljenimi podatki o imetniku, zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice. Na kartici mora biti zapisana enkratna vozovnica za relacijo A-B s terminom veljavnosti, ki je že izven dovoljenega intervala. Pred izvedbo testa se vsebina zapisa na kartici preveri z bralnikom COLISS in pripadajočo programsko opremo COLLIS za upravljanje s prebrano strukturo podatkov iz kartice, s čimer se pridobi vsebina 'batch' kartice pred validacijo. V testnem scenariju se uporabi še validator. Za pričetek testnega scenarija, se kartico z naloženim produktom enkratne vozovnice prisloni na označeno mesto validirne enote. Slednja v medsebojni interakciji prebere podatke, in preveri razpoložljiv produkt. Validator mora ugotoviti, da je produkt neveljaven. Neuspešna validacija se zapiše na kartico (glej dokument z definirano podatkovno strukturo – poglavje 3). Glede na poznano podatkovno strukturo brezstične kartice IJPP pred validacijo se s programskim paketom COLLIS in pripadajočim bralnikom kartic COLLIS

preveri zapisani podatek na za to definirano lokacijo (poglavje 3). V kolikor se z orodjem COLLIS ugotovi, da je kartica pravilno zapisana, in je zapis uspešno izveden, in se je transakcija zaključila v predpisanem časovnem okvirju in brez napak, se test obravnava kot uspešen. Pogoji za uspešen test je tudi ustrezna zvočna, svetlobna, tekstovna signalizacija validirne enote. To pomeni, da mora biti signalizacija za neuspešno validacijo skladna s testom 10.5.1.5. V primeru da validirna enota podpira zaslon, mora slednja prav tako izpisati sporočilo »Validacija neuspešna. Neveljavna vozovnica«. V kolikor kateri izmed naštetih kriterijev ni izpolnjen (zapis na napačno mesto, nepravilen zapis, napačna signalizacija, oz. napačno tekstovno sporočanje se test ovrednoti kot neuspešen.

Ker obstaja v sistemu IJPP več različnih ponudnikov validirnih enot je test potrebno izvesti na vsaj eni napravi vsakega ponudnika, s čimer se hkrati preverja unificirano delovanje validatorjev različnih ponudnikov. V kolikor ima en ponudnik validirnih enot na terenu več različnih modelov, se mora test izvesti na vseh modelih in na vsaj eni napravi dotičnega modela.

#### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske opreme validatorja testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je prišlo do sprememb programske opreme validirne enote, ki se tičejo procedur branja/zapisa podatkov na brezstično kartico. Test je obvezen in je predpogoj za izvajanje nadaljnjih validacijskih testov.

Test:	<b>VAL011</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.10.12 Test validacije vozovnice po preteku maksimalnega časa prestopanja**

S tem testom se ugotavlja ustreznost procedure, ki skrbi za preprečevanje akcij z vozovnicami, ki so izven veljavnega časa, ki dovoljuje prestopanje. Preverja se, ali se na validirni enoti preprečijo validacije vozovnic, ki so izven dovoljenega časa prestopanj, ter ali je interakcija (signalizacija, sporočila) validirne enote ustrezna.

*Identifikacijska oznaka testa: VAL012*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje kartica z že prednastavljenimi podatki o imetniku, zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice. Na kartici mora biti zapisana enkratna vozovnica za relacijo A-B s časom prestopanja, ki je izven dovoljenega intervala. Pred izvedbo testa se vsebina zapisa na kartici preveri z bralnikom COLISS in pripadajočo

programsko opremo COLLIS za upravljanje s prebrano strukturo podatkov iz kartice, s čimer se pridobi vsebina 'batch' kartice pred validacijo. V testnem scenariju se uporabi še validator. Za pričetek testnega scenarija, se kartico z naloženim produktom enkratne vozovnice s pretečenim intervalom prestopanja prisloni na označeno mesto validirne enote. Slednja v medsebojni interakciji prebere podatke, in preveri razpoložljiv produkt. Validator mora ugotoviti, da je produkt naložen, vendar pogoj za prestopanje ni več veljaven. Neuspešna validacija se zapiše na kartico (glej dokument z definirano podatkovno strukturo – poglavje 3). Glede na poznano podatkovno strukturo brezstične kartice IJPP pred validacijo se s programskim paketom COLLIS in pripadajočim bralnikom kartic COLLIS preveri zapisani podatek na za to definirano lokacijo (glej dokument o specifikirani strukturi kartice). V kolikor se z orodjem COLLIS ugotovi, da je kartica pravilno zapisana, in je zapis uspešno izveden, in se je transakcija zaključila v predpisanem časovnem okvirju in brez napak, se test obravnava kot uspešen. Pogoj za uspešen test je tudi ustrezna zvočna, svetlobna, tekstovna signalizacija validirne enote. To pomeni, da mora biti signalizacija za neuspešno validacijo skladna s testom 10.5.1.5. V primeru da validirna enota podpira zaslon, mora slednja prav tako izpisati sporočilo »Validacija neuspešna. Prestop ni dovoljen«. V kolikor kateri izmed naštetih kriterijev ni izpolnjen (zapis na napačno mesto, nepravilen zapis, napačna signalizacija, oz. napačno tekstovno sporočanje se test ovrednoti kot neuspešen.

Ker obstaja v sistemu IJPP več različnih ponudnikov validirnih enot je test potrebno izvesti na vsaj eni napravi vsakega ponudnika, s čimer se hkrati preverja unificirano delovanje validatorjev različnih ponudnikov. V kolikor ima en ponudnik validirnih enot na terenu več različnih modelov, se mora test izvesti na vseh modelih in na vsaj eni napravi dotičnega modela.

#### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske opreme validatorja testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je prišlo do sprememb programske opreme validirne enote, ki se tičejo procedur branja/zapisa podatkov na brezstično kartico. Test je obvezen in je predpogoj za izvajanje nadaljnjih validacijskih testov.

Test:	<b>VAL012</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.10.13 Test validacije vozovnice na neustrezni postaji, ki ne pripada relaciji A-B**

S tem testom se ugotavlja ustreznost procedure, ki skrbi za preprečevanje validacije enkratne vozovnice, ki ne pripada relaciji A-B. Preverja se, ali se na validirni enoti preprečijo validacije vozovnic, ki ne pripadajo relaciji B, (signalizacija, sporočila) validirne enote ustrezna.

*Identifikacijska oznaka testa: VAL013*



### Scenarij testa

Za izvedbo testa se v prvi fazi potrebuje kartica z že prednastavljenimi podatki o imetniku, zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice. Na kartici mora biti zapisana veljavna enkratna vozovnica za relacijo C-D, ki se bo v testu skušala validirati na validatorju, ki pokriva relacijo A-B. Pred izvedbo testa se vsebina zapisa na kartici preveri z bralnikom COLISS in pripadajočo programsko opremo COLLIS za upravljanje s prebrano strukturo podatkov iz kartice, s čimer se pridobi vsebina kartice pred validacijo (batch). V testnem scenariju se uporabi še validator kot v predhodnih testih, ki je namenjen za relacijo A-B. Za pričetek testnega scenarija, se kartico z naloženim veljavnim produktom enkratne vozovnice prisloni na označeno mesto validirne enote. Slednja v medsebojni interakciji prebere podatke, in preveri razpoložljiv produkt. Validator mora ugotoviti, da je naložen produkt (relacija C-D) neveljaven na relaciji A-B. Neuspešna validacija se zapiše na kartico (glej dokument z definirano podatkovno strukturo). Glede na poznano podatkovno strukturo brezstične kartice IJPP pred validacijo se s programskim paketom COLLIS in pripadajočim bralnikom kartic COLLIS preveri zapisani podatek na za to definirano lokacijo (glej dokument o specificirani strukturi kartice). Pri tem testu mora enkratna vozovnica ostati veljavna, validator pa mora uporabniku signalizirati neuspešno validacijo (glej razlago v nadaljevanju). V kolikor se z orodjem COLLIS ugotovi, da je kartica pravilno zapisana, in je zapis uspešno izveden, in se je transakcija zaključila v predpisanem časovnem okvirju in brez napak, se test obravnava kot uspešen. Pogoj za uspešen test je tudi ustrezna zvočna, svetlobna, tekstovna signalizacija validirne enote. To pomeni, da mora biti signalizacija za neuspešno validacijo skladna s testom 10.5.1.5. V primeru da validirna enota podpira zaslon, mora slednja prav tako izpisati sporočilo »Validacija neuspešna. Neveljavna karta relacije«. V kolikor kateri izmed naštetih kriterijev ni izpolnjen (zapis na napačno mesto, nepravilen zapis, napačna signalizacija, oz. napačno tekstovno sporočanje se test ovrednoti kot neuspešen.

Ker obstaja v sistemu IJPP več različnih ponudnikov validirnih enot je test potrebno izvesti na vsaj eni napravi vsakega ponudnika, s čimer se hkrati preverja unificirano delovanje validatorjev različnih ponudnikov. V kolikor ima en ponudnik validirnih enot na terenu več različnih modelov, se mora test izvesti na vseh modelih in na vsaj eni napravi dotičnega modela.

### Kdaj izvesti test

Test je potrebno izvajati dokler specifična verzija izdane programske opreme validatorja testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je prišlo do sprememb programske opreme validirne enote, ki se tičejo procedur branja/zapisa podatkov na brezstično kartico. Test je obvezen in je predpogoj za izvajanje nadaljnjih validacijskih testov.

Test:	VAL013	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.11**      *TESTI VERIFIKACIJ VOZOVNICE NA KARTOMATU OZ. PRODAJNEM MESTU*

Testi verifikacij vozovnice na kartomatu oz. prodajnem mestu služijo preverjanju podatkov, ki so bili zapisani na kartico ob nakupu vozovnice. Testi zajemajo verifikacijo takoj po uspešnem nakupu, verifikacijo po tem, ko je bila vozovnica dodana na črno listo, verifikacijo po tem, ko je bila kartica dodana na črno listo, verifikacijo po validaciji znotraj dovoljenega časa prestopanja in verifikacijo po preteku časa v katerem je dovoljeno prestopanje.

##### **10.6.11.1**      **Test uspešne verifikacije kartice in zapisanih podatkov**

S tem testom se ugotavlja ustreznost procedure, ki najprej na kartomatu zapiše produkt (enkratno testno vozovnico) na brezstično kartico, ter jo po končanem nakupu še verificira (prebere in prikaže vsebino naloženega produkta iz brezstične kartice). Preverja se, ali je zapisan produkt na kartici res to, kar je bilo izbrano na kartomatu oz. prodajnem mestu.

*Identifikacijska oznaka testa: VOZ004*

##### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje kartica z že prednastavljenimi podatki o imetniku, zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice. Na kartici skladno s specifikacijami le ta mora biti shranjena tudi datumsko veljavna enkratna vozovnica za relacijo A-B (glej test nakupa enkratne vozovnice – poglavje 10.6.9.1). Pred izvršbo testa se vsebina zapisa na kartici preveri z orodjem COLISS, s čimer se pridobi vsebina 'batch' kartice. V testnem scenariju se uporabi še kartomat, tokrat v vlogi bralnika vsebine kartice. Za pričetek testnega scenarija, se kartico z veljavnim produktom enkratne vozovnice prisloni na označeno mesto kartomata (oz. na prodajnem mestu samem). Kartomat (prodajno mesto) v medsebojni interakciji prebere podatke o predhodno naloženi enkratni vozovnici in jih prikaže na zaslon. Če se izpisani podatki na zaslonu ujemajo z naloženim testnim produktom (enkratna vozovnica, relacija, veljavnost, imetnik), je test že delno uspešen. Kot dodatni pogoj uspešnosti je zahtevana še preverba podatkov in celotne strukture kartice s programskim paketom COLLIS in pripadajočim bralnikom kartic COLLIS. Ker v testu ne gre za koriščenje produkta, mora vsebina kartice ostati nespremenjena, kar pomeni, da je struktura zapisanih podatkov brezstične kartice enaka pred in po izvršenem testu. Če to drži se test obravnava kot uspešen. Če katerikoli pogoj ne drži (izpisana vsebina produkta ne ustreza naloženemu produktu enkratne vozovnice, odstopanje vsebine kartice pred in po testu, ipd.) se test ovrednoti kot neuspešen in ga je potrebno ponoviti. V primeru neuspešnega testa prodajno mesto ne dobi odobritve za integracijo v IJPP sistem.

Ker lahko obstaja v sistemu IJPP več različnih ponudnikov prodajnih enot (kartomatov) je test potrebno izvesti na vsaj eni napravi vsakega ponudnika. V kolikor ima en ponudnik kartomatov na terenu več različnih modelov, se mora test izvesti na vseh modelih.

##### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske opreme kartomata testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz.

spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je prišlo do sprememb programske opreme kartomata, ki se tičejo procedur branja/zapisa podatkov na brezstično kartico. Test je obvezen in je predpogoj za izvajanje nadaljnjih verifikacijskih testov.

Test:	<b>VOZ004</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### 10.6.11.2 Test verifikacije izven časa veljavnosti enkratne vozovnice

S tem testom se ugotavlja ustreznost procedure verifikacije vozovnice na prodajnem mestu (kartomatu), kateri je potekla časovna veljavnost. Preverja se, ali je mogoče verificirati vozovnico, ki je izven časovne veljavnosti.

*Identifikacijska oznaka testa: VOZ005*

##### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje kartica z že prednastavljenimi podatki o imetniku, zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice. Na kartici skladno s specifikacijami le te (glej poglavje specifikacije IJPP kartice) mora biti shranjena datumsko neveljavna enkratna vozovnica za relacijo A-B (neveljavna pomeni sledeče: na brezstično kartico se naloži najprej veljavno enkratno vozovnico (glej poglavje 10.6.9.1), in se počaka toliko časa, da časovna veljavnost vozovnice poteče). Pred izvršbo testa se vsebina zapisa na kartici preveri z orodjem COLISS, s čimer se pridobi vsebina 'batch' kartice. V testnem scenariju se uporabi še kartomat, tokrat v vlogi bralnika vsebine kartice (verifikacija). Za pričetek testnega scenarija, se kartico s produktom enkratne vozovnice (časovno neveljavna vozovnica) prisloni na označeno mesto kartomata (oz. na prodajnem mestu samem). Kartomat (prodajno mesto) v medsebojni interakciji prebere podatke o naloženi enkratni vozovnici. Kartomat mora podatke o vozovnici prebrati in sporočiti uporabniku, da vozovnica ni več veljavna. Skladno z neveljavnostjo mora kartomat na kartico zapisati podatek o neuspešni verifikaciji. Kot dodatni pogoj uspešnosti testa je zahtevana še preverba podatkov in celotne strukture kartice s programskim paketom COLLIS in pripadajočim bralnikom kartic COLLIS po izvedeni neuspešni verifikaciji enkratne vozovnice. Ker v testu ne gre za koriščenje produkta, mora vsebina kartice po testu odražati zgolj spremembo, ki se nanaša na neuspešno verifikacijo. Vsi ostali podatki morajo po testu ostati nespremenjeni. Če to drži se test obravnava kot uspešen. Če katerikoli pogoj ne drži (izpisana vsebina produkta ne ustreza naloženemu produktu enkratne vozovnice, npr. uspešna verifikacija, neuspešen zapis neuspešne verifikacije na kartico ipd.) se test ovrednoti kot neuspešen in ga je potrebno ponoviti. V primeru neuspešnega testa prodajno mesto ne dobi odobritve za integracijo v IJPP sistem.

Ker lahko obstaja v sistemu IJPP več različnih ponudnikov prodajnih enot (kartomatov) je test potrebno izvesti na vsaj eni napravi vsakega ponudnika. V kolikor ima en ponudnik kartomatov na terenu več različnih modelov, se mora test izvesti na vseh modelih.

#### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske opreme kartomata testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je prišlo do sprememb programske opreme kartomata, ki se tičejo procedur branja/zapisa podatkov na brezstično kartico. Test je obvezen.

Test:	<b>VOZ005</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.11.3 Test verifikacije po uspešni validaciji in znotraj dovoljenega časa prestopanja**

S tem testom se ugotavlja ustreznost procedure verifikacije vozovnice na prodajnem mestu (kartomatu), po izvedeni validaciji. Preverja se, ali je enkratno vozovnico z določenim časovnim okvirjem prestopanja po uspešni validaciji moč verificirati.

*Identifikacijska oznaka testa: VOZ006*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje kartica z že prednastavljenimi podatki o imetniku, zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice. Na kartici skladno s specifikacijami le te (glej poglavje specifikacije IJPP kartice) mora biti shranjena datumsko veljavna enkratna vozovnica za relacijo A-B z aktivnim časovnim okvirjem, ki dovoljuje prestopanje. Pred izvršbo testa se vsebina zapisa na kartici preveri z orodjem COLISS, s čimer se pridobi vsebina 'batch' kartice. V testnem scenariju se uporabi še validator za validacijo vozovnice (glej test 10.6.10.8) in kartomat, tokrat v vlogi bralnika vsebine kartice (verifikacija). Za pričetek testnega scenarija, se kartico z veljavnim produktom enkratne vozovnice (dovoljeno prestopanje) prisloni na označeno mesto validatorja, s čimer se enkratna vozovnica validira (glej test 10.6.10.8). Validator skladno s testom 10.6.10.8 na kartico zapiše uspešno validacijo. Po validaciji ponovno z orodjem COLLIS (čitalnik kartic, programska oprema) preverimo vsebino zapisa kartice. V naslednjem koraku testa - po uspešni validaciji in znotraj časovnega okvirja, ki dovoljuje prestopanje, prislonimo kartico na označeno mesto kartomata, kjer izvedemo verifikacijo. Kartomat (prodajno mesto) mora vsebino prebrati in na zaslon izpisati podatke o veljavni enkratni vozovnici in preostali čas veljavnosti, do kdaj je karta še uporabna. Skladno z veljavnostjo mora kartomat na kartico zapisati podatek o uspešni verifikaciji. Kot dodatni

pogoj uspešnosti testa je zahtevana še preverba podatkov in celotne strukture kartice s programskim paketom COLLIS in pripadajočim bralnikom kartic COLLIS po izvedeni uspešni verifikaciji enkratne vozovnice. Na koncu testa se izvede še neposredna primerjava t.i. 'batche-v' pred validacijo, po validaciji in po verifikaciji, ki morajo odražati spremembe v podatkovni strukturi, kot jo predpisuje poglavje o strukturi kartice IJPP. Če so spremembe skladne s specifikacijo strukture kartice, in je verifikacija po validaciji uspešna se test obravnava kot uspešen. Če katerikoli pogoj ne drži (izpisana vsebina produkta ne ustreza naloženemu produktu enkratne vozovnice, npr. neuspešna verifikacija, neuspešen zapis verifikacije na kartico ipd.) se test ovrednoti kot neuspešen in ga je potrebno ponoviti. V primeru neuspešnega testa prodajno mesto ne dobi odobritve za integracijo v IJPP sistem.

Ker lahko obstaja v sistemu IJPP več različnih ponudnikov prodajnih enot (kartomatov) je test potrebno izvesti na vsaj eni napravi vsakega ponudnika. V kolikor ima en ponudnik kartomatov na terenu več različnih modelov, se mora test izvesti na vseh modelih.

#### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske opreme kartomata testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je prišlo do sprememb programske opreme kartomata, ki se tičejo procedur branja/zapisa podatkov na brezstično kartico. Test je obvezen.

Test:	VOZ006	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.11.4 Test verifikacije po uspešni validaciji in izven dovoljenega časa prestopanja**

S tem testom se ugotavlja ustreznost procedure verifikacije vozovnice na prodajnem mestu (kartomatu), po izvedeni validaciji in ob pogoju, da se je dovoljeni čas prestopanja iztekel. Preverja se, ali je enkratno vozovnico po validaciji in po preteku dovoljenega časa prestopanja moč verificirati.

*Identifikacijska oznaka testa: VOZ007*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje kartica z že prednastavljenimi podatki o imetniku, zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice. Na kartici skladno s specifikacijami le te (glej poglavje specifikacije IJPP kartice) mora biti shranjena datumsko veljavna enkratna vozovnica za relacijo A-B z aktivnim časovnim okvirjem, ki dovoljuje prestopanje. Pred izvršbo testa se vsebina zapisa na kartici preveri z orodjem COLISS, s čimer se pridobi vsebina

'batch' kartice. V testnem scenariju se uporabi še validator za validacijo vozovnice (glej test 10.6.10.8) in kartomat, tokrat v vlogi bralnika vsebine kartice (verifikacija). Za pričetek testnega scenarija, se kartico z veljavnim produktom enkratne vozovnice (dovoljeno prestopanje) prisloni na označeno mesto validatorja, s čimer se enkratna vozovnica validira (glej test 10.6.10.8). Validator skladno s testom 10.6.10.8 na kartico zapiše uspešno validacijo. Po validaciji ponovno z orodjem COLLIS (čitalnik kartic, programska oprema) preverimo vsebino zapisa kartice. Za nadaljevanje testa je nato potrebno počakati, da se časovni okvir dovoljenega prestopanja izteče. V naslednjem koraku testa - po uspešni validaciji in izven časovnega okvirja, ki dovoljuje prestopanje, prislonimo kartico na označeno mesto kartomata, kjer izvedemo verifikacijo. Kartomat (prodajno mesto) mora vsebino prebrati in na zaslon izpisati podatke o neveljavni enkratni vozovnici. Skladno z neveljavnostjo mora kartomat na kartico zapisati podatek o neuspešni verifikaciji. Kot dodatni pogoj uspešnosti testa je zahtevana še preverba podatkov in celotne strukture kartice s programskim paketom COLLIS in pripadajočim bralnikom kartic COLLIS po izvedeni verifikaciji enkratne vozovnice, ki ji je potekla veljavnost. Na koncu testa se izvede še neposredna primerjava t.i. 'batche-v' pred validacijo, po validaciji in po verifikaciji, ki morajo odražati spremembe v podatkovni strukturi, kot jo predpisuje poglavje o strukturi kartice IJPP. Če so spremembe skladne s specifikacijo strukture kartice, in je verifikacija po validaciji ustrezna (se ugotovi in zapiše neveljavnost vozovnice) se test obravnava kot uspešen. Če katerikoli pogoj ne drži (izpisana vsebina produkta ne ustreza naloženemu produktu enkratne vozovnice, npr. uspešna verifikacija, neuspešen zapis verifikacije na kartico ipd.) se test ovrednoti kot neuspešen in ga je potrebno ponoviti. V primeru neuspešnega testa prodajno mesto ne dobi odobritve za integracijo v IJPP sistem.

Ker lahko obstaja v sistemu IJPP več različnih ponudnikov prodajnih enot (kartomatov) je test potrebno izvesti na vsaj eni napravi vsakega ponudnika. V kolikor ima en ponudnik kartomatov na terenu več različnih modelov, se mora test izvesti na vseh modelih.

#### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske opreme kartomata testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je prišlo do sprememb programske opreme kartomata, ki se tičejo procedur branja/zapisa podatkov na brezstično kartico. Test je obvezen.

Test:	<b>VOZ007</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.11.5 Test verifikacije vozovnice s kartico, ki je na črni listi**

S tem testom se ugotavlja ustreznost procedure verifikacije vozovnice na prodajnem mestu (kartomatu), ki se nahaja na kartici katera je na seznamu črne liste. Preverja se, ali je enkratno vozovnico iz kartice ki je na črni listi mogoče verificirati.

*Identifikacijska oznaka testa: VOZ008*

#### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje kartica z že prednastavljenimi podatki o imetniku, zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice. Pred testom se na kartico skladno s specifikacijami le te (glej poglavje specifikacije IJPP kartice) naloži datumsko veljavna enkratna vozovnica za relacijo A-B (glej test 10.6.9.1). Ko je vozovnica naložena se serijsko številko kartice umesti na črno listo, kar predstavlja naslednji korak. Kartomat mora pred pričetkom tega testa razpolagati s posodobljenim seznamom kartic na črni listi, kjer se nahaja tudi testna kartica. Pred izvedbo naslednjega koraka v testu se najprej preveri vsebina zapisa na kartici z orodjem COLISS (bralnik kartic in pripadajoča programska oprema), s čimer se pridobi vsebina 'batch' kartice. Za pričetek testnega scenarija, se kartico (ki je sedaj že na seznamu črne liste) z veljavnim produktom enkratne vozovnice (dovoljeno prestopanje) prisloni na označeno mesto kartomata. Kartomat v tej iteraciji prebere podatke iz testne kartice in primerja serijsko številko z razpoložljivim seznamom. Kartomat mora ugotoviti, da je kartica na črni listi, in skladno s tem zapisati blokado na testno kartico. Hkrati z opravljeno blokado kartomat sporoči uporabniku naslednje sporočilo: »Blokirana kartica. Verifikacija neuspešna«. Naslednji korak predstavlja ponovno branje vsebine kartice z orodjem COLLIS (čitalnik kartic, programska oprema), s čimer se preveri vsebino zapisa kartice. Če sta podatka o blokadi kartice in neuspešni verifikaciji vozovnice ustrezno zapisani (glej standard kartice IJPP), in je algoritem verifikacije na kartomatu javil blokado kartice in neuspešno verifikacijo se test vrednoti kot uspešen. Če katerikoli pogoj ne drži (kartomat ne blokira kartice, ne izpiše ustreznega sporočila, ne zapiše blokade in neuspešne verifikacije po specifikacijah kartice IJPP, itd.) se test ovrednoti kot neuspešen in ga je potrebno ponoviti. V primeru neuspešnega testa prodajno mesto ne dobi odobritve za integracijo v IJPP sistem.

Ker lahko obstaja v sistemu IJPP več različnih ponudnikov prodajnih enot (kartomato) je test potrebno izvesti na vsaj eni napravi vsakega ponudnika. V kolikor ima en ponudnik kartomato na terenu več različnih modelov, se mora test izvesti na vseh modelih.

#### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske opreme kartomata testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je prišlo do sprememb programske opreme kartomata, ki se tičejo procedur branja/zapisa podatkov na brezstično kartico. Test je obvezen.

Test:	<b>VOZ008</b>
Tester: (Ime, Priimek)	
Datum testa (DD.MM.YYYY)	

Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### 10.6.11.6 Test verifikacije vozovnice, ki se nahaja na črni listi

S tem testom se ugotavlja ustreznost procedure verifikacije vozovnice na prodajnem mestu (kartomatu), ki se nahaja seznamu blokiranih vozovnic. Preverja se, ali je enkratno vozovnico iz črne liste mogoče verificirati.

*Identifikacijska oznaka testa: VOZ009*

##### *Scenarij testa*

Za izvedbo testa se v prvi fazi potrebuje kartica z že prednastavljenimi podatki o imetniku, zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice. Pred testom se na kartico skladno s specifikacijami le te (glej poglavje specifikacije IJPP kartice) naloži datumsko veljavna enkratna vozovnica za relacijo A-B (glej test 10.6.9.1). Ko je vozovnica naložena se serijsko številko vozovnice umesti na črno listo, kar predstavlja naslednji korak. Kartomat mora pred pričetkom tega testa razpolagati s posodobljenim seznamom vozovnic na črni listi, kjer se nahaja tudi testna enkratna vozovnica za relacijo A-B. Pred izvedbo naslednjega koraka v testu se najprej preveri vsebina zapisa na kartici z orodjem COLISS (bralnik kartic in pripadajoča programska oprema), s čimer se pridobi vsebina 'batch' kartice. Za pričetek testnega scenarija, se kartico z vozovnico, ki je sedaj že na seznamu črne liste vozovnic, prisloni na označeno mesto kartomata. Kartomat v tej iteraciji prebere podatke iz testne kartice in primerja serijsko številko produkta (vozovnice) z razpoložljivim seznamom vozovnic iz črne liste. Kartomat mora ugotoviti, da je vozovnica na črni listi, in skladno s tem zapisati blokado vozovnice na testno kartico. Hkrati z opravljeno blokado vozovnice kartomat sporoči uporabniku naslednje sporočilo: »Blokirana vozovnica. Verifikacija neuspešna«. Naslednji korak predstavlja ponovno branje vsebine kartice z orodjem COLLIS (čitalnik kartic, programska oprema), s čimer se preveri vsebino zapisa kartice. Če sta podatka o blokadi vozovnice in neuspešne verifikacije vozovnice ustrezno zapisani (glej standard kartice IJPP), in je algoritem verifikacije na kartomatu javil blokado vozovnice in neuspešno verifikacijo se test vrednoti kot uspešen. Če katerikoli pogoj ne drži (kartomat ne blokira vozovnice, ne izpiše ustreznega sporočila, ne zapiše blokade in neuspešne verifikacije vozovnice po specifikacijah kartice IJPP, itd.) se test ovrednoti kot neuspešen in ga je potrebno ponoviti. V primeru neuspešnega testa prodajno mesto ne dobi odobritve za integracijo v IJPP sistem.

Ker lahko obstaja v sistemu IJPP več različnih ponudnikov prodajnih enot (kartomатов) je test potrebno izvesti na vsaj eni napravi vsakega ponudnika. V kolikor ima en ponudnik kartomатов na terenu več različnih modelov, se mora test izvesti na vseh modelih.

##### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske opreme kartomata testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je prišlo do sprememb programske opreme kartomata, ki se tičejo procedur branja/zapisa podatkov na brezstično kartico. Test je obvezen.



Test:	VOZ009	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### 10.6.12 TESTI NADZOROV VOZOVNIC Z MOBILNIM VALIDATORJEM (KONTROLOR)

Testi nadzorov vozovnic so namenjeni ugotavljanju ustreznega delovanja mobilnih validatorjev, ki so namenjeni kontrolorjem. V ta sklop testov so zajeti naslednji testni scenariji: kontrola uspešno validirane vozovnice, kontrola nevalidirane enkratne vozovnice, kontrola vozovnice znotraj območja veljavnosti, kontrola vozovnice zunaj območja veljavnosti in kontrola istovetnosti osebe, ki razpolaga z vozovnico in njena upravičenost.

##### 10.6.12.1 Test kontrole z mobilnim validatorjem vozovnice, ki je bila uspešno validirana

Test sovпада s predhodno definiranim testom uspešne validacije vozovnice (10.6.10.8), kateri je predpogoj za izvedbo tega testa. Če test 10.6.10.8 ni uspešen se procedura tega testa ne izvaja. Test je namenjen ugotavljanju ustreznega delovanja mobilnega validatorja v primeru, ko je bila vozovnica že uspešno validirana.

*Identifikacijska oznaka testa: KON001*

##### *Scenarij testa*

V testu se uporabi IJPP kartica z že prednastavljenimi podatki o imetniku, zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice. Pred izvedbo testa mora biti na kartico naložena veljavna enkratna vozovnica (test 10.6.9.1) za relacijo A-B, in slednja mora biti hkrati uspešno validirana na validirni enoti (test 10.6.10.8). Na tem mestu testa se mora zabeležiti čas in datum validacije za potrebe kasnejše primerjave z izpisanim časom in datumom na mobilni validirni enoti. Ko so vse predhodne procedure izvedene se s paketom COLLIS (čitalnik kartic in pripadajoča programska oprema) pridobi vsebina kartice, ki v nadaljevanju služi kot osnova primerjalnega testa spremembe podatkov na kartici. Naslednji korak testa zajema postavitev IJPP kartice z validiranim veljavnim produktom v NFC polje (označeno mesto) mobilne validirne enote (kontrolor). V medsebojni interakciji mora mobilna validirna enota prebrati podatek o veljavni vozovnici in podatek o času in datumu validacije, ki je bila izvedena na klasični validirni enoti. Podatki o veljavni vozovnici (vrsta vozovnice, relacija, veljavnost), podatki validacije (čas in datum validacije) in podatki imetnika vozovnice se morajo izpisati na zaslonu mobilne validirne enote. Mobilni validator mora na kartico zapisati podatek (glej specifikacijo IJPP kartice) o izvedeni mobilni validaciji (kontroli). Ko je procedura izmenjave med kartico in mobilnim validatorjem zaključena se ponovno s paketom COLLIS pridobi t.i. 'batch' vsebine kartice, ki mora v primerjavi s predhodnim 'batch-em' odražati spremembo v sektorju

zapisane izvedene mobilne validacije (kontrole). Vsa ostala vsebina kartice mora ostati v testu nespremenjena. Če so podatki na mobilni enoti pravilno izpisani, in vsebina batchev odraža zgolj spremembo pri zapisu mobilne kontrole (validacije), in so vse procedure zaključene v zahtevanem času se test ovrednoti kot uspešen. V nasprotnem primeru (nepravilni in nepopolni podatki, napačen čas in datum prvotne validacije, nepravilno in nepravočasno zaključevanje transakcije, neustrezno beleženje transakcije, neustrezní zapis na kartico) se test ovrednoti kot neuspešen - mobilna validirna enota ne more pridobiti soglasja za integracijo v IJPP sistem.

Ker lahko obstaja v sistemu IJPP več različnih ponudnikov mobilnih validatorjev (kontrolorjev) je test potrebno izvesti na vsaj eni napravi vsakega ponudnika. V kolikor ima en ponudnik mobilnih validatorjev (kontrolorjev) na terenu več različnih modelov, se mora test izvesti na vseh modelih.

#### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske opreme mobilnega validatorja (kontrolorja) testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je prišlo do sprememb programske opreme mobilnega validatorja, ki se tičejo procedur branja/zapisa podatkov na brezstično kartico. Test je obvezen.

Test:	KON001	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.12.2 Test kontrole z mobilnim validatorjem vozovnice, ki ni bila validirana**

Test je namenjen ugotavljanju ustreznega delovanja mobilnega validatorja v primeru, ko še vozovnica ni bila uspešno validirana. Test povzema realen scenarij, kadar uporabnik pri vstopu na javno prevozno sredstvo ni validiral vozovnice.

*Identifikacijska oznaka testa: KON002*

#### *Scenarij testa*

V testu se uporabi IJPP kartica z že prednastavljenimi podatki o imetniku, zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice. Pred izvedbo testa mora biti na kartico naložena veljavna enkratna vozovnica (test 10.6.9.1) za relacijo A-B, in slednja ne sme biti validirana. Ko so vse predhodne procedure izvedene se s paketom COLLIS (čitalnik kartic in pripadajoča programska oprema) pridobi vsebina kartice, ki v nadaljevanju služi kot osnova primerjalnega testa spremembe podatkov na kartici. Naslednji korak testa zajema postavitve IJPP kartice z nevalidiranim veljavnim produktom vozovnice v NFC polje (označeno mesto) mobilne validirne enote (kontrolor). V

medsebojni interakciji mora mobilna validirna enota prebrati podatek o veljavni vozovnici in podatek morebitne validacije. Ker vozovnica ni bila validirana se mora podatek o nevalidirani vozovnici izpisati najprej na zaslonu mobilnega validatorja, šele nato se prikažejo preostali podatki o veljavnosti vozovnice. Ker gre za tip testa veljavne vozovnice se hkrati izvede validacija na mobilni validirni enoti (kontrolorju), ki se zapiše na kartico. Ko je procedura izmenjave podatkov med kartico in mobilnim validatorjem zaključena se ponovno s paketom COLLIS pridobi t.i. 'batch' vsebine kartice, ki mora v primerjavi s predhodnim 'batch-em' odražati spremembo v sektorju zapisane izvedene mobilne validacije (kontrole) in v sektorju izvedene validacije vozovnice. Vsa ostala vsebina kartice mora ostati v testu nespremenjena. Če so podatki na mobilni enoti pravilno izpisani, in vsebina batchev odraža zgolj spremembo pri zapisu mobilne kontrole (validacije) in zapisu validacije, in so vse procedure zaključene v zahtevanem času se test ovrednoti kot uspešen. V nasprotnem primeru (nepravilni in nepopolni podatki, napačni podatki vozovnice, nepravilno in nepravočasno zaključevanje transakcije, neustrezno beleženje transakcije, neustrezni zapis na kartico) se test ovrednoti kot neuspešen - mobilna validirna enota ne more pridobiti soglasja za integracijo v IJPP sistem.

Ker lahko obstaja v sistemu IJPP več različnih ponudnikov mobilnih validatorjev (kontrolorjev) je test potrebno izvesti na vsaj eni napravi vsakega ponudnika. V kolikor ima en ponudnik mobilnih validatorjev (kontrolorjev) na terenu več različnih modelov, se mora test izvesti na vseh modelih.

#### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske opreme mobilnega validatorja (kontrolorja) testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je prišlo do sprememb programske opreme mobilnega validatorja, ki se tičejo procedur branja/zapisa podatkov na brezstično kartico. Test je obvezen.

Test:	<b>KON002</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.12.3 Test kontrole vozovnice z mobilnim validatorjem znotraj območja veljavnosti**

Test je namenjen ugotavljanju ustreznega delovanja mobilnega validatorja v primeru, ko se veljavnost vozovnice preverja na trasi, za katero je vozovnica veljavna.

*Identifikacijska oznaka testa: KON003*

*Scenarij testa*

V testu se uporabi IJPP kartica z že prednastavljenimi podatki o imetniku, zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice. Pred izvedbo testa mora biti na kartico naložena veljavna enkratna vozovnica (test 10.6.9.1) za relacijo A-B, in slednja mora biti validirana na relaciji A-B (test 10.6.10.8). Mobilna validirna enota mora biti konfigurirana za preverjanje ustreznosti vozovnic na relaciji A-B. Ko so vse predhodne procedure izvedene se s paketom COLLIS (čitalnik kartic in pripadajoča programska oprema) pridobi vsebina kartice, ki v nadaljevanju služi kot osnova primerjalnega testa spremembe podatkov na kartici. Naslednji korak testa zajema postavitve IJPP kartice z že validiranim veljavnim produktom vozovnice v NFC polje (označeno mesto) mobilne validirne enote (kontrolor). V medsebojni interakciji mora mobilna validirna enota prebrati podatek o veljavni vozovnici in podatek validacije. Ker je bila vozovnica že validirana se mora podatek o veljavnosti vozovnice izpisati na zaslon mobilnega validatorja, skupaj s preostalimi podatki veljavne vozovnice. Ker je vozovnica veljavna za relacijo A-B, na kateri se izvaja kontrola, mora mobilni validator na zaslon javiti še sporočilo »Veljavna vozovnica za lokacijo preverjanja«. Mobilni validator v tem primeru zapiše podatek o uspešni mobilni validaciji (preverbi) na brezstično kartico IJPP. Ko je procedura izmenjave podatkov med kartico in mobilnim validatorjem zaključena se ponovno s paketom COLLIS pridobi t.i. 'batch' vsebine kartice, ki mora v primerjavi s predhodnim 'batch-em' odražati spremembo v sektorju zapisane izvedene mobilne validacije (kontrole). Vsa ostala vsebina kartice mora ostati v testu nespremenjena. Če so podatki na mobilni enoti pravilno izpisani, in vsebina batchev odraža zgolj spremembo pri zapisu mobilne kontrole (validacije), in so vse procedure zaključene v zahtevanem času se test ovrednoti kot uspešen. V nasprotnem primeru (nepravilni in nepopolni podatki, napačni podatki vozovnice, nepravilno in nepravočasno zaključevanje transakcije, neustrezno beleženje transakcije, neustrezní zapis na kartico, neustrezno ugotavljanje lokacije, zavrnitev veljavne vozovnice na relaciji A-B ipd.) se test ovrednoti kot neuspešen - mobilna validirna enota ne more pridobiti soglasja za integracijo v IJPP sistem.

Ker lahko obstaja v sistemu IJPP več različnih ponudnikov mobilnih validatorjev (kontrolorjev) je test potrebno izvesti na vsaj eni napravi vsakega ponudnika. V kolikor ima en ponudnik mobilnih validatorjev (kontrolorjev) na terenu več različnih modelov, se mora test izvesti na vseh modelih.

#### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske opreme mobilnega validatorja (kontrolorja) testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je prišlo do sprememb programske opreme mobilnega validatorja, ki se tičejo procedur branja/zapisa podatkov na brezstično kartico. Test je obvezen.

Test:	<b>KON003</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.12.4 Test kontrole vozovnice z mobilnim validatorjem izven območja veljavnosti**

Test je namenjen ugotavljanju ustreznega delovanja mobilnega validatorja v primeru, ko se veljavnost vozovnice preverja na trasi, za katero je vozovnica neveljavna. Test povzema realen primer, ko se uporabnik validira na trasi A-B, nato v vmesni točki izstopi in nadaljuje pot na trasi A-C brez vnovične validacije.

*Identifikacijska oznaka testa: KON004*

##### *Scenarij testa*

V testu se uporabi IJPP kartica z že prednastavljenimi podatki o imetniku, zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice. Pred izvedbo testa mora biti na kartico naložena veljavna enkratna vozovnica (test 10.6.9.1) za relacijo A-B, in slednja mora biti validirana na relaciji A-B (test 10.6.10.8). Mobilna validirna enota mora biti konfigurirana za preverjanje ustreznosti vozovnic na relaciji A-C. Ko so vse predhodne procedure izvedene se s paketom COLLIS (čitalnik kartic in pripadajoča programska oprema) pridobi vsebina kartice, ki v nadaljevanju služi kot osnova primerjalnega testa spremembe podatkov na kartici. Naslednji korak testa zajema postavitve IJPP kartice z že validiranim veljavnim produktom vozovnice za relacijo A-B v NFC polje (označeno mesto) mobilne validirne enote (kontrolor) – relacija A-C. V medsebojni interakciji mora mobilna validirna enota prebrati podatek o vozovnici in podatek validacije. Ker je bila vozovnica že validirana na relaciji A-B se mora podatek o neveljavnosti vozovnice na preverjeni lokaciji relacije A-C izpisati na zaslonu mobilnega validatorja, skupaj s preostalimi podatki vozovnice. Ker je vozovnica neveljavna za relacijo A-C, na kateri se izvaja kontrola, mora mobilni validator na zaslon javiti še sporočilo »Neveljavna vozovnica za lokacijo preverjanja«. Mobilni validator v tem primeru zapiše podatek o uspešni mobilni validaciji (uspešna ugotovitev neveljavne vozovnice na relaciji A-C) na brezstično kartico IJPP. Ko je procedura izmenjave podatkov med kartico in mobilnim validatorjem zaključena se ponovno s paketom COLLIS pridobi t.i. 'batch' vsebine kartice, ki mora v primerjavi s predhodnim 'batch-em' odražati spremembo v sektorju zapisane izvedene mobilne validacije (kontrole). Vsa ostala vsebina kartice mora ostati v testu nespremenjena. Če so podatki na mobilni enoti pravilno izpisani, in vsebina batchev odraža zgolj spremembo pri zapisu mobilne kontrole (validacije), in so vse procedure zaključene v zahtevanem času se test ovrednoti kot uspešen. V nasprotnem primeru (nepravilni in nepopolni podatki, napačni podatki vozovnice, nepravilno in nepravočasno zaključevanje transakcije, neustrezno beleženje transakcije, neustreznega zapisa na kartico, neustrezno ugotavljanje lokacije, mobilni validator ne ugotovi neveljavne vozovnice na relaciji A-C ipd.) se test ovrednoti kot neuspešen - mobilna validirna enota ne more pridobiti soglasja za integracijo v IJPP sistem.

Ker lahko obstaja v sistemu IJPP več različnih ponudnikov mobilnih validatorjev (kontrolorjev) je test potrebno izvesti na vsaj eni napravi vsakega ponudnika. V kolikor ima en ponudnik mobilnih validatorjev (kontrolorjev) na terenu več različnih modelov, se mora test izvesti na vseh modelih.

##### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske opreme mobilnega validatorja (kontrolorja) testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je

prišlo do sprememb programske opreme mobilnega validatorja, ki se tičejo procedur branja/zapisa podatkov na brezstično kartico. Test je obvezen.

Test:	<b>KON004</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### 10.6.12.5 Test istovetnosti imetnika IJPP kartice

Test je namenjen ugotavljanju istovetnosti uporabnika na mobilni validirni enoti, kateri razpolaga z IJPP brezstično kartico.

*Identifikacijska oznaka testa: KON005*

##### *Scenarij testa*

V testu se uporabi IJPP kartica z že prednastavljenimi podatki o imetniku (ime: Testni, priimek: Uporabnik), dovoljenjih imetnika (upravičenost), zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice. Pred izvedbo testa mora biti na kartico naložena veljavna enkratna vozovnica (test 10.6.9.1) za relacijo A-B, ki v tem testu ne rabi biti validirana. Mobilna validirna enota mora biti konfigurirana za preverjanje istovetnosti uporabnika. Ko so vse predhodne procedure izvedene se s paketom COLLIS (čitalnik kartic in pripadajoča programska oprema) pridobi vsebina kartice, ki v nadaljevanju služi kot osnova primerjalnega testa spremembe podatkov na kartici. Naslednji korak testa zajema postavitve IJPP kartice z veljavnim produktom vozovnice v NFC polje (označeno mesto) mobilne validirne enote (kontrolor), ki je konfigurirana za preverjanje istovetnosti. V medsebojni interakciji mora mobilna validirna enota prebrati podatek o vozovnici in podatke uporabnika, kateri (Testni Uporabnik) razpolaga z IJPP brezstično kartico. Mobilna validirna enota mora izpisati na zaslon podatke uporabnika (Testni Uporabnik). Mobilni validator v tem primeru zapiše na kartico podatek o uspešni mobilni validaciji (preverjanje istovetnosti) na brezstično kartico IJPP. Ko je procedura izmenjave podatkov med kartico in mobilnim validatorjem zaključena se ponovno s paketom COLLIS pridobi t.i. 'batch' vsebine kartice, ki mora v primerjavi s predhodnim 'batch-em' odražati spremembo v sektorju zapisane izvedene mobilne validacije (preverjanje istovetnosti). Vsa ostala vsebina kartice mora ostati v testu nespremenjena. Če so podatki na mobilni enoti pravilno izpisani, in vsebina batchev odraža zgolj spremembo pri zapisu mobilne kontrole (validacije – preverjanje istovetnosti), in so vse procedure zaključene v zahtevanem času se test ovrednoti kot uspešen. V nasprotnem primeru (nepravilni in nepopolni podatki, napačni podatki vozovnice, nepravilno in nepravočasno zaključevanje transakcije, neustrezno beleženje transakcije, neustrezni zapis na kartico, neustrezni podatki uporabnika ipd.) se test ovrednoti kot neuspešen - mobilna validirna enota ne more pridobiti soglasja za integracijo v IJPP sistem.

Ker lahko obstaja v sistemu IJPP več različnih ponudnikov mobilnih validatorjev (kontrolorjev) je test potrebno izvesti na vsaj eni napravi vsakega ponudnika. V kolikor ima en ponudnik mobilnih validatorjev (kontrolorjev) na terenu več različnih modelov, se mora test izvesti na vseh modelih.

#### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske opreme mobilnega validatorja (kontrolorja) testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je prišlo do sprememb programske opreme mobilnega validatorja, ki se tičejo procedur branja/zapisa podatkov na brezstično kartico. Test je obvezen.

Test:	<b>KON005</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

### **10.6.13 TESTI OBNOVE VOZOVNIC**

Testi obnove vozovnic so namenjeni ugotavljanju ustreznega delovanja prodajnih mest oz. kartomats, ki morajo omogočati podaljševanje neuporabljenih vozovnic. Proces obnove neuporabljene vozovnice skrbi za spremembo datuma veljavnosti obstoječe vozovnice. S testom se preverja ali proces obnove deluje v primeru neuporabljene in uporabljene vozovnice.

#### **10.6.13.1 Test obnove neuporabljene enkratne vozovnice**

Test je namenjen preverjanju ustreznega delovanja kartomata oz. prodajnega mesta, kjer se ugotavlja zmožnost podaljševanja vozovnic, ki še niso bile uporabljene (nevalidirane).

*Identifikacijska oznaka testa: VOZ010*

#### *Scenarij testa*

V testu se uporabi IJPP kartica z že prednastavljenimi podatki o imetniku (ime: Testni, priimek: Uporabnik), dovoljenjih imetnika (upravičenost), zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice. Pred izvedbo testa mora biti na kartico naložena veljavna enkratna vozovnica (test 10.6.9.1) za relacijo A-B, katera za potrebe testa ne sme biti validirana. V testu je uporabljena še periferija prodajnega mesta (kartomat). Za potrebe testa mora biti veljavnost vozovnice čim krajša. Na tem mestu v proceduri testa, ki ga izvajamo, preberemo vsebino kartice s orodjem COLLIS (čitalnik kartic in pripadajoča programska oprema) s čimer se pridobi vsebina kartice pred testom, za potrebe kasnejše primerjave vsebine po testu. Naslednji korak testa izvedemo šele, ko je interval časovne veljavnosti naložene vozovnice že potekel. Ko je ta pogoj

izpolnjen nadaljujemo test obnove vozovnice na kartomatu, kjer se izbere procedura obnove. Test nadaljujemo s prislonitvijo brezstične kartice z naloženo datumsko neveljavno vendar neuporabljeno vozovnico na označeno mesto kartomata (v magnetno polje čitalnika kartomata). Pri proceduri obnove vozovnice kartomat prebere podatke o vozovnici in mora kot prvi pogoj preveriti ali naložena vozovnica res ni bila validirana in je datumsko neveljavna. Kartomat mora v tem testu ugotoviti izpolnjenost obeh pogojev (datumsko neveljaven produkt in hkrati nevalidiran produkt), in avtomatično aplicirati spremembo datuma veljavnosti na brezstično kartico. Po končanem apliciranju novega okvirja veljavnosti, kartomat obvesti uporabnika o uspešni obnovi (»Obnova uspešna.«). Naslednji korak v testni proceduri zajema ponovno branje vsebine kartice po obnovi s paketom COLLIS, kjer se izvede neposredna primerjava z vsebino kartice pred obnovo. Če primerjava vsebine zapisa kartice odraža samo spremembo v sektorju datumске veljavnosti, in kartomat ustrezno ugotovi oba predhodno navedena pogoja se test obravnava kot uspešen. Če kateri izmed pogojev ni izpolnjen (nepravilen zapis na kartico, nepravilen datum, zavrnitev obnove v tem primeru ipd.) se test vrednoti kot neuspešen. V primeru neuspešnega testa se kartomatu ne odobri soglasja za integracijo v IJPP sistem.

Ker lahko obstaja v sistemu IJPP več različnih ponudnikov kartomats (prodajnih mest) je test potrebno izvesti na vsaj eni napravi vsakega ponudnika. V kolikor ima en ponudnik na terenu več različnih modelov kartomats, se mora test izvesti na vseh modelih.

#### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske opreme kartomata testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je prišlo do sprememb programske opreme kartomata, ki se tičejo procedur branja/zapisa podatkov na brezstično kartico. Test se izvede tudi v primeru vsakršne spremembe algoritmov za ugotavljanje pogojev, ki dovoljujejo oz. zavračajo proceduro obnove. Test je obvezen.

Test:	<b>VOZ010</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.13.2 Test preverjanja novega datuma veljavnosti po uspešni obnovi**

Test je namenjen preverjanju ustreznega delovanja kartomata oz. prodajnega mesta, kjer se ugotavlja ustreznost datuma podaljšane (obnovljene) vozovnice. Test je v osnovi nadaljevanje testa 10.6.13.1.

*Identifikacijska oznaka testa: VOZ011*

*Scenarij testa*



Gre za nadaljevanje testa 10.6.13.1 v katerem je že bila vozovnica uspešno obnovljena, in na kartico zapisan novi datum. V tem testu se preveri na novo zapisani datum veljavnosti. Nadaljevanje testa je praktična izvedba testa verifikacije vozovnice na kartomatu (test 10.6.11.1), ki vrne podatke o naloženem produktu (v tem primeru podatke podaljšane enkratne vozovnice – vrsta vozovnice, relacija in datum veljavnosti). Ker je test 10.6.11.1 že definiran in procedura testa specificirana, na tem mestu ne podvajamo specifikacij testa. V kolikor je procedura verifikacije vozovnice iz testa 10.6.11.1 uspešna in rezultati skladni s testom 10.6.11.1, se le ta ovrednoti kot uspešen.

Ker lahko obstaja v sistemu IJPP več različnih ponudnikov kartomatov (prodajnih mest) je test potrebno izvesti na vsaj eni napravi vsakega ponudnika. V kolikor ima en ponudnik na terenu več različnih modelov kartomatov, se mora test izvesti na vseh modelih.

#### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske opreme kartomata testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je prišlo do sprememb programske opreme kartomata, ki se tičejo procedur branja/zapisa podatkov na brezstično kartico. Test se izvede tudi v primeru vsakršne spremembe algoritmov za ugotavljanje pogojev, ki dovoljujejo oz. zavračajo proceduro obnove. Test je obvezen.

Test:	<b>VOZ011</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.13.3 Test obnove uporabljene enkratne vozovnice**

Test je namenjen preverjanju ustreznega delovanja kartomata oz. prodajnega mesta, kjer se ugotavlja zmožnost podaljševanja vozovnic, ki so bile uporabljene (validirane).

*Identifikacijska oznaka testa: VOZ012*

#### *Scenarij testa*

V testu se uporabi IJPP kartica z že prednastavljenimi podatki o imetniku (ime: Testni, priimek: Uporabnik), dovoljenjih imetnika (upravičenost), zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice. Pred izvedbo testa mora biti na kartico naložena veljavna enkratna vozovnica (test 10.6.9.1) za relacijo A-B, katera mora biti za potrebe testa že uspešno validirana (glej test 10.6.10.8). V testu je uporabljena še periferija prodajnega mesta (kartomat). V naslednjem koraku izvedbe testa preberemo vsebino kartice s orodjem COLLIS (čitalnik kartic in pripadajoča programska oprema) s čimer se pridobi vsebina kartice pred testom, za potrebe kasnejše primerjave vsebine po testu. Nadaljevanje testa poteka tako, da se na kartomatu najprej

izbere možnost za podaljševanje vozovnic (obnova). Test nadaljujemo s prislonitvijo brezstične kartice z naloženo datumsko veljavno in uporabljeno (validirano) vozovnico na označeno mesto kartomata (v magnetno polje čitalnika kartomata). Pri proceduri obnove vozovnice kartomat prebere podatke o vozovnici in mora kot prvi pogoj preveriti ali naložena vozovnica res ni bila validirana in če je slednja res datumsko neveljavna. Kartomat mora v tem testu ugotoviti, da je vozovnica že bila validirana in da je slednja tudi še datumsko veljavna. Tudi ostale kombinacije (validirana in datumsko neveljavna, validirana in datumsko veljavna) morajo v testu rezultirati zavrnitev obnove, kar mora kartomat uporabniku signalizirati v obliki naslednjega sporočila (»Obnova neuspešna!«). Naslednji korak v testni proceduri zajema ponovno branje vsebine kartice po proceduri neuspešne obnove, in sicer s paketom COLLIS, kjer se izvede neposredna primerjava z vsebino kartice pred neuspešno obnovo. Če primerjava vsebine zapisa kartice ne odraža sprememb (na kartico niso bili aplicirani nobeni podatki), in kartomat ustrezno ugotovi pogoje za zavrnitev obnove, in so vse procedure zaključene skladno s specifikacijami trajanja transakcije in specifikacijami obveščanja uporabnika, in je transakcija na kartomatu uspešno zabeležena se test vrednoti kot uspešen. V primeru neuspešnega testa (kartomat obnovi vozovnico – neustrezno obravnavani pogoji obnove, se spremeni vsebina kartice – neustrezno vpisovanje podatkov pri testu, ipd.) se test vrednoti kot neuspešen in se skladno s tem kartomatu ne odobri soglasja za integracijo v IJPP sistem.

Ker lahko obstaja v sistemu IJPP več različnih ponudnikov kartomатов (prodajnih mest) je test potrebno izvesti na vsaj eni napravi vsakega ponudnika. V kolikor ima en ponudnik na terenu več različnih modelov kartomатов, se mora test izvesti na vseh modelih.

#### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske opreme kartomata testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je prišlo do sprememb programske opreme kartomata, ki se tičejo procedur branja/zapisa podatkov na brezstično kartico. Test se izvede tudi v primeru vsakršne spremembe algoritmov za ugotavljanje pogojev, ki dovoljujejo oz. zavračajo proceduro obnove. Test je obvezen.

Test:	<b>VOZ012</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.14**      *TESTI REFUNDACIJE (POVRAČILA) ZNESKA VOZOVNICE*

Testi refundacije vozovnic so namenjeni ugotavljanju ustreznega delovanja prodajnih mest oz. kartomатов, ki morajo omogočati refundacije neuporabljenih vozovnic. Ta sklop zajema dva testa, in

sicer test zavrnitve refundacije v primeru, ko je bila vozovnice že uporabljena (validirana) in test odobritve refundacije zneska vozovnice v primeru ko vozovnica še ni bila uporabljena (nevalidirana).

#### **10.6.14.1 Test obnove uporabljene enkratne vozovnice**

Test je namenjen preverjanju ustreznega delovanja kartomata oz. prodajnega mesta, kateri mora refundacijo zavrniti v primeru že uporabljene vozovnice.

*Identifikacijska oznaka testa: VOZ013*

##### *Scenarij testa*

V testu se uporabi IJPP kartica z že prednastavljenimi podatki o imetniku (ime: Testni, priimek: Uporabnik), dovoljenjih imetnika (upravičenost), zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice. Pred izvedbo testa mora biti na kartico naložena veljavna enkratna vozovnica (test 10.6.9.1) za relacijo A-B, katera mora biti za potrebe testa še validirana (uporabljena – test 10.6.10.8). V testu je uporabljena še periferija prodajnega mesta (kartomat). V naslednjem koraku tega testa se prebere vsebina kartice z orodjem COLLIS (čitalnik kartic in pripadajoča programska oprema) s čimer se pridobi vsebina kartice pred testom, za potrebe kasnejše primerjave vsebine po testu. Na prodajnem mestu mora biti za potrebe izvedbe tega testa omogočena podpora refundaciji. Ko je ta pogoj izpolnjen test nadaljujemo tako, da brezstično kartico z naloženo datumsko veljavno in validirano vozovnico prislonimo na označeno mesto kartomata – prodajnega mesta (v magnetno polje čitalnika kartomata). Pri proceduri refundacije zneska vozovnice kartomat prebere podatke o vozovnici in mora kot prvi pogoj preveriti podatke o validaciji naložene enkratne vozovnice. Prodajno mesto mora ugotoviti, da je naložena testna vozovnica že bila validirana in skladno s tem zavrniti refundacijo. Prodajno mesto v tem primeru uporabnika obvesti z izpisanim sporočilom o neupravičeni refundaciji (»Refundacija neupravičena!«). V medsebojni interakciji neuspešne refundacije prodajno mesto ne sme spremeniti podatkov na kartici. Da lahko to v testu potrdimo, izvedemo v naslednjem koraku ponovno preverjanje vsebine kartice s paketom COLLIS. V kolikor je prebrana vsebina kartice pred testom in po testu identična, in je transakcija zaključena v predpisanem intervalu (glej specifikacije transakcij), in je prodajno mesto uspešno zabeležilo transakcijo ter je prodajno mesto v tem primeru s specifikiranim sporočilom zavrnilo refundacijo že validirane vozovnice se test ovrednoti kot uspešen. Če kateri izmed predhodno naštetih kriterijev ni izpolnjen, oz. je prišlo do sprememb v zapisani strukturi kartice po končanem testu, se test obravnava kot neuspešen. Skladno z neuspehom testa se prodajnemu mestu ne izda soglasja za integracijo v IJPP sistem.

