



IS-POKK - TEHNIČNA SPECIFIKACIJA ZA IZDELAVO INFORMACIJSKEGA SISTEMA POKK

Bron d.o.o.

Avtor:	Zlatko Suzić
Datum izdelave:	08. 05. 2025
Zadnja sprememba:	08. 06. 2025
Referenčna št:	
POKK_Tehnicna_Specifikacija.odt	
Verzija:	0.41

1	Uvod in opis projekta.....	11
1.1	Namen dokumenta.....	11
1.2	Obseg rešitve.....	11
1.2.1	Vsebinski obseg.....	11
1.2.2	Organizacijski obseg.....	12
1.2.3	Tehnološki obseg.....	12
1.3	Cilji in pričakovani rezultati.....	12
1.3.1	Glavni cilji.....	12
1.3.2	Pričakovani rezultati.....	13
1.4	Reference na standarde in smernice.....	13
1.4.1	Nacionalne smernice in metodologije.....	13
1.4.2	Mednarodni in pravni standardi.....	14
1.5	Terminologija.....	14
2	Poslovna analiza in funkcionalne zahteve.....	17
2.1	Splošne definicije in razmerja.....	17
2.1.1	Akterji-uporabniki.....	17
2.1.1.1	Obiskovalec.....	17
2.1.1.2	Uporabnik (Avtenticiran uporabnik).....	17
2.1.1.3	Administrator (Avtenticiran uporabnik).....	18
2.1.1.4	Varnostna shema in upravljanje vlog.....	18
2.2	Nefunkcionalne zahteve.....	19
2.2.1	Integracija na platformo eŽivinoreja.....	19
2.2.2	Uporaba API eŽivinoreja (api.zivinoreja.si).....	19
2.2.3	Lokacija strežnikov.....	20
2.2.4	Upoštevanje tehnične specifikacije platforme eŽivinoreja.....	20
2.2.5	Usklajenost z NOO projektom »Digitalizacija podatkovnih zbirk v živinoreji – DigŽiv«	20
2.3	Funkcionalne zahteve.....	20
2.3.1	Skupni predpogoji in merila uspešnosti funkcionalnih zahtev.....	20
2.3.1.1	Predpogoji.....	20
2.3.1.2	Merila uspešnosti.....	21
2.3.1.3	Tabela funkcionalnosti.....	21
2.3.2	SKL-VS: Sklop varnostna shema.....	22
2.3.2.1	SKL-VS-F-000: Vpis v sistem z uporabo uporabniškega imena in gesla.....	22
2.3.2.2	SKL-VS-F-100: Vpis v sistem z žetonom SSO v shemi federativne prijave.....	25
2.3.2.3	SKL-VS-F-200: Izbira zastopništva.....	27
2.3.2.4	SKL-VS-F-300: Odjava uporabnika iz sistema.....	29
2.3.2.5	SKL-VS-F-400: Samodejno podaljševanje seje.....	31
2.3.2.6	SKL-VS-F-500: Izpis osebnih podatkov (GDPR).....	33
2.3.2.7	SKL-VS-F-600: Urejanje osebnih podatkov uporabnika.....	35
2.3.2.8	SKL-VS-F-700: Urejanje podatkov zastopanega subjekta.....	37
2.3.2.9	SKL-VS-F-900: Nastavitve uporabniškega profila.....	39
2.3.2.10	SKL-VS-F-1100: Delegiranje pooblastil v okviru zastopništva.....	41
2.3.2.11	SKL-VS-F-1200: Dodeljevanje pravic zastopništva uporabniku.....	43
2.3.2.12	SKL-VS-F-1300: Upravljanje že dodeljenih zastopništev.....	45
2.3.2.13	SKL-VS-F-1400: Vključitev dvostopenjske avtentikacije (2FA).....	47
2.3.2.14	SKL-VS-F-1500: Preklic vseh aktivnih sej uporabnika.....	49
2.3.2.15	SKL-VS-F-1600: Dostop do zgodovine prijav in varnostnih dogodkov.....	51
2.3.2.16	SKL-VS-F-1700: Zaklepanje uporabniškega računa.....	52
2.3.2.17	SKL-VS-F-1800: Sumljive aktivnosti nad računom.....	54
2.3.2.18	SKL-VS-F-2000: Obnova pozabljenega gesla.....	56
2.3.2.19	SKL-VS-F-2200: Upravljanje uporabnikov.....	59

2.3.2.20 SKL-VS-F-2300: Upravljanje z osebami.....	62
2.3.2.21 SKL-VS-F-2400: Upravljanje z zastopništvu.....	64
2.3.2.22 SKL-VS-F-2500: Povezovanje uporabniškega računa z osebo.....	67
2.3.2.23 SKL-VS-F-2700: Upravljanje kataloga področij (Realm).....	69
2.3.2.24 SKL-VS-F-2800: Upravljanje vlog znotraj področja (Role-Realm).....	71
2.3.2.25 SKL-VS-F-2900: Upravljanje z vlogami zastopništva.....	73
2.3.3 SKL-Kom: Sklop komunikacijski modul.....	76
2.3.3.1 SKL-KOM-F-5000: Enrollment (avtorizacija) KMG v sistem MKGP/UVHVVR.....	78
2.3.3.2 SKL-KOM-F-5100: Sinhronizacija RKG.....	80
2.3.3.3 SKL-KOM-F-5200: Sinhronizacija CRŽ/CRK.....	82
2.3.3.4 SKL-KOM-F-5300: Sinhronizacija šifrantov (GERK, Naslovi, Prostor, Enote, ...).....	86
2.3.3.5 SKL-KOM-F-5700: Sinhronizacija CRK, CRŽ, RKG za podan RKG (MID).....	88
2.3.4 SKL-DW: Sklop podatkovno skladišče.....	91
2.3.4.1 SKL-DW-F-7000: Polni zajem (Full Snapshot Extract).....	91
2.3.4.2 SKL-DW-F-7100: Incremental zajem (Delta Extract).....	93
2.3.4.3 SKL-DW-F-7200: Upravljanje prenosov in zgodovine (Control Management).....	95
2.3.4.4 SKL-DW-F-7300: Čiščenje in standardizacija podatkov.....	97
2.3.4.5 SKL-DW-F-7400: Generiranje nadomestnih ključev (Surrogate Keys).....	99
2.3.4.6 SKL-DW-F-7500: Dimenzijsko modeliranje podatkov.....	101
2.3.4.7 SKL-DW-F-7600: Upravljanje zgodovinskih podatkov (Slowly Changing Dimensions – SCD).....	103
2.3.4.8 SKL-DW-F-7700: Validacija in integriteta transformiranih podatkov.....	104
2.3.4.9 SKL-DW-F-7800: Nalaganje podatkov v DW shemo (Load).....	107
2.3.5 SKL-Ziv: Sklop žival.....	109
2.3.5.1 SKL-ZIV-F-10000: Pregledovanje živali.....	109
2.3.5.2 SKL-ZIV-F-10100: Pregled podatkov o živali.....	110
2.3.5.3 SKL-ZIV-F-10200: Pregled zgodovine podatkov o živali.....	112
2.3.5.4 SKL-ZIV-F-10300: Urejanje podatkov živali.....	113
2.3.5.5 SKL-ZIV-F-10400: Statusno vodenje živali.....	115
2.3.5.6 SKL-ZIV-F-10500: Upravljanje z identifikatorji živali.....	116
2.3.5.7 SKL-ZIV-F-10600: Združevanje podatkov dveh živali.....	117
2.3.5.8 SKL-ZIV-F-10700: Povezovanje živali z uradno evidenco.....	119
2.3.5.9 SKL-ZIV-F-10800: Sinhronizacija podatkov živali z uradno evidenco.....	120
2.3.5.10 SKL-ZIV-F-10900: Spremljanje ključnih dejstev živali (Statusno spremljanje).....	121
2.3.5.11 SKL-ZIV-F-11000: Vse živali pri rejcu.....	123
2.3.5.12 SKL-ZIV-F-11100: Vnos skupine živali.....	124
2.3.6 SKL-Pas: Sklop pasma.....	126
2.3.6.1 SKL-PAS-F-12000: Pregled pasemskih klasifikacij (Pregled in filtriranje).....	126
2.3.6.2 SKL-PAS-F-12100: Kreiranje pasemske klasifikacije.....	127
2.3.6.3 SKL-PAS-F-12300: Posodabljanje pasemske klasifikacije.....	128
2.3.6.4 SKL-PAS-F-12400: Deaktivacija pasemske klasifikacije.....	131
2.3.6.5 SKL-PAS-F-12500: Upravljanje pasemskih podatkov pri živali.....	133
2.3.6.6 SKL-PAS-F-12600: Izračun deležev pasem.....	135
2.3.7 SKL-Gen: Sklop genealogija.....	136
2.3.7.1 SKL-GEN-F-14000: Upravljanje z genealogijo živali.....	136
2.3.7.2 SKL-GEN-F-14100: Potrjevanje ali zavrnitev genealoških povezav.....	138
2.3.8 SKL-Inf: Sklop splošne informacije.....	141
2.3.8.1 SKL-INF-F-15000: Upravljanje statičnih vsebin preko CMS.....	141
2.3.9 SKL-Dog: Sklop dogodki.....	142
2.3.9.1 SKL-DOG-F-17000: Pregled dogodkov.....	142
2.3.9.2 SKL-DOG-F-17100: Vnos splošnega datumskega dogodka.....	144
2.3.9.3 SKL-DOG-F-17600: Seme.....	145

2.3.9.4 SKL-DOG-F-17700: Pripust.....	146
2.3.9.5 SKL-DOG-F-17800: Tretiranje.....	148
2.3.9.6 SKL-DOG-F-17900: Izločitev.....	149
2.3.10 SKL-Drob: Sklop drobnica.....	150
2.3.10.1 SKL-DOG-F-17300: Meritve drobnica.....	150
2.3.10.2 SKL-DOG-F-18000: Izločitev TP Logatec.....	152
2.3.10.3 SKL-DOG-F-18200: Jagnjitev.....	153
2.3.10.4 SKL-DOG-F-18300: Označitev drobnice.....	154
2.3.10.5 SKL-DOG-F-18400: Odstavitev drobnice.....	156
2.3.10.6 SKL-DOG-F-18600: Mlečnost.....	157
2.3.10.7 SKL-DOG-F-18700: Telesne ocene.....	158
2.3.10.8 SKL-DOG-F-18800: Krvna analiza.....	160
2.3.10.9 SKL-DOG-F-18900: Odbira drobnice.....	162
2.3.10.10 SKL-DROB-F-22000: Molža.....	164
2.3.10.11 SKL-DROB-F-22100: Odkup.....	165
2.3.10.12 SKL-DROB-F-22200: Pregled drobnice pri rejcu.....	166
2.3.10.13 SKL-DROB-F-22300: Masovno nalaganje podatkov za drobnico.....	168
2.3.11 SKL-Pras: Sklop prašiči.....	169
2.3.11.1 SKL-DOG-F-17200: Tetoviranje.....	169
2.3.11.2 SKL-DOG-F-17400: Meritve prašičev.....	170
2.3.11.3 SKL-DOG-F-17500: Odstavitev prašičkov.....	172
2.3.11.4 SKL-DOG-F-18100: Prasitev.....	174
2.3.11.5 SKL-DOG-F-18500: Odbira plemenskega podmladka.....	175
2.3.11.6 SKL-PRAS-F-24000: Pregled prašičev pri rejcu.....	177
2.3.11.7 SKL-PRAS-F-24100: Masovni uvoz podatkov za prašiče.....	178
2.3.12 SKL-Kopit: Sklop kopitarji.....	179
2.3.13 eGovedo.....	180
2.3.14 Ečebele.....	180
Modul za področje čebelarstva (eČebele) predstavlja samostojen funkcionalni sklop v okviru platforme eŽivinoreja, ki ga je razvil KIS.....	180
2.3.15 SKL-GB: Sklop genska banka.....	180
2.3.16 SKL-Docs: Sklop dokumenti.....	181
2.3.16.1 SKL-DOCS-F-30000: Upravljanje predlog.....	181
2.3.16.2 SKL-DOCS-F-30100: Upravljanje ključnikov.....	183
2.3.16.3 SKL-DOCS-F-30200: Generiranje dokumentov.....	186
3 Arhitektura informacijske rešitve.....	188
3.1 Splošna arhitekturna zasnova.....	188
3.1.1 Logični sloji arhitekture.....	188
3.1.2 Integracijska arhitektura in sinhronizacija.....	188
3.1.3 Podatkovno skladišče in analitika.....	189
3.1.4 Tehnološke osnove.....	189
3.2 Diagrami arhitekture.....	190
3.3 Povezljivost in integracije.....	194
3.3.1 Tehnične značilnosti integracij.....	194
3.3.2 Formati izmenjave podatkov.....	194
3.4 Uporabljene tehnologije.....	195
3.4.1.1 3.4 Uporabljene tehnologije.....	195
4 Tehnične zahteve in specifikacije.....	195
4.1 Tehnični standardi.....	195
4.1.1 Pregled uporabljenih standardov.....	195
4.1.2 Utemeljitev uporabe.....	196
4.2 Zahteve glede izvajalnih okolij.....	197

4.2.1 Osnovna okolja.....	197
4.2.2 Minimalne sistemske zahteve.....	198
4.2.3 Posebne zahteve.....	198
4.3 Zmogljivost in skalabilnost.....	198
4.3.1 Ključni zmogljivostni kazalniki (KPI).....	198
4.3.2 Načela skalabilnosti.....	199
4.3.3 Podpora obremenitvenemu testiranju.....	199
4.4 Specifikacije integracij.....	199
4.4.1 Vrste integracij.....	200
4.4.2 Specifikacije komunikacije.....	200
4.4.3 Primeri ključnih zunanjih integracij.....	200
4.4.4 Mehanizmi odpornosti in napak.....	200
4.4.5 Sinhronizacijski tokovi.....	201
4.5 Tehnične omejitve.....	201
4.5.1 Tehnične omejitve po kategorijah.....	201
4.5.2 Posebne omejitve vezane na razvoj in razpis.....	202
5 Podatkovna arhitektura in modeli.....	202
5.1 Opis podatkovnih zbirk.....	202
5.1.1 Pregled podatkovnih zbirk.....	202
5.1.2 Organizacija podatkov po shemah.....	202
5.2 Integracijski vmesniki.....	203
5.2.1 Proxy sheme.....	203
5.2.2 Mehanizem dodeljevanja dostopa in način delovanja.....	203
5.2.3 Prednosti arhitekture.....	204
6 Varnostne zahteve.....	204
6.1 Varnostna politika in mehanizmi.....	204
6.1.1 Osnovna varnostna načela.....	204
6.1.2 Mehanizmi in orodja.....	205
6.1.3 Politika varnostnega testiranja.....	205
6.1.4 Upravljanje z incidenti.....	205
6.2 Skladnost s standardi.....	205
6.2.1 Obvezni standardi.....	205
6.2.2 Skladnost v praksi.....	206
6.3 Revizijske sledi.....	206
6.3.1 Journal shema.....	206
6.3.2 Aplikacijsko logiranje.....	207
6.3.3 Nivoji logiranja.....	207
6.3.4 Zahteve za vzdrževanje in arhiviranje.....	207
6.4 Preverjanje in testiranje.....	208
6.4.1 Varnostno preverjanje v okviru CI/CD.....	208
6.4.2 Periodično varnostno testiranje.....	208
6.4.3 Notranja preverjanja revizijske konsistence.....	208
6.4.4 Zahteve za izvajalce.....	208
7 Namestitve in konfiguracija.....	209
7.1 Postopek namestitve.....	209
7.1.1 Posopek namestitve nove verzije.....	209
7.1.2 Sledljivost in arhiviranje.....	210
7.1.3 Odgovornosti.....	210
7.2 Konfiguracijski parametri.....	210
7.2.1 Razvrstitev konfiguracijskih parametrov.....	210
7.2.2 Zahteve.....	211
7.3 Namestitvena navodila.....	211

7.3.1 Vsebina namestitvenih navodil.....	211
7.3.2 Format in dostopnost.....	212
7.4 Okolja.....	212
7.4.1 Vrste okolij in njihov namen.....	212
7.4.2 Ključne značilnosti izvajalnih okolij.....	212
7.4.3 Omejitve in varnost.....	213
8 Testiranje in validacija.....	213
8.1 Načrt testiranja.....	213
8.1.1 Vrste testiranja.....	213
8.1.2 Faze testiranja.....	214
8.1.3 Upravljanje testnih primerov in rezultatov.....	214
8.2 Testni scenariji in primeri uporabe.....	214
8.3 Merila uspešnosti.....	215
8.3.1 Osnovna načela validacije.....	215
8.3.2 Zahteve za testno pokritost.....	215
8.3.3 Prezemni kriterij.....	215
8.4 Dokumentiranje rezultatov.....	215
8.4.1 Osnovna načela.....	216
8.4.2 Struktura dokumentacije.....	216
8.4.3 Shranjevanje in dostopnost.....	216
8.4.4 Prezemna osnova.....	216
9 Upravljanje sprememb in vzdrževanje.....	216
9.1 Postopek upravljanja sprememb.....	216
9.1.1 Faze postopka.....	217
9.1.2 Vloge in odgovornosti.....	218
9.2 Načrt vzdrževanja.....	219
9.2.1 Vrste vzdrževanja.....	219
9.2.2 Organizacija vzdrževanja.....	219
9.2.3 Odzivni časi in prioriteta obravnava.....	219
9.2.4 Trajanje vzdrževanja.....	219
9.3 Revizijske sledi.....	220
9.3.1 Obvezne revizijske komponente.....	220
9.3.2 Dostopnost in hranjenje.....	221
9.4 Komunikacija in podpora.....	221
9.4.1 Kanali komunikacije.....	221
9.4.2 Klasifikacija in odzivni časi.....	221
9.4.3 Dokumentiranje komunikacije.....	221
9.4.4 Podporna dokumentacija.....	222
9.4.5 Usklajevanje in pregled dela.....	222
10 Dokumentacija in navodila za uporabo.....	222
10.1 Predhodni načrt dokumentacije.....	222
10.1.1 Zahtevana dokumentacija.....	222
10.1.2 Oblika in dostava.....	222
10.1.3 Končna priloga.....	222
10.2 Uporabniški priročniki.....	223
10.3 Tehnična dokumentacija.....	223
10.4 Navodila za namestitvev.....	223
10.5 Vzorčni dokumenti.....	223
11 Priloge in reference.....	223
11.1 Seznam prilog.....	223
11.2 Reference.....	237
11.2.1 Dokumenti.....	237

11.2.2 Standardi.....	237
11.3 Normativne povezave.....	238
11.4 Tehnični in vizualni materiali.....	238
12 Ocena del potrebnih za izvedbo.....	238
12.1 Izhodišča za ocenjevanje.....	238
12.2 Vrstni red izvedbe razvoja.....	239
12.3 Ocena del.....	239

Kazalo diagramov

Diagram 1: Primer uporabe razmerji med akterji in vključenost Avtentikacije in Avtorizacije.....	19
Diagram 2: Diagram poteka prijavnne logike.....	26
Diagram 3: Diagram primerov uporabe za SKL-VS-F-100.....	28
Diagram 4: Diagram poteka pri prijavi z SSO žetonom.....	29
Diagram 5: Diagram primerov uporabe za SKL-VS-F-200.....	31
Diagram 6: Diagram primerov uporabe za SKL-VS-F-300.....	33
Diagram 7: Diagram primerov uporabe za SKL-VS-F-400.....	35
Diagram 8: Diagram primerov uporabe za SKL-VS-F-500.....	37
Diagram 9: Diagram primerov uporabe za SKL-VS-F-600.....	39
Diagram 10: Diagram primerov uporabe za SKL-VS-F-700.....	42
Diagram 11: Diagram primerov uporabe za SKL-VS-F-900.....	44
Diagram 12: Diagram primerov uporabe za SKL-VS-F-1100.....	46
Diagram 13: Diagram primerov uporabe za SKL-VS-F-1200.....	49
Diagram 14: Diagram primerov uporabe za SKL-VS-F-1300.....	51
Diagram 15: Diagram primerov uporabe za SKL-VS-F-1400.....	53
Diagram 16: Diagram primerov uporabe za SKL-VS-F-1500.....	55
Diagram 17: Diagram primerov uporabe za SKL-VS-F-1600.....	57
Diagram 18: Diagram primerov uporabe za SKL-VS-F-1700.....	60
Diagram 19: Diagram primerov uporabe za SKL-VS-F-1800.....	62
Diagram 20: Diagram primerov uporabe za SKL-VS-F-2000.....	64
Diagram 21: Diagram primerov uporabe za SKL-VS-F-2200.....	68
Diagram 22: Diagram primerov uporabe za SKL-VS-F-2300.....	71
Diagram 23: Diagram primerov uporabe za SKL-VS-F-2400.....	74
Diagram 24: Diagram primerov uporabe za SKL-VS-F-2500.....	77
Diagram 25: Diagram primerov uporabe za SKL-VS-F-2700.....	79
Diagram 26: Diagram primerov uporabe za SKL-VS-F-2800.....	82
Diagram 27: Diagram primerov uporabe za SKL-VS-F-2900.....	86
Diagram 28: Diagram primerov uporabe za SKL-KOM-F-5000.....	91
Diagram 29: Diagram primerov uporabe za SKL-KOM-F-5100.....	94
Diagram 30: Diagram primerov uporabe za SKL-KOM-F-5200.....	98
Diagram 31: Diagram primerov uporabe za SKL-KOM-F-5300.....	101
Diagram 32: Diagram primerov uporabe za SKL-KOM-F-5700.....	104
Diagram 33: Diagram primerov uporabe za SKL-DW-F-7000.....	107
Diagram 34: Diagram primerov uporabe za SKL-DW-F-7100.....	109
Diagram 35: Diagram primerov uporabe za SKL-DW-F-7200.....	112
Diagram 36: Diagram primerov uporabe za SKL-DW-F-7300.....	115
Diagram 37: Diagram primerov uporabe za SKL-DW-F-7400.....	117
Diagram 38: Diagram primerov uporabe za SKL-DW-F-7500.....	119
Diagram 39: Diagram primerov uporabe za SKL-DW-F-7600.....	121
Diagram 40: Diagram primerov uporabe za SKL-DW-F-7700.....	123
Diagram 41: Diagram primerov uporabe za SKL-DW-F-7800.....	126
Diagram 42: Diagram primerov uporabe za SKL-ZIV-F-10000.....	128
Diagram 43: Diagram primerov uporabe za SKL-ZIV-F-10100.....	130
Diagram 44: Diagram primerov uporabe za SKL-ZIV-F-10200.....	132
Diagram 45: Diagram primerov uporabe za SKL-ZIV-F-10300.....	134
Diagram 46: Diagram primerov uporabe za SKL-ZIV-F-10400.....	136
Diagram 47: Diagram primerov uporabe za SKL-ZIV-F-10500.....	138
Diagram 48: Diagram primerov uporabe za SKL-ZIV-F-10600.....	140
Diagram 49: Diagram primerov uporabe za SKL-ZIV-F-10700.....	142
Diagram 50: Diagram primerov uporabe za SKL-ZIV-F-10800.....	144

Diagram 51: Diagram primerov uporabe za SKL-ZIV-F-10900.....	146
Diagram 52: Diagram primerov uporabe za SKL-ZIV-F-11000.....	148
Diagram 53: Diagram primerov uporabe za SKL-ZIV-F-11100.....	150
Diagram 54: Diagram primerov uporabe za SKL-PAS-F-12102.....	152
Diagram 55: Diagram primerov uporabe za SKL-PAS-F-12100.....	154
Diagram 56: Diagram primerov uporabe za SKL-PAS-F-12300.....	156
Diagram 57: Diagram primerov uporabe za SKL-PAS-F-12400.....	158
Diagram 58: Diagram primerov uporabe za SKL-PAS-F-12500.....	160
Diagram 59: Diagram primerov uporabe za SKL-PAS-F-12600.....	162
Diagram 60: Diagram primerov uporabe za SKL-GEN-F-14000.....	164
Diagram 61: Diagram primerov uporabe za SKL-GEN-F-14100.....	166
Diagram 62: Diagram primerov uporabe za SKL-INF-F-15000.....	168
Diagram 63: Diagram primerov uporabe za SKL-DOG-F-17000.....	170
Diagram 64: Diagram primerov uporabe za SKL-DOG-F-17100.....	172
Diagram 65: Diagram primerov uporabe za SKL-DOG-F-17600.....	174
Diagram 66: Diagram primerov uporabe za SKL-DOG-F-17700.....	176
Diagram 67: Diagram primerov uporabe za SKL-DOG-F-17800.....	178
Diagram 68: Diagram primerov uporabe za SKL-DOG-F-17900.....	180
Diagram 69: Diagram primerov uporabe za SKL-DOG-F-17300.....	182
Diagram 70: Diagram primerov uporabe za SKL-DOG-F-18000.....	184
Diagram 71: Diagram primerov uporabe za SKL-DOG-F-18200.....	186
Diagram 72: Diagram primerov uporabe za SKL-DOG-F-18300.....	188
Diagram 73: Diagram primerov uporabe za SKL-DOG-F-18400.....	190
Diagram 74: Diagram primerov uporabe za SKL-DOG-F-18600.....	192
Diagram 75: Diagram primerov uporabe za SKL-DOG-F-18700.....	194
Diagram 76: Diagram primerov uporabe za SKL-DOG-F-18800.....	196
Diagram 77: Diagram primerov uporabe za SKL-DOG-F-18900.....	198
Diagram 78: Diagram primerov uporabe za SKL-DROB-F-22000.....	200
Diagram 79: Diagram primerov uporabe za SKL-DROB-F-22100.....	202
Diagram 80: Diagram primerov uporabe za SKL-DROB-F-22200.....	204
Diagram 81: Diagram primerov uporabe za SKL-DROB-F-22300.....	206
Diagram 82: Diagram primerov uporabe za SKL-DOG-F-17200.....	208
Diagram 83: Diagram primerov uporabe za SKL-DOG-F-17400.....	210
Diagram 84: Diagram primerov uporabe za SKL-DOG-F-17500.....	212
Diagram 85: Diagram primerov uporabe za SKL-DOG-F-18100.....	214
Diagram 86: Diagram primerov uporabe za SKL-DOG-F-18500.....	216
Diagram 87: Diagram primerov uporabe za SKL-PRAS-F-24000.....	218
Diagram 88: Diagram primerov uporabe za SKL-PRAS-F-24100.....	220
Diagram 89: Diagram primera uporabe SKL-DOCS-F-30000.....	222
Diagram 90: Diagram primerov uporabe SKL-DOCS-F-30100.....	224
Diagram 91: Diagram primera uporabnosti SKL-DOCS-F-30200.....	225
Diagram 92: Diagram visokonivojske predstavitve arhitekture sistema.....	227
Diagram 93: Diagram srednjenivojske predstavitve arhitekture sistema.....	228

1 Uvod in opis projekta

1.1 Namen dokumenta

Namen tega dokumenta je podati **celovito tehnično specifikacijo** informacijskega sistema **POKK** v okviru platforme eŽivinoreja za živalske vrste prašiči, ovce, koze in kopitarjev, ki bo služil kot podlaga za izvedbo postopka javnega naročila po pravilih veljavne zakonodaje in skladno z usmeritvami iz dokumenta *Smernice_JN_IT.pdf*. Dokument predstavlja osnovo za:

- **oceno in oblikovanje ponudb** v postopku javnega naročila,
- **razvoj in implementacijo informacijskega sistema** s strani izbranega izvajalca,
- **pripravo projektne dokumentacije za izvedbo (PZI)** po sklenitvi pogodbe,
- **nadzor in preverjanje skladnosti** rešitve z vnaprej določenimi tehničnimi, funkcionalnimi in varnostnimi zahtevami.

Poleg neposredne uporabe v okviru razpisa je dokument pripravljen tudi kot **izhodišče za pripravo nadaljnjih tehničnih dokumentacij**, ki bodo podpirale dolgoročno vzdrževanje, nadgradnjo in vključitev sistema v širši kontekst digitalizacije javnih evidenc. Dokument služi tudi kot **temelj za definicijo vlog posameznih deležnikov in zaposlenih**, ki bodo sodelovali pri vzdrževanju, upravljanju in uporabi sistema POKK.

Specifikacija temelji na zahtevah naročnika, usklajenih z deležniki sistema (KIS, BF, VF, MKGP, UVHVVR) ter je oblikovana skladno z **metodologijo GTZ 2.3**, upoštevajoč **standardizirane tehnološke in varnostne prakse**, predpisane za informacijske rešitve v javnem sektorju.

1.2 Obseg rešitve

Informacijski sistem **POKK** je zasnovan kot celovita digitalna rešitev za podporo upravljanju populacij živali v rejnih in raziskovalnih okoljih, s posebnim poudarkom na ohranjanju genetske raznovrstnosti, spremljanju populacijskih kazalnikov in sledenju živalskih premikov. Sistem mora zagotoviti enoten, sledljiv, modularno zasnovan in dolgoročno vzdrževan informacijski okvir, v katerega bodo vključeni različni deležniki s področja raziskav, stroke in upravljanja rej.

1.2.1 Vsebinski obseg

Vsebinski obseg rešitve vključuje:

- **Evidenco populacij** in identifikacijo živali po različnih živalskih vrstah (govedo, drobnica, prašiči, konji ipd.), z upoštevanjem shem porekla, statusov in povezanih enot.
- **Sledenje premikom** živali znotraj in med rejami ter povezljivost z nacionalnimi ali obstoječimi evidencami (npr. veterinarske baze).
- **Upravljanje rodovniških podatkov** in podporo genetski analizi ter oceni in spremljanju genske raznolikosti.
- **Vmesnike za zajem podatkov** iz zunanjih virov in naprav (laboratoriji, tehtnice, čitalniki), kjer je to smiselno, ter možnost ročnega vnosa ali uvoza.

- **Upravljanje uporabnikov in vlog** z naprednim mehanizmom zastopanja in večnamenskega dostopa (multi-reprezentanca).
- **Modul za poročanje in analitiko**, ki omogoča pripravo strukturiranih poročil, pregledov po populacijah, statusih in rodovniških razmerjih.
- **Podporo za večjezikovno uporabo** (najmanj slovenščina in angleščina), lokalizacijo šifrantov in prilagoditev uporabniškega vmesnika.

1.2.2 Organizacijski obseg

Sistem bo vzpostavljen v sodelovanju med tremi ključnimi partnerji (KIS, BF, VF).

Vzpostavljen bo **enoten strokovni odbor**, ki bo usklajeval spremembe in nadzoroval skladnost s standardi.

Vključene bodo vse pravne osebe, ki sodelujejo kot rejci, rejske organizacije, javne službe, inšpekcijski organi, raziskovalci in druge strokovne ustanove.

1.2.3 Tehnološki obseg

Rešitev naj bo skladna z **metodologijo GTZ 2.3**, upoštevajoč **standardizirane tehnološke in varnostne prakse**, predpisane za informacijske rešitve v javni sektorju, z obvezno podporo za **izvajalna okolja za vsebnike s funkcijo orkestratorja po GTZ 2.3**

Podatkovna zasnova mora omogočati **več-shemsko strukturo**, ločevanje domen, podporo sinhronizaciji s tujimi sistemi in pripravo na podatkovno skladišče (DW).

Vzpostavljeni morajo biti **proxy mehanizmi za dostop** do vsebinskih podatkov in **revizijska sledljivost** preko journal shem.

Predvidena je dolgoročna možnost vključitve v **državno infrastrukturo DRO**, zato mora biti rešitev že v tej fazi skladna s pripadajočimi tehničnimi standardi (GTZ, seznam podprte opreme, varnostni standardi).

1.3 Cilji in pričakovani rezultati

Vzpostavitev informacijskega sistema POKK je usmerjena v dolgoročno podporo upravljanju genskih virov domačih živali živalskih vrst prašiči, ovce, koze in kopitarjev, spremljanju populacij, ter izboljšanju digitalne povezljivosti vseh deležnikov v živinorejski verigi. Projekt naslavlja tako potrebe po zanesljivi informacijski infrastrukturi kot tudi strateške cilje s področja trajnostnega upravljanja živalskih populacij.

1.3.1 Glavni cilji

Vzpostaviti enotno in konsolidirano informacijsko rešitev za obvladovanje vseh populacijskih, identifikacijskih in rodovniških podatkov genskih virov domačih živali.

Zmanjšati podvajanje funkcionalnosti in prehod od parcialnih, ločeno razvitih rešitev k skupnemu, standardiziranemu sistemu.

Zagotavljati sledljivost, revizijsko pokritost in varnost vseh operacij nad podatki, z vključenimi mehanizmi za nadzor dostopa, revizijske sledi in zaščito osebnih podatkov.

Omogočiti večnivojski dostop in zastopanje (multi-reprezentanca) uporabnikov, institucij in

strokovnih teles, skladno z njihovimi pristojnostmi.

Povečati informacijsko zrelost institucij s področja genskih virov z enotnim pristopom k upravljanju uporabnikov, vlog, podatkov in sprememb.

Uskladiti rešitev z nacionalnimi standardi in zahtevami za vključitev v DRO, pri čemer se že v tej fazi zagotavlja skladnost z GTZ metodologijo.

Zagotoviti vključevanje strokovnih deležnikov v vse faze razvoja in vzdrževanja sistema ter sistematično upoštevati dobre prakse s področja informacijske tehnologije, informacijske varnosti, modeliranja podatkov in modularne arhitekture.

1.3.2 Pričakovani rezultati

Z izvedbo projekta bodo doseženi naslednji konkretni rezultati:

- Delujoč in produkcijsko vzpostavljen **informacijski sistem POKK** z ločenimi moduli za posamezne funkcionalne sklope.
- **Migracija in uskladitev obstoječih podatkovnih virov**, z vzpostavljeno konsistentno shemo za prašiče, ovce, koze in kopitarje.
- Vzpostavitev **revizijskega mehanizma**, vključno z žurnalno shemo, zapisi o spremembah in centralizirano analitiko (OpenSearch).
- Implementacija **sistema vlog, uporabniških pravic in zastopanja** z možnostjo večnivojske avtorizacije.
- Dokumentiran, avtomatiziran in sledljiv proces **nameščanja, testiranja, uvajanja in podpore**, vključno z CI/CD in testnimi okolji.
- **Vzpostavljena praksa sodelovanja z deležniki na strokovni ravni**, ki omogoča transparentno, usklajeno in dolgoročno vzdrževanje ter nadgrajevanje sistema.
- Pripravljena podlaga za **vključitev sistema v nacionalno infrastrukturo DRO**, z uporabo dovoljene programske opreme in skladnostjo z arhitekturnimi smernicami.

1.4 Reference na standarde in smernice

Tehnična specifikacija za informacijski sistem POKK temelji na naslednjih nacionalnih smernicah in mednarodnih standardih, ki določajo metodološke, varnostne, arhitekturne in tehnološke zahteve za razvoj informacijskih rešitev v javnem sektorju.

1.4.1 Nacionalne smernice in metodologije

Upoštevajo se nacionalne smernice in metodologije:

- *Genericne tehnološke zahteve GTZ 2.3* – metodološki okvir za modularno in varnostno skladno načrtovanje sistemov.
- *Smernice za pripravo tehničnih specifikacij IT (Smernice_JN_IT.pdf)* – pripomoček za skladno in primerljivo pripravo razpisne dokumentacije.
- *Seznam podprte programske opreme DRO (Podprta-programska-oprema-DRO_V1.0.pdf)* – omejitev na dovoljene tehnologije in orodja v rešitvah, ki bodo lahko vključene v državne registre in infrastrukturo.

1.4.2 Mednarodni in pravni standardi

Upoštevajo naslednji mednarodni in pravni standardi:

- **ISO/IEC 27002:2022** – standard za upravljanje varnosti informacij in oblikovanje varnostnih politik in kontrol.
- **OWASP TOP 10** – priporočila za varnost spletnih aplikacij in zaščito pred najpogostejšimi ranljivostmi.
- **TLS 1.3** – obvezen protokol za šifriranje komunikacij v sistemu.
- **OAuth 2.0 / JWT** – standard za varno avtorizacijo in podpisane prijavnne žetone.
- **ZVOP-2** – Zakon o varstvu osebnih podatkov, ki določa pravni okvir za obdelavo osebnih podatkov v RS.

Podrobnosti o sklicevanju, vlogi posameznega vira in razmejitvi med obveznimi in informativnimi referencami so navedene v poglavju **237 - Reference** in **238 - Normativne povezave**.

1.5 Terminologija

2FA	dvofaktorska avtentikacija
ACID	atomicity, consistency, isolation, durability - funkcionalnosti podatkovne zbirke
AKTRP	Agencija za kmetijske trge in razvoj podeželja
API	Application Program Interface – programski vmesnik
AVG	povprečje
BASE	basically available, soft state, and eventually consistent – funkcionalnost podatkovne zbirke
BF	Biotehnična fakulteta
BI	Business intelligence – poslovno poročanje
CI/CD	continuous integration (CI) and continuous delivery (CD)
CLOB	Character Large Object – standardna oblika SQL standarda
CMS	Content Management System
CRG	Centralni register gospodarstev
CRK	Centralni register kopitarjev
CRŽ	Centralni register živali
CSFR	internetni napad
CSS	Cascading Style Sheets – opis stila maske
CSV	Coma separated values – vrednosti ločene z vejico – oblika zapisa podatkov
CVSS	Common Vulnerability Scoring System
DAL	Data acces layer
DAST	Dynamic application security testing
DB link	Database link

DigŽiv	Projekt »Digitalizacija podatkovnih zbirk v živinoreji«
DMZ	demilitarizirano območje,
DOCX	format dokumentov
DRO	Državni računalniški oblak
DŠ	davčna številka
DW	Datawarehouse - podatkovno skladišče
EMŠO	enotna matična številka občana
EPL	internetna povezava
ER-model	Entitetno relacijski model
ES	Evropska skupnost
EŠD	enota stavbnega dela
ETL	Extract, Transfer, Load
EU	Evropska Unija
FK	tuji ključ – v podatkovni zbirki
G-MID	Identifikacijska številka gospodarstva
GDPR	Splošna uredba Evropske Unije o varstvu podatkov
GERK	Grafična enota rabe kmetijskega gospodarstva
GTZ	Generalne tehnološke zahteve
GUI	Grafični uporabniški vmesnik
GURS	Geodetska Uprava republike Slovenije
HA	Visoka razpoložljivost
HKOM	Centralno komunikacijsko omrežje državne uprave
HTML	standard markup language za web strani
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IDP	Identity provider
IP	Internet Protocol
ISO/IEC	Mednarodni standard določen s strani mednarodne organizacije za standardizacijo in mednarodnim komitejem za elektrotehniko
IT	informacijska tehnologija
JDBC	Java Database Connectivity API
JN	Javno naročilo
JN IT	Javni naročilo informacijske tehnologije
JPG	Slikovni format
JSKS	Javna služba kmetijskega svetovanja
JSON	JavaScript Object Notation
JWT	JASON web token
KIS	Kmetijski Inštitut Slovenije
KMG-MID	identifikacijska številka kmetijskega gospodarstva
MDP	Ministrstvo za digitalno preobrazbo
MIT	Tehnološki institut Massachusettsa
MJU	Ministrstvo za javno upravo

MKGP	Ministrstvo za kmetijstvo, gozdarstvo in prehrano
MS	Microsoft
NOO	Načrt okrevanja in obnove
NSP	neznan starševski prispevek
NTP	Network Time protocol
Oauth	Industrijski standard za avtorizacijo
ODBC	Open Database Connectivity protocol
OLAP	Online analytical processing
OLTP	Online transaction processin
OWASP	Open Web Application Security Project
PDF	Portable Document Format (Adobe Acrobat)
PHYPHAR	Register FFS
PK	primarni ključ pri podatkovnih zbirkah
PV	plemenske vrednosti
POKK	Informacijski sistem za prašiče, ovce, koze in konje
PZI	Projekt za izvedbo
QR koda	matrična oz. Dvorazsežna črtna koda
RBAC	Role based access control
RDBMS	Relational database management system
REST API	Representational State Transfer, nabor pravil za povezovanje med servisi
RFC	Request for Change
RKG	Register kmetijskih gospodarstev
RS	Republika Slovenija
SAST	Static application security testing
SCD	Slowly Changing Dimensions
SI-PASS	enotna točka za preverjanje identitete različnih uporabnikov
SIEM	sistem za spremljanje incidentov
SMS	Short Message Service – kratka sporočila
smtp	Simple mail transfer protocol
SP	Service provider
SP	samostojni podjetnik
SQL	Structured Query Language
SSO	Single Sign-on, shema preverjanja pristnosti, ki uporabnikom omogoča prijavo v več programskih sistemov z eno samo identiteto
SUM	seštevek
TCP	Transmission Control Protocol
TLS	Transport Layer security
TOTP	Time-based One-Time Password
UAT	User acceptance test
UI	User interface
UML	Unified Modeling Language – jezik za modeliranje
UVHVVR	Urad za varno hrano, veterinarstvo in varstvo rastlin

UX/UI	User xperience / user interface
VF	Veterinarska fakulteta
VPN	Virtual private network
VS	Varnostna shema
WCAG	Web Content Accessibility Guidelines
XLSX	format razpredelnic
XML	Extensible Markup Language
YAML	human-readable data serialization language
ZDSMA	Zakon o dostopnosti spletišč in mobilnih aplikacij
ZIP	Protokol za stiskanje podatkov
ZIV	žival
ZVOP	Za-kon o varstvu osebnih podatkov

2 Poslovna analiza in funkcionalne zahteve

2.1 Splošne definicije in razmerja

2.1.1 Akterji-uporabniki

Informacijski sistem temelji na vlogah, ki jih prevzemajo različni tipi akterjev glede na način dostopa, avtentikacije in avtorizacije. Akterji morajo biti usklajeni z uporabniki v okviru platforme eŽivinoreja.

2.1.1.1 Obiskovalec

Akter tipa obiskovalec, ni prijavljen v sistem in nima dodeljene vloge ali zastopniškega konteksta. Njegove možnosti so omejene na dostop do javno objavljenih informacij na uporabniškem vmesniku. Sistem zanj ne vzpostavlja uporabniške seje in ne beleži operacij, razen osnovnih revizijskih dogodkov na ravni IP-naslova in uporabniškega agenta (npr. za spremljanje dostopov do javnih virov).

2.1.1.2 Uporabnik (Avtenticiran uporabnik)

Akter tipa uporabnik je vsak obiskovalec, ki je uspešno opravil postopek prijave v sistem z enim izmed podprtih mehanizmov (lokalna prijava, federativna prijava preko SI-PASS, API ključ, žeton itd.). Po prijavi mora izbrati subjekt, ki ga zastopa, in v okviru tega konteksta deluje kot nosilec ene ali več vlog. Akter tega tipa tako pridobi razširjene lastnosti in funkcionalnosti poleg dedovanih lastnosti obiskovalca.

V to skupino sodijo:

- **interaktivni uporabniki** (npr. rejci, svetovalci, administratorji),
- **tehnični uporabniki** (npr. integracijski moduli, avtomatske sinhronizacije), ki imajo skupne avtorizacijske mehanizme in dodeljene vloge prek varnostne sheme sistema.

Dostop do funkcionalnosti je pogojen s kombinacijo:

- **identitete uporabnika,**
- **izbranega zastopništva (subjekta),**

- **dodeljenih vlog znotraj konteksta zastopništva.**

Sistem vsakič preveri avtorizacijo z vključeno funkcionalnostjo “Avtoriziraj uporabnika”, ki potrdi veljavnost seje, prisotnost zastopništva in ustreznost vloge za želeno operacijo.

2.1.1.3 Administrator (Avtenticiran uporabnik)

Akter tipa administrator je hkrati uporabnik sistema, s to izjemo, da lahko vedno trdimo, da ima ta uporabnik dodeljene vloge administratorja. V kontekstu procesa to običajno pomeni, da ima administratorski uporabnik dovolj velike (široke) pravice, ki mu dovoljujejo izvedbo funkcionalnosti mimo običajnih mehanizmov zastopništva.

2.1.1.4 Varnostna shema in upravljanje vlog

Informacijski sistem POKK uporablja večnivojski model avtorizacije na osnovi t. i. *področij* (*Realm*) in *vlog* (*Role*), kjer velja naslednje:

- **Področje (Realm)** označuje funkcionalni del sistema (npr. ZIV, PAS, DOG) oziroma tip uporabnika (Veterinar, Revizor, Genetik),
- **Vloga (Role)** definira, kaj uporabnik lahko počne v tem področju.

Sistem ne vsebuje vnaprej določenega nabora vlog in področji – vsi objekti so **dinamično definirani in upravljeni s strani administratorja**, ki ima sistemsko dodeljeno vlogo `SISTEMSKI_ADMINISTRATOR`. Administrator lahko ustvari nova področja, vloge in povezave med področji in vlogami, ki se uporabljajo pri ročnem ali avtomatičnem dodeljevanju uporabnikom.

Za vse vloge se priporoča standardizirana konvencija poimenovanja:

- `IME_VLOGE_READ`
- `IME_VLOGE_WRITE`

Kjer:

- `READ` pomeni dostop za vpogled (npr. pregled podatkov, poročil),
- `WRITE` pomeni dostop za spreminjanje, upravljanje ali izvajanje operacij.

Sistem dovoljuje, da ima posamezen uporabnik več zastopništev hkrati, pri čemer ima v vsakem zastopniškem kontekstu lahko dodeljen drugačen nabor vlog. Vloge so aktivne le znotraj izbranega zastopništva in veljajo le za trajanje aktivne avtorizacijske seje.

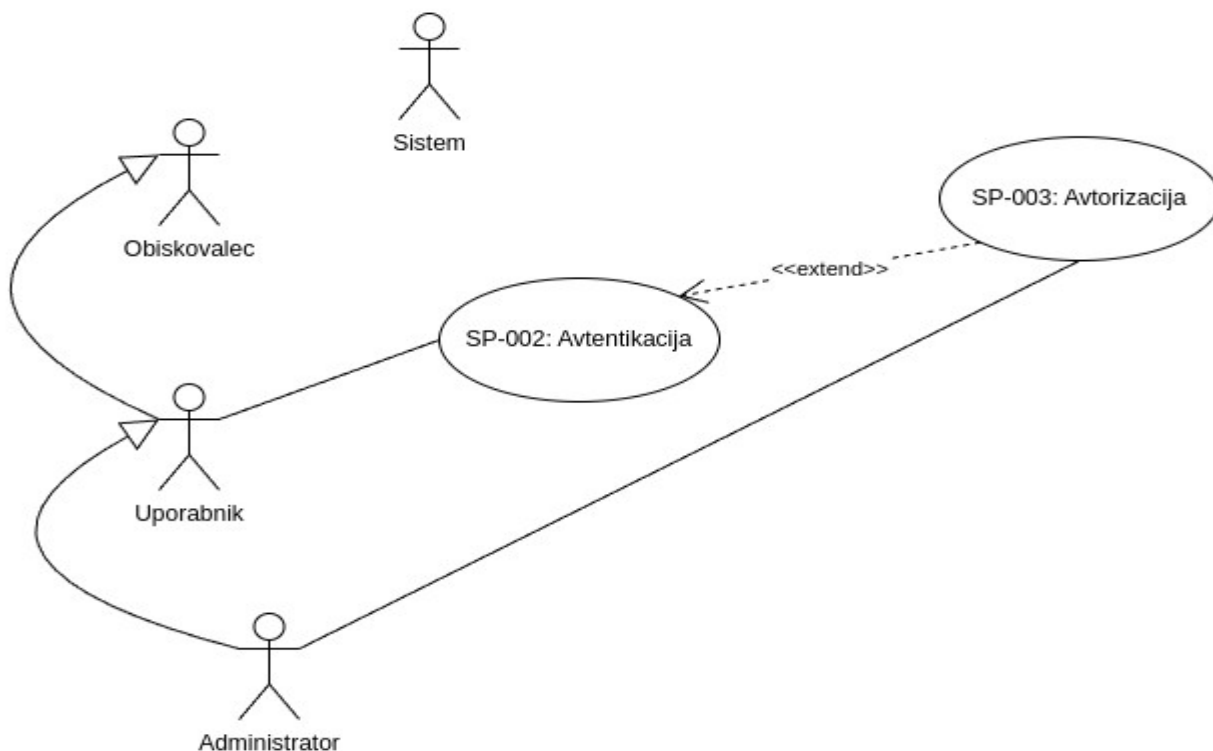


Diagram 1: Primer uporabe razmerji med akterji in vključenost Avtentikacije in Avtorizacije.

2.2 Nefunkcionalne zahteve

2.2.1 Integracija na platformo eŽivinoreja

Vse rešitve, razvite v okviru informacijskega sistema POKK, morajo biti popolnoma integrirane v obstoječo platformo eŽivinoreja (<https://e.zivinoreja.si/>), ki predstavlja skupno informacijsko platformo namenjena upravljanju s podatki v živinoreji in osrednjo vstopno točko za uporabnike javnih služb, rejcev, strokovnih organizacij in nadzornih organov. Integracija vključuje tako enotno uporabniško izkušnjo (SSO, skupni uporabniški vmesnik, konsistentna navigacija) kot tudi tehnično povezljivost (uporaba skupnih šifrantov, API komunikacija z obstoječimi moduli, standardizirani formati izmenjave podatkov).

V skladu s specifikacijami projekta POKK se vse aplikacije in moduli razvijejo z mislijo na sprotno vključevanje v platformo eŽivinoreja, pri čemer mora biti vsaka komponenta najkasneje do 1.1.2026 operativno vključena. Zagotoviti je treba tudi popolno interoperabilnost z obstoječimi rešitvami, skladnost z varnostnimi politikami platforme ter vključitev v skupni nabor storitev za identifikacijo, avtorizacijo in beleženje dejavnosti uporabnikov. Tehnična integracija mora temeljiti na arhitekturnih in varnostnih zahtevah iz GTZ 2.3 in ISO in biti skladna s smernicami za DRO in dokumentom „Podprta programska oprema“.

2.2.2 Uporaba API eŽivinoreja (api.zivinoreja.si)

Vsi novi API vmesniki v okviru sistema POKK se razvijajo izključno v primerih, ko enakovredna funkcionalnost ni že vzpostavljena v okviru obstoječega sistema api.zivinoreja.si. Obvezna je analiza že obstoječih storitev in ponovna uporaba funkcionalnosti, kjer je to možno. Podvajanje funkcionalnosti ni dopustno, nova implementacija pa mora biti predhodno strokovno utemeljena in potrjena s strani skupnega razvojnega odbora.

2.2.3 Lokacija strežnikov

Fizična namestitvev strežnikov, na katerih deluje informacijski sistem POKK se uskladi s projektnimi partnerji. Namestitev vključuje tako aplikacijsko kot podatkovno plast, pri čemer se vzpostavi ustrezna povezljivost z nacionalnimi registri in zunanjo komunikacijo skladno z varnostnimi zahtevami projekta.

2.2.4 Upoštevanje tehnične specifikacije platforme eŽivinoreja

Razvoj, arhitektura in integracija informacijskega sistema POKK morajo biti skladni z zahtevami in usmeritvami, določenimi v dokumentu *eŽiv_teh_specifikacija* v2. Ta dokument predstavlja obvezno izhodišče za vse vidike povezovanja z obstoječo platformo eŽivinoreja, vključno z:

- uporabo obstoječih komunikacijskih vmesnikov (API na naslovu <https://api.zivinoreja.si>),
- upoštevanjem varnostnih in infrastrukturnih zahtev,
- zagotavljanjem uporabniške konsistence in kompatibilnosti z grafičnimi elementi uporabniškega vmesnika,
- spoštovanjem načel modularnosti in odgovornosti za implementacijo in vzdrževanje komponent.

Odstopanja od specifikacije so dopustna le ob predhodni odobritvi skupnega razvojnega odbora ter morajo biti tehnično in vsebinsko utemeljena v PZI dokumentaciji.

2.2.5 Usklajenost z NOO projektom »Digitalizacija podatkovnih zbirk v živinoreji – DigŽiv«

Informacijski sistem POKK mora biti v celoti usklajen z namenom in cilji projekta »Digitalizacija podatkovnih zbirk v živinoreji – DigŽiv«, financiranega iz Načrta za okrevanje in odpornost (NOO). Rešitev mora podpirati centralizacijo živinorejskih podatkov, digitalizacijo javnih služb, vzpostavitev skupne vstopne točke za uporabnike ter povezljivost z državnim informacijskim sistemom.

Vse komponente morajo prispevati k integraciji obstoječih sistemov partnerjev (KIS, UL BF, UL VF), zagotavljati funkcionalno nadgradnjo platforme eŽivinoreja in biti skladne z določili dokumentacije projekta in investicijskega programa NOO.

2.3 Funkcionalne zahteve

2.3.1 Skupni predpogoji in merila uspešnosti funkcionalnih zahtev

V nadaljevanju so navedeni **splošni predpogoji, merila uspešnosti in tabela funkcionalnosti**, ki veljajo **za vse funkcionalnosti sistema POKK**, razen kadar je v posamezni funkcionalni zahtevi izrecno določeno drugače. Ta določila predstavljajo osnovni okvir za zagotavljanje skladnosti, interoperabilnosti in pričakovane kakovosti izvajanja vseh funkcionalnosti v produkcijskem okolju.

2.3.1.1 Predpogoji

- **Uporabniški račun:** Uporabnik mora biti avtenticiran z uporabo mehanizma, ki ga sistem podpira (lokalna prijava, federativna prijava – SSO).
- **Avtorizacija:** Uporabnik mora imeti ustrezne pravice za izvajanje posamezne

funkcionalnosti (ustrezna vloga dodeljena preko zastopništva). **Ta razmerja se eksplicitno ne modelirajo v UML diagramih, saj se predpostavlja, da jih vsaka funkcionalnost inherentno vključuje.**

- **Dostopnost komponent:** Vse vključene tehnične komponente (npr. podatkovna baza, sejni sistem, 2FA mehanizem, komunikacijski strežnik) morajo biti ob izvajanju funkcionalnosti dostopne in operativne.
- **Konfiguracija:** Sistemske nastavitve (npr. dolžina sej, število dovoljenih napak, razpoložljivost storitev) morajo biti uspešno naložene in skladne z veljavno konfiguracijsko politiko.
- **Revizijsko beleženje:** Aktivno mora biti beleženje dogodkov za vse operacije, ki vplivajo na varnost, integriteto ali sledljivost podatkov. **Primeri uporabe revizijskega beleženja se v nadaljevanju praviloma ne modelirajo, saj se predpostavlja, da jih vsaka funkcionalnost inherentno vključuje (kjer je to potrebno).**

2.3.1.2 Merila uspešnosti

- **Časovni odziv:** Vsaka funkcionalnost mora zagotavljati odziv v času, ki je ustrezen glede na naravo zahteve:
 - ≤ 1000 ms za osnovne operacije (pridobi, shrani, prikaži),
 - ≤ 2 s za potrditve z več koraki (npr. preverjanje kode, preverjanje pravic),
 - asinhronne operacije (npr. izvoz podatkov) morajo sprožiti potrditev sprejema zahteve v ≤ 2 s.
- **Zanesljivost:** V 99,9 % primerov mora funkcionalnost izvesti zahtevek uspešno (brez napake zaradi sistemskih komponent).
- **Revizijska sledljivost:** Vsi dogodki, ki vplivajo na podatke ali dostop, morajo biti zabeleženi z vsaj naslednjimi metapodatki: uporabnik, čas, IP naslov, izvorna komponenta, rezultat.
- **Varnost:** Vse funkcionalnosti morajo spoštovati varnostne politike sistema (TLS 1.3, OAuth2/JWT, 2FA, zaklepne politike, zaščita pred CSRF).
- **Uporabniška informiranost:** Uporabnik mora biti vedno seznanjen z rezultatom izvedene akcije (uspeh, napaka, opozorilo) in morebitnimi nadaljnjimi koraki.
- **Dostopnost in odzivnost vmesnika:** Uporabniški vmesnik mora zagotavljati vizualno odzivnost in validacijo v realnem času tam, kjer je to možno (npr. vnosne forme, preverjanje pravilnosti formata).

2.3.1.3 Tabela funkcionalnosti

ID	Naziv	Opis	Pogoj uporabe	Opombe
SP-001	Logiranje dogodkov v OpenSearch	Vse funkcionalnosti sistema POKK generirajo varnostno in revizijsko sled v centralnem iskalniku	Vedno prisotno	Zaradi prisotnosti v vseh funkcionalnostih se v posameznih poglavjih ne modelira posebej.

ID	Naziv	Opis	Pogoj uporabe	Opombe
		(OpenSearch), vključno z vsemi metapodatki.		
SP-002	Avtentikacija uporabnika	Uporabnik mora biti prijavljen z veljavnim avtentikacijskim mehanizmom (lokalno ali SSO), kadar funkcionalnost vključuje uporabniško interakcijo.	Zahtevano, če funkcionalnost ni javna ali anonimizirana	Če ni potrebna prijava, se ne zahteva niti avtorizacija.
SP-003	Avtorizacija uporabnika	Sistem preveri, ali ima uporabnik pravico izvajati zahtevan ukrep (npr. dostop, urejanje, nadzor, delegiranje) znotraj svojega konteksta.	Zahtevano vedno, kadar je zahtevana avtentikacija	Brez izjeme velja: če je funkcionalnost avtentificirana, mora biti tudi avtorizirana.
SP-004	Mehko brisanje zapisov	Vsi podatki, ki se "brišejo" v okviru funkcionalnosti, se zgolj označijo kot logično izbrisani – fizično ostanejo v podatkovni bazi.	Vedno, kadar funkcionalnost vsebuje operacijo "brisanja"	V podatkovnem modelu mora obstajati atribut deleted, valid_to, status ipd.; brisanje je reverzibilno.

2.3.2 SKL-VS: Sklop varnostna shema

2.3.2.1 SKL-VS-F-000: Vpis v sistem z uporabo uporabniškega imena in gesla

2.3.2.1.1 Opis funkcionalnosti

Funkcionalnost omogoča uporabniku prijavo v informacijski sistem POKK z uporabo kombinacije **uporabniškega imena in gesla**, z dodatno podporo za **dvostopenjsko avtentikacijo (2FA)** in **preverjanjem soglasij po prijavi**.

Postopek vključuje naslednje korake:

1. Preverjanje obstoja računa:

Če uporabniško ime ne obstaja, sistem vrne generično sporočilo o napačnih prijavnih podatkih.

2. Preverjanje statusa računa:

Sistem preveri, ali je račun:

- aktiven,
- zaklenjen (začasno),
- trajno onemogočen.

Uporabnik dobi ustrezno sporočilo z razlogom in (če je zaklep začasen) tudi s časom odklepa.

3. Validacija gesla:

Geslo se ob prijavi zgošči z enosmernim algoritmom (s soljo) in primerja s hranjeno vrednostjo.

4. Preverjanje 2FA (če je aktiviran):

- Če ima uporabnik aktivirano 2FA, se sproži dodatni varnostni korak:
 - **TOTP:** uporabnik vnese trenutno generirano kodo.
 - **SMS:** uporabnik prejme potrditveno kodo na svoj mobilni telefon.
- Napačen vnos kode pomeni neuspešen poskus prijave in se zabeleži.

5. Preverjanje soglasij:

Po uspešni avtentikaciji (in 2FA, če je aktiviran) sistem preveri, ali so vsa **obvezna soglasja** podana:

- Če soglasja manjkajo ali obstaja nova verzija soglasja, se uporabniku prikaže modal za podajo odločitve.
- Nadaljevanje je omogočeno šele po potrditvi vseh zahtev, v skladu s funkcionalnostjo ~1900.

6. Dodelitev seje in preusmeritev:

Ob uspešni potrditvi soglasij se uporabniku dodeli sejni piškotek (Secure, HttpOnly) in preusmeritev v ~200: **Izbira zastopništva**.

7. Revizijsko beleženje:

Vsak korak prijave, uspešen ali neuspešen, se zabeleži skupaj z IP naslovom, User-Agent, časom, tipom prijave in rezultatom.

2.3.2.1.2 Predpogoji

- Uporabniški račun obstaja in ni zaklenjen ali onemogočen.
- Vpisano geslo ustreza hranjeni zgoščeni vrednosti.
- Če je aktiviran 2FA, mora uporabnik uspešno opraviti dodatno preverjanje.
- Uporabnik mora imeti podana vsa obvezna soglasja v trenutni veljavni verziji.
- Revizijski sistem in sejni mehanizem morata biti dosegljiva.

2.3.2.1.3 Merila uspešnosti

- Postopek prijave (vključno z 2FA in preverjanjem soglasij) se izvede v ≤ 1500 ms.
- Napačni podatki, napačna 2FA koda ali manjkajoča soglasja so jasno sporočeni uporabniku.
- Ob uspešni prijavi se sejni identifikator dodeli in uporabnik se preusmeri v ~200.
- Vsi dogodki so zabeleženi z natančnimi metapodatki.
- Manjkajoča soglasja se zahtevajo enkratno na vsako spremembo verzije.

2.3.2.1.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~000	Vpis v sistem	Osnovna prijava z uporabniškim imenom in geslom.	Uporabniški račun obstaja, geslo nastavljeno.	Avtentikacija uspešna, prehod v ~200.
~001	Preverjanje statusa računa	Preveri ali je račun aktiven, začasno zaklenjen ali trajno	Uporabniško ime obstaja.	Sistem poda ustrezno informacijo ali nadaljuje

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
		onemogočen.		prijava.
~002	Validacija gesla	Geslo se zgošči in primerja s hranjeno vrednostjo.	Vpisano geslo.	Ustreden algoritem, rezultat natančen.
~003	Preverjanje aktivacije 2FA	Če je 2FA aktiviran, sistem sproži dodatni korak preverjanja.	2FA je omogočen na profilu.	Uporabnik je pozvan k vnosu kode (TOTP ali SMS).
~004	Vnos in preverjanje 2FA kode	Uporabnik vnese potrditveno kodo in jo sistem preveri.	Uporabnik ima aktiven 2FA način.	Vnos ustreza sistemski kodi, prijava se nadaljuje.
~005	Dodelitev seje in preusmeritev	Ob uspešni prijavi in podanih soglasjih se dodeli piškotek in sproži preusmeritev na izbiro zastopništva.	Vsi prejšnji koraki uspešno zaključeni.	Seja aktivna, uporabnik preusmerjen.
~006	Beleženje prijavnih dogodkov	Sistem beleži vse prijave poskuse z vsemi metapodatki.	OpenSearch in varnostni logging sta aktivna.	Vsi dogodki zabeleženi, dostopni po klasifikaciji (uspeh, IP, User-Agent).
~007	Preverjanje soglasij po prijavi	Sistem po avtentikaciji preveri, ali so podana vsa veljavna obvezna soglasja.	Modul za upravljanje soglasij aktiven.	Če soglasij ni ali obstaja nova verzija, uporabnik mora potrditi, sicer nadaljevanje ni mogoče.
~008	Preusmeritev na aplikacijo po končanem postopku prijave	Sistem po avtentikaciji in avtorizacijo uporabnika preusmeri na ciljno aplikacijo.	Naslov ciljne aplikacije je ob začetku postopka podan (returnUrl, onSuccessUrl, onCancelUrl...).	Ob končanem postopku se uporabnik preusmeri na željeno aplikacijo.