Ker lahko obstaja v sistemu IJPP več različnih ponudnikov kartomatov (prodajnih mest) je test potrebno izvesti na vsaj eni napravi vsakega ponudnika. V kolikor ima en ponudnik na terenu več različnih modelov kartomatov, se mora test izvesti na vseh modelih.

##### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske opreme kartomata (prodajnega mesta) testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je prišlo do sprememb programske opreme kartomata, ki se tičejo procedur branja/zapisa podatkov na brezstično kartico. Test se izvede tudi v primeru vsakršne spremembe algoritmov za ugotavljanje pogojev, ki dovoljujejo oz. zavračajo proceduro refundacije (povračila). Test je obvezen.

Test:	VOZ013	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### 10.6.14.2 Test obnove neuporabljene enkratne vozovnice

Test je namenjen preverjanju ustreznega delovanja kartomata oz. prodajnega mesta, kateri mora refundacijo odobriti v primeru še neuporabljene vozovnice.

*Identifikacijska oznaka testa: VOZ014*

##### *Scenarij testa*

V testu se uporabi IJPP kartica z že prednastavljenimi podatki o imetniku (ime: Testni, priimek: Uporabnik), dovoljenjih imetnika (upravičenost), zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice. Pred izvedbo testa mora biti na kartico naložena veljavna enkratna vozovnica (test 10.6.9.1) za relacijo A-B, katera za potrebe testa ne sme biti validirana (status: neuporabljena vozovnica). V testu je uporabljena še periferija prodajnega mesta (kartomat). V naslednjem koraku tega testa se prebere vsebina kartice z orodjem COLLIS (čitalnik kartic in pripadajoča programska oprema) s čimer se pridobi vsebina kartice pred testom, za potrebe kasnejše primerjave vsebine po testu. Na prodajnem mestu mora biti za potrebe izvedbe tega testa omogočena podpora refundaciji. Ko je ta pogoj izpolnjen test nadaljujemo tako, da brezstično kartico z naloženo datumsko veljavno in nevalidirano vozovnico prislonimo na označeno mesto kartomata – prodajnega mesta (v magnetno polje čitalnika kartomata). Pri proceduri refundacije zneska vozovnice kartomat prebere podatke o vozovnici in mora kot prvi pogoj preveriti podatke o morebitni validaciji naložene enkratne vozovnice. Prodajno mesto mora ugotoviti, da testna vozovnica še ni bila validirana in skladno s tem odobriti refundacijo. Prodajno mesto v tem primeru iz kartice izbriše produkt (prva sprememba na kartici), na kartico zapiše podatek o uspešni refundaciji (druga sprememba podatkov kartice), znesek refundacije se povrne imetniku kartice v obliki gotovine. Skladno po končani uspešni interakciji izmenjave podatkov med prodajnim mestom in kartico, kartomat obvesti uporabnika z izpisanim sporočilom o upravičeni refundaciji (»Refundacija uspešna«). Da so vsi omenjeni podatki izvedene refundacije uspešno zapisani na kartico preverimo v naslednjem koraku s paketom COLLIS vsebino kartice po testu. Če medsebojna primerjava podatkov kartice pred/po testu odraža spremembe produkta (izbrisan produkt) in podatek o izvedeni refundaciji, in je transakcija zaključena v predpisanem intervalu (glej specifikacije transakcij), in je prodajno mesto uspešno zabeležilo transakcijo ter je prodajno mesto v tem primeru s specifikiranim sporočilom odobrilo refundacijo nevalidirane vozovnice, se test ovrednoti kot uspešen. Če kateri izmed predhodno naštetih kriterijev ni izpolnjen, oz. je prišlo do napačnih sprememb v zapisani strukturi kartice (glej specifikacije strukture

podatkov IJPP kartice) se test vrednoti kot neuspešen. Skladno z neuspehom testa se prodajnemu mestu ne izda soglasja za integracijo v IJPP sistem.

Ker lahko obstaja v sistemu IJPP več različnih ponudnikov kartomatov (prodajnih mest) je test potrebno izvesti na vsaj eni napravi vsakega ponudnika. V kolikor ima en ponudnik na terenu več različnih modelov kartomatov, se mora test izvesti na vseh modelih.

#### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske opreme kartomata (prodajnega mesta) testa ne opravi. Prav tako mora naprava opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je prišlo do sprememb programske opreme kartomata, ki se tičejo procedur branja/zapisa podatkov na brezstično kartico. Test se izvede tudi v primeru vsakršne spremembe algoritmov za ugotavljanje pogojev, ki dovoljujejo oz. zavračajo proceduro refundacije (povračila). Test je obvezen.

Test:	VOZ014	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.14.3 Test blokade enkratne vozovnice v sistemu**

Test je namenjen preverjanju ustreznega delovanja procedur blokade vozovnice na vseh segmentih sistema IJPP. S testom se preverja, ali se vse komponente sistema na katerih se kartica uporablja (kartomati, prodajna mesta, validatorji, mobilni validatorji,...) odzivajo na enak način in v skladu s tem testom prepoznajo blokirano vozovnico.

*Identifikacijska oznaka testa: VOZ015*

#### *Scenarij testa*

Prvi del tega testa predstavlja test 10.6.11.6, v katerem se ugotavlja ustreznost delovanja mehanizmov upravljanja z vozovnicami na črni listi. S testom 10.6.11.6 se poskrbi, da je brezstična kartica primerno konfigurirana s produktom na črni listi. Zraven brezstične kartice se za nadaljnje izvajanje testa uporabi še naslednja periferija: prodajno mesto, kartomat, validator, mobilni validator (kontrolor), zaledni sistem prevoznika in zaledni sistem upravljavca. Ker je s testom 10.6.11.6 že poskrbljeno, da je enkratna vozovnica na črni listi je potrebno slednjo preizkusiti najprej na kartomatu (preverjanje informacij o produktu), validatorju, mobilnem validatorju in prodajnem mestu samem. Za uspešno izveden test je pogoj, da vse predhodno naštetе periferne enote prepoznajo naložen produkt kot blokirani produkt, hkrati pa se mora informacija o blokirani vozovnici nahajati tako v zalednem sistemu prevoznika, kot tudi zalednem sistemu upravljavca. Če je ta pogoj izpolnjen se test vrednoti kot uspešen. V primeru neustrezne razpoznave blokirane produkta na posamezni periferni enoti

(kartomat, validator, prodajno mesto, mobilni validator) oziroma manjkajočega podatka v zalednem sistemu prevoznika oz. upravljavca se test smatra kot neuspešen. V tem primeru sistem kot celota ne dobi soglasja za operativo v sistemu IJPP. Periferija, ki testa ne opravi, mora test ponavljati, ponudnik periferne enote pa mora zanjo poskrbeti z ustrezno nadgradnjo (programsko, strojno).

Ker lahko obstaja v sistemu IJPP več različnih ponudnikov perifernih enot, je test potrebno izvesti na vsaj eni periferni enoti specifičnega modela in v navezi z vsemi kombinacijami preostale periferije s katero se pojavlja v operativnem sistemu.

#### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske in strojne opreme oz. različne verzije kombinirane programske in strojne opreme različnih proizvajalcev periferije testa ne opravijo. Prav tako mora prodajno mesto opraviti test, ko je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je prišlo do sprememb programske opreme prodajnega mesta, ki se tičejo procedur branja/zapisa podatkov na brezstično kartico. Test se izvede tudi v primeru vsakršnih sprememb algoritmov na zalednem sistemu, ki skrbijo za seznam vozovnic na črni listi. Test je obvezen.

Test:	<b>VOZ015</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.14.4 Test deblokade enkratne vozovnice v sistemu**

Test je namenjen preverjanju ustreznega delovanja procedur deblokade vozovnice na vseh segmentih sistema IJPP. S testom se preverja, ali se vse komponente sistema na katerih se kartica uporablja (kartomati, prodajna mesta, validatorji, mobilni validatorji,...) odzivajo v skladu s tem testom in omogočajo nemoteno uporabo deblokirane vozovnice.

*Identifikacijska oznaka testa: VOZ016*

#### *Scenarij testa*

Predpogoj za izvedbo tega testa je izvedba testa 10.6.11.6 s katerim se vozovnico postavi na črno listo. Ker je procedura predhodno omenjenega testa 10.6.11.6 že definirana in opisana, je v tem segmentu ne obravnavamo podrobneje. Zraven brezstične kartice se za nadaljnje izvajanje testa uporabi še naslednja periferija: prodajno mesto, kartomat, validator, mobilni validator (kontrolor), zaledni sistem prevoznika in zaledni sistem upravljavca. V nadaljevanju testa se blokirano enkratno vozovnico umakne iz črne liste na vseh nivojih. To pomeni, da naprave, ki razpolagajo s črnimi listami (kartomati, prodajna mesta, validatorji, mobilni validatorji i zaledni sistemi) nemudoma posodobijo status črnih list. Ko je procedura posodobitve črnih list izvedena na vseh nivojih, se kartico z blokirano vozovnico preizkusi

na vsej naštetih periferiji, ki mora kot uspešen rezultat testa vozovnico sedaj prepoznati kot veljavno (test 10.6.10.8). Za uspešno izveden test je pogoj, da vse predhodno naštete periferne enote prepoznajo naložen produkt kot sedaj veljaven produkt, hkrati pa se mora informacija o deblokirani vozovnici nahajati tako v zalednem sistemu prevoznika, kot tudi zalednem sistemu upravljavca. Če je ta pogoj izpolnjen se test vrednoti kot uspešen. V primeru neustrezne razpoznave deblokiranega produkta na posamezni periferni enoti (kartomat, validator, prodajno mesto, mobilni validator) oziroma manjkajočega podatka v zalednem sistemu prevoznika oz. upravljavca se test smatra kot neuspešen. V tem primeru sistem kot celota ne dobi soglasja za operativno v sistemu IJPP. Periferija, ki testa ne opravi, mora test ponavljati, ponudnik periferne enote pa mora zanjo poskrbeti z ustrezno nadgradnjo (programsko, strojno).

Ker lahko obstaja v sistemu IJPP več različnih ponudnikov perifernih enot, je test potrebno izvesti na vsaj eni periferni enoti specifičnega modela in v navezi z vsemi kombinacijami preostale periferije s katero se pojavlja v operativnem sistemu.

#### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske in strojne opreme oz. različne verzije kombinirane programske in strojne opreme različnih proizvajalcev periferije testa ne opravijo. Prav tako mora prodajno mesto opraviti test, ko je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je prišlo do sprememb programske opreme prodajnega mesta, ki se tičejo procedur branja/zapisa podatkov na brezstično kartico. Test se izvede tudi v primeru vsakršnih sprememb algoritmov na zalednem sistemu, ki skrbijo za seznam aktivnih vozovnic na črni listi. Test je obvezen.

Test:	<b>VOZ016</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		

#### **10.6.14.5 Test odstranitve enkratne vozovnice iz kartice**

Test je namenjen preverjanju ustreznega delovanja procedur odstranitve enkratne testne vozovnice iz kartice na prodajnem mestu. S testom se preverja, ali se enkratna vozovnica uspešno odstrani iz kartice.

*Identifikacijska oznaka testa: VOZ017*

#### *Scenarij testa*

V testu se uporabi IJPP kartica z že prednastavljenimi podatki o imetniku (ime: Testni, priimek: Uporabnik), dovoljenjih imetnika (upravičenost), zapisanimi diverzificiranimi ključi in ostalimi inicialnimi podatki, kot jih določa standard IJPP kartice. Pred izvedbo testa mora biti na kartico naložen

produkt enkratne vozovnice (test 10.6.9.1) za relacijo A-B. V testu je uporabljena še periferija prodajnega mesta (kartomat). V naslednjem koraku tega testa se najprej prebere vsebina kartice z orodjem COLLIS (čitalnik kartic in pripadajoča programska oprema) s čimer se pridobi vsebina kartice pred testom, za potrebe kasnejše primerjave vsebine po testu (po odstranjenem produktu enkratne vozovnice). Na prodajnem mestu mora biti za potrebe izvedbe tega testa omogočena odstranitev produkta (enkratne vozovnice). Ko je ta pogoj izpolnjen test nadaljujemo tako, da brezstično kartico z naloženim produktom enkratne vozovnice prislonimo na označeno mesto prodajnega mesta. Pri proceduri odstranitve produkta prodajno mesto pobriše vse podatke, ki se tičejo produkta enkratne vozovnice, tako da slednji na kartici več ne obstaja. Skladno po končani uspešni interakciji izmenjave podatkov med prodajnim mestom in kartico, slednje obvesti uporabnika z izpisanim sporočilom o izvedeni proceduri brisanja produkta (»Produkt odstranjen.«). Preverjanje vsebine izbrisanega produkta iz kartice izvedemo s paketom COLLIS, s katerim pridobimo vsebino kartice po izvršenem testu (odstranjen produkt). Če medsebojna primerjava podatkov kartice pred/po testu odraža zgolj spremembe v strukturi zapisa produkta (izbrisan produkt), in je transakcija zaključena v predpisanem intervalu (glej specifikacije transakcij), in je prodajno mesto uspešno zabeležilo transakcijo ter je prodajno mesto v tem primeru s specifičnim sporočilom odstranitev produkta vozovnice, se test ovrednoti kot uspešen. Če kateri izmed predhodno naštetih kriterijev ni izpolnjen, oz. je prišlo do napačnih sprememb v zapisani strukturi kartice (glej specifikacije strukture podatkov IJPP kartice) se test vrednoti kot neuspešen. Skladno z neuspehom testa se prodajnemu mestu ne izda soglasja za integracijo v IJPP sistem.

Ker lahko obstaja v sistemu IJPP več različnih ponudnikov prodajnih mest je test potrebno izvesti na vsaj eni napravi vsakega ponudnika. V kolikor ima en ponudnik na terenu več različnih modelov prodajnih mest, se mora test izvesti na vseh modelih.

#### *Kdaj izvesti test*

Test je potrebno izvajati dokler specifična verzija izdane programske in strojne opreme prodajnega mesta testa ne opravi. Prav tako mora sklop naprav, ki sestavljajo prodajno mesto opraviti test, kadar je prišlo do sprememb na segmentu algoritmov oz. spremembi strukture zapisa vsebine na IJPP kartico, in tudi v primerih, ko je prišlo do sprememb programske opreme prodajnega mesta, ki se tičejo procedur branja/zapisa podatkov (brisanje produkta) na brezstično kartico. Test se izvede tudi v primeru vsakršne spremembe algoritmov za ugotavljanje pogojev, ki dovoljujejo oz. zavračajo proceduro brisanja produkta(ov). Test je obvezen.

Test:	<b>VOZ017</b>	
Tester: (Ime, Priimek)		
Datum testa (DD.MM.YYYY)		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>
Ugotovitve:		



## 10.7 TESTI METOD SPLETNE STORITVE IJPP

Integraciji zalednega sistema s sistemi, ki jih uporabljajo naši partnerji bo v veliki meri potekala preko metod spletnih storitev IJPP. Metode spletne storitve IJPP nam omogočajo pridobitev podatkov iz zalednega sistema IJPP, prav tako pa nam omogočajo prenos podatkov v zaledni sistem IJPP. Za brezhibno delovanje komunikacije med zalednim sistemom IJPP in sistemi, ki jih uporabljajo partnerji je potrebno brezhibno delovanje metod spletne storitve IJPP. V nadaljevanju smo definirali testne scenariječitalnika za vsako metodo spletne storitve IJPP. Teste metod spletnih storitev IJPP je potrebno izvesti ob začetku uporabe le-teh s čimer se preverja ustreznost komunikacije med metodami spletnih storitev IJPP in programsko opremo. Prav tako se preverja, ali metode vračajo rezultate v skladu s podanimi deklaracijami oz. zahtevami, ki so bile podane v specifikaciji.

### 10.7.1 TESTI METODE IDENTIFYTRANSACTIONPARAMETERS

#### Scenarij testa 1

Pri testiranju metode preverimo če le ta pravilno sprejema vhodne parametre in vrne predviden rezultat pri izhodnih parametrih. Test metode poteka tako, da najprej kot vhodni podatek pošljemo Id terminala, ki ne obstaja v sistemu. V tem primeru pošljemo za TerminalId = **0**. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = **-501** ; sResultDescription = **Terminal not found**; mIdentifyData = **null**

Test:	1 (IdentifyTransactionParameters)	
Tester:		
Datum testa:		
Scenarij:	<b>Scenarij testa 1</b> <b>Vhodni parametri:</b> TerminalId = 0;	
Pričakovan rezultat:	iResultCode = <b>-501</b> ; sResultDescription = <b>Terminal not found</b> ; mIdentifyData = <b>null</b> ;	
Dobljen rezultat:		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>

#### Scenarij testa 2

Test metode poteka tako, da kot vhodni podatek pošljemo Id terminala ki obstaja v sistemu. V tem primeru pošljemo za TerminalId = **7xxx**. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi : iResultCode = **0**; sResultDescription = **OK**; mIdentifyData = struktura{iTarifIds; iBlacklistId; iBlacklistId; iLineListId; iTarifClassListId }

### 10.7.2 TESTI METODE GETBLACKLIST

#### Scenarij testa 1

Pri tem testu se preverja ali metoda pravilno sprejema vhodne parametre in vrne predviden rezultat v izhodnih parametrih. Metoda sprejema en obvezni in en opsijski vhodni parameter. Pri prvem scenariju preverimo, če metoda pravilno obravnava terminal Id, ki ne obstaja v sistemu. V tem primeru pošljemo za TerminalId = **0**; iLastBlackListId = **null** . Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = **-501**; sResultDescription = **Terminal not found**; mBlackListData = **null** .

#### Scenarij testa 2

Pri tem scenariju se preverja ali metoda pravilno sprejme terminal ID, ki obstaja v sistemu in vrne celotni seznam črne liste. V tem primeru pošljemo za TerminalId = **7xxx**; iLastBlackListId = **null**. Metoda mora vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = **0**; sResultDescription = **OK**; mBlackListData = struktura{ **ICuid, iBlacklistId, iBlackListCRC** }.

Test:	2 (GetBlackList)	
Tester:		
Datum testa:		
Scenarij:	<b>Scenarij testa 2</b> <b>Vhodni parametri:</b> TerminalId = <b>7xxx</b> ;	
Pričakovan rezultat:	iResultCode = <b>0</b> ; sResultDescription = <b>OK</b> ; mBlackListData = struktura{ <b>ICuid, iBlacklistId, iBlackListCRC</b> };	
Dobljen rezultat:		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>

### 10.7.3 TESTI METODE GETCARDSTATUS

#### Scenarij testa 1

Pri tem scenariju preverimo če metoda pravilno obravnava vhodne parametre in vrne predviden rezultat pri izhodnih parametrih. Pri tem scenariju pošljemo kot vhodne parametre obstoječi terminalID = **7xxx** in neobstoječi CUID = **xxxxxy** kartice. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = **-713** ; sResultDescription = **Card not found** ; ICardStatus = **null** .

#### Scenarij testa 2

Pri tem scenariju testa preverimo ali metoda vrne pravilen rezultat, če preko vhodnih parametrov pošljemo obstoječi terminalID = **7xxx** in obstoječo CUID = **xxxxxx** kartice. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = **0** ; sResultDescription = **OK**; ICardStatus = **1** (vrednost parametra ICardStatus je odvisna od statusa na kartici).

Test:	3 (GetCardStatus)	
Tester:		
Datum testa:		
Scenarij:	<b>Scenarij testa 2</b> <b>Vhodni parametri:</b> TerminalId = <b>7xxx</b> ; CUID = <b>xxxxxx</b>	
Pričakovan rezultat:	iResultCode = <b>0</b> ; sResultDescription = <b>OK</b> ; ICardStatus = <b>1</b> ; (odvisno od statusa kartice)	
Dobljen rezultat:		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>

#### 10.7.4 TESTI METODE SETNEWCARDSTATUS

##### Scenarij testa 1

Pri tem scenariju se preverja ali metoda pravilno obravnava vhodne parametre in vrne pravilen rezultat pri izhodnih parametrih. Kot vhodne parametre pošljemo obstoječi terminalID = **7xxx**, obstoječi CUID = **xxxxxx** kartice in neveljaven status kartice iCardStatus = **-1**. Metoda mora vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = **-714**; sResultDescription = **Invalid card status**; iOldCardStatus = **null**.

##### Scenarij testa 2

Pri tem scenariju se preverja ali metoda vrne na izhod pravilen rezultat, če smo vnesli pravilne podatke. Kot vhodne parametre pošljemo obstoječi terminalID = **7xxx**, obstoječi CUID = **xxxxxx** kartice in veljaven status kartice iCardStatus = **1**. Metoda mora vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = **0**; sResultDescription = **OK**; iOldCardStatus = **0**.

Test:	4 (SetNewCardStatus)	
Tester:		
Datum testa:		
Scenarij:	<b>Scenarij testa 2</b> <b>Vhodni parametri:</b> TerminalId = <b>7xxx</b> ; CUID = <b>xxxxxx</b> iCardStatus = <b>1</b> ;	
Pričakovan rezultat:	iResultCode = <b>0</b> ; sResultDescription = <b>OK</b> ; iOldCardStatus = <b>0</b> ; (odvisno od statusa kartice)	
Dobljen rezultat:		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>

### 10.7.5 TESTI METODE GETEXECUTABLELIST

#### Scenarij testa 1

Pri tem scenariju se preverja ali metoda pravilno obravnava vhodne parametre in vrne pravilen rezultat pri izhodnih parametrih. Kot vhodni parameter pošljemo neobstoječi terminalID = **0**, iCompactType = **0** in lCuid = **null**. Metoda mora vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = **-501**; sResultDescription = **Terminal not found**; mExecutableData = **null**.

#### Scenarij testa 2

Pri tem scenariju preverimo če metoda vrne na izhodu pravilen rezultat, če smo vnesli pravilne podatke. Kot vhodne parametre pošljemo obstoječi terminalID = **7xxx** ternaslednje parametre iCompactType = **1** in lCuid = **xxxx**. Metoda mora vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = **0**; sResultDescription = **OK**; mExecutableData = struktura{**lCuid, iAction, mAdditionalCardObject** }.

Test:	5 (GetExecutableList)	
Tester:		
Datum testa:		
Scenarij:	<b>Scenarij testa 2</b> <b>Vhodni parametri:</b> TerminalId = <b>7xxx</b> ; iCompactType = <b>1</b> ; lCuid = <b>xxxxx</b> ;	
Pričakovan rezultat:	iResultCode = <b>0</b> ; sResultDescription = <b>OK</b> ; mExecutableData= struktura{ <b>lCuid, iAction, mAdditionalCardObject</b> };	
Dobljen rezultat:		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>

### 10.7.6 TESTI METODE GETTARIFF

#### Scenarij testa 1

Pri tem scenariju se preverja ali metoda pravilno obravnava vhodne parametre in vrne pravilen rezultat pri izhodnih parametrih. Kot vhodni parameter pošljemo neobstoječi terminalID = **0** in iTariffIds = **null**. Metoda mora vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = **-501**; sResultDescription = **Terminal not found**; mTariffData = **null**.

#### Scenarij testa 2

Pri tem scenariju se preverja ali metoda vrne na izhodu pravilen rezultat, če smo vnesli pravilne podatke. Kot vhodne parametre pošljemo obstoječi terminalID = **7xxx** in iTariffIds = **null**. Metoda mora vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = **0**; sResultDescription = **OK**; mTariffData = struktura{ **iTariffFormat**, **mServiceProviderTariffData**}.

Test:	6 (GetTariff)	
Tester:		
Datum testa:		
Scenarij:	<b>Scenarij testa 2</b> <b>Vhodni parametri:</b> TerminalId = <b>7xxx</b> ;	
Pričakovan rezultat:	iResultCode = <b>0</b> ; sResultDescription = <b>OK</b> ; mTariffData =struktura{ <b>iTariffFormat</b> , <b>mServiceProviderTariffData</b> }.	
Dobljen rezultat:		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>

### 10.7.7 TESTI METODE GETTARIFFCLASSTRANSITIONMATRIX

#### Scenarij testa 1

Pri tem scenariju se preverja ali metoda pravilno obravnava vhodne parametre in vrne pričakovan rezultat pri izhodnih parametrih. Kot vhodni parameter pošljemo neobstoječi terminalID = **0**. Metoda mora vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = **-501**; sResultDescription = **Terminal not found**; mTariffClassTransitionData = **null**.

#### Scenarij testa 2

Pri tem scenariju se preverja ali metoda vrne na izhodu pravilen rezultat, če smo vnesli pravilne podatke. Kot vhodne parametre pošljemo obstoječi terminalID = **7xxx**. Metoda mora vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = **0**; sResultDescription = **OK**; mTariffClassTransitionData = struktura{**TariffLocationData**, **LocationTransitionMatrixData**, **iTariffClassTransitionsId** }

Test:	7 (GetTariffClassTransitionMatrix)	
Tester:		
Datum testa:		
Scenarij:	<b>Scenarij testa 2</b> <b>Vhodni parametri:</b> TerminalId = <b>7xxx</b> ;	
Pričakovan rezultat:	iResultCode = <b>0</b> ; sResultDescription = <b>OK</b> ; mTariffClassTransitionData=struktura{ <b>TariffLocationData</b> , <b>LocationTransitionMatrixData</b> , <b>iTariffClassTransitionsId</b> }.	
Dobljen rezultat:		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>

### 10.7.8 TESTI METODE PROCESSTRANSACTIONS

#### Scenarij testa 1

Pri tem scenariju se preverja ali metoda pravilno obravnava vhodne parametre in vrne pravilen rezultat pri izhodnih parametrih. Kot vhodni parameter pošljemo prazno spremenljivko mIJPPTransactionData = **null**. Metoda mora vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = **-732**; sResultDescription = **Missing input data**;

#### Scenarij testa 2

Pri tem scenariju se preverja ali metoda vrne na izhodu pravilen rezultat, če smo vnesli pravilne podatke. Kot vhodni parameter pošljemo spremenljivko mIJPPTransactionData katere struktura vsebuje več parametrov, za vse obvezne parametre vnesemo veljavne vrednosti. Če so podatki v pravilni obliki mora metoda vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = **0**; sResultDescription = **OK** (pomeni da so bile transakcije uspešno sprejete).

Test:	8 (ProcessTransactions)	
Tester:		
Datum testa:		
Scenarij:	<b>Scenarij testa 2</b> <b>Vhodni parametri:</b> mIJPPTransactionData = struktura{ ITransactionExecutionId, iTransactionType, dtStartTime , ICUID,...}	
Pričakovan rezultat:	iResultCode = <b>0</b> ; sResultDescription = <b>OK</b> ;	
Dobljen rezultat:		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>



## 10.8 VERIFIKACIJA METOD SPLETNE STORITVE IJPP - ADMINISTRACIJA

### 10.8.1 TESTI METODE SETPASSENGER

#### Scenarij testa 1

Pri tem scenariju preverimo, če metoda pravilno obravnava vhodne parametre in vrne pravilen rezultat pri izhodnih parametrih. Kot vhodni parameter pošljemo prazno spremenljivko `mPassengerData = null`. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: `iResultCode = -732` ; `sResultDescription = Missing input data`;

#### Scenarij testa 2

Pri tem scenariju preverimo če metoda vrne na izhodu pravilen rezultat, če smo vnesli pravilne podatke. Kot vhodni parameter pošljemo spremenljivko `mPassengerData` katere struktura vsebuje več parametrov, za vse obvezne parametre vnesemo veljavne vrednosti. Če so podatki v pravilni obliki nam mora metoda vrniti izhodne parametre z naslednjimi vrednostmi: `iResultCode = 0` ; `sResultDescription = OK` . (Kar pomeni da so bili podatki za potnika uspešno vneseni v zaledni sistem.)

Test:	1 (SetPassenger)	
Tester:		
Datum testa:		
Scenarij:	<b>Scenarij testa 2</b> <b>Vhodni parametri:</b> <code>mPassengerData = struktura { sFirstName, sLastName, sAddress,...}</code>	
Pričakovan rezultat:	<code>iResultCode = 0</code> ; <code>sResultDescription = OK</code> ;	
Dobljen rezultat:		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>

### 10.8.2 TESTI METODE GETPASSENGER

#### Scenarij testa 1

Pri tem scenariju preverimo če metoda pravilno obravnava vhodne parametre in vrne pričakovan rezultat pri izhodnih parametrih. Kot vhodni parameter pošljemo neobstoječi iPid (identifikator potnika) = **0**. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = -**716**; sResultDescription = **Passenger not found.**;

#### Scenarij testa 2

Pri tem scenariju preverimo če metoda vrne na izhodu pravilen rezultat, če smo vnesli pravilne podatke. Kot vhodni parameter pošljemo obstoječi iPid (identifikator potnika) = **1x**. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = **0** ; sResultDescription = **OK**; mPassengerData = struktura { **sFirstName, sLastName, sAddress,...**};

Test:	2 (GetPassenger)	
Tester:		
Datum testa:		
Scenarij:	<b>Scenarij testa 2</b> <b>Vhodni parametri:</b> Pid = <b>1x</b> ;	
Pričakovan rezultat:	iResultCode = <b>0</b> ; sResultDescription = <b>OK</b> ; mPassengerData = struktura { <b>sFirstName, sLastName, sAddress,...</b> };	
Dobljen rezultat:		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>

### 10.8.3 TESTI METODE SETCARD

#### Scenarij testa 1

Pri tem scenariju preverimo, če metoda pravilno obravnava vhodne parametre in vrne pravilen rezultat pri izhodnih parametrih. Kot vhodni parameter pošljemo prazno spremenljivko `mCardData = null`. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: `iResultCode = -732` ; `sResultDescription = Missing input data`;

#### Scenarij testa 2

Pri tem scenariju preverimo če metoda vrne na izhodu pravilen rezultat, če smo vnesli pravilne podatke. Kot vhodni parameter pošljemo spremenljivko `mCardData` katere struktura vsebuje več parametrov, za vse obvezne parametre vnesemo veljavne vrednosti. Če so podatki v pravilni obliki nam mora metoda vrniti izhodne parametre z naslednjimi vrednostmi: `iResultCode = 0` ; `sResultDescription = OK` . (Kar pomeni da so bili podatki za kartico uspešno vneseni v zaledni sistem.)

Test:	3 (SetCard)	
Tester:		
Datum testa:		
Scenarij:	<b>Scenarij testa 2</b> <b>Vhodni parametri:</b> <code>mCardData = struktura { ICUID,dtInitDate, dtValidToDate,...}</code>	
Pričakovan rezultat:	<code>iResultCode = 0</code> ; <code>sResultDescription = OK</code> ;	
Dobljen rezultat:		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>

#### 10.8.4 TESTI METODE UPDATECARD

##### Scenarij testa 1

Pri tem scenariju preverimo, če metoda pravilno obravnava vhodne parametre in vrne pravilen rezultat pri izhodnih parametrih. Kot vhodni parameter pošljemo prazno spremenljivko `mCardData = null`. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: `iResultCode = -732` ; `sResultDescription = Missing input data`;

##### Scenarij testa 2

Pri tem scenariju preverimo če metoda vrne na izhodu pravilen rezultat, če smo vnesli pravilne podatke. Kot vhodni parameter pošljemo spremenljivko `mCardData` katere struktura vsebuje več parametrov, za vse obvezne parametre vnesemo veljavne vrednosti. Če so podatki v pravilni obliki nam mora metoda vrniti izhodne parametre z naslednjimi vrednostmi: `iResultCode = 0` ; `sResultDescription = OK` . (Kar pomeni da so bili podatki za kartico uspešno posodobljeni v zalednem sistemu.)

Test:	4 (UpdateCard)	
Tester:		
Datum testa:		
Scenarij:	<b>Scenarij testa 2</b> <b>Vhodni parametri:</b> <code>mCardData = struktura { ICUID,dtInitDate, dtValidToDate,...}</code>	
Pričakovan rezultat:	<code>iResultCode = 0</code> ; <code>sResultDescription = OK</code> ;	
Dobljen rezultat:		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>

### 10.8.5 TESTI METODE GETCARD

#### Scenarij testa 1

Pri tem scenariju preverimo če metoda pravilno obravnava vhodne parametre in vrne pričakovan rezultat pri izhodnih parametrih. Kot vhodni parameter pošljemo neobstoječi ICUID (identifikator kartice) = **0**. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: **iResultCode = -713**; **sResultDescription = Card not found.**;

#### Scenarij testa 2

Pri tem scenariju preverimo če metoda vrne na izhodu pravilen rezultat, če smo vnesli pravilne podatke. Kot vhodni parameter pošljemo obstoječi ICUID (identifikator kartice) = **7xxxx**. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: **iResultCode = 0** ; **sResultDescription = OK**; **mCardData = struktura { ICUID,dtInitDate, dtValidToDate,...}**;

Test:	5 (GetCard)	
Tester:		
Datum testa:		
Scenarij:	<b>Scenarij testa 2</b> <b>Vhodni parametri:</b> ICUID = 7xxxx;	
Pričakovan rezultat:	iResultCode = 0 ; sResultDescription = OK; mCardData = struktura { ICUID,dtInitDate, dtValidToDate,...};	
Dobljen rezultat:		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>

### 10.8.6 TESTI METODE UPDATECARDPASSENGERSTATUS

#### Scenarij testa 1

Pri tem scenariju preverimo, če metoda pravilno obravnava vhodne parametre in vrne pravilen rezultat pri izhodnih parametrih. Kot vhodni parameter pošljemo prazno spremenljivko `mCardStatusData = null`. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: `iResultCode = -732` ; `sResultDescription = Missing input data`;

#### Scenarij testa 2

Pri tem scenariju preverimo če metoda vrne na izhodu pravilen rezultat, če smo vnesli pravilne podatke. Kot vhodni parameter pošljemo spremenljivko `mCardStatusData` katere struktura vsebuje več parametrov, za vse obvezne parametre vnesemo veljavne vrednosti. Če so podatki v pravilni obliki nam mora metoda vrniti izhodne parametre z naslednjimi vrednostmi: `iResultCode = 0` ; `sResultDescription = OK` . (Kar pomeni da so bile pravice za potnika uspešno vnesene/posodobljene v zalednem sistemu.)

Test:	6 (UpdateCardPassengerStatus)	
Tester:		
Datum testa:		
Scenarij:	<b>Scenarij testa 2</b> <b>Vhodni parametri:</b> <code>mCardStatusData = struktura { ICUID, iTicketStatusID, dtTicketStatusStartDate ,...}</code>	
Pričakovan rezultat:	<code>iResultCode = 0</code> ; <code>sResultDescription = OK</code> ;	
Dobljen rezultat:		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>

### 10.8.7 TESTI METODE *GETCARDPASSENGERSTATUS*

#### Scenarij testa 1

Pri tem scenariju preverimo če metoda pravilno obravnava vhodne parametre in vrne pričakovan rezultat pri izhodnih parametrih. Kot vhodni parameter pošljemo neobstoječi ICUID (identifikator kartice) = **0**. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: **iResultCode = -713**; **sResultDescription = Card not found.**;

#### Scenarij testa 2

Pri tem scenariju preverimo če metoda vrne na izhodu pravilen rezultat, če smo vnesli pravilne podatke. Kot vhodni parameter pošljemo obstoječi ICUID (identifikator kartice) = **7xxxx**. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: **iResultCode = 0** ; **sResultDescription = OK**; **mCardStatusData = struktura { ICUID, iTicketStatusID, dtTicketStatusStartDate ,...}**;

Test:	7 (GetCardPassengerStatus)	
Tester:		
Datum testa:		
Scenarij:	<b>Scenarij testa 2</b> <b>Vhodni parametri:</b> ICUID = 7xxxx;	
Pričakovan rezultat:	iResultCode = 0 ; sResultDescription = OK; mCardStatusData = struktura { ICUID, iTicketStatusID, dtTicketStatusStartDate ,...};	
Dobljen rezultat:		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>

### 10.8.8 TESTI METODE GETRELATION

#### Scenarij testa 1

Pri tem scenariju preverimo če metoda pravilno obravnava vhodne parametre in vrne pričakovan rezultat pri izhodnih parametrih. Kot vhodna parametra pošljemo neobstoječi imeni začetne in končne postaje sStartStation = **xy..** in sEndStation = **zy..**. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = **-736**; sResultDescription = **Relation not found.**;

#### Scenarij testa 2

Pri tem scenariju preverimo če metoda vrne na izhodu pravilen rezultat, če smo vnesli pravilne podatke. Kot vhodna parametra pošljemo obstoječi imeni začetne in končne postaje sStartStation = **axy..** in sEndStation = **bzy...**. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = **0** ; sResultDescription = **OK**; mRelationData = struktura {sStartStationName, sEndStationName, iRelationID,...};

Test:	8 (GetRelation)	
Tester:		
Datum testa:		
Scenarij:	<b>Scenarij testa 2</b> <b>Vhodni parametri:</b> sStartStation = <b>axy..</b> ; sEndStation = <b>bzy..</b> ;	
Pričakovan rezultat:	iResultCode = <b>0</b> ; sResultDescription = <b>OK</b> ; mRelationData = struktura {sStartStationName, sEndStationName, iRelationID,...};	
Dobljen rezultat:		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>



### 10.8.9 TESTI METODE *GE*CARDPRODUCTS

#### Scenarij testa 1

Pri tem scenariju preverimo če metoda pravilno obravnava vhodne parametre in vrne pričakovan rezultat pri izhodnih parametrih. Kot vhodni parameter pošljemo neobstoječi ICUID (identifikator potnikove kartice) = **0**. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: **iResultCode = -713**; **sResultDescription = Card not found.**;

#### Scenarij testa 2

Pri tem scenariju preverimo če metoda vrne na izhodu pravilen rezultat, če smo vnesli pravilne podatke. Kot vhodni parameter pošljemo obstoječi ICUID (identifikator potnikove kartice) = **7xxxx**. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: **iResultCode = 0** ; **sResultDescription = OK**; **mCardProductData = struktura { ICUID, bAction, lProductPurchaseId,..}**

Test:	10 (GetCardProducts)	
Tester:		
Datum testa:		
Scenarij:	<b>Scenarij testa 2</b> <b>Vhodni parametri:</b> ICUID = 7xxxx;	
Pričakovan rezultat:	iResultCode = 0 ; sResultDescription = OK; mCardProductData = struktura { ICUID, bAction,lProductPurchaseId,..};	
Dobljen rezultat:		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>

### 10.8.10 TESTI METODE GETPRODUCTPRICE

#### Scenarij testa 1

Pri tem scenariju preverimo, če metoda pravilno obravnava vhodne parametre in vrne pravilen rezultat pri izhodnih parametrih. Kot vhodni parameter pošljemo prazno spremenljivko `mProductPriceIn = null`. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: `iResultCode = -732` ; `sResultDescription = Missing input data`;

#### Scenarij testa 2

Pri tem scenariju preverimo če metoda vrne na izhodu pravilen rezultat, če smo vnesli pravilne podatke. Kot vhodni parameter pošljemo spremenljivko `mProductPriceIn` katere struktura vsebuje več parametrov, za vse obvezne parametre vnesemo veljavne vrednosti. Če so podatki v pravilni obliki nam mora metoda vrniti izhodne parametre z naslednjimi vrednostmi: `iResultCode = 0` ; `sResultDescription = OK` . (Kar pomeni da je bila uspešno pridobljena cena produkta.) `mProductPriceOut = struktura { iTariffId, fPriceValue}`

Test:	11 (GetProductPrice)	
Tester:		
Datum testa:		
Scenarij:	<b>Scenarij testa 2</b> <b>Vhodni parametri:</b> <code>mProductPriceIn = struktura{ iProductId, iTariffClassId , iTicketStatus,.. };</code>	
Pričakovan rezultat:	<code>iResultCode = 0</code> ; <code>sResultDescription = OK</code> ; <code>mProductPriceOut = struktura { iTariffId, fPriceValue};</code>	
Dobljen rezultat:		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>

### 10.8.11 TESTI METODE SETPOS

#### Scenarij testa 1

Pri tem scenariju preverimo, če metoda pravilno obravnava vhodne parametre in vrne pravilen rezultat pri izhodnih parametrih. Kot vhodni parameter pošljemo prazno spremenljivko `mPosData = null`. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: `iResultCode = -732` ; `sResultDescription = Missing input data`;

#### Scenarij testa 2

Pri tem scenariju preverimo če metoda vrne na izhodu pravilen rezultat, če smo vnesli pravilne podatke. Kot vhodni parameter pošljemo spremenljivko `mPosData` katere struktura vsebuje več parametrov, za vse obvezne parametre vnesemo veljavne vrednosti. Če so podatki v pravilni obliki nam mora metoda vrniti izhodne parametre z naslednjimi vrednostmi: `iResultCode = 0` ; `sResultDescription = OK` . (Kar pomeni da so bili podatki za prodajno mesto uspešno vneseni v zaledni sistem.)

Test:	12 (SetPos)	
Tester:		
Datum testa:		
Scenarij:	<b>Scenarij testa 2</b> <b>Vhodni parametri:</b> <code>mPosData = struktura { iPosId, iCompanyId, sName,...}</code>	
Pričakovan rezultat:	<code>iResultCode = 0</code> ; <code>sResultDescription = OK</code> ;	
Dobljen rezultat:		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>

### 10.8.12 TESTI METODE GETPOS

#### Scenarij testa 1

Pri tem scenariju preverimo če metoda pravilno obravnava vhodne parametre in vrne pričakovan rezultat pri izhodnih parametrih. Kot vhodni parameter pošljemo neobstoječi iPosId (identifikator prodajnega mesta) = **0**. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = **-717**; sResultDescription = **Pos not found.**;

#### Scenarij testa 2

Pri tem scenariju preverimo če metoda vrne na izhodu pravilen rezultat, če smo vnesli pravilne podatke. Kot vhodni parameter pošljemo obstoječi iPosId (identifikator prodajnega mesta) = **1x**. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = **0**; sResultDescription = **OK**; mPosData = struktura { **iPosId, iCompanyId, sName,...**}

Test:	13 (GetPos)	
Tester:		
Datum testa:		
Scenarij:	<b>Scenarij testa 2</b> <b>Vhodni parametri:</b> iPosId = <b>1x</b> ;	
Pričakovan rezultat:	iResultCode = <b>0</b> ; sResultDescription = <b>OK</b> ; mPosData = struktura { <b>iPosId, iCompanyId, sName,...</b> };	
Dobljen rezultat:		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>

### 10.8.13 TESTI METODE SETTERMINAL

#### Scenarij testa 1

Pri tem scenariju preverimo, če metoda pravilno obravnava vhodne parametre in vrne pravilen rezultat pri izhodnih parametrih. Kot vhodni parameter pošljemo prazno spremenljivko `mTerminalData = null`. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: `iResultCode = -732` ; `sResultDescription = Missing input data`;

#### Scenarij testa 2

Pri tem scenariju preverimo če metoda vrne na izhodu pravilen rezultat, če smo vnesli pravilne podatke. Kot vhodni parameter pošljemo spremenljivko `mTerminalData` katere struktura vsebuje več parametrov, za vse obvezne parametre vnesemo veljavne vrednosti. Če so podatki v pravilni obliki nam mora metoda vrniti izhodne parametre z naslednjimi vrednostmi: `iResultCode = 0` ; `sResultDescription = OK` . (Kar pomeni da so bili podatki terminala uspešno vneseni v zaledni sistem.)

Test:	14 (SetTerminal)	
Tester:		
Datum testa:		
Scenarij:	<b>Scenarij testa 2</b> <b>Vhodni parametri:</b> <code>mTerminalData = struktura { iId, sSerialNumber, iTerminalTypeId,...}</code>	
Pričakovan rezultat:	<code>iResultCode = 0</code> ; <code>sResultDescription = OK</code> ;	
Dobljen rezultat:		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>

### 10.8.14 TESTI METODE GETTERMINAL

#### Scenarij testa 1

Pri tem scenariju preverimo če metoda pravilno obravnava vhodne parametre in vrne pričakovan rezultat pri izhodnih parametrih. Kot vhodni parameter pošljemo neobstoječi iTId (identifikator terminala) = **0**. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = -**501**; sResultDescription = **Terminal not found.**;

#### Scenarij testa 2

Pri tem scenariju preverimo če metoda vrne na izhodu pravilen rezultat, če smo vnesli pravilne podatke. Kot vhodni parameter pošljemo obstoječi iTId (identifikator terminala) = **1x**. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = **0**; sResultDescription = **OK**; mTerminalData = struktura { iTId, sSerialNumber, iTerminalTypeId,...}

Test:	15 (GetTerminal)	
Tester:		
Datum testa:		
Scenarij:	<b>Scenarij testa 2</b> <b>Vhodni parametri:</b> iTId = 1x;	
Pričakovan rezultat:	iResultCode = <b>0</b> ; sResultDescription = <b>OK</b> ; mTerminalData = struktura { iTId, sSerialNumber, iTerminalTypeId,...};	
Dobljen rezultat:		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>

### 10.8.15 TESTI METODE SETBLACKLIST

#### Scenarij testa 1

Pri tem scenariju preverimo če metoda pravilno obravnava vhodne parametre in vrne pričakovan rezultat pri izhodnih parametrih. Kot vhodni parameter pošljemo neobstoječi ICUID (identifikator potnikove kartice) = **0**. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = **-713**; sResultDescription = **Card not found.**;

#### Scenarij testa 2

Pri tem scenariju preverimo če metoda vrne na izhodu pravilen rezultat, če smo vnesli pravilne podatke. Kot vhodni parameter pošljemo obstoječi ICUID (identifikator kartice) = **7xxxx**. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: iResultCode = **0** ; sResultDescription = **OK**; iBlackListId = **1xx** ;(Identifikator zadnje aktivne verzije črne liste.)

Test:	16 (SetBlackList)	
Tester:		
Datum testa:		
Scenarij:	<b>Scenarij testa 2</b> <b>Vhodni parametri:</b> ICUID = <b>7xxxx</b> ;	
Pričakovan rezultat:	iResultCode = <b>0</b> ; sResultDescription = <b>OK</b> ; iBlackListId = <b>1xx</b> ;	
Dobljen rezultat:		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>

### 10.8.16 TESTI METODE SETEXECUTABLELIST

#### Scenarij testa 1

Pri tem scenariju preverimo, če metoda pravilno obravnava vhodne parametre in vrne pravilen rezultat pri izhodnih parametrih. Kot vhodni parameter pošljemo prazno spremenljivko `mExecutableData = null`. Metoda nam mora vrniti izhodne parametre z naslednjimi vrednostmi: `iResultCode = -732` ; `sResultDescription = Missing input data`;

#### Scenarij testa 2

Pri tem scenariju preverimo če metoda vrne na izhodu pravilen rezultat, če smo vnesli pravilne podatke. Kot vhodni parameter pošljemo spremenljivko `mExecutableData` katere struktura vsebuje več parametrov, za vse obvezne parametre vnesemo veljavne vrednosti. Če so podatki v pravilni obliki nam mora metoda vrniti izhodne parametre z naslednjimi vrednostmi: `iResultCode = 0` ; `sResultDescription = OK` . (Kar pomeni da so bili podatki izvršne liste uspešno vneseni v zaledni sistem.)

Test:	17 (SetExecutableList)	
Tester:		
Datum testa:		
Scenarij:	<b>Scenarij testa 2</b> <b>Vhodni parametri:</b> <code>mExecutableData = struktura { ICUID, iAction, mAdditionalCardObject, ...}</code>	
Pričakovan rezultat:	<code>iResultCode = 0</code> ; <code>sResultDescription = OK</code> ;	
Dobljen rezultat:		
Rezultat:	Uspešen test <input type="checkbox"/>	Neuspešen test <input type="checkbox"/>



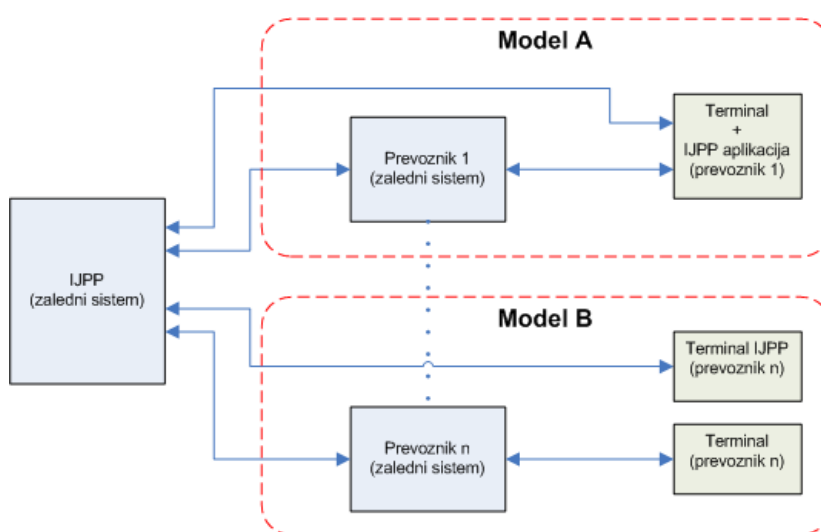
## 10.9 REFERENCE

- [1] [www.ijpp-standard.si](http://www.ijpp-standard.si).
- [2] [www.ijpp-priglasitev.si](http://www.ijpp-priglasitev.si)

## 11 VARNOSTNA SHEMA SISTEMA

Opisana varnostna shema sistema se nanaša na delovanje in procesiranje transakcijskih podatkov na terminalski opremi in komunikacijo z zalednimi sistemom.

Slika 11.1 prikazuje komunikacijo med posameznimi moduli in sistemi po modelu A, kjer se mora terminalska oprema prevoznika povezovati na zaledni sistem prevoznika in IJPP in po modelu B, kjer se IJPP terminalska oprema povezuje samo na zaledni sistem IJPP.



SLIKA 11.1: KOMUNIKACIJSKE POVEZAVE PRI MODELU A IN MODELU B

### 11.1 RAVNANJE Z OBČUTLJIVIMI PODATKI

V sistemu IJPP poteka obdelava in izmenjava velikega števila podatkov, ki morajo biti ustrezno zaščiteni. V to zaščito spada varnostna politika in ocena varnosti posameznega akterja, segmenta oz. dela sistema.

Med varnostno občutljive podatke v sistemu IJPP uvrščeno kriptografske ključne, osebne podatke uporabnikov in transakcijske podatke. Ti podatki ne smejo biti na voljo nepooblaščenim osebam. Osebe oz. službe, ki upravljajo oz. procesirajo varnostno občutljive podatke so:

- Upravljavec sistema IJPP.
- Upravljavec zalednega sistema prevoznika.
- Skrbniki strojne in informacijske opreme.
- Dobavitelj varnostnih elementov.
- Služba za personalizacije varnostnih elementov.
- Služba za integracijo in aktivacijo terminalske opreme.
- Služba za pomoč uporabnikom.

- Služba za tehnično podporo.

Vsi ki pri svojem delu uporabljajo in upravljajo z občutljivimi informacije morajo biti seznanjeni z varnostnimi zahtevami pri njihovem delu. Upravljaivec oz. pogodbeni izvajalec mora imeti izdelano varnostno politiko, ki vsebuje možna tveganja, korektivne procedure in procedure ob nastanku kršitve. Varnostna politika predstavlja metode za varovanje informacij pred notranjimi in zunanjimi vplivi, z namenom zagotavljanja stabilnosti, zaupnosti in varnosti. Na podlagi tega je razvit celovit program varnosti, ki mora vsebovati:

- Opredelitev fizičnega dostopa do sistema (omejevanje, beleženje).
- Definicijo načina identifikacije uporabnika za dostop (fizična in elektronska).
- Definicijo dodeljevanja, sprememb in prenehanja dostopnih pravic uporabnikov.
- Varovanje kritične strojne opreme (lokacija nameščene opreme, kako je oprema nameščena, zaščita pred nesrečami,...).
- Procedure za nemoteno delovanje sistema ob izpadu električnih in telekomunikacijskih povezav.
- Načrt vzdrževanja informacijskega sistema in preprečevanja nameščanja ali uporabe zlonamerne programske opreme.
- Metode kriptiranja za zaščito zaupnosti podatkov.

#### **11.1.1 KLJUČI IN CERTIFIKATI**

Za zagotavljanje varnega sporočanja, so v sistemu uporabljeni različni kriptografski ključi in certifikati. Poleg same vrednosti ključa morajo biti opremljeni še z informacijami o verziji, datumu veljavnosti, načinu in mestu uporabe.

V IJPP sistemu se uporabljajo naslednji ključi:

- $K_{SAK\_M}$  – glavni ključ, ki omogoča upravljanje z varnostnim elementom.
- $K_{SAK}$  - ključ, ki omogoča avtentikacijo med terminalsko opremo in varnostnim elementom (nameščen ob namestitvi in aktivaciji varnostnega elementa).
- $K_{APP}$  - ključ, ki omogoča preverjanje pristnosti aplikacije na terminalski opremi.
- $K_{NFC\_M}$  - glavni ključ, ki je v lasti izdajatelja brezstične kartice.
- $K_{NFC\_R}$  - bazni ključ IJPP brezstične aplikacije.
- $K_{NFC\_I}$  – privzeti avtentikacijski ključ nove IJPP brezstične aplikacije.
- $K_{NFC\_IJPP}$  - avtentikacijski ključ IJPP brezstične aplikacije, ki omogoča njeno uporabo.
- $K_{IJPP\_T}$  - ključ oz. certifikat terminalske opreme, ki omogoča varno komunikacijo terminalske opreme z zalednim sistemom IJPP.
- $K_{IJPP\_Z}$  - ključ oz. certifikat, zalednimega sistema IJPP.

Tabela 11.1 prikazuje lastnosti posameznih ključev in certifikatov.

TABELA 11.1: LASTNOSTI KLJUČEV IN CERTIFIKATOV V SISTEMU IJPP

Oznaka	Tip	Velikost [bit]	Veljavnost	Opomba
$K_{SAK\_M}$	AES	192	*	*Do preklica
$K_{SAK}$	AES	128	*	*Do preklica
$K_{APP}$	*	*	*	*Odvisno od ponudnika terminalske opreme
$K_{NFC\_M}$	3KDES	3 x 56	*	*Do preklica
$K_{NFC\_R}$	3KDES	3 x 56	*	*Do preklica
$K_{NFC\_I}$	3KDES	3 x 56	*	*Do preklica
$K_{NFC\_IJPP}$	3KDES	3 x 56	*	*Do preklica
$K_{IJPP\_T}$	RSA	256 - 2048	*	*Določi izdajatelj
$K_{IJPP\_Z}$	RSA	256 - 2048	*	*Določi izdajatelj

Za upravljanje s ključi mora upravljavec IJPP poskrbeti za sistem za upravljanje s ključi, ki skrbi za generiranje, hrambo in izmenjavo ključev. Takšen sistem sestoji iz baze za hrambo ključev in njihovih lastnosti, servisov oz. vmesnikov za komunikacijo s sistemom, priporočen strojni modul za kriptografske operacije (HSM) in uporabniški vmesnik.

Dostop do sistema za upravljanje s ključi mora biti fizično in logično omejen. Priporočena je uporaba varne sobe:

- omejen dostop skozi ena vrata, ki so pod video nadzorom; dostop mora biti beležen in omočen samo pooblaščenim osebam,
- oprema v sobi ne uporablja zunanjih komunikacijskih povezav; uporablja se samo lokalna mreža.

Za transport ključev se uporabljajo varni mediji v obliki pametnih kartic, ki omogočajo varno hrambo. Pri transportu se uporablja transportni ključ, ki se generira naključno in se končnemu uporabniku sporoči preko drugega transportnega kanala.

Upravljavec mora zagotoviti vsaj minimalne zahteve za sistem upravljanja s ključi:

- upravljanje s ključi skozi celotni življenjski cikel,
- metode za obnovitev ključev v primeru okvare,
- beleženje dostopov do vpogleda v bazo podatkov,
- onemogočen dostop uporabnikov do vidne vrednosti ključa,
- kriptirana baza za shranjevanje ključev.

### ***11.1.2 OSEBNI PODATKI***

Obdelava in dostop do osebnih podatkov uporabnikov IJPP brezstične aplikacije mora biti v skladu z veljavno zakonodajo o varovanju osebnih podatkov, ki velja na območju RS.

V sistemu IJPP morajo biti osebni podatki, ki se pošiljajo med različnimi sistemi, pretvorjeni v psevdonim oz. identifikacijsko številko.

Dostop do baze uporabnikov je mogoč le ob uspešni identifikaciji dveh pooblaščenih uporabnikov.

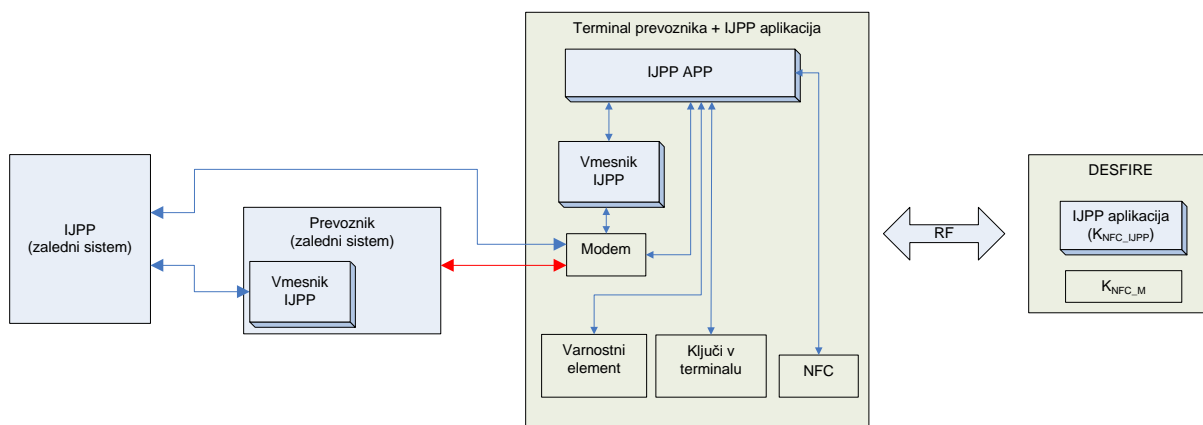
### ***11.1.3 TRANSAKCIJSKI PODATKI***

Transakcijski podatki, ki se prenašajo v sistemu IJPP morajo biti kriptirani s čimer se zagotovi zaupnost in celovitost informacij.