2.3.2.1.5 Diagram primerov uporabe

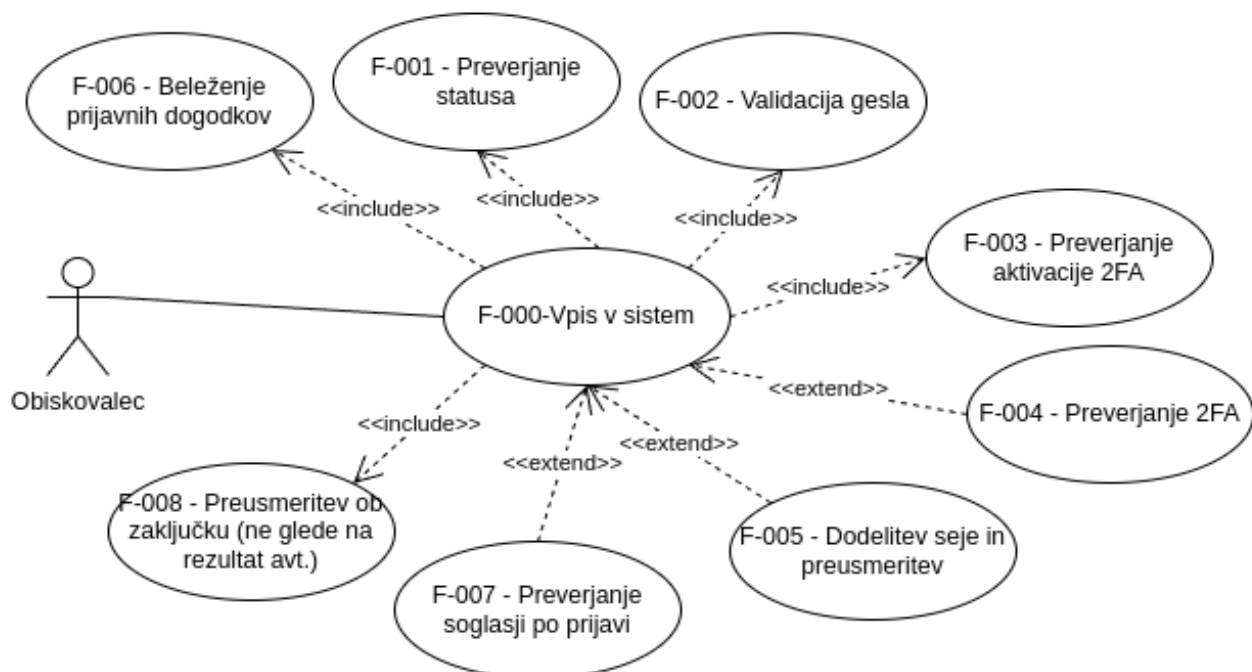


Figure 1: Diagram primerov uporabe za SKL-VS-F-000.

2.3.2.1.6 Diagram poteka

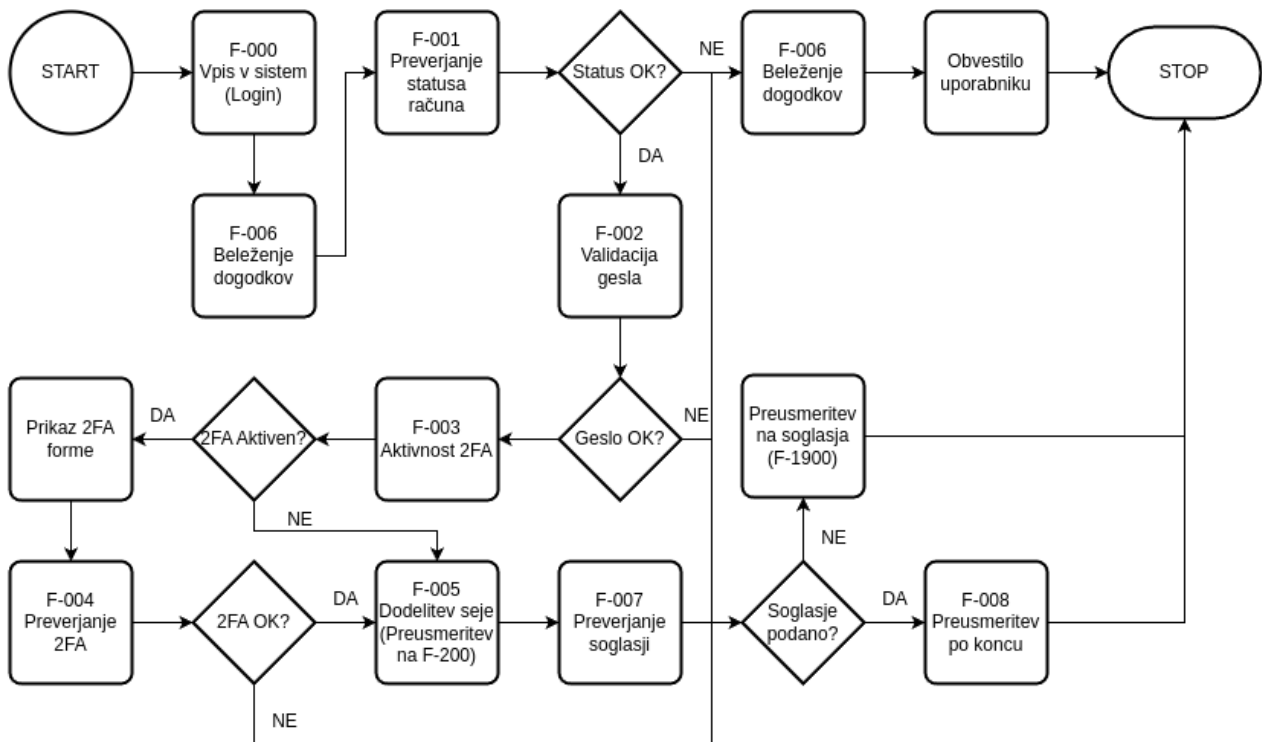


Diagram 2: Diagram poteka prijavnne logike.

2.3.2.2 SKL-VS-F-100: Vpis v sistem z žetonom SSO v shemi federativne prijave

2.3.2.2.1 Opis funkcionalnosti

Funkcionalnost omogoča uporabniku prijavo v informacijski sistem POKK z uporabo **SSO žetona**, izdanega s strani zaupanja vrednega **zunanjega identitetnega ponudnika (IdP)** v okviru **federativne prijavnne sheme**. Sistem POKK deluje kot **SP (Service Provider)** in popolnoma zaupa izidu avtentikacije, izvedene s strani IdP.

Postopek prijave vključuje naslednje korake:

1. Validacija žetona:

- Preveri se veljavnost, izdajatelj (issuer), podpis, časovna veljavnost in nedvoumnost identitete.
- Sistem ne izvaja 2FA lokalno – predpostavlja se, da je varnostna politika izvajana na ravni IdP.

2. Iskanje uporabnika v sistemu:

- Če uporabnik že obstaja, se z njim poveže aktivna seja.
- Če uporabnik še ne obstaja, sistem na podlagi atributov iz žetona samodejno kreira nov uporabniški račun (glej ~102).

3. Preverjanje soglasij:

- Po uspešni avtentikaciji sistem preveri, ali so vsa **obvezna soglasja** (glej ~1900) podana.

- Če soglasja manjkajo ali obstaja nova verzija, se uporabniku prikaže modal za potrditev pogojev.
- Brez potrditve soglasij nadaljevanje ni mogoče.

4. Dodelitev seje in preusmeritev:

- Uporabniku se dodeli sejni identifikator, ki ga sistem označi kot »avtentificiran«.
- Nato sledi preusmeritev v funkcionalnost **~200: Izbira zastopništva**, kjer se izvede avtorizacija.

5. Revizijsko beleženje:

- Vsi dogodki so zabeleženi, vključno z izdajateljem žetona, rezultatom validacije, identifikatorjem uporabnika, IP naslovom, User-Agent itd.

2.3.2.2.2 Predpogoji

- Sistem mora biti registriran kot SP pri zaupanja vrednem IdP.
- Prejeti žeton mora biti kriptografsko veljaven (podpisan, nepreklican, časovno veljaven).
- Atributi žetona morajo omogočati enolično identifikacijo uporabnika.
- Sistem mora imeti omogočeno preverjanje soglasij po prijavi.

2.3.2.2.3 Merila uspešnosti

- Žeton se validira v ≤ 200 ms.
- Obstoječi uporabnik je povezan brez nepotrebne zamude.
- Novi uporabnik je kreiran avtomatsko z ustrezno revizijsko sledjo.
- Soglasja se preverijo takoj po avtentikaciji; uporabnik je jasno obveščen, če katero manjka.
- Uporabnik se preusmeri v izbor zastopništva samo, če so vsi pogoji (vključno s soglasji) izpolnjeni.
- Vsi dogodki so popolno revizijsko sledeni.

2.3.2.2.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~100	Prijava z žetonom SSO	Sistem sprejme in validira žeton, izdan s strani federativnega identitetnega ponudnika.	Sistem registriran kot SP, veljaven žeton.	Prijava uspešna v ≤ 200 ms, uporabnik povezan ali ustvarjen.
~101	Validacija žetona	Preveri se podpis, veljavnost, issuer, časovni okvir.	Ključni IdP so znani in aktivni.	Žeton je validiran in nepreklican.
F-001	Preverjanje statusa	Preveri ali je račun aktiven, začasno zaklenjen ali trajno onemogočen.	Uporabniški račun za ta SSO je najden.	Sistem poda ustrezno informacijo ali nadaljuje prijavo.
~102	Samodejna registracija uporabnika	Če uporabnik še ne obstaja, se kreira profil na podlagi atributov iz žetona.	Žeton vsebuje vse zahtevane podatke.	Račun se ustvari brez vmesnega potrjevanja, soglasja pa se zahtevajo kasneje.

2.3.2.2.5 Diagram primerov uporabe

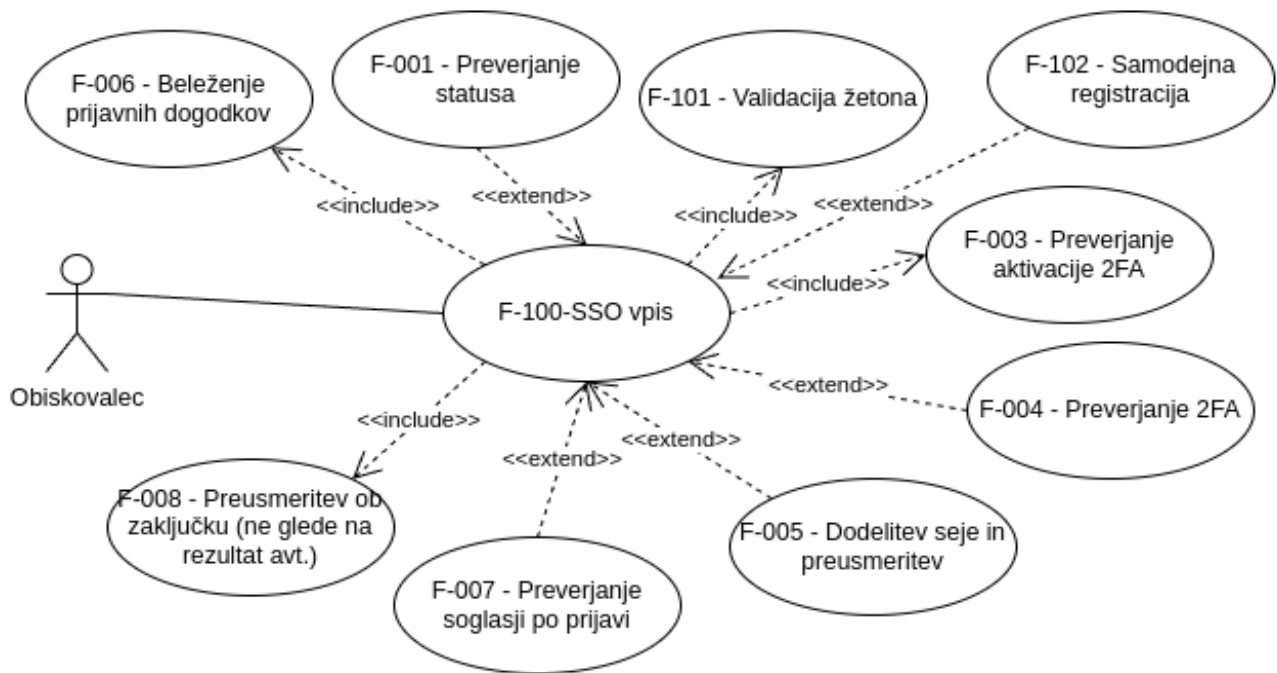


Diagram 3: Diagram primerov uporabe za SKL-VS-F-100.

2.3.2.2.6 Diagram poteka

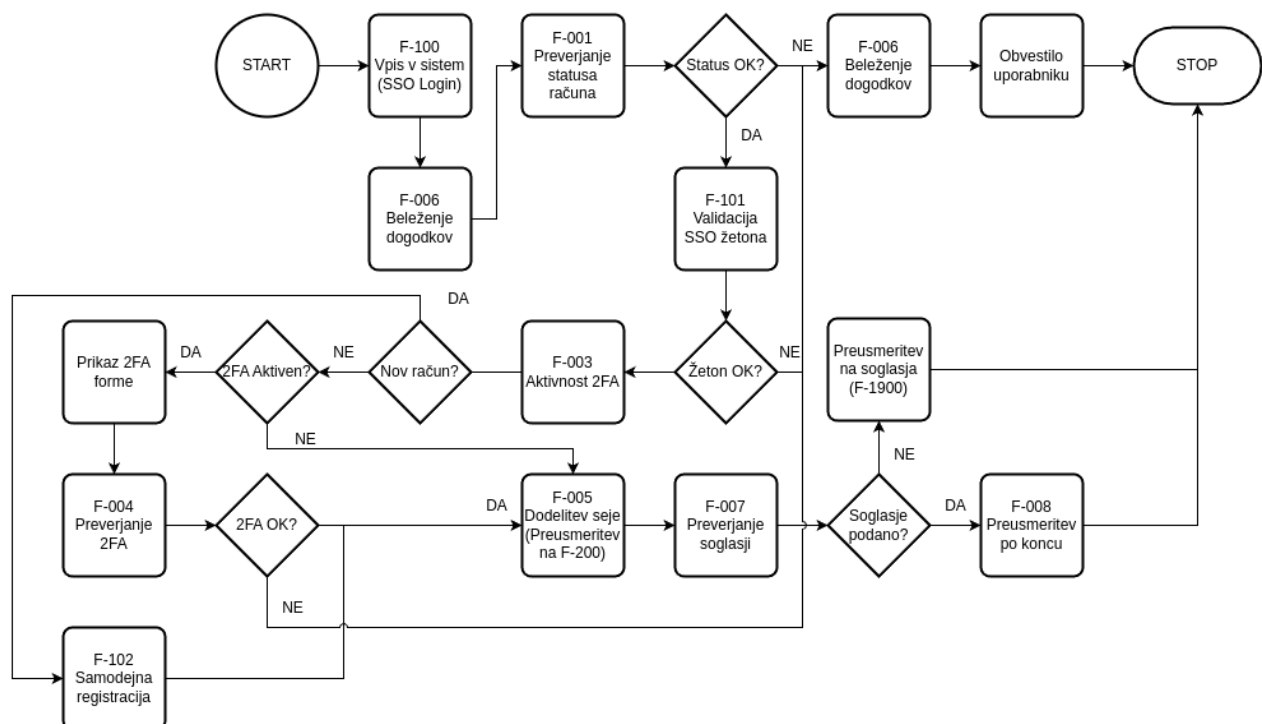


Diagram 4: Diagram poteka pri prijavi z SSO žetonom.

2.3.2.3 SKL-VS-F-200: Izbira zastopništva

2.3.2.3.1 Opis funkcionalnosti

Po uspešni avtentikaciji se uporabnik preusmeri na namensko stran za izbor zastopništva. Na tej

strani sistem uporabniku prikaže seznam subjektov (npr. rejec, rejska organizacija, institucija), ki jih je pooblaščen zastopati. Vsako zastopništvo predstavlja kontekst, v okviru katerega bo uporabnik izvajal operacije v sistemu.

Zastopništva so definirana v podatkovni zbirki in se periodično osvežujejo s pomočjo namenskih sinhronizacijskih servisov iz zunanjih virov (npr. CRG, rejske organizacije).

Če uporabnik nima nobenega dodeljenega zastopništva, ga sistem o tem obvesti in prikaže navodila za vzpostavitev zastopniškega razmerja – običajno z usmeritvijo k kontaktu pooblaščen osebe v instituciji.

Uporabnik lahko v vsakem trenutku:

- **izbere zastopništvo**, kar pomeni začetek avtorizacijske seje z dodeljenimi pravicami, vezanimi na izbran subjekt (zastopniška pravila, omejitve dostopa, pristojnosti),
- **prekine prijavo**, kar ima za posledico takojšnjo invalidacijo avtentikacijskega žetona in uničenje seje – uporabnik ni več prijavljen v sistem.

Po potrditvi zastopništva se izvede avtorizacija in uporabnik je preusmerjen na vnaprej podani naslov `onSuccessURL`, ali na privzeto začetno stran sistema.

2.3.2.3.2 Predpogoji

- Uporabnik mora biti uspešno avtentificiran (ima veljaven avtentikacijski žeton).
- Baza zastopništev mora biti dosegljiva in osvežena s pomočjo sinhronizacijskih servisov.
- Uporabniški vmesnik za izbor mora biti na voljo.
- Storitve za avtorizacijo mora biti dosegljiva.

2.3.2.3.3 Merila uspešnosti

- Seznam zastopništev se prikaže v ≤ 300 ms za >95 % uporabnikov.
- Če uporabnik nima zastopništev, se prikaže sporočilo s kontaktnimi informacijami.
- Po izbiri zastopništva se dodelijo ustrezne pravice in začne nova avtorizacijska seja.
- Prekinitev postopka prijave prekliče vse sejne podatke in deaktivira avtentikacijski žeton.
- Sistem omogoča sledljivost vsake izbire (kdo je koga zastopal, kdaj, s katerimi pravicami).

2.3.2.3.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~200	Izbira zastopništva	Uporabnik izbere subjekt, v imenu katerega bo uporabljal sistem. Izbira določa pravice, ki se mu dodelijo v nadaljevanju seje.	Uporabnik je avtentificiran.	Zastopništva se prikažejo, avtorizacija uspešno dodeli pravice, seja se nadaljuje ali pravilno prekine.
~201	Prikaz in izbor subjektov	Seznam subjektov (rejec, organizacija ipd.), ki jih uporabnik lahko zastopa, se dinamično prikaže glede na podatke iz podatkovne baze.	Sinhronizacija z virom zastopništev je bila uspešna.	Seznam se naloži hitro, uporabnik lahko izbere le aktivna in veljavna zastopništva.

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~202	Uporabnik nima zastopništva	Sistem zazna, da uporabnik nima dodeljenih zastopništev, in mu prikaže jasna navodila za kontakt s pooblaščen osebo.	Uporabnik nima vpisov v relacijski tabeli zastopništev.	Obvestilo vsebuje kontaktne podatke; preprečen nadaljnji vstop v sistem brez veljavnega zastopništva.
~300	Prekinitev postopka prijave (odjava)	Uporabnik lahko kadarkoli prekine prijavo; Izvede se popolna odjava skladno z F-300.	Uporabnik je v fazi izbire zastopništva.	Seja je nemudoma deaktivirana, nadaljnji dostop do sistema brez ponovne prijave ni mogoč.
~204	Avtorizacija po izbiri zastopništva	Po izbiri zastopništva sistem dodeli pravice vezane na izbran kontekst. Nadaljnja uporaba sistema je vezana na ta kontekst in njegove omejitve.	Izbrano je veljavno in aktivno zastopništvo.	Dodeljene pravice so pravilno prenesene iz podatkovne baze, potrjene v seji in zabeležene v revizijski sled.

2.3.2.3.5 Diagram primerov uporabe

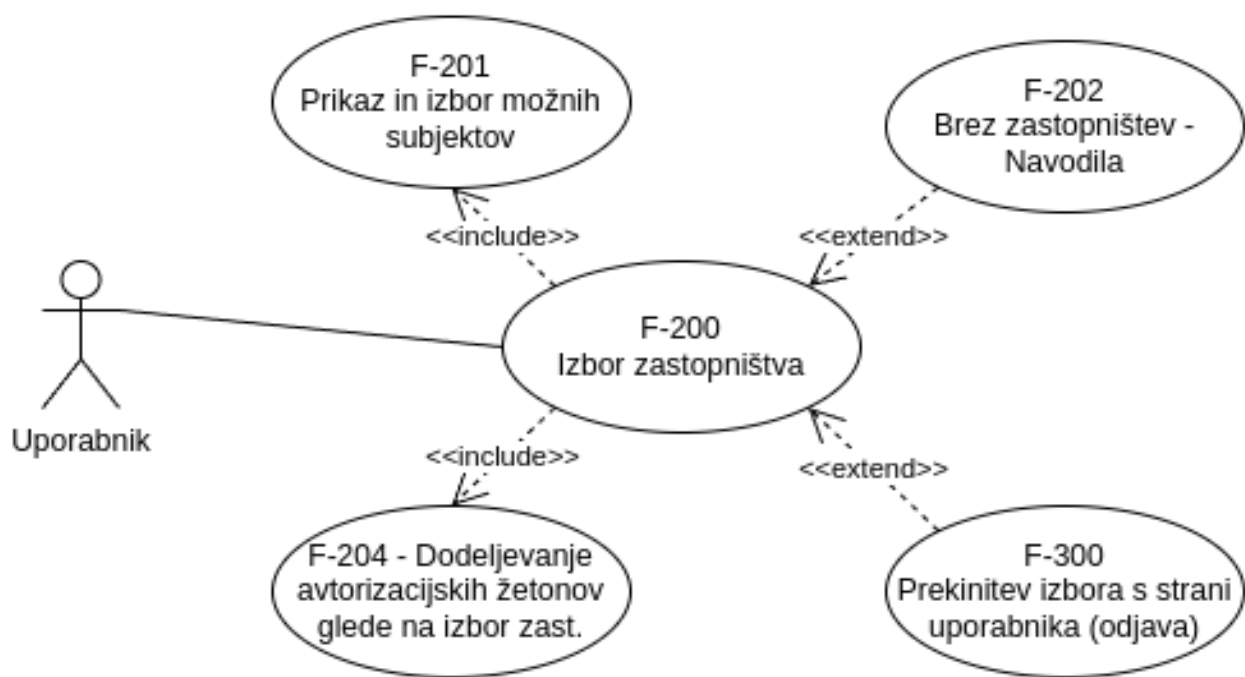


Diagram 5: Diagram primerov uporabe za SKL-VS-F-200.

2.3.2.4 SKL-VS-F-300: Odjava uporabnika iz sistema

2.3.2.4.1 Opis funkcionalnosti

Funkcionalnost omogoča varno in popolno odjavo uporabnika iz informacijskega sistema POKK. Uporabnik se lahko kadarkoli med uporabo sistema odjavi bodisi ročno (s klikom na gumb »Odjava«) bodisi avtomatsko (npr. ob poteku seje).

Ob odjavi sistem izvede naslednje korake:

1. **Prekinitev avtorizacijske seje:** Seja se zaključi, vsi podatki o uporabnikovem kontekstu (zastopništvo, pravice) se počistijo iz aktivnega pomnilnika in morebitnega začasnega

skladišča (npr. Redis).

2. **Invalidacija avtentikacijskega žetona:** Avtentikacijski žeton (npr. JWT ali sejni ID) se označi kot neveljaven, tako da ponovna uporaba ni več mogoča.
3. **Preusmeritev uporabnika:** Uporabnik je preusmerjen na privzeto odjavno stran ali ustrezen naslov `onLogoutRedirectUrl`, če je bil podan.
4. **Beleženje dogodka:** Odjava se beleži v revizijski sistem z identifikacijo uporabnika, časom in razlogom (ročno, samodejno, varnostno).
5. **SSO kontekst (če obstaja):** Če je bila seja vzpostavljena prek federativne prijave, se izvede odjava tudi iz IdP sistema (npr. prek end-session endpoint-a).

2.3.2.4.2 Predpogoji

- Uporabnik mora imeti aktivno avtentikacijsko sejo.
- Sistem za upravljanje sej in revizijsko beleženje mora biti dosegljiv.
- V primeru federativne prijave mora biti konfiguriran mehanizem za SSO-odjavo.

2.3.2.4.3 Merila uspešnosti

- Seja se zaključi v ≤ 200 ms.
- Avtentikacijski žeton ni več veljaven (sistem zavrne vse nadaljnje klice).
- Uporabnik je preusmerjen na potrjeno odjavno stran.
- Vsi dogodki odjave so revizijsko zabeleženi z ustreznim statusom.
- V primeru federativne prijave je sprožena tudi globalna odjava iz IdP.

2.3.2.4.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~300	Odjava uporabnika iz sistema	Uporabnik se varno odjavi iz sistema. Seja se konča, žeton se deaktivira, uporabnik je preusmerjen na odjavno stran.	Aktivna seja	Seja je ukinjena, žeton neveljaven, revizijski zapis uspešen.
~301	Zaključek avtorizacijske seje	Seja, povezana z uporabniškim kontekstom, se uniči.	Uporabnik je prijavljen	Vsi podatki seje so odstranjeni iz pomnilnika in začasnega skladišča.
~302	Invalidacija avtentikacijskega žetona	Sejni žeton se označi kot neveljaven in ni več sprejet v nadaljnjih klicih.	Avtentikacijski žeton obstaja	Poskusi z neveljavnim žetonom sprožijo napako 401 (Unauthorized).
~303	Preusmeritev na odjavno stran	Uporabnik je po odjavi preusmerjen na vnaprej določen <code>onLogoutRedirectUrl</code> ali sistemsko privzeto stran.	Seja zaključena	Preusmeritev je izvedena v <200 ms.
~304	Revizijsko beleženje odjave	Dogodek odjave se zabeleži s časom, uporabniškim ID-jem in	Mehanizem za logiranje je	Zapis je dostopen v OpenSearch-u v

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
		razlogom (ročno, avtomatsko, varnostno).	dosegljiv	roku 2 s po odjavi.
~305	SSO-odjava iz IdP	Sproži se globalna odjava iz IdP (npr. OpenID Connect end-session).	Seja zaključena	SSO-odjava je potrjena s statusom 200 iz IdP sistema.

2.3.2.4.5 Diagram primerov uporabe

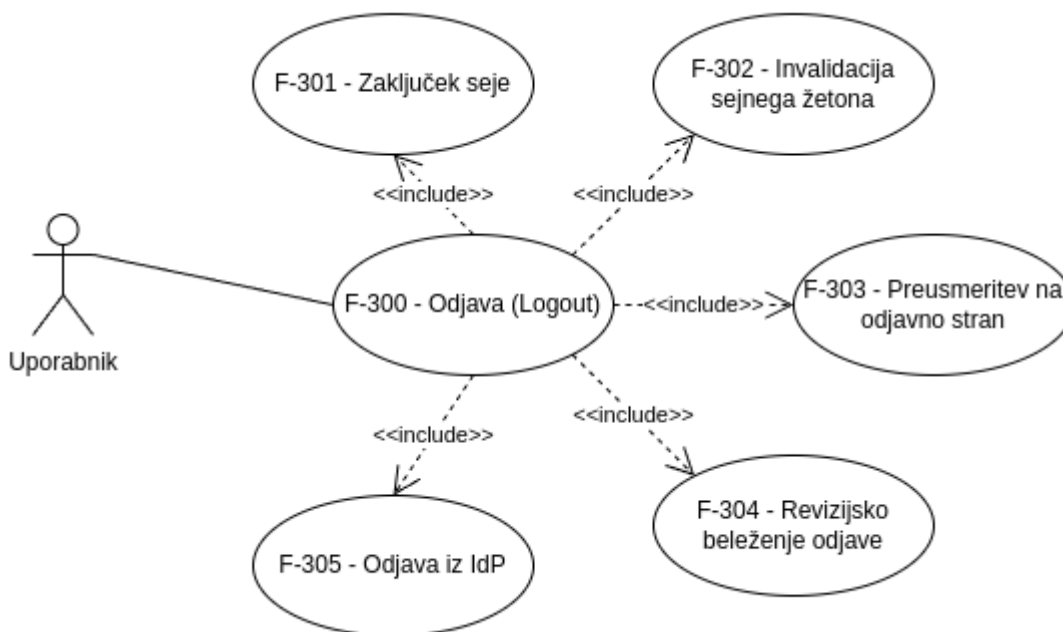


Diagram 6: Diagram primerov uporabe za SKL-VS-F-300.

2.3.2.5 SKL-VS-F-400: Samodejno podaljševanje seje

2.3.2.5.1 Opis funkcionalnosti

Funkcionalnost omogoča samodejno podaljševanje avtorizacijske seje uporabnika, dokler ta aktivno uporablja informacijski sistem POKK. Sistem sledi vsakemu upravičenemu zahtevku (npr. zahteva na API, klik v uporabniškem vmesniku) in na tej osnovi podaljša veljavnost uporabniške seje.

Seja ima sicer definirano **najdaljšo dopustno trajanje** (npr. 16 ur), ki velja ne glede na uporabnikovo aktivnost – s čimer se zmanjša tveganje zlorabe ukradenih avtentikacijskih žetonov (npr. v primeru kraje piškotka ali SSO seje).

Parametri glede trajanja in mehanizmov podaljševanja se berejo iz zunanje konfiguracijske datoteke, ki omogoča ločeno nastavljanje časovnih omejitev za:

- lokalne prijave (uporabniško ime/geslo),
- federativne prijave (SSO žeton),
- specifične skupine uporabnikov ali aplikacijske vloge.

Po poteku največje dovoljene dolžine seje se uporabniku seja prekine in sproži standardni mehanizem odjave (~300).

2.3.2.5.2 Predpogoji

- Uporabnik mora biti aktivno prijavljen.

- Sejni mehanizem mora omogočati osveževanje sejnega časovnika na podlagi aktivnosti.
- Konfiguracijska datoteka s parametri trajanja se mora uspešno naložiti ob zagonu sistema.
- Čas strežnika mora biti sinhroniziran (npr. prek NTP).

2.3.2.5.3 Merila uspešnosti

- Seja se podaljšuje ob vsaki upravičeni uporabniški aktivnosti.
- Sistem ne preseže konfigurirane največje veljavnosti seje.
- Po poteku časa uporabnik prejme informacijo o izteku seje in se mora ponovno prijaviti.
- Vse prekinitve zaradi poteka veljavnosti se zabeležijo v revizijski sistem.
- V primeru napačne konfiguracije se sistem odzove z varnim privzetim časom (fail-safe mehanizem).

2.3.2.5.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~400	Samodejno podaljševanje seje	Uporabniku se seja ob vsakem zahtevku avtomatsko podaljša, dokler je aktiven. Po poteku največje dolžine seje se uporabnik samodejno odjavi.	Aktivna prijava; sejni mehanizem omogoča nadzor nad trajanjem.	Seja se podaljšuje z vsako aktivnostjo, a ne preseže konfiguriranega maksimuma.
~401	Konfiguracija parametrov trajanja	Sistem ob zagonu naloži parametre za maksimalno dolžino seje, ločeno po vrstah prijave.	Konfiguracijska datoteka je na voljo.	Uporabljene vrednosti so sledljive in v primeru napake se uporabi privzeta (npr. 8 ur).
~402	Samodejna prekinitve po izteku	Če seja doseže končni čas trajanja, se uporabnik samodejno odjavi in obvesti o izteku.	Presežen je čas trajanja seje.	Seja se zaključi, uporabniku se prikaže obvestilo in sproži odjava ~300.
~403	Revizijsko beleženje prekinitve seje	Sistem zabeleži razlog prekinitve (nedejavnost, pretek časa, izrecna odjava, napaka).	Revizijsko beleženje je aktivno.	Zapis je dostopen v OpenSearch in vsebuje uporabnika, čas, razlog, IP in UserAgent.
~404	Fail-safe mehanizem za napake v nastavitvah	Če konfiguracija ni dosegljiva ali vsebuje neveljavne podatke, se uporabi varna privzeta vrednost trajanja seje.	Konfiguracija ni naložena ali vsebuje napako.	Seja se ne podaljšuje neskončno – sistem jo zaključi po 4–8 urah (glede na varne privzete vrednosti).