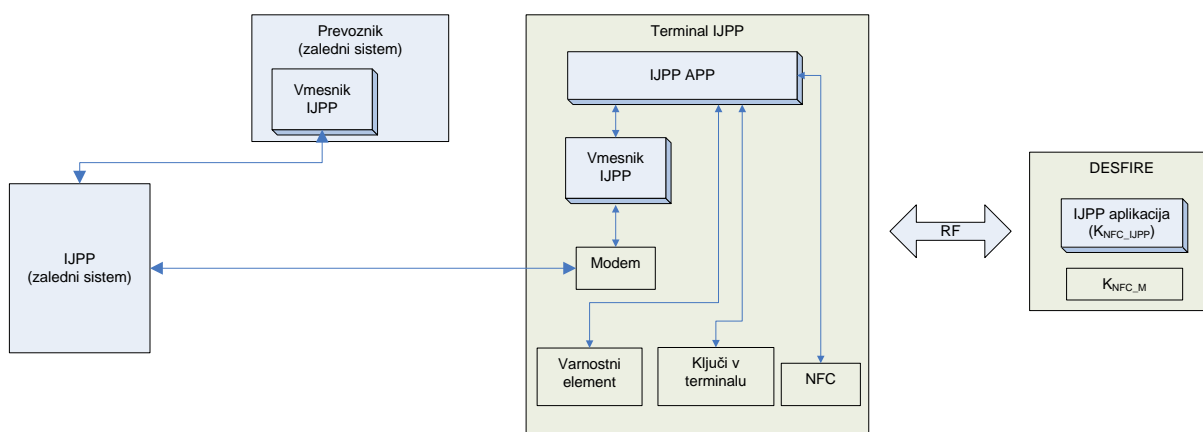
## **11.2 VAROVANJE PODATKOV ZNOTRAJ SISTEMA IJPP**

Glede na zgradbo sistema, lahko ločimo posamezne segmente sistema in informacije, ki jih ta segment zajema, in za njih določimo minimalne varnostne zahteve. Glavni deli sistema IJPP so:

- Zaledni sistem IJPP, ki izvaja procesiranje transakcijskih podatkov IJPP, hrani bazo vključene terminalske opreme in izdanih brezstičnih kartic z IJPP aplikacijo.
- Terminalska oprema, ki zajema validacijske terminale, terminale za nakup produktov in terminale za izvajanje kontrole (veljavnost produktov ob koriščenju).
- Brezstična kartica, ki ima nameščeno IJPP aplikacijo (in druge aplikacije, v kolikor je to dovoljeno).
- Varnostni element, ki omogoča uporabo terminalske opreme v sistemu IJPP.
- Brezstična IJPP aplikacija, ki se nahaja na brezstični kartici in vsebuje podatke o produktih in uporabniku.
- IJPP aplikacija na terminalski opremi, ki skrbi za izvajanje IJPP transakcije. Vključuje uporabo strojne opreme terminalske opreme in varnostnega elementa.



SLIKA 11.2: KOMUNIKACIJSKE POVEZAVE V SISTEMU IJPP (MODEL A); RDEČA POVEZAVA PONAŽARJA INFORMACIJE, KI NISO DEFINIRANE ZNOTRAJ SISTEMA IJPP



SLIKA 11.3: KOMUNIKACIJSKE POVEZAVE V SISTEMU IJPP (MODEL B)

### 11.2.1 ZALEDNI SISTEM IJPP

Zaledni sistem IJPP, preko vstopne točke za terminale, omogoča izvedbo zahteve oz. transakcij, prejetih iz terminalske opreme. Zaledni sistem ponuja več hkratnih vstopnih točk, ki so zaščitene z naslednjimi mehanizmi:

- Varna povezava (SSL , TLS, IPSec, PPTP, ...); implementirana mora biti vsaj ena možnost.
- Požarni zid.
- Dostop mogoč le aktivnim terminalom.
- Mehanizmi za preprečevanje vdora in prekinitve delovanja.

Vsa terminalska oprema na terenu, ki se želi povezati z zalednim sistemom IJPP zahteva uporabo certifikata, ki je nameščen na terminalski opremi (oz. varnostnem elementu) -  $K_{IJPP\_T}$ . Poleg tega mora imeti terminalska oprema status aktivne naprave v sistemu.

### 11.2.2 TERMINALSKA OPREMA

Terminalska oprema, s pomočjo varnostnega elementa, omogoča uporabo brezstične IJPP aplikacije. Ponudnik terminalske opreme mora zagotoviti upravljanje z njegovim življenjskim ciklom in zagotoviti mehanizem za vnos potrebnih ključev in namestitvev IJPP aplikacije. Prav tako mora poskrbeti, da je vsak terminal skladen s predpisi IJPP.

Terminalska oprema ima nameščene naslednje ključe:

- $K_{SAK\_M}$  – glavni ključ, ki omogoča upravljanje z varnostnim elementom.
- $K_{SAK}$  - ključ, ki omogoča avtentikacijo med terminalsko opremo in varnostnim elementom (nameščen ob namestitvi in aktivaciji varnostnega elementa).

Ključ, ki omogoča uporabo varnostnega elementa v terminalski opremi -  $K_{SAK}$ , mora zagotoviti ponudnik varnostnega elementa. Vnos ključa v terminal se opravi po postopku, ki ga zahteva njegov ponudnik.

### **11.2.3 BREZSTIČNA KARTICA**

Brezstična kartica ima nameščeno brezstično IJPP aplikacijo, ki omogoča njeno uporabo v sistemu IJPP. Na njej so shranjeni podatki o produktih, ki jih je uporabnik kupil oz. porabil in informacije o lastniku ali uporabniku kartice. Poleg IJPP aplikacije so lahko na kartici tudi druge aplikacije, v kolikor le-te ne onemogočajo delovanja IJPP aplikacije.

Brezstične kartice imajo nameščene naslednje ključe:

- $K_{NFC\_M}$  - Glavni ključ, ki je v lasti izdajatelja brezstične kartice.
- $K_{NFC\_IJPP}$  - Avtentikacijski ključ IJPP brezstične aplikacije, ki omogoča njeno uporabo.

Izdajatelj brezstične kartice je njen lastnik, ki ima v lasti glavni ključ  $K_{NFC\_M}$ . Z njim je omogočen zapis, spreminjanje in izbris novih aplikacij na brezstični kartici. Pri zapisu aplikacije IJPP se ustvari nov ključ  $K_{NFC\_IJPP}$  za vsako kartico.

### **11.2.4 VARNOSTNI ELEMENT**

Uporaba varnostnega elementa je zahtevana v terminalski opremi, ki je vključena v sistem IJPP. Omogoča varno shrambo in distribucijo na terenu.

Varnostni element ima nameščene naslednje ključe:

- $K_{SAK\_M}$  – glavni ključ, ki omogoča upravljanje z varnostnim elementom.
- $K_{SAK}$  - ključ, ki omogoča avtentikacijo med terminalsko opremo in varnostnim elementom (nameščen ob namestitvi in aktivaciji varnostnega elementa).
- $K_{NFC\_R}$  - bazni ključ IJPP brezstične aplikacije.
- $K_{NFC\_I}$  – privzeti avtentikacijski ključ nove IJPP brezstične aplikacije.

### **11.2.5 BREZSTIČNA IJPP APLIKACIJA**

Brezstična aplikacija IJPP omogoča validacijo oz. uporabo produktov, ki jih ima uporabnik. Vsaka aplikacija je zaščitena s ključem za avtentikacijo v sistem IJPP -  $K_{NFC\_IJPP}$ . Ključ je zapisan na brezstični kartici ob zapisu aplikacije.

#### **11.2.6 IJPP APLIKACIJA NA TERMINALSKI OPREMI**

IJPP aplikacija na terminalski opremi mora biti izdelana in validirana po specifikacijah IJPP. Za namestitev in delovanje na izbrani terminalski opremi mora biti podpisana s ključem ponudnika terminalske opreme -  $K_{APP}$ .

### **11.3 VAROVANJE POVEZAV Z ZUNANJIMI SISTEMI**

Za komunikacijo do zunanjih sistemov je potrebno zagotoviti varno povezavo, ki omogoča varno sporočanje. Sistemi s katerimi se sistem IJPP lahko povezuje so:

- Zaledni sistem prevoznika, ki sporoča informacije v zaledni sistem IJPP in omogoča komunikacijo s terminalsko opremo tega prevoznika.

#### **11.3.1 ZALEDNI SISTEM PREVOZNIKA**

Zaledni sistem posameznega prevoznika, ki je vključen v sistem IJPP mora omogočati povezavo do zalednega sistema IJPP, preko varne povezave. Le-ta je zavarovana z uporabo:

- Varna povezava (SSL , TLS, IPSec, PPTP, ...); implementirana mora biti vsaj ena možnost.
- Požarni zid.

Distribucija certifikatov za komunikacijo z zalednimi sistemi prevoznikov se opravi preko dogovorjenega varnega distribucijskega kanala.

### **11.4 VAROVANJE POVEZAV PROTI KONČNIM UPORABNIKOM (ADMINISTRACIJA, SPLETNI PORTAL)**

Končni uporabniki, ki lahko dostopajo do sistema IJPP morajo imeti določene uporabniške pravice, ki določajo stopnjo dostopa. S stopnjami dostopa so definirane vrste informacij in dovoljene operacije v sistemu.



## 12 PODPORA SISTEMA

### 12.1 DEFINICIJA PODPORE SISTEMA

Podpora sistema definira korake in postopke, katerim je potrebno slediti pri razreševanju problema v določenem časovnem okvirju. Podpora sistema je urejena na več nivojev. Vsak nivo ima definirane svoje pristojnosti, odgovorne osebe in časovne okvirje, v katerih mora biti problem razrešen. Znotraj podpore sistema se definirajo postopki obveščanja in okoliščine, pri katerih se problem posreduje višjim nivojem. Posredovanje problema višjim nivojem je primarno določeno s časovnimi okvirji. V primeru, da določen nivo podpore ne razreši problema, v naprej določenem časovnem okvirju, le tega posreduje višjemu nivoju. Možne so izjeme, kjer se problem posreduje neposredno višjim nivojem podpore.

Podpora sistema velja za programske rešitve sistema IJPP, systemske rešitve IJPP in strojne opreme v okviru upravljavca IJPP.

### 12.2 DEFINICIJE

- **Incident, težava**  
Dogodek, zaradi katerega je stranka kontaktirala podporo sistema.
- **Problem**  
Dejanska napaka oziroma sum napake, ki ne povzroči spremembe delovanja izdelka.
- **Napaka**  
Napaka v produktu, ki povzroča, da izdelek ne deluje v skladu s specifikacijami.
- **Popravek**  
Sprememba v programski kodi, binarni kodi ali izvršni kodi, katera razreši napako v delovanju izdelka.
- **Začasna rešitev**  
Sprememba v izvajanju postopka, kateremu stranka sledi, brez bistvenega poslabšanja uporabe izdelka.

### 12.3 NIVOJI PODPORE

Nivoji podpore so določeni za čim bolj uspešno razrešitev problema. Definirane so napake in problemi, ki določajo prehod med nivoji podpore. Za uspešno razrešitev problema, v določenem časovnem okvirju, so definirane procedure, ki določajo odgovornost za razrešitev ter informiranje vseh oseb, katere so del ekipe pri razreševanju problema.

### **12.3.1 PRVI NIVO**

Na prvem nivoju se razrešujejo problemi s pomočjo pred pripravljene dokumentacije. Dokumentacija zajema uporabniška navodila, tehnične zapiske, dokument pogostih vprašanj (FAQ), itd.

Postopki omogočajo pregled celotnega opisa simptomov problema, informacijo o konfiguraciji in verziji izdelka ter informacijo o možni ponovitvi problema.

#### **12.3.1.1 Naloge**

- Nuditi podporo uporabnikom 24/7
- Izobraževanje končnih uporabnikov
- Izobraževanje uporabnikov IJPP izdelkov
- Dobava potrošnega materiala
- Raziskovanje odpovedi izdelka
- Logistika izdelka (dostava, posodobitve informacij)
- Dodelitev prodajnega mesta
- Spremljanje transakcij

#### **12.3.1.2 Potek delovanja**

Klicni center sprejme pritožbo oziroma problem uporabnika, zajame njegove podatke in pošlje obvestilo preko elektronske pošte. Če lahko reši pritožbo direktno ali kličočega podučni glede uporabe, to tudi izvede in zabeleži to znotraj obvestila, ki je bil poslan na elektronsko pošto, drugače se počaka odgovor s strani servisne službe oziroma drugega nivoja podpore.

#### **12.3.1.3 Odgovorna oseba**

Profesionalni klicni center z ustreznimi podatki in razpoložljivostjo in povezavo do drugega nivoja podpore.

### **12.3.2 DRUGI NIVO**

Na drugem nivoju podpore se preveri, ali je problem samostojen ali je v povezavi še s katerimi drugimi. Poskuša se identificirati problem v specifikacijah izdelka in razpoznavanje nepravilnega delovanja izdelka.

Drugi nivo razišče tudi začasno predlagano rešitev, da se lahko stranka izogne problemu.

#### **12.3.2.1 Naloge**

- Nadzor nad delovanjem sistema 24/7
- Nameščanje, nadgradnja in vzdrževanje programske opreme
- Izvajanje izobraževanje za naročnika in partnerje naročnika
- Konfiguracija izdelka (posodobitve, nastavitve, ...)
- Lažje mehanično popravilo izdelka (baterija, tipkovnica, ekran,...)
- Napredna logistika izdelka
- Pridobivanje statistike transakcij
- Usklajevanje podpore

#### **12.3.2.2 Potek delovanja**

Spremlja in odgovarja se elektronska pošta, ki prispe na naslov odgovorne osebe. Če problema oziroma pritožbe ni možno rešiti, se posreduje elektronska pošta do 3. nivoja, osebi, ki je odgovorna za določeno problematiko oziroma segment. Po dobljenem odgovoru se le ta posreduje do končnega uporabnika.

#### **12.3.2.3 Odgovorna oseba**

Znotraj upravljavca IJPP se določijo osebe, ki skrbijo za drugi nivo podpore.

### **12.3.3 TRETJI NIVO**

Tretji nivo razišče in pojasni znane podatke o napaki. Prav tako preuči napako v izvorni kodi ali strojni opremi. V primeru, da se ugotovi možnost začasne rešitve, le-to preučijo in potrdijo.

Na tem nivoju se izvedejo zadolžitve za popravke.

Vsi postopki so skrbno in natančno dokumentirani.

#### **12.3.3.1 Potek delovanja**

Po dobljeni elektronski pošti, se razišče problem in odgovori, ali delna ali celotna rešitev, pošiljatelju elektronske pošte, znotraj 24 ur, če so napake javljene v rednem delovnem času; v kolikor so napake javljene izven rednega delovnega časa, pa morajo biti odpravljene najkasneje prvi delovni dan do 15. ure. Po potrebi se kontaktira 4. nivo.

#### **12.3.3.2 Odgovorna oseba**

Podpora tehnološkega ponudnika.

#### **12.3.4 ČETRTI NIVO**

Za ugotovitev in popravilo napake četrtega nivoja je potrebno strokovno znanje. Za uspešno razrešitev je potrebno sodelovanje z razvojno ekipo.

Ko je pripravljen popravek, se posodobitev le-tega načrtuje glede na nivo problema. Lahko se izdela tudi poseben popravek, ki je prilagojen samo za določeno stranko.

V primeru kritične napake se popravek lahko hitro posodobi, vendar ni nujno, da je popravek nujno testiran.

##### **12.3.4.1 Potek delovanja**

Po dobljeni elektronski pošti ali opozorilu 3. stopnje, se razižče problem in se odgovori, ali delna ali celotna rešitev, do 2. stopnje v 24 urah, če so napake javljene v rednem delovnem času oziroma najkasneje prvi delovni dan do 16. ure, če so napake javljene izven rednega delovnega časa

##### **12.3.4.2 Odgovorna oseba**

Razvojna ekipa tehnološkega ponudnika.

#### **12.4 DEFINICIJE RAVNI RESNOSTI PROBLEMA**

Sledijo definicije ravni, katere definirajo resnost problema, ki so razdeljene na več kategorij. Ravni resnosti so razdeljene glede na vpliv problema, kakršnega zazna stranka.

##### **12.4.1 KRITIČNI POSLOVNI UČINEK**

IJPP sistem je doživel celoten izpad funkcionalnosti oziroma nima sredstev za delovanje. Začasna rešitev ne obstaja. Celotni sistem je neuporaben.

###### **12.4.1.1 Primeri kritičnih učinkov**

- IJPP sistem je neodziven. Stranke ne morejo izvesti transakcij.
- Nihče od strank določene regije oziroma določenega prevoznika ne more uporabljati sistema.
- Nihče od potnikov ne mora izvesti validacije na katerem koli vozilu.
- Onemogočen je nakup vozovnic na vseh prodajnih mestih IJPP.

##### **12.4.2 RESNI POSLOVNI UČINEK**

IJPP sistem ne deluje v celoti. Del sistema je neodziven oziroma nedelujoč. Povečana je poslovna izguba. Za razrešitev problema je lahko ponujena začasna rešitev, ki omogoči delno uporaben sistem.

#### **12.4.2.1 Primeri resnih učinkov**

- Pojavljajo se težave s komunikacijo.
- Storitve niso na voljo.
- Sistem pade v zanko in postane neodziven za določen čas.

#### **12.4.3 SREDNJI POSLOVNI UČINEK**

IJPP sistem deluje z manjšimi omejitvami. Na samo stranko in transakcije ni problematičnega učinka. Pojavlja se problem, da sistem ne deluje po specifikacijah oziroma kakšen manjši del ni delujoč.

#### **12.4.3.1 Primeri srednjih učinkov**

- Terminal se ne prižge.
- Terminal ne deluje za prodajalca, vendar pod skrbnikom deluje.
- Ni mogoče spremeniti lastnosti terminala, vendar jih lahko izbrišemo in ponovno naložimo.
- Pogosto se izvaja servisna storitev.

#### **12.4.4 MANJŠI OZIROMA NIČELNI POSLOVNI UČINEK**

IJPP sistem deluje brez težav. Stranka ne zazna nikakršnih težav okvare iz strani sistema. Problem je lahko manjše nezadovoljstvo pri uporabi izdelka. To nezadovoljstvo je lahko problem nerazumevanja sistema oziroma nepravilno izvedeno izobraževanje.

#### **12.4.4.1 Primeri manjših, ničelnih učinkov**

- Izdelek ima prazno baterijo in ga je potrebno priključiti na zunanje napajanje.
- Osvetlitev zaslona na izdelku ne deluje.

#### **12.4.5 DODATNA ZAHTEVA**

Stranka želi oziroma predlaga spremembo v delovanju izdelka.

## 12.5 ODZIVNI ČASI PODPORE

Po prejetju obvestila o problemu na določeni nivo podpore, je potrebno posredovati povratno informacijo o prejetju problema in informacij. Spodnja tabela prikazuje časovne okvirje, kateri so cilj odziva.

Časovni okvirji se delijo glede na resnost problema in glede na to, kateri nivo podpore je zahtevan.

Odziv je sestavljen iz potrditve o prejemu, zahtevi po dodatnih informacijah in informaciji na kakšen način bo potekalo razreševanje problema.

TABELA 12.1: TABELA ODZIVNIH ČASOV

<i>Nivo podpore</i> <i>Resnost problema</i>	<i>Prvi nivo</i>	<i>Drugi nivo</i>	<i>Tretji nivo</i>	<i>Četrti nivo</i>
<b>Kritična</b>	Do 15 min	Do 1 ure	Znotraj 2 ur če je bilo posredovano med delovnim dnevom, drugače 5 ur	Znotraj 2 ur če je bilo posredovano med delovnim dnevom, drugače 5 ur
<b>Resna</b>	Do 15 min	Do 1 ure	Do 8 ur	Do 8 ur
<b>Srednja</b>	Do 15 min	Do 1 ure	Do 16. ure naslednjega delovnega dne	Do 16. ure naslednjega delovnega dne
<b>Manjša, ničelna</b>	Do 15 min	Do 1 ure	Do 16. ure naslednjega delovnega dne	Do 16. ure naslednjega delovnega dne
<b>Zahtevek izboljšave</b>	V skladu z dogovorom in v odvisnosti od zahtevnosti posega.	V skladu z dogovorom in v odvisnosti od zahtevnosti posega.	V skladu z dogovorom in v odvisnosti od zahtevnosti posega	V skladu z dogovorom in v odvisnosti od zahtevnosti posega

Delovni dnevi: Ponedeljek – Petek (razen prazniki)

Delovni čas: 8:00 – 16:00

## 12.6 RAZDELITEV ODGOVORNOSTI

### 12.6.1 PRVI NIVO

Profesionalni klicni center z ustreznimi podatki in razpoložljivostjo ter povezavo do drugega nivoja podpore.

### 12.6.2 DRUGI NIVO

Znotraj upravljavca IJPP se določijo osebe, ki skrbijo za drugi nivo podpore.

### 12.6.3 TRETJI NIVO

Podpora tehnološkega ponudnika.

### 12.6.4 ČETRTI NIVO

Razvojna ekipa tehnološkega ponudnika.

## 12.7 PREHOD MED NIVOJI PODPORE

Tabela v nadaljevanju prikazuje meje, ki jih je potrebno upoštevati pri stopnjevanju problema na višjo raven podpore. Na primer, kritični problemi se predajo na drugi nivo takoj, ko se prvotno zberejo podatki.

TABELA 12.2: PREHODI MED NIVOJI

<b>Nivo podpore</b> <b>Resnost problema</b>	<b>Prvi nivo</b>	<b>Drugi nivo</b>	<b>Tretji nivo</b>	<b>Četrti nivo</b>
<b>Kritična</b>	Ko se problem potrdi in se zberejo podatki, je potrebno takoj predati problem na višji nivo. (24/7 ur)	Takoj opozoriti tretjo stopnjo in predati podatke. Problem se poskuša rešiti v roku 4h. Po preteku tega časa se problem preda na tretjo stopnjo.	24 ur	Dokler se problem ne reši

<b>Resna</b>	Ko se problem potrdi in se zberejo podatki, je potrebno sporočiti informacije drugemu nivoju. Problem se poskuša odpraviti v roku 4h. Po preteku časa se problem preda na višji nivo(24/7 ur)	Takoj obvestiti tretjo stopnjo in poskusiti rešiti problem v 1 dnevno do predaje	24 ur	Dokler se problem ne reši
<b>Srednja</b>	1 delovni dan (delovne ure)	1 delovni teden (delovne ure)	1 delovni dan (delovne ure)	Časovnica dogovorjena s stranko
<b>Manjša, ničelna</b>	1 delovni dan (delavne ure)	1 delovni teden (delovne ure)	1 delovni teden (delovne ure)	Časovnica dogovorjena s stranko
<b>Zahtevek izboljšave</b>	1 delovni dan (delovne ure)	1 delovni teden (delovne ure)	1 delovni teden (delovne ure)	Časovnica dogovorjena s stranko

## 12.8 SLEDENJE ZAHTEV

Vse napake se beležijo v sistem beleženja napak. Ko se napaka prenese na drugo stopnjo podpore, podporni vodja stranki vsakodnevno omogoča pregled o stanju napredka, dokler problem ni rešen.

## 12.9 NAČINI KOMUNIKACIJE MED NIVOJI

Komunikacija med drugo in tretjo stopnjo se ureja preko telefona, elektronske pošte in IJPP podporne spletne strani. Jezik komunikacije je slovenski. Vse komunikacije se sledi v sistemu beleženja napak na IJPP podporni spletni strani.

## 12.10 DOTIČNA TOČKA



Odgovorne osebe morajo biti dodeljene na obeh straneh podpornih stopenj, za učinkovito vodenje in nadzor poteka reševanja problema.

## 12.11 VZOREC OBRAZCA NA KLICNEM CENTRU

Ob prijavi problema je potrebno zajeti čim več podatkov, ki so v pomoč pri razreševanju problema. Pomembni podatki ob prijavi so čas, datum in način prijave. Potrebno je pridobiti kontaktne podatke osebe, katera je naredila prijavo. Najpomembnejša informacija je seveda opis problema.

Potrebni podatki na obrazcu:

- zaporedna številka prijave,
- datum prijave,
- čas prijave,
- način kontakta,
- vrsta problema,
- podjetje,
- identifikacijska številka kartice,
- povratni klic,
- pomembnost razrešitve,
- posredovanje problema,
- opis problema,
- status razreševanja.

## 13 POSTOPEK SERVISIRANJA TERMINALOV IJPP

Poglavje se navezuje na prevoznike, ki bodo prevzeli terminalsko opremo od upravljavca IJPP.

### 13.1 ENTITETE VKLJUČENE V POSTOPEK SERVISA

#### **Upravljavec IJPP:**

- bo vzdrževal terminale tako, da bodo le-ti optimalno delovali; ta obveznost vključuje tudi primere izrednega vzdrževanja, ko bi potreba po popravilu terminalov nastala zaradi nestrokovnega ravnanja prevoznika (npr. politje s tekočino, izpostavljanje terminalov direktnemu soncu ali padavinam, poškodovanje terminalov z ostrim predmetom,

poškodovanje terminala zaradi metanja ali padca, uporaba in polnjenje baterije v drugih napravah) ali drugačnih zunanjih vplivov, zaradi priključitve (uporabe) tujih izdelkov brez soglasja Upravljavca IJPP ali zaradi izvajanja del na terminalih s strani drugih oseb ali drugih firm in ne s strani Upravljavca IJPP, vendar pod pogoji nadomestila stroškov določenih v poglavju 13.2 (Poškodbe, uničenje ali izguba in/ali kraja terminala ter povračilo škode).

**Prevoznik:**

- bo s terminalom ravnal skrbno kot dober gospodar, v skladu s prejetimi navodili Upravljavca IJPP, pri čemer vse rizike in stroške, ki bi nastali zaradi neupoštevanja navodil Upravljavca IJPP ali od njega pooblaščen osebe, ter dejanji navedenih v poglavju 13.2. pokriva prevoznik sam;
- ne bo v nobenem primeru posegal v terminal oziroma delovanje terminala;
- bo imel terminal ob vsakem času v svoji posesti in ga ne bo posojal nepooblaščenim (tretjim) osebam;
- bo v delovnem času omogočil Upravljavcu IJPP ali od njega pooblaščen osebi dostop do terminala;
- bo nemudoma prijavil vse napake in pomanjkljivosti v delovanju terminala ter morebitne poškodbe ali izgube terminala znotraj spletne aplikacije;

Terminali IJPP imajo določeno garancijo, posledično obravnava garancijske zahteve Upravljavec IJPP.

## 13.2 POŠKODBE, UNIČENJE ALI IZGUBA IN/ALI KRAJA TERMINALA TER POVRAČILO ŠKODE

Prevoznik je odgovoren za vsako poškodovanje, uničenje ali izgubo terminala, četudi je do njega prišlo brez krivde prevoznika in/ali njegovih delavcev.

Prevoznik je dolžan Upravljavcu IJPP povrniti vso škodo, ki mu je nastala zaradi poškodovanja, uničenja ali izgube terminala, za kar Upravljavec IJPP izstavi račun.

Povračilo škode prevoznika je odvisno od narave poškodbe. V kolikor je poškodba terminala takšne narave, da terminal postane neuporaben, kakor tudi v primeru izgube in/ali kraja terminala, je prevoznik dolžan povrniti vrednost novega terminala.

Če pride do okvare ali poškodbe opreme zaradi nepravilne uporabe, bo Upravljavec IJPP sam oziroma preko svojega pogodbenega partnerja poskušal okvare odpraviti. Če to ne bo mogoče, bo opremo nadomestil z novo, prevoznik pa se zavezuje Upravljavcu IJPP poravnati vse stroške popravila oziroma zamenjave. Račun bo izstavljen po opravljenem popravilu oziroma zamenjavi opreme.

## 13.3 SPLOŠNI POTEK DOGODKOV

- Uporabnik (prevoznik) prijavi zahtevek znotraj spletne aplikacije.
- Glede na tip zahtevka Upravljavec IJPP tole dodeli ustreznemu nivoju podpore IJPP.
- Nadaljnji koraki in časovne omejitve so definirani znotraj poglavja **Podpore sistema**.
- V kolikor se znotraj servisnega postopka ugotovi upravičenost do garancije, se bo material ustrezno popravil in/ali nadomestil.

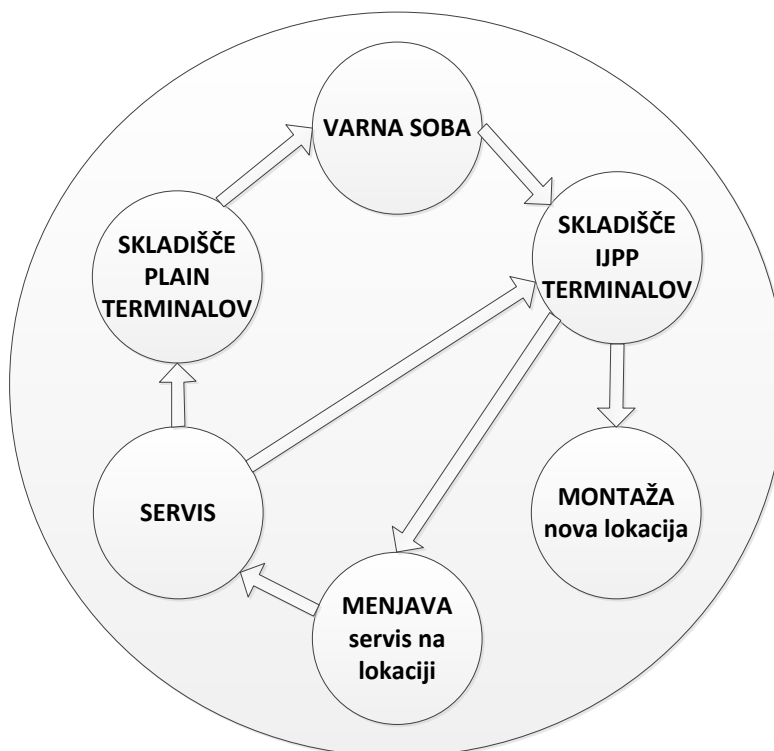
## 13.4 UGOTAVLJANJE GARANCIJ

### Osnovna določila

- V primeru, da se znotraj servisnega postopka ugotovi, da uveljavljanje garancije ni upravičeno, Upravljavec IJPP ni odgovoren za (a) izgubo podatkov ali (b) storitve obnovitve podatkov. Na primer, Upravljavec IJPP nima nikakršne odgovornosti povrniti ali plačati stroške povezane z obnovitvijo podatkov.
- V primeru, da se znotraj servisnega postopka ugotovi, da uveljavljanje garancije ni upravičeno, je prevoznik odgovoren, da ne pride do (a) izgube podatkov ali (b) storitve obnovitve podatkov. Na primer, prevoznik krije stroške povezane z obnovitvijo podatkov.
- Upravljavec IJPP vzdržuje sledenje serijskih števil. Katerikoli terminal, za katerega se ugotovi, da je ukraden, ni več v obsegu garancije. Zamenjava ali denarno povračilo za takšen terminal ne bosta izdana.
- Vse izmenjave opreme morajo biti potrjene s podpisom izročitvene pogodbe v dveh identičnih kopijah, kjer je jasno navedena vsa vročena oprema.

## 13.5 CIKEL TERMINALA IJPP

Slika 13.1 prikazuje življenjski cikel IJPP terminala, ki ga prevoznik prevzame od upravljavca IJPP.



SLIKA 13.1: ŽIVLJENJSKI CIKEL IJPP TERMINALA

## 13.6 SKLADIŠČE NOVIH TERMINALOV

V skladišču se nahajajo terminali, ki so pripravljeni za programiranje v varni sobi. Terminali morajo imeti nameščen aktiviran varnostni element, preden zapustijo skladišče, sicer nadaljnja procedura ni izvedljiva.

## 13.7 VARNA SOBA

V varni sobi poteka konfiguriranje in programiranje zaupnih kriptografskih podatkov, ki so namenjeni delovanju terminalov.

Po programiranju ima terminal nameščeno programsko opremo, ki podpira branje brezstičnih kartic in povezavo z zalednim sistemom, s katerim se lahko opravi osnovno testiranje.

Pripravi se potrebna tabela, v kateri se določi namembnost terminala in ostali podatki, ki so potrebni za vpis v zaledni sistem. Vpis v zaledni sistem se opraviti že v tej fazi, posodobitvena datoteka pa se še ne določi. Ta se bo vklopila za prenos šele takrat, ko bo terminal poslan na teren. Zraven omenjene

datoteke se posreduje xml datoteka s ključi zalednega sistema, z namenom, da bo terminal sposoben opraviti servisno storitev.

Po uspešno zaključenem programiranju, je potrebno testiranje terminala.

## 13.8 POSTOPEK TESTIRANJA TERMINALA

- NFC test (test z servisno kartico)
- Modem test – Test GPRS OK
- Vizualni test (preveriti LCD ekran in LED signalizacijo)
- Audio test (preveriti ali se zvok iz zvočnika sliši znotraj tolerančnega območja glasnosti in kvalitete)

### 13.8.1 STANJA NA ZASLONU OB TESTIRANJU

V nadaljevanju so opisani osnovni postopki testiranja, ko na terminalu ni naložena še nobena aplikacija.

#### 13.8.1.1 Normalno stanje

Na zaslonu je ves čas prikazan ključ.

#### 13.8.1.2 Branje servisne kartice

Ob približanju servisne kartice se na zaslonu izpiše ura in datum ter, v spodnjem levem kotu, status terminala, ki se s časom spreminja.

Če terminal nima posodobljene ure in datuma, oziroma, če je prišlo do katere koli druge napake v zvezi z uro in datumom, se na terminalu prikaže napaka.

Do tega pride, če terminal ni opravil še nobene uspešne servisne transakcije, razlog za to je razviden v statusu. Če je terminal opravil uspešno servisno transakcijo, je status »OK« in mora prikazovati čas, sicer je napaka na tiskanem vezju – potreben servis. Terminal sme v naslednjo fazo le v primeru, da prikazuje status »OK«.

### 13.8.2 ZAGON TERMINALA

Vstavi se testna SIM kartica, ob zagonu terminal prehaja skozi naslednja stanja:

UNKNOWN ERR → SIM ERR → SIM OK → GPRS OK → OK

### 13.8.3 ČITANJE OSTALIH KARTIC

Preveri se branje različnih tipov kartic, pri čemer terminal ne sme podpirati nobenih ostalih tipov kartic.

## 13.9 SKLADIŠČE IJPP TERMINALOV

V skladišču IJPP terminalov so terminali, ki vsebujejo že omenjeno aktivirano SIM kartico, opravili so postopek varne sobe in osnovne teste. S tem so pripravljeni na montažo. Vsebujejo potrebno programsko kodo, ki omogoča samodejno posodobitev terminala.

## 13.10 MONTAŽA

Ko pooblaščen oseba, za vgradnjo, prevzame terminal, je potrebno sporočiti zalednemu sistemu, da se temu terminalu dodeli posodobitvena datoteka – trenutna produkcijska verzija programske kode.

Preden se terminal dodeli avtobusu/vozilu, je potrebno počakati in preveriti, da je terminal uspešno posodobljen. V primeru, da ni prišlo do uspešnega zaključka posodobitve, je potrebno terminal ponovno zagnati in počakati 10 minut. Če še vedno ni uspešnega zaključka, je potrebno uporabiti drug terminal in ponoviti postopek.

Ko so vse posodobitve (črne liste, izvršne liste, tarife,...) uspešno opravljene, terminal ne sme prikazovati nobenih alarmov ali opozoril na zaslonu.

Sledi končni test terminala z uradno servisno kartico.

## 14 SISTEMI ZA PROMETNO PLANIRANJE

Sistem za prometno planiranje (v nadaljevanju SPP) je orodje, ki omogoča Upravljavcu IJPP celovito in učinkovito upravljanje s prometno-tehničnimi podatki za javni potniški promet v Republiki Sloveniji, kar vključuje:

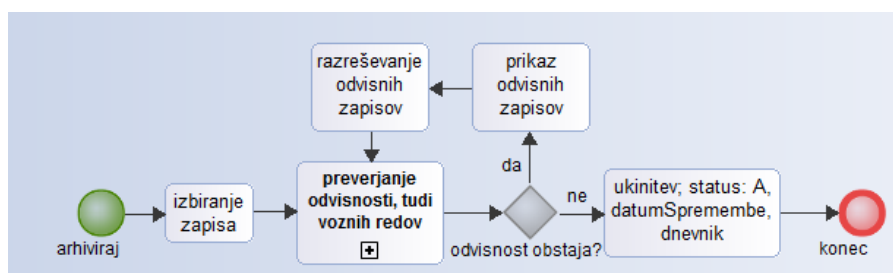
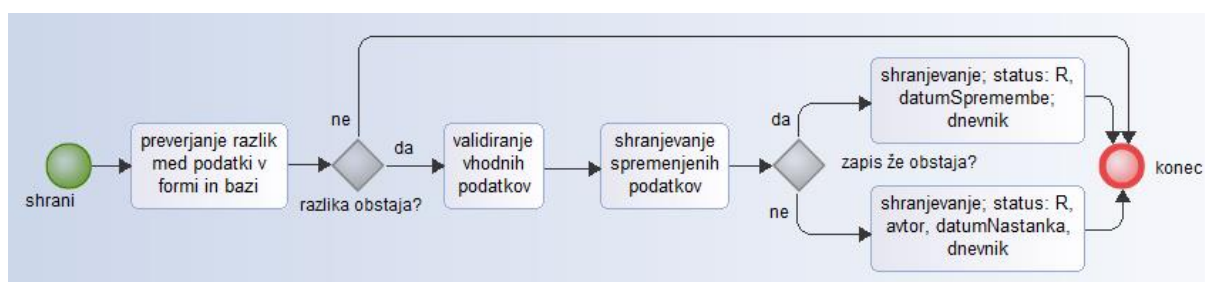
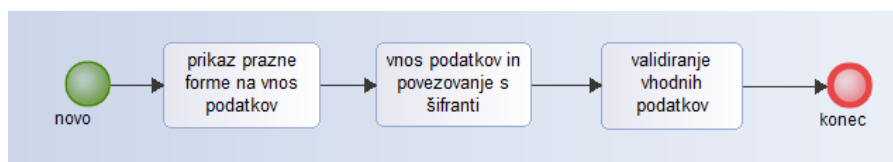
- Enotni daljinar za opis prometne infrastrukture javnega potniškega prometa (tj. postajališča, postajne točke, relacije, linijski odseki, prestopne točke in prestopi),
- Nacionalni vozni red (vozni redi in režimi) s pripadajočimi moduli za izvajanje analiz (npr. standard dostopnosti za okvirni vozni red) in
- Portal za potnike, ki črpa podatke iz enotna baze IJPP in zagotavlja prikaz kontekstno odvisnih podatkov glede na zahteve potnika (npr. postajni vozni red).

Sistemi za prometno planiranje so izdelani kot spletna aplikacija, kar omogoča vsem deležnikom vnos potrebnih prometno-tehničnih podatkov ob uvedbi / spremembi / ukinitvi obravnavane entitete.

### 14.1 OPIS PROCESOV

#### 14.1.1 OPIS PROCESOV ZA ENOTNI DALJINAR

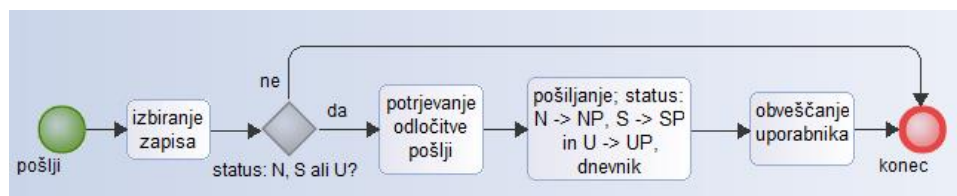
Enotni daljinar je namenjen opisu prometne infrastrukture za izvajanje javnega potniškega prometa, ki se uporablja za hitrejšo in bolj avtomatizirano pripravo in urejanje voznih redov. Dodajanje, urejanje in ukinjanje podatkov je v domeni Upravljavca IJPP, zato ostali uporabniki neposredno ne morejo uporabljati funkcionalnosti Enotnega daljinara. Programski modul Enotni daljinar omogoča dodajanje, spreminjanje in arhiviranje vseh entitet Enotnega daljinara. Procesi so opisani z naslednjimi diagrami:



### 14.1.2 OPIS PROCESOV ZA NACIONALNI VOZNI RED

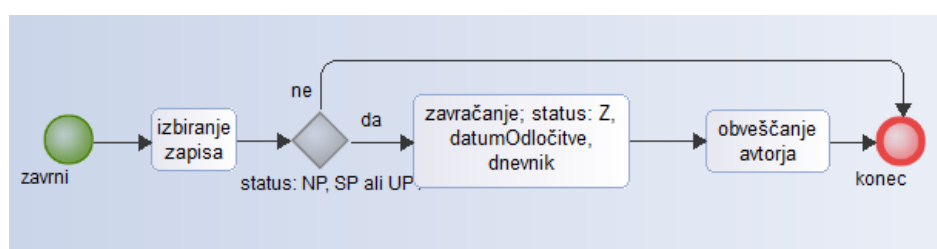
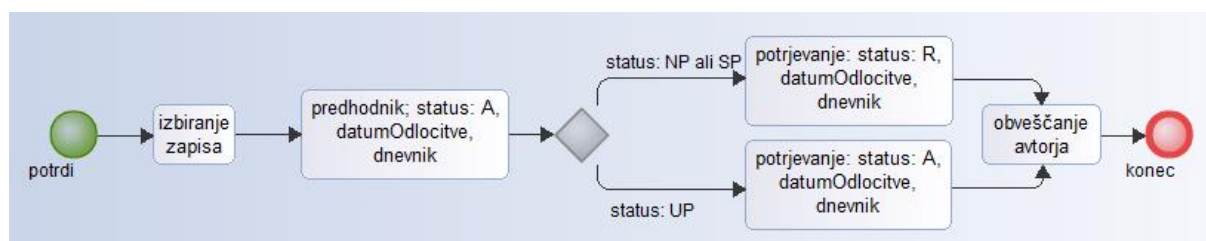
Upravljavec IJPP v SPP vnese roke za predložitev sprememb in dopolnitev voznorednih podatkov (tj. voznih redov in režimov). Sistem nato samodejno obvesti / pozove prevoznike k posodobitvi podatkov za katere so odgovorni oz. se neposredno nanašajo na njihovo dejavnost.

V naprej predpisanem roku prevozniki posodobijo / dodajo / ukinejo voznoredne podatke in predloge pošljejo Upravljavcu IJPP. Proces pošiljanja predlogov prevoznikov je opisan z naslednjim modelom:



Iz modela je razvidno, da imajo predlogi prevoznika lahko različne statuse, kot so: N (nov), S (spremenjen) ali U (ukinjen). Predlogi, ki so poslani v obravnavo Upravljavcu IJPP spremenijo svoj status, in sicer N v NP, S v SP in U v UP.

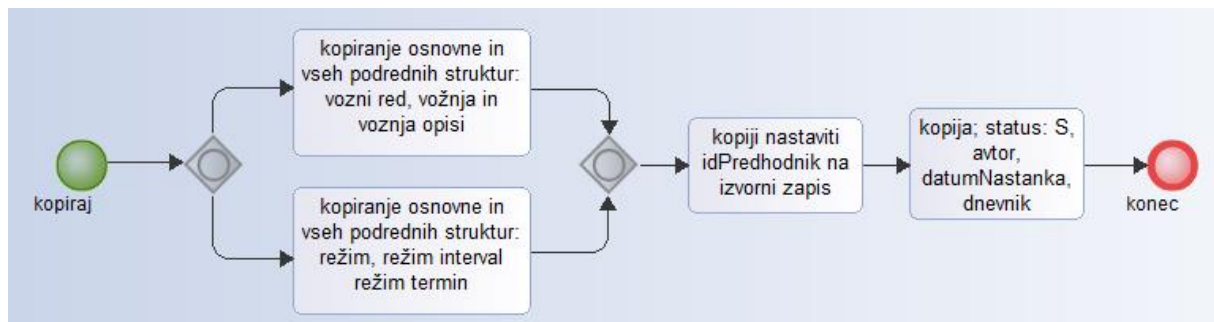
Zapise voznorednih podatkov s statusom NP, SP ali UP Upravljavec IJPP lahko obravnava. Glede na njihovo vsebino in vsebino ostalih prometno-tehničnih podatkov se odgovorna oseba odloči ali predlog sprejme ali zavrne. Proces potrjevanja oz. zavračanja je opisana v naslednjima diagramoma:



V primeru, ko Upravljavec IJPP odobri zahtevek, vnesene spremembe postanejo aktivne, hkrati pa sistem obvesti prevoznika o odobritvi oz. zavrnitvi. Prevoznik lahko spremlja v kateri fazi se poslani zahtevek nahaja.

Kadar Upravljavec IJPP odloči, da je potrebno zahtevek dopolniti, le-tega zavrne pod opombe pa zapiše, kaj mora prevoznik še zagotoviti za odobritev. Sprememba vnosa se zapiše v bazo, sistem pa prevozniku pošlje obvestilo o zavrnitvi z opombami. Če se prevoznik odloči za dopolnitev vloge, potem iz zavrnenega zapisa izdela kopijo. Proces kopiranja je opisan z naslednjim diagramom:





Nad kopiranim zapisom lahko prevoznik izvede popravke upoštevajoč opombe ter zapis pošlje v ponovno odločanje Upravljavcu IJPP. Poslani predlog vsebuje tudi povezavo na svojega predhodnika, zato SPP mogoča prikaz prehodno poslanega predloga in tako olajša delo Upravljavcu IJPP.

## 15 DEFINICIJE METOD IN PODATKOVNIH STRUKTUR ZA SPLETNE STORITVE

### – PROMETNI SISTEMI IJPP

#### 15.1 DEFINICIJE METOD ZA SPLETNE STORITVE – PROMETNI SISTEMI IJPP

##### 15.1.1 GETPREVOZNIKI

Metoda GetPrevozniki vrne seznam tistih podatkovnih struktur Prevoznik, katerih identifikatorji so navedeni v polju idPrevozniki. Če je argument idPrevozniki enak NULL, metoda vrne vse zapise.

Deklaracija metode	int <b>GetPrevozniki</b> ( <b>int</b> [] idPrevozniki [IN], <b>Prevoznik</b> []prevozniki [OUT], <b>int</b> rezultat [OUT], <b>string</b> rezultatOpis [OUT])		
Parameter	Tip parametra	Obveznost parametra / vhodni - izhodni	Opis parametra
[]idPrevozniki	integer, null	DA / vhodni	Identifikatorji prevoznikov; če je NULL sledi vračanje vseh registriranih zapisov
[]prevozniki	<b>Prevoznik</b>	- / izhodni	Seznam s podatkovnimi strukturami Prevoznik
rezultat	integer	- / izhodni	Tip kode odgovora
rezultatOpis	string	- / izhodni	Opis tipa odgovora

rezultat	rezultatOpis	
-721	Invalid input data	Identifikator ne obstaja.
0	ok	

### 15.1.2 GETPOSTAJALISCA

Metoda GetPostajalisca vrne seznam tistih podatkovnih struktur Postajalisce, katerih identifikatorji so navedeni v polju idPostajalisca. Če je argument idPostajalisca enak NULL, metoda vrne vse zapise.

Deklaracija metode	int <b>GetPostajalisca</b> ( <b>int</b> []idPostajalisca [IN], <b>int</b> iReturnMode [IN], <b>Postajalisce</b> []postajalisca [OUT], <b>int</b> rezultat [OUT], <b>string</b> rezultatOpis [OUT])		
Parameter	Tip parametra	Obveznost parametra / vhodni - izhodni	Opis parametra
[]idPostajalisca	integer, null	DA / vhodni	Identifikatorji postajališč, če je NULL sledi vračanje vseh registriranih zapisov
iReturnMode	integer, null	NE / vhodni	Način vračanja zapisov, in sicer: <ul style="list-style-type: none"> <li>1. null ali 0: vračajo se veljavni zapisi s statusom R.</li> <li>2. 1: vračajo se veljavni zapisi s statusom R in P.</li> </ul>
[]postajalisca	<b>Postajalisce</b>	- / izhodni	Seznam s podatkovnimi strukturami Postajalisce
Rezultat	integer	- / izhodni	Tip kode odgovora
rezultatOpis	string	- / izhodni	Opis tipa odgovora

rezultat	rezultatOpis	
-721	Invalid input data	Identifikator ne obstaja.
0	ok	

### 15.1.3 GETPOSTAJALISCA\_PLADENJ

Metoda GetPostajalisca\_Pladenj je identična metodi GetPostajalisca, sicer se uporablja za potrebe **Pladenj** sistema.

Metoda vrača zapise s statusom P in R.

Deklaracija metode	int <b>GetPostajalisca_Pladenj</b> (int []idPostajalisca [IN], <b>Postajalisce_Pladenj</b> []postajalisca [OUT], int rezultat [OUT], string rezultatOpis [OUT])		
Parameter	Tip parametra	Obveznost parametra / vhodni - izhodni	Opis parametra
[]idPostajalisca	integer, null	DA / vhodni	Identifikatorji postajališč, če je NULL sledi vračanje vseh registriranih zapisov
[]postajalisca	<b>Postajalisce_Pladenj</b>	- / izhodni	Seznam s podatkovnimi strukturami Postajalisce
Rezultat	integer	- / izhodni	Tip kode odgovora
rezultatOpis	string	- / izhodni	Opis tipa odgovora

rezultat	rezultatOpis	
-721	Invalid input data	Identifikator ne obstaja.
0	ok	

### 15.1.4 GETRELACIJE

Metoda GetRelacije vrne seznam tistih podatkovnih struktur Relacija, katerih identifikatorji so navedeni v polju idRelacije. Če je argument idRelacije enak NULL, metoda vrne vse zapise.

Deklaracija metode	int <b>GetRelacije</b> (int []idRelacije [IN], Relacija []relacije [OUT], int rezultat [OUT], string rezultatOpis [OUT])		
Parameter	Tip parametra	Obveznost parametra / vhodni - izhodni	Opis parametra
[]idRelacije	integer, null	DA / vhodni	Identifikatorji relacij, če je NULL sledi vračanje vseh registriranih zapisov
[]relacije	Relacija	- / izhodni	Seznam s podatkovnimi strukturami Relacija
rezultat	integer	- / izhodni	Tip kode odgovora
rezultatOpis	string	- / izhodni	Opis tipa odgovora

rezultat	rezultatOpis	
-721	Invalid input data	Identifikator ne obstaja.
0	ok	

### 15.1.5 GETPOSTAJNETOCKE

Metoda GetPostajneTocke vrne seznam tistih podatkovnih struktur PostajnaTocka, katerih identifikatorji so navedeni v polju idPostajneTocke. Če je argument idPostajneTocke enak NULL, metoda vrne vse zapise.

Deklaracija metode	int <b>GetPostajneTocke</b> ( <b>int</b> []idPostajneTocke [IN], <b>int</b> iReturnMode [IN], <b>PostajnaTocka</b> []postajneTocke [OUT], <b>int</b> rezultat [OUT], <b>string</b> rezultatOpis [OUT])		
<b>Parameter</b>	<b>Tip parametra</b>	<b>Obveznost parametra / vhodni – izhodni</b>	<b>Opis parametra</b>
[]idPostajneTocke	integer, null	DA / vhodni	Identifikatorji postajnih točk, če je NULL sledi vračanje vseh registriranih zapisov
iReturnMode	integer, null	NE / vhodni	Način vračanja zapisov, in sicer: <ul style="list-style-type: none"> <li>1. null ali 0: vračajo se veljavni zapisi s statusom R.</li> <li>2. 1: vračajo se veljavni zapisi s statusom R in P.</li> </ul>
[]postajneTocke	<b>PostajnaTocka</b>	- / izhodni	Seznam s podatkovnimi strukturami PostajnaTocka
rezultat	integer	- / izhodni	Tip kode odgovora
rezultatOpis	string	- / izhodni	Opis tipa odgovora

rezultat	rezultatOpis	
-721	Invalid input data	Identifikator ne obstaja.
0	ok	

### 15.1.6 GETCONE

Metoda GetCone vrne seznam tistih podatkovnih struktur Cona, katerih identifikatorji so navedeni v polju idCone. Če je argument idCone enak NULL, metoda vrne vse zapise.

Deklaracija metode	int <b>GetCone</b> (int []idCone [IN], Cona []cone [OUT], int rezultat [OUT], string rezultatOpis [OUT])		
Parameter	Tip parametra	Obveznost parametra / vhodni – izhodni	Opis parametra
[]idCone	integer, null	DA / vhodni	Identifikatorji con, če je NULL sledi vračanje vseh registriranih zapisov
[]cona	Cona	- / izhodni	Seznam s podatkovnimi strukturami Cona
rezultat	integer	- / izhodni	Tip kode odgovora
rezultatOpis	string	- / izhodni	Opis tipa odgovora

rezultat	rezultatOpis	
-721	Invalid input data	Identifikator ne obstaja.
0	V redu	

### 15.1.7 GETLOKACIJE

Metoda GetLokacije vrne seznam tistih podatkovnih struktur Lokacija, katerih identifikatorji so navedeni v polju idLokacije. Če je argument idLokacije enak NULL, metoda vrne vse zapise.

Deklaracija metode	int <b>GetLokacije</b> (int []idLokacije [IN], <b>Lokacija</b> []lokacije [OUT], int rezultat [OUT], string rezultatOpis [OUT])		
Parameter	Tip parametra	Obveznost parametra / vhodni – izhodni	Opis parametra
[]idLokacije	integer, null	DA / vhodni	Identifikatorji lokacij, če je NULL sledi vračanje vseh registriranih zapisov
[]lokacije	<b>Lokacija</b>	- / izhodni	Seznam s podatkovnimi strukturami Lokacija
rezultat	integer	- / izhodni	Tip kode odgovora
rezultatOpis	string	- / izhodni	Opis tipa odgovora

rezultat	rezultatOpis	
-721	Invalid input data	Identifikator ne obstaja.
0	V redu	



### 15.1.8 GETLINIJSKIODESKI

Metoda GetLinijskiOdseki vrne seznam tistih podatkovnih struktur LinijskiOdsek, katerih identifikatorji so navedeni v polju idLinijskiOdseki. Če je argument idLinijskiOdseki enak NULL, metoda vrne vse zapise brez geometrije. V primeru, da atribut *linijskiOdseki* vsebuje seznam identifikatorjev linijskih odsekov, metoda zanje vrača tudi geometrijske podatke.

Deklaracija metode	int <b>GetLinijskiOdseki</b> (int []idLinijskiOdseki [IN], int iReturnMode [IN], <b>LinijskiOdsek</b> []linijskiOdseki [OUT], int rezultat [OUT], <b>string</b> rezultatOpis [OUT])		
Parameter	Tip parametra	Obveznost parametra / vhodni - izhodni	Opis parametra
[]idLinijskiOdseki	integer, null	DA / vhodni	Identifikatorji linijskih odsekov, če je NULL sledi vračanje vseh registriranih zapisov brez geometrije. Za konkretno navedene linijske odseke metoda vrača tudi geometrijo.
iReturnMode	integer, null	NE / vhodni	Način vračanja zapisov, in sicer: <ol style="list-style-type: none"> <li>1. null ali 0: vračajo se veljavni zapisi s statusom R.</li> <li>2. 1: vračajo se veljavni zapisi s statusom R in P.</li> </ol>
[]linijskiOdseki	<b>LinijskiOdsek</b>	- / izhodni	Seznam s podatkovnimi strukturami LinijskiOdsek
rezultat	integer	- / izhodni	Tip kode odgovora
rezultatOpis	string	- / izhodni	Opis tipa odgovora

rezultat	rezultatOpis	
-721	Invalid input data	Identifikator ne obstaja.
0	V redu	

### 15.1.9 GETCASOVNIREZIMI

Metoda GetCasovniRezimi vrne podatkovno strukturo za časovni režim linijskega odseka. Omogoča prilagajanje časa vožnje, ki je potreben za pot po linijskem odseku. Obdobje se določi z datumom od-do ter časom od-do.

Deklaracija metode	int <b>GetCasovniRezimi</b> (int []idCasovniRezimi [IN], <b>CasovniRezim</b> []casovniRezimi [OUT], int rezultat [OUT], string rezultatOpis [OUT])		
Parameter	Tip parametra	Obveznost parametra / vhodni - izhodni	Opis parametra
[]idCasovniRezimi	integer, null	DA / vhodni	Identifikator časovnega režima, če je NULL sledi vračanje vseh registriranih zapisov
[]casovniRezimi	<b>CasovniRezim</b>	- / izhodni	Seznam s podatkovnimi strukturami časovnega režima
rezultat	integer	- / izhodni	Tip kode odgovora
rezultatOpis	string	- / izhodni	Opis tipa odgovora

rezultat	rezultatOpis	
-721	Invalid input data	Identifikator ne obstaja.
0	V redu	

### 15.1.10 GETPRESTOPNETOCKE

Metoda GetPrestopneTocke vrne seznam tistih podatkovnih struktur PrestopnaTocka, katerih identifikatorji so navedeni v polju idPrestopneTocke. Če je argument idPrestopneTocke enak NULL, metoda vrne vse zapise.

Deklaracija metode	int <b>GetPrestopneTocke</b> ( <b>int</b> []idPrestopneTocke [IN], <b>PrestopnaTocka</b> []prestopneTocke [OUT], <b>int</b> rezultat [OUT], <b>string</b> rezultatOpis [OUT])		
Parameter	Tip parametra	Obveznost parametra / vhodni - izhodni	Opis parametra
[]idPrestopneTocke	integer, null	DA / vhodni	Identifikatorji prestopnih točk, če je NULL sledi vračanje vseh registriranih zapisov
[]prestopneTocke	<b>PrestopnaTocka</b>	- / izhodni	Seznam s podatkovnimi strukturami PrestopnaTocka
rezultat	integer	- / izhodni	Tip kode odgovora
rezultatOpis	string	- / izhodni	Opis tipa odgovora

rezultat	rezultatOpis	
-721	Invalid input data	Identifikator ne obstaja.
0	V redu	

### 15.1.11 GETPRESTOPI

Metoda GetPrestopi vrne seznam tistih podatkovnih struktur Prestop, katerih identifikatorji so navedeni v polju idPrestopi. Če je argument idPrestopi enak NULL, metoda vrne vse zapise.

Deklaracija metode	int <b>GetPrestopi</b> (int []idPrestopi [IN], <b>Prestop</b> []prestopi [OUT], int rezultat [OUT], string rezultatOpis [OUT])		
Parameter	Tip parametra	Obveznost parametra / vhodni - izhodni	Opis parametra
[]idPrestopi	integer, null	DA / vhodni	Identifikatorji prestopov, če je NULL sledi vračanje vseh registriranih zapisov
[]prestopi	<b>Prestop</b>	- / izhodni	Seznam s podatkovnimi strukturami Prestop
rezultat	integer	- / izhodni	Tip kode odgovora
rezultatOpis	string	- / izhodni	Opis tipa odgovora

rezultat	rezultatOpis	
-721	Invalid input data	Identifikator ne obstaja.
0	V redu	

### 15.1.12 GETREZIMI

Metoda GetRezimi vrne seznam tistih podatkovnih struktur RezimExt, katerih identifikatorji so navedeni v polju idRezimi. Če je argument idRezimi enak NULL, metoda vrne vse zapise.

Deklaracija metode	int <b>GetRezimi</b> (int []idRezimi [IN], RezimExt []rezimi [OUT], int rezultat [OUT], string rezultatOpis [OUT])		
Parameter	Tip parametra	Obveznost parametra / vhodni - izhodni	Opis parametra
[]idRezimi	integer, null	DA / vhodni	Identifikatorji režimov, če je NULL sledi vračanje vseh registriranih zapisov
[]rezimi	RezimExt	- / izhodni	Seznam z razširjenimi podatkovnimi strukturami RezimExt
rezultat	integer	- / izhodni	Tip kode odgovora
rezultatOpis	string	- / izhodni	Opis tipa odgovora

rezultat	rezultatOpis	
-721	Invalid input data	Identifikator ne obstaja.
0	V redu	

### 15.1.13 GETREZIMIZAPREVOZNIKA

Metoda GetRezimiZaPrevoznika vrne seznam režimov (razširjena struktura) za prevoznika, ki vsebuje tudi termine in intervale. Funkcija vrača samo režime, ki so uporabljeni v registriranih in veljavnih vozniških redih prevoznika običajne certifikate (terminali) in vse registrirane za virtualne certifikate (zaledni sistemi).

Deklaracija metode	int <b>GetRezimiZaPrevoznika</b> (int idPrevoznik [IN], int []idRezimi, <b>RezimExt</b> []rezimi [OUT], int rezultat [OUT], string rezultatOpis [OUT])		
Parameter	Tip parametra	Obveznost parametra / vhodni - izhodni	Opis parametra
idPrevoznik	integer	DA / vhodni	Identifikator prevoznika
[]idRezimi	integer, null	NE / vhodni	Seznam identifikatorjev režimov za katere funkcija vrača režime; če je NULL sledi vračanje vseh registriranih zapisov prevoznika. Rezultat vsebuje tudi splošne režime, tj. režimi upravljavca.
[]rezimi	<b>RezimExt</b>	- / izhodni	Seznam režimov z vsebovanima podstrukturama RezimTermin in RezimInterval
rezultat	integer	- / izhodni	Tip kode odgovora
rezultatOpis	string	- / izhodni	Opis tipa odgovora

rezultat	rezultatOpis	
-721	Invalid input data	[-1] Neustrezni identifikator za prevoznika.
		[-2] Neustrezni identifikatorji režimov.
0	V redu	

### 15.1.14 GETVOZNIREDIZAPREVOZNIKA

Metoda GetVozniRediZaPrevoznika vrne seznam voznih redov (razširjena struktura) za prevoznika, ki vsebuje tudi vožnje, opise voženj in režime. Terminali bodo prejeli vozne rede, ki so registrirani in veljavni v trenutki klica metode, medtem kot naprave z virtualnimi certifikati (tj. zaledni sistemi prevoznikov), pa bodo prejeli vse vozni rede, ki so registrirani (veljavne in vse, ki še bodo veljavni).

Deklaracija metode	int <b>GetVozniRediZaPrevoznika</b> (int idPrevoznik [IN], int []idVozniRedi [IN], int []idVoznje [IN], bool isExtStructure [IN], <b>VozniRed*</b> []vozniRedi [OUT], int rezultat [OUT], string rezultatOpis [OUT])		
Parameter	Tip parametra	Obveznost parametra / vhodni - izhodni	Opis parametra
idPrevoznik	integer	DA / vhodni	Identifikator prevoznika
[]idVozniRedi	integer, null	NE / vhodni	Seznam identifikatorjev voznih redov, za katere funkcija vrača voznoredne podatke, če je NULL sledi vračanje vseh registriranih zapisov prevoznika
[]idVoznje	integer, null	NE / vhodni	Seznam identifikatorjev voženj, za katere funkcija vrača voznoredne podatke, če je NULL sledi vračanje vseh registriranih zapisov prevoznika
isExtStructure	boolean	NE / vhodni	Atribut je opcijski in določa katera struktura voznega reda se bo vračala, in sicer:  - isExtStructure = true; vrača se VozniRedExt  - isExtStructure = false; vrača se VozniRedEmb
[]vozniRedi	<b>VozniRed*</b>	- / izhodni	Seznam voznih redov z vsebovanima podstrukturama VoznjaExt in VoznjaOpis
rezultat	Integer	- / izhodni	Tip kode odgovora
rezultatOpis	String	- / izhodni	Opis tipa odgovora

\*vrača se podatkovna struktura **VozniRedExt** ali **VozniRedEmb** odvisno od atributa isExtStructure, kjer velja:

- isExtStructure = true; vrača se **VozniRedExt**
- isExtStructure = false; vrača se **VozniRedEmb**

Struktura VozniRedExt je namenjena zalednim sistemom prevoznikov in vrača vse voznoredne podatke. Ker vsi podatki niso potrebni za delovanje terminalov, metoda omogoča vračanje okrnjene strukture, ki se imenuje VozniRedEmb. Slednja je namenjena terminalom na vozilih.

rezultat	rezultatOpis	
-721	Invalid input data	[-1] Neustrezni identifikator za prevoznika.
		[-2] Neustrezni identifikatorji voznih redov.
		[-3] Neustrezni identifikatorji voženj.
0	V redu	



### 15.1.15 *GetVOZNIREDINDEXZAPREVOZNIKA*

Metoda *GetVozniRedIndexZaPrevoznika* vrne seznam voznih redov, ki vsebujejo le identifikatorje za vožnje in opise voženj. Rezultat je odvisen od vrste certifikata. Terminali bodo prejeli vozne rede, ki so registrirani in veljavni v trenutki klica metode, medtem kot naprave z virtualnimi certifikati (tj. zaledni sistemi prevoznikov), pa bodo prejeli vse vozni rede, ki so registrirani (veljavne in vse, ki še bodo veljavni).

Deklaracija metode	int <b>GetVozniRedIndexZaPrevoznika</b> (int idPrevoznik [IN], <b>VozniRedIndex</b> []vozniRedi [OUT], int rezultat [OUT], string rezultatOpis [OUT])		
Parameter	Tip parametra	Obveznost parametra / vhodni - izhodni	Opis parametra
idPrevoznik	integer	DA / vhodni	Identifikator prevoznika
[]vozniRedi	<b>VozniRedIndex</b>	- / izhodni	Seznam voznih redov z vsebovanima podstrukturama VoznjaExt in VoznjaOpis, kjer so navedeni samo identifikatorji
rezultat	integer	- / izhodni	Tip kode odgovora
rezultatOpis	string	- / izhodni	Opis tipa odgovora

rezultat	rezultatOpis	
-721	Invalid input data	[-1] Neustrezni identifikator za prevoznika.
		[-2] Neustrezni identifikatorji voznih redov.
0	V redu	

### 15.1.16 GETPOSTAJNETOCKEZAPREVOZNIKA

Metoda GetPostajneTockeZaPrevoznika vrne seznam postajnih točk, ki so vsebovane v vozni redih prevoznika. Podatkovna struktura postajne točke vsebuje tudi podatke o postajališču, coni in lokaciji.

Deklaracija metode	int <b>GetPostajneTockeZaPrevoznika</b> (int idPrevoznik [IN], <b>PostajnaTockaExt</b> []postajneTocke [OUT], int rezultat [OUT], string rezultatOpis [OUT])		
Parameter	Tip parametra	Obveznost parametra / vhodni - izhodni	Opis parametra
idPrevoznik	integer	DA / vhodni	Identifikator prevoznika
[]postajneTocke	<b>PostajnaTockaExt</b>	- / izhodni	Seznam postajnih točk, ki vsebuje tudi podatke o postajališču, coni in lokaciji. V seznamu so le postajne točke, ki so vsebovane v vozni redih prevoznika
Rezultat	integer	- / izhodni	Tip kode odgovora
rezultatOpis	string	- / izhodni	Opis tipa odgovora

rezultat	rezultatOpis	
-721	Invalid input data	[-1] Neustrezní identifikator za prevoznika.
		[-2] Neustrezní identifikatorji postajnih točk.
0	V redu	

### 15.1.17 SETDELOVNAHALOGAZAPREVOZNIKA

Metoda SetDelovnaNalogaZaPrevoznika omogoča vpis delovnih nalog v sistem IJPP. V kolikor gre za vpis delovne naloge z že obstoječo številko delovne naloge, tedaj sistem preveri morebitno časovno prekrivanje. V kolikor se vpisana delovna naloga časovno ne prekriva z ostalimi delovnimi naloga, ki imajo enako številko, tedaj se izvede vpis delovne naloge. Uporabnik mora sam poskrbeti za ustrezna obdobja veljavnosti (glej metodo SpremeneniVeljavnostDelovneNaloge).

Časovni konflikti med vožnjami se ne preverjajo.

Deklaracija metode	int <b>SetDelovnaNalogaZaPrevoznika</b> ( <b>int</b> idPrevoznik [IN], <b>DelovnaNaloga</b> delovnaNaloga [IN], <b>int</b> rezultat [OUT], <b>string</b> rezultatOpis [OUT])		
Parameter	Tip parametra	Obveznost parametra / vhodni - izhodni	Opis parametra
idPrevoznik	integer	DA / vhodni	Identifikator prevoznika
delovnaNaloga	<b>DelovnaNaloga</b>	DA / vhodni	Delovna naloga za prevoznika. Pri vpisovanju posamezne delovne naloge se izvede validacija identifikatorjev voženj, ki morajo pripadati veljavnih in registriranim voznim redom prevoznika.
rezultat	integer	- / izhodni	Tip kode odgovora
rezultatOpis	string	- / izhodni	Opis tipa odgovora

rezultat	rezultatOpis	
-721	Invalid input data	[-1] Manjka identifikator delovne naloge.
		[-2] Neustrezní identifikator prevoznika (manjka ali ne obstaja).
		[-3] Neustrezna številka delovne naloge*.
		[-4] Neustrezno določena atributa veljavnostDo in veljavnostDo**.
		[-5] Neustrezno določene vezane vožnje***.
0	V redu	

\* Delovne naloge z enakimi številkami (tj. številkaDelovneNaloge) se časovno ne smejo prekrivati za posameznega prevoznika.

**\*\* Atribut veljavnostDo mora biti večji od atributa veljavnostOd.**

**\*\*\* Vezane vožnje se morajo stikati v istih postajnih točka, kar pomeni, da je končna postajna točka predhodne vožnje hkrati tudi začetna postajna vožnja naslednje vezane vožnje.**

### 15.1.18 *SPREMENI VELJAVNOST DELOVNE NALOGE*

Metoda omogoča spreminjanje obdobja veljavnosti delovne naloge. Pri spremembi veljavnosti, se izdela kopija originala, prvotni zapisa pa se arhivira.

Deklaracija metode	int <b>SpremeniVeljavnostDelovneNaloge</b> (int idPrevoznik [IN], <b>DelovnaNaloga</b> delovnaNaloga [IN], <b>datetime</b> newVeljavnostOd [IN], <b>datetime</b> newVeljavnostDo [IN], <b>int</b> rezultat [OUT], <b>string</b> rezultatOpis [OUT])		
Parameter	Tip parametra	Obveznost parametra / vhodni - izhodni	Opis parametra
idPrevoznika	integer	DA / vhodni	Identifikator prevoznika
delovnaNaloga	<b>DelovnaNaloga</b>	DA / vhodni	Obvezno vpisati: številka delovne naloge, veljavnost od in veljavnost do. Ostali atributi objekta so lahko prazni.
newVeljavnostOd	datetime	DA / vhodni	Nov datum začetka veljavnosti delovne naloge. Primer: 2016-01-01T00:00:00
newVeljavnostDo	datetime	DA / vhodni	Nov datum zaključka veljavnosti delovne naloge. Primer: 2016-01-01T23:59:59
rezultat	integer	- / izhodni	Tip kode odgovora
rezultatOpis	string	- / izhodni	Opis tipa odgovora

rezultat	rezultatOpis	
-721	Invalid input data	[-1] Delovna naloga ne obstaja
		[-2] Neustrezní identifikator prevoznika (manjka ali ne obstaja).
		[-3] Atribut veljavnostDo mora biti večji od atributa veljavnostOd
0	V redu	Podatki delovne naloge so se uspešno zapisali

### 15.1.19 GETNEARESTSTATIONS

Metoda vrača najbližja postajališča za določeni **HSMID** (Identifikator hišne številke, postavljen s strani GURSa).

Privzeta konfiguracija metode je:

- Največ 10 postajališč
- Maksimalni radius za navadna postajališča je 7km
- Maksimalni radius za glavna postajališča je 7km

Deklaracija metode	int <b>GetNearestStations</b> (int HSMID [IN], StationDetails []stations [OUT], int rezultat [OUT], string rezultatOpis [OUT])		
Parameter	Tip parametra	Obveznost parametra / vhodni - izhodni	Opis parametra
HSMID	integer	DA / vhodni	Identifikator hišne številke
[]stations	StationDetails	- / izhodni	Seznam postajališč
rezultat	integer	- / izhodni	Tip kode odgovora
rezultatOpis	string	- / izhodni	Opis tipa odgovora

rezultat	rezultatOpis	
0	V redu	Podatki delovne naloge so se uspešno zapisali

### 15.1.20 GETNEARESTSTATIONS\_PLADENJ

Metoda GetNearestStations\_Pladenj je identična metodi GetNearestStations, sicer se uporablja za potrebe sistema **Pladenj**. Ima identično privzeto konfiguracijo.

Deklaracija metode	int <b>GetNearestStations_Pladenj</b> (int HSMID [IN], <b>StationDetails_Pladenj</b> []stations [OUT], int rezultat [OUT], string rezultatOpis [OUT])		
Parameter	Tip parametra	Obveznost parametra / vhodni - izhodni	Opis parametra
HSMID	integer	DA / vhodni	Identifikator hišne številke
[]stations	<b>StationDetails_Pladenj</b>	- / izhodni	Seznam postajališč
rezultat	integer	- / izhodni	Tip kode odgovora
rezultatOpis	string	- / izhodni	Opis tipa odgovora

rezultat	rezultatOpis	
0	V redu	Podatki delovne naloge so se uspešno zapisali

### 15.1.21 SETLOCATION

Za zbiranje dinamičnih podatkov o lokacijah terminalov se uporabi metoda **SetLocation** (na tem mestu je govora o lokacijah vozil in metoda nima povezave s podatkovno strukturo Lokacija). Pošiljajo se izključno lokacije vozil za IJPP vožnje.

Deklaracija metode	<b>int SetLocation</b> ( <b>Package</b> []mPackages [IN], <b>int</b> iNumOfPackages[IN], <b>int</b> iResults [OUT], <b>string</b> sResultsDescription [OUT], <b>PackageStatus</b> []mPackageStatus [OUT])		
Parameter	Tip parametra	Obveznost parametra / vhodni - izhodni	Opis parametra
mPackages	<b>Package</b> []	DA / vhodni	Seznam paketov podatkov o lokaciji vozila. Seznam mora vsebovati vsaj en paket. Maksimalno število paketov je 1000.
iNumOfPackages	integer	DA / vhodni	Število preostalih še ne poslanih paketov na terminalu. Paketi, ki so v pošiljanju niso všteti. V običajnih razmerah je tako vrednost praviloma 0.
iResult	integer	- / izhodni	Tip kode odgovora
sResultDescription	string	- / izhodni	Opis tipa odgovora
mPackageStatus	<b>PackageStatus</b> []	- / izhodni	Opis tipa odgovora

iResult	sResultDescription	
0	OK	Paketi so bili uspešno sprejeti
-501	Terminal not found.	Terminal ni najden v sistemu
-721	Invalid input data.	Nepravilni vhodni podatki, splošne napake: [-4] Seznam paketov mora vsebovati vsaj en paket. [-5] Maksimalno število paketov je preseženo (max. = 100).



		Napake znotraj paketa so v parametru PackageStatus.
-733	Not all elements were successfull.	Vsi paketi niso bili sprejeti, kateri niso bili sprejeti glej parameter PackageStatus.

## 15.2 DEFINICIJE PODATKOVNIH STRUKTUR – PROMETNI SISTEMI IJPP

### 15.2.1 PREVOZNIK

**Prevoznik** je pravna oseba, ki izvaja prevoz avtobusnih linijskih prevozov ali železniškega prometa.

Opisan je s podatkovno strukturo **Prevoznik**, ki vsebuje naslednje parametre:

Ime spremenljivke	Tip spremenljivke	Opis
idPrevoznik	integer	
oznakaPrevoznika	string, null	
ime	string	
naslov	string	
telefon	string, null	
email	string, null	
URL	string, null	
FareURL	string, null	Spletni naslov cenika.
kooperant	boolean	false – prevoznik, true – kooperant privzeto false
kontaktnaOseba	string, null	
davcnaStevilka	string, null	
maticnaStevilka	string, null	
TRR	string, null	
TRRPriBanki	string, null	

### 15.2.2 POSTAJALIŠČE

**Postajališče** je skupina bližnjih postajnih točk, ki imajo enako ime, ter predstavlja njihovo geografsko težišče. Postajališče je predstavljeno s podatkovno strukturo **Postajalisce**, ki vsebuje naslednje parametre:

Ime spremenljivke	Tip spremenljivke	Opis
idPostajalisce	integer	Identifikator postajališča
sifra	string	
oznaka	string, null	oznaka
ime	string	Ime postajališča
opomba	string, null	
idObcina	integer	
idVrstaPostajalisca	integer	1-Železniško, 2-Cestno
idNaselje	integer, null	
idVrstaUpravljavca	integer	1-Država, 2-Občina
idUpravljavca	integer	Seznam občin po GURS, DRSI, Tujina
idLokacija	integer, null	Identifikator lokacije
geometrija	Geometrija	Točka (x,y), format WGS84
jeAktivno	integer, null	Identifikator ali je postajališče prisotno v trenutno aktivnih vozniških redih določenega tipa:  0 - ni uporabljeno postajališče v nobenem VR 1 - mestni 2 - medkrajevni 3 - mestni in medkrajevni
[]idTipPostajalisca	varbinary, null	To je bitno polje, kjer vsak bit pomeni oznako, Biti pomenijo sledeče:  0x80 – MK - medkrajevno, 0x50 – MEp + PL, 0x40 – MEp – mestno in medkrajevno,

		0x30 - ME + PL, 0x20 – ME - mestno, 0x10 – PL – posebno linijsko
idPredhodnik	Integer, null	Identifikator predhodnega zapisa, tj. zapisa iz katerega se je izvajala sprememba.
veljavnostOd	Datetime, null	Začetni datum veljavnosti postajališča
veljavnostDo	Datetime, null	Končni datum veljavnosti postajališča
idStatus	Integer, null	Status zapisa; zaloga vrednosti: 4 – R (registriran) 13 – P (potrjen); bo registriran z dnem veljavnosti. Zapis je pripravljen za uporabo, a še ni v uporabi.

### 15.2.3 POSTAJALIŠČE\_PLADENJ

**Postajališče\_Pladenj** je podatkovna struktura ki vsebuje podobne (skor identične) podatke kot struktura *Postajališče*, sicer se uporablja za potrebe sistema **Pladenj**.

Ime spremenljivke	Tip spremenljivke	Opis
idPostajalisce	integer	Identifikator postajališča
sifra	string	
ime	string	Ime postajališča
Veljavnost od	datetime, null	Začetni datum veljavnosti postajališča
Veljavnost do	datetime, null	Končni datum veljavnosti postajališča
opomba	string, null	
idObcina	integer	
idVrstaPostajalisca	integer	1-Železniško, 2-Cestno
idVrstaPrevoza	integer	1-Consko&Relacijski, 2-Conski in 3-Relacijski
idNaselje	integer, null	
idVrstaUpravljavca	integer	1-Država, 2-Občina
idUpravljavec	integer	Seznam občin po GURS
idLokacija	integer, null	Identifikator lokacije
geometrija	Geometrija	Točka (x,y), format WGS84
glavnoPostajalisce	boolean	true – glavno postajališče false – ostala postajališča Privzeto – false
jeAktivno	integer, null	To je bitno polje, kjer vsak bit pomeni oznako, ali je postajališče prisotno v trenutno aktivnih vozniških redih določenega tipa. Biti pomenijo sledeče:

		0x01 – consko postajališče, 0x02 – relacijsko postajališče, 0x04 – mednarodno postajališče, 0x08 – postajališče posebnega linijskega prevoza, 0x10 – mejni prehod, 0x20 - tuje postajališče
[]idTipPostajalisca	integer, null	1 – consko postajališče, 2 – relacijsko postajališče, 3 – mednarodno postajališče, 4 – postajališče posebnega linijskega prevoza, 5 – mejni prehod, 6 - tuje postajališče

#### 15.2.4 RELACIJA

**Relacija** je povezava med dvema zaporednima postajališčema v itinerarju, ki pa fizično nista nujno tudi sosednja. Relacije je predstavljena s podatkovno strukturo **Relacija**, ki vsebuje naslednje parametre:

Ime spremenljivke	Tip spremenljivke	Opis
idRelacija	integer	
idPostajalisceOd	integer	
idPostajalisceDo	integer	

### 15.2.5 POSTAJNA TOČKA

**Postajna točka** je točka, ki predstavlja fizično mesto oziroma težišče območja (peron), kjer potniki (praviloma) vstopajo na vozilo JPP. Postajna točka je predstavljena s podatkovno strukturo

**PostajnaTocka**, ki vsebuje naslednje parametre:

Ime spremenljivke	Tip spremenljivke	Opis
idPostajnaTocka	integer	Identifikator postajne točke
idPostajalisce	integer	Identifikator postajališča
idCona	integer, null	Identifikator cone, kateri pripadajo postajne točke
jeAktivna	integer, null	Identifikator ali je postajališče prisotno v trenutno aktivnih voznih redih določenega tipa:  0 - ni uporabljeno postajališče v nobenem VR 1 - mestni 2 - medkrajevni 3 - mestni in medkrajevni
sifra	string	
oznaka	string, null	
opomba	string, null	
geometrija	Geometrija	Točka (x,y), format WGS84
casPostanka	integer	Postanek izražen v minutah  Privzeto – 0
idPredhodnik	Integer, null	Identifikator predhodnega zapisa, tj. zapisa iz katerega se je izvajala sprememba.
veljavnostOd	Datetime, null	Začetni datum veljavnosti postajne točke
veljavnostDo	Datetime, null	Končni datum veljavnosti postajne točke
idStatus	Integer, null	Status zapisa; zaloga vrednosti:  4 – R (registriran) 13 – P (potrjen); bo registriran z dnem veljavnosti. Zapis je pripravljen za uporabo, a še ni v uporabi.



### 15.2.6 POSTAJNA TOČKA – EXT

Razširjena podatkovna struktura **PostajnaTočkaExt** združuje osnovne attribute o postajni točki, postajališču, coni in lokaciji. Struktura je naslednja:

Ime spremenljivke	Tip spremenljivke	Opis
idPostajnaTočka	integer	Identifikator postajne točke
idPostajnaTočkaExternal	string, null	Identifikator postajne točke, ki se nahaja v zalednem sistemi prevoznika. Ne vrača za kooperanta.
idPostajalisce	integer	Identifikator postajališča, kateremu postajna točka pripada.
idVrstaPrevoza	integer, null	1 – consko & relacijski, 2 – conski, 3 – relacijski
imePostajalisca	string	Ime postajališča, kateremu postajna točka pripada.
idCona	integer, null	Identifikator cone, kateri postajna točka pripada.
idLokacija	integer, null	Identifikator lokacije, kateri postajna točka pripada.
geometrija	Geometrija	Geografska lokacija postajne točke, format WGS84

### 15.2.7 TEHNIČNI ELEMENTI POSTAJNE TOČKE

Za postajno točko so določeni podatki tehničnih elementov zbrani v tabeli

PostajnaTočkaTehnicniElement. Struktura je naslednja:

Ime spremenljivke	Tip spremenljivke	Opis
idPostajnaTočkaTehnicniElement	integer	Identifikator tehničnega elementa postajne točke
idPostajnaTočka	integer	Identifikator postajne točke
idTehnicniElement	integer	Identifikator tehničnega elementa. Zaloga vrednosti:  1 – Prometni znak III-54 2 – Talna označba 3 – Nadstrešnica 4 – Čakališče 5 – Ima vozni red 6 – Je v naselju 7 – Je na vozišču 8 – Je širina vozišča min. 5,5 m 9 – Je pregledno naprej 10 – Je pregledno nazaj

### 15.2.8 GEOMETRIJA

Podatkovna struktura **Geometrija** vsebuje x in y koordinati za opis točke v ravnini. Struktura je naslednja:

Ime spremenljivke	Tip spremenljivke	Opis
X	double	X koordinata
Y	double	Y koordinata

### 15.2.9 CONA

Entiteta **Cona** združuje postajne točke, ki se nahajajo znotraj določenega geografskega območja.

Cona je predstavljena s podatkovno strukturo **Cona**, ki vsebuje naslednje parametre:

Ime spremenljivke	Tip spremenljivke	Opis
idCona	integer	Identifikator cone
ime	string	Ime cone
opis	string, null	Opis cone
[]geometrija	Geometrija	Lomljenka, sestavljena iz seznama točk (x,y), format WGS84
idLokacija	integer	Identifikator lokacije

### 15.2.10 LOKACIJA

Entiteta **Lokacija** združuje cone, glede na pripadajočo tarifo. Lokacija je predstavljena s podatkovno strukturo **Lokacija**, ki vsebuje naslednje parametre:

Ime spremenljivke	Tip spremenljivke	Opis
idLokacija	integer	Identifikator lokacije
ime	string	Ime lokacije
tipLokacije	int	0 - mestna, 1- medkrajevna, 2 - splošna

### 15.2.11 LINIJSKI ODSEK

**Linijski odsek** je usmerjena povezava med dvema zaporednima postajnama točkama v itinerarju, ki pa fizično nista nujno tudi sosednja, in poteka po osi ceste oz. proge. Linijski odsek je predstavljen s podatkovno strukturo **LinijskiOdsek**, ki vsebuje naslednje parametre:

Ime spremenljivke	Tip spremenljivke	Opis
idLinijskiOdsek	integer	Identifikator linijskega odseka
idPostajnaTockaOd	integer	Začetna postajna točka
idPostajnaTockaDo	integer	Končna postajna točka
dolzinaAVRIS	double, null	Administrativna dolžina, ki pa ni nujno enaka kot dejanska dolžina na terenu.
dolzina	double	Dolžina, ki se upošteva pri nadaljnjih izračunih
dolzina2D	double, null	Dolžina izračunana iz geometrije linijskega odseka
dolzina3D	double, null	Dolžina izračunana iz geometrije linijskega odseka
opomba	string, null	
casVoznje	integer, null	Čas v minutah, ki je potreben za linijskih odsek v normalnih razmerah.
idRelacija	integer	
[.]geometrija	Geometrija, null	Lomljenka, sestavljena iz seznama točk (x,y), format WGS84
idPredhodnik	Integer, null	Identifikator predhodnega zapisa, tj. zapisa iz katerega se je izvajala sprememba.
veljavnostOd	Datetime, null	Začetek veljavnosti zapisa. Z dnem veljavnosti se prične uporaba.
veljavnostDo	Datetime, null	Konec veljavnosti zapisa. Z naslednjim dnem se zaključi uporaba zapisa.
idStatus	Integer, null	Status zapisa; zaloga vrednosti:  4 – R (registriran) 13 – P (potrjen); bo registriran z dnem veljavnosti. Zapis je pripravljen za uporabo, a še ni v uporabi.

### 15.2.12 ČASOVNI REŽIM

**Časovni režim** je neposredno povezan z linijskim odsekom in določa variabilnost časov prevozov v določenih časovnih intervalih (npr. zaradi prometnih konic). Definiran je s podatkovno strukturo **CasovniRezim**, ki vsebuje naslednje parametre:

Ime spremenljivke	Tip spremenljivke	Opis
idCasovniRezim	integer	identifikator
ime	string	Ima oz. naziv časovnega režima
jeGlobalni	boolean	Značka je če je časovni režim globalni. Globalni je lahko sam eden!  Privzeto – false
[]casovniFaktor	CasovniFaktor, null	Seznam časovnih faktorjev, kjer je navedeno časovno obdobje (čas od – do) in časovni faktor.

### 15.2.13 ČASOVNI FAKTOR

**Časovni faktor** je struktura, ki omogoča določitev časovnega faktorja za izbrano časovno obdobje in ima naslednjo strukturo:

Ime spremenljivke	Tip spremenljivke	Opis
idCasovniFaktor	integer	identifikator
idCasovniRezim	integer	Identifikator časovnega režima, ki mu pripada.
casOd	integer	Začetek časovnega obdobja. Število sekund od 00:00:00.
casDo	integer	Konec časovnega obdobja. Število sekund od 00:00:00.
naziv	string	Naziv faktorja
opis	string, null	Kratek opis časovnega faktorja
faktor	float	Faktor časa potovanja v določenem časovnem obdobju

#### 15.2.14      *PRESTOPNA TOČKA*

**Prestopna točka** je pomembnejše postajališče ali skupina bližnjih postajališč, kjer potnik lahko zamenja linijo ali prevozno sredstvo. Definirana je s podatkovno strukturo **PrestopnaTocka**, ki vsebuje naslednje parametre:

Ime spremenljivke	Tip spremenljivke	Opis
idPrestopnaTocka	integer	Identifikator prestopne točke
oznaka	string, null	
ime	string	
opomba	string, null	

### 15.2.15 *PRESTOP*

**Prestop** je mesto zamenjave linije ali prevoznega sredstva znotraj istega postajališča ali pešpot med dvema postajališčema znotraj iste prestopne točke, ki je potrebna za zamenjavo linije ali prevoznega sredstva. Definiran je s podatkovno strukturo **Prestop**, ki vsebuje naslednje parametre:

Ime spremenljivke	Tip spremenljivke	Opis
idPrestop	integer	Identifikator prestopa
idPrestopnaTocka	integer	Identifikator prestopne točke
idPostajalisceOd	integer	
idPostajalisceDo	integer	
intervalPrestopanja	integer	Izraženo v minutah
opomba	string, null	



### 15.2.16 VOZNI REDI – INDEX

Podatkovna struktura **VozniRedIndex** vsebuje le identifikatorje strukture VoznjaIndex.

Ime spremenljivke	Tip spremenljivke	Opis
idVozniRed	integer	Identifikator voznega reda.
veljavnostOd	string	Začetek veljavnosti voznega reda, primer: 2016-01-01T00:00:00
veljavnostDo	string	Konec veljavnosti voznega reda, primer: 2016-01-01T00:00:00
[]voznje	VoznjaIndex	Seznam indeksnih struktur voženj, ki pripadajo voznemu redu

### 15.2.17      *VOŽNJA – INDEX*

Podatkovna struktura **VoznjaIndex** vsebuje le identifikatorje opisov voženj.

Ime spremenljivke	Tip spremenljivke	Opis
idVoznja	integer	Identifikator vožnje
idRezim	integer	Identifikator režima

### 15.2.18 VOZNI RED – EXT

Razširjena podatkovna struktura **VozniRedExt** vsebuje zraven parametrov voznega reda tudi polje razširjene podatkovne strukture VoznjaExt.

Ime spremenljivke	Tip spremenljivke	Opis
idVozniRed	integer	Identifikator voznega reda
idVozniRedExternal	string, null	Identifikator voznega reda, ki se nahaja v zalednem sistemu prevoznika
idPredhodnik	integer, null	Identifikator predhodno veljavnega zapisa. Pri ukinjanju in spreminjanju se obstoječi podatki ne brišejo, temveč se naredi kopija kateri se nato spremenijo podatki.
idPrevoznik	integer	Identifikator prevoznika
idVrstaPrevoza	integer	2-conski, 3-relacijski; v opuščanju.
idTipPrevoza	integer, null	4 - Medkrajevni 5 - Mestni 9 - Posebni linijski 10 - Covid19
veljavnostOd	string	Začetek veljavnosti voznega reda, primer: 2016-01-01T00:00:00
veljavnostDo	string	Konec veljavnosti voznega reda, primer: 2016-01-01T00:00:00
stevilka	integer	
verzija	integer, null	Privzeto – 0
idPostajalisceOd	integer	
postajalisceVia	string, null	
idPostajalisceDo	integer	
opomba	string, null	

[ ]voznje	VoznjaExt	Seznam voženj voznega reda, s pripadajočimi opisi (čas in prostor)
-----------	-----------	--

### 15.2.19 VOŽNJA – EXT

Razširjena podatkovna struktura **VoznjaExt** vsebuje zraven parametrov vožnje tudi polje podatkovne strukture **VoznjeOpisExt**.

Ime spremenljivke	Tip spremenljivke	Opis
idVoznja	integer	Identifikator vožnje
idVoznjaExternal	string, null	Identifikator vožnje, ki se nahaja v zalednem sistemu prevoznika
idPredhodnik	integer, null	Identifikator predhodno veljavnega zapisa. Pri ukinjanju in spreminjanju se obstoječi podatki ne brišejo, temveč se naredi kopija kateri se nato spremenijo podatki.
idVozniRed	integer	Identifikator voznega reda.
oznaka	string, null	Oznaka vožnje, lahko se uporabi tudi kratko ime.
oznakaPoti	string, null	
stevilka	integer, null	Številka vožnje
idRezim	integer	Identifikator režima
idKooperant	integer, null	Identifikator prevoznika, ki opravlja določeno vožnjo, a nima koncesije
[]voznjeOpisi	VoznjaOpisExt	Seznam vseh opisov vožnje (čas in prostor)

### 15.2.20 VOŽNJA OPIS – EXT

Opis posamezne vožnje omogoča podatkovna struktura **VoznjaOpisExt**, kjer so nanizane zaporedne postajne točke s časi prihodov in odhodov. Struktura **VoznjaOpisExt** vsebuje naslednje parametre:

Ime spremenljivke	Tip spremenljivke	Opis
idVoznjaOpis	integer	Identifikator opisa vožnje.
idPredhodnik	integer, null	Identifikator predhodno veljavnega zapisa. Pri ukinjanju in spreminjanju se obstoječi podatki ne brišejo, temveč se naredi kopija kateri se nato spremenijo podatki.
idVozniRed	integer	Identifikator voznega reda
idVoznja	integer	Identifikator vožnje
zapStPostajalisca	integer	Zaporedna številka postajane točke v itinerarju.
idPostajnaTocka	integer	Postanek prevoznega sredstva na postajališču, kjer potniki vstopajo.
casPrihoda	integer	Zapis časa se določi kot število sekund od 00:00:00. Izjema je čas prihoda za prvo postajno točko v itinerarju (tedaj je null).
casOdhoda	integer	Zapis časa se določi kot število sekund od 00:00:00. Izjema je čas odhoda iz zadnje postajne točke v itinerarju (tedaj je null).
postanek	boolean, null	Določa postanke vozila (true – ustavlja, false – ne ustavlja)  Privzeto – true
polnoc	boolean, null	V kolikor vožnja traja čez polnoč, npr. pričetek ob 22:45 in konec ob 02:32. Zapisi, ki so čez polnoč imajo značko polnoč = true. Ostali imajo vrednost false.  Privzeto – false

### 15.2.21 VOZNI RED – EMB

Osnovna podatkovna struktura **VozniRedEmb** vsebuje zraven parametrov voznega reda tudi polje osnovne podatkovne strukture VoznjaEmb.

Ime spremenljivke	Tip spremenljivke	Opis
idVozniRed	integer	
idVrstaPrevoza	integer	2-conski, 3-relacijski; v opuščanju
veljavnostOd	string	Začetek veljavnosti voznega reda, primer: 2016-01-01T00:00:00
veljavnostDo	string	Konec veljavnosti voznega reda, primer: 2016-01-01T00:00:00
[]voznje	VoznjaEmb	Seznam voženj voznega reda, s pripadajočimi opisi (čas in prostor)

### 15.2.22 VoŽNJA – EMB

Osnovna podatkovna struktura **VoznjaEmb** vsebuje zraven parametrov vožnje tudi polje podatkovne strukture **VoznjeOpisEmb**.

Ime spremenljivke	Tip spremenljivke	Opis
idVoznja	integer	
oznaka	string, null	Oznaka vožnje, lahko se uporabi tudi kratko ime.
idRezim	integer	Identifikator režima
idKooperant	integer, null	Identifikator prevoznika, ki opravlja določeno vožnjo, a nima koncesije
[]voznjeOpisi	VoznjaOpisEmb	Seznam vseh opisov vožnje (čas in prostor)



### 15.2.23 VOŽNJA – GRP

Osnovna podatkovna struktura **VoznjaGrp** vsebuje zraven parametrov vožnje dodatne parametre, ki vožnjo povezujejo z ostalimi vožnjami, npr. vezane vožnje, podaljšanje vožnje itd. Ti parametri so definirani v podatkovni strukturi **VoznjaGrp**. Za vsako vožnjo (idVoznja) se lahko določijo dodatne lastnosti, kot so vezanje ali podaljševanje voženj. Povezave med vožnjami se določijo s parametrom idPredhodnaVoznja, tip povezave pa z idTip. Privzeto je tip 1, kar pomeni za vožnja ni vezana ali podaljšana.

Ime spremenljivke	Tip spremenljivke	Opis
idVoznja	integer (32 bitov)	Identifikator vožnje
idPredhodnaVoznja	integer (32 bitov), null	Identifikator predhodne vožnje
idTip	integer (32 bitov)	1 – običajna vožnja, 2 – vezana vožnja  Privzeto – 1.

### 15.2.24 VOŽNJA OPIS – EMB

Opis posamezne vožnje omogoča osnovna podatkovna struktura **VoznjaOpisEmb**, kjer so nanizane zaporedne postajne točke s časi prihodov in odhodov. Struktura **VoznjaOpisEmb** vsebuje naslednje parametre:

Ime spremenljivke	Tip spremenljivke	Opis
idVoznjaOpis	integer	Identifikator opisa vožnje
zapStPostajalisca	integer	Zaporedna številka postajne točke v itinerarju
idPostajnaTocka	integer	Postanek prevoznega sredstva na postajališču, kjer potniki vstopajo.
casPrihoda	integer	Zapis časa se določi kot število sekund od 00:00:00
casOdhoda	integer	Zapis časa se določi kot število sekund od 00:00:00
postanek	boolean, null	Določa postanke vozila (true – ustavlja, false – ne ustavlja)  Privzeto – true
polnoc	boolean, null	V kolikor vožnja traja čez polnoč, npr. pričetek ob 22:45 in konec ob 02: 32. Zapisi, ki so čez polnoč imajo značko polnoč = true. Ostali imajo vrednost false.  Privzeto – false

### 15.2.25 REŽIM – EXT

**RezimExt** določa posebne časovne pogoje za izvajanje JPP in je definiran s podatkovno strukturo **RezimExt** s pripadajočima podstrukturama **RezimInterval** in **RezimTermin**, ki vsebuje naslednje parametre:

Ime spremenljivke	Tip spremenljivke	Opis
idRezim	integer	Identifikator režima
idPredhodnik	integer, null	Identifikator predhodno veljavnega zapisa. Pri ukinjanju in spreminjanju se obstoječi podatki ne brišejo, temveč se naredi kopija kateri se nato spremenijo podatki.
oznaka	string	
opis	string	
privzetiDelovniVektor	integer	Atribut določa privzete vrednosti delovnega vektorja, in sicer: 0 – vse vrednosti delovnega vektorja so 0 (vožnja se NE izvaja), 1 – vse vrednosti delovnega vektorja so 1 (vožnja se izvaja).
[ ]termini	RezimTermin, null	Seznam terminov, ki določajo vrednosti delovnega vektorja za določen dan.
[ ]intervali	RezimInterval, null	Seznam intervalov, ki določajo vrednosti delovnega vektorja za določeno obdobje.

### 15.2.26 REŽIM TERMIN

**Režim termin** določa datume za katere se določi ali se vožnja izvaja ali ne. **RežimTermin** ima naslednje parametre:

Ime spremenljivke	Tip spremenljivke	Opis
idRezimTermin	integer	Identifikator termina
idRezim	integer	Identifikator režima, ki mu obravnavani termin pripada.
termin	string	Datum termina, primer 2016-01-04T00:00:00
akcija	integer	0-ne vozi, 1-vozi

### 15.2.27 REŽIM INTERVAL

**Režim interval** omogoča določitev katere dneve v izbranem intervalu se izvaja vožnja. **RežimInterval** ima naslednje parametre:

Ime spremenljivke	Tip spremenljivke	Opis
idRezimInterval	integer	
idRezim	integer	Identifikator režima, ki mu obravnavani termin pripada.
ponedeljek	boolean	Določijo se posamezni dnevi (true ali false) za katere se izvajajo prevozi po voznem redu
torek	boolean	
sreda	boolean	
cetrtek	boolean	
petek	boolean	
sobota	boolean	
nedelja	boolean	
praznik	boolean	
danOd	integer	Določi se obdobje v letu za katerega velja, da se izvajajo vožnje voznega reda. Upoštevajoč celotni februar je potrebno zapisati danDo = 29.
mesecOd	integer	
danDo	integer	
mesecDo	integer	
akcija	integer, null	0 – vožnja ni operativna  1 – vožnja je operativna

### 15.2.28 DELOVNA NALOGA

**Delovna naloga** je agregacija voženj posameznega prevoznika na posamezne delovne enote.

Podatkovna struktura **DelovnaNaloga** ima naslednje parametre:

Ime spremenljivke	Tip spremenljivke	Opis
idDelovniNalog	integer	
stevilkaDelovnegaNaloga	integer	Številka delovne naloge, ki se uporablja za operativne namene na vozilih
veljavnostOd	string	Datum začetka veljavnosti delovne naloge, primer: 2016-01-01T00:00:00
veljavnostDo	string	Datum konca veljavnosti delovne naloge, primer: 2016-01-01T00:00:00
[]idVoznje	integer	<u>Seznam identifikatorjev voženj, ki pripadajo delovni nalogi.</u> <u>Atribut se bo opustil, zato je potrebno uporabiti polje</u> <u>voznjeGrp</u>
[]voznjeGrp	VoznjaGrp	Objekt vsebuje identifikator vožnje in identifikator predhodnje vožnje v primeru povezovanja voženj.

### 15.2.29 STATIONDETAILS

**StationDetails** je struktura ki se uporablja pri pridobivanju najbližjih postajlišč z metodo `GetNearestStations`.

Ime spremenljivke	Tip spremenljivke	Opis
StationId	integer	
Longitude	double	Zemljepisne dolžine
Latitude	double	Zemljepisne širine
IsMainStation	boolean	Zastavica, ki določa ali je postajlišče <i>glavno postajališče</i>
StationCategory	integer	Tip postajališča
TransportMode	integer	Tip prevoza

### 15.2.30 STATIONDETAILS\_PLADENJ

**StationDetails\_Pladenj** je podatkovna struktura ki vsebuje podobne (skoraj identične) podatke kot strukturo *StationDetails*, sicer se uporablja za potrebe sistema **Pladenj**.

Ime spremenljivke	Tip spremenljivke	Opis
StationId	integer	
Longitude	double	Zemljepisne dolžine
Latitude	double	Zemljepisne širine
IsMainStation	boolean	Zastavica, ki določa ali je postajlišče <i>glavno postajališče</i>
StationCategory	integer	Tip postajališča
TransportMode	integer	Tip prevoza



### 15.2.31 PACKAGESTATUS

Struktura **PackageStatus** omogoča definicijo odgovora pri vpisu lokacije vozila v sistem IJPP.

Ime spremenljivke	Tip spremenljivke	Opis
IPackageld	long	Unikatni identifikator paketa, ki je sestavljen iz identifikatorja terminala (zgornji 4 bajti) in trenutnega časa (spodnji 4 bajti).
iStatus	integer	Status paketa pri obdelavi.  0 = OK  -1 = Paket zavrnjen zaradi sistemske napake (potrebno <b>ponovno</b> pošiljanje)  -2 = Paket že obstaja v sistemu (ni ponovnega pošiljanja)  -3 = Paket zavrnjen zaradi nepravilnih podatkov (potrebno odpraviti pomanjkljivost v času razvoja – ni ponovnega pošiljanja)
sDescription	integer	Opis napake:  iStatus = -3:  [-6] Manjka ali neustrezni identifikator paketa.  [-7] Manjka ali neustrezni tip paketa.  [-8] Manjka ali neustrezni podatki o datumu in uri.  [-9] Manjka ali neustrezni identifikator IJPP vožnje.

### 15.2.32 PACKAGE

Struktura **Package** definira podatkovno strukturo paketa pri vpisu lokacije vozila v sistem IJPP.

Ime spremenljivke	Tip spremenljivke	Opis
IPackageId	long	Unikatni identifikator paketa, ki je sestavljen iz identifikatorja terminala (zgornji 3 bajti) in zaporedne številke transakcije na terminalu (spodnji 4 bajti).
iPackageType	integer	Tip paketa: <ol style="list-style-type: none"> <li>1. vožnja po liniji</li> <li>2. prihod na postajo</li> <li>3. odhod iz postaje</li> <li>4. vožnja mimo postaje</li> </ol>
dGPSLongitude	decimal, null	Geometrijski podatki lokacije; WGS 84  V primeru, če terminal nima GPS signala oz. če lokacija vozila ni mogoča se pošljejo vrednosti »null«
dGPSLatitude	decimal, null	
dGPSAltitude	decimal, null	
dtLocationTime	datetime	Datum in čas zajema podatkov o lokaciji
iTripId	integer	Identifikator IJPP vožnje
iWorkOrderId	integer, null	Številka delovne naloge, ki je definirana v sistemu IJPP
iDirection	integer, null	Smer vožnje izražena v stopnjah.  Zaloga vrednosti [0-359]
fVelocity	float, null	Hitrost vožnje izražena v km/h
IDriverId	long, null	Identifikator voznika
iStopId	integer, null	Identifikator postajne točke; samo za paket tipa: prihod na postajo, odhod iz postaje, vožnja mimo postaje.
iSequence	integer, null	Zaporedna števila postaje za posamezno vožnjo. Začetna vrednost je 1. Za vsako prevoženo postajo (odhod iz postaje ali vožnja mimo postaje) se vrednost poveča za ena. Maksimalna vrednost je 999.

## 16 APLIKACIJA ZA VALIDACIJO IN NAKUP

Namen dokumenta je zagotoviti vse potrebne specifikacije in informacije, ki so potrebne za implementacijo storitev IJPP na mobilni terminal, ki se nahaja na IJPP prevoznem sredstvu (avtobus, vlak). Primarno je dokument namenjen ponudnikom tehnoloških rešitev, ki se želijo vključiti v IJPP program s svojo terminalsko opremo. Upoštevanje specificiranih priporočil je ključnega pomena za končnega uporabnika, saj mora terminalska oprema različnih tehnoloških ponudnikov delovati usklajeno in po enakem principu. Informacije ki se podajajo uporabniku preko uporabniških vmesnikov morajo biti poenotene, unificirane in jasne. Upoštevanje priporočil tega dokumenta daje jasno vizijo orientiranosti IJPP rešitev, ki stremijo k poenotenemu delovanju terminalske strojne in programske opreme vključno s komponentami zalednega sistema različnih tehnoloških ponudnikov.

Dokument sestavlja šest osnovnih poglavij izmed katerih je *drugo poglavje* namenjeno opisu splošnih funkcionalnosti mobilnega terminala, nameščenega na prevozno sredstvo IJPP. Poglavje zajema opis splošne funkcionalnosti, kamor sodi postopek prijave/odjave voznika v/iz mobilnega terminala, izbiro potrebnih parametrov, kot osnovo za nemoteno delovanje opreme, postopek branja informacij iz IJPP kartice, postopek validacije za različne vrste prevozov (mestni potniški promet, medkrajevni potniški promet), postopek nakupa vozovnic (upravičenost), prav tako je podan splošen opis in namen možnosti stornacije zadnjega nakupa produkta oz. skupine produktov, podani pa so tudi postopki in mehanizmi za preprečevanje zlorab, kamor sodi blokada vozovnice in blokada IJPP kartice. Ker se v sistemu IJPP uporablja elektronsko poslovanje se v sklopu tega poglavja podaja še splošen opis funkcionalnosti za preverjanje uspešnosti zapisa kupljenih produktov na IJPP brezstično kartico, s čimer se končni uporabnik prepriča, da je prejel vse kar je plačal. Za potrebe vodenja evidenc so na koncu drugega poglavja podani napotki in smernice za ustvarjanje poročil prodaje produktov, validacij, listnih vozovnic ipd.

*Tretje poglavje* je namenjeno specifikacijam uporabniškega vmesnika za vse aktivnosti, ki se izvajajo s takšnim mobilnim terminalom. Zajete so specifikacije za mobilne terminale z zasloni in integriranimi tipkovnicami, oziroma za terminale, ki premorejo tudi zaslone na dotik. Vzporedno je dodana še definicija uporabe svetlobnega indikatorja, v kolikor ga terminal premore za naslednja stanja: nepripravljenost terminala za interakcije z uporabniki, pripravljenost terminala za interakcije z uporabniki, čakanje na uporabnikovo aktivnost, uspešno izvedena procedura, neuspešno izvedena procedura, napaka ipd. Da je interakcija uporabnika s terminalom povsem nedvoumna, je kot dodaten pogoj uspešne/neuspešne izvedbe posameznega koraka vpeljana še zvočna signalizacija, ki zajema zvočni zapis za uspešno izvedeno proceduro in zvočni zapis za neuspešno izvedeno proceduro. Uporabniški vmesnik je definiran od nižjih nivojev (avtentikacija z zalednim sistemom), do višjih nivojev, ki zadevajo voznika (prenašanje tarife in tarifnih razredov, prenašanje črnih list s seznamami kartic in vozovnic ki se nahajajo na njih, izvajanje servisne storitve in posodabljanje systemskega časa, prijava/odjava voznika) do nivojev, ki zadevajo končnega uporabnika (validacija, nakup, stornacija, nakup listne vozovnice, poizvedba o naloženih produktih (ang. card info) ipd.). Za vsako stanje je točno definirana vsebina zaslona, ki se mora izpisati na zaslonu, v kombinaciji s svetlobnim indikatorjem in zvočnim zapisom.

*Četrto poglavje* vsebuje definicije funkcionalnosti v obliki pravil in diagramov poteka, ki jo mora mobilni terminal podpirati. Sem spada protokol med mobilnim terminalom in zalednim sistemom izveden z

uporabo web-service, sestava tarife in tarifnih razredov, struktura črnih list in izvršnih list ter postopek prenosa iz zalednega sistema na mobilni terminal, definicija vrst posodobitev, ki obstajajo za tak terminal vključno s postopkom prenosa. V zaključku četrtega poglavja so zajeti še tako imenovani 'Batchi' namenjeni prenašanju informacij o izvedenih aktivnostih na terminalu k zalednemu sistemu.

*Peto poglavje* zajema podrobno strukturo datotečnega sistema IJPP brezstične kartice. To pomeni, da je struktura kartice sestavljena iz različnih datotek (xxx), kjer ima vsaka izmed datotek definirano podatkovno strukturo, ki jo sestavljajo posamezni podatki. Za vsak posamezni podatek specifične datoteke na kartici je točno definirana velikost v Byte-ih. S pomočjo poznanih indeksov datotek na kartici, in velikosti posameznega podatka datotečne strukture lahko tehnološki ponudnik zagotovi potrebne algoritme za branje/pisanje iz/na brezstično kartico in skladno z definicijo uporabniškega vmesnika prikaže specifične informacije na zaslon specifično akcijo.

*Šesto poglavje* je namenjeno protokolom komunikacije, ki se uporabljajo na relaciji čitalnik-brezstična IJPP kartica. Osnova komunikacijskemu protokolu je standard ISO/IEC 14443-3 in ISO/IEC 14443-4. V nadaljevanju so definirani vsi ukazi, ki se uporabljajo v procedurah upravljanja s kartico, pisanja po kartici (nakup produktov, stornacija nakupa, blokada aplikacije IJPP, blokada kartice, blokada vozovnice), branja vsebine iz kartice (validacija, card info, preverjanje uspešnosti zapisa vozovnice ...), varnostni ukazi ipd. Vsak izmed omenjenih segmentov je natančno definiran skladno s potrebnimi ukazi in komandami za posamičen segment. Na ta način lahko tehnološki ponudnik opreme zagotovi poenoteno, usklajeno in brezhibno delovanje svoje naprave v sklopu IJPP sistema. Vsak potreben segment implementacije na mobilni enoti je specificiran in podrobno predstavljen v sklopu vseh šestih poglavij.

Ob vsem naštetem velja poudariti, da je lahko na takšnem mobilnem terminalu nameščenih več aplikacij, ena izmed takšnih je IJPP. To omogoča uporabo enega mobilnega terminala za več različnih aplikacij.

## 16.1 FUNKCIONALNOST (SPLOŠNO)

Mobilni terminal - validator s prodajo podpira naslednje funkcije:

- Podpora prijava/odjava uporabnika terminala (npr. prijava/odjava z identifikacijsko kartico)
- Podpora prikaza informacij o kartici IJPP, statusu in o prisotnih produktih na kartici
- Podpora validacije obstoječih produktov IJPP na kartici
- Podpora prodaje vozovnic IJPP (vozovnice na aplikaciji IJPP, listna vozovnica)
  - Terminal podpira prodajo produktov, ki ustrezajo zapisanemu statusu in drugim informacijam na kartici.
  - Na terminalu je potrebno vnesti ustrezne informacije o nakupu vozovnice (vnesemo cono ali izberemo relacijo za katero želimo kupiti vozovnico).
  - Prikaz vseh možnih produktov, ki jih lahko kupimo ter izbira zelenega produkta.
  - V primeru nakupa relacijskih vozovnic, morajo biti na kartici zapisane ustrezne informacije o relaciji (relacija).
  - Prikaz postaj za dano linijo ter izbira relacije za katero lahko kupimo listno vozovnico.

- Terminal omogoča tiskanje potrdila o nakupu vozovnice ter tiskanje listne vozovnice.
- Komunikacija z zalednim sistemom prevoznika
  - Pridobivanje podatkov (tarifa, črna lista, izvršna lista, cone, vozni red, vzdrževalni klic, prejemanje posodobitev,...)
  - Sporočanje podatkov o transakcijah (nakup, validacija, prijava/odjava, stornacija,...)
- Prikazovanje informacij na zaslonu (uspešnost validacije, nakupa produkta, prijave, odjave, ...)
- Tiskanje finančnih poročil v dveh izvodih (poročila o opravljenih gotovinskih transakcijah)

## 16.2 KOMUNIKACIJA Z ZALEDNIM SISTEMOM

Pred prvo uporabo mora terminal iz zalednega sistema sprejeti najnovejšo različico podatkov, ki jih potrebuje za normalno delovanje tarife, črne liste, izvršne liste, posodobitve. Informacije o trenutnih različicah sistemskih podatkov na zalednem sistemu se terminalu sporočijo po vsaki uspešno opravljeni transakciji. Takoj po zagonu terminala, pa je smotrno, da se opravi servisna transakcija. Po servisni oz. po vsaki uspešni transakciji preverimo, če se verzija sistemski podatkov na terminalu razlikuje od teh, ki je na zalednem sistemu. V primeru, da je potrebno sistemske podatke posodobiti, je potrebno opraviti ustrezno transakcijo. Pred pričetkom ustrezne transakcije se na zaslonu pojavi ustrezna statusna ikona. V primeru, da je postopek transakcije kritičen oz., da je potrebno preprečiti nadaljnjo uporabo terminala, ker bi sicer lahko prišlo do morebitne zlorabe sistema, se terminal zaklene. Nadaljnja uporaba terminala pa je omogočena šele po uspešni transakciji. Terminal mora ves čas imeti nameščene najnovejše sistemske podatke:

- Tarife,
- črne liste,
- izvršne liste,
- vozni redi,
- seznam pravil prehodov med conami.

## 16.3 PRIJAVA/ODJAVA

Po vklopu terminala je za nadaljnjo uporabo potrebno opraviti prijavo z ustrezno identifikacijsko kartico. Ko identifikacijsko kartico približamo k terminalu, se izvrši preverjanje ustreznosti prislonjene identifikacijske kartice. Če kartica ustreza pogojem prijave se izvrši prijava. Odjavo pa lahko v terminalu izvedemo z izbiro ustrezne funkcije na terminalu.

Namen funkcije prijava/odjava:

- preprečitev nepooblaščen uporabe terminalske opreme
- evidentiranje uporabnika
- prijava uporabnika in s tem aktivacija terminala

Prijava se izvede s pomočjo posebne identifikacijske kartice. V terminal se shranijo podatki povezani s prijavo uporabnika.