2.3.2.5.5 Diagram primerov uporabe

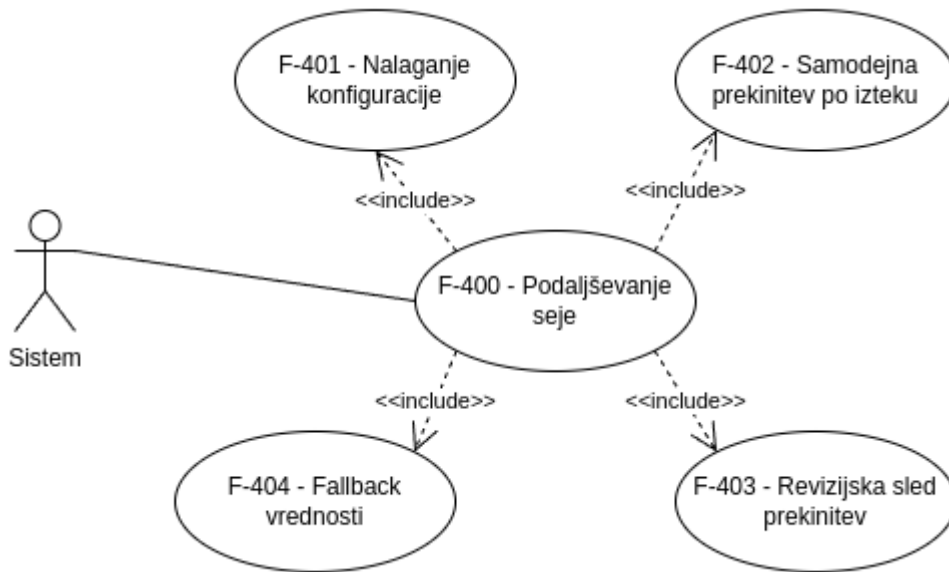


Diagram 7: Diagram primerov uporabe za SKL-VS-F-400.

2.3.2.6 SKL-VS-F-500: Izpis osebnih podatkov (GDPR)

2.3.2.6.1 Opis funkcionalnosti

Funkcionalnost omogoča uporabniku, da skladno z določili Splošne uredbe o varstvu podatkov (GDPR – člen 15 in člen 20) pridobi izvoz vseh svojih osebnih podatkov, ki jih o njem hrani informacijski sistem POKK.

Postopek zahteve za izpis osebnih podatkov poteka na naslednji način:

1. **Zahteva uporabnika:** Uporabnik v uporabniškem vmesniku odda zahtevo za izvoz svojih podatkov.
2. **Asinhrona obdelava:** Sistem sproži asinhroni proces (job), ki pridobi vse osebne podatke uporabnika iz vseh relevantnih domen sistema (osebni profil, prijave, revizijske sledi, zastopništva ipd.).
3. **Generacija datoteke:** Podatki se zberejo in zapakirajo v zaščiteno ZIP datoteko, ki vsebuje jasno strukturirane podatke v standardizirani obliki (npr. JSON, XML, CSV).
4. **Obveščanje uporabnika:**
 - Na elektronski naslov uporabnika se pošlje enkratna povezava (SSO link) za prenos datoteke.
 - Na uporabnikov mobilni telefon se pošlje potrditvena koda, ki služi kot geslo za odklepanje datoteke.
5. **Varovanje vsebine:** Datoteka je zaščitena z geslom in na voljo omejen čas (npr. 72 ur). Prenos poteka prek TLS povezave.

Ta funkcionalnost omogoča sledljivost, nadzor dostopov in skladnost z vsemi zahtevami GDPR, vključno s pravico do dostopa, prenosljivosti in zaščite osebnih podatkov.

2.3.2.6.2 Predpogoji

- Uporabnik mora biti uspešno avtenticiran in avtoriziran.
- Sistem mora imeti konfigurirane povezave do vseh domen z osebnimi podatki.
- Vzpostavljen mora biti mehanizem za asinhrono izvajanje zahtevkov (job scheduler).
- Na voljo mora biti konfiguriran poštni strežnik in storitev za pošiljanje SMS sporočil.
- Telefonska številka in e-poštni naslov uporabnika morata biti preverjena.

2.3.2.6.3 Merila uspešnosti

- Zahteva za izvoz se sprejme in potrdi v ≤ 1 s.
- Asinhroni proces pripravi datoteko v ≤ 10 minutah (odvisno od obsega podatkov).
- SSO povezava in geslo sta poslana ločeno, varno in v realnem času.
- ZIP datoteka vsebuje vse relevantne podatke, strukturirane po področjih.
- Datoteka je dostopna omejen čas, prenos pa je zaščiten.
- Vsaka zahteva in dostop do podatkov se beleži v revizijski sled.

2.3.2.6.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~500	Izvoz osebnih podatkov (GDPR)	Uporabnik zahteva izvoz vseh osebnih podatkov, ki jih sistem hrani o njem, skladno z GDPR.	Avtentikacija, avtorizacija, vnos kontakta uporabnika.	Podatki so celovito zbrani in varno dostavljeni v ustrezni obliki.
~501	Oddaja zahteve za izvoz podatkov	Uporabnik sproži zahtevek prek uporabniškega vmesnika.	Prijavljen uporabnik, aktivna seja.	Zahteva potrjena in sprožen asinhroni proces.
~502	Asinhrona priprava podatkov	Sistem v ozadju zbere vse podatke uporabnika iz različnih domen sistema.	Konfiguriran podatkovni dostop, job sistem deluje.	Podatki so konsistentni, popolni in pripravljeni v strukturirani obliki.
~503	Kreiranje zaščitene ZIP datoteke	Podatki se zapakirajo v zaščiten ZIP datoteko z ločenim geslom.	Uporabnikov kontaktni kanal je preverjen.	ZIP je zaščiten, standardiziran, vsebina validna.
~504	Obveščanje uporabnika	Na email se pošlje SSO povezava, na telefon pa ločeno geslo za dostop do datoteke.	Preverjen elektronski naslov in mobilna številka.	Email in SMS sta uspešno poslani, povezava je enkratna in časovno omejena.
~505	Prenos datoteke	Uporabnik prenese datoteko glede na avtorizaicjski žeton iz poslane elektronske pošte.	Datoteka je kreirana.	Datoteka se prenese.

2.3.2.6.5 Diagram primerov uporabe

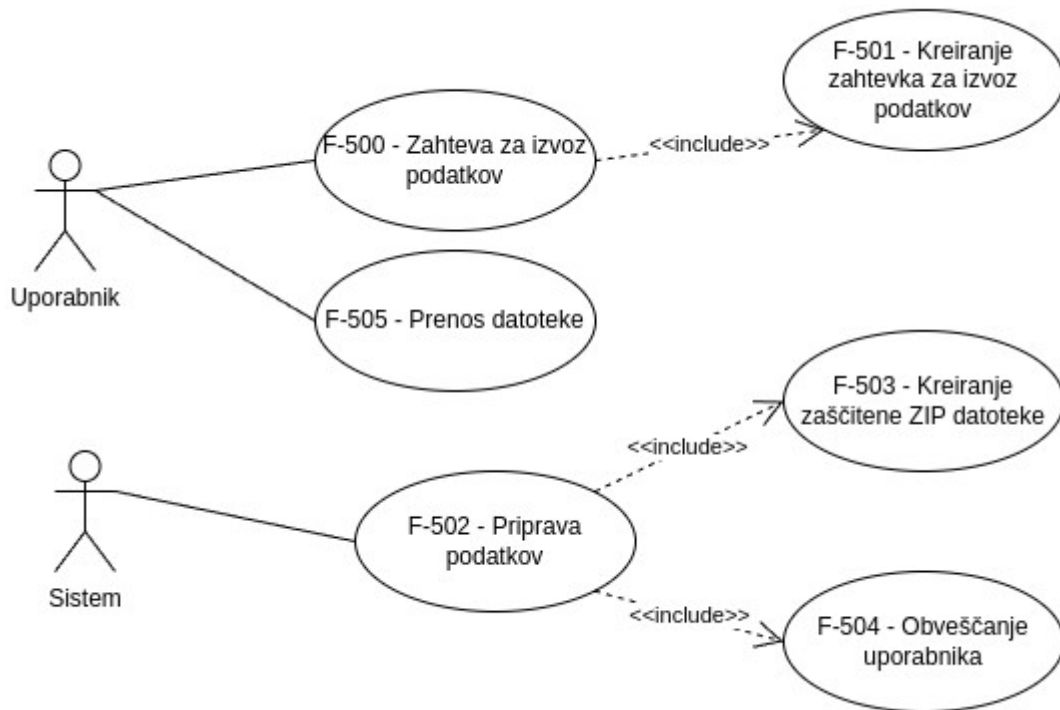


Diagram 8: Diagram primerov uporabe za SKL-VS-F-500.

2.3.2.7 SKL-VS-F-600: Urejanje osebnih podatkov uporabnika

2.3.2.7.1 Opis funkcionalnosti

Funkcionalnost omogoča uporabniku, da samostojno ureja tiste osebne podatke, ki niso pridobljeni iz uradnih registrov oziroma prek integracij z državnimi sistemi. Ti podatki predstavljajo **dodatne informacije**, ki jih uporabnik prostovoljno vnese in so lahko v pomoč pri nadaljnjih procesih znotraj informacijskega sistema POKK (npr. preferenčni kontaktni kanal, akademski naziv, interno označevanje uporabnika, vloga znotraj organizacije ipd.).

Sistem loči med:

- **urejevalnimi polji** – ki jih uporabnik lahko posodablja,
- **integriranimi podatki** – ki so informativni in se ne morejo spreminjati znotraj aplikacije (npr. davčna številka, naslov stalnega prebivališča).

Vsaka sprememba dodatnih podatkov:

- se validira glede na vnaprej določena pravila (npr. obveznost polj, dolžina, tip),
- se shrani kot **nova verzija** zapisa z časovnim žigom,
- omogoča **revizijski vpogled v zgodovino** vseh izvedenih sprememb (kdo, kdaj, kaj se je spremenilo).

Podatki se hranijo skladno z načelom **verzioniranja**, kar omogoča transparentnost, auditabilnost in skladnost z določili GDPR.

2.3.2.7.2 Predpogoji

- Uporabnik mora biti avtenticiran in avtoriziran za urejanje lastnega profila.
- Podatkovni model mora ločevati med integriranimi in urejevalnimi polji.

- Sistem mora imeti omogočeno verzioniranje in revizijsko beleženje.
- UI mora zagotavljati jasen prikaz, kaj je urejevalno in kaj ne.

2.3.2.7.3 Merila uspešnosti

- Sistem uporabniku prikaže zgolj tista polja, ki jih lahko ureja.
- Vsaka sprememba sproži novo verzijo podatkov z unikatnim ID-jem in časovno oznako.
- Pretekle verzije so dostopne prek zgodovine sprememb.
- Nedovoljen poskus urejanja integriranih podatkov se zavrne in zabeleži.
- Vse spremembe so revizijsko sledene.

2.3.2.7.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~600	Urejanje osebnih podatkov uporabnika	Uporabnik lahko ureja podatke, ki niso pridobljeni po uradni dolžnosti, npr. alternativni kontakt, akademski naziv, interna opomba ipd.	Uporabnik ima aktiven profil; sistem ločuje urejevalna in integrirana polja.	Urejanje omejeno na določena polja, validacija podatkov, spremembe se beležijo.
~601	Ločevanje urejevalnih in integriranih polj	Uporabniku se prikažejo le tista polja, ki jih lahko spreminja. Ostali podatki so prikazani samo za branje.	Metapodatkovna oznaka za vsak atribut (editable vs. readonly).	Uporabnik nima možnosti manipulacije z integriranimi podatki.
~602	Validacija sprememb	Vnesene vrednosti se validirajo glede na poslovna pravila (tip, dolžina, format).	Definirana pravila za vsako polje.	Napačno vneseni podatki se zavrnejo, uporabnik prejme povratno informacijo.
~603	Verzioniranje uporabniških podatkov	Vsaka sprememba ustvari novo verzijo podatkovnega zapisa.	Verzijska podpora v podatkovni plasti.	Vsaka verzija ima časovni žig in avtorja spremembe.
~604	Revizijski vpogled	Uporabnik in administratorji lahko dostopajo do zgodovine sprememb.	Aktivno beleženje dogodkov.	Prikaz zgodovine urejanj, možnost izpisa starejših verzij v berljivi obliki.

2.3.2.7.5 Diagram primerov uporabe

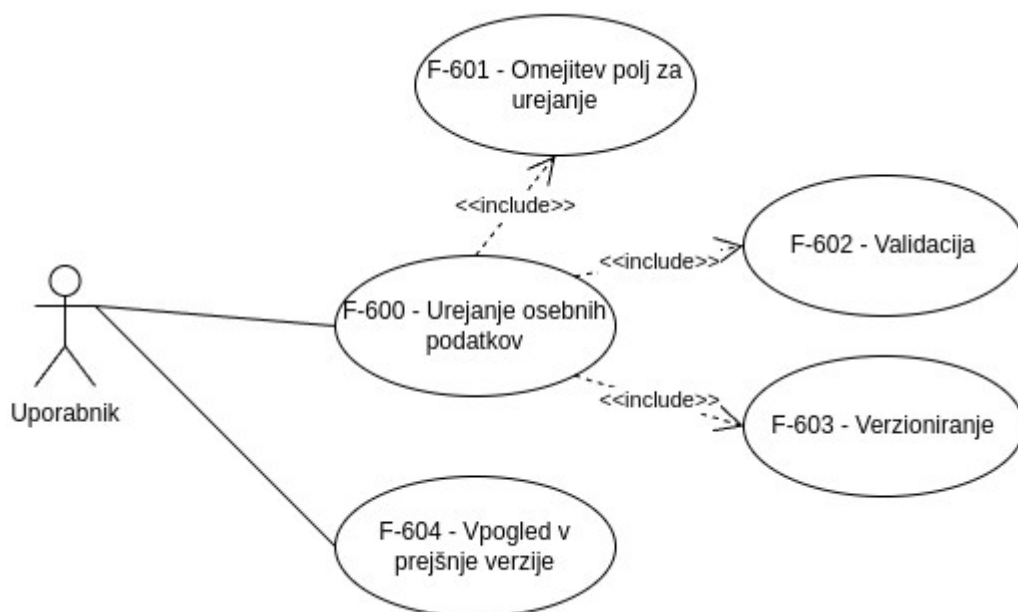


Diagram 9: Diagram primerov uporabe za SKL-VS-F-600.

2.3.2.8 SKL-VS-F-700: Urejanje podatkov zastopanega subjekta

2.3.2.8.1 Opis funkcionalnosti

Funkcionalnost omogoča uporabniku urejanje **dodatnih podatkov** vezanih na subjekt, ki ga zastopa v okviru aktivne avtorizacijske seje. Tako kot pri osebnih podatkih uporabnika tudi tukaj velja, da lahko ureja **le tista polja**, ki niso določena kot **integrirana** oziroma **uradno pridobljena** iz zunanjih evidenc (npr. CRG, AKTRP, poslovni register).

Urejevalna polja lahko vključujejo:

- dodatne kontaktne podatke (alternativni e-poštni naslov, telefonska številka),
- interna pojasnila ali oznake za posamezne module (npr. vrsta subjektove aktivnosti, status v sistemu),
- nastavitve obveščanja in preferenc.

Vsaka sprememba nad podatki zastopanega subjekta:

- se **validira** glede na ustrezna pravila,
- ustvari **ново verzijo podatkov**, ki omogoča popoln zgodovinski vpogled,
- se **revizijsko zabeleži**, vključno z identifikacijo uporabnika, ki je spremembo izvedel, časom in vrednostjo spremembe.

2.3.2.8.2 Predpogoji

- Uporabnik mora biti avtenticiran, avtoriziran in mora imeti izbran subjekt (zastopništvo).
- Sistem mora poznati, katera polja so urejevalna in katera integrirana.
- Verzijsko beleženje mora biti omogočeno tudi na ravni entitet subjekta.
- Vloga uporabnika mora imeti dovoljenje za urejanje.

2.3.2.8.3 Merila uspešnosti

- Prikazane so le tiste informacije, ki so urejevalne narave.
- Vsaka sprememba sproži novo verzijo zapisa.
- Nedovoljeni poskusi sprememb integriranih podatkov so zavrjeni.
- Zgodovina vseh sprememb je dostopna in transparentna.
- Vse spremembe so sledene v revizijskem sistemu.

2.3.2.8.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~700	Urejanje podatkov zastopanega subjekta	Uporabnik lahko ureja dodatne podatke o subjektu, ki ga zastopa, če za to obstaja pravica in polja niso uradno integrirana.	Avtentikacija, avtorizacija, veljavno zastopništvo.	Urejanje dovoljeno le nad definiranimi polji, vse spremembe verzionirane in revizijsko sledene.
~701	Identifikacija urejevalnih polj	Sistem zna ločiti med polji, ki jih je dovoljeno urejati, in tistimi, ki so integrirana iz uradnih virov.	Metapodatki o izvoru podatka.	Polja so jasno označena kot nespremenljiva ali urejevalna; UI preprečuje nedovoljene spremembe.
~702	Validacija sprememb	Vnesene vrednosti se validirajo glede na poslovna pravila (tip, dolžina, format).	Definirana pravila za vsako polje.	Napačno vneseni podatki se zavrnejo, uporabnik prejme povratno informacijo.
~703	Verzioniranje subjektnih podatkov	Vsaka sprememba nad dodatnimi podatki subjekta povzroči nastanek nove verzije z ustreznim časovnim žigom in avtorjem spremembe.	Verzijska podpora v podatkovnem modelu.	Pretekle verzije so na voljo za vpogled ali revizijo.
~704	Revizijsko beleženje sprememb subjekta	Sistem beleži vse spremembe podatkov subjekta, vključno z uporabnikom, časom, izvorno in spremenjeno vrednostjo.	Aktivno revizijsko sledenje sprememb.	Vsi vnosi dostopni prek OpenSearch, z ustrežno klasifikacijo dogodka.

2.3.2.8.5 Diagram primerov uporabe

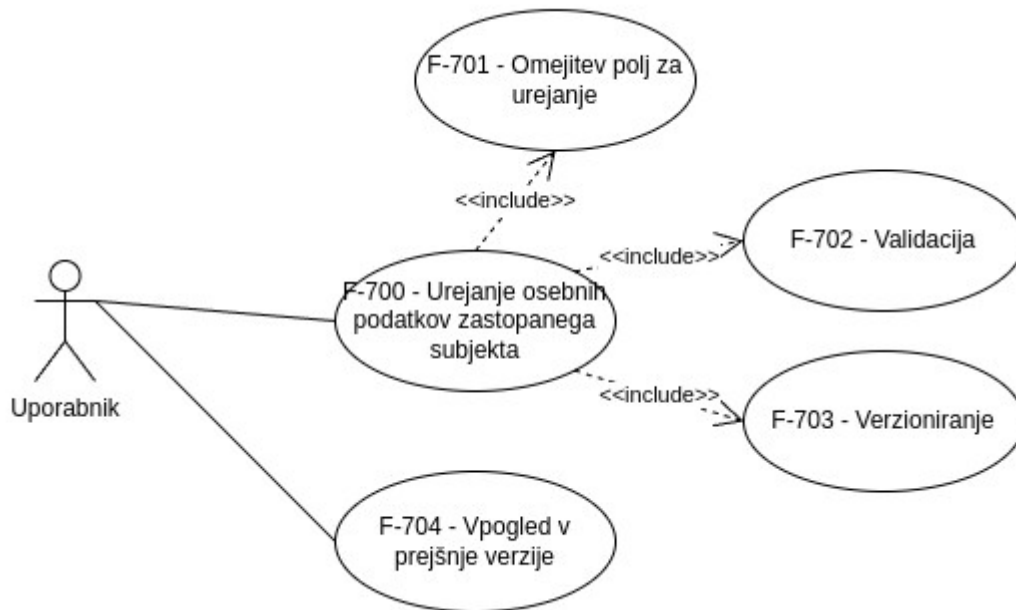


Diagram 10: Diagram primerov uporabe za SKL-VS-F-700.

2.3.2.9 SKL-VS-F-900: Nastavitve uporabniškega profila

2.3.2.9.1 Opis funkcionalnosti

Funkcionalnost omogoča uporabniku nastavitve in prilagoditev izbranih aspektov lastnega profila za uporabo sistema POKK. Te nastavitve ne sodijo med osebne podatke in niso pridobljene iz uradnih evidenc, temveč predstavljajo **operativne preference**, ki vplivajo na delovanje uporabniškega vmesnika in uporabniško izkušnjo.

Med nastavitve profila sodijo zlasti:

- izbrani jezik uporabniškega vmesnika,
- privzeti način obveščanja (email, SMS, obvestila znotraj sistema),
- nastavitve prikaza podatkov (npr. število vrstic na stran, privzeti filter),
- dovoljenja za prikaz vsebin po domenah (npr. sklopi, vrstni red zavihkov),
- aktivacija dvostopenjske avtentikacije, če je omogočena.

Nastavitve so **specifične za uporabnika**, neodvisne od zastopanega subjekta, in veljajo v vseh kontekstih prijave. Vsaka sprememba se zabeleži v sistemu in je povezljiva z uporabnikom ter časom spremembe.

2.3.2.9.2 Predpogoji

- Uporabnik mora biti prijavljen in avtoriziran.
- Podatkovna shema mora imeti vzpostavljen mehanizem za hranjenje uporabniških nastavitvev.
- Uporabniški vmesnik mora omogočati dostop in upravljanje nastavitvev.
- Sistem mora podpirati večjezičnost in dinamično prilagajanje UI-ja.

2.3.2.9.3 Merila uspešnosti

- Sprememba posamezne nastavitve se shrani in začne veljati brez ponovne prijave.
- Vse nastavitve so validirane pred shranjevanjem.
- V primeru napake se ohrani prejšnje stanje.
- Uporabnik lahko povrne nastavitve na privzete vrednosti.
- Vse spremembe se beležijo v revizijsko sled.

2.3.2.9.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~900	Nastavitve uporabniškega profila	Uporabnik ureja lastne preference, ki vplivajo na uporabniški vmesnik in način interakcije s sistemom.	Avtentikacija, aktivna seja	Spremembe so takoj uveljavljene, shranjene in sledljive.
~901	Nastavitev jezika	Uporabnik izbere želeni jezik uporabniškega vmesnika.	Sistem podpira večjezičnost	UI se samodejno prevede, izbira se shrani kot trajna preferenca.
~902	Način obveščanja	Uporabnik določi, ali želi sistemska obvestila prejemati po emailu, SMS ali zgolj znotraj aplikacije.	Aktivirani kontaktni podatki	Sistem uporablja izbran kanal za vsa nadaljnja sistemska obvestila.
~903	Prikaz podatkov	Uporabnik nastavi privzeti način prikaza (vrstice na stran, sortiranje, prikaz filtrov).	Na voljo ustrezne možnosti v uporabniškem vmesniku	Izbrani prikaz velja pri vseh modulih, kjer je prikaz parametriziran.
~905	Aktivacija 2FA (če je omogočeno)	Uporabnik vklopi ali izklopi dvostopenjsko prijavo (2FA), če jo sistem podpira (npr. TOTP, SMS koda).	2FA je aktivirano na ravni sistema	Sistem vodi uporabnika skozi postopek aktivacije, z varnostnim preverjanjem.
~906	Povrnitev privzetih nastavitev	Uporabnik lahko ponastavi vse nastavitve na sistemske privzete vrednosti.	Uporabniški profil obstaja	Po potrditvi se vse nastavitve povrnejo, obstoječe vrednosti pa se arhivirajo (revizijsko).

2.3.2.9.5 Diagram primerov uporabe

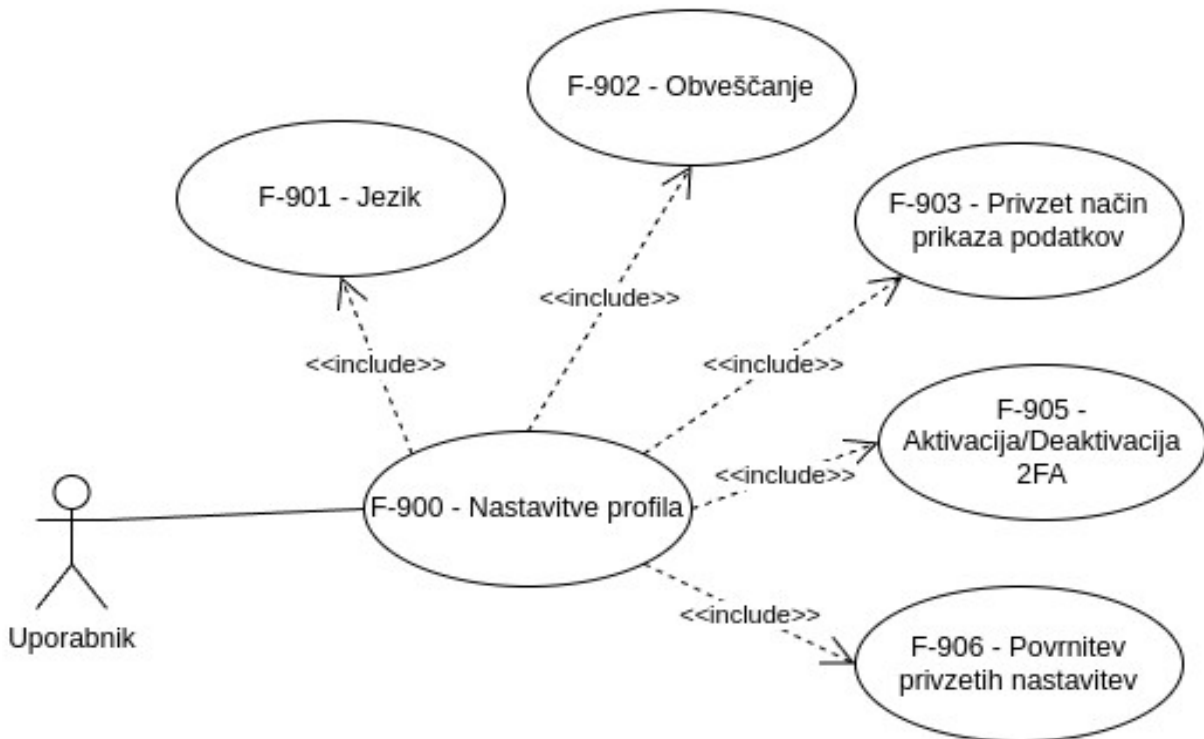


Diagram 11: Diagram primerov uporabe za SKL-VS-F-900.

2.3.2.10 SKL-VS-F-1100: Delegiranje pooblastil v okviru zastopništva

2.3.2.10.1 Opis funkcionalnosti

Funkcionalnost omogoča, da uporabnik, ki zastopa določen subjekt, dodeli **pravico zastopanja istega subjekta** tudi drugim uporabnikom. Gre za **delegacijo zastopništva**, ki mora biti zamejena na obseg pravic, ki jih delegator sam razpolaga – kar pomeni, da ni mogoče podeliti višjih pravic od lastnih.

Postopek vključuje:

1. **Identifikacijo prejemnika zastopništva:** Uporabnik vnese elektronski naslov drugega uporabnika.
2. **Določitev obsega pravic:** Uporabnik izbere nivo pravic ali konkretne funkcionalne sklope, ki jih želi dodeliti.
3. **Validacija pooblastil:** Sistem preveri, ali uporabnik res razpolaga z vsemi izbranimi pravicami ter ali je predmet zastopanja veljaven.
4. **Sprožitev postopka dodeljevanja:** S potrditvijo se sproži funkcionalnost **SKL-VS-F-1200: Dodeljevanje pravic zastopništva uporabniku**, ki obdeluje tehnično implementacijo prenosa pravic in pošiljanje obvestil prejemniku.

Dodeljevanje se beleži v revizijski sistem, prejemnik pa mora zastopništvo aktivirati ob naslednji prijavi (ali prek posebne potrditvene povezave, če tako določa sistemska politika).

2.3.2.10.2 Predpogoji

- Uporabnik mora biti avtoriziran za zastopanje subjekta.
- Pravice, ki jih želi dodeliti, morajo biti manjše ali enake njegovim lastnim.
- Prejemnik mora imeti veljaven uporabniški profil ali mora biti omogočeno samodejno ustvarjanje ob potrditvi.
- Konfiguriran mora biti podproces SKL-VS-F-1200.

2.3.2.10.3 Merila uspešnosti

- Sistem prepreči dodeljevanje višjih pravic od dovoljenih.
- Uporabniku je jasno prikazano, katere pravice lahko podeli in komu.
- Po oddaji zahteve je prejemnik ustrezno obveščen (email, sistemsko sporočilo).
- Revizijska sled vsebuje: delegatorja, prejemnika, subjekt, datum, obseg pravic.
- Vse spremembe so reverzibilne v okviru upravljanja dodeljenih pravic.

2.3.2.10.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~1100	Dodeljevanje zastopništev ostalim uporabnikom	Uporabnik delegira pravico zastopanja subjekta drugemu uporabniku, z omejitvijo obsega glede na lastne pravice.	Avtentikacija, avtorizacija, aktivno zastopništvo	Dodeli se le dovoljen obseg pravic, sprožitev SKL-VS-F-1200 uspešna, revizijski zapis prisoten.
~1101	Vnos prejemnika zastopništva	Uporabnik vnese elektronski naslov druge osebe, ki bo prejela pravice do zastopanja.	Vnos e-naslova je obvezen in mora biti veljaven.	Sistem validira email in preveri, ali že obstaja povezljiv uporabniški račun.
~1102	Izbira pravic za delegacijo	Uporabnik izbere nabor pravic ali sklopov, ki jih želi prenesti na prejemnika.	Dovoljene le pravice manjše ali enake obstoječim.	Izbrani obseg je ustrezno validiran in omejen.
~1103	Preverjanje pogojev dodelitve	Sistem preveri upravičenost uporabnika in veljavnost subjekta.	Subjekt mora biti aktiven, uporabnik mora imeti vlogo zastopnika.	Napačne kombinacije ali prekoračitve se zavrnejo s sporočilom o napaki.
~1104	Sprožitev SKL-VS-F-1200	Po potrditvi se izvede funkcionalnost za dejansko dodeljevanje pravic uporabniku.	Komunikacija z modulom SKL-VS-F-1200 omogočena.	Uporabnik je obveščen o uspešni zahtevi, prejemnik prejme vabilo.

2.3.2.10.5 Diagram primerov uporabe

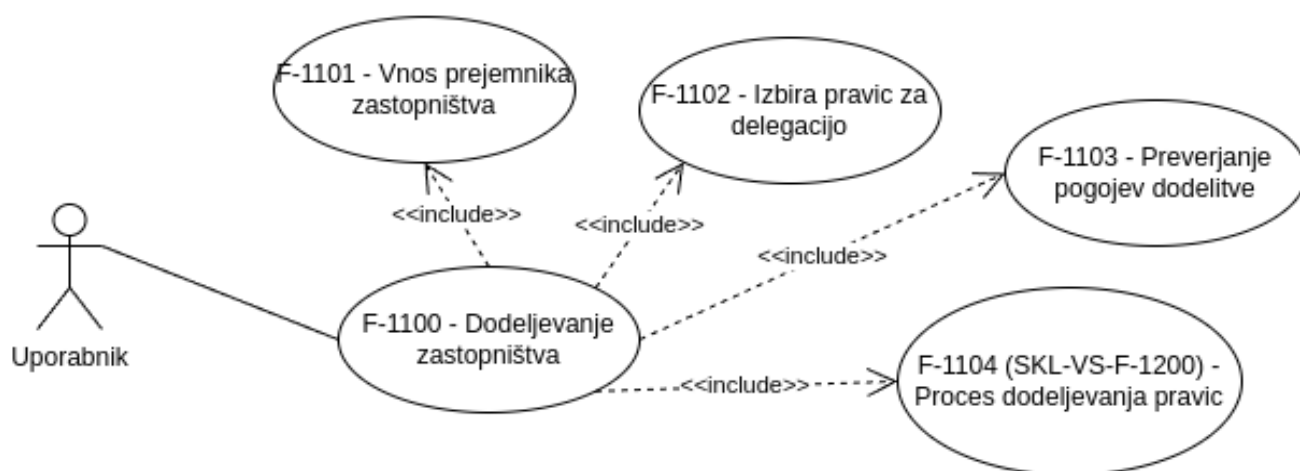


Diagram 12: Diagram primerov uporabe za SKL-VS-F-1100.