- odjava uporabnika in s tem zaklep terminala  
Odjava uporabnika se izvede z izbiro ustrezne funkcije na terminalu.
- Na zaledni sistem se pošljejo podatki o prijavi in odjavi uporabnika.

## 16.4 IZBIRA PARAMETROV DELOVANJA

Pred izvajanje validacij kartic je v terminal potrebno vnesti ustrezne podatke:

- ID voznega reda
- ID vozila,

Po uspešni prijavi voznika, mora voznik vnesti ID vozila ter izbrati ustrezen vozni red po katerem opravlja svojo vožnjo. To lahko izvede ročno iz nabora vseh voznih redov, ali z vnosom šifre ustreznega voznega reda. Po uspešno izbranem voznem redu, se postaje na terminalu s pomočjo podatkov o trenutni lokaciji samodejno predstavljajo. Voznik mora pred izvajanjem validacije preveriti ali se nahaja na pravi postaji.

## 16.5 BRANJE INFO KARTICE

Na terminalu aktiviramo aplikacijo branje info in približamo kartico za katero želimo prikazati informacije k terminalu. Po uspešnem postopku se informacije o kartici in o vozovnicah na kartici shranijo v ustrezno podatkovno strukturo, nato pa se prikažejo na zaslonu.

## 16.6 VALIDACIJA

Ko je na terminalu izbrana in se izvaja ustrezna aplikacija za validacijo oz. aktivacijo vozovnic, lahko kadarkoli k terminalu pristonimo kartico, ki jo želimo validirati oz. aktivirati, da se izvede postopek validacije oz. aktivacije. Po uspešno izvedenem postopku se informacije o validaciji izpišejo na zaslonu.

### 16.6.1 VALIDACIJA CONSKI PROMET

Pri validaciji mestnega prometa, gre za validacijo conskega prometa. Na podlagi voznega reda je možno določiti, katere linije in/ali postaje sodijo v določeno cono, ali so uvrščene v medkrajevni promet. Po pravilnem postopku prijave in izbire parametrov delovanja, validator sam preklaplja med postajami.

Pred izvedbo validacije, pa mora voznik vseeno preveriti na kateri postaji se nahaja. Znotraj conskega prometa se preverja ali ima potnik ustrezno vozovnico na kartici.

## **16.6.2 VALIDACIJA RELACIJSKI PROMET**

Za validacijo medkrajevnih oz. relacijskih vozovnic potrebujemo informacijo o trenutni postaji. Znotraj validacije se preverja ali ima potnik ustrezno vozovnico na kartici in če se nahaja na pravilni relaciji, glede na podatke, ki so zapisani na kartici in s katerimi razpolaga validator ob validaciji.

## **16.7 NAKUP VOZOVNIC**

Ko je na terminalu izbrana in se izvaja ustrezna aplikacija za nakup vozovnic, lahko k terminalu prislonimo kartico, za katero želimo opraviti nakup vozovnice. Najprej se izvede branje info oz. ustreznih informacij o kartici na podlagi katerih se lahko na zaslonu prikažejo vsi ustrezni produkti, ki jih lahko za to kartico kupimo. Po izbiri ustreznega produkta, kartico ponovno prislonimo k terminalu. Po uspešno izvedenem nakupu se na zaslonu izpišejo informacije o nakupu vozovnice.

Glede na postajo, kjer se nahaja vozilo se smatra ali gre za consko območje ali za medkrajevni promet.

### **16.7.1 NAKUP CONSKI PROMET**

Nakup conskih vozovnic je odvisen od statusa posamezne kartice ter določitve različnih tipov vozovnic, ki jih je za posamezni status moč kupiti.

#### **16.7.1.1 BREZSTIČNA KARTICA**

Na brezstično kartico terminal zapiše ustrezno vozovnico, ki jo želimo kupiti.

#### **16.7.1.2 LISTNA**

V primeru, da je možna izbira med različnimi conami, voznik izbere za koliko con želi potnik vozovnico. Na listno vozovnico se nato natisnejo podatki za katero cono/cone je bila kupljena vozovnica. Tako kupljena vozovnica NE omogoča prestopanja.

## 16.7.2 NAKUP RELACIJSKI PROMET

Kadar gre za nakup medkrajevne vozovnice imamo različne opcije:

- Podaljševanje terminske vozovnice, ki je bila že zapisana na kartici – relacija je zapisana na kartici in se izvede nakup terminske vozovnice za dano relacijo. Na kartici je lahko tudi več relacij, posledično je pred izvedbo nakupa potrebno izbrati ustrezno.
- Nakup se lahko izvede za neko relacijo, ki jo izbere voznik. Kot vstopna postaja se smatra trenutna postaja, izstopna postaja, pa se mora nahajati na isti liniji kot je vstopna postaja. Takšna vozovnica NE omogoča prestopanja. Vozovnica se lahko zapiše/izda:
  - Na brezstično kartico
  - Kot listna vozovnica

### 16.7.2.1 BREZSTIČNA KARTICA

Na brezstično kartico se lahko podaljša terminska vozovnica za že obstoječo relacijo na kartici. Ob prisotnosti več relaciji mora voznik izbrati ustrezno relacijo.

Zapiše pa se lahko tudi enkratna vožnja z relacijo, ki jo izbere voznik in definira vstopno in izstopno postajo. Takšna vozovnica NE omogoča prestopanja.

### 16.7.2.2 LISTNA

V obliki listne vozovnice se izda enkratna vozovnica za določeno relacijo, ki se nahaja na liniji vozila in vsebuje vstopno in izstopno postajo in ne omogoča prestopanja.

## 16.8 STORNO

Stornacija je možna le za predhodno transakcijo. Torej, če se je izvedel nakup terminske vozovnice, je možno stornacijo izvesti takoj, če se vmes izvede kakšna druga operacija na istem terminalu, npr. validacija, nakup drugega produkta ipd., stornacija predhodne terminske vozovnice ne bo več izvedljiva. Izvedbo stornacije je potrebno predhodno izbrati funkcijo storno. Vsaka stornacija se zavede v zaledni sistem in se nahaja tudi znotraj poročil izmene.

Ko je na terminalu izbrana in se izvaja ustrezna aplikacija za storno nakup vozovnice, lahko k terminalu prislonimo kartico, za katero želimo opraviti storno nakup vozovnice. Najprej se izvede branje info oz. ustreznih informacij o kartici na podlagi katerih se lahko na zaslonu prikažejo vsi ustrezni produkti, ki jih za to kartico lahko storniramo. Storniramo lahko vozovnice, ki so bile predhodno kupljene na istem terminalu. Po izbiri ustreznega produkta, kartico ponovno prislonimo k terminalu. Po uspešno izvedenem nakupu se na zaslonu izpišejo informacije o storno postopku. Imetniku kartice pa se



povrnejó sredstva v obliki gotovine. Predpogoj za storno nakupa vozovnice je uspešen predhoden nakup vozovnice na istem terminalu.

## 16.9 BLOKADA APLIKACIJE

Blokada aplikacije se izvede samodejno pri postopku branja info kartice v primeru, da se UID kartice nahaja na črni listi.

## 16.10 BLOKADA VOZOVNICE

Blokada aplikacije se izvede samodejno pri postopku branja info kartice v primeru, da se UID kartice nahaja na izvršni listi. Z ustreznim parametrom ob UID se terminalu sporoči katero vozovnico je potrebno blokirati.

## 16.11 PREVERJANJE USPEŠNOSTI ZAPISA VOZOVNICE

Poseben mehanizem preverjanja uspešnosti zapisa vozovnice se aktivira v primeru, da je bila kartica prehitro odstranjena od terminala. V redkih primerih se zgodi, da se podatki ustrezno zapišejo na kartico medtem ko, se operacija aktivacije ali nakupa vozovnice prekine zaradi prehitre odstranitve kartice.

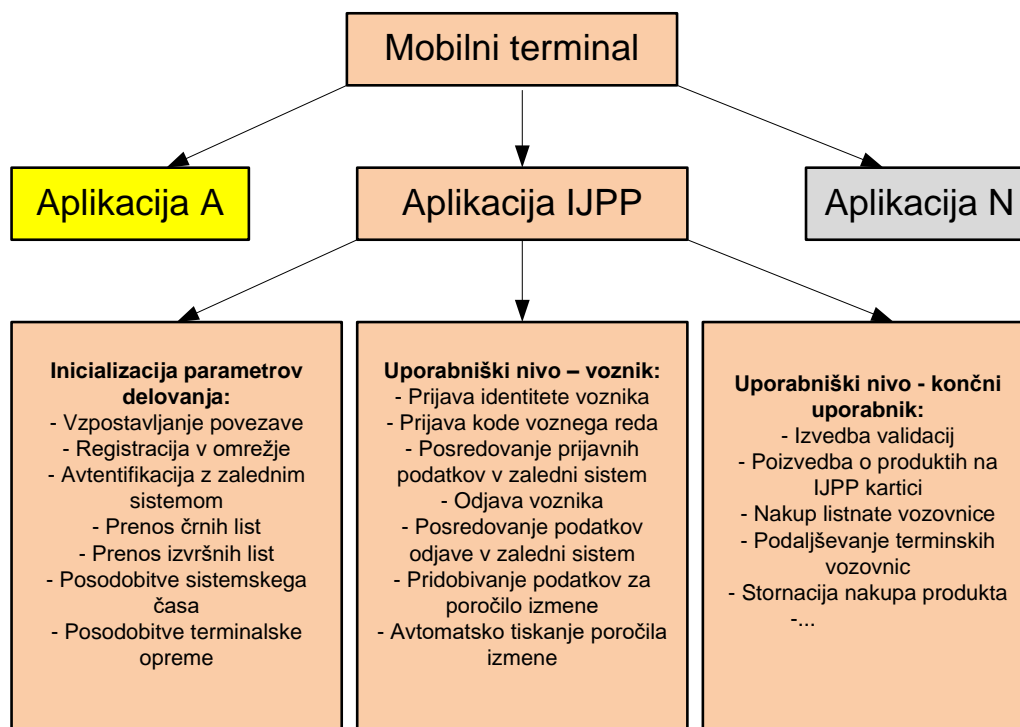
## 16.12 FINANČNA POROČILA IZMENE

Finančna poročila so namenjena pregledu prodaje vozovnic v času izmene. Ko se izvede odjava voznika se poročilo izmene natisne avtomatsko.

## 16.13 GUI

V tem sklopu dokumenta so podane specifikacije oz. priporočila za implementacijo uporabniškega vmesnika mobilnega terminala, ki je namenjen validacijam in nakupom produktov na prevoznem sredstvu (avtobus, vlak). Dokument je primarno namenjen ponudnikom tehnoloških rešitev (ponudniki strojne in programske opreme), ki se želijo vključiti v IJPP sistem v navezi s prevoznikom, ki do sedaj ni uporabljal nobene izmed poznanih obstoječih rešitev, in kateri ima namen uporabljati rešitve kakšnega drugega ponudnika. V dokument so zajeta vsa stanja, ki jih takšnem terminal podpira, in sicer:

prijava/odjava, pripravljenost terminala za izvajanje validacij, pripravljenost terminala za nakup produktov itd.

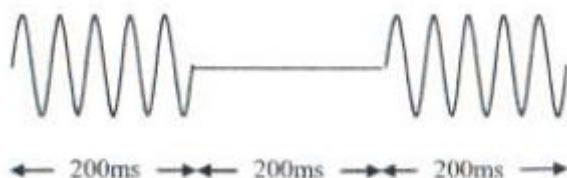


Priporočila se navezujejo na terminale, ki premorejo samo zaslon oz. terminale, ki zraven zaslona premorejo še svetlobni/e indikator/je.

### 16.13.1 DEFINICIJE

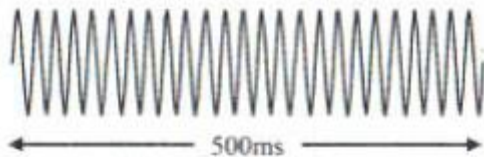
#### **Definicija zvoka neuspešne interakcije**

Za neuspešno izvedeno proceduro na validirni enoti se uporabi frekvenca sinusnega vala 750Hz, ki predstavlja dvojni pisk s periodo 200ms predvajanja zvoka nato sledi perioda 200ms tišine, in nato ponovno perioda 200ms predvajanja zvoka za neuspešno proceduro (glej spodnjo sliko).



#### **Definicija zvoka uspešne interakcije**

Zvočni signal za uspešno izvedeno proceduro na validirni napravi mora biti sinusni val, frekvenca 1500Hz in periode 500ms.



### **Definicije pomena svetlobnega indikatorja**

- Izključena led pri izključenem terminalu
- ◼ Led utripa v rdeči barvi – terminal ni pripravljen za končnega uporabnika
- ◼ Led gori v rdeči barvi – terminal ni pripravljen za končnega uporabnika, napaka
- ◼ Led utripa v zeleni barvi – terminal je pripravljen za interakcijo z uporabnikom
- ◼ Led gori v zeleni barvi – terminal signalizira uspešno interakcijo

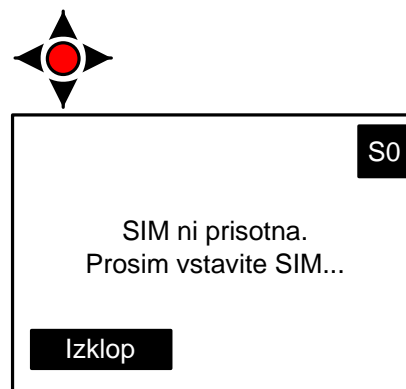
Pri povezovanju terminala z zalednim sistemom se lahko koristijo različni mediji, ki so danemu ponudniku tehnološke opreme na razpolago. Sem lahko spada komunikacija preko GPRS omrežja, komunikacija preko brezžičnih omrežij WiMax, WiFi, satelitskih omrežij ipd. V dokumentu je podan primer uporabe GPRS prenosnega medija, kot ene izmed obstoječih in predhodno omenjenih rešitev. Za ostale uporabljene medije se uporabi podoben koncept, kot ga podaja primer GPRS v nadaljevanju.

#### **16.13.2 GUI - MOBILNI TERMINAL NE RAZPOLAGA S SIM KARTICO**

Mobilni terminal mora vsebovati mehanizme za ugotavljanje prisotnosti SIM kartice, ki jo potrebuje za povezavo v GPRS omrežje in za paketni prenos podatkov po GPRS omrežju. Implementacija mehanizma za ugotavljanje prisotnosti SIM kartice običajno temelji na ukazih, ki se upravljajo na strani GPRS modema. V kolikor ukazi modema ne prejmejo odziva SIM kartice, se mora na zaslonu mobilnega terminala prikazati naslednje:

- Sporočilo: "SIM ni prisotna. Prosim vstavite SIM." Sporočilo mora ostati prisotno ves čas, dokler mobilni terminal ne prepozna pravilno SIM kartice.
- V primeru, da mobilni terminal premore še svetlobni indikator naj slednji v tem primeru utripa z rdečo barvo v razmiku 500ms in se sekvenca utripanja ponavlja na periodo 5 sekund. Svetlobni signal hkrati uporabnika opozarja, da naprava ni pripravljena za izvajanje interakcij.
- Za lažjo razpoznavo akcij mobilnega terminala (kaj dejansko počne) se priporoča še uporaba majhnih ikon, ki jo sestavlja črka in številka. V tem primeru "S0", kar pomeni da terminal ne prepozna SIM kartice, ali da slednja ni vstavljena, oz. ni pravilno vstavljena.

- Prikaz majhnih statusnih ikon na zaslonu naj se izvaja v zgornji vrstici desne polovice zaslona.
- Ostale interakcije z mobilnim terminalom se v tem stanju preprečijo.
- Ponovno prepoznavanje SIM kartice se izvede ob ponovnem vklopu naprave.
- V primeru, da se zahteva PIN (vključena opcija), se v naslednjem stanju (16.13.3) vnese PIN oz. PUK, v nasprotnem primeru se izvede registracija v GPRS omrežje (16.13.5).
- Pravilna razpoznavna SIM kartice je predpogoj za nadaljnje operacije z mobilnim terminalom, kot je registracija v GPRS omrežje, prenos tarife, črnih list, posodobitve sistemskega časa.
- Po uspešnem prepoznavi SIM kartice, se predhodno sporočil odstrani, in ugasne utripajoč svetlobni indikator, terminal pa preide v naslednje stanje registracije v GPRS omrežje (16.13.5).

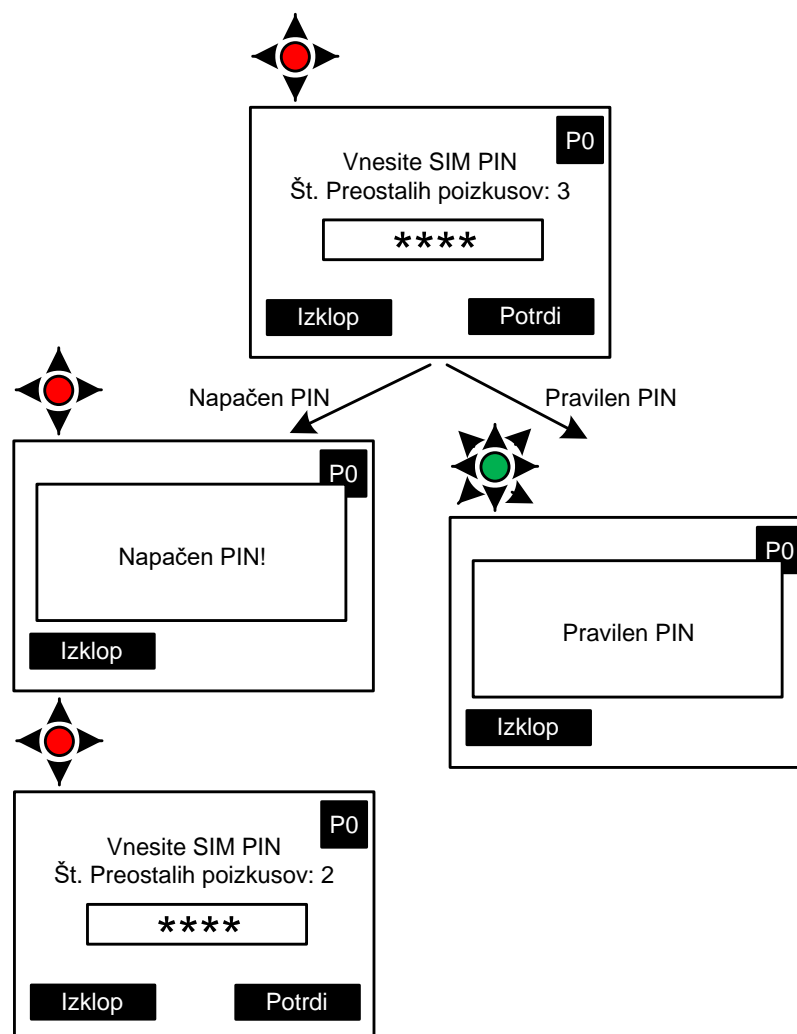


### 16.13.3 GUI - VNOS SIM PIN

Če je SIM kartica v predhodnem koraku pravilno razpoznana, in je hkrati omogočen vnos SIM PIN številke, mora operater mobilnega terminala s pomočjo bodisi zaslona na dotik, bodisi integrirane tipkovnice vnesti 4-mestni PIN SIM kartice. Vsebino zaslona pri tej proceduri naj sestavlja:

- Sporočilo: "Vnesite SIM PIN. Število preostalih poizkusov:3". Sporočilo mora ostati prisotno ves čas, dokler vnesen PIN ni potrjen. Če je potrjen vnos PIN-a pravilen terminal preide v stanje (16.13.5). Če je potrjen vnos PIN številke napačen se naj najprej na zaslon za obdobje 3 sekund pokaže sporočilo: "Napačen PIN!". Po preteku 3 sekund se predhodno sporočilo odstrani in prikaže naslednje: "Vnesite SIM PIN. Število preostalih poizkusov: 2" itn. Če je vnos PIN številke trikrat zaporedoma napačen, terminal preide v stanje (16.13.4).
- Polje za vnos PIN številke
- Vsak numerični vnos se naj maskira s simbolom "\*", kar pomeni da številke, ki sestavljajo PIN niso vidne.
- V primeru, da mobilni terminal premore še svetlobni indikator naj slednji v tem primeru pravilnega vnosa PIN številke zasveti zeleno v dolžini svetlobnega pulza 0.5 sekunde. V primeru nepravilnega vnosa naj svetlobni indikator zasveti rdeče za enako obdobje (3 sekunde), kot je prikazano sporočilo o neuspešnem vnosu.
- Za lažjo razpoznavo akcij mobilnega terminala (kaj dejansko počne) se priporoča še uporaba majhnih ikon, ki jo sestavlja črka in številka. V tem primeru "P0", kar pomeni da mobilni terminal zahteva vnos PIN številke.

- Prikaz majhnih statusnih ikon na zaslonu naj se izvaja v zgornji vrstici desne polovice zaslona.
- Ostale interakcije z mobilnim terminalom razen vnosa SIM PIN številke se v tem stanju preprečijo.
- Pravilen vnos SIM PIN številke je predpogoj za kasnejšo registracijo terminala v GPRS omrežje (stanje 16.13.5).



#### 16.13.4 GUI - VNOS SIM PUK

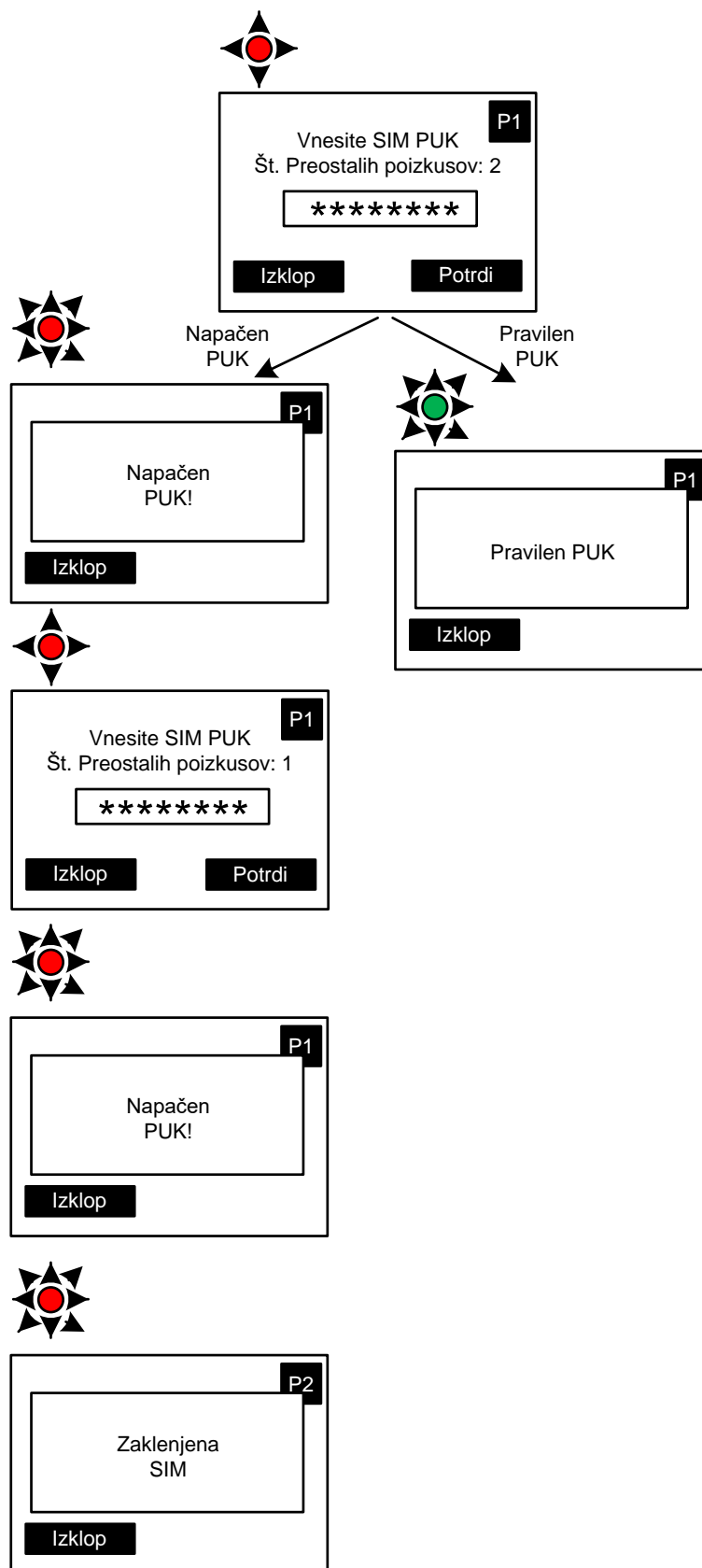
Če je operater v predhodnem koraku trikrat napačno vnesel in potrdil SIM PIN številko (4-mestna), mora mobilni terminal preiti v stanje za vnos SIM PUK številke (8-mestna). Vnos SIM PUK številke izvrši bodisi s pomočjo integrirane tipkovnice, oziroma preko zaslona na dotik v kolikor ga takšen mobilni terminal premore. Vsebino zaslona pri vnosu SIM PUK naj sestavlja:

- Sporočilo: "Vnesite SIM PUK. Število preostalih poizkusov:2". Sporočilo mora ostati prisotno ves čas, dokler vnesen PUK ni potrjen. Če je potrjen vnos PUK-a pravilen terminal preide v stanje (16.13.5). Če je potrjen vnos PUK številke napačen se naj najprej na zaslon za obdobje 3 sekund pokaže sporočilo: "Napačen PUK!". Po preteku 3 sekund se predhodno sporočilo odstrani in prikaže naslednje: "Vnesite SIM PUK. Število preostalih poizkusov: 1" itn. Če je vnos

PUK številke 2 zaporedoma napačen, terminal pobriše vso vsebino iz zaslona, in v sam center terminalskega zaslona umesti sporočilo "Zaklenjena SIM".

- Polje za vnos 8-mestne PUK številko.
- Vsak numerični vnos se naj maskira s simbolom "\*", kar pomeni da številke, ki sestavljajo PUK niso vidne.
- V primeru, da mobilni terminal premore še svetlobni indikator naj slednji v primeru pravilnega vnosa PUK številke zasveti zeleno v dolžini svetlobnega pulza 1 sekunde. V primeru nepravilnega vnosa naj svetlobni indikator zasveti rdeče za enako obdobje (3 sekunde), kot je prikazano sporočilo o neuspešnem vnosu PUK številke.
- Za lažjo razpoznavo akcij mobilnega terminala (kaj dejansko počne) se priporoča še uporaba majhnih ikon, ki jo sestavlja črka in številka. V tem primeru "P1", kar pomeni da mobilni terminal zahteva vnos PUK številke.
- V primeru dveh zaporednih napačnih vnosov PUK številke (zaklenjena SIM), naj se prikaže ikona "P2", ki zraven predhodno omenjenega sporočila signalizira zaklenjeno SIM kartico. V tem primeru se naj trajno prižge svetlobni indikator v rdeči barvi, ki ostane aktiven skupaj s sporočilom vse do ugasnitve (ang. Shut down) mobilnega terminala. Če operater v tem času ne zamenja SIM kartice, in mobilni terminal ponovno prižge se ponovno prikaže vsebina zaslona s sporočilom o zaklenjeni SIM kartici, svetlobni indikator pa se ponovno trajno prižge v rdeči barvi. Če operater zamenja kartico z drugo, in je na njej omogočen SIM PIN se terminal vrne v stanje (16.13.3) oziroma, če kartica ni pravilno vstavljena v stanje (16.13.2).
- Prikaz majhnih statusnih ikon na zaslonu naj se izvaja v zgornji vrstici desne polovice zaslona.
- Ostale interakcije z mobilnim terminalom razen vnosa SIM PUK številke in možnosti izklopa mobilnega terminala se v tem stanju preprečijo.

Pravilen vnos SIM PIN številke je predpogoj za kasnejšo registracijo terminala v GPRS omrežje (stanje 16.13.5). Registracija v GPRS omrežje



### 16.13.5 GUI - REGISTRACIJA MOBILNEGA TERMINALA V GPRS OMREŽJE

Če je operater v predhodnem koraku pravilno vnesel SIM PIN/SIM PUK številko se prične procedura registracije naprave v GPRS omrežje (iskanje razpoložljivega omrežja, registracija v omrežje in prijava). V tem primeru se naj na zaslon mobilnega terminala prikaže naslednja vsebina:

- Sporočilo: "Povezovanje z GPRS omrežjem. Prosim počakajte..."
- Indikator napredka (ang. progress bar), ki je lahko v obliki dejanskega napredka, ali v obliki animacije, s čimer signalizira, da je naprava aktivna v proceduri registracije v GPRS omrežje.
- V primeru, da mobilni terminal premore še svetlobni indikator naj slednji v tem primeru sveti brez premora z rdečo barvo, kar signalizira nepripravljenost za izvajanje interakcij z uporabnikom
- Za lažjo razpoznavo akcij mobilnega terminala (kaj dejansko počne) se priporoča še uporaba majhnih ikon, ki jo sestavlja črka in številka. V tem primeru "G0", kar ravno predstavlja postopek registracije naprave v GPRS omrežje.
- Ostale interakcije z mobilnim terminalom se v tem stanju preprečijo.
- Če registracija v GPRS omrežje ne uspe (npr. področje brez razpoložljivega omrežja), se mora procedura ponavljati do uspešne registracije in prijave v samo omrežje.
- Po uspešni registraciji in prijavi v omrežje, se pobriše vsa aktivna vsebina zaslona, vključno s statusno ikono, in se prikaže vsebina naslednjega stanja (glej stanje 16.13.6).



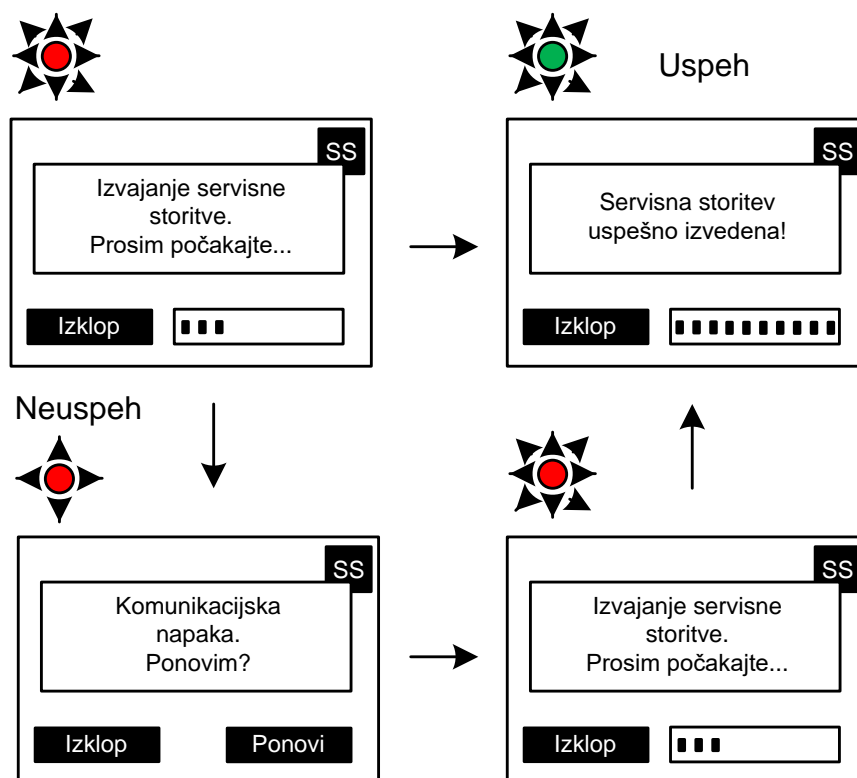
#### 16.13.6 GUI - AVTENTIKACIJA MOBILNEGA TERMINALA Z ZALEDNIM SISTEMOM - SERVISNA STORITEV

Ko je terminal uspešno prijavljen v omrežje mobilnega operaterja, mehanizmi za preverjanja stanja terminala (čas delovanja v offline režimu, čas zadnje prijave v zaledni sistem ipd.) preverjajo pogoje za proženje servisne storitve. V kolikor mehanizmi ugotovijo potrebo po izvedbi servisnega klica, se le ta izvede. V tem primeru se naj na zaslon mobilnega terminala prikaže naslednja vsebina:

- Sporočilo: "Izvajanje servisne storitve. Prosim počakajte..."



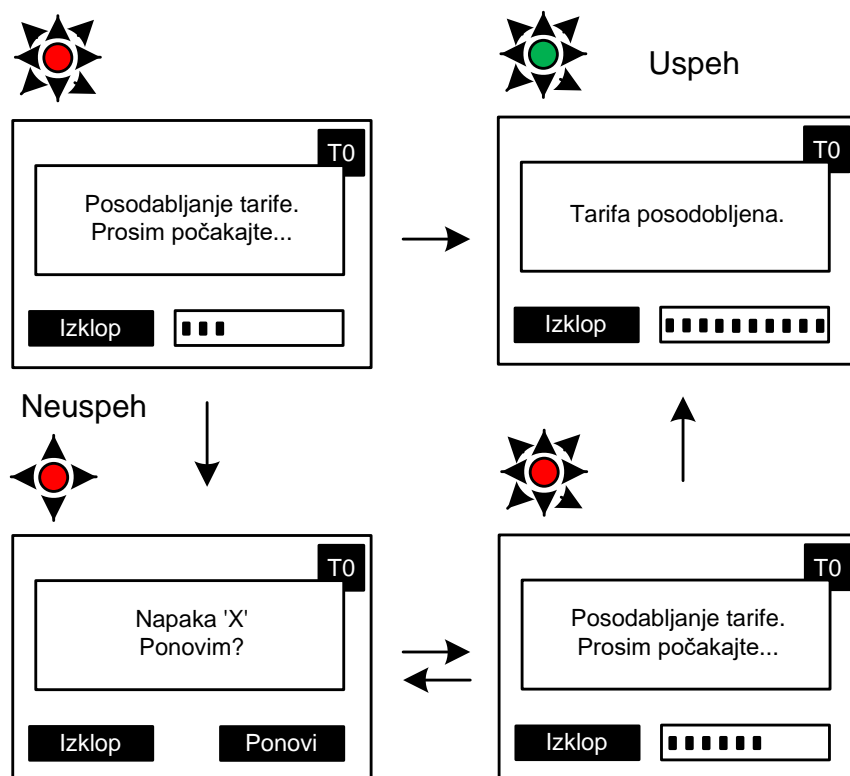
- Indikator napredka (ang. progress bar), ki je lahko v obliki dejanskega napredka, ali v obliki animacije, s čimer signalizira, da je naprava aktivna v proceduri izvajanja transakcije servisne storitve.
- V primeru, da mobilni terminal premore še svetlobni indikator naj bo slednji v tem primeru vključen (sveti z rdečo barvo), kar signalizira nepripravljenost za izvajanje interakcij z uporabnikom
- Za lažjo razpoznavo akcij mobilnega terminala (kaj dejansko počne) se priporoča še uporaba majhnih ikon, ki jo sestavljata dve črki. V tem primeru "SS", kar ravno predstavlja postopek izvajanja servisne storitve.
- Ostale interakcije z mobilnim terminalom se v tem stanju preprečijo.
- Če izvedba servisne storitve ne uspe (npr. nepravilno vpisan terminal v zaledni sistem, komunikacijska napaka ipd.), se procedura ponavlja do uspešnega zaključka servisne storitve.
- V primeru, da servisna storitev ne uspe, mora terminal na zaslon za čas 3 sekund izpisati vrsto napake, pri čemer mora biti predhodna vsebina v celoti pobrisana, vključno z indikatorjem poteka;
  - "Neuspešna avtentikacija"
  - "Komunikacijska napaka"
  - "Napaka v kriptografiji (ključi)"
  - ...
- Po prikazu sporočila v primeru neuspešne servisne storitve (po preteku 3 sekund), se vsebina sporočila o napaki pobriše, proceduro pa vrne na sam začetek ponovnega izvajanja servisne storitve, pri čemer se ponovno prikaže sporočilo "Izvajanje servisne storitve. Prosim počakajte...", indikator poteka itn.
- V proceduri servisne storitve mobilni terminal prav tako preverja informacije tarife (poglavje 16.13.7 črnih list (poglavje 16.13.8, 16.13.9), statusa časa (poglavje 16.13.10) itn. V kolikor mobilni terminal ugotovi, da ga čaka posamezno navedena posodobitev, jo nemudoma prične izvajati v skladu, kot je definirano v sklopih (16.13.7, 16.13.8, 16.13.9 in 16.13.10).
- Če ne obstaja nobena posodobitev se terminal po izvedbi servisne storitve vrne v prvo stanje, ki je namenjeno uporabniku (sklop 16.13.11) - vpisu/odjavi voznika prevoznega sredstva IJPP.



### 16.13.7 GUI - MOBILNI TERMINAL NE RAZPOLAGA Z AKTUALNO TARIFO

Priporočilo se nanaša na prvi zagon mobilnega terminala. Ko je tarifa enkrat že nameščena na mobilnem terminalu uporabniku ni potrebno čakati na posodobljanje tarife, saj se posodobitev izvaja v ozadju, kar indicira s statusno ikono 'T0'. Kadar mobilni terminal na prevoznem sredstvu sploh ne razpolaga z nobeno različico tarife, oz. slednje še sploh ni prenesel, se mora postopek validacij v tej proceduri preprečiti. Na začetni zaslon se mora v tem primeru prikazati:

- Sporočilo: "Posodobljanje tarife, prosim počakajte..." (samo v primeru prvega zagona terminala – ne razpolaga z nobeno tarifo).
- Indikator napredka (ang. progress bar), ki je lahko v obliki dejanskega napredka, ali v obliki animacije, s čimer signalizira, da je naprava aktivna v proceduri posodobitve tarife.
- V primeru, da mobilni terminal premore še svetlobni indikator naj slednji v tem primeru sveti brez premora z rdečo barvo, kar signalizira nepripravljenost za izvajanje interakcij
- Za lažjo razpoznavo akcij mobilnega terminala (kaj dejansko počne) se priporoča še uporaba majhnih ikon, ki jo sestavlja črka in številka. V tem primeru "T0", kar pomeni prenašanje tarife
- Prikaz majhnih statusnih ikon na zaslonu naj se izvaja v zgornji vrstici desne polovice zaslona.
- Ostale interakcije z mobilnim terminalom se v tem stanju preprečijo.
- Če prenos tarife ne uspe, se mora procedura ponavljati do uspešnega prenosa.
- Po uspešnem prenosu, terminal preide v naslednje stanje za prenos seznama brezstičnih kartic na črni listi (naslednje GUI stanje).

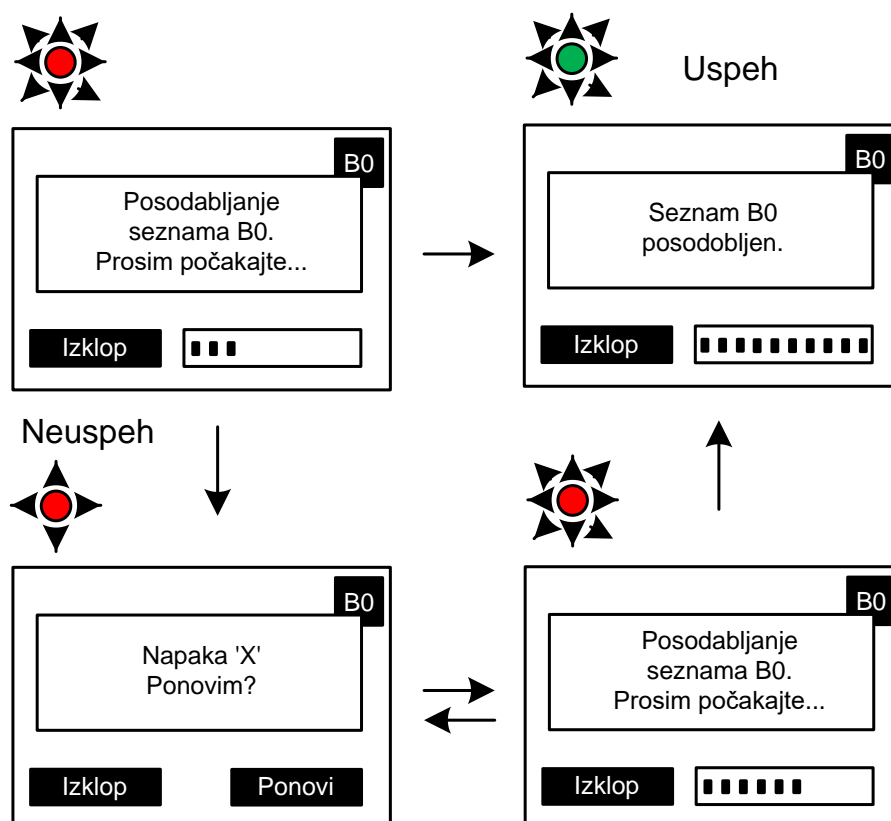


### 16.13.8 GUI - MOBILNI TERMINAL NE RAZPOLAGA Z AKTIVNIM SEZNAMOM KARTIC NA ČRNI LISTI

Podobno kot za predhodno stanje, je procedura definirana za prvi zagon terminala, kateri še ne razpolaga s seznamom IJPP kartic na črni listi. Ko terminal enkrat razpolaga s seznamom se vsak naslednji prenos dotičnega seznama vrši v ozadju, na kar signalizira statusna ikona 'B0'. Pri prenosu seznama v ozadju lahko uporabnik nemoteno izvaja ineterakcije z mobilnim terminalom. Kadar mobilni terminal na prevoznem sredstvu ne razpolaga z nobeno različico seznama kartic na črni listi, oz. slednjega še sploh ni prenesel, se morajo vse interakcije mobilnega terminala na relaciji z uporabnikom preprečiti. Terminal mora biti v takšnem primeru v t.i. 'zaklenjenem' načinu, kar pomeni, da izvaja nujne posodobitve, akcije s strani uporabnika pa so preprečene. Sporočilo o nepripravljenosti za uporabnika mora biti jasno in nedvoumno. Če uporabnik kljub temu prisloni brezstično IJPP kartico se ne sme zgoditi nič (čitalec v nepripravljenosti). Na začetni zaslon se mora v tem primeru prikazati:

- Sporočilo: "Posodabljanje seznama B0, prosim počakajte..."
- Indikator napredka (ang. progress bar), ki je lahko v obliki dejanskega napredka, ali v obliki animacije, s čimer signalizira, da je naprava aktivna v proceduri posodobitve seznama kartic na črni listi.
- V primeru, da mobilni terminal premore še svetlobni indikator naj slednji v tem primeru sveti brez premora z rdečo barvo, kar signalizira nepripravljenost za izvajanje interakcij z uporabnikom.

- Za lažjo razpoznavo akcij mobilnega terminala (kaj dejansko počne) se priporoča še uporaba majhnih ikon, ki jo sestavlja črka in številka. V tem primeru "B0", kar pomeni prenašanje seznama kartic, ki so na črni listi
- Prikaz majhnih statusnih ikon na zaslonu naj se izvaja v zgornji vrstici desne polovice zaslona.
- Ostale interakcije z mobilnim terminalom se v tem stanju preprečijo.
- Če prenos seznama kartic na črni listi ne uspe, se mora procedura ponavljati do uspešnega prenosa.
- Po uspešnem prenosu, terminal preide v naslednje stanje za prenos seznama vozovnic na črni listi (naslednje GUI stanje).

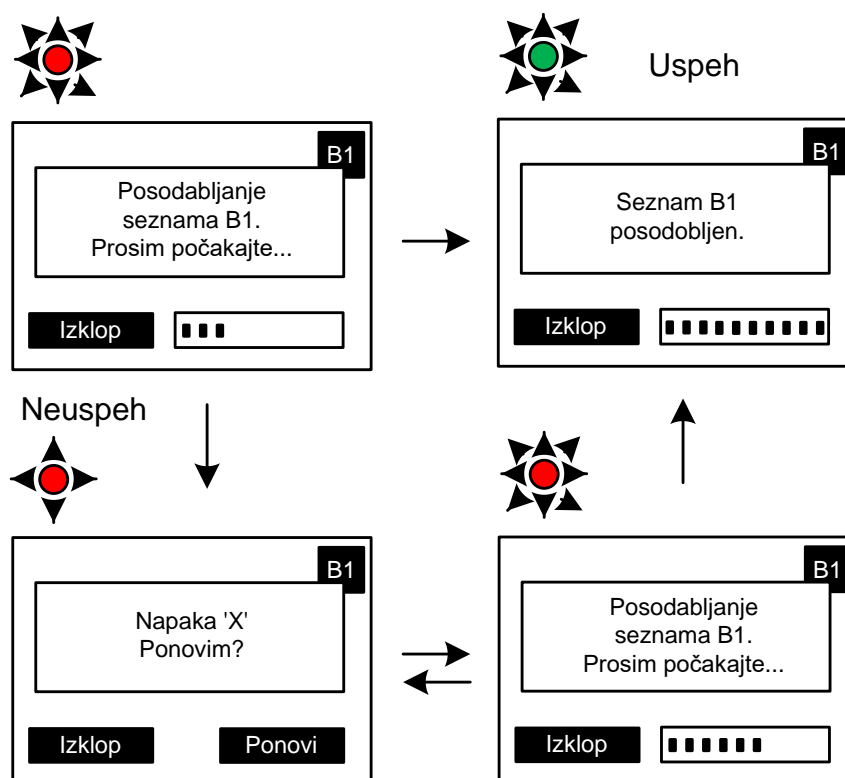


#### 16.13.9 GUI - MOBILNI TERMINAL NE RAZPOLAGA Z AKTIVNIM SEZNAMOM VOZOVNIC NA ČRNI LISTI

Procedura je definirana za prvi zagon terminala, ki ne razpolaga z črno listo vozovnic. Za ta primer velja spodaj definirana procedura. Ko ima terminal enkrat že nameščeno črno listo vozovnic, se nadgradnja izvaja v ozadju, kar signalizira statusna ikona. Ko se nadgradnja izvaja v ozadju se lahko interakcije med uporabnikom in terminalom izvajajo nemoteno. Kadar mobilni terminal na prevoznem sredstvu ne razpolaga z zadnjo različico seznama vozovnic na črni listi, oz. slednjega še sploh ni prenesel, se morajo vse interakcije mobilnega terminala na relaciji z uporabnikom preprečiti. Terminal mora biti v takšnem primeru v t.i. 'zaklenjenem' načinu, kar pomeni, da izvaja nujne posodobitve, akcije s strani uporabnika pa so preprečene. Sporočilo o nepripravljenosti za uporabnika mora biti jasno in nedvoumno. Če

uporabnik kljub temu prisloni brezstično IJPP kartico se ne sme zgoditi nič (čitalec v nepripravljenosti). Na začetni zaslon se mora v tem primeru prikazati:

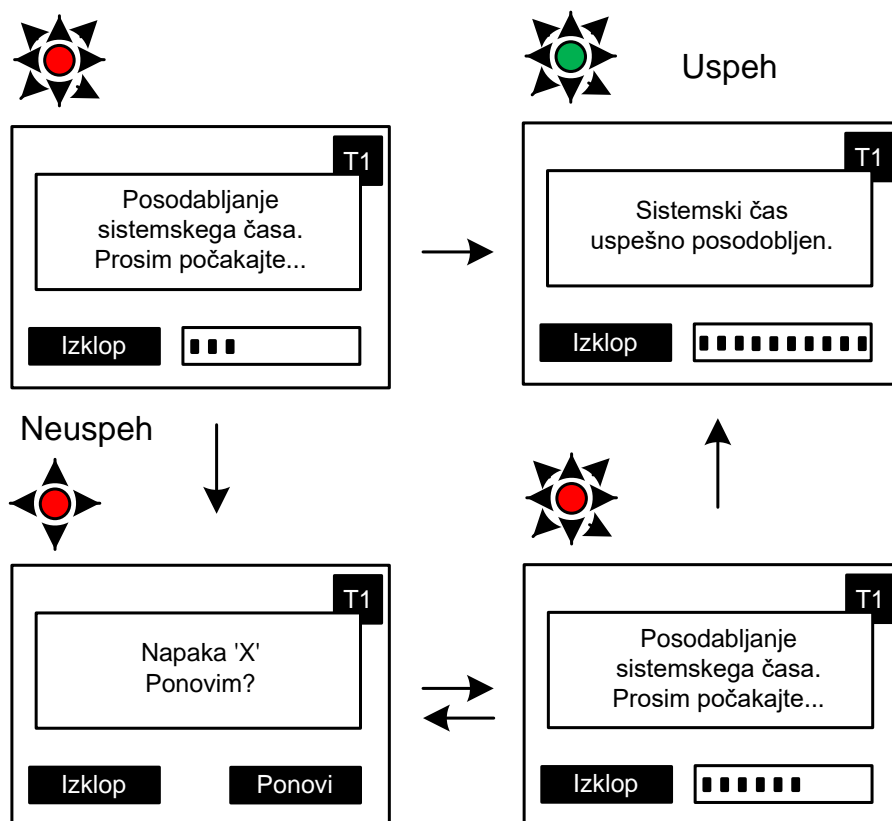
- Sporočilo: "Posodabljanje seznama B1, prosim počakajte..."
- Indikator napredka (ang. progress bar), ki je lahko v obliki dejanskega napredka, ali v obliki animacije, s čimer signalizira, da je naprava aktivna v proceduri posodobitve seznama vozovnic na črni listi.
- V primeru, da mobilni terminal premore še svetlobni indikator naj slednji v tem primeru sveti brez premora z rdečo barvo, kar signalizira nepripravljenost za izvajanje interakcij z uporabnikom.
- Za lažjo razpoznavo akcij mobilnega terminala (kaj dejansko počne) se priporoča še uporaba majhnih ikon, ki jo sestavlja črka in številka. V tem primeru "B1", kar pomeni prenašanje seznama vozovnic, ki so na črni listi
- Prikaz majhnih statusnih ikon na zaslonu naj se izvaja v zgornji vrstici desne polovice zaslona.
- Ostale interakcije z mobilnim terminalom se v tem stanju preprečijo.
- Če prenos seznama kartic na črni listi ne uspe, se mora procedura ponavljati do uspešnega prenosa.
- Po uspešnem prenosu, terminal preide v naslednje stanje za prenos seznama vozovnic na črni listi (naslednje GUI stanje).



#### 16.13.10 GUI - MOBILNI TERMINAL NE RAZPOLAGA Z DEJANSKIM ČASOM

Kadar mobilni terminal na prevoznem sredstvu ne razpolaga z dejanskim časom se morajo interakcije mobilnega terminala z uporabnikom preprečiti. Terminal mora v tem primeru najprej uskladiti čas z zalednim sistemom v t.i. 'zaklenjenem' načinu. Uskladitev časa je pomembna iz dveh vidikov. Prvi vidik se nanaša na pravilni izpis časa o opravljeni validaciji, nakupu, poizvedbi ipd., drugi vidik pa se nanaša na usklajevanje poročil na zalednem sistemu, ki se generirajo na osnovi poslanih 'batchev', katere posreduje mobilni terminal v zaledni sistem preko GPRS podatkovnega kanala. Sporočilo o nepripravljenosti za uporabnika mora biti jasno in nedvoumno. Če uporabnik kljub temu prisloni brezstično IJPP kartico se ne sme zgoditi nič (čitalec v nepripravljenosti). Na začetni zaslon se mora v tem primeru prikazati:

- Sporočilo: "Posodabljanje časa, prosim počakajte..."
- Indikator napredka (ang. progress bar), ki je lahko v obliki dejanskega napredka, ali v obliki animacije, s čimer signalizira, da je naprava aktivna v proceduri posodobitve časa.
- V primeru, da mobilni terminal premore še svetlobni indikator naj slednji v tem primeru sveti brez premora z rdečo barvo, kar signalizira nepripravljenost za izvajanje interakcij z uporabnikom.
- Za lažjo razpoznavo akcij mobilnega terminala (kaj dejansko počne) se priporoča še uporaba majhnih ikon, ki jo sestavlja črka in številka. V tem primeru "T1", kar pomeni usklajevanje systemskega časa terminala s časom zalednega sistema.
- Prikaz statusnih ikon na zaslonu naj se izvaja v zgornji vrstici desne polovice zaslona.
- Ostale interakcije z mobilnim terminalom se v tem stanju preprečijo.
- Če posodobitev časa ne uspe, se mora procedura ponavljati do uspešne posodobitve.
- Po uspešnem prenosu, in uspešnih predhodnih procedurah (prisotna in aktivna SIM kartica, registracija v GPRS omrežje, prenesena tarifa, prenesen seznam kartic na črni listi, prenesen seznam vozovnic na črni listi) se mobilni terminal postavi v stanje pripravljenosti za interakcijo z uporabnikom (naslednje GUI stanje).



#### 16.13.11 GUI - PRIJAVA VOZNIKA PREVOZNEGA SREDSTVA IJPP

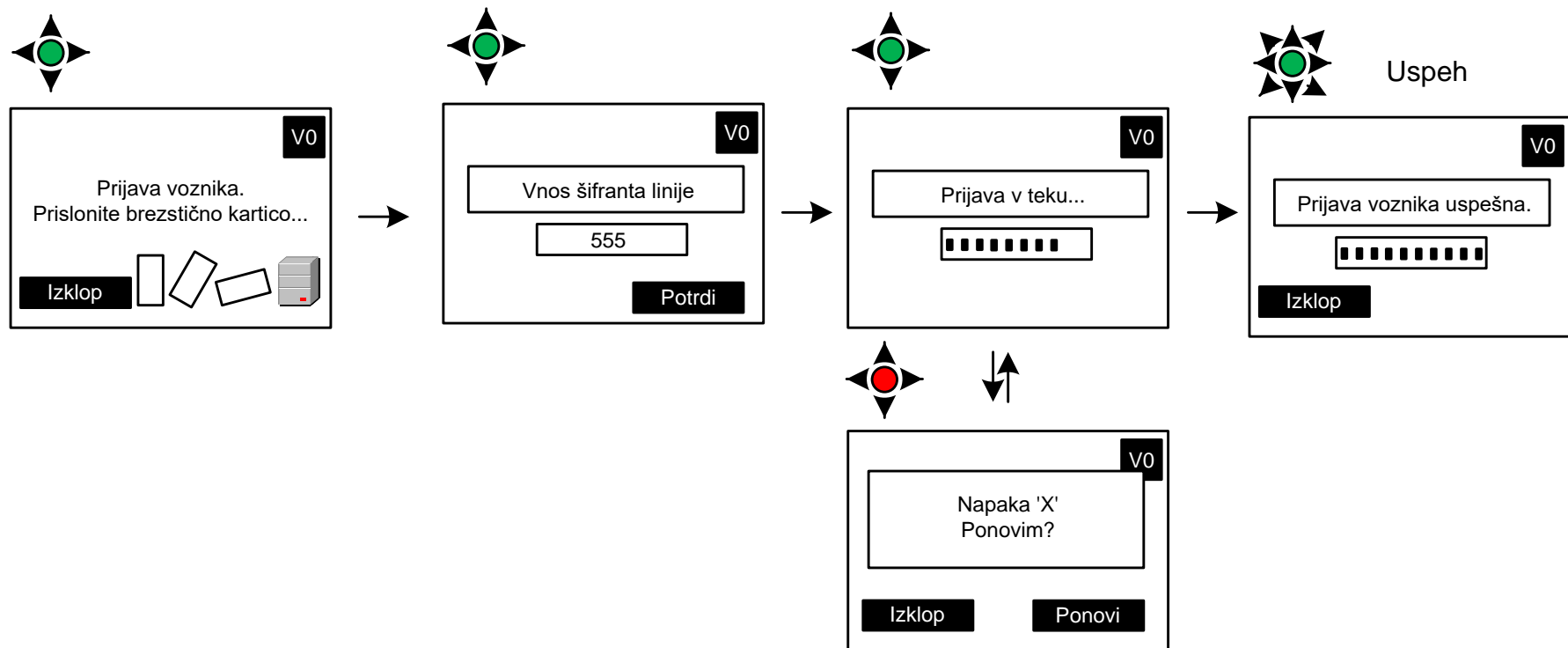
Ko so predhodni pogoji v celoti realizirani in izpolnjeni sledi stanje za prijavo voznika v mobilni terminal na specifičnem prevoznem sredstvu (vlak, avtobus). Stanje se avtomatično izvede, če so vsi predhodni (opisani) postopki uspešno zaključeni. Možna je prijava samo enega uporabnika (voznika, sprevodnika), drugi se lahko prijavi šele, ko se prvi odjavi. Zaslonski zaslon za prijavo voznika naj vsebuje naslednje informacije:

- Sporočilo: "Prijava voznika. Prislonite brezstično kartico..."
- Animacija: slednja naj vsebuje postopek koračnega približevanja kartice NFC bralno-pisalni enoti mobilnega terminala.
- V primeru, da mobilni terminal premore še svetlobni indikator naj slednji v tem primeru utripa na vsakih 500ms z zeleno barvo, s čimer signalizira pripravljenost za prvo interakcijo z uporabnikom (voznikom).
- Ostale interakcije z mobilnim terminalom razen prijave voznika se v tem stanju preprečijo.
- Za lažjo razpoznavo akcij mobilnega terminala (kaj dejansko počne) se priporoča še uporaba majhnih ikon, ki jo sestavlja črka in številka. V tem primeru "V0", kar pomeni da terminal čaka na prijavo voznika.
- V trenutku prislonitve brezstične kartice voznika k mobilnemu terminalu se predhodno opisana vsebina zaslona zamenja z odstotnim indikatorjem poteka, ki ponazarja potek transakcije v kateri se v zaledni sistem posreduje voznikov ID in čas prijave. Naslednji korak v proceduri prijave voznika je vnos šifrantne linije na kateri vozi avtobus. Na osnovi šifrantne validator

enostavno ugotovi, katere postaje pripadajo liniji, s čimer omogoča pravilno izvedbo validacij. Po uspešni izvedbi prijave in vnosa šifrant terminal za kratek čas prikaže t.i. 'Pop-Up' sporočilo o uspešni prijavi za čas cca. 3 sekund. Zraven uspešne prijave se priporoča še uporaba zvočnega signala, in sicer uporaba zvoka za uspešno izvedeno interakcijo (glej definicije na začetku poglavja o GUI).

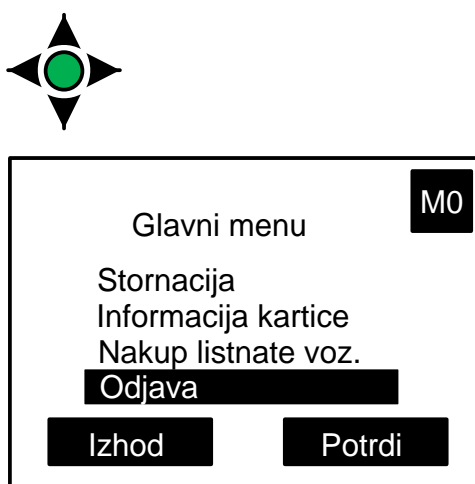
- Po podani informaciji o uspešni prijavi mobilni terminal preide v osnovno stanje za izvajanje validacij (t.i. idle stanje).
- V tem stanju naj v levem zgornjem kotu, terminal prikazuje ime in prvo črko priimka voznika (npr. Andrej K.).
- Svetlobni indikator začne po uspešni prijavi svetiti zeleno za 3 sekunde in po preteku 3 sekund - vrnitvi v osnovno stanje utripati na vsakih 1000ms, kar uporabnikom sporoča, da je v stanju pripravljenosti za izvajanje validacij (glej stanje 16.13.13). Na zaslonu so prikazani podatki trenutno prijavljenega uporabnika.
- Če prijava voznika v mobilni terminal ne uspe, se slednjega prav tako obvesti s pojavnim sporočilom (cca. 3 sekunde), da je prijava neuspešna. Zraven neuspešne prijave se priporoča še uporaba zvočnega signala, in sicer zvoka za neuspešno interakcijo (glej definicijo na začetku dokumenta). V tem primeru se prav tako za ta čas prižge svetlobni indikator v rdeči barvi.
- Po neuspešni prijavi se ponovno izpiše identična vsebina zaslona, kot je definirana na začetku tega stanja.
- Uspešna prijava voznika je predpogoj za izvajanje naslednjih korakov, kot so validacije, nakupi, in nenazadnje tudi odjava voznika.





### 16.13.12 GUI - ODJAVA VOZNIKA PREVOZNEGA SREDSTVA IJPP IN AVTOMATIČNO TISKANJE POROČIL

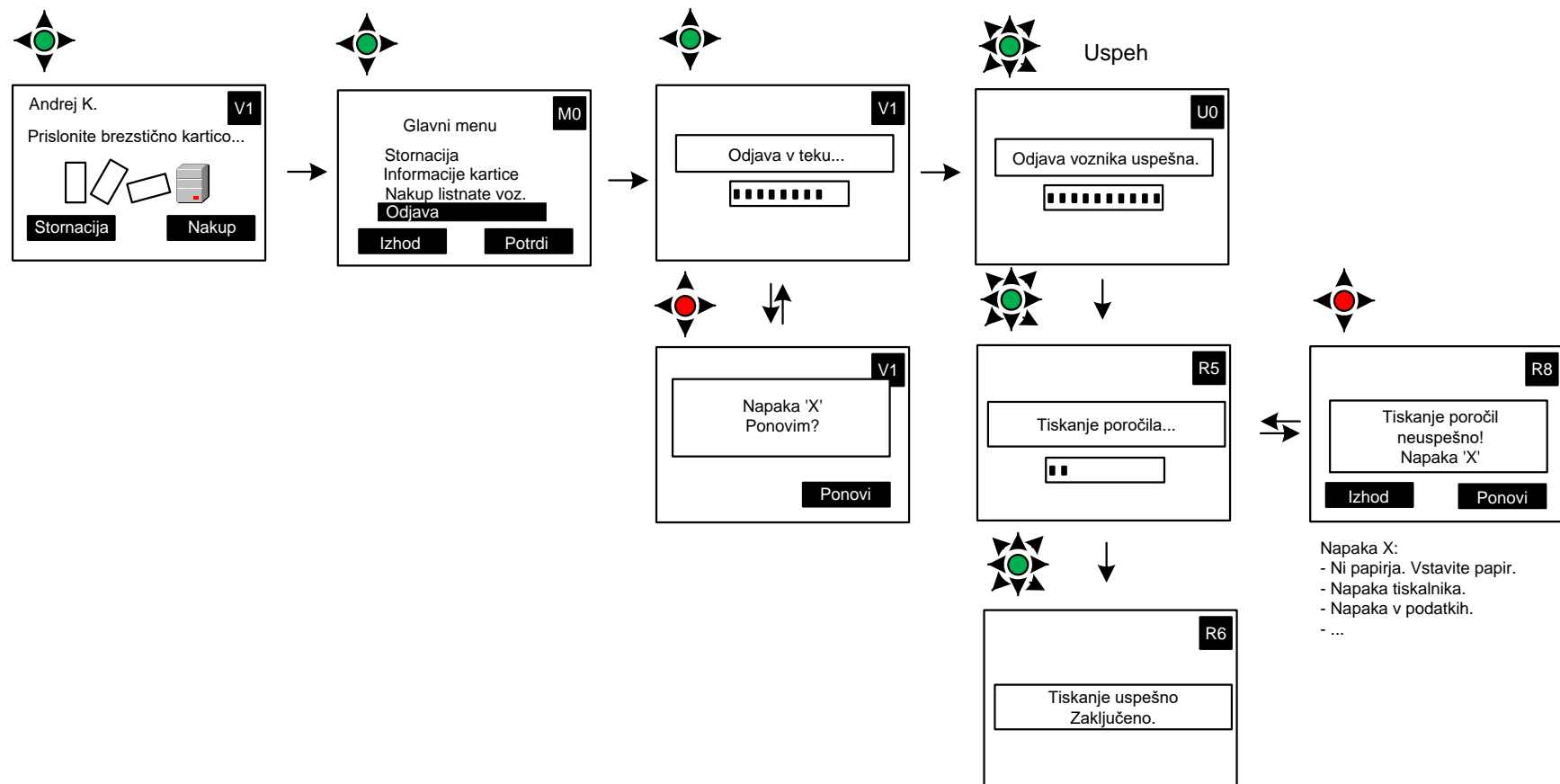
Ko je voznik v mobilni terminal že prijavljen se mora po izteku voženj v delavniku tudi odjaviti. Odjava je izvedena z dostopom do te možnosti preko glavnega menija. Do menija lahko uporabnik dostopa s pomočjo funkcijske tipke 'Menu' na terminalu. Glavni menu z možnostjo odjave prikazuje naslednja slika:



Po izbiri in potrditvi možnosti 'Odjava', mora terminal voznika odjaviti, in pri tem prikazati sledečo vsebino:

- V trenutku odjave voznika se predhodno opisana vsebina zaslona zamenja z odstotnim indikatorjem poteka, ki ponazarja potek transakcije v kateri se v zaledni sistem posreduje voznikov ID in čas odjave. Po uspešni izvedbi odjave in transakcije terminal za kratek čas prikaže t.i. 'Pop-Up' sporočilo o uspešni odjavi za čas cca. 3 sekund. Zraven uspešne odjave se priporoča še uporaba zvočnega signala, in sicer zvoka za uspešno izvedeno interakcijo. Svetlobni indikator v primeru uspeha za ta čas sveti zeleno.
- 
- Ostale interakcije, razen ponovne prijave so po tem koraku z mobilnim terminalom onemogočene.
- Po uspešni odjavi se izvede avtomatsko tiskanje finančnih poročil, ki jih je voznik opravil v danem terminu od prijave do odjave.
- Po uspešnem tiskanju se za lažjo razpoznavo akcij mobilnega terminala (kaj dejansko počne) priporoča še uporaba statusnih ikon, ki jo sestavlja črka in številka. V tem primeru "V0", kar pomeni da terminal čaka na ponovno prijavo voznika.

- Po podani informaciji o uspešni odjavi in tiskanju finančnih poročil mobilni terminal preide v stanje prijave (glej stanje 16.13.11), kar omogoča naslednjemu vozniku prijavo v mobilni terminal in nadaljnje izvajanje interakcij s končnimi uporabniki.
- Svetlobni indikator naj po povratku v stanje prijave (stanje 16.13.11) utripa na vsakih 1000ms z zeleno barvo, s čimer ponovno signalizira pripravljenost za prvo interakcijo z uporabnikom (voznikom).
- Če odjava voznika v mobilni terminal ne uspe, se slednjega prav tako obvesti s pojavnim sporočilom (cca. 3 sekunde), da je odjava neuspešna ("Odjava neuspešna!"). Zraven 'popup' sporočila o neuspešni odjavi se priporoča še uporaba zvočnega signala za neuspešno interakcijo (glej definicijo na začetku dokumenta). V tem času naj svetlobni indikator sveti rdeče.
- Po neuspešni odjavi se ponovno izpiše identična vsebina zaslona, kot je definirana na začetku tega stanja (začetek stanja 16.13.13).



Primer poročila izmene IJPP prevoznega sredstva, ki se avtomatično natisne ob odjavi voznika:

**Primer poročila izmene**

<b>LOGOTIP IJPP</b>		
Potniški promet XY d.o.o. Pod Rjavčevo 36, 2000 Maribor		
Datum/čas: 28.10.2015/14:57:57 ID Terminala: 77777 ID prevoznega sredstva: 27-MB ID voznika: 36066995138405380		
<hr/>		
<b>Poročilo izmene</b>		
Od: 28.10.2015 / 08:00:27 Do: 28.10.2015 / 14:01:27		
<hr/>		
<b>Nakup</b>		
LISTNATA VOZOVNICA		
Zap. št.:	Datum/Čas	Vrednost:
1	28.10.2015/08:02:22	1,30€
2	28.10.2015/10:04:17	3,30€
3	28.10.2015/13:57:07	3,10€
Skupaj transakcij:		3
Vrednost:		7,70€
<hr/>		
<b>Skupno</b>		
Skupaj transakcij:		3
Vrednost:		7,70€

**16.13.13 GUI – VALIDACIJA – MESTNI, MEDKRAJEVNI PROMET**

Ko so vsi predhodni kriteriji že izpolnjeni (pravilno vstavljena SIM, registracija v omrežje, posodobljena tarifa, črne liste, prijava voznika, ...) je mobilni terminal pripravljen za izvajanje validacij. Ko je mobilni terminal v pripravljenosti izvršiti validacijo, mora uporabniški vmesnik (GUI) vsebovati naslednje komponente:

**Pred validacijo:**

- Sporočilo: " Prislomite brezstično kartico..."
- Animacija: slednja naj vsebuje postopek koračnega približevanja kartice NFC bralno-pisalni enoti mobilnega terminala.
- V primeru, da mobilni terminal premore še svetlobni indikator mora slednji v tem stanju pripravljenosti imeti vključen indikator, ki odraža zeleno barvo in utripa na periodo 1000ms. S tem signalizira pripravljenost za interakcije z uporabniki.

- Na primarnem zaslonu mora terminal ponujati še možnost za nakup, ki mora biti usklajena s pripadajočo tipko v primeru integrirane tipkovnice oz. podprto možnost na dotik v primeru uporabe zaslona na dotik.
- Vse ostale interakcije z mobilnim terminalom so v tem primeru omogočene (nakup, validacija, odjava).
- Za lažjo razpoznavo akcij mobilnega terminala (kaj dejansko počne) se priporoča še uporaba statusnih ikon, ki jo sestavlja črka in številka. V tem primeru "V1", kar pomeni da terminal čaka v pripravljenosti za izvedbo validacije.

#### ***Med validacijo:***

- V trenutku prislonitve brezstične kartice IJPP k mobilnem terminalu nameščenem pri vozniku prevoznega sredstva IJPP se pri proceduri branja najprej spremeni stanje svetlobnega indikatorja, ki v tem postopku sveti zeleno, brez prekinitve in utripanja.

#### ***Po uspešni validaciji:***

- Po uspešni izvedbi validacije, mora mobilni terminal to signalizirati z zvokom za uspešno izvedeno interakcijo (glej definicijo na začetku poglavja o GUI).
- Po uspešno izvedeni validaciji naj svetlobni indikator ostane vključen za čas cca. 3 sekunde, v katerem sveti zeleno, brez utripanja.
- V tem času se morajo na zaslon izpisati naslednji podatki o uspešni validaciji za

##### *Mestni promet:*

- Sporočilo: "Validacija uspešna"
- Čas izvedene validacije v obliki: hh:mm:ss
- Datum izvedene validacije v obliki: dd:mm:yyyy
- Informacijo o časovni veljavnosti
- Informacijo o veljavnosti cone

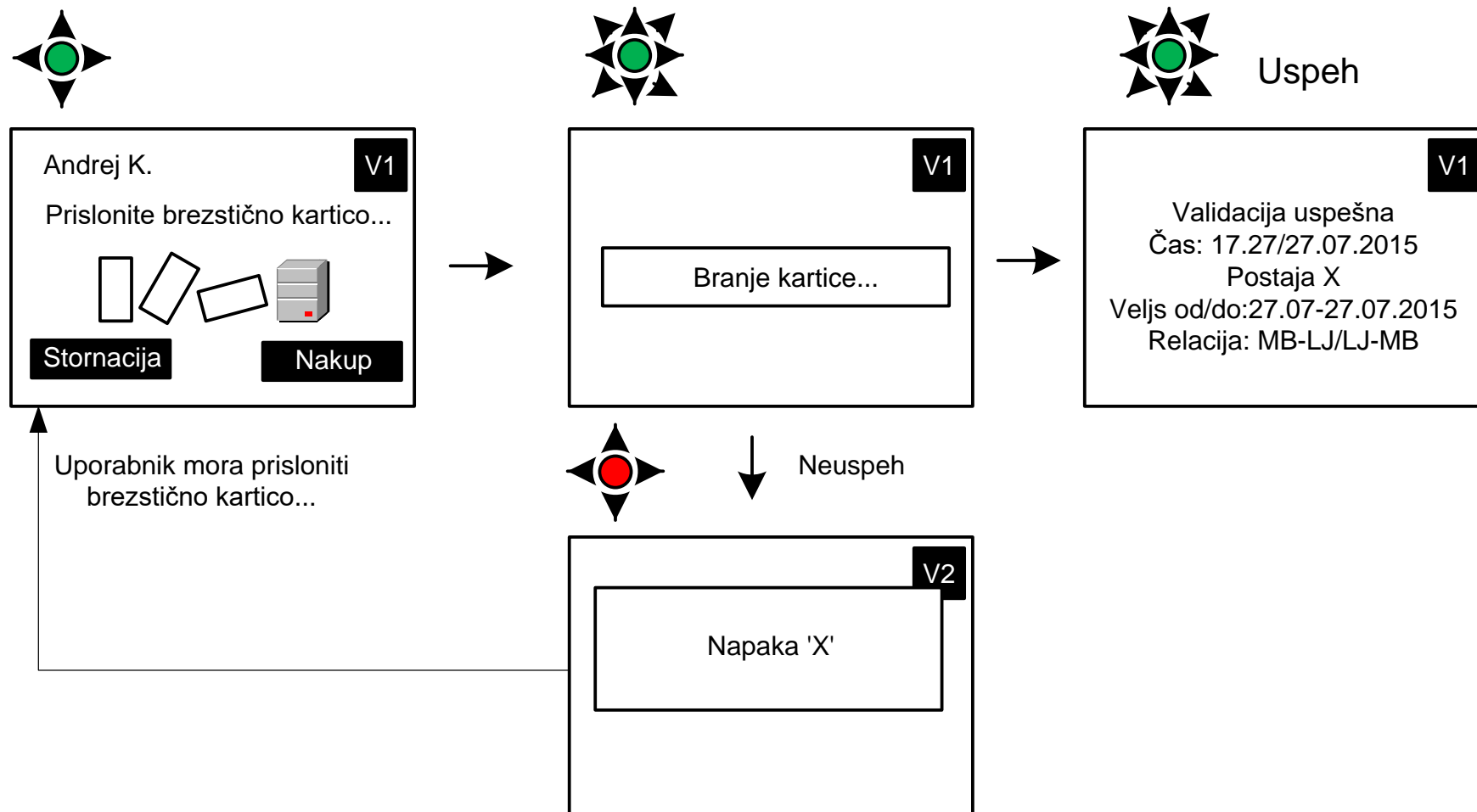
##### *Medkrajevni promet:*

- Sporočilo: "Validacija uspešna"
- Čas izvedene validacije v obliki: hh:mm:ss
- Datum izvedene validacije v obliki: dd:mm:yyyy
- Informacijo o časovni veljavnosti vozovnice za medkrajevni promet
- Informacijo o vrsti vozovnice (enosmerna, povratna)
- Informacijo o liniji za katero je namenjena vozovnica
- Za lažjo razpoznavo akcij mobilnega terminala (kaj dejansko počne) se priporoča še uporaba statusnih ikon, ki jo sestavlja črka in številka. V tem primeru "V1", kar pomeni da terminal prikazuje podatke uspešne validacije.
- Za prikaz statusne ikone se naj uporablja prva vrstica zgornjega desnega dela zaslona.

- Po preteku omenjenega časa 3 sekunde, se mora terminal vrniti v začetno stanje pripravljenosti za sprejem nove kartice, kot je opisano v začetku tega sklopa (Idle stanje s sporočilom za prislonitev kartice in možnostjo nakupa). Skladno s avtomatskim preходом v to stanje se mora tudi svetlobni indikator postaviti v stanje pripravljenosti (utripanje z zeleno barvo v periodi 1000ms).
- Terminal (čitalna komponenta) je po uspešno izvedeni validaciji pripravljen za branje nove IJPP kartice, čeprav zaslon in svetlobni indikatorji, prikazujejo še uspešnost validacije.

***Po neuspešni validaciji:***

- Po neuspešni izvedbi validacije, mora mobilni terminal to signalizirati z zvokom za neuspešno izvedeno interakcijo (glej definicijo na začetku poglavja o GUI).
- Po neuspešno izvedeni validaciji naj svetlobni indikator ostane vključen za čas cca. 3 sekunde, v katerem sveti rdeče, brez utripanja.
- V tem času se mora na zaslon izpisati naslednja vsebina:
  - Sporočilo: "Validacija neuspešna!"
  - Vzrok za neuspešno validacijo, npr.:
    - "Neustrezen produkt"
    - "Neveljaven produkt"
    - "Ni produkta"
    - "Kartica na črni listi"
    - "Vozovnica na črni listi"
    - "Že validiran produkt"
    - "Neveljavna kartica"
    - "Nepoznana kartica"
    - "..."
- Za lažjo razpoznavo akcij mobilnega terminala (kaj dejansko počne) se priporoča še uporaba statusnih ikon, ki jo sestavlja črka in številka. V tem primeru "V2", kar pomeni da terminal prikazuje sporočilo neuspešne validacije.
- Za prikaz statusne ikone se naj uporablja prva vrstica zgornjega desnega dela zaslona.
- Po preteku omenjenega časa 3 sekunde, se mora terminal vrniti v začetno stanje pripravljenosti za sprejem nove kartice, kot je opisano v začetku tega sklopa (Idle stanje s sporočilom za prislonitev kartice in možnostjo nakupa). Skladno s avtomatskim preходом v to stanje se mora tudi svetlobni indikator postaviti v stanje pripravljenosti (utripanje z zeleno barvo s periodo 1000ms).
- Terminal (čitalna komponenta) je po neuspešno izvedeni validaciji ponovno v pripravljenosti za branje nove IJPP kartice, čeprav zaslon in svetlobni indikatorji prikazujejo še sporočilo o neuspešni validaciji.





#### **16.13.14 GUI – PODALJŠEVANJE TERMINSKIH VOZOVNIC NA MOBILNEM TERMINALU, MESTNI IN MEDKRAJEVNI PROMET**

Ko je terminal v stanju pripravljenosti interakcije z uporabnikom, se lahko na njemu izvede tudi nakup produkta iz skupine vseh razpoložljivih za dano lokacijo. Predpogoj za izvedbo nakupa je aktivna brezstična kartica IJPP z vsemi statusi in pravicami uporabnika, ki jo je pridobil pri ponudniku. Z izbiro opcije nakupa v osnovnem stanju (izbira preko tipke v primeru integrirane tipkovnice oz. izbira preko zaslona na dotik), se uporabniku najprej pokaže meni, ki vsebuje naslednje informacije:

##### ***Poziv za prislonitev brezstične kartice:***

- Naslov sporočila: "Za nakup prislonite brezstično kartico"
- Animacija: slednja naj vsebuje postopek koračnega približevanja kartice NFC bralno-pisalni enoti mobilnega terminala.
- V primeru, da mobilni terminal premore še svetlobni indikator mora slednji v tem stanju imeti vključen indikator, ki odraža zeleno barvo in utripa na periodo 1000ms. S tem signalizira pripravljenost za interakcijo z uporabnikom.
- Terminal mora uporabniku prav tako omogočiti prehod v predhodno (začetno) stanje, bodisi preko korespondenčne tipke za opcijo "Izhod" oziroma preko opcije na zaslonu občutljivem na dotik - če ga le ta premore.

##### ***Prislonitev kartice za pričetek procedure nakupa vozovnice:***

- V trenutku prislonitve brezstične kartice IJPP k mobilnem terminalu nameščenem pri vozniku prevoznega sredstva IJPP se pri proceduri branja najprej spremeni stanje svetlobnega indikatorja, ki v tem postopku sveti zeleno, brez prekinitve in utripanja.
- Glede na uspešnost/neuspešnost branja se uporabnika opozori tudi z zvočnim signalom za uspeh/neuspeh (glej definicijo)

##### ***Po postopku prislonitve brezstične kartice za pričetek procedure nakupa vozovnice:***

Terminal glede na status uporabnika s pomočjo tarifnih razredov slednjemu najprej ponudi v izbiro možnosti za kateri promet želi terminsko vozovnico:

- Conski promet
- Medkrajevni promet

V kolikor terminal iz podatkov IJPP kartice razbere, da se uporabniku dovoljuje nakup samo ene izmed naštetih opcij, prikaže na zaslon samo to možnost. Ko je predhodna možnost potrjena terminal preide na naslednje stanje. V tem primeru terminal glede na status uporabnika s pomočjo tarifnih razredov prikaže v novem meniju možne nakupe vozovnic, npr.:

- Dnevna
- Tedenska

- Mesečna
  - Polletna
  - Letna
  - itn.
- V primeru, da mobilni terminal premore še svetlobni indikator mora slednji v tem stanju pripravljeno imeti vključen indikator, ki odraža zeleno barvo in utripa na periodo 1000ms. S tem signalizira pripravljenost za interakcije z uporabniki.
  - Terminal mora uporabniku prav tako omogočiti prehod v predhodno (začetno) stanje bodisi preko korespondenčne tipke za možnost "Izhod" oziroma opcije na zaslonu občutljivem na dotik.
  - Uporabnik iz menija izbere ustrezno opcijo, in jo potrdi s korespondenčnimi tipkami, npr. uporaba numeričnih, navigacijskih tipk za posamezno opcijo izbire v primeru integrirane tipkovnice, oziroma s pomočjo dotika na zeleno opcijo na zaslonu občutljivem na dotik, v kolikor mobilni terminal le tega premore.
  - Za lažjo razpoznavo akcij mobilnega terminala (kaj dejansko počne) se priporoča še uporaba majhnih ikon, ki jo sestavlja črka in številka. V tem primeru "N0", kar pomeni da terminal prikazuje meni z možnimi opcijami za nakup.

***Po potrditvi opcije za nakup se na zaslonu prikažejo sledeči podatki:***

- Znesek nakupa, ki ga uporabnik poravnava vozniku samo z denarnim plačilnim sredstvom oz. možna izbira plačilnega sredstva.
- Vrsta vozovnice, ki se kupuje (npr. tedenska, relacija "XY" za medkrajevni promet, oz. npr. tedenska, cona 'B-MB' za mestni promet)
- Veljavnost od/do v obliki (3.10. - 10.10.2015)

Po potrditvi izbrane opcije nakupa se podatki posredujejo do voznika, ki preveri vnešene podatke in jih potrdi. Šele ko voznik potrdi željen nakup se uporabnika pozove k prislonitvi brezstične IJPP kartice:

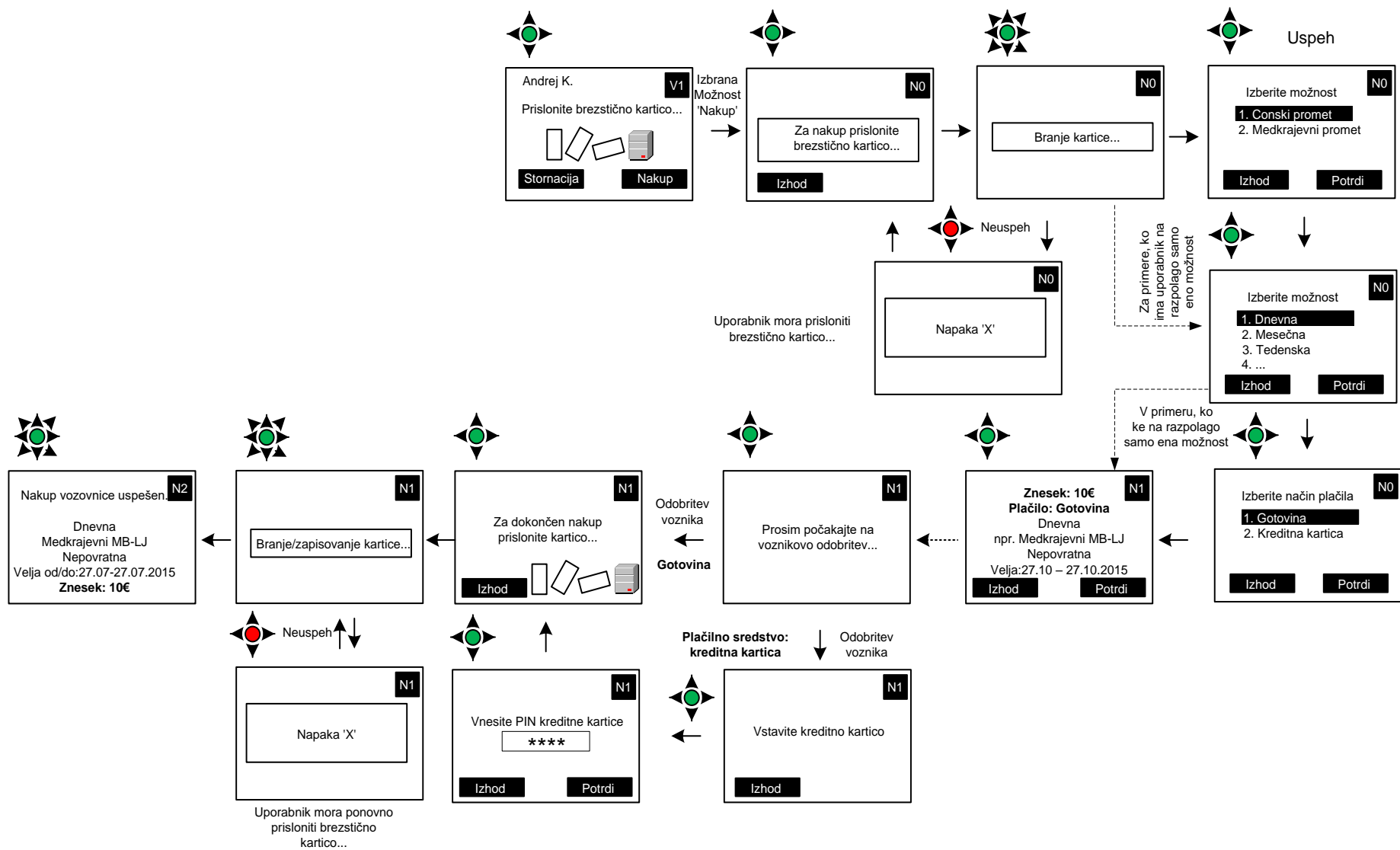
- Naslov sporočila: "Za dokončen nakup ponovno prislonite brezstično kartico"
- Animacija: slednja naj vsebuje postopek koračnega približevanja kartice NFC bralno-pisalni enoti mobilnega terminala.
- V primeru, da mobilni terminal premore še svetlobni indikator mora slednji v tem stanju imeti vključen indikator, ki odraža zeleno barvo in utripa na periodo 1000ms. S tem signalizira pripravljenost za interakcijo z uporabnikom.
- Terminal mora uporabniku prav tako omogočiti prehod v predhodno (začetno) stanje, bodisi preko korespondenčne tipke za opcijo "Izhod" oziroma preko opcije na zaslonu občutljivem na dotik - če ga le ta premore.
- Za lažjo razpoznavo akcij mobilnega terminala (kaj dejansko počne) se priporoča še uporaba statusnih ikon, ki jo sestavlja črka in številka. V tem primeru "N1", kar pomeni da je terminal v zadnjem koraku pred izvršitvijo dokončnega nakupa (čaka samo še na prislonitev kartice).

### ***Prislonitev brezstične kartice za finalizacijo procedure nakupa vozovnice:***

- V trenutku prislonitve brezstične kartice IJPP k mobilnem terminalu nameščenem pri vozniku prevoznega sredstva IJPP se pri proceduri branja najprej spremeni stanje svetlobnega indikatorja, ki v tem postopku sveti zeleno, brez prekinitve in utripanja.
- Glede na uspešnost/neuspešnost branja se uporabnika opozori tudi z zvočnim signalom za uspeh/neuspeh (glej definicijo)

### ***Po prislonitvi brezstične kartice za zaključek nakupa vozovnice:***

- Terminal glede na izveden zaslon za čas cca. 3 sekunde izpiše podatke o nakupu, ki so sledeči:
  - Sporočilo o nakupu: "Nakup vozovnice uspešen."
  - Vrsta vozovnice (dnevna, tedenska, mesečna, polletna, letna)
  - V primeru mestnega prometa: Cono uporabnosti kupljene vozovnice
  - V primeru medkrajevnega prometa: relacijo za katero je bila kupljena vozovnica
  - Veljavnost od/do (npr. 3.10.2015-10.10.2015)
  - Znesek
- V času prikazovanja predhodnih podatkov mora terminal prav tako natisniti potrdilo o nakupu, na katerem mora biti zraven zgoraj že navedenih podatkov še:
  - Datum nakupa
  - Referenčna številka
  - ID naprave na kateri se je izvršil nakup
  - ID prevoznikaUporabniki se omogoči hkrati tudi tisaknje duplikata z istimi podatki, kot so na potrdilu o nakupu za potnika.
- V primeru, da mobilni terminal premore še svetlobni indikator mora imeti v tem časovnem intervalu (cca. 3 sekunde) vključen indikator, ki odraža zeleno barvo in ne sme utripati. S tem signalizira uspešen zaključek interakcije.
- Po preteku definiranega časa se mora mobilni terminal avtomatično vrniti v začetno stanje, kot je definirano (pripravljenost za izvedbo validacije z možnostjo nakupa, utripanje zelenega svetlobnega indikatorja na periodo 1000ms).
- Za lažjo razpoznavo akcij mobilnega terminala (kaj dejansko počne) se priporoča še uporaba statusnih ikon, ki jo sestavlja črka in številka. V tem primeru "N2", kar pomeni da je uporabnik na mobilnem terminalu uspešno izvedel nakup specifične vozovnice.



### **16.13.15 GUI – STORNACIJA VOZOVNIC NA MOBILNEM TERMINALU, MESTNI, MEDKRAJEVNI PROMET**

Ko je terminal v stanju pripravljenosti interakcije z uporabnikom, se lahko na njemu izvede tudi stornacija nakupa produkta oziroma več produktov. Stornacija je lahko izvedena samo za ravnokar kupljen produkt in hkrati na terminalu na katerem se je izvedel nakup, s čimer se preprečijo zlorabe, kjer bi lahko tretja oseba v primeru tatvine odtujila denar v protivrednosti naloženih produktov. Predpogoj za izvedbo nakupa je aktivna brezstična kartica IJPP z vsemi statusi in pravicami uporabnika, ki jo je pridobil pri ponudniku. Do stornacijskega menija se dostopa s pomočjo tipke 'Menu' če jo terminal pripomore, oziroma ekvivalentne povezave, ki je jasno označena in razumljiva uporabnikom. Z izbiro opcije stornacije se pokaže sledeča vsebina:

#### ***Poziv za prislonitev brezstične kartice:***

- Naslov sporočila: "Za stornacijo vozovnice prislonite brezstično kartico"
- Animacija: slednja naj vsebuje postopek koračnega približevanja kartice NFC bralno-pisalni enoti mobilnega terminala.
- V primeru, da mobilni terminal premore še svetlobni indikator mora slednji v tem stanju imeti vključen indikator, ki odraža zeleno barvo in utripa na periodo 1000ms. S tem signalizira pripravljenost za interakcijo z uporabnikom.
- Terminal mora uporabniku prav tako omogočiti prehod v predhodno (začetno) stanje, bodisi preko korespondenčne tipke za opcijo "Izhod" oziroma preko opcije na zaslonu občutljivem na dotik - če ga le ta premore.

#### ***Prislonitev kartice za pričetek procedure stornacije vozovnice:***

- V trenutku prislonitve brezstične kartice IJPP k mobilnem terminalu nameščenem pri vozniku prevoznega sredstva IJPP se pri proceduri branja najprej spremeni stanje svetlobnega indikatorja, ki v tem postopku sveti zeleno, brez prekinitev in utripanja.
- Glede na uspešnost/neuspešnost branja se uporabnika opozori tudi z zvočnim signalom za uspeh/neuspeh (glej definicijo)

#### ***Po postopku prislonitve brezstične kartice za pričetek procedure stornacije vozovnice:***

Če terminal ugotovi, da je bil produkt prodan na isti napravi omogoči uporabniku stornacijo, tako, da pozove uporabnika za ponovno prislonitev kartice, za dokončno izvedbo stornacije produkta.

- Sporočilo: »Stornacija produktov«
- Stornirani produkti: »Produkt A«
- Znesek vračila: npr. 27,70€ (krepko)

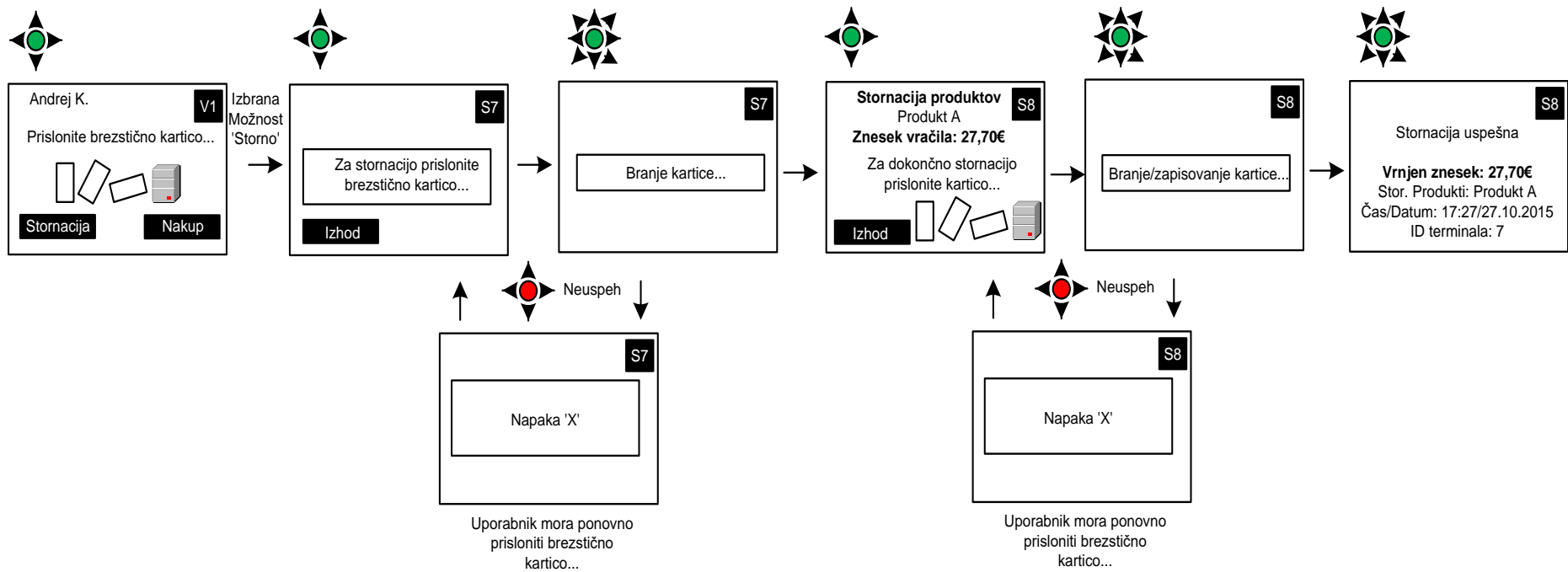
- Sporočilo za prislonitev kartice. »Ponovno prislonite brezstično kartico za zaključitev stornacije«.
- Animacijo s koračnim približevanjem brezstične kartice k bralno-pisalni enoti mobilnega terminala.
- V primeru, da mobilni terminal premore še svetlobni indikator mora slednji v tem stanju pripravljenosti imeti vključen indikator, ki odraža zeleno barvo in utripa na periodo 1000ms. S tem signalizira pripravljenost za interakcije z uporabniki.
- Terminal mora uporabniku prav tako omogočiti prehod v predhodno (začetno) stanje bodisi preko korespondenčne tipke za možnost "Izhod" oziroma opcije na zaslonu občutljivem na dotik.
- Za lažjo razpoznavo akcij mobilnega terminala (kaj dejansko počne) se priporoča še uporaba statusnih ikon, ki jo sestavlja črka in številka. V tem primeru "S7", kar pomeni da terminal prikazuje informacijo o produktu, ki je na voljo za stornacijo.

#### ***Prislonitev brezstične kartice za pričetek končne procedure stornacije vozovnice:***

- V trenutku prislonitve brezstične kartice IJPP k mobilnem terminalu nameščenem pri vozniku prevoznega sredstva IJPP se pri proceduri branja/pisanja najprej spremeni stanje svetlobnega indikatorja, ki v tem postopku sveti zeleno, brez prekinitve in utripanja.
- Glede na uspešnost/neuspešnost branja/pisanja se uporabnika opozori tudi z zvočnim signalom za uspeh/neuspeh (glej definicijo)

#### ***Po prislonitvi brezstične kartice za finalizacijo stornacije vozovnice:***

- Terminal glede na izveden zaslon za čas cca. 3 sekunde izpiše podatke o stornaciji, ki so sledeči:
  - Sporočilo o stornaciji: "Stornacija uspešna."
  - Vrnjen znesek: 27,70€ (krepko)
  - Stornirani produkti: »Produkt A«
  - Čas in datum stornacije v obliki: »Čas/datum storno: 17:27/27.10.2015«
  - Identiteta terminala na katerem se je izvršila stornacija
- V času prikazovanja predhodnih podatkov mora terminal prav tako natisniti potrdilo o stornaciji, na katerem mora biti zraven zgoraj že navedenih podatkov še:
  - Referenčna številka
  - ID prevoznika
- V primeru, da mobilni terminal premore še svetlobni indikator mora imeti v tem časovnem intervalu (cca. 3s) vključen indikator, ki odraža zeleno barvo in ne sme utripati. S tem signalizira uspešen zaključek interakcije.
- Po preteku definiranega časa se mora mobilni terminal avtomatično vrniti v začetno stanje, kot je definirano (pripravljenost za izvedbo validacije z možnostjo nakupa, utripanje zelenega svetlobnega indikatorja na periodo 1000ms).
- Za lažjo razpoznavo akcij mobilnega terminala (kaj dejansko počne) se priporoča še uporaba statusnih ikon, ki jo sestavlja črka in številka. V tem primeru "S8", kar pomeni da je uporabnik na mobilnem terminalu uspešno izvedel stornacijo produktov.

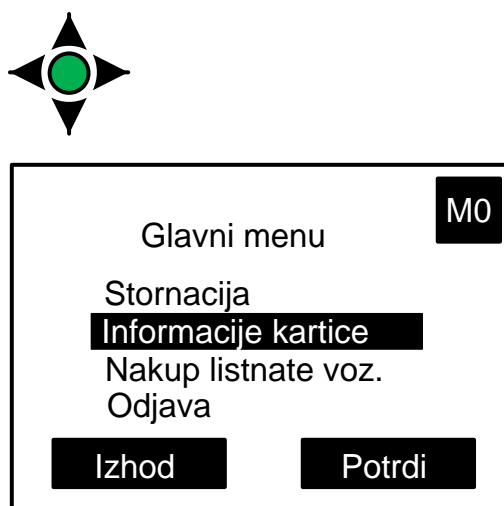


### 16.13.16 GUI – POIZVEDBA O VSEBINI BREZSTIČNE IJPP KARTICE (INFORMACIJA KARTICE)

Ko je terminal v stanju pripravljenosti interakcije z uporabnikom mora uporabnik preko korespondenčne tipke 'Menu' oziroma ekvivalentne opcije, ki je razumljiva uporabniku dostopati do glavnega terminalskega menija. V meniju se mora izbrati možnost 'Informacija kartice', s čimer se spriži postopek za poizvedbo informacij, ki so shranjene na kartici (naloženi produkti - vozovnice). Storitev je namenjena vsem uporabnikom, in jim omogoča vpogled v produkte, ki so jih kupili. Predpogoj za izvedbo procedure poizvedbe je aktivna brezstična kartica IJPP z vsemi statusi in pravicami uporabnika, ki jo je pridobil pri ponudniku.

#### **Poziv za prislonitev brezstične kartice:**

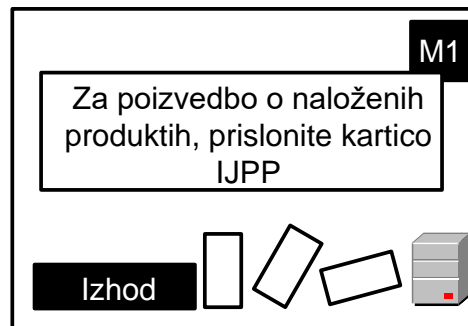
Po pritisku tipke 'Menu' oziroma ekvivalentne povezave se uporabniku prikaže sledeča vsebina zaslona:



Uporabnik na zaslonu izbere možnost 'Informacije kartice' in jo potrdi s pomočjo dotika (zaslon na dotik) oziroma s pomočjo navigacijskih in potrditvenih korespondenčnih tipk. Po izbrani možnosti 'Card info' glavnega terminalskega menija, se uporabniko prikaže naslednji zaslon z naslednjo vsebino:

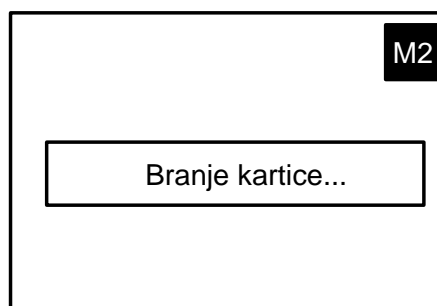
- Sporočilo: »Za poizvedbo o naloženih produktih, prislonite brezstično kartico IJPP«.
- V primeru svetlobnega indikatorja, mora slednji v tem koraku utripati na periodo 1000ms z zeleno barvo.
- Animacija, ki prikazuje postopek prislonitve brezstične IJPP kartice k mobilnemu terminalu.
- Statusno ikono 'M1', ki signalizira stanje terminala v katerem se pričakuje prislonitev IJPP kartice za izvedbo procedure poizvedbe o naloženih produktih.





#### ***Prislonitev kartice za pričetek procedure poizvedbe o naloženih produktih***

- V trenutku prislonitve brezstične kartice IJPP k mobilnem terminalu nameščenem pri vozniku prevoznega sredstva IJPP se pri proceduri branja najprej spremeni stanje svetlobnega indikatorja, ki v tem postopku sveti zeleno, brez prekinitve in utripanja.
- Glede na uspešnost/neuspešnost branja se uporabnika opozori tudi z zvočnim signalom za uspeh/neuspeh (glej definicijo).
- V proceduri branja informacij o kartici, se statusna ikona spremeni v stanje 'M2'.



#### ***Po postopku prislonitve brezstične kartice v proceduri 'Informacija kartice'***

*V primeru uspešnega branja, naj terminal pokaže naslednjo vsebino, če so na kartici naloženi produkti:*

- Zvok za uspeh (glej definicijo)
- Kartica aktivna
- Status kartice

- Podatki imetnika kartice
- Naloženi produkti:
  - Mesečna (Cona A/MB)
  - Dnevna (Relacija MB-KP)
  - Letna (Cona B/LJ)
  - ...



Kartica aktivna/veljavna

Imetnik: Jože Dolgonogi

Naloženi produkti

Mesečna (Cona A/MB)

Dnevna (Relacija MB/KP) ▲

Letna (Cona B/LJ) ▼

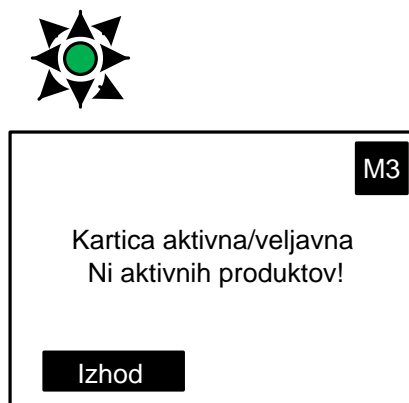
M3

Izhod

Vsebina se mora izpisovati do nastopa 'timeouta' (cca. 5 sekund) oziroma do uporabnikovega ročnega izhoda z izbiro možnosti 'Izhod'. V vsem tem času mora biti svetlobni indikator vključen, in neprestano brez utripanja svetiti zeleno. V primeru večih produktov, ki ne gredo na en zaslon, mora biti izvedena podpora 'scroll down/up' – pomikanje navzol in navzgor.

*V primeru uspešnega branja, naj terminal pokaže naslednjo vsebino, če na kartici ni naloženih produktov:*

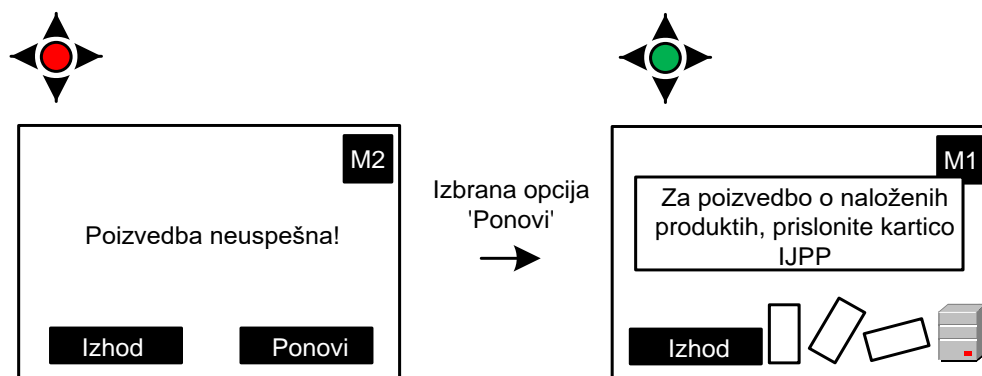
- Zvok za uspeh (glej definicijo)
- Kartica aktivna
- Status kartice
- Podatki imetnika kartice
  - Sporočilo: »Ni aktivnih produktov!«



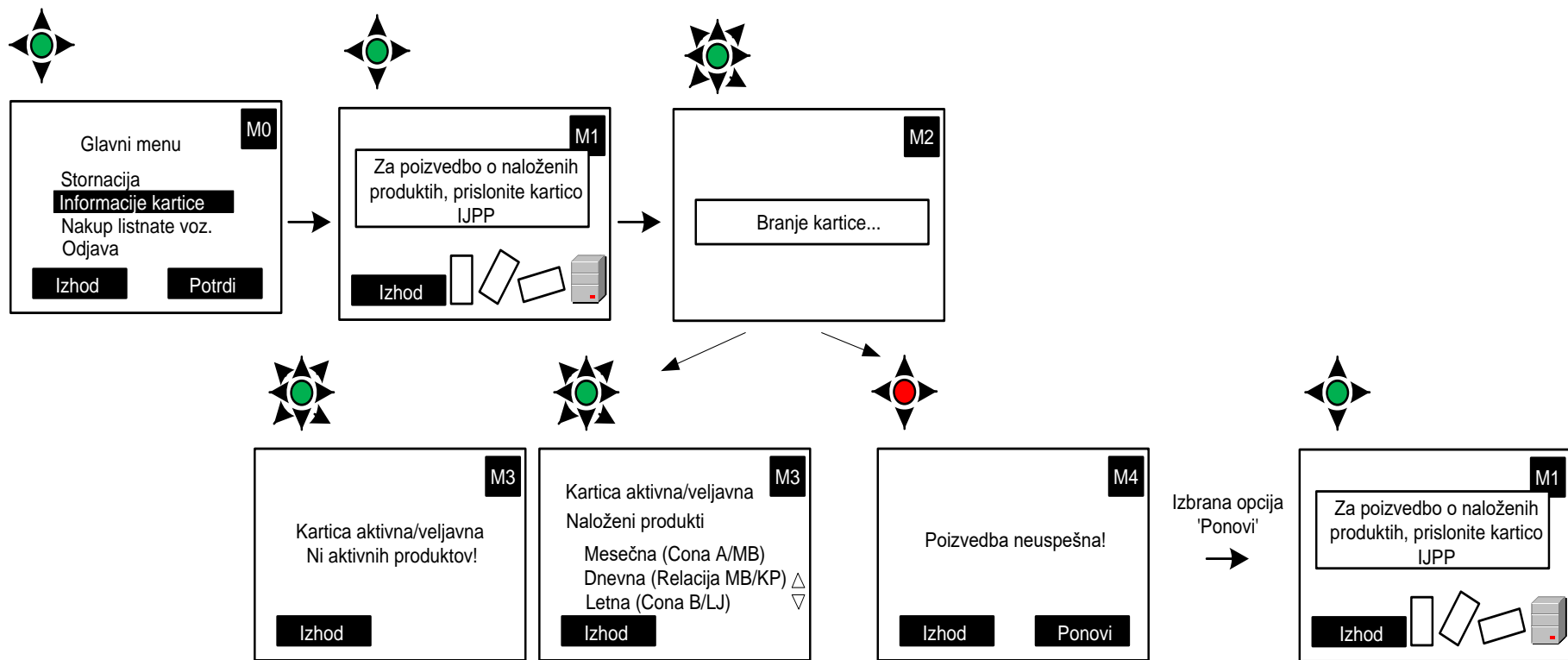
Vsebina se mora izpisovati cca. 5 sekund, oziroma do uporabnikovega ročnega izhoda z izbiro možnosti 'Izhod'. V vsem tem času mora biti svetlobni indikator vključen, in neprestano brez utripanja svetiti zeleno.

*V primeru neuspešnega branja:*

- Zvok za neuspeh (glej definicijo)
- Sporočilo: »Poizvedba neuspešna!«



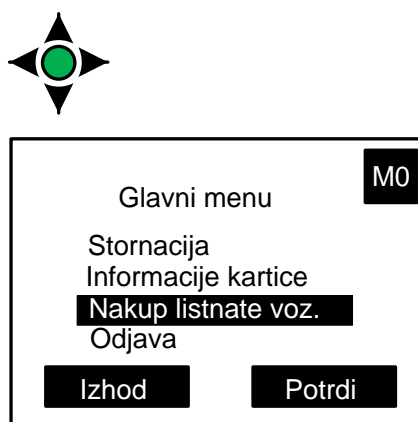
Vsebina se mora izpisovati cca. 5 sekund, oziroma do uporabnikovega ročnega izhoda z izbiro možnosti 'Izhod' oziroma vnovične ponovitve procedure Informacija kartice' s ponovno prislonitvijo. V času prikazovanja sporočila o neuspešni poizvedbi mora biti vključen svetlobni indikator, ki mora svetiti rdeče vse do uporabnikove izbire oz. timeout-a.



### 16.13.17 GUI – NAKUP LISTNE VOZOVNICE v TISKANI OBLIKI oz. KOT PRODUKT SHRANJEN NA IJPP KARTICO

Ko je terminal v stanju pripravljenosti interakcije z uporabnikom, se lahko na njemu izvede nakup listne vozovnice. Za listno vozovnico obstajata dve možnosti, kar pomeni da se jo lahko ob nakupu natisne (papirnata vozovnica), lahko pa se jo shrani tudi na IJPP kartico. Do možnosti za izvedbo nakupa listne vozovnice lahko uporabnik dostopa s pomočjo tipke 'Menu' če jo terminal pripomore, oziroma ekvivalentne korespondenčne povezave, ki je jasno označena in razumljiva uporabnikom.

Po pritisku tipke 'Menu' oziroma ekvivalentne povezave se uporabniku prikaže sledeča vsebina zaslona:



Uporabnik na zaslonu izbere možnost 'Nakup listne voz.' in jo potrdi s pomočjo dotika (zaslon na dotik) oziroma s pomočjo navigacijskih in potrditvenih korespondenčnih tipk. Po izbrani možnosti 'Nakup listne voz.' glavnega terminalskega menija, se uporabniku prikaže naslednji zaslon z naslednjo vsebino:

- Sporočilo: »Listna vozovnica«.
- Vsebina:
  - Število voženj: 1 (enkratna)
  - Relacija: npr. Kungota – Maribor
  - Ča in datum v obliki: hh:mm/dd.mm.yyyy
  - »Za plačilo: 5,00€« (krepko)
- Statusna ikona 'M5', ki ponazarja, da je mobilni terminal v stanju prikazovanja informacij o nakupu listne vozovnice
- Svetlobni indikator naj v tem primeru utripa s periodo 1000ms v zeleni barvi.



Listnata vozovnica

M5

Št. voženj: 1 (enkratna)  
Relacija: Kungota - Maribor  
Čas/Datum: 11:27/27.10.2015  
**Za plačilo: 5,00€**

Izhod

Potrdi

V naslednjem koraku se uporabniku ponudita dve možnosti, in sicer: možnost za tiskanje vozovnice in možnost za shranjevanje listne vozovnice na IJPP kartico:



Izberite možnost

M5

**Tiskanje vozovnice**  
Zapis na IJPP kartico

Izhod

Potrdi

Glede na izbiro in potrditev izbire se v naslednjem stanju uporabniku ponudi možnost za izbiro plačila. V primeru, ko je na razpolago zgolj ena možnost se ta korak preskoči:



Izberite način plačila

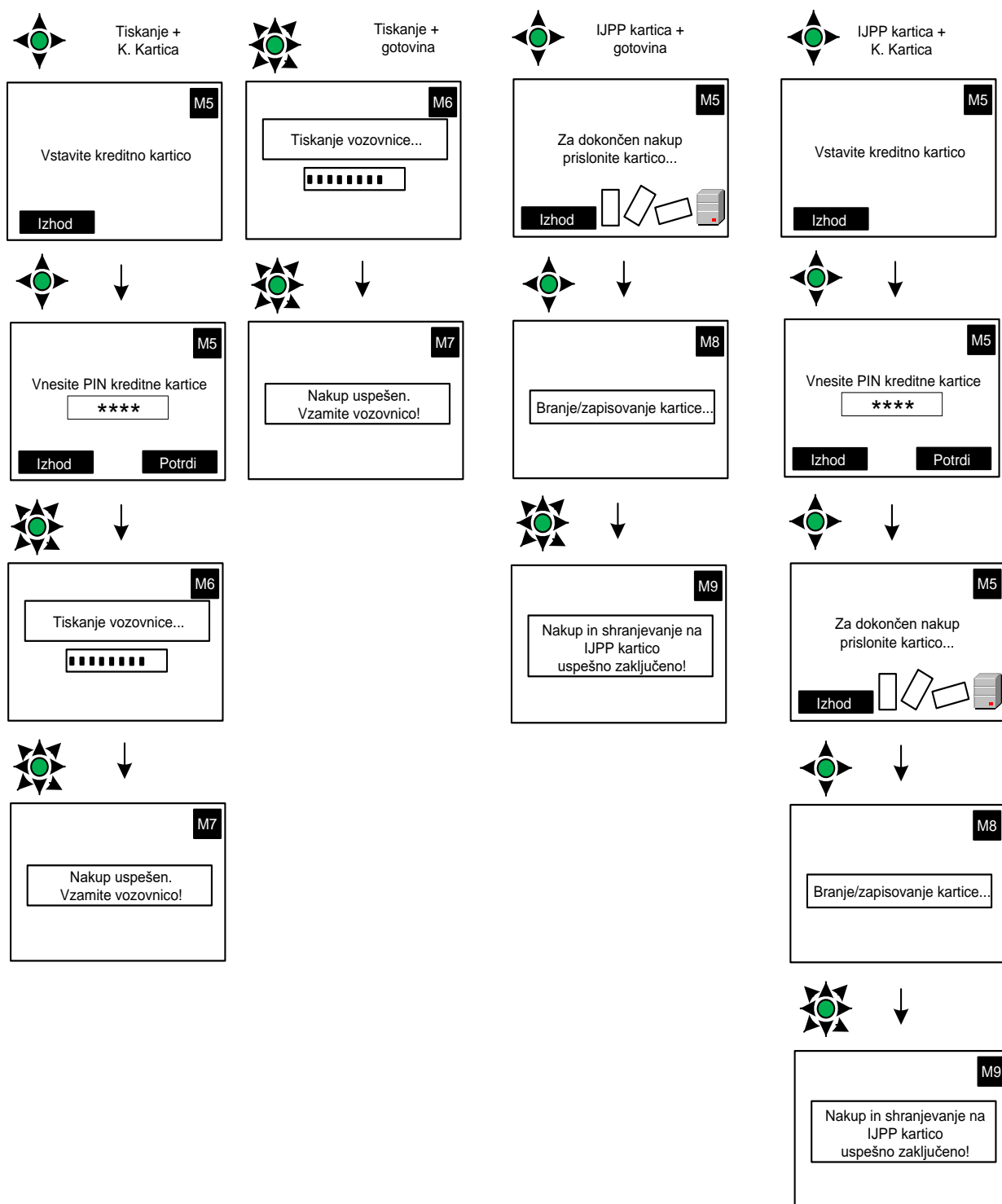
M5

**1. Gotovina**  
2. Kreditna kartica

Izhod

Potrdi

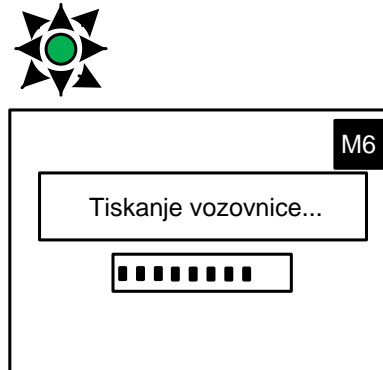
Glede na predhodno izbrani možnosti se uporabniku ponudi ustrezna nadaljnja procedura za nakup listne vozovnice:



Ko je plačilo izvedeno in je bila izbrana možnost za tiskanje vozovnice se na zaslonu pojavi naslednja vsebina:

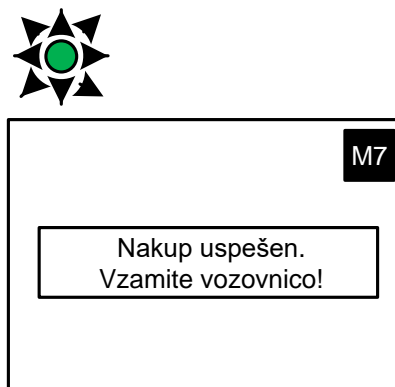
- Sporočilo: »Tiskanje vozovnice...«

- Indikator napredka, ki naj bo v sorazmerju deleža že natisnjenega dela potrdila.
- Statusna ikona 'M6', ki ponazarja, da je terminal v proceduri tiskanja.
- Led indikator naj v tem primeru sveti v zeleni barvi, brez utripanja.