2.3.2.11 SKL-VS-F-1200: Dodeljevanje pravic zastopništva uporabniku

2.3.2.11.1 Opis funkcionalnosti

Funkcionalnost omogoča sistemu, da izvede **dejanski prenos zastopniških pravic** drugemu uporabniku na podlagi zahteve, sprožene prek funkcionalnosti **SKL-VS-F-1100**. Gre za centralno procesno točko, ki obdeluje dodeljevanje pravic, generacijo SSO povezav, preverjanje obstoja uporabnika in pripravo komunikacije s prejemnikom.

Sistem izvede naslednje korake:

1. Preverjanje obstoja uporabnika:

- Če uporabnik že obstaja v sistemu, se mu pošlje **SSO povezava** za neposreden dostop do sistema z vnaprej dodeljenim kontekstom zastopništva.
- Če uporabnik še ne obstaja, se mu pošlje **registracijska SSO povezava**, ki ga vodi na stran za ustvarjanje računa. Po uspešni registraciji se aktivira zastopništvo in sproži postopek določitve gesla.

2. Avtorizacijska preverjanja:

- Sistem preveri, ali ima delegator ustrezne pravice za dodelitev pravic.
- Preveri se, ali prejemnik ni že zastopnik z višjimi pravicami za isti subjekt.

3. Tehnična implementacija:

- Zastopniška povezava (relacija) se ustvari v podatkovni bazi, vendar je aktivna šele ob potrditvi prijave (first login).
- Po uspešni avtentikaciji prejemnika se aktivira seja z ustreznim kontekstom in dodeljenimi pravicami.

4. Obveščanje:

- Prejemnik prejme sporočilo z obrazloženo vsebino zastopništva, varnostnim opozorilom in navodili za prijavo ali registracijo.

2.3.2.11.2 Predpogoji

- Uporabnik, ki dodeljuje pravice, mora biti avtoriziran za zastopanje subjekta in imeti ustrezen obseg pooblastil.
- Prejemnikov elektronski naslov mora biti validiran (RFC 5322).
- Sistem mora imeti omogočeno generiranje SSO žetonov in podporo za postopke prve prijave.
- Komunikacijski kanal (email) mora biti dostopen in varen.

2.3.2.11.3 Merila uspešnosti

- SSO povezava je veljavna enkratno in časovno omejena (npr. 72 ur).
- Če uporabnik že obstaja, se po prijavi takoj aktivira zastopništvo in pravice.
- Če uporabnik še ne obstaja, se po registraciji aktivira zastopništvo, določitev gesla pa se izvede prek varnostnega postopka.
- Vse aktivnosti (pošiljanje povezave, registracija, aktivacija pravic) so revizijsko pokrite.
- Sistem prepreči podvajanja ali konfliktne relacije za istega uporabnika.

2.3.2.11.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~1200	Dodeljevanje pravic zastopništva uporabniku	Sistem izvede dodeljevanje zastopništva drugemu uporabniku – z aktivacijo ob prvi prijavi ali registraciji.	Delegator ima ustrezne pravice; prejemnikov email je znan.	Povezava uspešno dostavljena, prijava ali registracija aktivira zastopništvo.
~1201	Preverjanje obstoja uporabnika	Sistem preveri, ali prejemnik že ima uporabniški račun.	Email je veljaven.	Razvejitev med prijavno in registracijsko potjo.
~1202	Pošiljanje SSO povezave	Prejemniku se pošlje povezava za prijavo (če obstaja račun) ali registracijo (če ne obstaja).	Komunikacijski kanal je delujoč.	Povezava je unikatna, časovno omejena, zaščitena pred zlorabo.
~1203	Aktivacija zastopništva po prijavi	Ob uspešni prijavi (ali registraciji) se aktivira vnaprej pripravljena relacija zastopništva.	Avtentikacija prejemnika uspešna.	Pravice so naložene v aktivno sejo, revizijsko zabeležene.
~1204	Ustvarjanje računa in inicialna nastavitve	Če uporabnik ne obstaja, se mu omogoči ustvarjanje računa in določitev gesla (npr. po koraku potrditve SSO).	Uporabnik uspešno opravi postopek registracije.	Račun je aktiven, geslo določeno, zastopništvo aktivirano.
~1205	Revizijsko beleženje dodelitve in aktivacije	Vse dodelitve, potrditve, zavrnitve in aktivacije se zabeležijo z vsemi ključnimi parametri (kdo, komu, kaj, kdaj, kako).	OpenSearch aktiven, logiranje omogočeno.	Revizijska evidenca je popolna, preverljiva in neizbrisljiva.

2.3.2.11.5 Diagram primerov uporabe

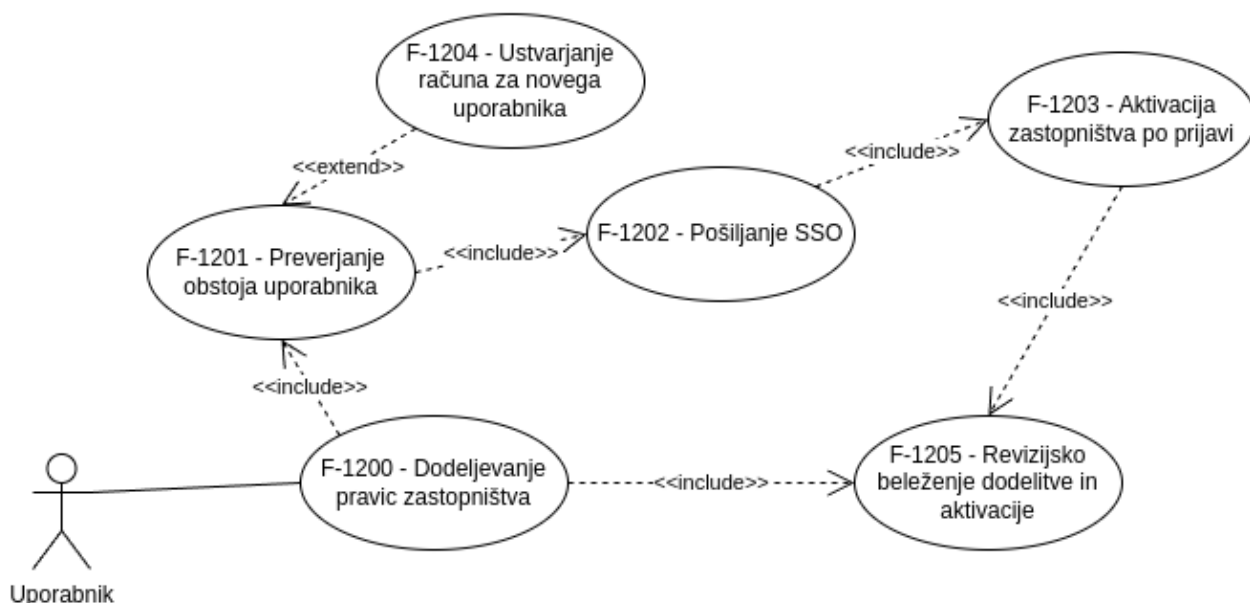


Diagram 13: Diagram primerov uporabe za SKL-VS-F-1200.

2.3.2.12 SKL-VS-F-1300: Upravljanje že dodeljenih zastopništev

2.3.2.12.1 Opis funkcionalnosti

Funkcionalnost omogoča uporabniku, da pregleda in upravlja vsa zastopništva, ki jih je dodelil drugim uporabnikom za subjekt, ki ga zastopa. Upravljanje vključuje:

- pregled trenutnih zastopnikov in njihovih pravic,
- **spreminjanje obsega pravic** (znotraj meja lastnega pooblastila),
- **odvzem zastopništva** posameznemu uporabniku,
- **časovno omejevanje** dodeljenih zastopništev (npr. časovna veljavnost ali preklic z datumom).

Sistem zagotavlja, da uporabnik ne more:

- povzdigniti obsega pravic nad svojimi,
- upravljati zastopništvo, ki jih sam ni podelil,
- odvzeti pravic uporabnikom, ki so jih dodelili drugi uporabniki (razen administratorja).

Vse spremembe se beležijo v revizijski sistem, skupaj z razlogom spremembe, uporabnikom, ki je spremembo izvedel, in časovnim žigom.

2.3.2.12.2 Predpogoji

- Uporabnik mora imeti veljavno zastopništvo z ustreznim obsegom pravic (npr. upravljanje delegacij).
- Obstoječe relacije zastopanj morajo biti vzpostavljene in aktivne.
- UI mora omogočati ustrezne kontrole (omejitve, potrditve, filtri).

2.3.2.12.3 Merila uspešnosti

- Prikaz vseh aktivnih in poteklih zastopniških relacij je hiter in pregleden.
- Spremembe pravic se validirajo glede na obseg lastnega pooblastila.
- Odvzemi so takojšnji in učinkoviti – uporabnik izgubi dostop v isti seji.
- Vse spremembe so revizijsko evidentirane in časovno označene.
- Sistem preprečuje nenamerne odvzeme (potrditveno okno, dvofaktorska avtorizacija, če je omogočena).

2.3.2.12.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~1300	Upravljanje že dodeljenih zastopništev	Uporabnik upravlja zastopništva, ki jih je dodelil drugim za subjekt, ki ga trenutno zastopa.	Veljavno aktivno zastopništvo, pravica za delegiranje	Prikaz, sprememba, odvzem in časovna omejitev so podprti in revizijsko sledeni.
~1301	Prikaz aktivnih zastopništev	Seznam uporabnikov z dodeljenimi pravicami, obseg pravic, datum podelitve, dodeljevalec, veljavnost.	Sistem dostopa do tabele zastopništev	Seznam se naloži v ≤ 500 ms, podatki so filtrabilni in sortirani.
~1302	Sprememba obsega pravic	Uporabnik lahko zniža ali spremeni obseg pravic znotraj lastnih pooblastil.	Delegator ima dovoljene pravice	Nova nastavitev začne veljati takoj, aktivna seja prejemnika je prekinjena ali osvežena.
~1303	Odvzem zastopništva	Uporabnik lahko odvzame že dodeljeno zastopništvo in s tem onemogoči nadaljnjo uporabo sistema s tem subjektom.	Dodelitev je bila izvedena s strani istega uporabnika	Prejemnik takoj izgubi dostop; sprememba je zabeležena.
~1304	Časovna omejitev zastopanja	Dodelitev se lahko omeji z datumom poteka veljavnosti. Po tem datumu se pravice samodejno prekinejo.	Sistem podpira atribut »velja_do«	Datum poteka se upošteva avtomatsko, seja se ob naslednji prijavi ne aktivira.
~1305	Revizijsko sledenje sprememb	Vse spremembe nad zastopniškimi relacijami se beležijo z razlogom, časom, uporabnikom in vsebino spremembe (iz starega v novo stanje).	OpenSearch delujoč, revizijska struktura na voljo	Vsi dogodki dostopni prek iskanja, klasificirani po vrsti spremembe.

2.3.2.12.5 Diagram primerov uporabe

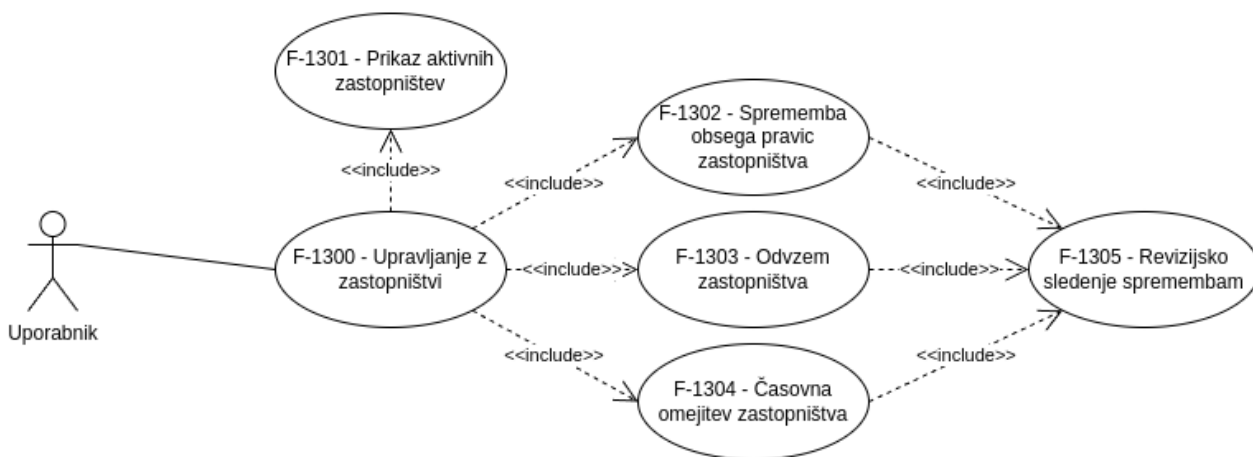


Diagram 14: Diagram primerov uporabe za SKL-VS-F-1300.

2.3.2.13 SKL-VS-F-1400: Vklop dvostopenjske avtentikacije (2FA)

2.3.2.13.1 Opis funkcionalnosti

Funkcionalnost omogoča uporabniku, da v svojem profilu aktivira **dvostopenjsko avtentikacijo (2FA)**, ki zagotavlja dodatno raven varnosti pri prijavi v sistem POKK. Mehanizem temelji na kombinaciji osnovne avtentikacije (uporabniško ime in geslo ali SSO žeton) in dodatne komponente, ki jo generira ali prejme uporabnik:

- **SMS koda**, poslana na preverjeno mobilno številko,
- **TOTP (Time-based One-Time Password)**, generirana prek aplikacije, združljive z RFC 6238 (npr. Google Authenticator, Authy).

Ob vklopu 2FA sistem izvede naslednje:

1. **Preverjanje kontaktnih podatkov:** Uporabnik mora imeti preverjeno mobilno številko ali registrirano TOTP napravo.
2. **Izbira metode:** Uporabnik izbere želeni način – SMS ali aplikacijo.
3. **Postopek aktivacije:**
 - V primeru TOTP sistem prikaže QR kodo in enkratno kodo za povezavo.
 - V primeru SMS se na telefon pošlje potrditvena koda.
4. **Potrditev vklopa:** Uporabnik vnese preveritveno kodo, s čimer se 2FA aktivira.
5. **Revizijsko beleženje:** Aktivacija in vse spremembe povezane z 2FA se beležijo.

2FA se uporablja pri vseh nadaljnjih prijavah. Uporabnik lahko 2FA kadarkoli deaktivira, če za to ima ustrezna pooblastila, ali pa to zahteva preko podpore (odvisno od sistemske politike).

2.3.2.13.2 Predpogoji

- Uporabnik mora biti prijavljen in imeti dostop do svojega profila.
- Za SMS mora biti mobilna številka predhodno preverjena.
- Za TOTP mora imeti uporabnik ustrezno napravo z avtentikatorsko aplikacijo.

- Sistem mora imeti aktiviran modul za 2FA.

2.3.2.13.3 Merila uspešnosti

- Uporabnik uspešno zaključi postopek aktivacije v največ 2 minutah.
- Napake (napačna koda, neveljavna številka) so jasno prikazane.
- 2FA velja ob vsaki prijavi in je preverljiva.
- Revizijska sled vsebuje čas vklopa, metodo, uporabnika in status.
- Deaktivacija je mogoča le z dodatno potrditvijo.

2.3.2.13.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~1400	Vklop dvostopenjske avtentikacije	Uporabnik v profilu vključi 2FA za povečano varnost prijave.	Prijavljen uporabnik, preverjena kontaktna točka (telefon ali TOTP naprava).	2FA aktivirana v največ 2 minutah, sprememba zabeležena, prijava zahteva dodatno preverjanje.
~1401	Izbira metode (SMS / TOTP)	Uporabnik izbere način 2FA – SMS ali aplikacijo (TOTP).	Kontaktne podatke so potrjeni.	Pravilna izbira omogoča nadaljevanje aktivacije.
~1402	Aktivacija TOTP	Sistem prikaže QR kodo za povezavo z avtentikatorsko aplikacijo.	Aplikacija podpira RFC 6238.	Koda je pravilno generirana in validirana.
~1403	Aktivacija preko SMS	Sistem pošlje preveritveno kodo na registrirano številko.	Telefonska številka preverjena.	Koda dostavljena in potrjena, 2FA aktivirana.
~1404	Potrditev in shranjevanje nastavitvev	Uporabnik vnese preveritveno kodo in potrdi aktivacijo.	Uporabnik vnese pravilno kodo.	2FA označena kot aktivna v uporabniškem profilu.
~1405	Revizijsko sledenje 2FA operacij	Vse spremembe nad 2FA (aktivacija, deaktivacija, neuspešni poskusi) se beležijo.	OpenSearch aktiven, logiranje omogočeno.	Dogodki dostopni prek iskanja, z vsemi relevantnimi metapodatki.

2.3.2.13.5

Diagram primerov uporabe

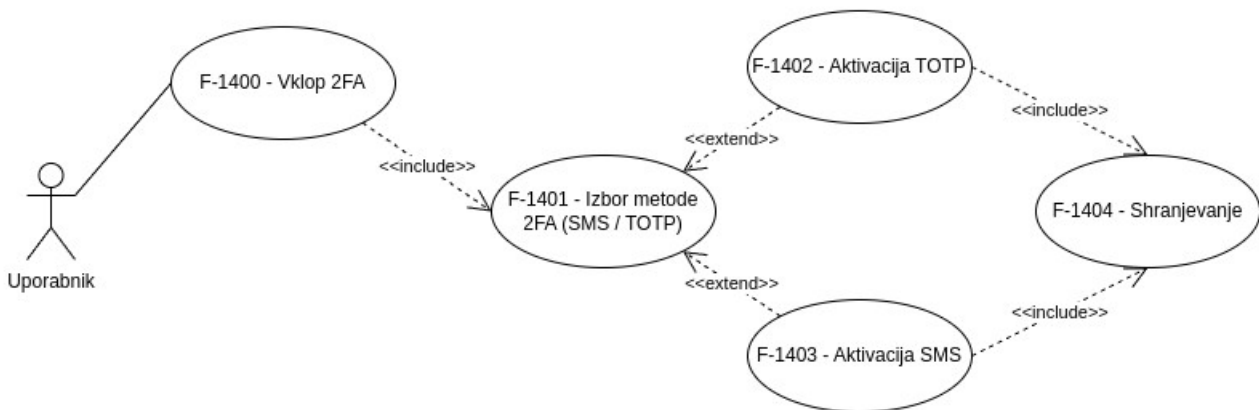


Diagram 15: Diagram primerov uporabe za SKL-VS-F-1400.

2.3.2.14 SKL-VS-F-1500: Preklic vseh aktivnih sej uporabnika

2.3.2.14.1 Opis funkcionalnosti

Funkcionalnost omogoča uporabniku (ali administratorju, kjer je to ustrezno) izvedbo varnostnega ukrepa – **preklic vseh trenutno aktivnih sej uporabnika** znotraj informacijskega sistema POKK. S tem se zagotavlja varnost v primerih, kot so:

- sum zlorabe računa ali kraje žetona,
- izguba nadzora nad sejo v tuji napravi,
- sprememba varnostnih parametrov (geslo, 2FA),
- ročna intervencija s strani uporabnika ali skrbnika.

Preklic zajema:

- vse seje, aktivne prek spletnega vmesnika, mobilnih aplikacij ali API dostopov,
- seje ustvarjene preko federativne prijave (SSO), z ločenim upravljanjem od IdP sistema,
- vse sejne identifikatorje in avtentikacijske žetone.

Po preklicu:

- se vse aktivne seje uničijo,
- morebitne odprte povezave se zavrnejo z napako 401,
- uporabniku se prikaže obvestilo o izteku seje,
- vse aktivnosti se zabeležijo v revizijsko sled.

2.3.2.14.2 Predpogoji

- Uporabnik mora biti prijavljen in imeti potrjen profil (ali pa ima ustrezne pravice za upravljanje s sejami drugih – npr. administrator).
- Sistem mora podpirati centralizirano upravljanje sej (npr. Redis, baza žetonov).
- Mehanizem za preklic mora zajemati vse kontekste in tipe sej.

2.3.2.14.3 Merila uspešnosti

- Preklic se izvede v ≤ 2 sekundah (neodvisno od števila sej).

- Vse seje so nedvoumno prekinjene, nadaljnji zahtevki zavrtnjeni.
- Dogodek je ustrezno zabeležen (uporabnik, IP, čas, razlog).
- UI uporabniku prikaže potrditveno sporočilo.
- V primeru napake se uporabnik opozori in ponudi ponoven poskus.

2.3.2.14.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~1500	Preklic vseh aktivnih sej uporabnika	Uporabnik prekliče vse trenutno aktivne seje svojega računa, z namenom zaščite dostopa.	Uporabnik mora biti avtenticiran.	Preklic se izvede nemudoma, aktivne seje so prekinjene, UI potrdi izvedbo.
~1501	Identifikacija vseh aktivnih sej	Sistem pridobi seznam vseh sej za danega uporabnika iz sistema za upravljanje sej.	Centralni sejni register delujoč (Redis, baza).	Seje zajemajo vse tipe (spletni UI, API, mobilne aplikacije).
~1502	Preklic sej v realnem času	Vse aktivne seje se označijo kot neveljavne in izbrišejo iz registracijskega sistema.	Dovoljenje za preklic.	Vse nadaljnje interakcije s starimi sejami so zavrnjene z napako 401.
~1503	Obvestilo uporabniku	Sistem uporabniku prikaže informacijo, da je bil preklic uspešen in da je potrebna ponovna prijava.	UI komunikacija aktivna.	Obvestilo vključuje datum, čas in napotek k nadaljnji prijavi.
~1504	Revizijsko beleženje operacije	Preklic sej se zabeleži skupaj z razlogom (ročno, varnostno, administrativno), izvorno IP lokacijo in uporabnikom, ki je zahtevek sprožil.	Aktivno beleženje v OpenSearch.	Zapis vsebuje vse ključne metapodatke, dostopen administratorjem in samemu uporabniku.

2.3.2.14.5 Diagram primerov uporabe

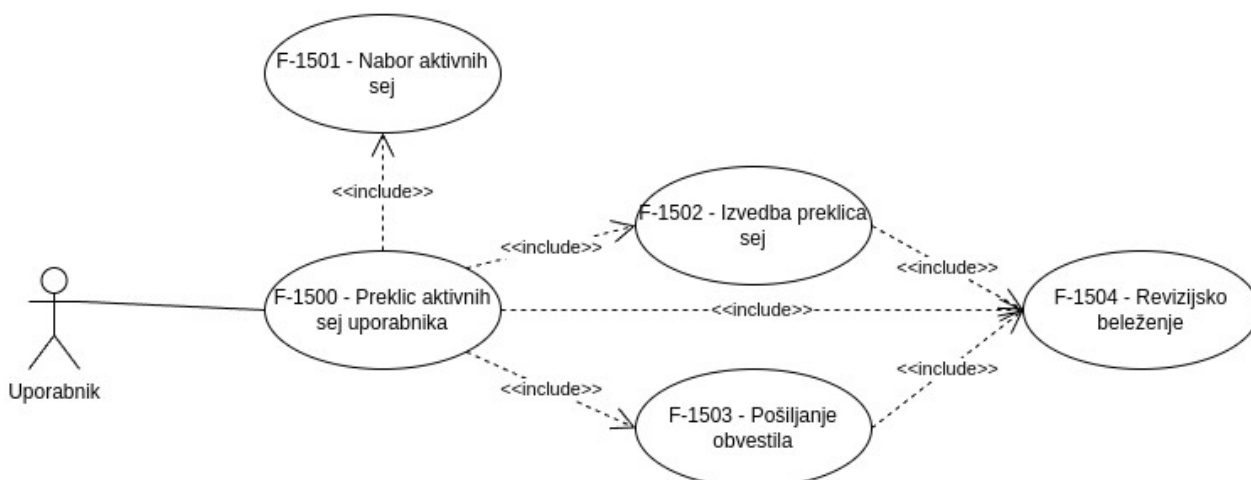


Diagram 16: Diagram primerov uporabe za SKL-VS-F-1500.

2.3.2.15 SKL-VS-F-1600: Dostop do zgodovine prijav in varnostnih dogodkov

2.3.2.15.1 Opis funkcionalnosti

Funkcionalnost omogoča uporabniku **vpogled v lastno zgodovino prijav in varnostno relevantnih dogodkov**, zabeleženih v sistemu. Namen funkcionalnosti je:

- **povečanje transparentnosti,**
- **samonadzor uporabnika nad aktivnostmi,** povezanimi z dostopi do sistema,
- **zaznavanje nepooblaščenih ali nenavadnih prijav.**

Podatki so pridobljeni iz revizijskega sistema (OpenSearch) in vključujejo naslednje informacije:

- datum in čas dogodka,
- tip dogodka (prijava, neuspešna prijava, odjava, preklic seje, aktivacija 2FA ipd.),
- lokacija/IP naslov,
- uporabljena platforma (npr. spletni brskalnik, mobilna aplikacija),
- rezultat dogodka (uspeh/neuspeh),
- vir (ročno, SSO, avtomatsko).

Uporabniški vmesnik omogoča:

- kronološki pregled z možnostjo **filtriranja po tipu dogodka, časovnem obdobju, lokaciji,**
- **označevanje sumljivih dogodkov** (uporabnik jih lahko izrecno označi za revizijo),
- **prikaz metapodatkov** za vsak dogodek (headerji, User-Agent, identifikatorji sej, ipd.).

2.3.2.15.2 Predpogoji

- Uporabnik mora biti prijavljen.
- Revizijski sistem (OpenSearch) mora biti vzpostavljen in sinhroniziran.
- Uporabniški profil mora imeti pravico do vpogleda v lastne dogodke.

2.3.2.15.3 Merila uspešnosti

- Prikaz zgodovine je na voljo v ≤ 1 sekundi za tipične uporabnike.
- Vsak dogodek vsebuje minimalen sklop podatkov (čas, rezultat, vir, IP).
- Filter omogoča vsaj iskanje po obdobju, tipu dogodka in rezultatu.
- Uporabnik lahko s klikom označi dogodek kot sumljiv.
- Vsi dogodki so dostopni za vsaj zadnjih 180 dni.

2.3.2.15.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~1600	Dostop do zgodovine prijav in	Uporabnik vidi seznam vseh lastnih prijav, odjav in drugih varnostnih	Avtentikacija, aktiven revizijski sistem	Seznam je popoln, točen, dostopen v realnem času.

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
	dogodkov	dogodkov.		
~1601	Prikaz dogodkov z metapodatki	Vsak dogodek vključuje informacije o IP, lokaciji, tipu dostopa, User-Agentu in rezultatu.	OpenSearch beleži popolne podatke	Metapodatki so razvidni v uporabniškem vmesniku na klik posameznega dogodka.
~1602	Filtriranje in iskanje po zgodovini	Uporabnik lahko filtrira po časovnem obdobju, vrsti dogodka, rezultatu (uspeh/neuspeh) in izvoru (SSO/lokalno).	UI omogoča uporabo filtrirnih kontrol	Rezultat filtra se prikaže takoj, brez ponovnega nalaganja strani.
~1603	Označevanje sumljivih dogodkov	Uporabnik lahko ročno označi dogodke, ki mu niso znani ali so potencialno zlonamerni.	UI omogoča oznako in vnos opombe	Označen dogodek se zabeleži kot varnostno opozorilo in posreduje v sistem za spremljanje incidentov (npr. SIEM).

2.3.2.15.5 Diagram primerov uporabe

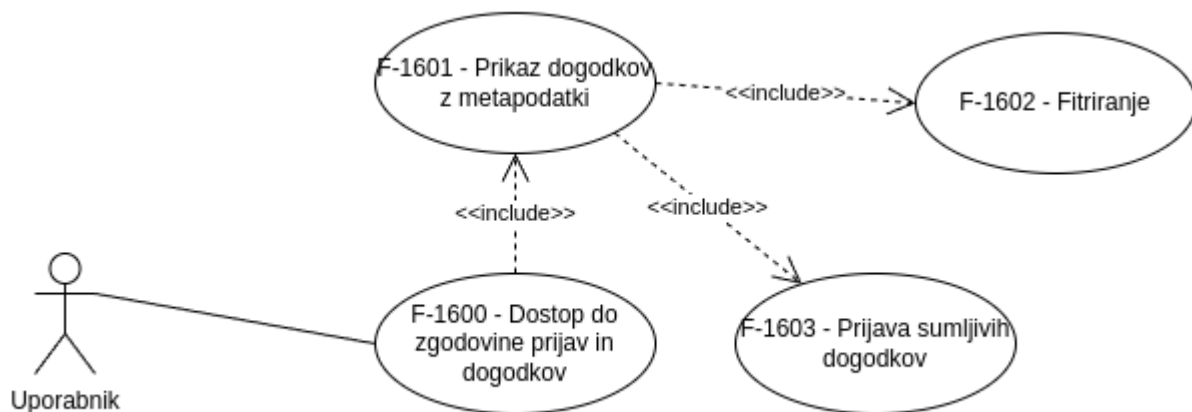


Diagram 17: Diagram primerov uporabe za SKL-VS-F-1600.

2.3.2.16 SKL-VS-F-1700: Zaklepanje uporabniškega računa

2.3.2.16.1 Opis funkcionalnosti

Funkcionalnost zagotavlja zaščitni mehanizem za **začasno ali trajno zaklepanje uporabniškega računa** po določenem številu zaporednih neuspešnih prijav. Gre za kritičen element obrambe pred napadi z ugibanjem gesel (npr. brute-force, credential stuffing) in za ukrep skladnosti z varnostnimi priporočili (npr. ISO/IEC 27002).

Logika zaklepa je naslednja:

- Števec neuspešnih poskusov se vodi po uporabniškem imenu in IP naslovu.
- Če število zaporednih neuspešnih prijav preseže **vnapij konfigurirano mejo** (npr. 5), se uporabniški račun:
 - **začasno zaklene** za določeno obdobje (npr. 15 minut),
 - ali **trajno onemogoči**, če sistemska politika to zahteva.

- Uporabnik dobi jasno informacijo o zaklepu in (če je začasen) tudi čas ponovne možnosti prijave.
- Vse napake se **revizijsko beležijo**, skupaj z IP, časom, User-Agent, in razlogom.
- Sistem omogoča tudi **ročni odklep** preko administrativnega vmesnika ali na zahtevo uporabnika prek kontaktne točke.

2.3.2.16.2 Predpogoji

- Uporabniški račun mora biti aktiven.
- Sistem mora beležiti neuspešne poskuse prijave.
- Konfiguracijska politika za zaklepanje mora biti aktivna in pravilno nastavljena.
- Mehanizem za revizijsko sledenje mora delovati.

2.3.2.16.3 Merila uspešnosti

- Uporabniški račun se zaklene točno ob preseženi meji neuspešnih poskusov.
- Sistem uporabniku poda jasno in neinvazivno sporočilo o razlogu za zaklep.
- Po preteku časa zaklepa je možna ponovna prijava (če ni trajni zaklep).
- Vsi dogodki so ustrezno zabeleženi in pregledni za administratorje.
- Sistem podpira odpornost proti t. i. lockout-abuse napadom (npr. zaklep iz zlonamernega IP naslova za tuji račun).

2.3.2.16.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~1700	Uporabniški zaklep po več neuspešnih prijavah	Sistem zaklene uporabniški račun po preseženi meji neuspešnih prijav.	Prijavni mehanizem omogoča spremljanje neuspešnih poskusov.	Zaklep se sproži pravočasno in velja za predvideno trajanje.
~1701	Konfiguracija pogojev zaklepa	Politika zaklepa se definira v konfiguracijski datoteki (npr. 5 poskusov v 10 minutah → 15-minutni zaklep).	Sistem prebere veljavne konfiguracijske vrednosti.	Vrednosti so uveljavljene ob vsakem poskusu prijave.
~1702	Obvestilo uporabniku	Uporabnik ob zaklepu prejme jasno sporočilo z razlogom in (če je primerno) časom do odklepa.	UI mehanizem za prijavo je aktiven.	Sporočilo ne razkriva, ali uporabniško ime obstaja, s čimer se zmanjša možnost informacijske zlorabe.
~1703	Beleženje poskusov in zaklepa	Vsak neuspešen poskus in sproženi zaklep se beleži v revizijsko sled.	OpenSearch oz. drug revizijski sistem deluje.	Zapis vključuje uporabniško ime, IP, User-Agent, čas in rezultat.
~1704	Odklep računa	Po preteku zaklepa ali po administrativnem posegu	Časovna politika ali administratorski	Ponovna prijava je mogoča brez dodatnih

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
		se račun ponovno aktivira.	ukaz	ukrepov, če ni bil izveden trajni zaklep.
~1705	Odpornost proti zlorabam zaklepanja	Sistem preprečuje zaklepanje računov iz zlonamernih IP naslovov (npr. zaklep za druge uporabnike).	IP-based omejitve in sistemi za zaznavo anomalij	Zaklep se izvaja selektivno in ne vodi do onemogočanja večjega števila uporabnikov zaradi enega napadalca.

2.3.2.16.5 Diagram primerov uporabe

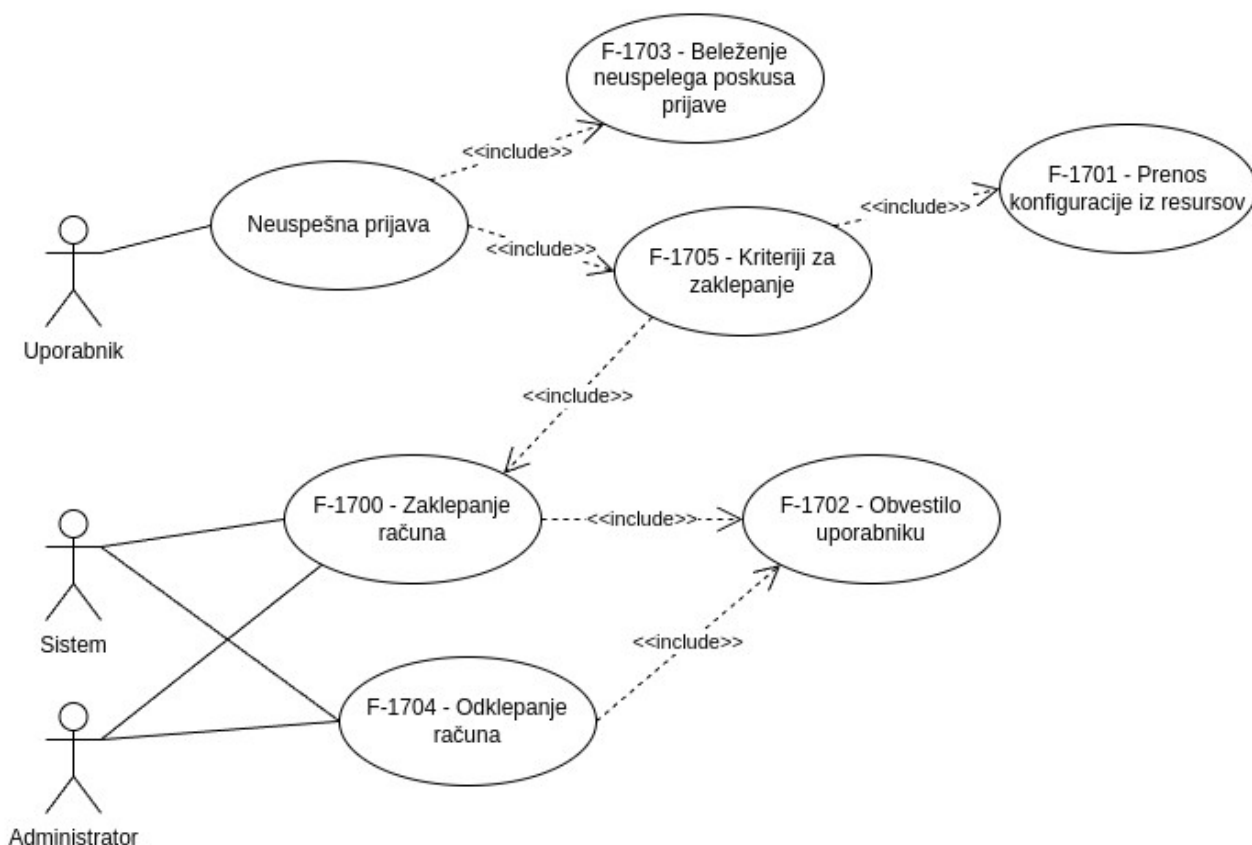


Diagram 18: Diagram primerov uporabe za SKL-VS-F-1700.

2.3.2.17 SKL-VS-F-1800: Sumljive aktivnosti nad računom

2.3.2.17.1 Opis funkcionalnosti

Funkcionalnost zagotavlja, da je uporabnik **takoj obveščen o varnostno sumljivih dogodkih**, povezanih z njegovim uporabniškim računom. Sistem na osnovi vnaprej definiranih pravil sproži obvestilo uporabniku, kadar zazna anomalijo ali dogodek, ki odstopa od običajnega vedenja. Namen je povečanje preglednosti, hitrejši ukrepanje uporabnika in zmanjšanje tveganja zlorabe.

Dogodki, ki sprožijo obvestilo:

- **prijava iz nove naprave ali geografske lokacije**, ki ni bila doslej zabeležena,
- **več neuspešnih poskusov prijave** v kratkem času (npr. ≥ 3 v 5 minutah),
- **zaklep računa**,

- **onemogočanje ali sprememba 2FA,**
- **preklic aktivnih sej** brez predhodno potrjene aktivnosti uporabnika,
- **dostop do zgodovine prijav iz novega vira.**

Obvestilo se posreduje na:

- elektronski naslov uporabnika,
- sistemska obvestila v uporabniškem vmesniku (po prijavi),
- SMS (če je aktiviran).

Vsebina obvestila vključuje:

- datum in čas dogodka,
- vrsto dogodka in njegovo oznako,
- lokacijo oz. IP naslov (če je na voljo),
- navodila za nadaljnje ukrepanje (npr. sprememba gesla, stik s podporo).

2.3.2.17.2 Predpogoji

- Uporabnik mora imeti registriran in preverjen elektronski naslov.
- Mehanizmi za zaznavanje anomalij morajo biti aktivni (logika znotraj varnostnega modula).
- Sistem mora imeti vzpostavljen kanal za pošiljanje elektronske pošte (SMTP), obvestil in SMS (če se uporablja).

2.3.2.17.3 Merila uspešnosti

- Obvestilo se sproži v največ 30 sekundah po zaznavi dogodka.
- Vsebina je jasno strukturirana, ne vsebuje občutljivih podatkov.
- Uporabnik lahko obvestilo potrdi kot "znano" ali "neznano" – za slednje se sproži varnostni ukrep.
- Vsi dogodki in odzivi so revizijsko sledeni.
- Napačne sprožitve (false positives) so omejene na manj kot 1 %.

2.3.2.17.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~1800	Sumljive aktivnosti nad računom	Sistem obvesti uporabnika o sumljivih dogodkih, povezanih z njegovim računom (lokacija, zaklep, spremembe).	Preverjen kontaktni naslov, aktiven varnostni modul	Obvestilo sproženo v ≤ 30 s, vsebina razumljiva, revizijsko zabeležena.
~1801	Detekcija sumljivih dogodkov	Sistem zazna anomalije glede na pravila (geolokacija, hitrost poskusov prijave, status sej, dogodki sprememb).	Pravila definirana v konfiguraciji sistema	Zaznava se sproži samo za netipične dogodke, brez ponavljajočih obvestil.

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~1802	Oblikovanje vsebine obvestila	Sistem pripravi obvestilo s strukturirano vsebino: čas, dogodek, lokacija, IP, priporočeno ukrepanje.	Podatki iz revizijskega sistema dosegljivi	Vsebina je standardizirana, brez občutljivih informacij (npr. geslo, JWT).
~1803	Pošiljanje prek različnih kanalov	Obvestilo se pošlje po e-pošti, kot sistemsko obvestilo ali prek SMS (če omogočeno).	SMTP in drugi komunikacijski kanali delujoči	Obvestilo dostavljeno v ≤60 s, redundanca kanalov po potrebi.
~1804	Uporabniška povratna informacija	Uporabnik lahko označi obvestilo kot znano ali neznano. V slednjem primeru se sproži opozorilo za varnostno podporo ali samodejni ukrep.	UI podpora, obrazec za odziv	Oznaka sproži dodatne postopke (npr. zaklep, zaznamovanje za SIEM).

2.3.2.17.5 Diagram primerov uporabe

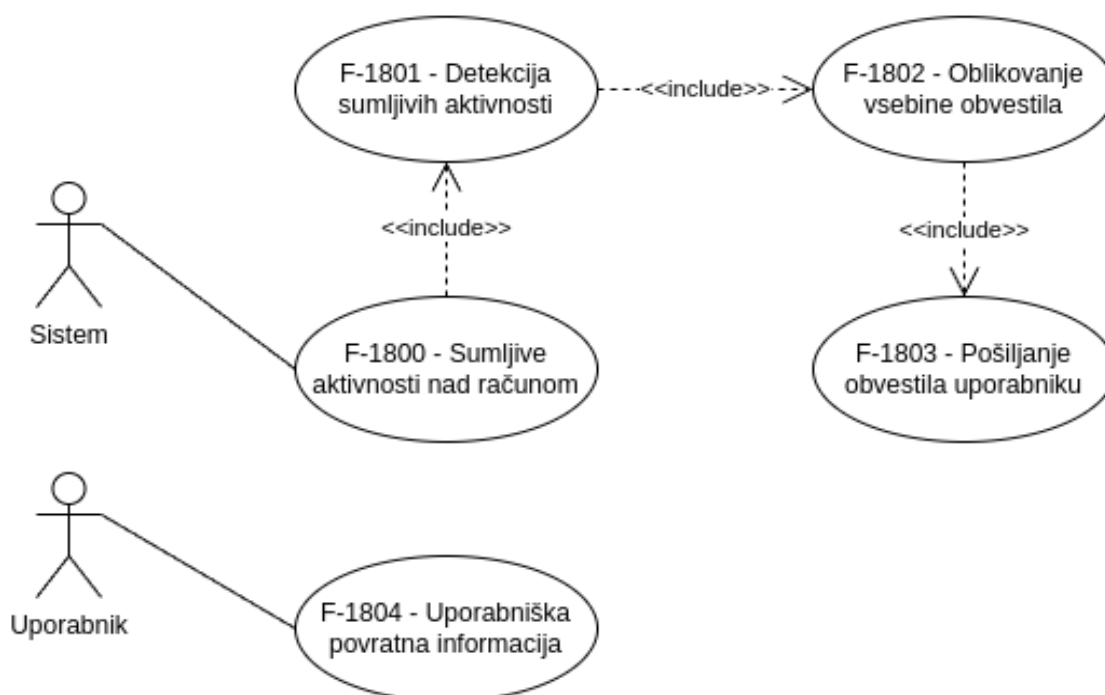


Diagram 19: Diagram primerov uporabe za SKL-VS-F-1800.

2.3.2.18 SKL-VS-F-2000: Obnova pozabljenega gesla

2.3.2.18.1 Opis funkcionalnosti

Funkcionalnost omogoča registriranim uporabnikom sistema POKK **samostojno obnovo pozabljenega gesla** na varen, sledljiv in preverljiv način. Mehanizem je zasnovan skladno z varnostnimi smernicami (npr. OWASP) in vključuje preverjanje identitete uporabnika prek več kanalske potrditve.

Postopek obnove gesla vključuje naslednje korake:

1. Vnos identifikatorja:

- Uporabnik vnese svoj **elektronski naslov** (oz. drug prijavni identifikator).
- Sistem preveri, ali obstaja aktiven uporabniški račun, povezan s tem naslovom.

2. Pošiljanje potrditvene povezave:

- Na elektronski naslov uporabnika se pošlje **enkratna SSO povezava**, ki vodi na vmesnik za nastavitev novega gesla.
- Povezava ima časovno omejeno veljavnost (npr. 30 minut), enkratno uporabo in je vezana na IP ter User-Agent zahtevek.

3. Dodatna potrditev (če je aktiviran 2FA):

- Če ima uporabnik aktivirano 2FA (SMS ali TOTP), se za obnovo gesla zahteva tudi **potrditev s kodo**, poslano na registriran kanal.

4. Vnos novega gesla:

- Uporabnik vnese novo geslo, ki mora ustrezati sistemsko definiranim pravilom (dolžina, kompleksnost).
- Geslo se zgošči z enosmernim algoritmom z uporabo individualne soli in shrani v bazo.

5. Zaključek postopka:

- Ob uspešni spremembi gesla se vse obstoječe seje prekličejo.
- Uporabnik prejme potrditev po e-pošti.
- Revizijsko se zabeleži, kdo, kdaj in s katerega IP naslova je bil postopek izveden.

2.3.2.18.2 Predpogoji

- Elektronski naslov mora biti registriran in preverjen.
- SMTP kanal mora biti aktiven.
- Če je 2FA aktiviran, mora biti na voljo registrirana potrdilna komponenta (SMS številka ali TOTP naprava).
- Revizijski sistem mora biti aktiven.

2.3.2.18.3 Merila uspešnosti

- Povezava za obnovo gesla je dostavljena v ≤ 60 s po oddaji zahtevka.
- Nova gesla morajo biti preverjena po veljavnih pravilih varnosti.
- Postopek se zaključi v ≤ 5 minutah (ob normalni uporabniški interakciji).
- Po spremembi gesla so vse prejšnje seje deaktivirane.
- Vsi dogodki so revizijsko zabeleženi.

2.3.2.18.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~2000	Obnova pozabljenega	Uporabnik z registriranim elektronskim naslovom sproži	Elektronski naslov obstaja in	Povezava poslana, nova gesla veljavna,

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
	gesla	postopek za obnovo pozabljenega gesla.	je preverjen.	vse seje preklicane.
~2001	Pošiljanje povezave za obnovo	Sistem generira enkratno povezavo za nastavitev novega gesla.	SMTP strežnik dosegljiv	Povezava časovno omejena, vezana na IP/User-Agent.
F-003	Preverjanje 2FA (če aktiviran)	V primeru aktivnega 2FA se zahteva potrditev s kodo pred nastavljenjem novega gesla.	Uporabnik ima 2FA aktiven	Koda uspešno preverjena ali postopek zavrnjen.
~2003	Vnos in validacija novega gesla	Uporabnik vnese novo geslo, ki se preveri glede na varnostna pravila in shrani v bazo.	Povezava še ni potekla	Geslo ustreza pogojem, zgoščeno in shranjeno.
F-1500	Preklic obstoječih sej	Po uspešni spremembi gesla se vse obstoječe seje prekličejo.	Seja aktivna	Uporabnik mora po spremembi gesla ponovno prijaviti.

2.3.2.18.5 Diagram primerov uporabe

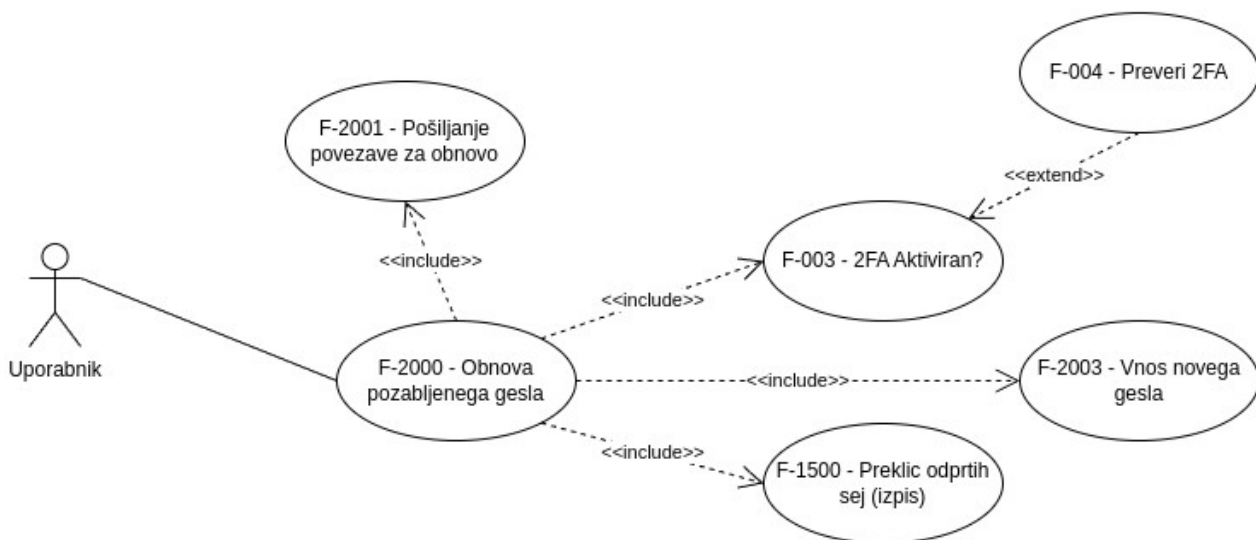


Diagram 20: Diagram primerov uporabe za SKL-VS-F-2000.

2.3.2.19 SKL-VS-F-2200: Upravljanje uporabnikov

2.3.2.19.1 Opis funkcionalnosti

Funkcionalnost omogoča **internim uporabnikom z ustreznimi pravicami (READ/WRITE za uporabniške podatke)** pregled in upravljanje vseh uporabniških računov znotraj sistema POKK.

Sistem omogoča **vse ključne administrativne posege**, potrebne za obvladovanje življenjskega cikla uporabniškega profila, skladno z varnostnimi politikami, zahtevami GDPR in revizijsko sledljivostjo.

Vsaka izvedena operacija se vedno **revizijsko zabeleži v OpenSearch**, hkrati pa tudi v **dnevnik akcij nad uporabnikom**, ki omogoča kronološki pregled vseh upravnih posegov (glej spodaj: pregled zgodovine aktivnosti).

Glavne zmožnosti funkcionalnosti:

1. Pregled uporabniškega profila:

- Prikaz osnovnih podatkov (ime, email, status, zadnja prijava, tip uporabnika, metoda avtentikacije),
- Preverjanje aktivacije 2FA in statusov prijave.

2. Urejanje atributov uporabnika:

- Sprememba: naziva, interne oznake, organizacije, aktivnih vlog, klasifikacije uporabnika (interni/zunanji/sistemski),
- Možnost dodajanja opombe.

3. Ustvarjanje novega uporabnika:

- Vnos minimalnega nabora podatkov (ime, email),
- Na vpisani elektronski naslov se pošlje povezava za prvo prijavo in nastavitev gesla,
- Uporabniški račun ostane neaktiven, dokler ni uspešno dokončana aktivacija.

4. Proženje ponastavitve gesla:

- Uporabniku se posreduje povezava za ponastavitev gesla (funkcionalnost ~2000),
- Vse trenutne seje se ob uspešni spremembi gesla prekličejo.

5. Zaklepanje / odklepanje računa:

- Ročno zaklepanje z nastavitvijo **časovnega obdobja** (do datuma),
- Odklep s takojšnjim učinkom.

6. Popolna deaktivacija računa:

- Račun je onemogočen za nadaljnje prijave do ponovne aktivacije.

7. Izbris uporabniškega računa (GDPR zahteva):

- Izbris se izvede samo, če je zahteva za izbris podana s strani posameznika,

- Administrator mora navesti **pisno obrazložitev** (obvezno),
- Izveden je mehki izbris, z revizijsko sledjo.

8. Ponovni začetek postopka 2FA (reset):

- Možen samo, če je **trenutno prijavljen administrator** sam zaščiten z aktivnim 2FA,
- Pred izvedbo mora administrator svojo zahtevo **elektronsko podpisati z 2FA** (npr. TOTP koda),
- Postopek onemogoči obstoječe 2FA metode uporabnika in omogoči ponovni onboarding.

9. Pregled podanih soglasij uporabnika:

- Pregled vseh aktivnih in zgodovinskih soglasij, ki jih je uporabnik podal, skupaj z verzijami in časovnimi žigi,
- Samo branje – brez možnosti spremembe odločitev.

10. Pregled zgodovine aktivnosti uporabnika:

- Prikaz dogodkov, kot so: prijave, neuspešni poskusi, zaklepi, spremembe profilov, oddaja soglasij, varnostne operacije,
- Prikaz z metapodatki: čas, IP, vir dogodka, status.

11. Referenca:

- Za pregled nad zastopniškimi vlogami uporabnika glej **SKL-VS-F-2201: Pregled zastopništev uporabnika**.

2.3.2.19.2 Predpogoji

- Uporabnik ima avtorizirano vlogo z WRITE pravicami za uporabniške podatke.
- Komponente za pošiljanje email obvestil, upravljanje sej, 2FA in OpenSearch morajo biti aktivne.
- Za reset 2FA mora biti administrator sam zaščiten z aktivnim 2FA in potrditi zahtevo s kodo.
- Za delo z uporabniki mora biti administrator sam zaščiten z aktivnim 2FA.

2.3.2.19.3 Merila uspešnosti

- Vsi ukrepi (kreacija, zaklep, reset 2FA, izbris) so izvedeni v ≤ 1 s.
- Povezave za prvo prijavo ali obnovo gesla so dostavljene v ≤ 60 s.
- Vsi dogodki so zabeleženi v revizijski sistem z metapodatki.
- V primeru GDPR izbriša se račun loči od vseh aktivnih povezav, a ostane revizijsko označen.
- Ponovni onboarding 2FA se omogoči le, če je bil administrator ustrezno preverjen.

2.3.2.19.4

Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~2200	Upravljanje uporabnikov	Interni uporabnik z ustreznimi pravicami lahko upravlja z uporabniškimi računi in njihovim statusom.	WRITE pravice za uporabnike, OpenSearch aktiven	Vse akcije izvedene z nizko latenco in popolno revizijsko sledljivostjo.
~2201	Ustvarjanje uporabniškega računa	Uporabnik se doda ročno, na email se pošlje povezava za prvo prijavo.	Email naslov preverjen, SMTP aktiven	Račun je ustvarjen, uporabnik aktivira dostop sam.
~2202	Urejanje atributov uporabnika	Sprememba internih podatkov profila: naziv, oznaka, vloga, organizacija.	UI za urejanje omogočen	Spremembe zabeležene, prikazane ob naslednjem dostopu.
F-2000	Ponastavitev gesla	Administrator sproži funkcionalnost ~2000.	Uporabniški račun obstaja, email dosegljiv	Povezava dostavljena, seja preklicana po spremembi.
~2204	Zaklep in odklep uporabnika	Ročno zaklepanje za določeno obdobje ali takojšnje odklepanje.	Prijavni mehanizem aktiviran	Status se spremeni takoj, uporabnik obveščen (če je omogočeno).
~2205	Deaktivacija računa	Račun se popolnoma onemogoči do ročnega ponovnega vklopa.	-	Prijava ni več mogoča, povezane seje deaktivirane.
~2206	Izbris po GDPR	Ob ustrezni zunanji zahtevi se uporabnika logično izbriše; obrazložitev je obvezna.	Zunanja zahteva + obrazložitev	Račun označen kot izbrisan, a podatki ohranjeni za sledljivost.
~2207	Reset 2FA (s preverbo administratorja)	Administrator lahko prekliče 2FA uporabnika le, če je sam zaščiten z 2FA in to preveri s kodo.	Administrator ima aktiven 2FA	Reset 2FA se potrdi s kodo; uporabnik lahko začne onboarding znova.
~2209	Pregled zgodovine aktivnosti	Prikaz revizijskih dogodkov za uporabnika: prijave, zaklepi, spremembe, soglasja, varnostne operacije.	OpenSearch aktiven	Dogodki prikazani s klasifikacijo in metapodatki (čas, IP, vir, status).

2.3.2.19.5

Diagram primerov uporabe

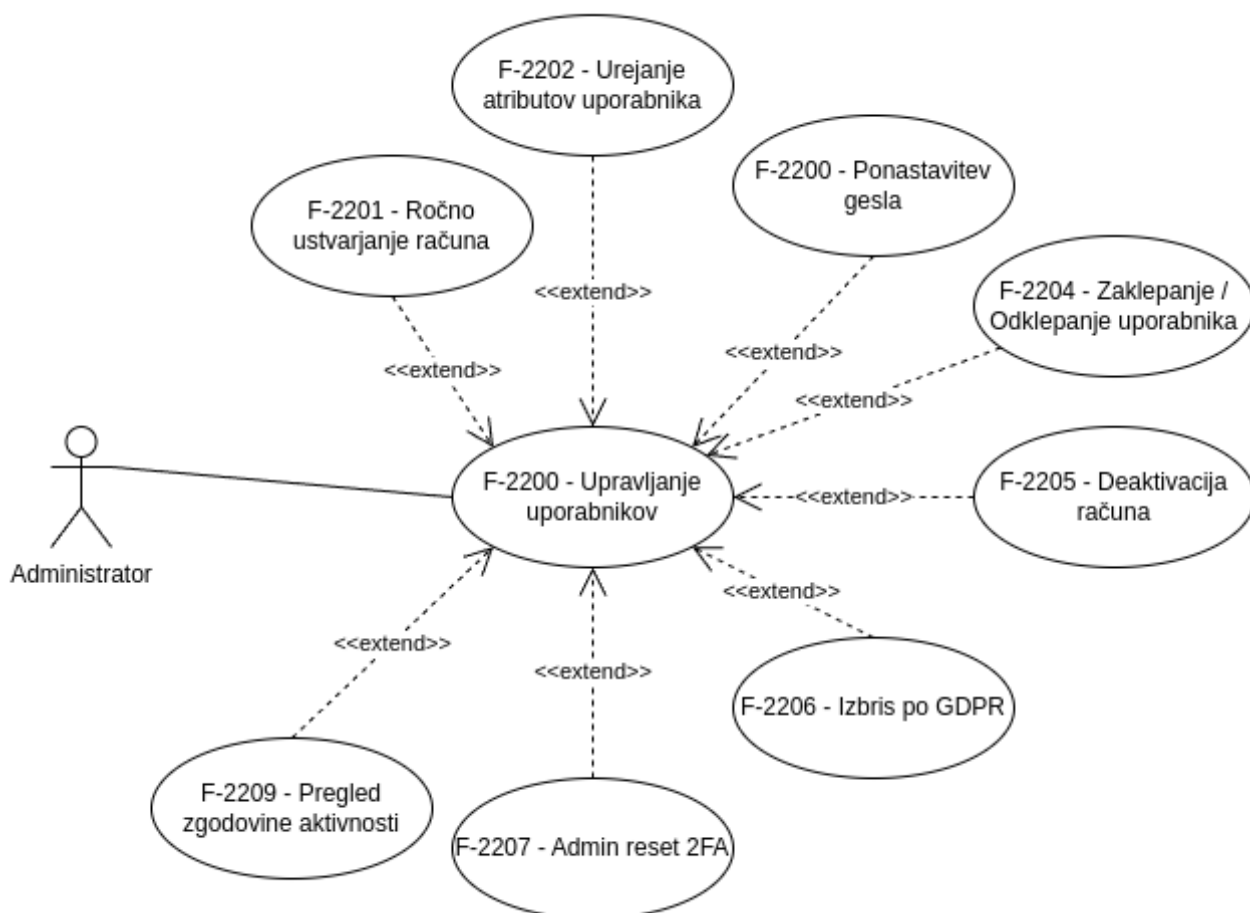


Diagram 21: Diagram primerov uporabe za SKL-VS-F-2200.

2.3.2.20 SKL-VS-F-2300: Upravljanje z osebami

2.3.2.20.1 Opis funkcionalnosti

Funkcionalnost omogoča **internim uporabnikom z ustreznimi pravicami (READ/WRITE za entiteto oseba)** pregled in upravljanje **oseb** v sistemu POKK. Oseba je entiteta, ki predstavlja:

- fizično osebo (naravna oseba),
- ali **tehnično osebo** (npr. API uporabnik), vezano na pravno osebo.

Vse osebe so vključene v enoten sistemski model, kar omogoča poenoten nadzor nad:

- **zastopništvu** (oseba ↔ subjekt),
- **vlogami** in pravicami (znotraj zastopništva),
- **uporabniškimi računi** (preko vezave uporabnik ↔ oseba).

Vsaka oseba vsebuje ključne identifikatorje:

- **davčna številka (DŠ)** (uporablja se za identifikacijo in povezavo s subjektom).

Sistem omogoča:

- ustvarjanje, pregled in urejanje oseb (če so lokalnega izvora),

- klasifikacijo osebe po tipu (FIZIČNA, TEHNIČNA, ZUNANJI_VIR),
- revizijsko spremljanje vseh sprememb.

Tehnična oseba (API klient):

- Modelirana je kot običajna oseba z atributom `tip_osebe = TEHNIČNA`.
- Te osebe so praviloma povezane z uporabniškimi računi, ki uporabljajo **avtentikacijo z API ključi, certifikati ali drugimi neinteraktivnimi mehanizmi**.
- Pravice teh oseb so omejene z dodeljenimi vlogami (npr. READ, EXPORT).

Glavne zmožnosti:

1. Iskanje in pregled oseb:

- Po: DŠ, imenu, priimku, tipu osebe, viru podatkov.
- Vizualna oznaka za TEHNIČNA, FIZIČNA, ZUNANJI_VIR.

2. Ustvarjanje nove osebe:

- DŠ mora biti unikatna .
- Ob vnosu se določi **tip osebe**, ki vpliva na pravila validacije in prikaza.

3. Urejanje obstoječe osebe:

- DŠ **se ne smespreminjati** po ustvarjanju.
- Osebe iz zunanjih virov (npr. CRG, RKG) so **read-only**.

4. Pregled zgodovine sprememb:

- Vse spremembe se zabeležijo: atribut, staro, novo, čas, uporabnik.

2.3.2.20.2 Predpogoji

- Administrator ima WRITE pravice za entiteto oseba.
- DŠ je edinstvena .
- Tip osebe mora biti definiran (FIZIČNA, TEHNIČNA, ZUNANJI_VIR).
- Osebe iz zunanjih virov se ne morejo spreminjati.

2.3.2.20.3 Merila uspešnosti

- Oseba se ustvari le, če je kombinacija DŠ unikatna.
- Za tehnične osebe je DŠ **obvezen in mora ustrezati subjektu**, .
- Vse spremembe so revizijsko zabeležene.
- Osebe so prikazane z jasnim statusom (tip, vir, aktivnost).