Ko je vozovnica uspešno natisnjena, se mora najprej predvajati zvok za uspešno transakcijo, sočasno s tem pa se na zaslon enkrana izpiše pojavno sporočilo in naslednja vsebina:

- Pojavno sporočilo: »Nakup uspešen. Vzamite vozovnico!«
- Statusna ikona 'M7', ki ponazarja stanje, da je terminal uspešno natisnil listno vozovnico, in o tem obvešča uporabnika.
- Sporočilo naj se prikazuje 5 sekund, in v tem času naj bo svetlobni indikator ves čas vključen in sveti zeleno. Po preteku 5 sekund, se terminal vrne v začetno stanje, kjer čaka na novo akcijo uporabnika.



*Ko je plačilo izvedeno in je bila izbrana možnost za shranjevanje vozovnice na IJPP kartico, se na zaslonu pojavi naslednja vsebina:*

- Sporočilo: »Za dokončen nakup prislomite kartico IJPP«

Ob prislomitvi IJPP kartice se na zaslon izpiše naslednja vsebina:

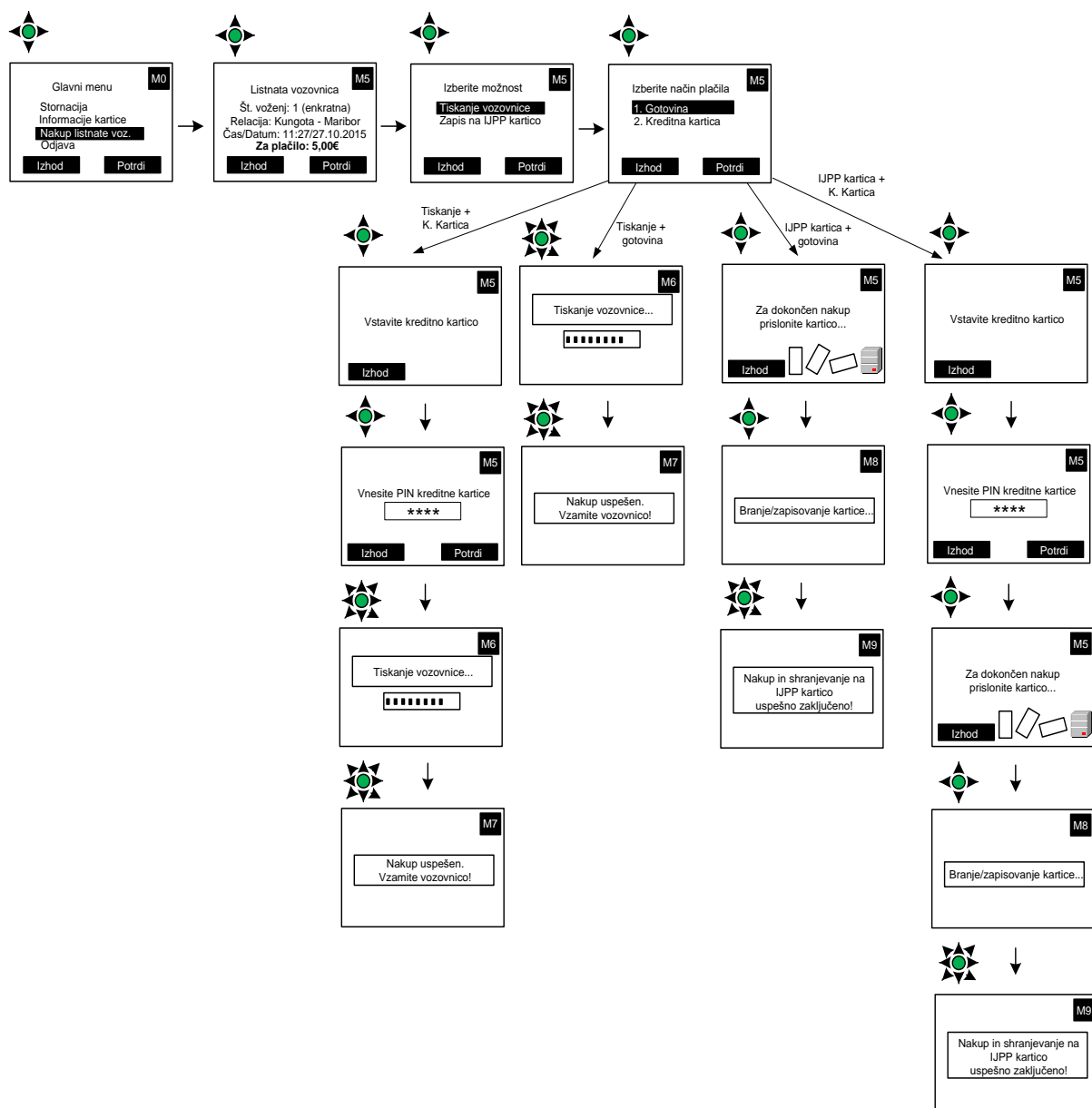


- Pojavno sporočilo z vsebino: »Branje/zapisovanje kartice...«
- Statusna ikona 'M8', ki ponazarja proceduro zapisovanja vozovnice na IJPP kartico.
- Led indikator naj v tem primeru utripa zeleno s periodo 500ms.

Po končani proceduri zapisovanja listne vozovnice na IJPP kartico se na zaslon izpiše naslednja vsebina:

- Sporočilo: »Nakup in shranjevanje na IJPP kartico uspešno zaključeno!«
- Statusna ikona 'M9', ki ponazarja, da je terminal uspešno shranil listno vozovnico na IJPP kartico.
- Led indikator naj v tem primeru sveti v zeleni barvi, brez utripanja.

Celotna procedura nakupa listne vozovnice:



Listna vozovnica naj vsebuje:

- Logotip IJPP
- Kontakni podatki in drugi podatki prevoznika
- Datum in čas prodaje vozovnice
- Id javnega prevoznega sredstva
- Id mobilnega terminala
- Referenčna številka
- Relacija vozovnice
- Znesek listne vozovnice
- ...

**Primer listnate  
vozovnice - relacija**

<b>LOGOTIP IJPP</b>	
Potniški promet XY d.o.o. Pod Rjavčevo 36, 2000 Maribor ID za DDV: SI12345678	
LISTNATA VOZOVNICA	
Datum/čas nakupa: 28.10.2015/14:57:57 ID Terminala: 77777	
ID prevoznega sredstva: 27-MB Ref. št.: 05122 77891 44531 77777	
Relacija: MB-LJ	
<b>Vrednost vozovnice:</b>	<b>10,85€</b>
Osnova: 10,00€      8.50 DDV: 0,85€	
<b>Znesek za plačilo:</b>	<b>10,85€</b>
Vozovnica velja kot račun, zato jo shranite	

**Primer listnate  
vozovnice - cona**

<b>LOGOTIP IJPP</b>	
Potniški promet XY d.o.o. Pod Rjavčevo 36, 2000 Maribor ID za DDV: SI12345678	
LISTNATA VOZOVNICA	
Datum/čas nakupa: 28.10.2015/14:57:57 ID Terminala: 77777	
ID prevoznega sredstva: 277-MB Ref. št.: 05122 77891 44531 77777	
Cona: Cona A/MB	
<b>Vrednost vozovnice:</b>	<b>3,10€</b>
Osnova: 2,86€      8.50 DDV: 0,24€	
<b>Znesek za plačilo:</b>	<b>3,10€</b>
Vozovnica velja kot račun, zato jo shranite	

### 16.13.18 POMEN STATUSNIH IKON

- S0 – mobilni terminal ne razpolaga s SIM kartico ali je ne prepozna
- P0 - mobilni terminal pričakuje vnos 4 mestne SIM PIN kode
- P1 – mobilni terminal pričakuje vnos 8 mestne SIM PUK kode, vendar samo v primeru, ko je vnos PIN kode v predhodnih poizkusih bil neuspešen
- P2 – SIM kartica je zaradi napačnih vnosov PIN kode in nato še PUK kode dokončno zaklenjena
- G0 – mobilni terminal izvaja postopek registracije v GPRS omrežje
- G1 – mobilni terminal javlja napake GPRS modema
- SS – mobilni terminal izvaja servisno storitev
- T0 – mobilni terminal posodablja tarifne razrede (tarifo)
- T1 – mobilni terminal usklajuje svoj sistemski čas s časom zalednega sistema
- B0 – mobilni terminal prenaša seznam kartic, ki so na črni listi
- B1 – mobilni terminal prenaša seznam vozovnic, ki so na črni listi

- V0 – mobilni terminal čaka na prijavo voznika
- U0 – mobilni terminal signalizira uspešno odjavo voznika
- V1 – mobilni terminal je v pripravljenosti izvesti validacijo, nakup, stornacijo
- V2 – mobilni terminal prikazuje informacije o neuspešni validaciji
- N0 – mobilni terminal prikazuje menu z možnimi opcijami za nakup produkta
- N1 – mobilni terminal je v zadnjem koraku pred dokončno izvršitvijo nakupa produkta
- N2 – mobilni terminal signalizira uspešno zaključen postopek nakupa produkta
- S7 – mobilni terminal prikazuje menu z možnimi produkti, ki jih je mogoče stornirati
- S8 – mobilni terminal signalizira uspešno izvedeno proceduro stornacije produkta ali skupine produktov
- M0 – mobilni terminal prikazuje glavni menu v katerem so možne izbire za stornacijo, poizvedbo – 'card info' in nakup listne vozovnice
- M1 – mobilni terminal je v stanju, v katerem se pričakuje prislonitev IJPP kartice za izvedbo 'card info' storitve
- M2 – mobilni terminal je v proceduri branja informacij o kartici
- M3 – mobilni terminal je uspešno zaključil poizvedbo 'card info' in prikazal ustrezne podatke
- M4 – mobilni terminal ni uspešno zaključil procedure branja informacij kartice, in prikazuje sporočilo o neuspehu.
- M5 – mobilni terminal prikazuje informacije o listni vozovnici pred nakupom
- M6 – mobilni terminal je v proceduri tiskanja listne vozovnice
- M7 – mobilni terminal je uspešno natisnil listno vozovnico, in prikazuje sporočilo o uspešnem tiskanju za čas petih sekund.
- M8 – mobilni terminal je v proceduri zapisovanja listne vozovnice na IJPP kartico.
- M9 – mobilni terminal je uspešno zapisal listno vozovnico na IJPP brezstično kartico.
- R5 – mobilni terminal je v proceduri avtomatskega tiskanja poročila ob odjavi voznika.
- R6 – mobilni terminal je uspešno natisnil poročilo v celoti (po odjavi voznika).
- R8 – tiskanje poročil na mobilnem terminalu se ni izvedlo uspešno.

### 16.13.19 SPOROČILA NAPAK

Koda napake	Pomen napake
0x00	Ni napake.
0x01	Nepravilna/napačna kartica (napačen format, napačni ključi, crypto,...)
0x02	Kartica je potekla.
0x03	Kartica še ni aktivirana.
0x04	Kartica je zaklenjena. Kartica se nahaja na črni listi.
0x05	Produkt ne obstaja (primer: poslan tarifni razred, ki ne obstaja v tarifnem sistemu).
0x06	Napačen tip produkta (primer: poslan service provider, ki ne obstaja). Preveriti je potrebno tarifnik, cono, pravi service provider... nato ponovi ukaz.
0x07	Podatkov o izbranem produktu ne najdemo oz. jih ne moremo prebrati (priporočena posodobitev tarifnega sistema, pošiljanje drugačnih parametrov).
0x08	Nedovoljen nakup mesečne ali letne karte, ki se prekriva na kakršenkoli način z enako karto. Možnost ponovitve le z drugimi podatki: mesec, leto.
0x09	Željen tarifni razred na kartici ni najden.

0x0A	Polna kartica. Nemogoč nakup produktov, ker je kartica polna.
0x0B	Cena produkta se razlikuje od cene, ki je v terminalovi tarifi.
0x0C	Produkt potekel. Željeni tarifni razred je še veljaven, vendar je zahtevan produkt za čas, ko več ne bo veljaven.
0x0D	Napaka pri branju podatkov iz kartice.
0x0E	Manjka podatek o položaju avtobusa (pri validaciji). V primeru, ko je več conski sistem in ko se avtobus vozi skozi več con je potrebno nujno imeti informacijo o položaju.
0x0F	Napaka z RTC. Prišlo je do desinhronizacije.
0x10	Nakup produktov ni dovoljen s prednastavljenim tarifnim razredom.
0x11	Validator je v fazi posadabljanja tarife. Validacija vozovnice ni mogoča.
0x12	Napaka v NFC komunikaciji.
0x13	Na kartici ni nobene vozovnice.
0x14	Pri nakupu produktov je bilo ugotovljeno, da je maksimalno število dnevnih ali tedenskih kart preseženo.
0x15	Terminal je zaznal, da je predolgo v offline načinu, čaka na povezavo s centrom.
0x16	Terminal ne vsebuje veljavnega tarifnega razreda (počakaj maksimalno 5 minut, da lahko terminal presname novo tarifo in nato ponovi. V primeru, da terminal ponovno pošlje enako kodo napake je potreben reset terminala).
0x17	Terminal ne vsebuje veljavne črne liste (počakaj maksimalno 5 minut, da lahko terminal presname novo black listo in nato ponovi. V primeru, da terminal ponovno pošlje enako kodo napake je potreben reset terminala).
0x18	Terminal je zaznal, da ga na strani centra čaka kritična posodobitev (počakaj maksimalno 20 minut, da se terminal posodobi in nato ponovi).
0x19	Terminal je zaseden ali se nahaja v 'offline' načinu (počakaj maksimalno 1 minuto, da konča z operacijo in nato ponovi).
0x1A	Terminal je zaznal, da je pomnilnik za transakcije poln.  (počakaj maksimalno 1 uro, da lahko terminal prenese transakcije na zaledni sistem in nato ponovi. V primeru, da terminal ponovno pošlje enako kodo napake je potreben reset terminala).
0x1B	Terminal je zaznal napačno nastavitev ure, potrebno je počakati na povezavo s centrom, da se posodobi ura.
0x1C	Terminal je predolgo časa v offline, zaradi tega mora narediti transakcijo s centrom (počakaj maksimalno 5 minut, da se lahko terminal poveže s centrom in nato ponovi. V primeru, da terminal ponovno pošlje enako kodo napake je potreben reset terminala).
0x1D	Ukaz ni bil uspešno izveden. Ponovi ukaz.
0x1E	Napaka modema.
0x1F	Napaka v GPRS povezavi (ang. communication error).
0x20	Napaka. SIM ni prisotna.
0x21	Napaka PIN.
0x22	Napaka PUK.
0x23	Napaka v avtentikaciji.
0x24	Že validirana vozovnica.
0x25	Nepoznan tip brezstične IJPP kartice.
0x26	Neustrezen produkt.
0x27	Neveljaven produkt.
0x28	Vozovnica na črni listi.
0x29	Vozovnica ni aktivna.

0x2A	Vozovnica je potekla.
0x2B	Napaka pri prenosu podatkov.
0x2C	Sinhronizacija neuspešna.
0x2D	Napaka. Sprejem samo vozovnic tipa 'X'
0x2E	Napaka. Storitev ni na voljo.

## 16.14 SISTEMSKA FUNKCIONALNOST (PRAVILA/DIAGRAMI)

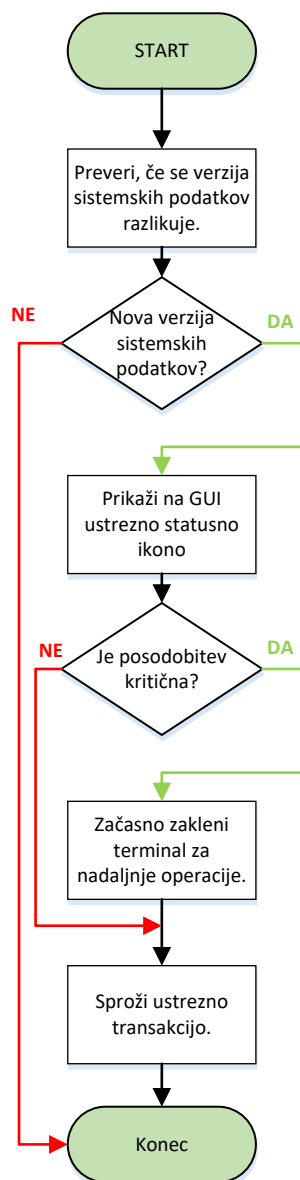
### 16.14.1 INICIALIZACIJA APLIKACIJE IN SISTEMSKIH PODATKOV

Pred prvo uporabo mora terminal iz zalednega sistema sprejeti najnovejšo različico podatkov, ki jih potrebuje za normalno delovanje. To so:

- Tarife,
- črne liste,
- izvršne liste,
- vozni redi,
- seznam pravila prehodov med conami.

Informacije o trenutnih različicah sistemskih podatkov na zalednem sistemu se terminalu sporočijo po vsaki uspešno opravljeni transakciji. Takoj po zagonu terminala, pa je smotno, da se opravi servisna transakcija. Po servisni oz. po vsaki uspešni transakciji preverimo, če se verzija sistemski podatkov na terminalu razlikuje od teh, ki je na zalednem sistemu. V primeru, da je potrebno sistemske podatke posodobiti, je potrebno opraviti ustrezno transakcijo. Pred pričetkom ustrezne transakcije se na zaslonu pojavi ustrezna statusna ikona. V primeru, da je postopek transakcije kritičen oz., da je potrebno preprečiti nadaljnjo uporabo terminala, ker bi sicer lahko prišlo do morebitne zlorabe sistema, se terminal zaklene. Nadaljnja uporaba terminala pa je omogočena šele po uspešni transakciji. Terminal zaklenemo v naslednjih primerih:

- Če se terminal daljši čas ni povezal na zaledni sistem, je nujno potrebno opraviti servisno transakcijo.
- Če je na zalednem sistemu pripravljena nova črna ali izvršna lista.
- Napačen format tarife, prazna tarifa oz. terminal še ni prejel ustrezne tarife.
- Če terminalu zmanjka prostora za shranjevanje kartičnih transakcij. Nujno je potrebno začeti pošiljati kartične transakcije na zaledni sistem, da se sprost prostor za nadaljnje shranjevanje transakcij.
- Ko terminal uspešno prejme programsko posodobitev in začne s postopkom posodabljanja programske opreme.
- Če NFC naprava na terminalu ne deluje pravilno.
- Pojavi se napaka na tiskalniku.
- Pojavi se napaka na bateriji.



SLIKA 16.1: POSTOPEK PRIČETKA USTREZNE TRANSAKCIJE

#### 16.14.2 KOMUNIKACIJA Z ZALEDNIM SISTEMOM

Način komunikacije z zalednim sistemom bo potekal s pomočjo spletnih storitev (web service).

#### 16.14.3 SERVISNA TRANSAKCIJA

Servisna transakcija bo izvedena z metodo **IdentifyTransactionPartameters**, ki jo uporabljamo za ugotavljanje trenutnih verzij sistemskih podatkov zalednega sistema, potrebnih za pravilno delovanje

terminalov v sistemu IJPP. V zalednem sistemu so verzije podatkov ločene z identifikatorji, kjer vsako verzijo sistemske podatkovne strukture določa unikaten identifikator.

Terminali s periodičnim klicanjem metode **IdentifyTransactionPartameters** pridobivajo identifikatorje zadnjih verzij tarife, črne liste, izvršne liste, vozni redov, seznama pravil prehodov med conami. Identifikatorji se hranijo na terminalih. Ob spremembi posamezne verzije določenih podatkov na zalednem sistemu, mora terminal izvesti prenos nove verzije ustreznih podatkov iz zalednega sistema.

Servisna transakcija se mora opraviti takoj po vklopu terminala in po vsaki drugi uspešni transakciji.

#### 16.14.4 TARIFE

Tarife so pomemben del sistema IJPP saj zajemajo informacije o vseh možnih vozovnicah in njihovih lastnostih.

Če je rezultat uspešne servisne transakcije ugotovitev, da se je identifikator za tarifo na zalednem sistemu posodobil, kar pomeni, da je na zalednem sistemu pripravljena nova tarifa, mora terminal opraviti prenos nove tarife.

Zahtevo za novo tarifo terminal opravi s pomočjo metode **GetTariff**.

#### 16.14.5 ČRNE LISTE

Črne liste zajemajo informacije o karticah oz. njihove UID številke, za katere je bila podana zahteva za blokado in posledično odstranitev iz sistema IJPP. Terminal mora za vsako kartico najprej preveriti, če se nahaja na črni listi, preden lahko nadaljuje s postopki komunikacije s kartico.

Če je rezultat uspešne servisne transakcije ugotovitev, da se je identifikator za črno listo na zalednem sistemu posodobil, kar pomeni, da je na zalednem sistemu pripravljena nova črna lista, mora terminal opraviti prenos nove črne liste.

Zahtevo za novo črno listo terminal opravi s pomočjo metode **GetBlackList**. Metoda deluje v dveh načinih: običajni način in inkrementalni način. Pri običajnem načinu izhodni podatki vsebujejo vse kartice na črni listi. Pri inkrementalnem načinu so v izhodnih podatkih kartice, ki so spremenile status, oziroma so bile na novo dodane ali odvzete iz črne liste. Pri inkrementalnem načinu je pomemben vhodni parameter **LastBlackListId**, ki pove od katere verzije črne liste naprej se upoštevajo dodane ali odvzete kartice na črni listi. Če parameter **LastBlackListId** ni prisoten (NULL), metoda deluje v običajnem načinu, sicer deluje metoda v inkrementalnem načinu.

#### 16.14.6 IZVRŠNE LISTE

Izvršne liste zajemajo informacije o karticah oz. njihove UID številke, za katere je bila podana zahteva za izvedbo določenega opravila: prenos vozovnice na kartico, brisanje vozovnice na kartici, posodobitev statusa ali specifične vozovnice na kartici ali začasno blokiranje kartice.

Če je rezultat uspešne servisne transakcije ugotovitev, da se je identifikator za izvršno listo na zalednem sistemu posodobil, kar pomeni, da je na zalednem sistemu pripravljena nova izvršna lista, mora terminal opraviti prenos nove izvršne liste.

Zahtevo za novo izvršno listo terminal opravi s pomočjo metode **GetExecutableList**.

#### 16.14.7 KARTIČNE TRANSAKCIJE

Informacije o kartičnih transakcijah se prenesejo v zaledni sistem s pomočjo metode **ProcessTransactions**, ki se uporabljamo za prenos uspešno izvedenih transakcij o nakupih, preverjanjih in uporabi vozovnic na terminalih do procesnega centra zalednega sistema. Te transakcije imenujemo IJPP Transakcije. Služijo nam za generiranje poročil za poravnavo. Zaradi tega je pomembno, da so te transakcije:

- Atomične – vsaka uspešno izvedena transakcija na terminalu mora biti zavedena kot uspešna tudi v zalednem sistemu. Transakcije izvedene na terminalih morajo biti prenesene v čim krajšem času na zaledni sistem, kjer so procesirane za izdelavo poročil na podlagi katerih izvajamo poravnavo.
- Konsistentne – transakcijski podatki morajo ustrezati pravilom za izvedbo transakcij tako, da vsaka transakcija izvedena na terminalih pravilno spremeni stanje podatkovne baze v zalednem sistemu.
- Izolirane – rezultati procesiranja transakcij morajo biti enaki, če transakcije procesiramo zaporedno ali hkrati v paketu po več skupaj.
- Trajne – rezultati procesiranih transakcij v podatkovni bazi so trajni – po končanem procesiranju se transakcije trajno hranijo v podatkovni bazi.

#### 16.14.8 SEZNAMI VOZNIH REDOV

Seznami voznih redov sestavljajo informacije o vseh voznih redih v sistemu IJPP, ki vsebujejo natančnejše podatke o linijah, časih prihodov in odhodov, režimih ter informacijah o posameznih postajah (koordinate, ime postaje, ID postaje, ID zone, ID mesta).

Če je rezultat uspešne servisne transakcije ugotovitev, da se je identifikator za seznam voznih redov na zalednem sistemu posodobil, kar pomeni, da so na zalednem sistemu pripravljeni novi vozni redi, mora terminal opraviti prenos novih voznih redov.

Zahtevo za sprejem novih voznih redov terminal opravi s pomočjo metode **GetLineDefinitions**.



#### 16.14.9 SEZNAM PRAVIL PREHODOV MED CONAMI

Seznam pravil prehodov med conami definira pravila prehodov med conami. Na podlagi teh informacij lahko znotraj posamezne cone določamo veljavnost conskih vozovnic.

Če je rezultat uspešne servisne transakcije ugotovitev, da se je identifikator za seznam pravil prehodov med conami na zalednem sistemu posodobil, kar pomeni, da je na zalednem sistemu pripravljen nov seznam pravil prehodov, mora terminal opraviti prenos novega seznama pravil prehodov med conami.

Zahtevo za nov seznam pravil prehodov med conami terminal opravi s pomočjo metode **GetTariffClassTransitionMatrix**.

#### 16.14.10 POSODOBITVE

Posodobitve služijo posodobitvi terminalske programske opreme.

#### 16.14.11 DOLOČITEV USTREZNOSTI CONSKE VOZOVNICE

Ustreznost mestne vozovnice lahko določimo s pomočjo seznama pravil prehodov med conami. Seznam pravil prehodov si lahko predstavljamo kot dvo-dimenzionalno polje, ki definira vse možne kombinacije med conami ter podaja število prestopov za posamezno kombinacijo. Ustrezno število prestopov za posamezno vozovnico tako torej določimo na podlagi vstopne cone vozovnice, ki se zapiše na vozovnico ob aktivaciji in podatka o coni trenutne postaje, kjer poteka validacija. Podatek o coni trenutne postaje dobimo s pomočjo informacije o lokaciji trenutne postaje in informacij o postajah, ki se nahajajo na seznamu voznih redov.

#### 16.14.12 DOLOČITEV USTREZNOSTI RELACIJSKE VOZOVNICE

Ustreznost medkrajevne vozovnice lahko določimo na podlagi informacije o lokaciji trenutne postaje, seznama postaj oz. relacije, ki je zapisana na kartici ter seznama voznih redov, ki je shranjen na čitalniku. Bistvo določitve ali je določena medkrajevna vozovnica veljavna na določeni liniji je, da ugotovimo, ali se trenutna postaja nahaja znotraj relacije, ki je za ustrezno vozovnico zapisana na kartici. Kar za ustrezno vozovnico na določeni liniji pomeni, da je začetek relacije na liniji definiran bodisi s postajo pred trenutno postajo bodisi s trenutno postajo. Konec relacije na liniji pa mora biti definiran s postajo za trenutno postajo. V kolikor želimo izpisati začetno in končno postajo oz. prestopno postajo na trenutni liniji, kjer je vozovnica še veljavna, jih je potrebno poiskati s primerjavo postaj relacije na kartici s postajami trenutne linije. Relacija na kartici je namreč lahko sestavljena iz ustrezne kombinacije različnih linij. V nadaljevanju bomo predstavili dva algoritma, ki se uporabljata za preverjanje enosmernih in dvosmernih relacijskih vozovnic.

#### 16.14.12.1 Algoritem za preverjanje dvosmerne relacijske vozovnice

Algoritem za preverjanje dvosmernih relacijskih vozovnic je neodvisen od numeričnih identifikatorjev oz. vrstnega reda številčenja posameznih postaj.

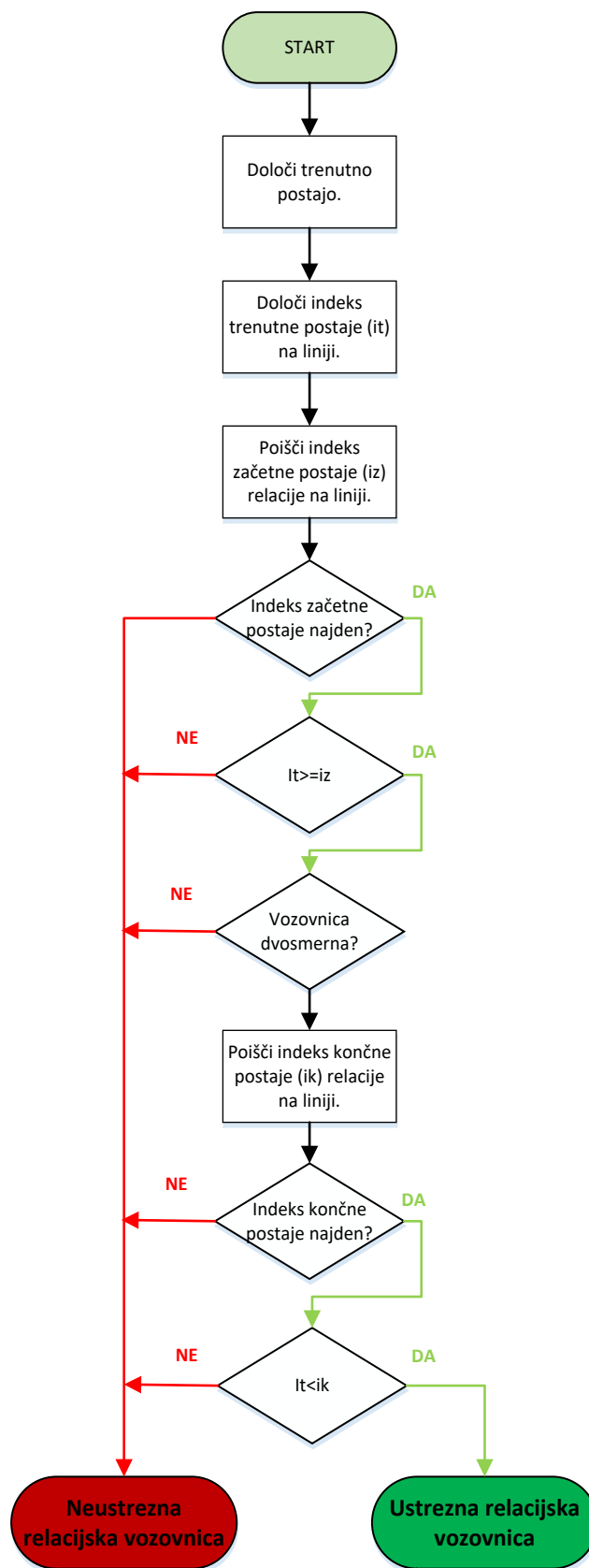
1. Določimo trenutno postajo na podlagi podatkov o trenutni lokaciji in informacij iz voznega reda.
2. Določimo zaporedno številko oz. indeks trenutne postaje ( $i_t$ ) v liniji na podlagi primerjave ID trenutne postaje z ID-ji vseh postaj na liniji.
3. Poiščemo indeks začetne postaje ( $i_z$ ) na liniji, ki odgovarja relaciji na kartici. Indeks poiščemo s ciklično primerjavo vseh posameznih ID postaj na liniji z vsemi ID postajami na kartici. Začnemo s prvo postajo na liniji ( $j=0$ ) in s prvo postajo na kartici ( $i=0$ ). Nato indeksa ustrezno povečujemo ter nadaljujemo s primerjavami tako, da vse postaje na kartici ( $i++$ ) primerjamo s posameznimi postajami na liniji ( $j++$ ).
4. Če se ID postaje na liniji ujema z ID postaje na kartici, zapišemo indeks postaje na liniji v indeks začetne postaje ( $i_z=j$ ). Če smo preiskali vse postaje, algoritem prekinemo.
5. Če velja, da je indeks trenutne postaje večji ali enak indeksu začetne postaje ( $i_t \geq i_z$ ), algoritem nadaljujemo, drugače, algoritem prekinemo, saj vozovnica trenutni postaji ne ustreza.
6. Če je vozovnica dvosmerna nadaljujemo z algoritmom. Če je vozovnica enosmerna, nadaljujemo z algoritmom za preverjanje enosmernih vozovnic.
7. Poiščemo indeks končne postaje ( $i_k$ ) na liniji, ki ustreza relaciji na kartici. Zadnjo postajo poiščemo s ciklično primerjavo vseh posameznih ID postaj na liniji z vsemi ID postajami na kartici. Začnemo z zadnjo postajo na liniji ( $j=\text{število postaj na liniji}$ ) in s prvo postajo na kartici ( $i=0$ ). Nato indeksa ustrezno povečujemo oz. zmanjšujemo ter nadaljujemo s primerjavami tako, da vse postaje na kartici ( $i++$ ) primerjamo s posameznimi postajami na liniji ( $j--$ ).
8. Če se ID postaje na liniji ujema z ID postaje na kartici, zapišemo indeks postaje na liniji v indeks končne postaje ( $i_k=j$ ) in algoritem nadaljujemo. Če smo preiskali vse postaje in nismo našli ujemanja postaj, algoritem prekinemo.
9. Če je indeks trenutne postaje manjši od indeksa končne postaje na liniji ( $i_t < i_k$ ), je kartica ustrezna, drugače kartica ni ustrezna in algoritem prekinemo.

#### 16.14.12.2 Algoritem za preverjanje enosmerne relacijske vozovnice

Algoritem za preverjanje enosmerne relacijskih vozovnic je neodvisen od numeričnih identifikatorjev oz. vrstnega reda številčenja posameznih postaj. Pomemben pa je vrstni red zapisanih postaj na kartici, ki odraža smer relacije. Algoritem za preverjanje enosmerne relacijske vozovnice se nadaljuje v sedmi točki algoritma za preverjanje dvosmerne relacijske vozovnice. Bistvena razlika med posameznima algoritmoma je v sedmi točki. Algoritem za dvosmerne vozovnice išče začetno in končno postajo ne

glede na vrstni red postaj zapisanih na kartici. Algoritem za enosmerne vozovnice pa pri iskanju končne postaje upošteva vrstni red postaj zapisanih na kartici.

1. Zapišemo indeks postaje na kartici v indeks začetne postaje ( $i_z=i$ ). indeks končne postaje ( $i_k$ ) na liniji, ki ustreza relaciji na kartici. Zadnjo postajo poiščemo s ciklično primerjavo vseh posameznih ID postaj na kartici z vsemi ID postajami na liniji. Začnemo z zadnjo postajo na kartici ( $i$ =število postaj na kartici) in s prvo postajo na liniji ( $j=0$ ). Nato indeksa ustrezno povečujemo oz. zmanjšujemo ter nadaljujemo s primerjavami tako, da vse postaje na liniji ( $j++$ ) primerjamo s posameznimi postajami na kartici ( $i--$ ,  $i>i_z$ ).
2. Če se ID postaje na liniji ujema z ID postaje na kartici, zapišemo indeks postaje na liniji v indeks končne postaje ( $i_k=j$ ) in algoritem nadaljujemo. Če smo preiskali vse postaje in nismo našli ujemanja postaj, algoritem prekinemo, saj vozovnica trenutni postaji ne ustreza.
3. Če je indeks trenutne postaje manjši od indeksa končne postaje ( $i_k < i_k$ ) na liniji, je kartica ustrezna, drugače kartica ni ustrezna in algoritem prekinemo.



SLIKA 16.2: ALGORITEM ZA DOLOČANJE USTREZNOSTI RELACIJSKE VOZOVNICE

## 16.15 PROTOKOLI KOMUNIKACIJE MED ČITALNIKOM IN KARTICO

NFC (near field communication) je tehnologija, ki omogoča brezstično komunikacijo med napravami, ki so v neposredni bližini. Prav zaradi nujno potrebne neposredne bližine med napravami je prisluškovanje izredno oteženo, kar ima očitne prednosti kar se tiče varstva podatkov. Naslednja prednost, ki je prav tako povezana z nujno neposredno bližino med posameznimi napravami je lastnost, da je možna komunikacija tudi med aktivno in pasivno napravo. Kar pomeni, da je lahko ena naprava brez lastnega napajanja in posledično za to poskrbi nasprotna naprava. Pasivna naprava se tako napaja s pomočjo induktivnosti in elektromagnetnega valovanja, ki služi za prenos komunikacije, kar je značilno za komunikacijo med čitalnikom, ki komunicira z brezstično pametno kartico. Za takšen primer je značilen način komunikacije način Način Branja/Pisanja. V načinu branja in pisanja lahko naprava oz. čitalnik prejema podatke od druge naprave ali pa ji podatke pošilja. Pri tem načinu je ključno, da naprava, ki vzpostavi iniciativo pri komunikaciji prevzame generiranje RF elektromagnetnega polja. Kar pomeni, da je druga naprava pasivna in za svoje delovanje ne potrebuje lastnega napajanja.

V sistemu IJPP se bodo čitalniki uporabljali način delovanja Branje/Pisanje. S tem bodo komunicirali z brezstičnimi pametnimi karticami, ki predstavljajo pasivne naprave. Pasivna komunikacija je natančneje definirana v drugem delu ISO/IEC 14443[1]:

- Pobudnik komunikacije oz. čitalnik generira elektromagnetno polje oz. 13.56 MHz nosilni signal. Elektromagnetno polje se uporablja za napajanje pasivne naprave.
- Prenos podatkov s strani čitalnika proti pasivni napravi oz. brezstični pametni kartici poteka z direktno amplitudno modulacijo (ASK) nosilnega signala.
- Prenos podatkov s strani brezstične pametne kartice v smeri čitalnika pa poteka s pomočjo modulacije bremena, kar se pozna na strani čitalnika v obliki padcev elektromagnetnega polja.

Preden pa lahko začnemo prenašati podatke, med kartico in terminalom pa je potrebno izvesti implementacijo komunikacijskih protokolov, ki so natančneje obravnavani v ISO/IEC 14443 [1]. V nadaljevanju bomo na kratko predstavili komunikacijske postopke oz. ukaze, ki jih je potrebno implementirati za uspešno inicializacijo kartice, ki je nepogrešljivi pri nadaljnji komunikaciji.

## 16.16 APLIKACIJSKI PROTOKOL KOMUNIKACIJE MED ČITALNIKOM IN KARTICO

V nadaljevanju se bomo osredotočili na aplikacijski protokol oz. posamezne postopke komunikacije med čitalnikom in kartico.

### 16.16.1 PRIJAVA/ODJAVA

Po vklopu terminala je za nadaljnjo uporabo potrebno opraviti prijavo z ustrezno identifikacijsko kartico. Ko identifikacijsko kartico približamo k terminalu, se podatki o kartici preberejo in posledično shranijo v podatkovno strukturo. Posledično lahko preverimo ustreznost prislone identifikacijske kartice in izvršimo prijavo. Odjavo pa lahko v terminalu izvedemo bodisi z izbiro ustrezne funkcije na terminalu bodisi s ponovno prislone identifikacijske kartice in s postopkom, ki je identičen postopku prijave.

#### 16.16.1.1 Protokol Prijave

Cilj postopka komunikacije pri prijavi je ustrezen protokol komunikacije med čitalnikom in kartico. Ustrezen protokol omogoča, da se iz ustrezne kartice preberejo in zapišejo v ustrezno podatkovno strukturo na terminalu ustrezne informacije o identifikacijski kartici. Postopek branja informacij o kartici je razdeljen v različna zaporedna stanja. V posameznih stanjih se izvajajo določeni postopki branja posameznih datotek na kartici.

##### 1. Stanje pripravljenosti

- **Akcije**

1. V stanju pripravljenosti se opravijo različni inicializacijski postopki podatkovne strukture, ki se uporablja pri procesu.
2. V strukturo se zapiše **Čas transakcije**
3. V strukturo se zapiše **UID kartice**
  1. UID se prebere iz kartice s postopkom izbire oz. antikolizije. Tako je UID pripravljen takoj, ko je kartica pripravljena za nadaljnja branja in pisanja podatkov.

Sledi operacija izbire ustrezne aplikacije (AID) s katero želi čitalnik komunicirati čitalnik pošlje ukaz:

- **Ukaz:** Select Application
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešna izbira aplikacije.

##### 2. Stanje izbiri aplikacijo

- **Akcije**

1. V stanju izberi aplikacijo čitalnik pošlje ukaz za avtentikacijo.
2. Postopek avtentikacije.

- **Ukaz:** Authenticate  
Aktivira se postopek avtentikacije med čitalnikom in kartico.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešna avtentikacija med čitalnikom in kartico.

### 3. Stanje avtentikacija

V stanju avtentikacija se po uspešno zaključeni avtentikaciji lahko prične postopek branja in pisanja v zelenih datotekah.

- **Akcije**
  1. Branje verzije kartice oz. datoteke INFO.
- **Ukaz:** Read Data  
Z ukazom aktiviramo postopek branja datoteke INFO na kartici, ki je namenjena informacijam o verziji kartice.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje datoteke INFO.

### 4. Stanje branje verzije (datoteka INFO)

- **Akcije:**
  1. V stanju branje verzije shranimo podatke o verziji kartice v ustrezno podatkovno strukturo.
  2. Branje osebnih podatkov oz. datoteke CARD INFO.
- **Pogoji:**

Primerjamo prebrane podatke o zapisani verziji na kartici z želeno verzijo kartice za nadaljnjo komunikacijo.

  1. Če verzija trenutne kartice ni ustrezna, prekinemo operacijo nakupa vozovnice in obvestimo uporabnika o nastali napaki.
  2. Če je verzija ustrezna nadaljujemo nakup vozovnice.
- **Ukaz:** Read Data  
Z ukazom aktiviramo postopek branja datoteke CARD INFO, ki vsebuje splošne informacije o kartici in osebne podatke o imetniku kartice.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje datoteke CARD INFO.

### 5. Stanje branje info (datoteka CARD INFO)

- **Akcije:**
  1. V tem stanju shranimo splošne informacije o kartici in imetniku kartice v ustrezno podatkovno strukturo.

## 2. Branje datoteke DATA INDEX.

- **Pogoji:**
  1. Preverimo veljavnost kartice. Če ni veljavna prekinemo operacijo nakupa in uporabnika obvestimo o poteku veljavnosti kartice.
- **Ukaz: Read Data**

Z ukazom aktiviramo postopek branja indeksa posameznih statusov oz datoteke DATA INDEX.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešno branje datoteke DATA INDEX.

## 6. Stanje branje indeksov (datoteka DATA INDEX)

- **Akcije:**
  1. V stanju shranimo informacije o mestu in dolžini zapisa podatkov, ki predstavljajo posamezni status, ki je zapisan v datoteki DATA INDEX na kartici.
  2. Branje datoteke DATA FILE.
- **Ukaz: Read Data**

Z ukazom aktiviramo branje posameznega statusa na kartici oz. datoteka DATA FILE. V primeru, da je na kartici več statusov, preberemo vse statuse.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešno branje datoteke DATA FILE.

## 7. Stanje branje statusa (datoteka DATA FILE)

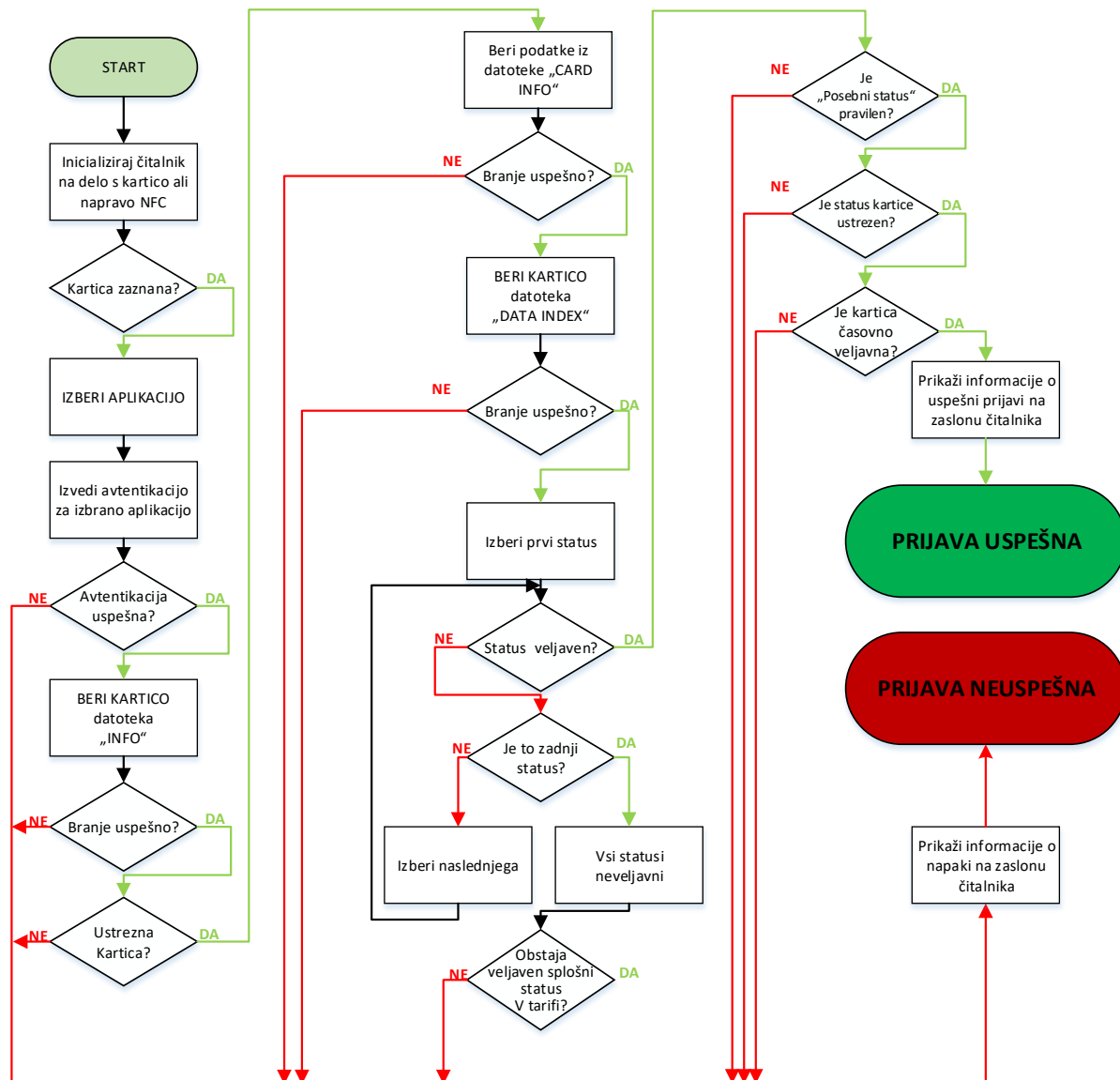
- **Akcije:**
  1. V stanju shranimo ustrezni status v podatkovno strukturo. V primeru, da je na kartici več statusov, je potrebno pregledati vse statuse na kartici in izbrati ustreznega.
- **Pogoji:**

V primeru, da so na kartici neveljavni statusi, nastavimo splošni status, ki ga preberemo iz tarifnika na čitalniku. Ob branju splošnega statusa iz tarifnika preverimo ustreznost tarifnika.

  1. Če tarifnik ni ustrezen prekinemo operacijo nakupa vozovnice in uporabnika obvestimo o napaki.
  2. Preverimo ustreznost statusa kartice. Če je neustrezen prekinemo postopek uporabnika obvestimo o napaki.
  3. Preverimo ustreznost parametra posebnega statusa kartice (vrsta kartice) v datoteki CARD INFO. Če je neustrezen prekinemo postopek in uporabnika obvestimo o napaki.



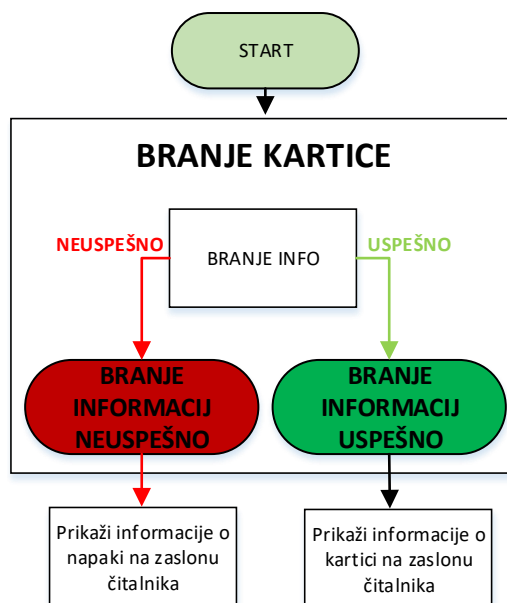
S stanjem **Stanje branje statusa (datoteka DATA FILE)** se postopek prijave konča, če kartica ustreza pogojem identifikacijske kartice, postopek končamo. Na zaslonu terminala prikažemo sporočilo, da je prijava uspešna.



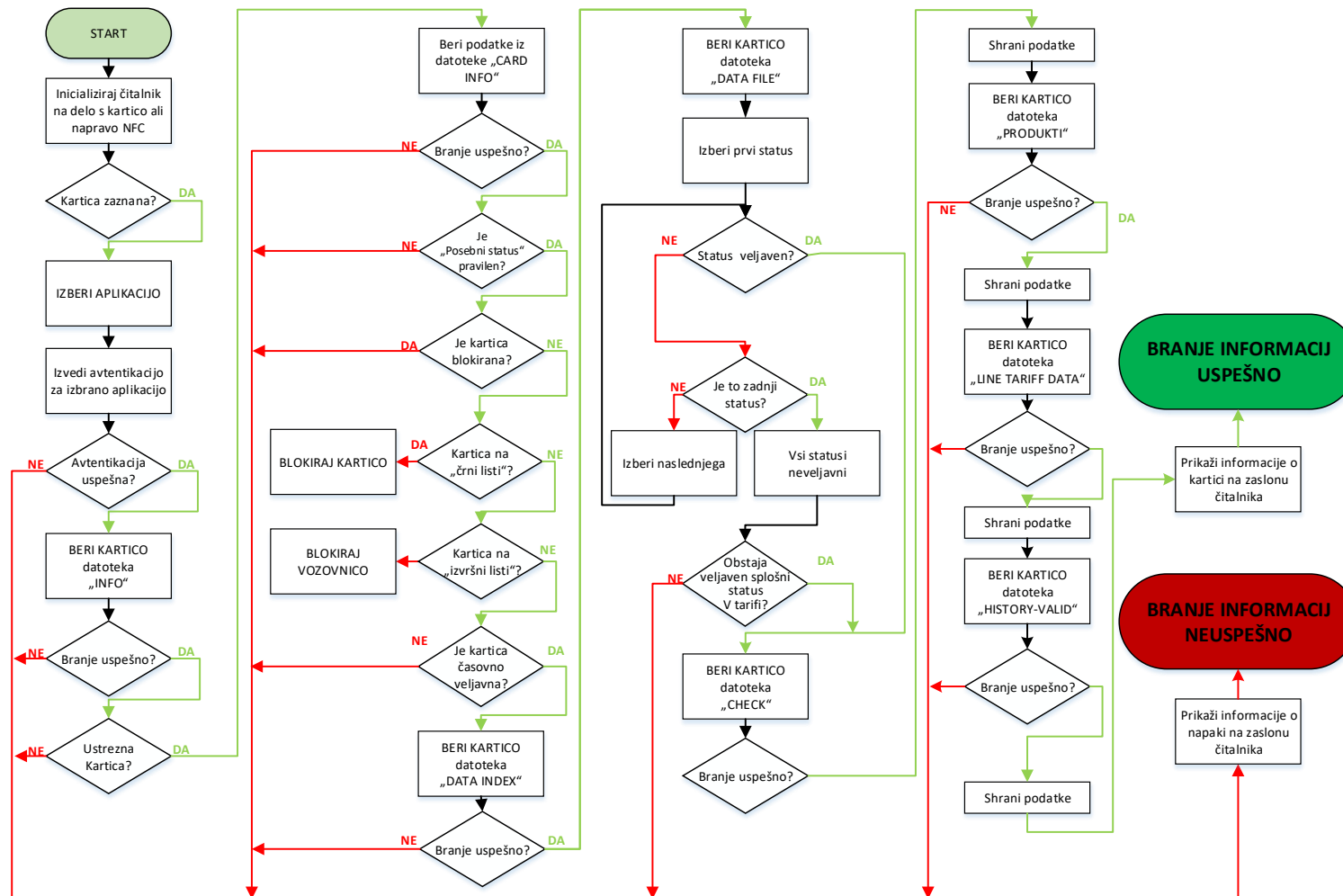
SLIKA 16.3: PRIJAVA

### 16.16.2 BRANJE INFO KARTICE

Na terminalu aktiviramo aplikacijo Branje Info in približamo kartico za katero želimo prikazati informacije k terminalu. Po uspešnem postopku se informacije o kartici in o vozovnicah na kartici shranijo v ustrezno podatkovno strukturo, ki jih nato lahko prikažemo na zaslonu. V nadaljevanju je natančneje predstavljen protokol komunikacije.



SLIKA 16.4: BRANJE INFO KARTICE – SPLOŠEN PREGLED



### 16.16.2.1 Protokol Branje INFO kartice

Cilj postopka komunikacije pri pregledovanju informacij o kartici in o obstoječih vozovnicah je ustrezen protokol komunikacije med čitalnikom in kartico. Ustrezen protokol omogoča, da se iz ustrezne kartice preberejo in zapišejo v ustrezno podatkovno strukturo na terminalu informacije o vseh obstoječih vozovnicah na kartici. Postopek branja informacij o kartici in o obstoječih vozovnicah je razdeljen v različna zaporedna stanja. V posameznih stanjih se izvajajo določeni postopki branja posameznih datotek na kartici. Ko se podatki uspešno prebrani in posledično shranjeni v podatkovni strukturi, lahko informacije prikažemo na zaslonu terminala.

#### 1. Stanje pripravljenosti

- **Akcije**

1. V stanju pripravljenosti se opravijo različni inicializacijski postopki podatkovne strukture, ki se uporablja pri procesu.
2. V strukturo se zapiše **Čas transakcije**
3. V strukturo se zapiše **UID kartice** - UID se prebere iz kartice s postopkom izbire oz. antikolizije. Tako je UID pripravljen takoj, ko je kartica pripravljena za nadaljnja branja in pisanja podatkov.

Sledi operacija izbire ustrezne aplikacije (AID) s katero želi čitalnik komunicirati čitalnik pošlje ukaz:

- **Ukaz:** Select Application
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešna izbira aplikacije.

#### 2. Stanje izbiri aplikacijo

- **Akcije**

1. V stanju izberi aplikacijo čitalnik pošlje ukaz za avtentikacijo.
2. Postopek avtentikacije.

- **Ukaz:** Authenticate  
Aktivira se postopek avtentikacije med čitalnikom in kartico.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešna avtentikacija med čitalnikom in kartico.

#### 3. Stanje avtentikacija

V stanju avtentikacija se po uspešno zaključeni avtentikaciji lahko prične postopek branja in pisanja v zelenih datotekah.

- **Akcije**

1. Branje verzije kartice oz. datoteke INFO.

- **Ukaz:** Read Data

Z ukazom aktiviramo postopek branja datoteke INFO na kartici, ki je namenjena informacijam o verziji kartice.

- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešno branje datoteke INFO.

#### 4. Stanje branje verzije (datoteka INFO)

- **Akcije:**

1. V stanju branje verzije shranimo podatke o verziji kartice v ustrezno podatkovno strukturo.
2. Branje osebnih podatkov oz. datoteke CARD INFO.

- **Pogoji:**

Primerjamo prebrane podatke o zapisani verziji na kartici z želeno verzijo kartice za nadaljnjo komunikacijo.

1. Če verzija trenutne kartice ni ustrezna, prekinemo operacijo nakupa vozovnice in obvestimo uporabnika o nastali napaki.
2. Če je verzija ustrezna nadaljujemo nakup vozovnice.

- **Ukaz: Read Data**

Z ukazom aktiviramo postopek branja datoteke CARD INFO, ki vsebuje splošne informacije o kartici in osebne podatke o imetniku kartice.

- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešno branje datoteke CARD INFO.

#### 5. Stanje branje info (datoteka CARD INFO)

- **Akcije:**

1. V tem stanju shranimo splošne informacije o kartici in imetniku kartice v ustrezno podatkovno strukturo.
2. Branje datoteke DATA INDEX.

- **Pogoji:**

1. Če je STATUS kartice blokirana ali ni aktivirana, vrnemo napako in prekinemo operacijo nakupa vozovnice.
2. Če se kartica nahaja na črni listi, sprožimo postopek blokade kartice.
3. Če se kartica nahaja na izvršni listi, sprožimo postopek blokade vozovnice.
4. Preverimo ustreznost parametra POSEBNI STATUS. Če ni ustrezen prekinemo operacijo nakupa vozovnice in uporabnika obvestimo o napaki.
5. Preverimo veljavnost kartice. Če ni veljavna prekinemo operacijo nakupa in uporabnika obvestimo o poteku veljavnosti kartice.

Če so zgornji pogoji ustrezni in se postopek nakupa vozovnice lahko nadaljuje, operacijo nadaljujemo z branjem indeksa statusov na kartici.

- **Ukaz: Read Data**

Z ukazom aktiviramo postopek branja indeksa posameznih statusov oz datoteke DATA INDEX.

- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje datoteke DATA INDEX.