2.3.2.20.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~2300	Upravljanje z	Pregled, vnos in urejanje oseb	WRITE pravice	Osebe validirane, jasno

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
	osebami	(fizičnih in tehničnih), ki predstavljajo nosilce zastopništva v sistemu.	za entiteto oseba	klasificirane, vse spremembe revizijsko sledljive.
~2301	Iskanje in filtriranje	Iščemo po , DŠ, imenu, tipu osebe, viru.	Indeksirani atributi	Povratni rezultat v ≤ 500 ms.
~2302	Ustvarjanje nove osebe	Vnos nove osebe z atributi in klasifikacijo (FIZIČNA, TEHNIČNA, ZUNANJI_VIR).	Kombinacija DŠ še ne obstaja	Nova oseba je aktivna in takoj dostopna.
~2303	Urejanje obstoječe osebe	Urejanje izbranih atributov (razen DŠ), če oseba ni iz zunanjega vira.	Vloga in vir to dovoljujeta	Vse spremembe zabeležene.
~2304	Pregled zgodovine sprememb	Prikaz vseh sprememb z metapodatki (čas, uporabnik, staro/novo, razlog).	Revizijsko sledenje aktivno	Zgodovina je popolna in prikazana v uporabniškem vmesniku.

2.3.2.20.5 Diagram primerov uporabe

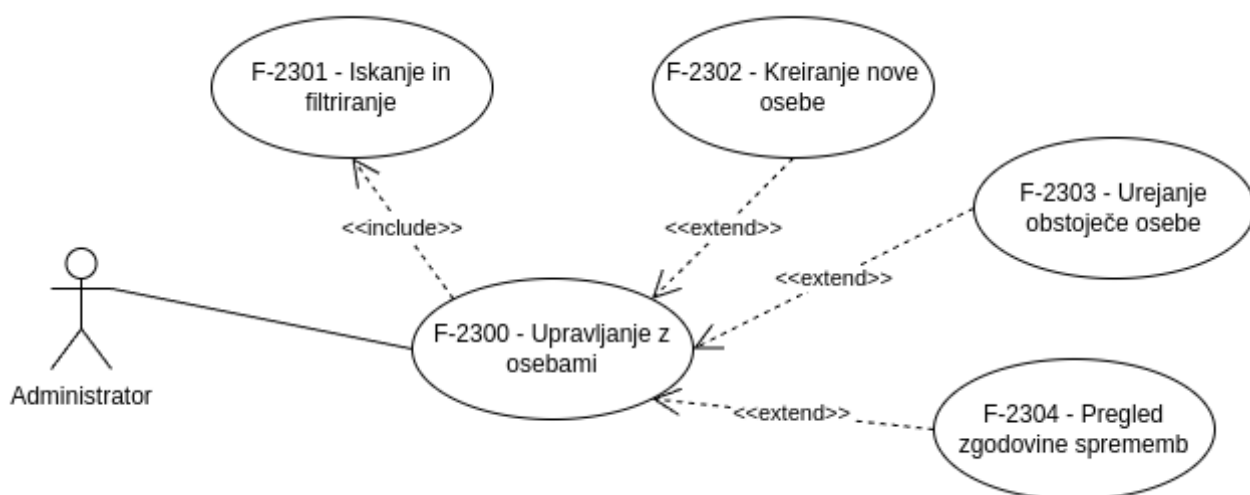


Diagram 22: Diagram primerov uporabe za SKL-VS-F-2300.

2.3.2.21 SKL-VS-F-2400: Upravljanje z zastopništvu

2.3.2.21.1 Opis funkcionalnosti

Funkcionalnost omogoča **internim uporabnikom z ustreznimi pravicami (WRITE za entiteto zastopništvo)** upravljanje z razmerji **med osebami (fizične osebe) in subjekti** (pravne osebe, kmetijska gospodarstva, organizacije).

Vsako **zastopništvo predstavlja razmerje**, v okviru katerega ima oseba pravico **zastopati določeni subjekt** z določenim obsegom pravic. Na osnovi tega razmerja se nato uporabnikom lahko dodeli pravice za delovanje v imenu subjekta (glej **SKL-VS-F-2200**).

Zastopništvo definira:

- **Osebo** (vezano na DŠ, glej SKL-VS-F-2300),
- **Subjekt** (pravna/fizična entiteta, iz referenčnega registra subjektov),
- **Časovno obdobje veljavnosti** (velja_od, velja_do – opsijsko),

- **Nabor pravic**, ki jih ima oseba v okviru tega razmerja (glej *tabelo pravic*).

Glavne zmožnosti:

1. Dodajanje novega zastopništva:

- Izbor obstoječe osebe (fizična oseba) in subjekta,
- Nastavitev obdobja veljavnosti (*velja_od*, *velja_do*),
- Dodelitev pravic po matrici pravic.

2. Urejanje obstoječega zastopništva:

- Sprememba datuma veljavnosti ali nabora pravic.

3. Odvzem (deaktivacija) zastopništva:

- Zaprtje razmerja z nastavitvijo *velja_do* ali brisanje (mehko).

4. Pregled aktivnih in poteklih zastopništev:

- Po osebi, subjektu, časovnem obdobju ali pravici.

5. Zgodovina sprememb:

- Vsi dogodki (dodano, spremenjeno, deaktivirano) se revizijsko beležijo.

☞ Ta funkcionalnost služi kot **pravna osnova za nadaljnjo dodelitev zastopniških pravic uporabnikom** – na osnovi povezave: **uporabnik** → **oseba** → **subjekt**.

2.3.2.21.2 Predpogoji

- Oseba in subjekt morata biti že evidentirana v sistemu.
- Uporabnik mora imeti avtorizacijo za urejanje zastopništev.
- Časovni okviri morajo biti logično pravilni (*velja_do* > *velja_od*).
- Matrika pravic mora biti dosegljiva.

2.3.2.21.3 Merila uspešnosti

- Zastopništvo se vzpostavi v ≤ 1 s.
- Veljavnost in pravice začnejo veljati takoj, če je današnji datum znotraj obdobja.
- Zgodovina vseh sprememb nad razmerjem je zabeležena.
- Sistem ne omogoča prekrivajočih se razmerij med isto osebo in subjektom (razen, če je namenoma omogočeno).
- Pravice so skladne z dovoljeno matriko (glej *tabelo pravic*).

2.3.2.21.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~2400	Upravljanje z zastopništv	Interni uporabnik ureja razmerja med osebami in subjekti – vzpostavlja formalna zastopniška razmerja.	Oseba in subjekt obstajata, pravice definirane	Razmerja konsistentna, podatki validirani, spremembe sledljive.

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~2401	Dodelitev zastopništva	Vzpostavi se novo razmerje oseba ↔ subjekt z veljavnostjo in pravicami.	Nobeno obstoječe razmerje za isti par v tem obdobju	Novo razmerje začne veljati glede na "velja od".
~2402	Urejanje zastopništva	Možna sprememba datuma veljavnosti ali pravic.	Zastopništvo obstaja, ni zaklenjeno	Nova vrednost nadomesti obstoječo, sprememba zabeležena.
~2403	Deaktivacija (odvzem) zastopništva	Z razmerjem se prekine veljavnost (nastavi se velja_do , ali se status označi kot neaktiven).	Avtorizacija aktivna	Razmerje preneha učinkovati, uporabniki izgubijo povezane avtorizacije.
~2404	Pregled zastopniških razmerij	Iskanje in filtriranje po subjektu, osebi, statusu, obdobju in pravicah.	Podatki indeksirani	Povratni rezultat v ≤ 500 ms, prikaz v strukturirani obliki.
~2405	Pregled zgodovine sprememb nad razmerjem	Prikaz vseh sprememb (kdaj, kdo, kaj), vključno z verzijami pravic in spremembami časovne veljavnosti.	Revizijsko sledenje aktivno	Vsaka sprememba je zabeležena z metapodatki (čas, uporabnik, sprememba).

Opomba:

Glej tabelo pravic, kjer je opredeljena sistemska matrika pravic za zastopniška razmerja. Te pravice se uporabljajo za kontekstualno avtorizacijo v modulu uporabnikov (~2200).

2.3.2.21.5 Diagram primerov uporabe

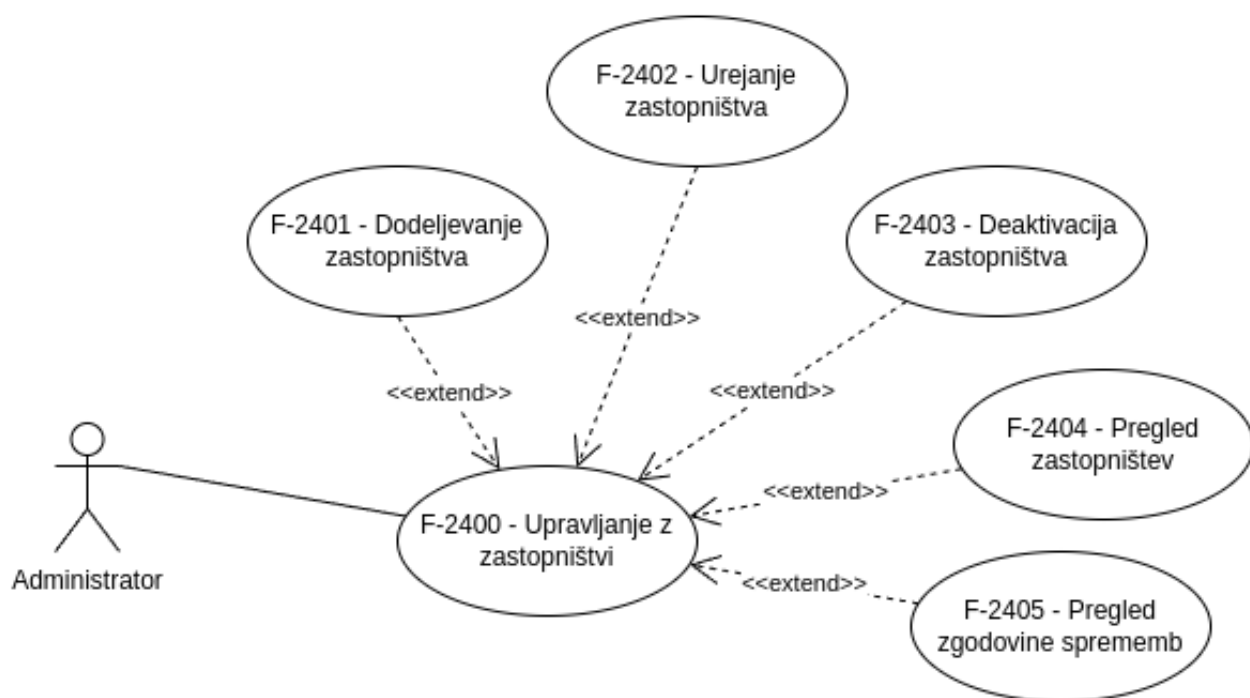


Diagram 23: Diagram primerov uporabe za SKL-VS-F-2400.

2.3.2.22 SKL-VS-F-2500: Povezovanje uporabniškega računa z osebo

2.3.2.22.1 Opis funkcionalnosti

Funkcionalnost omogoča **ustrezno avtoriziranemu internemu uporabniku**, da **vzpostavi, pregleda ali prekine povezavo med uporabniškim računom in entiteto fizične osebe (oseba)**.

Ta povezava predstavlja osnovo za določanje **pravnega konteksta delovanja uporabnika v sistemu POKK**, saj s tem uporabnik pridobi posreden dostop do subjektov, ki jih ta oseba zastopa (glej SKL-VS-F-2400).

Ključna pravila povezovanja:

- Povezavo med uporabniškim računom in osebo je mogoče ustvariti **samo ročno**, kadar uporabnik **ni bil registriran preko SI-PASS** (federativna prijava).
- V primeru SI-PASS registracije je bila povezava vzpostavljena že ob registraciji na podlagi davčne številke – taka vezava je **zaklenjena in se ne more ročno spreminjati**.
- Ena oseba je lahko **povezana z največ enim uporabniškim računom**.
- En uporabniški račun je lahko **povezan z največ eno osebo**.
- Vsaka operacija se **beleži v revizijsko sled in zgodovino vezav**.

Glavne zmožnosti:

1. Vzpostavitev povezave:

- Iz entitete **uporabnik**: izbira osebe (preko imena),
- Iz entitete **oseba**: izbira uporabniškega računa (po emailu ali ID-ju),
- Sistem preveri, da:
 - uporabnik ni registriran prek SI-PASS,
 - nobena od entitet ni že vezana.

2. Prikaz statusa povezave:

- V obeh entitetah (oseba, uporabnik) je jasno prikazano:
 - ali povezava obstaja,
 - komu je vezana,
 - kako je bila ustvarjena (SI-PASS ali ročno),
 - kdaj in kdo jo je ustvaril.

3. Prekinitev povezave (odvezava):

- Administrator lahko povezavo **prekine**, razen če je bila vzpostavljena prek SI-PASS.
- Ob prekinitvi sistem opozori na posledice (izguba zastopniških pravic).
- Potreben je vnos **pisne obrazložitve** (obvezen atribut).

4. Pregled zgodovine povezav:

- Prikaz vseh zgodovinskih vezav:

- kdaj je bila vzpostavljena,
- kdo jo je ustvaril ali prekinil,
- vključno z razlogom za prekinitev,
- tip povezave (ročno / SI-PASS).

2.3.2.22.2 Predpogoji

- Uporabnik ima WRITE pravice za modul vezav uporabnik-oseba.
- Oba entiteta (uporabnik in oseba) morata obstajati v sistemu.
- Uporabniški račun ni registriran prek SI-PASS (če želiš ročno povezavo).
- Niti uporabnik niti oseba **nista že povezani** (ni podvojenih vezav).

2.3.2.22.3 Merila uspešnosti

- Sistem vedno preveri veljavnost pogojev povezovanja.
- Povezava je enolična in revizijsko dokumentirana.
- Prekinitve povezave zahtevajo obrazložitev in so reverzibilne le prek administrativnega postopka.
- Vsak uporabnik in oseba imata jasno prikazan status vezave.
- Zgodovina vezav je popolna in časovno sledljiva.

2.3.2.22.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~2500	Povezovanje uporabniškega računa z osebo	Administrator poveže uporabnika s fizično osebo, če ta povezava še ne obstaja in ni bila že ustvarjena preko SI-PASS.	Oseba in uporabnik obstajata, uporabnik ni iz SI-PASS	Povezava je edinstvena in takoj vidna.
~2501	Vzpostavitev povezave	Izvedba povezave iz smeri uporabnika ali osebe.	Nobena entiteta ni že vezana	Sistem preveri pogoje in ustrezno označi vir povezave.
~2502	Prikaz statusa povezave	Prikaz trenutne vezave na obeh straneh (oseba ↔ uporabnik), z izvorom, časom in ustvarjalcem.	Entiteta obstaja	Podatki so prikazani v preglednem formatu, vključno z opozorili.
~2503	Prekinitve povezave (odvezava)	Administrator lahko prekine vezavo, če ni bila ustvarjena preko SI-PASS; vnos obrazložitve je obvezen.	Povezava ni zaklenjena	Povezava se takoj prekine, vsa povezana avtorizacija preneha veljati.
~2504	Zgodovina vezav	Sistem prikazuje vse pretekle povezave med uporabniki in osebami, vključno z razlogi, spremembami in ustvarjalci.	Revizijsko sledenje aktivno	Kronološki zapis, razviden tip povezave, razlog za morebitno prekinitev.

2.3.2.22.5

Diagram primerov uporabe

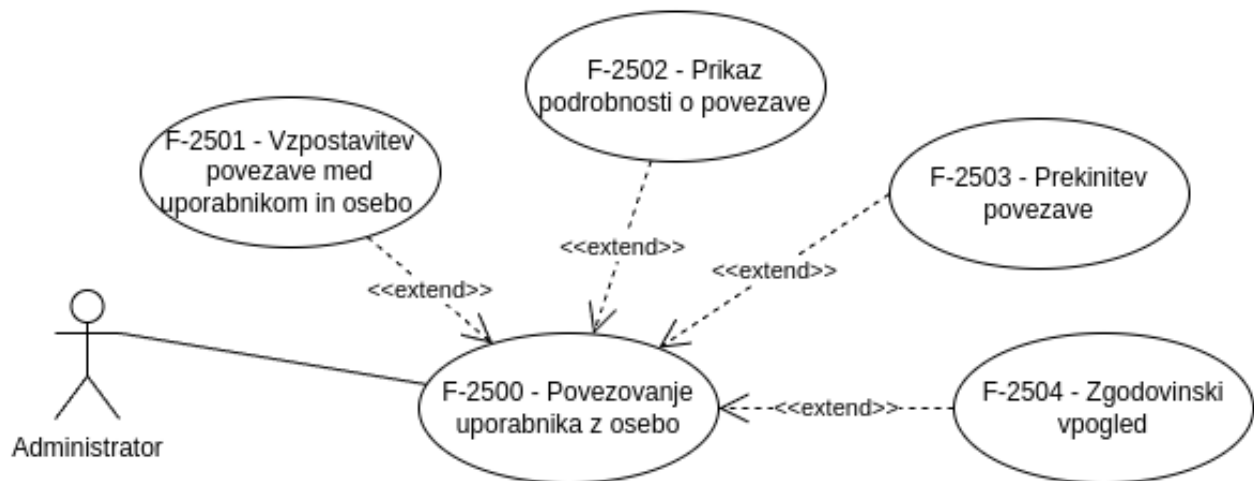


Diagram 24: Diagram primerov uporabe za SKL-VS-F-2500.

2.3.2.23

SKL-VS-F-2700: Upravljanje kataloga področij (Realm)

2.3.2.23.1

Opis funkcionalnosti

Funkcionalnost omogoča **ustrezno avtoriziranemu administratorju** upravljanje s sistemskim katalogom **področij delovanja sistema** – t. i. “**Realm-ov**”. Področje (Realm) je **logična enota avtorizacije**, ki se uporablja za:

- **organizacijo funkcionalnosti** po vsebinskih sklopih (npr. »Subjekti«, »Živali«, »Soglasja«, »Evidenca zdravljenj«),
- **grupiranje pravic**, vezanih na posamezne funkcionalnosti in uporabniške vloge.

S tem se vzpostavi sistemska podlaga za nadaljnjo izgradnjo **matrike pravic**, ki povezuje:

- **področje (realm)**,
- **funkcionalnost**,
- **pravico** (npr. READ, WRITE, DELETE, ADMIN).

Glavne zmožnosti:

1. Kreiranje novega področja (Realm):

- Uporabnik vnese ime področja (unikatno znotraj kataloga),
- Ime mora biti jasno, opisno in enoznačno (npr. »SUBJEKTI«, »ZDRAVILA«, »DOKUMENTI«),
- Sistem zagotovi enoličnost (npr. s tehničnim ključem).

2. Preimenovanje obstoječega področja:

- Ime področja se lahko spremeni, vendar mora biti razlog preimenovanja zabeležen,
- Sistem preveri, da sprememba ne povzroči konflikta z obstoječimi vezavami.

3. Pregled seznama področij:

- Prikaz vseh področij s tehničnim ID, nazivom in časom zadnje spremembe,

- Možnost iskanja in filtriranja po nazivu.

4. Revizijsko sledenje:

- Vse spremembe nad katalogom področij se zabeležijo v revizijsko sled.

2.3.2.23.2 Predpogoji

- Uporabnik ima WRITE pravice za modul "Katalog področij".
- Ime področja je unikatno.
- Sistem mora imeti omogočeno revizijsko beleženje.

2.3.2.23.3 Merila uspešnosti

- Kreiranje in preimenovanje se izvršita v ≤ 500 ms.
- Vsaka sprememba je revizijsko zabeležena z uporabnikom, časom in razlogom.
- Vnos ne dovoljuje praznih ali podvojenih imen.
- Sistem prikazuje vedno aktualen seznam področij.

2.3.2.23.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~2700	Upravljanje kataloga področij (Realm)	Administrator ustvarja in upravlja s področji, ki služijo kot nosilci funkcionalnih sklopov v sistemski avtorizaciji.	WRITE pravice za modul Realm	Vnos uspešen, unikatnost zagotovljena, vse spremembe zabeležene.
~2701	Kreiranje novega področja	Dodajanje novega realm-a z unikatnim imenom.	Ime ne obstaja	Realm vključen v katalog, takoj dostopen za uporabo v avtorizacijski matrici.
~2702	Editiranje obstoječega področja	Sprememba podatkov obstoječega realm-a z utemeljitvijo.	Realm obstaja	Sprememba evidentirana, ne povzroči podvajanj.
~2703	Pregled seznama področij	Seznam vseh obstoječih realm-ov z atributi (ime, ID, status, čas spremembe).	-	Povratni rezultat ≤ 500 ms, seznam vedno ažuren.

2.3.2.23.5 Diagram primerov uporabe

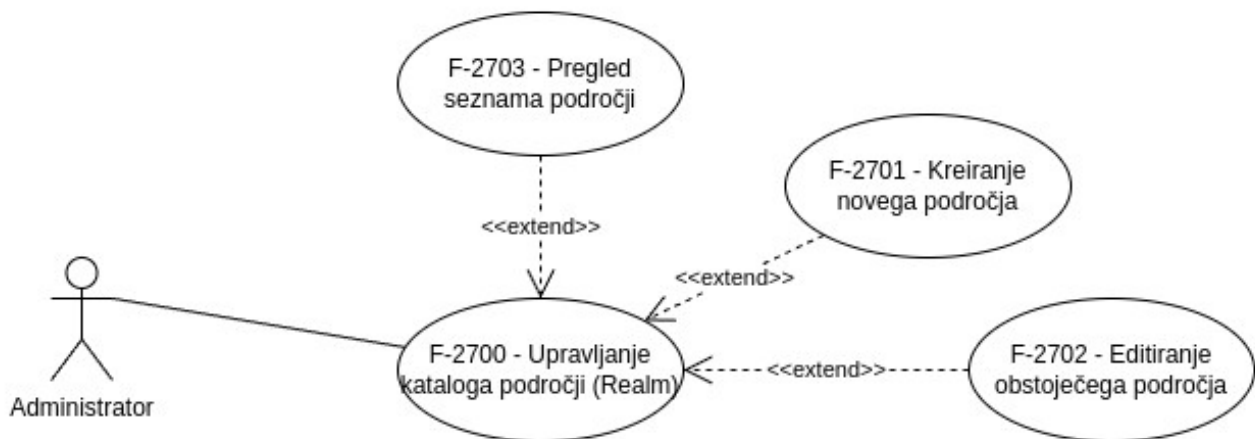


Diagram 25: Diagram primerov uporabe za SKL-VS-F-2700.

2.3.2.24 SKL-VS-F-2800: Upravljanje vlog znotraj področja (Role-Realm)

2.3.2.24.1 Opis funkcionalnosti

Funkcionalnost omogoča **administratorju z ustreznimi pravicami (WRITE za katalog vlog)** upravljanje z **vlogami**, ki so **vsebinsko vezane na področja (Realm)** in določajo **matriko pravic**, ki jih uporabniki lahko uveljavljajo znotraj posameznega funkcionalnega sklopa.

Vsaka vloga (Role) predstavlja **logično ime pravno-funkcijske vloge uporabnika**, npr. *UREJEVALEC_SUBJEKTA*, *SVETOVALEC*, *VETERINAR_ZUNANJI*, ki znotraj določenega področja (npr. »SUBJEKTI«, »ŽIVALI«) uveljavlja pravice (READ, WRITE, DELETE, ADMIN...).

Vloge niso neposredno vezane na uporabnike – za to služi ločena funkcionalnost dodeljevanja vlog uporabnikom.

Vsaka vloga vključuje:

- **Naziv vloge** (unikaten znotraj sistema),
- **Področje (Realm)**, na katerega je vezana,
- **Opis vloge** (v namen razumevanja funkcionalnega pomena),
- **Tip vloge** (sistemska, aplikativna, zunanja, storitvena),
- **Seznam dovoljenih pravic** (iz nabora: READ, WRITE, DELETE, ADMIN, EXECUTE, EXPORT...),
- **Sklic na funkcionalnosti**, na katere se te pravice nanašajo (iz kataloga funkcionalnosti),
- **Status (aktivna / neaktivna).**

Glavne zmožnosti:

1. Ustvarjanje nove vloge

- Vnos vseh atributov vloge, vezava na obstoječe področje.

- Seznam dovoljenih pravic je izbran iz sistemskega kataloga.
- Vloga se shrani kot aktivna in je na voljo za dodeljevanje uporabnikom.

2. Urejanje obstoječe vloge

- Sprememba naziva, opisa, tipa, pravic in vezanih funkcionalnosti.
- Spremembe se ne smejo uporabljati retroaktivno brez revizije (če je vloga že dodeljena uporabnikom).

3. Onemogočanje vloge

- Sprememba statusa na "neaktivna", kar prepreči nadaljnje dodeljevanje.
- Obstoječe dodelitve ostanejo aktivne, razen če sistemska politika določa drugače.

4. Kopiranje vloge

- Administrator lahko obstoječo vlogo duplicira kot osnovo za novo.
- Kopirajo se ime (z novo oznako), opis, pravice, funkcionalnosti, vezava na področje.

5. Pregled vseh vlog znotraj sistema

- Možnost filtriranja po področju, statusu, tipu, imenu ali pravici.
- Jasna indikacija, ali je vloga uporabljena v sistemu (vezana na uporabnike).

6. Revizijsko sledenje

- Vsaka sprememba nad vlogo (kreacija, sprememba, deaktivacija) se zabeleži v revizijski sistem z uporabnikom, časom in vsebino spremembe.

2.3.2.24.2 Predpogoji

- Uporabnik ima WRITE pravice za modul "vloge" oziroma celotni RBAC.
- Področje (realm), na katerega se vloga veže, mora že obstajati.
- Pravice in funkcionalnosti morajo biti definirane v katalogu.
- Revizijsko sledenje je omogočeno.

2.3.2.24.3 Merila uspešnosti

- Vloge so unikatne po imenu.
- Shranjevanje se izvede v ≤ 1 s.
- Spremembe se ne uveljavijo retroaktivno brez opozorila.
- Status vloge (aktivna/neaktivna) je jasno prikazan in upoštevan.
- Vse operacije nad vlogami so popolnoma revizijsko sledljive.

2.3.2.24.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~2800	Upravljanje vlog znotraj področja	Administrator ustvarja in upravlja z vlogami, ki uveljavljajo pravice znotraj	WRITE pravice za RBAC	Vloge so unikatne, funkcionalno popolne in konsistentno vezane

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
		funkcionalnega področja (Realm).		na področje.
~2801	Ustvarjanje nove vloge	Vnos vseh atributov vloge, pravic in funkcionalnosti.	Področje obstaja, katalog pravic/funkcionalnosti dosegljiv	Vloga je takoj na voljo za dodeljevanje uporabnikom.
~2802	Urejanje obstoječe vloge	Možnost popravkov naziva, opisa, pravic in funkcionalnosti.	Vloga obstaja	Sistem opozori, če je vloga že dodeljena uporabnikom.
~2803	Deaktivacija vloge	Vloga se označi kot neaktivna – ni več na voljo za nadaljnjo uporabo.	Vloga obstaja	Aktivne dodelitve se ohranijo, nadaljnje dodeljevanje onemogočeno.
~2804	Kopiranje vloge	Ustvarjanje nove vloge z uporabo obstoječe kot predloge.	Izbrana izvorna vloga	Nova vloga dobi novo ime, status "aktivna", enake pravice.
~2805	Pregled kataloga vlog	Prikaz vseh vlog, z možnostjo filtriranja po področju, tipu, statusu, vsebini.	-	Vloge prikazane s statusom, tipom, povezavami.

2.3.2.24.5 Diagram primerov uporabe

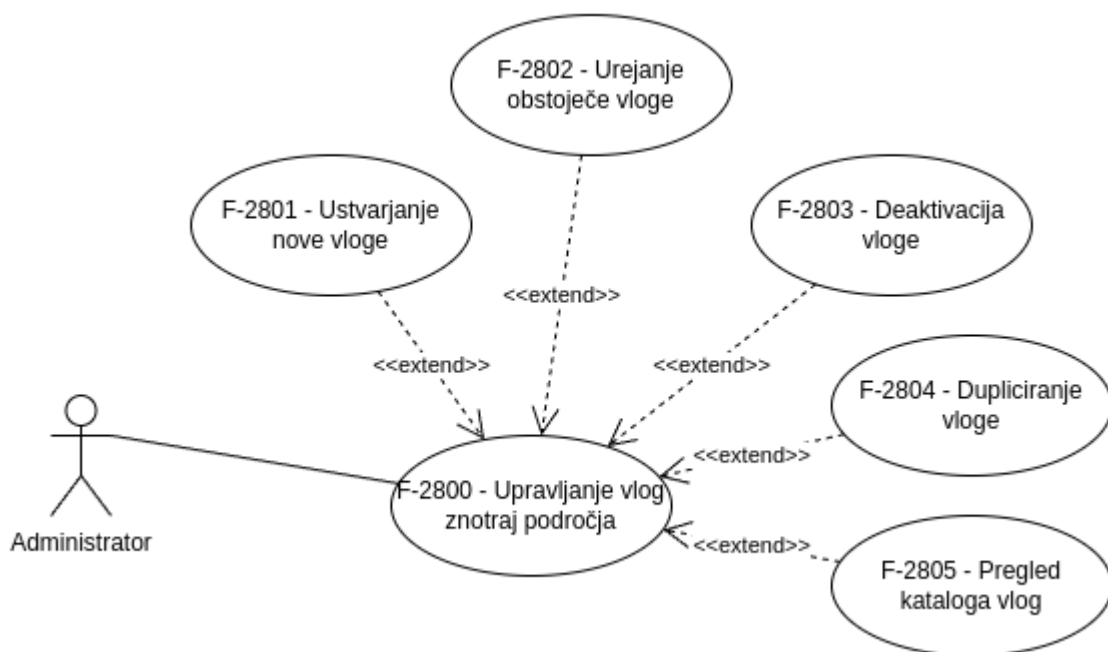


Diagram 26: Diagram primerov uporabe za SKL-VS-F-2800.

2.3.2.25 SKL-VS-F-2900: Upravljanje z vlogami zastopništva

Opis funkcionalnosti

Funkcionalnost omogoča **ustrezno avtoriziranemu administratorju**, da **znotraj obstoječega zastopništva (oseba ↔ subjekt)** upravlja **vloge (roles)**, ki določajo **pravice uporabnikov** v okviru tega zastopniškega konteksta.

Zastopništvo določa pravni okvir delovanja, vloge pa definirajo **funkcionalni obseg**, ki ga ima ta

zastopnik pri delovanju v imenu določenega subjekta. Vloge so vezane na področja (Realme) in vključujejo kombinacije pravic (READ, WRITE, DELETE, ADMIN...).

Glavne zmožnosti:

1. Dodajanje vlog zastopništvu

- Posamezno: izbor ene ali več vlog za izbrani realem.
- Masovno: dodajanje vseh vlog za določen realem, ali več realmov.
- Možnost določanja **časovne veljavnosti** vloge (neobvezno): `velja_od`, `velja_do`.

2. Odvzem vlog zastopništvu

- Posamezno: odvzem izbranih vlog.
- Masovno: odvzem vseh vlog izbranega realma ali vseh vlog hkrati.
- Obvezna je navedba **razloga za odvzem**, ki se shrani v revizijsko zgodovino.

3. Pregled trenutnega stanja vlog

- Prikaz vseh trenutno aktivnih vlog, z metapodatki:
 - naziv vloge, realem, sklop pravic,
 - časovna veljavnost (če podana),
 - status (aktivna/potekla),
 - izvor (npr. ročno dodeljena, sistemska, pogojena s funkcijo).

4. Zgodovina sprememb

- Kronološki prikaz vseh sprememb nad vlogami zastopništva:
 - kdo je dodelil ali odvzel,
 - kdaj, katero vlogo, zakaj,
 - status pred/po spremembi.

5. Varnostni mehanizmi

- Dodeljevanje sistemskih vlog (npr. ADMIN) lahko zahteva dodatno potrditev ali višjo raven avtorizacije.
- Vloge niso prenosljive med zastopništvu – dodeljene so izključno v okviru konkretnega razmerja oseba ↔ subjekt.