## 6. Stanje branje indeksov (datoteka DATA INDEX)

- **Akcije:**
  1. V stanju shranimo informacije o mestu in dolžini zapisa podatkov, ki predstavljajo posamezni status, ki je zapisan v datoteki DATA INDEX na kartici.
  2. Branje datoteke DATA FILE.
- **Ukaz: Read Data**  
Z ukazom aktiviramo branje posameznega statusa na kartici oz. datoteka DATA FILE. V primeru, da je na kartici več statusov, preberemo vse statuse.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje datoteke DATA FILE.

## 7. Stanje branje statusa (datoteka DATA FILE)

- **Akcije:**
  1. V stanju shranimo ustrezni status v podatkovno strukturo. V primeru, da je na kartici več statusov, je potrebno pregledati vse statuse na kartici in izbrati ustreznega.
  2. Branje datoteke CHECK.
- **Pogoji:**

V primeru, da so na kartici neveljavni statusi, nastavimo splošni status, ki ga preberemo iz tarifnika na čitalniku. Ob branju splošnega statusa iz tarifnika preverimo ustreznost tarifnika.

  1. Če tarifnik ni ustrezen prekinemo operacijo nakupa vozovnice in uporabnika obvestimo o napaki.
- **Ukaz: Read Data**  
Z ukazom aktiviramo branje kontrolnih podatkov oz datoteke CHECK.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje datoteke CHECK.

## 8. Stanje branje kontrolnih podatkov (datoteka CHECK)

- **Akcije:**
  1. Shranimo kontrolne podatke v strukturo.
  2. Branje dela datoteke PRODUCTS, kjer so zapisani statusi posameznih vozovnic.
- **Ukaz: Read Value**

Z ukazom aktiviramo branje statusov obstoječih vozovnic na kartici.

- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje podatkov o statusih posameznih vozovnic.

#### 9. Stanje branje statusov vozovnic (datoteka PRODUCTS)

- **Akcije:**
  1. Shranimo statute posameznih vozovnic, ki so prisotne na kartici.
  2. Branje informacij o vozovnicah oz. preostanek datoteke PRODUCTS. Beremo le tisti del datoteke, ki ustreza veljavnim vozovnicam.
- **Ukaz: Read Data**  
Z ukazom aktiviramo branje vozovnic prisotnih na kartici.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje podatkov o vozovnicah v datoteki PRODUCTS.

#### 10. Stanje branje vozovnic (datoteka PRODUCTS)

- **Akcije:**
  1. Shranimo posamezne prebrane informacije o vozovnicah v podatkovno strukturo.
- **Pogoji:**
  1. Če se na kartici nahajajo relacijske vozovnice, nadaljujemo z branjem informacij o relacijah oz. datoteke LINE TARIFF DATA.
  2. Če se na kartici ne nahajajo relacijske vozovnice nadaljujemo z branjem zgodovine validacij oz. datoteke HISTORY-VALID.
- **Ukaz: Read Data**
  1. Z ukazom aktiviramo branje datoteke LINE TARIFF DATA (aktiviramo branje statusov relacij) ali datoteke HISTORY-VALID
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno branje podatkov.

#### 11. Stanje branje statusov relacij (datoteka LINE TARIFF DATA)

- **Akcije:**
  1. Shranimo statute posameznih relacij, ki so prisotne na kartici.
  2. Branje informacij o relacijah oz. preostanek datoteke LINE TARIFF DATA. Beremo le tisti del datoteke, ki ustreza veljavnim relacijam.
- **Ukaz: Read Data**  
Z ukazom aktiviramo branje ustreznih relacij prisotnih na kartici.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešno branje podatkov o relacijah v datoteki LINE TARIFF DATA.

## 12. Stanje branje relacij (datoteka LINE TARIFF DATA)

- **Akcije:**
  1. Shranimo posamezne prebrane informacije o relacijah v podatkovno strukturo.
- **Ukaz: Read Data**

Z ukazom aktiviramo branje zgodovine validacij oz. datoteke HISTORY-VALID.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešno branje podatkov o zgodovini validacij oz. datoteke HISTORY-VALID.

## 13. Stanje branje zgodovine validacij (datoteka HISTORY-VALID)

- **Akcije:**
  1. Shranimo informacije o zgodovini validacij v podatkovno strukturo.

S stanjem **Stanje branje zgodovine validacij (datoteka HISTORY-VALID)** se protokol BRANJE INFO KARTICE konča. Po uspešno izvedenem postopku branja prikažemo informacije o kartici in o vozovnicah na zaslonu.

### 16.16.3 VALIDACIJA IN AKTIVACIJA

Ko je na terminalu izbrana in se izvaja ustrezna aplikacija za validacijo oz. aktivacijo vozovnic, lahko kadarkoli k terminalu prislonimo kartico, ki jo želimo validirati oz. aktivirati, da se izvede postopek validacije oz. aktivacije. Po uspešno izvedenem postopku se informacije o validaciji izpišejo na zaslonu.

#### 16.16.3.1 Protokol VALIDACIJA in AKTIVACIJA

Cilj postopka validacije/aktivacije vozovnice je ustrezen protokol komunikacije med čitalnikom in kartico, ki omogoča, da se na kartici validira oz. aktivira obstoječa veljavna vozovnica. Postopek validacije/aktivacije vozovnice je razdeljen v različna zaporedna stanja. V posameznih stanjih se izvajajo določeni postopki branja ali pisanja v določene datoteke na kartici. Pred nadaljnjimi operacijami validacije ali aktivacije vozovnic se izvede postopek branje INFO (Slika 16.4). S tem s kartice preberemo vse potrebne informacije o kartici in o obstoječih vozovnicah na kartici. Postopek validacije ali aktivacije nadaljujemo v zadnjem stanju postopka branje INFO po sledečem protokolu.

#### 1. Stanje branje zgodovine validacij (datoteka HISTORY-VALID) Akcije:

- **Akcije:**



1. Shranimo zgodovino validacij v podatkovno strukturo.
  2. Poiščemo vozovnice, katerih veljavnost se je iztekla in sprožimo postopek brisanja oz. spremembe statusa teh vozovnic.
  3. Sprožimo postopek VALIDACIJA ali AKTIVACIJE.
- **Pogoji:**
    1. Če so na kartici neveljavne conske vozovnice sprožimo postopek spremembe STATUS-a (brisanje) vozovnic v neveljavne po naslednjem postopku:
      - a. **Stanje zapis statusov vozovnic (datoteka PRODUCTS)**  
Spremenimo status neveljavnih vozovnic v NEVELJAVNA.
      - b. **Stanje potrdi zapis podatkov**  
Potrdimo zapisane podatke ne vozovnico.
      - c. **Stanje branje vozovnic (datoteka PRODUCTS)**  
Ponovno preberemo vozovnice na kartici ter nadaljujemo postopek VALIDACIJE ali AKTIVACIJE.
    2. Če je na kartici neveljavna relacijska vozovnica pri kateri je potrebno pobrisati relacijo sprožimo postopek spremembe STATUS-a (brisanje) relacij v neveljavne po naslednjem postopku:
      - a. **Stanje zapis statusov relacij (datoteka PRODUCTS)**  
Spremenimo status neveljavnih relacijskih vozovnic v NEVELJAVNA.
      - b. **Stanje potrdi zapis podatkov**  
Potrdimo zapisane podatke ne vozovnico.
      - c. **Stanje branje vozovnic (datoteka PRODUCTS)**  
Ponovno preberemo vozovnice na kartici ter nadaljujemo postopek VALIDACIJE ali AKTIVACIJE.
    3. Če je na kartici ustrezna veljavna vozovnica gre za postopek VALIDACIJE. Postopek nadaljujemo v stanju:
      - a. **Stanje zapis validacije v zgodovino validacij (datoteka HISTORY-VALID).**
    4. Če je na kartici ustrezna veljavna vozovnica vendar še ni aktivirana gre za postopek AKTIVACIJE. Postopek nadaljujemo v stanju
      - a. **Stanje zapis statusov vozovnic (datoteka PRODUCTS).**
    5. Če na kartici ni ustrezne veljavne vozovnice, uporabnika obvestimo, da na kartici ni ustrezne vozovnice in prekinemo operacijo.
  - **Ukaz v primeru VALIDACIJE:** Write Record  
Z ukazom aktiviramo postopek zapisovanja podatkov validacije v ciklično datoteko vnosov HISTORY-VALID.
  - **Ukaz v primeru AKTIVACIJE:** Write Data  
Z ukazom aktiviramo postopek zapisa statusa vozovnice v datoteki PRODUCTS.
  - **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.

## 2. Stanje zapis statusov vozovnic (datoteka PRODUCTS)

V stanju aktiviramo ukaz za zapis podatkov vozovnice.

- **Ukaz: Write Data**  
Z ukazom aktiviramo zapis preostalih informacij o produktu oz. vozovnici, ki jo želimo aktivirati ali zapisati na kartico.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.

### 3. Stanje zapis vozovnic (datoteka PRODUCTS)

Če želimo aktivirati še kakšno vozovnico nadaljujemo z zapisovanjem statusa naslednje vozovnice v **stanju zapis statusov vozovnic (datoteka PRODUCTS)**. Ko smo aktivirali vse vozovnice nadaljujemo v **stanju zapis validacije v zgodovino validacij (datoteka HISTORY-VALID)**

- **Ukaz: Write Record**  
Z ukazom aktiviramo zapis podatkov validacije v zgodovino validacij v ciklično datoteko vnosov HISTORY-VALID.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.
- **Stanje zapis validacije v zgodovino validacij (datoteka HISTORY-VALID)**  
V stanju zapišemo informacije o validaciji v zgodovino validacij.
- **Ukaz: Write Data**  
Z ukazom aktiviramo zapis naključnega števila v datoteko CHECK.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.

### 4. Stanje zapis naključnega števila (datoteka CHECK)

Po uspešnem zapisu naključnega števila na kartico zapišemo še čas transakcije.

- **Ukaz: Write Data**  
Z ukazom aktiviramo postopek zapisa časa transakcije na kartico v datoteko CHECK.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.

### 5. Stanje zapis časa zadnje transakcije (datoteka CHECK)

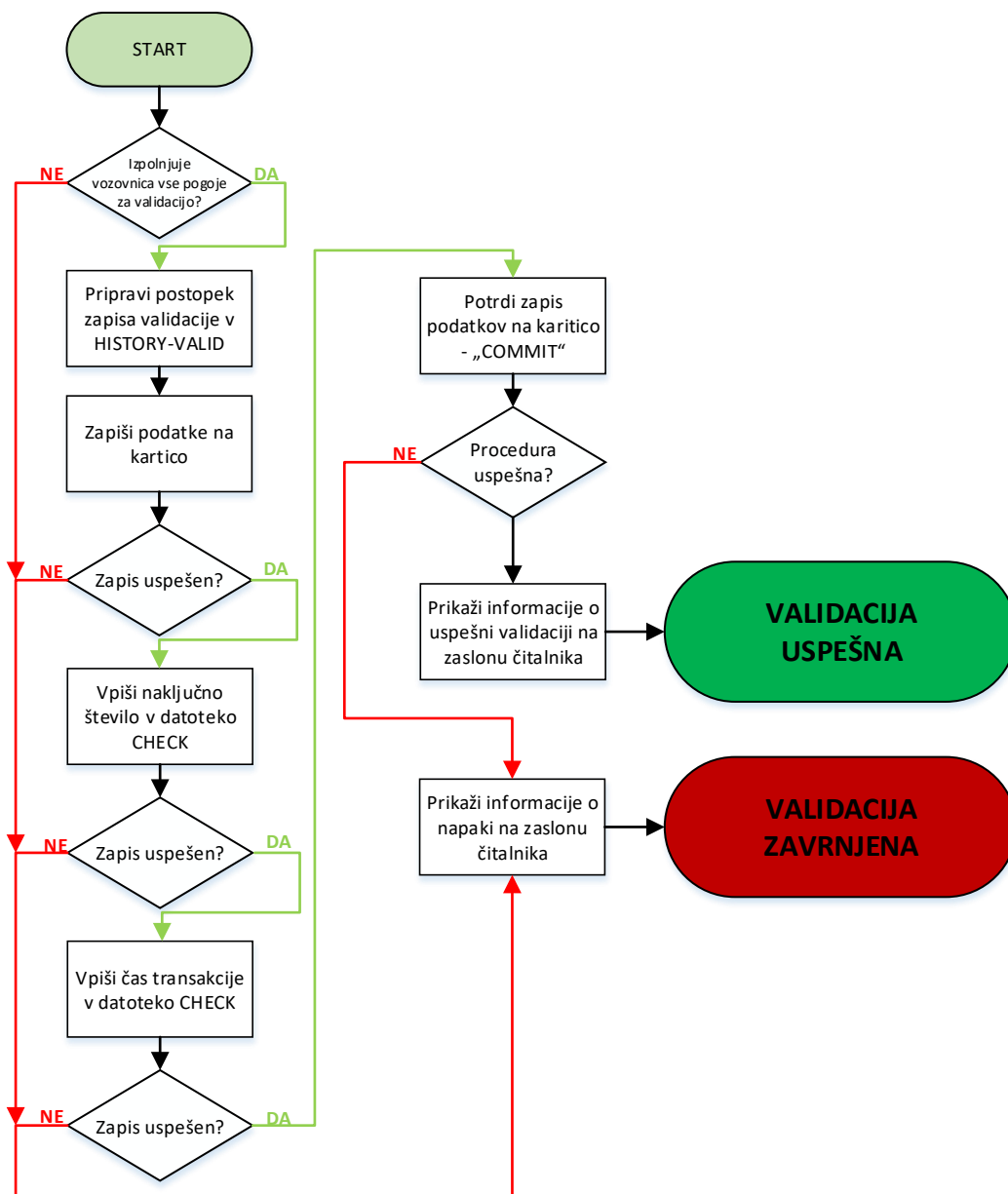
V stanju izvedemo ukaz za potrditev zapisanih podatkov na kartico.

- **Ukaz: Commit Transaction**  
Z ukazom aktiviramo postopek potrjevanja zapisanih podatkov na kartico.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno potrjevanje podatkov.

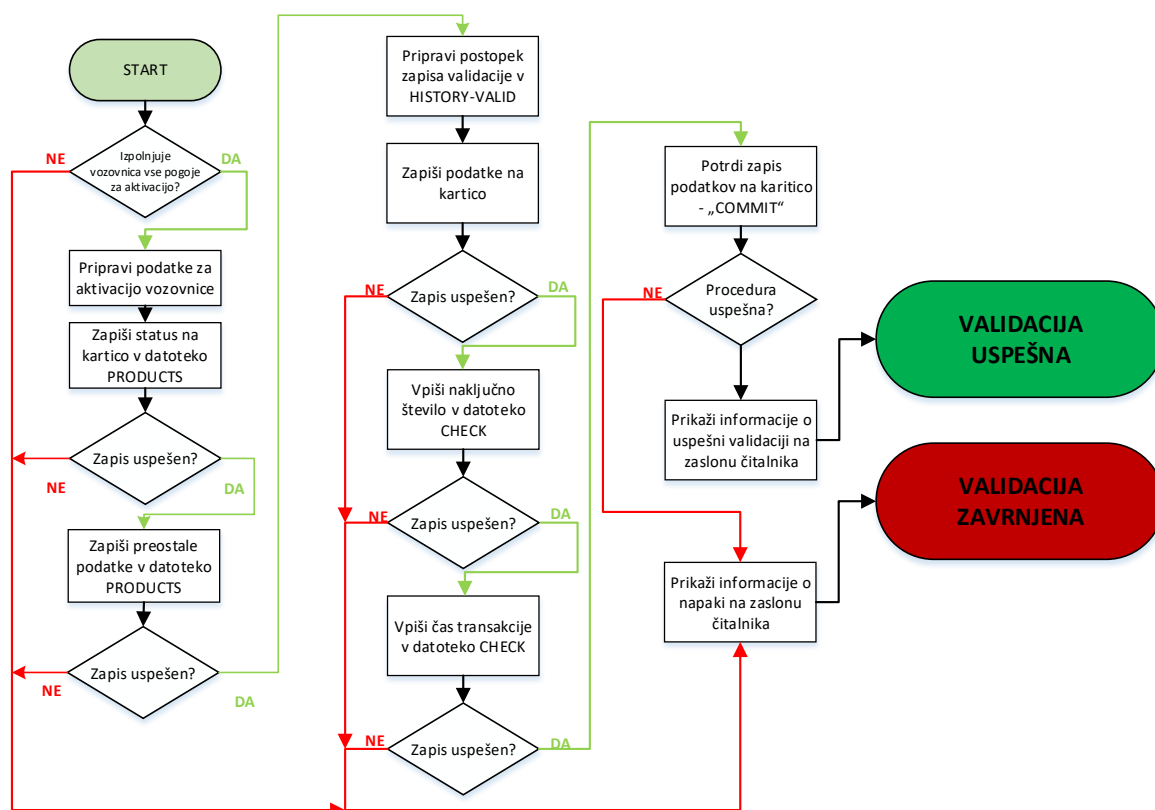
### 6. Stanje potrdi zapis podatkov

Po uspešnem potrjevanju podatkov na kartici uporabnika obvestimo o uspešni transakciji.





SLIKA 16.7: POSTOPEK VALIDACIJE



SLIKA 16.8: POSTOPEK AKTIVACIJE VOZOVNICE

#### 16.16.4 NAKUP VOZOVNICE

Ko je na terminalu izbrana in se izvaja ustrezna aplikacija za nakup vozovnic, lahko k terminalu prislomimo kartico, za katero želimo opraviti nakup vozovnice. Najprej se izvede branje INFO oz. ustreznih informacij o kartici na podlagi katerih se lahko na zaslonu prikažejo vsi ustrezni produkti, ki jih lahko za to kartico kupimo. Po izbiri ustreznega produkta, kartico ponovno prislomimo k terminalu. Po uspešno izvedenem nakupu se na zaslonu izpišejo informacije o nakupu vozovnice.

##### 16.16.4.1 Protokol NAKUP VOZOVNICE

Cilj postopka nakupa vozovnice je ustrezen protokol komunikacije med čitalnikom in kartico, ki omogoči, da se na ustrezno kartico zapiše izbrana vozovnica. Postopek nakupa je razdeljen v različna zaporedna stanja. V posameznih stanjih se izvajajo določeni postopki branja ali pisanja v določene datoteke na kartici. Pred nadaljnjimi operacijami nakupa vozovnice se izvede postopek branje INFO (Slika 16.4). S tem s kartice preberemo vse potrebne informacije o kartici in o obstoječih vozovnicah na kartici. Po izbiri zelenega produkta, postopek nakupa vozovnice nadaljujemo po sledečem protokolu.

### 1. Stanje branje zgodovine validacij (datoteka HISTORY-VALID) Akcije

- **Akcije:**
  1. Shranimo zgodovino validacij v podatkovno strukturo.
  2. Aktiviramo zapis statusa nove vozovnice.
- **Ukaz:** Write Data  
Z ukazom aktiviramo zapis statusa nove vozovnice, ki jo želimo zapisati datoteko PRODUCTS.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.

### 2. Stanje zapis statusa (datoteka PRODUCTS)

V stanju aktiviramo ukaz za zapis podatkov vozovnice.

- **Ukaz:** Write Data  
Z ukazom aktiviramo zapis preostalih informacij o produktu oz. vozovnici, ki jo želimo zapisati v datoteko PRODUCTS.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.

### 3. Stanje zapis podatkov vozovnice (datoteka PRODUCTS)

Če želimo zapisati na kartico še kakšno vozovnico nadaljujemo z zapisovanjem statusa naslednje vozovnice in nadaljujemo v stanju zapis statusa. V primeru, da smo na kartico uspešno zapisali vse vozovnice pa nadaljujemo z zapisovanjem naključnega števila.

- **Ukaz:** Write Data  
Z ukazom aktiviramo zapis naključnega števila v podatkovno datoteko CHECK.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.

### 4. Stanje zapis naključnega števila (datoteka CHECK)

Po uspešnem zapisu naključnega števila na kartico zapišemo še čas transakcije.

- **Ukaz:** Write Data  
Z ukazom aktiviramo postopek zapisa časa transakcije v datoteko CHECK.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.

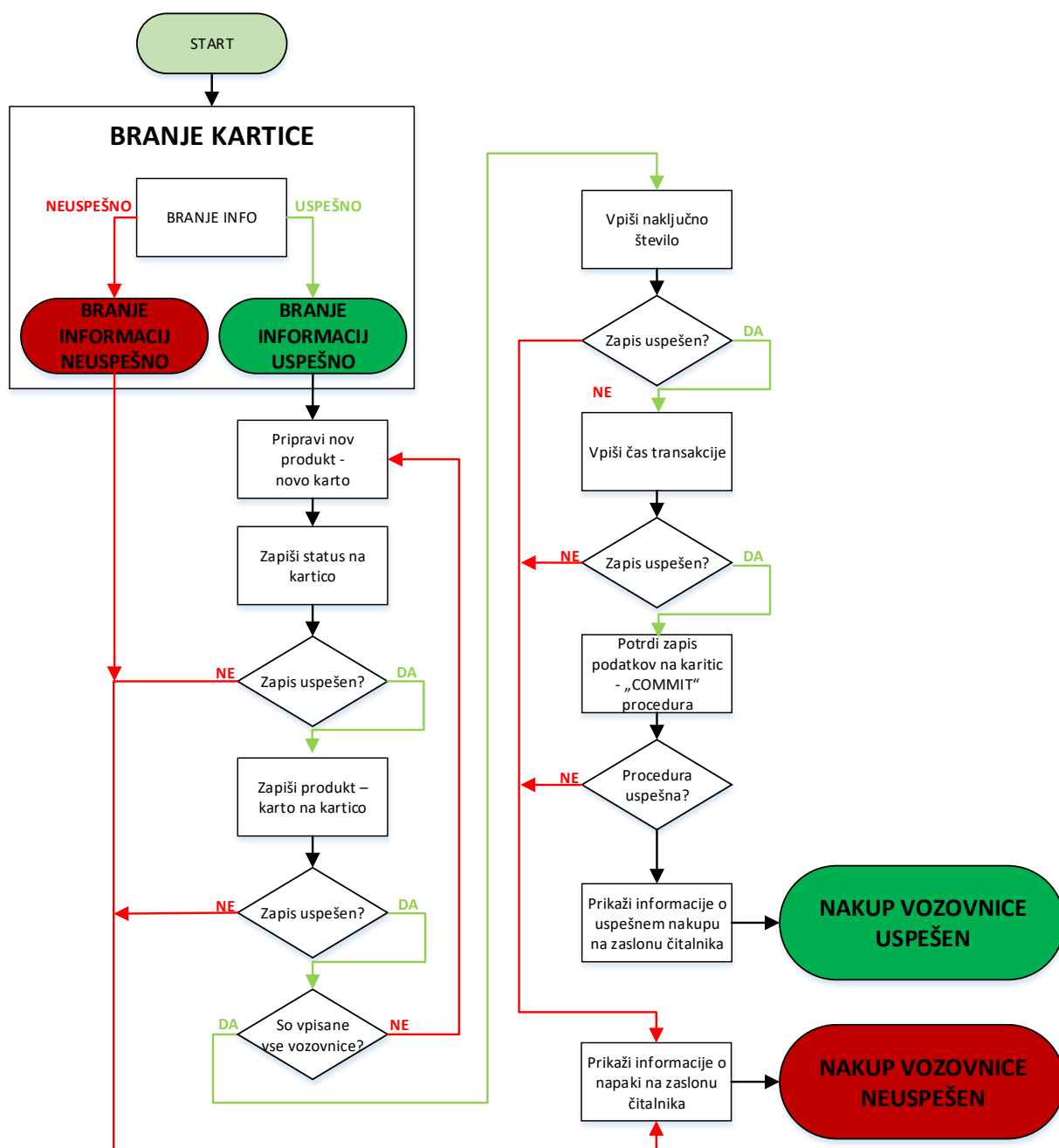
### 5. Stanje zapis časa zadnje transakcije (datoteka CHECK)

V stanju izvedemo ukaz za potrditev zapisanih podatkov na kartico.

- **Ukaz:** Commit Transaction  
Z ukazom aktiviramo postopek potrjevanja zapisanih podatkov.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno potrjevanje podatkov.

### 6. Stanje potrdi zapis podatkov

Po uspešnem potrjevanju podatkov na kartici uporabnika obvestimo o uspešni transakciji.



SLIKA 16.9: NAKUP VOZOVNICE

#### 16.16.5 STORNO NAKUPA VOZOVNICE

Ko je na terminalu izbrana in se izvaja ustrezna aplikacija za storno nakup vozovnice, lahko k terminalu prislonimo kartico, za katero želimo opraviti storno nakup vozovnice. Najprej se izvede branje INFO oz. ustreznih informacij o kartici na podlagi katerih se lahko na zaslonu prikažejo vsi ustrezni produkti, ki jih za to kartico lahko storniramo. Po branju informacij terminal preveri naključno število zapisano na

kartici in ga primerja z naključnim številom, ki se je ustvarilo pri zadnji transakciji na terminalu. S tem lahko določimo, če se vozovnice na kartici lahko stornirajo. Storniramo namreč lahko vozovnice, ki so bile predhodno kupljene na istem terminalu. Po izbiri ustreznega produkta, kartico ponovno prislonimo k terminalu. Po uspešno izvedenem nakupu se na zaslonu izpišejo informacije o storno postopku. Imetniku kartice pa se povrnejo sredstva v obliki gotovine. Predpogoj za storno nakupa vozovnice je uspešen predhodni nakup vozovnice na istem čitalniku.

#### **16.16.5.1 Protokol STORNO NAKUPA VOZOVNICE**

Cilj postopka storno nakupa vozovnice je ustrezen protokol komunikacije med čitalnikom in kartico, ki omogoči, da se zadnji nakup na določenem čitalniku stornira. Postopek storno je razdeljen v različna zaporedna stanja. V posameznih stanjih se izvajajo določeni postopki branja in pisanja določenih datotek na kartici. Pred nadaljnjimi operacijami storno nakupa vozovnice se izvede postopek branje INFO (Slika 16.4). S tem s kartice preberemo vse potrebne informacije o kartici in o obstoječih vozovnicah na kartici. Postopek storno nakupa vozovnice nadaljujemo po izbiri produkta, ki ga želimo stornirati, po sledečem protokolu.

##### **1. Stanje branje zgodovine validacij (datoteka HISTORY-VALID) Akcije**

- **Akcije:**
  1. Shranimo zgodovino validacij v podatkovno strukturo.
  2. Aktiviramo pisanje statusa NEVELJAVNA izbrane vozovnice.
- **Ukaz: Write Data**  
Z ukazom aktiviramo pisanje statusa vozovnice v datoteko PRODUCTS.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno pisanje podatkov.

##### **2. Stanje zapis statusa (datoteka PRODUCTS)**

V stanju aktiviramo ukaz za zapis naključnega števila.

- **Ukaz: Write Data**  
Z ukazom aktiviramo zapis naključnega števila v datoteko CHECK.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.

##### **3. Stanje zapis naključnega števila (datoteka CHECK)**

Po uspešnem zapisu naključnega števila na kartico zapišemo še čas transakcije.

- **Ukaz: Write Data**  
Z ukazom aktiviramo postopek zapisa časa transakcije v datoteko CHECK.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešen zapis podatkov na kartico.

##### **4. Stanje zapis časa zadnje transakcije (datoteka CHECK)**

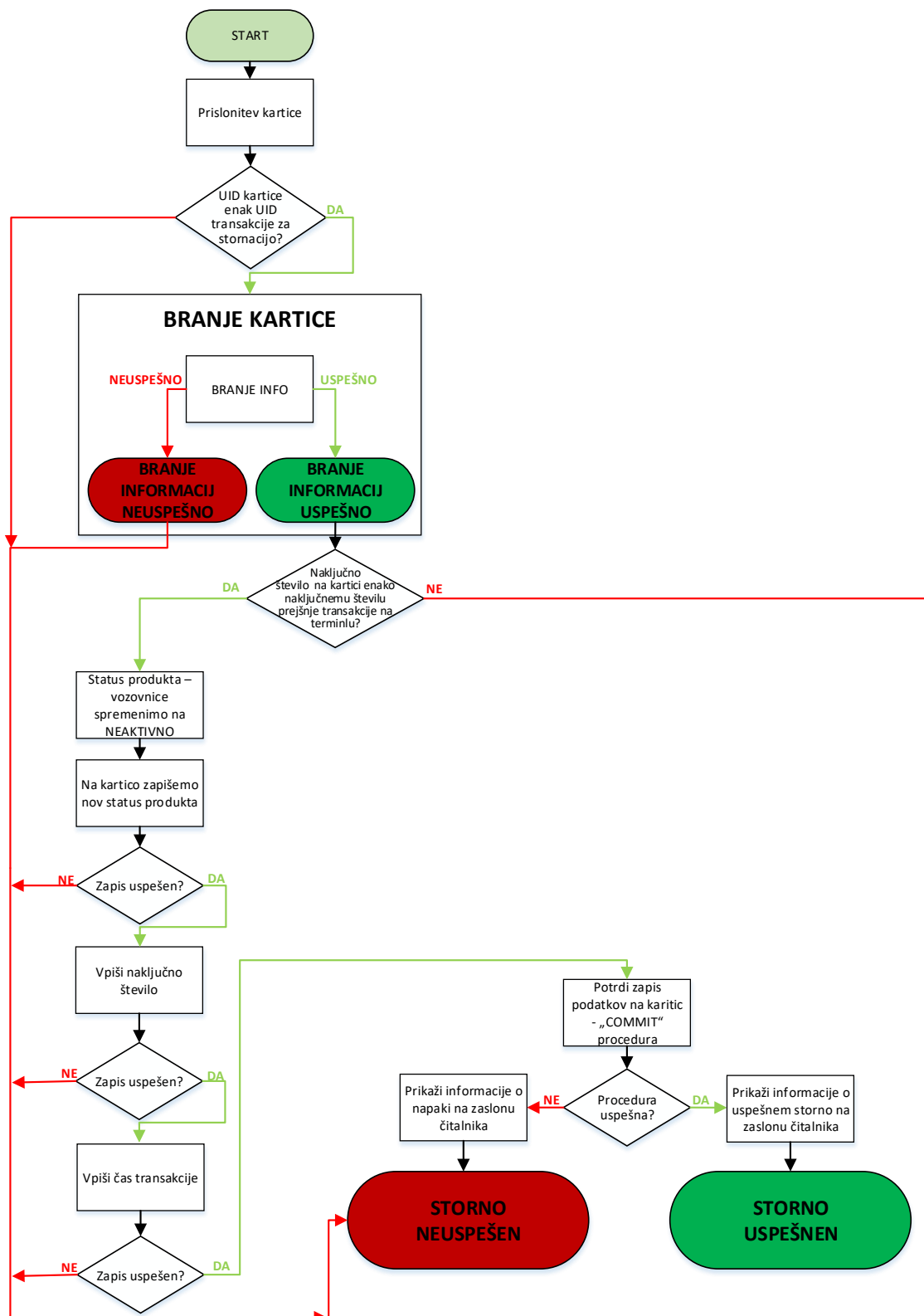
V stanju izvedemo ukaz za potrditev zapisanih podatkov na kartico.



- **Ukaz:** Commit Transaction  
Z ukazom aktiviramo postopek potrjevanja zapisanih podatkov na kartico.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno potrjevanje podatkov.

#### **5. Stanje potrdi zapis podatkov**

Po uspešnem potrjevanju podatkov na kartici uporabnika obvestimo o uspešni storno transakciji.



SLIKA 16.10: STORNO NAKUPA VOZOVNICE

## 16.16.6 BLOKADA APLIKACIJE IJPP

Blokada aplikacije se izvede samodejno pri postopku branja info kartice v primeru, da se UID kartice nahaja na črni listi.

### 16.16.6.1 Protokol BLOKADA APLIKACIJE IJPP

Cilj postopka blokade aplikacije IJPP je ustrezen protokol komunikacije med čitalnikom in kartico, ki omogoči, da se STATUS aplikacije IJPP nastavi na vrednost BLOKIRANA. Postopek blokade aplikacije IJPP je razdeljen v različna zaporedna stanja. V posameznih stanjih se izvajajo določeni postopki branja ali pisanja v določene datoteke na kartici

#### 1. Stanje pripravljenosti

- **Akcije**

1. V stanju pripravljenosti se opravijo različni inicializacijski postopki podatkovne strukture, ki se uporablja pri procesu.
2. V strukturo se zapiše **Čas transakcije**
3. V strukturo se zapiše **UID kartice**  
UID se prebere iz kartice s postopkom izbire oz. antikolizije. Tako je UID pripravljen takoj, ko je kartica pripravljena za nadaljnja branja in pisanja podatkov.

Sledi operacija izbire ustrezne aplikacije (AID) s katero želi čitalnik komunicirati čitalnik pošlje ukaz:

- **Ukaz:** Select Application
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešna izbira aplikacije.

#### 2. Stanje izbiri aplikacijo

- **Akcije**

1. V stanju izberi aplikacijo čitalnik pošlje ukaz za avtentikacijo.
2. Postopek avtentikacije.

- **Ukaz:** Authenticate  
Aktivira se postopek avtentikacije med čitalnikom in kartico.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešna avtentikacija med čitalnikom in kartico.

#### 3. Stanje avtentikacija

V stanju avtentikacija se po uspešno zaključeni avtentikaciji lahko prične postopek branja in pisanja v zelenih datotekah.

- **Akcije**

1. Branje verzije kartice oz. datoteke INFO.

- **Ukaz:** Read Data

Z ukazom aktiviramo postopek branja datoteke INFO na kartici, ki je namenjena informacijam o verziji kartice.

- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešno branje datoteke INFO.

**4. Stanje branje verzije (datoteka INFO)**

- **Akcije:**

1. V stanju branje verzije shranimo podatke o verziji kartice v ustrezno podatkovno strukturo.
2. Branje osebnih podatkov oz. datoteke CARD INFO.

- **Pogoji:**

Primerjamo prebrane podatke o zapisani verziji na kartici z želeno verzijo kartice za nadaljnjo komunikacijo.

1. Če verzija trenutne kartice ni ustrezna, prekinemo operacijo nakupa vozovnice in obvestimo uporabnika o nastali napaki.
2. Če je verzija ustrezna nadaljujemo nakup vozovnice.

- **Ukaz:** Read Data

Z ukazom aktiviramo postopek branja datoteke CARD INFO, ki vsebuje splošne informacije o kartici in osebne podatke o imetniku kartice.

- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešno branje datoteke CARD INFO.

**5. Stanje branje info (datoteka CARD INFO)**

- **Akcije:**

1. V tem stanju shranimo splošne informacije o kartici in imetniku kartice v ustrezno podatkovno strukturo.

- **Pogoji:**

1. Če je STATUS kartice blokirana ali ni aktivirana vrnemo napako in prekinemo operacijo nakupa vozovnice.
2. Preverimo, če se kartica nahaja na črni listi sprožimo postopek blokade kartice.
3. Preverimo ustreznost parametra POSEBNI STATUS. Če ni ustrezen prekinemo operacijo nakupa vozovnice in uporabnika obvestimo o napaki.
4. Preverimo veljavnost kartice. Če ni veljavna prekinemo operacijo nakupa in uporabnika obvestimo o poteku veljavnosti kartice.

Če je kartica na črni listi se nadaljuje postopek blokade kartice.

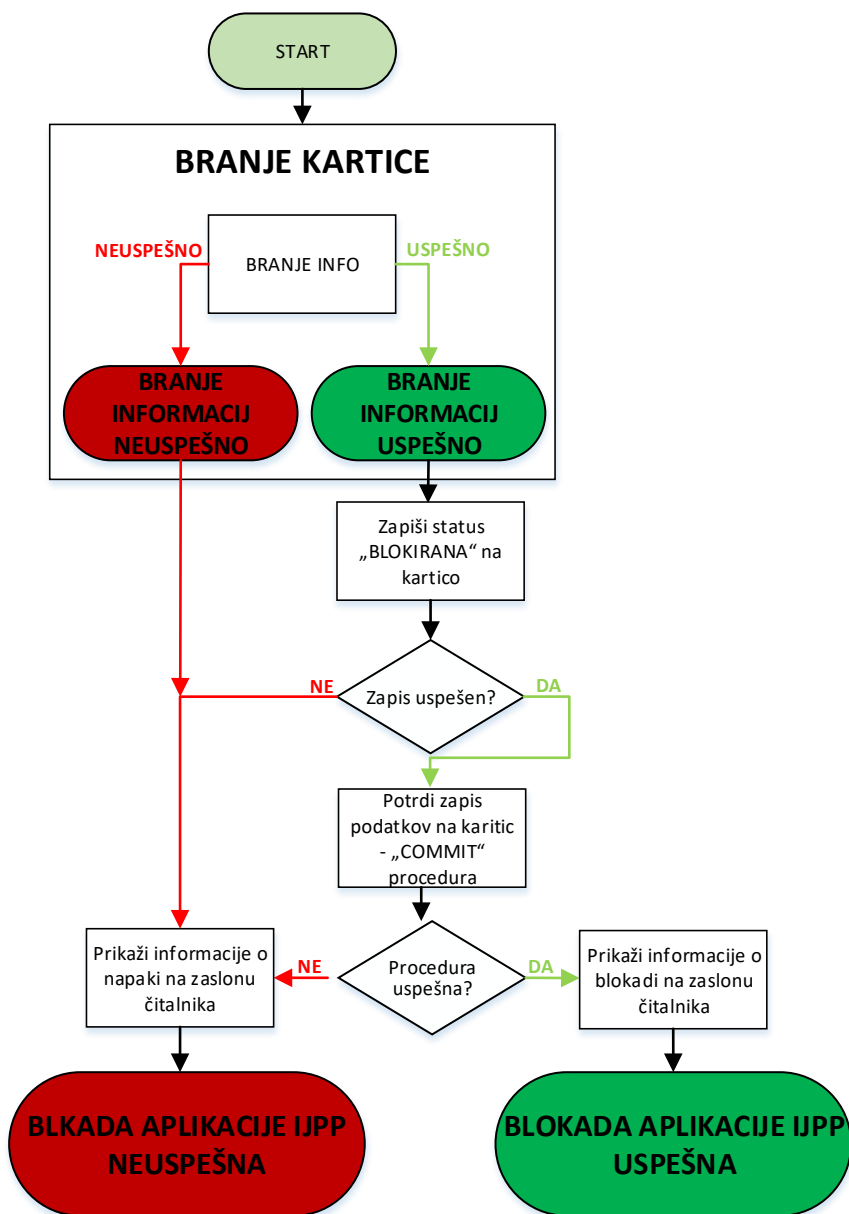
- **Ukaz:** Write Data

Z ukazom aktiviramo postopek pisanja STATUSA kartice, kjer zapišemo vrednost BLOKIRANA.

- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešno pisanje podatkov.

## 6. Stanje potrdi zapis podatkov

Po uspešnem potrjevanju podatkov na kartici uporabnika obvestimo o uspešni blokadi kartice.



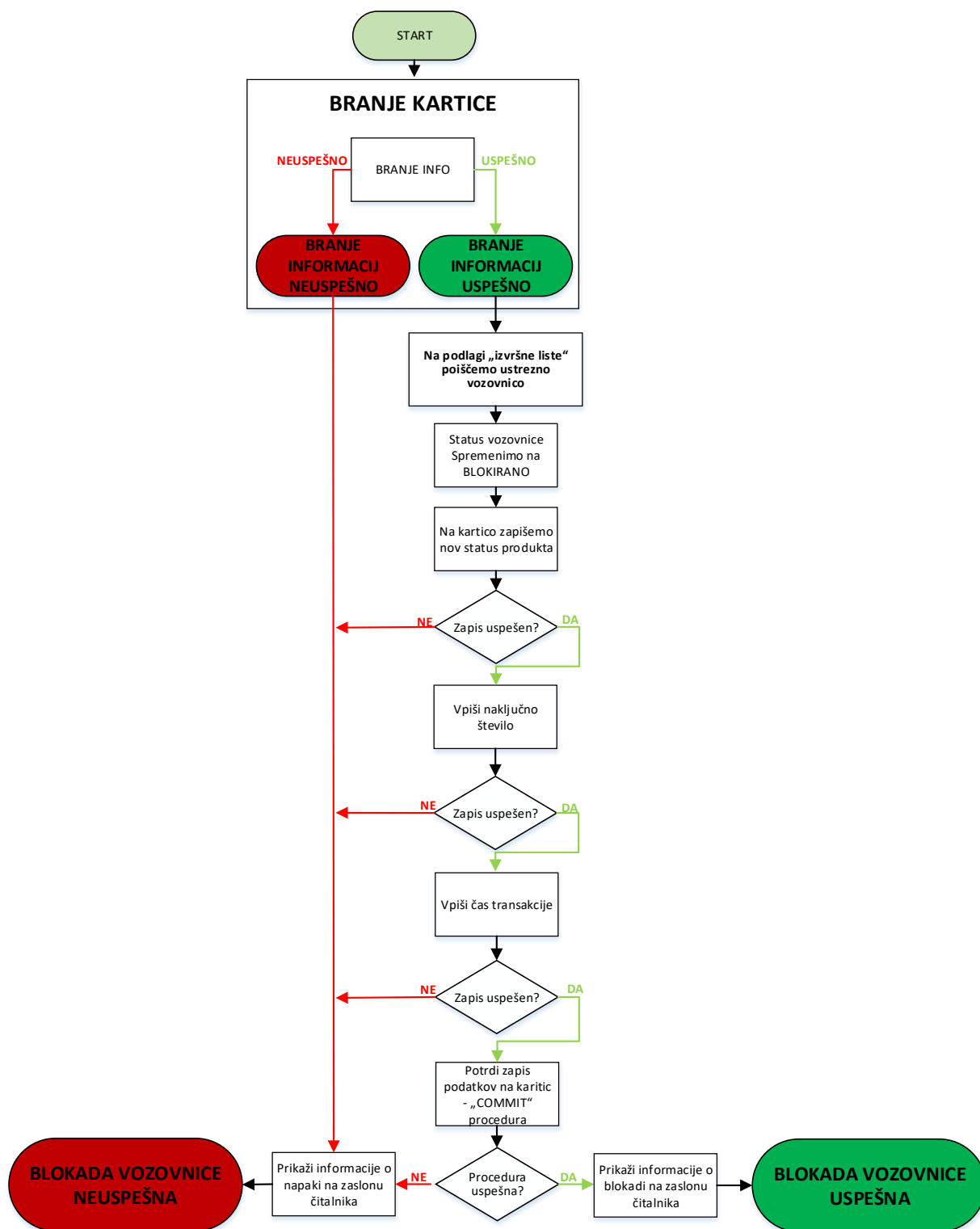
SLIKA 16.11: BLOKADA APLIKACIJE IJPP

### **16.16.7      BLOKADA VOZOVNICE**

Blokada vozovnice se izvede samodejno pri postopku branja info kartice v primeru, da se UID kartice nahaja na izvršni listi. Najprej se izvede branje INFO oz. ustreznih informacij o kartici na podlagi katerih se lahko locira ustrezna vozovnica, ki jo potrebno na podlagi informacij izvršne liste blokirati. Nato se izvrši zapis statusa ustrezne vozovnice na enak način kot pri postopku storno nakupa vozovnice.

#### **16.16.7.1      Protokol BLOKADA VOZOVNICE**

Cilj postopka blokade vozovnice je ustrezen protokol komunikacije med čitalnikom in kartico, ki omogoči, da se na ustrezni kartici blokira ustrezna vozovnica. Postopek blokade vozovnice je razdeljen v različna zaporedna stanja. V posameznih stanjih se izvajajo določeni postopki branja ali pisanja v določene datoteke na kartici. Operacijo blokade vozovnice izvedemo s pomočjo postopka branje INFO in postopka storno nakupa vozovnice.



SLIKA 16.12: BLOKADA VOZOVNICE

#### 16.16.8 PREVERJANJE USPEŠNOSTI ZAPISA VOZOVNICE

Poseben mehanizem preverjanja uspešnosti zapisa vozovnice se aktivira v primeru, da je bila kartica prehitro odstranjena od terminala. V redkih primerih se zgodi, da se podatki ustrezno zapišejo na kartico medtem ko, se operacija aktivacije ali nakupa vozovnice prekine zaradi prehitre odstranitve kartice.

#### 16.16.8.1 Protokol PREVERJANJE USPEŠNOSTI ZAPISA VOZOVNICE

Cilj postopka preverjanja uspešnosti zapisa vozovnice je ustrezen protokol komunikacije med čitalnikom in kartico, ki omogoči, da se na ustrezni kartici preveri uspešnost zapisa vozovnice. Postopek preverjanja uspešnosti zapisa vozovnice je razdeljen v različna zaporedna stanja. V posameznih stanjih se izvajajo določeni postopki branja ali pisanja v določene datoteke na kartici. Operacija preverjanja uspešnosti zapisa se izvede samodejno po postopku aktivacije ali nakupa vozovnice v primeru, da se kartica predčasno odmakne in se s tem postopek prekine.

Če NFC na terminalu zazna, da je bila kartica odstranjena v zadnjem stanju (**Stanje potrdi zapis podatkov**) postopka aktivacije ali nakupa vozovnice, se na terminalu shranijo UID kartice in naključno število zadnje transakcije, ki se je zapisalo v stanju (**Stanje zapis naključnega števila (datoteka CHECK)**). Mehanizem preverjanja uspešnosti zapisa vozovnice se nadaljuje, ko uporabnik ponovno prisloni kartico k terminalu. Postopek preverjanja uspešnosti zapisa se nadaljuje po naslednjih stanjih:

##### 1. Stanje pripravljenosti

- **Akcije**

1. V stanju pripravljenosti se opravijo različni inicializacijski postopki podatkovne strukture, ki se uporablja pri procesu.
2. V strukturo se zapiše **Čas transakcije**
3. V strukturo se zapiše **UID kartice**  
UID se prebere iz kartice s postopkom izbire oz. antikolizije. Tako je UID pripravljen takoj, ko je kartica pripravljena za nadaljnja branja in pisanja podatkov.

- **Pogoji:**

Preveri se UID trenutne kartice in UID, ki se je shranil ob prekinitvi predhodnega postopka.

1. Če je trenutni UID enak UID-eju predhodnega postopka. Nadaljujemo postopek preverjanja uspešnosti zapisa vozovnice.
2. Če UID-eja nista enaka nadaljujemo postopek bodisi aktivacije vozovnice bodisi nakupa vozovnice.

Sledi operacija izbire ustrezne aplikacije (AID) s katero želi čitalnik komunicirati čitalnik pošlje ukaz:

- **Ukaz:** Select Application
- **Pogoj za nadaljevanje postopka v naslednje stanje:**  
Uspešna izbira aplikacije.



## 2. Stanje izbiri aplikacijo

- **Akcije**
  1. V stanju izberi aplikacijo čitalnik pošlje ukaz za avtentikacijo.
  2. Postopek avtentikacije.
- **Ukaz: Authenticate**

Aktivira se postopek avtentikacije med čitalnikom in kartico.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešna avtentikacija med čitalnikom in kartico.

## 3. Stanje avtentikacija

V stanju avtentikacija se po uspešno zaključeni avtentikaciji lahko prične postopek branja in pisanja v zelenih datotekah.

- **Akcije**
  1. Branje verzije kartice oz. datoteke INFO.
- **Ukaz: Read Data**

Z ukazom aktiviramo postopek branja datoteke CHECK na kartici.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešno branje datoteke CHECK.

## 4. Stanje branje kontrolnih podatkov (datoteka CHECK)

- **Akcije:**
  1. Shranimo kontrolne podatke v strukturo.
  2. Branje dela datoteke PRODUCTS, kjer so zapisani statusi posameznih vozovnic.
- **Pogoji:**

Preverimo, če se naključno število na trenutni kartici ujema z naključnim številom predhodnega postopka aktivacije ali nakupa vozovnice. Če se trenutno naključno število ne ujema s predhodnim naključnim številom, lahko ugotovimo, da se vozovnica pri predhodnem postopku ni zapisala na kartico. Če se naključni števili ujemata, ugotovimo, da se je vozovnica pri predhodnem postopku uspešno zapisala na kartico.

  1. Če se naključni števili ne ujemata, nadaljujemo s postopkom aktivacije ali nakupa vozovnice.
  2. Če se naključni števili ujemata, nadaljujemo s postopkom preverjanja uspešnega zapisa vozovnice.
- **Ukaz: Read Value**

Z ukazom aktiviramo branje statusov obstoječih vozovnic na kartici.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešno branje podatkov o statusih posameznih vozovnic.

## 5. Stanje branje statusov vozovnic (datoteka PRODUCTS)

- **Akcije:**
  1. Shranimo statuse posameznih vozovnic, ki so prisotne na kartici.
  2. Branje informacij o vozovnicah oz. preostanek datoteke PRODUCTS. Beremo le tisti del datoteke, ki ustreza veljavnim vozovnicam.
- **Ukaz: Read Data**

Z ukazom aktiviramo branje vozovnic prisotnih na kartici.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešno branje podatkov o vozovnicah v datoteki PRODUCTS.

## 6. Stanje branje vozovnic (datoteka PRODUCTS)

- **Akcije:**
  1. Shranimo posamezne prebrane informacije o vozovnicah v podatkovno strukturo.
- **Pogoji:**
  1. Če se na kartici nahajajo relacijske vozovnice, nadaljujemo z branjem informacij o relacijah oz. datoteke LINE TARIFF DATA.
  2. Če se na kartici ne nahajajo relacijske vozovnice nadaljujemo z branjem zgodovine validacij oz. datoteke HISTORY-VALID.
- **Ukaz: Read Data**
  1. Z ukazom aktiviramo branje datoteke LINE TARIFF DATA (aktiviramo branje statusov relacij) ali datoteke HISTORY-VALID
- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešno branje podatkov.

## 7. Stanje branje statusov relacij (datoteka LINE TARIFF DATA)

- **Akcije:**
  1. Shranimo statuse posameznih relacij, ki so prisotne na kartici.
  2. Branje informacij o relacijah oz. preostanek datoteke LINE TARIFF DATA. Beremo le tisti del datoteke, ki ustreza veljavnim relacijam.
- **Ukaz: Read Data**

Z ukazom aktiviramo branje ustreznih relacij prisotnih na kartici.
- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešno branje podatkov o relacijah v datoteki LINE TARIFF DATA.

## 8. Stanje branje relacij (datoteka LINE TARIFF DATA)

- **Akcije:**
  1. Shranimo posamezne prebrane informacije o relacijah v podatkovno strukturo.
- **Ukaz: Read Data**

Z ukazom aktiviramo branje zgodovine validacij oz. datoteke HISTORY-VALID.

- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešno branje podatkov o zgodovini validacij oz. datoteke HISTORY-VALID.

## 9. Stanje branje zgodovine validacij (datoteka HISTORY-VALID)

- **Akcije:**

1. Shranimo informacije o zgodovini validacij v podatkovno strukturo.
2. Sprožimo postopek spremembe STATUS-a (brisanje) vozovnic, ki so bile v predhodnem postopku zapisane na kartico.

- **Ukaz: Write Data**

Z ukazom aktiviramo postopek zapisa statusa vozovnice v datoteki PRODUCTS.

- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešen zapis podatkov na kartico.

- **Stanje zapis statusov vozovnic (datoteka PRODUCTS)**

V stanju izvedemo ukaz za potrditev zapisanih podatkov na kartico.

- **Ukaz: Commit Transaction**

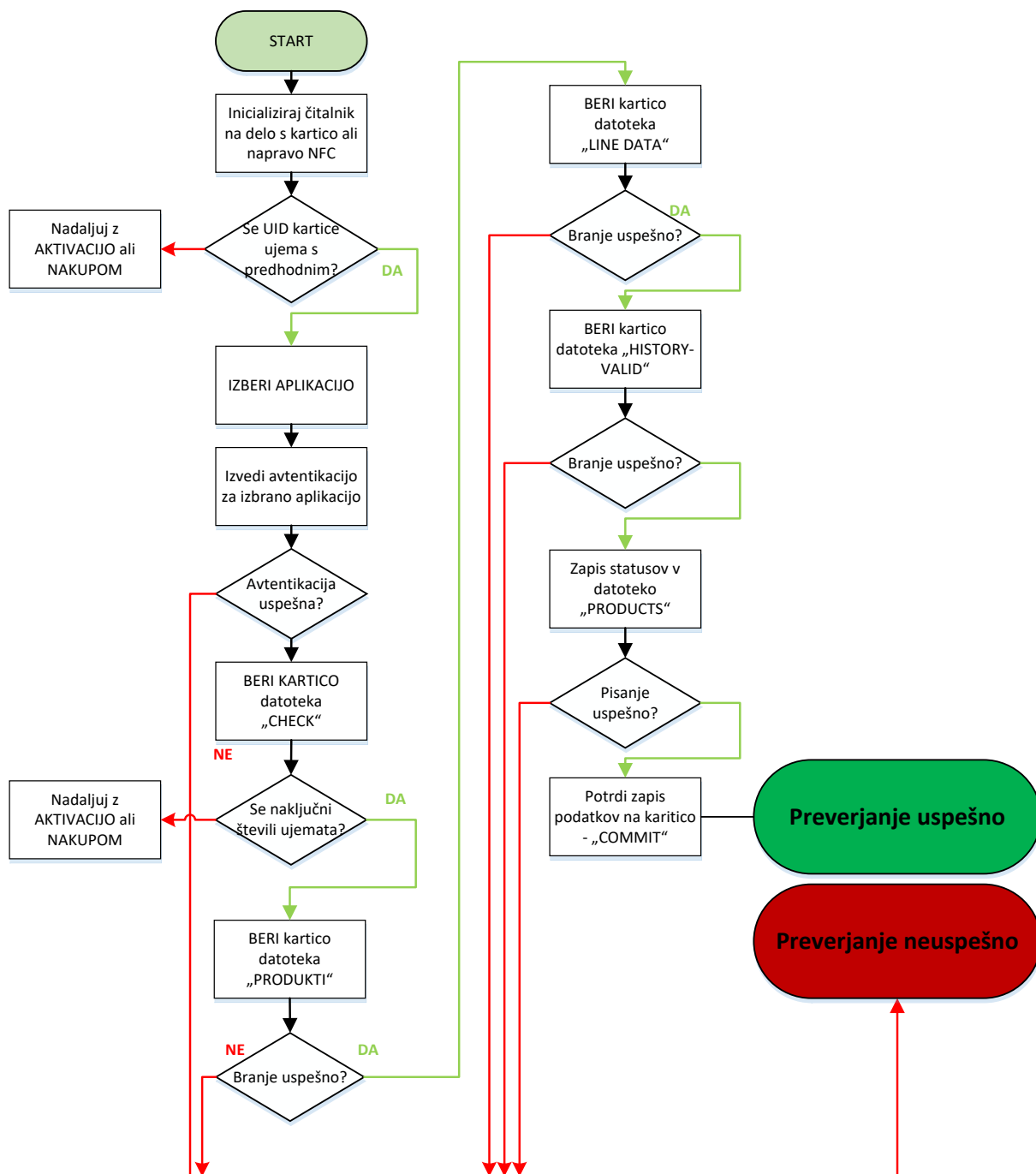
Z ukazom aktiviramo postopek potrjevanja zapisanih podatkov na kartico.

- **Pogoj za nadaljevanje postopka v naslednje stanje:**

Uspešno potrjevanje podatkov.

## 10. Stanje potrdi zapis podatkov

Po uspešnem potrjevanju podatkov na kartici ponovimo prvotni postopek aktivacija ali nakupa vozovnice.



SLIKA 16.13: PREVERJANJE USPEŠNOSTI ZAPISA VOZOVNICE

## 16.17 REFERENCE

[1] – ISO/IEC Standard – ISO/IEC 14443 Identification cards – Contactless integrated circuit cards – Proximity cards.

[2] – ISO/IEC Standard – ISO/IEC 7816 Identification cards – Integrated circuit.

[3] – MF3ICDx21\_41\_81 MIFARE DESFire EV1 contactless multi-application IC Rev. 3.1 — 21 December 2010, 145631.

## 17 DODATEK OBRAZCI ZA TESTIRANJE

## 18 DODATEK OBRAZEC ZA SERVISIRANJE

<b>I. Podatki o mestu posega:</b>							
Družba/podjetnik:		<input style="width: 95%;" type="text"/>					
Prodajno mesto:		<input style="width: 95%;" type="text"/>					
Kontaktna oseba:		<input style="width: 40%;" type="text"/>		Telefon:		<input style="width: 40%;" type="text"/>	
<b>II. Opis opraviła</b>							
	Menjava	Dostava	Popravilo		Menjava	Dostava	Popravilo
Baterija:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nosilec:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Napajalnik:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Terminal:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Papir:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Vračilo terminala:		<input type="button" value="DA"/>		<input type="button" value="NE"/>			
<b>III. IJPP storitev</b>							
Identifikacijski podatki terminala							
Vrsta terminala:	<input type="checkbox"/> Validacijski		<input type="checkbox"/> Prodajni				
Serijska številka (S/N):	<input style="width: 95%;" type="text"/>						
TID:	<input style="width: 95%;" type="text"/>						
SIM PIN:	<input style="width: 95%;" type="text"/>						
Identifikacijski podatki terminala - nadomestni							
Vrsta terminala:	<input type="checkbox"/> Validacijski		<input type="checkbox"/> Prodajni				
Serijska številka (S/N):	<input style="width: 40%;" type="text"/>		_____:		<input style="width: 40%;" type="text"/>		
TID:	<input style="width: 40%;" type="text"/>		_____:		<input style="width: 40%;" type="text"/>		
Vrnjeni terminal je fizično poškodovan: <input type="checkbox"/> DA <input type="checkbox"/> NE Opis poškodbe: <input style="width: 95%;" type="text"/>							
<b>IV. Opombe</b>							
<input style="width: 95%;" type="text"/>							
<input style="width: 95%;" type="text"/>							
<input style="width: 95%;" type="text"/>							
Stranka se v celoti strinja s postopkom / načinom posega v terminal(e) ter ugotovitvami glede stanja vrnjenih oz. prevzetih terminalov.							
IJPP prevzel:		Prevoznik <input style="width: 95%;" type="text"/>					
Ime in priimek	<input style="width: 40%;" type="text"/>			Ime in priimek	<input style="width: 40%;" type="text"/>		
Podpis	<input style="width: 40%;" type="text"/>			Podpis	<input style="width: 40%;" type="text"/>		
Datum in čas:	<input style="width: 95%;" type="text"/>						