2.3.2.25.1 Predpogoji

- Zastopništvo (oseba ↔ subjekt) mora obstajati (glej SKL-VS-F-2400).
- Administrator ima WRITE pravice za upravljanje vlog zastopništev.
- Katalog vlog in pravic mora biti dosegljiv (glej SKL-VS-F-2800).
- Revizijsko sledenje sprememb mora biti omogočeno.

2.3.2.25.2 Merila uspešnosti

- Vloge so validirane proti katalogu in vezane na ustrezna področja.
- Časovna veljavnost je logično konsistentna ($velja_do > velja_od$).
- Masovne operacije se izvedejo brez napake in z zabeleženim povzetkom.
- Vsaka sprememba (dodelitev/odvzem) je revizijsko zabeležena s časom, uporabnikom in razlogom.
- Prikaz vlog se izvede v ≤ 500 ms.

2.3.2.25.3 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~2900	Upravljanje z vlogami zastopništva	Administrator ureja vloge znotraj obstoječega zastopništva (oseba ↔ subjekt), ki določajo funkcionalne pravice v imenu subjekta.	Zastopništvo obstaja, WRITE pravice	Vloge validirane, spremembe zabeležene, sistem ostane dosleden.
~2901	Dodeljevanje vlog	Dodajanje ene ali več vlog, ročno ali masovno po realmih; možnost nastavitve časovne veljavnosti.	Katalog vlog in pravic dosegljiv	Vloge aktivne glede na veljavnost, ni podvajanj.
~2902	Odvzem vlog	Odstranitev posamezne vloge ali več vlog hkrati; razlog za odvzem je obvezen.	Vloge morajo biti aktivne za to zastopništvo	Vloge odstranjene, revizijsko zabeležen razlog.
~2903	Pregled dodeljenih vlog	Prikaz vseh trenutno veljavnih vlog z vsemi pripadajočimi atributi.	-	Seznam ažuren, filtrabilen po realmu, statusu, vlogi.
~2904	Zgodovina sprememb nad vlogami	Kronološki vpogled v spremembe vlog: dodeljeno, odvzeto, čas, uporabnik, razlog, veljavnost.	Revizijsko sledenje aktivno	Popolna sledljivost sprememb, uporabno za nadzor in revizije.

2.3.2.25.4 Diagram primerov uporabe



Diagram 27: Diagram primerov uporabe za SKL-VS-F-2900.

2.3.3 SKL-Kom: Sklop komunikacijski modul

Komunikacijski modul predstavlja namenski sklop sistema POKK, ki skrbi za **integracijo z zunanjimi informacijskimi sistemi, pravilno in časovno usklajeno osveževanje lokalnih podatkovnih shem** ter za izpostavitve podatkov **podrejenim sistemom** preko varno dostopnih **API točk**. Pridobivanje podatkov iz zunanjih virov (npr. CRŽ– MKGP, UVHVVR) poteka centralizirano prek podatkovnega skladišča (DW) na KIS, kamor se podatki najprej sinhronizirajo preko namenskih replikacijskih API točk. Posamezni podsistemi, vključno s POKK, dostopajo do teh podatkov iz DW, skladno s svojo pristojnostjo.

Modul združuje:

- **uvozne vmesnike (pull integracije)** za sinhronizacijo podatkov iz uradnih evidenc,
- **API vmesnike** za podrejene sisteme, ki **sami izvajajo poizvedbe (pull)** po podatkih iz POKK,
- **cron mehanizme**, ki nadzorujejo in izvajajo periodične posodobitve podatkov,
- **popolno integracijo z varnostno shemo sistema**, ki zagotavlja, da so vsi podatki dostopni le glede na pravice uporabnikov ali tehničnih odjemalcev.

Ključne naloge komunikacijskega modula:

1. Uvoz podatkov iz uradnih evidenc (pull integracije)

Modul skrbi za **prenos, validacijo in shranjevanje podatkov** iz zunanjih sistemov, kot so:

- **OSEBE** – fizične osebe, registrirane v evidencah CRG, CRŽ, CRK,
- **SUBJEKTI** – pravne osebe, kmetijska gospodarstva (RKG, AKTRP),
- **ZASTOPNIŠTVA** – relacije oseb do subjektov (nosilci KMG, pooblastila),
- **GERK** – prostorske enote, ki pripadajo subjektom, z vira UVHVVR,
- **PROSTORSKI PODATKI** – naslovi, prostorske enote, parcele (povezano z evidencami GURS, MKGP),

Osveževanje se izvaja znotraj sistema kot **cron naloge**, s frekvenco, definirano v konfiguraciji (praviloma dnevno).

2. Izpostavitve podatkov podrejenim sistemom (API pull model)

POKK vsebuje **API točke**, ki omogočajo varno in strukturirano poizvedovanje po podatkih:

- Vsak zunanji sistem, ki želi dostopati do podatkov, mora imeti ustrezno avtorizacijo (preko uporabniškega računa ali tehničnega API odjemalca),
- API spoštuje **matriko pravic**, definirano v sistemu (glej SKL-VS-F-2900),
- Vmesniki omogočajo poizvedbe po osebah, subjektih, zastopništvih, GERK, pravnih osnovah itd.

POKK **ne izvaja aktivnega push-a**, temveč omogoča nadzorovano uporabo svojih podatkov drugim sistemom.

3. Integracije in povezani sistemi

- **MKGP**: dostop do RKG, CRŽ, CRK; možnost izpostavitve API za preverjanje podatkov o subjektih, GERK in dovoljenjih.
- **UVHVVR**: podatki o GERK, subjektih, evidencah zdravljenja – z opcijsko uporabo tehničnih uporabnikov.
- **VF (VOLOS)**: dostop do podatkov o uporabnikih, subjektih, zastopništvih.
- **KIS**: replikacijska uporaba podatkov POKK z namenom porazdeljene obdelave v znanstveno-raziskovalne namene.

4. Varnost in pravice

- Vsi dostopi do podatkov so zaščiteni z:
 - **avtentikacijo (certifikat/token)**,
 - **avtorizacijo glede na pravice**, ki so določene znotraj **zastopniške vloge** tehničnega uporabnika,
 - **revizijsko sledjo**, ki beleži čas, vir, namen in rezultat vsakega klica.
- API spoštuje enotno **varnostno shemo** sistema POKK.

5. Enrollment KMG (PUSH mehanizem)

Poleg sinhronizacije in API točk komunikacijski modul vključuje tudi funkcionalnost **enrollmenta kmetije (KMG)**, ki omogoča, da sistem POKK **aktivno obvesti državne sisteme MKGP/UVHVVR**, da je določen **subjekt** (kmetijsko gospodarstvo) **podal soglasje za vključitev svojih podatkov** v obdelavo znotraj sistema POKK.

Ta korak predstavlja **formalno registracijo kmetije v kontekstu sistema POKK**, in je **pogoj** za nadaljnjo avtomatsko replikacijo podatkov (npr. GERK, RKG, pooblastila, CRŽ...).

Ključne značilnosti enrollment mehanizma:

- **Sprožitev enrollmenta** nastopi:
 - ob potrditvi soglasja s strani zastopnika kmetije (glej SKL-VS-F-1900),
 - ob prvi potrjeni povezavi oseba ↔ subjekt, ki še ni bil enroled,
 - ali ročno s strani administratorja (če je to predvideno).
- **Obvestilo se pošlje** na predhodno dogovorjeno PUSH točko MKGP/UVHVVR (npr. `POST /api/enrollment/kmg`), skupaj z:
 - identifikatorjem KMG (MID, DŠ),
 - podatki o soglasju,
 - podatki o zastopniški osebi.

- Po uspešno izvedenem enrollmentu se:
 - **označi subjekt kot enroled** (status v sistemu POKK),
 - **odprejo replikacijske poti**, ki omogočajo sinhronizacijo entitet (npr. GERK, subjekti, osebe, parcelni podatki).
- V primeru neuspeha se rezultat zabeleži in sistem redno ponavlja postopek do uspeha (z omejenim številom ponovitev).
- **Enrollment ne prenaša nobenih drugih podatkov** razen izrecnega soglasja in identifikatorjev – podatkovni promet se nato nadaljuje izključno preko registriranih pull integracij.

2.3.3.1 SKL-KOM-F-5000: Enrollment (avtorizacija) KMG v sistem MKGP/UVHVVR

2.3.3.1.1 Opis funkcionalnosti

Funkcionalnost omogoča **formalni vpis (enrollment)** subjekta tipa **KMG (kmetijsko gospodarstvo)** v sistem MKGP/UVHVVR preko **namenskih API-jev**, s čimer se uradnemu informacijskemu sistemu sporoči, da je subjekt podal soglasje za sodelovanje v sistemu POKK.

Enrollment predstavlja **avtorizacijski korak**, ki je **pogoj** za nadaljnjo podatkovno interoperabilnost med POKK in uradnimi registri (RKG, CRŽ, CRK itd.).

Ključne značilnosti:

1. Sprožitev enrollmenta:

- Enrollment se izvede **sinhrono**, kot del:
 - postopka **registracije subjekta** v sistemu POKK,
 - ročne administrativne potrditve (npr. klik v UI),
 - ali druge systemske akcije (npr. potrditev soglasja uporabnika).
- Sistem preveri:
 - ali subjekt še ni enroled (glede na notranji status),
 - ali obstaja potrjeno soglasje za posredovanje podatkov.

2. Klic na MKGP/UVHVVR:

- Na podlagi internega ID-ja sistema POKK se oblikuje payload z:
 - **identifikatorjem subjekta** (npr. MID, DŠ, naziv),
 - **atributi soglasja** (verzija, datum, način podaje),
 - podatki o **zastopniku** (ime, DŠ – če so potrebni),
- Sistem pošlje sinhroni **HTTP POST** na predpisan API endpoint.

3. Prejem in obdelava odgovora:

- Sistem zabeleži status odgovora (uspeh / zavrnitev),
- Če je enrollment uspešen, se:

- v podatkovni model zapiše atribut `status_enrollment = "OK"`,
- sproži sprostitev pull replikacij (npr. aktivira cron naloge),
- V primeru napake:
 - sistem shrani odziv (koda, telo),
 - status označi kot `PENDING` ali `FAILED`,
 - omogoči ročno ponovno proženje.

4. Revizija in ponovljivost:

- Vsak poskus enrollmenta se beleži v revizijsko sled,
- Vsak subjekt ima lahko **največ en aktiven enrollment** (ponovljeni klici morajo biti dovoljeni le, če status \neq "OK").

2.3.3.1.2 Predpogoji

- Subjekt mora obstajati v sistemu kot KMG in biti označen kot primeren za enrollment.
- Obstajati mora potrjeno soglasje za posredovanje podatkov.
- Konfiguracijska povezava na MKGP/UVHVVR API mora biti dosegljiva in aktivna.
- Revizijsko beleženje mora biti omogočeno.

2.3.3.1.3 Merila uspešnosti

- Enrollment se izvede v ≤ 5 s (sinhrona komunikacija).
- Napake se zabeležijo in so obravnavane kot ponovljive (retryable).
- Po uspešnem odgovoru se atribut statusa enrollmenta ustrezno osveži.
- Revizijska sled je popolna (čas, payload, rezultat, uporabnik/sistem).

2.3.3.1.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~5000	Enrollment KMG v MKGP/UVHVVR	Sistem POKK preko sinhronega API klika na MKGP/UVHVVR prijavi subjekt, da je podal soglasje in se želi vključiti v obdelavo.	Subjekt obstaja, soglasje potrjeno, API konfiguracija aktivna	Status enrollmenta osvežen, rezultat zabeležen, napake upravljane.
~5001	Sestava podatkov za enrollment	Sistem pripravi strukturo s podatki o subjektu, soglasju in zastopniku.	Subjekt z atributi, soglasje potrjeno	Payload oblikovan skladno z zahtevo MKGP/UVHVVR.
~5002	Klic API MKGP/UVHVVR	HTTP POST klic na zunanji API za namen enrollmenta.	Zunanja točka dosegljiva, avtentikacija uspešna	Odziv zabeležen, status v sistemu posodobljen.
~5003	Obdelava odgovora in	Glede na rezultat API klika se subjektu določi status	Odziv validiran	Sistem ob uspehu aktivira nadaljnje

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
	označevanje statusa	enrollmenta (OK, FAILED, PENDING).		sinhronizacijske tokove.
~5004	Revizijsko sledenje	Vsi poskusi enrollmenta se revizijsko beležijo z natančno vsebino in rezultatom.	Revizijski sistem delujoč	Vsaka izvedba vključno z napakami je sledljiva po statusu, ID-ju in času.

2.3.3.1.5 Diagram primerov uporabe

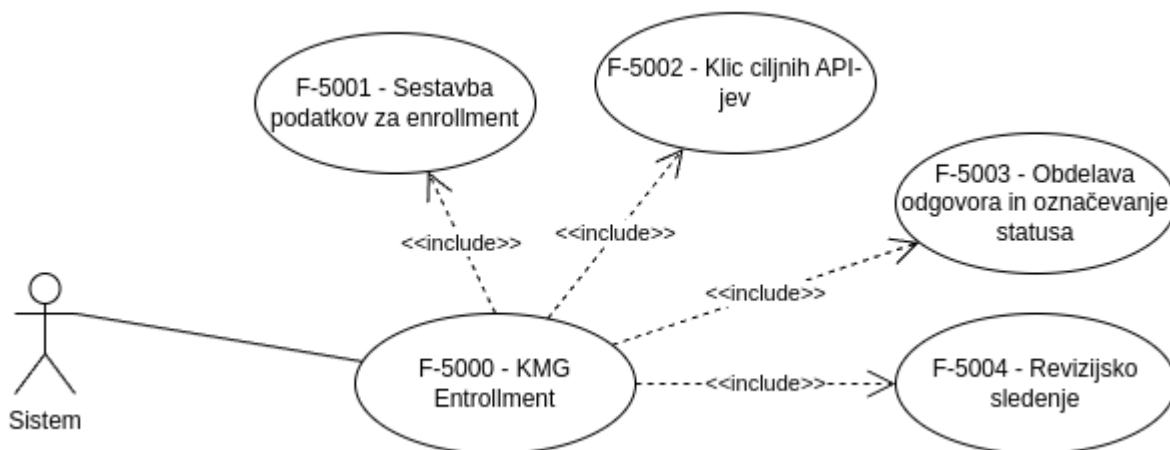


Diagram 28: Diagram primerov uporabe za SKL-KOM-F-5000.

2.3.3.2 SKL-KOM-F-5100: Sinhronizacija RKG

2.3.3.2.1 Opis funkcionalnosti

Funkcionalnost omogoča sinhronizacijo podatkov iz **Registrske kmetijske evidence (RKG)** preko uradnih API vmesnikov MKGP/UVHVVR.

Podatki se prenašajo postopoma, po spremembah, z uporabo parametra *posodobljeno_od*, in se vedno obdelujejo v zaporedju od najnovejših do najstarejših sprememb (sortirano padajoče po datumu posodobitve).

Sistem omogoča **avtomatiziran prenos** podatkov preko konfigurabilnih **cron nalog** ter opsijsko **ročno sprožitev ponovnega poizkusa**, v primeru napake ali odpovedi sinhronizacije.

Podprti tipi podatkov iz RKG:

- **Subjekti** (pravne in fizične osebe, nosilci kmetij),
- **KMG** (kmetijska gospodarstva z MID identifikacijo),
- **GERK** (grafične enote rabe kmetijskih zemljišč),
- **Kmetijska zemljišča in parcele**,
- **Povezane osebe in vloge v subjektih (nosilci, uporabniki).**

Potek sinhronizacije:

1. Pridobivanje sprememb

Sistem periodično pošilja klic na RKG API z zadnjim znanim *posodobljeno_od*.

Prejme nabor zapisov, razvrščenih po *posodobljeno* v padajočem vrstnem redu.

2. Uvoz in validacija

Vsak prejeti zapis se validira in pripravi za uvoz.

V primeru, da je katerikoli zapis neustrezen, se celoten paket **ne uvozi** – uveljavljen je model **transakcijske konsistence (vse ali nič)**.

3. Zapis sinhronizacijskega rezultata (job log)

Ob vsakem poskusu sinhronizacije se zabeležijo:

- čas izvedbe,
- število uspešnih in neuspešnih zapisov (če je uvoz delno pripravljen, a ni zaključen),
- tehnični ID sinhronizacijske naloge,
- status (OK, FAILED, SKIPPED),
- ID-je prejetih zapisov (če so na voljo),
- čas posodobljeno_od in posodobljeno_do.

4. Ponovljivost in ročni ponovni poizkus

V primeru napake se sistemska sinhronizacija samodejno označi kot **neuspešna**, podatki se ne shranijo, status ostane nespremenjen.

Administrator lahko iz UI sproži **ponovni poizkus sinhronizacije**, ki uporabi isti časovni žig kot zadnji neuspešen poskus.

5. Revizijska sled

Vsak poskus sinhronizacije je popolnoma zabeležen z metapodatki in odzivom zunanjega sistema.

Povzetek sinhronizacije je prikazan v uporabniškem vmesniku (zgodovina nalog).

2.3.3.2.2 Predpogoji

- Povezava do RKG API mora biti aktivna in ustrezno konfigurirana.
- Sistem mora imeti zapisan čas zadnje uspešne sinhronizacije.
- Uporabljeni API-ji morajo podpirati razvrščanje po datumu **posodobljeno** in filtriranje po časovnem žigu.
- Revizijsko sledenje mora biti omogočeno.
- Cron mehanizem mora biti konfiguriran.

2.3.3.2.3 Merila uspešnosti

- Prenos se zaključi z zapisom statusa sinhronizacije v ≤ 10 s.
- V primeru napake se podatki **ne shranijo** (vse ali nič).
- Vsaka sinhronizacija beleži število obdelanih enot, ID-je in status.
- Sistemska zgodovina sinhronizacij vsebuje vsaj zadnjih 100 zabeleženih prenosov.
- Administrator lahko kadarkoli sproži ponovni prenos.

2.3.3.2.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~5100	Sinhronizacija RKG	Prenos podatkov iz RKG preko MKGP API z uporabo mehanizma »posodobljeno_od« in celovito beleženje rezultatov.	API konfiguriran, časovni žig na voljo, cron ali ročni zagon	Vse ali nič sinhronizacija, statusno označeno, revizijsko zabeleženo.
~5101	Obdelava RKG podatkov po entitetah	Sistem ločeno obdeluje subjekte, GERK, zemljišča, KMG in nosilce.	Razlikovanje po endpointih ali tipu podatka	Obdelava enot po tipu, validacija in transakcijska obdelava.
~5102	Beleženje rezultata sinhronizacije	Zapis o izvajanju sinhronizacije (job), z metapodatki in statusi.	Dostop do interne evidence jobov	Povzetek je dostopen v UI in API.
~5103	Ponovni poizkus sinhronizacije	Ročna sprožitev ponovnega prenosa po neuspešni sinhronizaciji.	UI ali API funkcionalnost za ponovno proženje	Uporabi enak posodobljeno_od, rezultat se beleži kot nova naloga.
~5104	Revizijsko sledenje	Vsaka sinhronizacija in vsak zapis se beleži s statusom, časom in rezultatom.	OpenSearch ali lokalni revizijski sistem	Vsaj osnovni metapodatki: kdaj, kdo (cron/sistem), status, število zapisov.

2.3.3.2.5 Diagram primerov uporabe

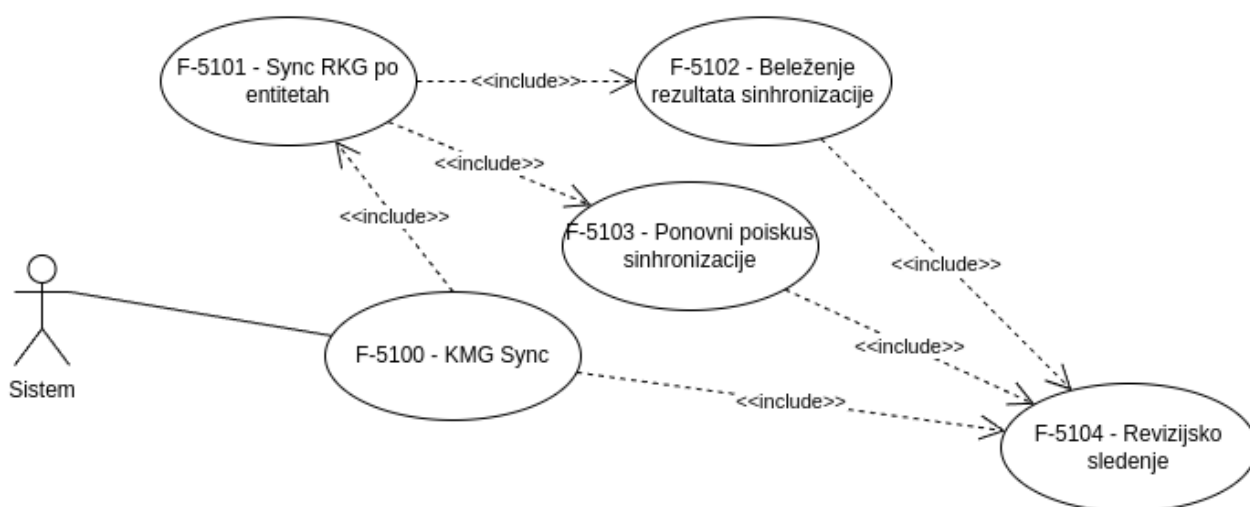


Diagram 29: Diagram primerov uporabe za SKL-KOM-F-5100.

2.3.3.3 SKL-KOM-F-5200: Sinhronizacija CRŽ/CRK

2.3.3.3.1 Opis funkcionalnosti

Funkcionalnost omogoča **periodično sinhronizacijo podatkov o živalih** iz uradnih registrov **CRŽ (Centralni register živali)** in **CRK (Centralni register kopitarjev)** preko uradnih API vmesnikov, ki jih zagotavljata MKGP in/ali UVHVVR.

Sinhronizacija poteka po principu **časovnega zajema (delta pull)**, kjer sistem POKK v določenem časovnem intervalu pridobiva **vse spremenjene podatke od zadnjega uspešnega prenosa dalje**.

Podatki so urejeni po **posodobljeno** v padajočem vrstnem redu. Vsak prenos se izvaja kot

transakcijsko zaprt blok (vse ali nič). V primeru neuspeha se podatki ne shranijo, rezultat pa se beleži kot **neuspešna sinhronizacija**.

Vključeni podatkovni sklopi:

1. ŽIVALI

- osnovni identifikacijski podatki:
 - številka (ušesna ali druga),
 - vrsta živali,
 - datum rojstva,
 - status (aktivna, neaktivna, zaklana, poginula),
 - KMG lastništvo / skrbništvo.

2. IDENTIFIKATORJI

- vse oznake, dodeljene posamezni živali:
 - primarna ušesna številka,
 - alternativni identifikatorji (čipi, mikročipi, transponderji, pasmične številke),
 - časovna veljavnost posamezne oznake.

3. DOGODKI

- pomembni statusni dogodki, vezani na žival:
 - registracija,
 - sprememba statusa,
 - sprememba lokacije,
 - veterinarski dogodki (če so zajeti v CRŽ/CRK),
 - odjave in aktivacije.

4. PREMIKI

- beleženje selitev živali med KMG enotami:
 - datum premika,
 - izvorna in ciljna KMG (MID),
 - razlog premika,
 - potrdila o izvedbi.

Potek sinhronizacije:

1. Sprožitev sinhronizacije

- avtomatsko preko `cron` nalog,
- ali ročno (administrativni nadzor),

- s časovnim parametrom `posodobljeno_od` glede na zadnji uspešen prenos.

2. Pridobivanje podatkov

- sistem pošlje sinhroni klic na ustrezen API za CRŽ in CRK (ločeno ali združeno, odvisno od strukture),
- API vrne zapise razvrščene po `posodobljeno` (padajoče).

3. Validacija in transakcijski uvoz

- vsak zapis se validira po strukturi in vsebini,
- če pride do napake (npr. manjkajoč MID, neveljaven ID), se celoten paket zavrne.

4. Beležba sinhronizacijske naloge

- zabeležijo se:
 - časovni okvir,
 - število obdelanih zapisov po sklopih (živali, identifikatorji, dogodki, premiki),
 - status (OK, FAILED, SKIPPED),
 - seznam napak (če delno analizirano),
 - oznaka vira (CRŽ ali CRK).

5. Ročni ponovni poizkus

- omogočen v UI,
- uporabi isti `posodobljeno_od` časovni žig kot prejšnji neuspešen prenos.

6. Revizijska sled

- zabeleženi so:
 - vsi vhodni podatki,
 - rezultat validacije in transformacije,
 - razlog za neuspeh (če obstaja).

2.3.3.3.2 Predpogoji

- Povezava do CRŽ/CRK API mora biti vzpostavljena in ustrezno avtorizirana.
- V sistemu mora biti zabeležen čas zadnjega uspešnega prenosa.
- Cron naloge ali ročni nadzor morajo biti omogočeni.
- Revizijsko beleženje mora biti vključeno.

2.3.3.3.3 Merila uspešnosti

- Vsi podatki se prenesejo in uvozijo v ≤ 10 s na 1000 zapisov.
- Če je prenos neuspešen, se podatki ne shranijo (atomicnost).
- Vsaka entiteta je validirana ločeno; neuspešni paketi se zavrnejo in zabeležijo.

- Vsaka uspešna sinhronizacija osveži notranji časovni žig.
- Napake so revizijsko sledljive in odpravljive s ponovnim poizkusom.

2.3.3.3.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~5200	Sinhronizacija CRŽ/CRK	Periodični prenos podatkov o živalih, identifikatorjih, dogodkih in premikih iz registrov CRŽ/CRK.	API aktiven, časovni žig na voljo	Obdelava vseh sklopov, statusni pregled po nalogi, revizijska sled.
~5201	Obdelava po sklopih	Obdelava entitet: živali, identifikatorji, dogodki, premiki; ločeno po tipu.	Tip zapisa določen ali razpoznan	Vsaka entiteta obdelana znotraj svojega modula.
~5202	Beleženje rezultata sinhronizacije	Če je katerikoli zapis neveljaven, se celoten prenos zavrne in ne uveljavi. Rezultat se zabeleži.	Validacija podatkov pred zapisom	Vse ali nič model – brez delnih zapisov. Zapis rezultata sinhronizacije.
~5203	Ponovni poizkus sinhronizacije	Ročno ali samodejno proženje ponovnega prenosa na osnovi predhodnega neuspeha.	UI ali API omogoča retry funkcionalnost	Enak časovni žig uporabljen, novi status zabeležen.
~5204	Revizijsko beleženje	Vsaka sinhronizacija beleži: čas, rezultat, vir, število zapisov, napake (če obstajajo).	Revizijski sistem aktiven	Sinhronizacijski log prikazan v administracijskem pregledu.

2.3.3.3.5 Diagram primerov uporabe

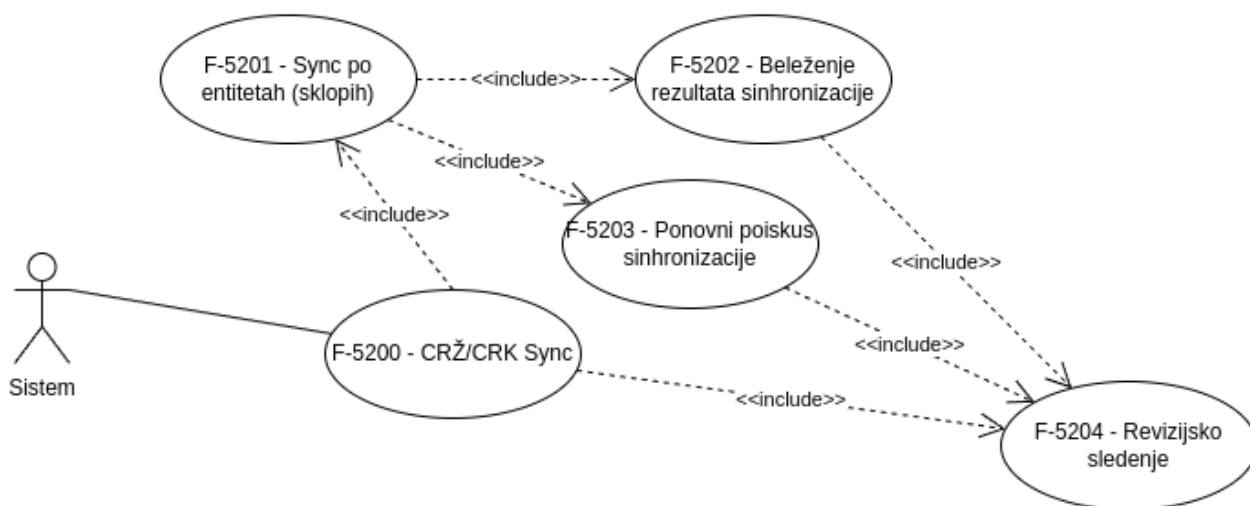


Diagram 30: Diagram primerov uporabe za SKL-KOM-F-5200.

2.3.3.4 SKL-KOM-F-5300: Sinhronizacija šifrantov (GERK, Naslovi, Prostor, Enote, ...)

2.3.3.4.1 Opis funkcionalnosti

Funkcionalnost omogoča **sinhronizacijo vseh prostorskih, naslovnih in splošnih šifrantov**, ki se uporabljajo v sklopih POKK, in sicer iz uradnih evidenc MKGP, UVHVVR in GURS.

Prenos poteka po principu **delta sinhronizacije** (na osnovi posodobljeno_od), v kombinaciji z možnostjo **periodične popolne obnove**, kjer je to potrebno (npr. pri nestabilnih šifrantskih virih).

Vključeni šifranti in prostorski sklopi:

1. GERK (grafične enote rabe kmetijskih zemljišč)

- GERK ID, MID, površina, datum veljavnosti, geometrija (WKT/GeoJSON/GML),
- povezava z vrsto rabe in subjektom (MID),
- status GERK (aktiven, neaktiven).

2. Vrsta rabe (šifrant)

- Kodna oznaka, naziv, hierarhija (npr. kmetijska, gozdna),
- klasifikacija po MKGP šifrantu,
- povezljivost z GERK.

3. Naslovi

- Ulica, hišna številka, naselje, občina, poštna številka,
- geografske koordinate (centroid, stavba),
- EŠD (enota stavbnega dela) in ID stavbe, če dostopno.

4. Prostorske enote

- Upravne enote, statistične regije, katastrske občine,
- povezava z naslovi in KMG,
- klasifikacije za potrebe filtriranja, regionalne razmejitve.

5. Merske enote (šifrant)

- Enote površine (m², ha), dolžine, količine,
- povezava z drugimi podatkovnimi sklopi (GERK, raba...).

6. Splošni šifranti

- Tabele vrednosti, ki se uporabljajo v prostorskih kontekstih ali drugih evidencah:
 - statusi, klasifikacije, tipi evidenc,
 - razlogi za spremembe (npr. deaktivacija GERK, sprememba naslova).

Potek sinhronizacije:

1. Sprožitev (cron/ročno)

- Periodično (npr. dnevno/tedensko),
- ali ročno ob potrebi za prisilno osvežitev.

2. Pridobivanje podatkov

- Preko uradnih API točk (MKGP, UVHVVR, GURS),
- za posamezne sklope posebej (modularna izvedba).

3. Validacija in transakcijski zapis

- Vsi podatki so validirani po strukturi in vsebini,
- ob napaki se celoten prenos zavrne (model vse ali nič),
- zapis o neuspehu z metapodatki in razlogom.

4. Beležba sinhronizacijske naloge

- Zabeleži se:
 - vir (npr. GURS, MKGP),
 - tip sinhronizacije (šifrant, prostor),
 - status (OK, FAILED),
 - število prenešenih zapisov,
 - ID-ji in oznake posodobljenih enot.

5. Ročni ponovni poizkus

- Na voljo v UI,
- omogoča ponovitev sinhronizacije za določen sklop.

6. Povezljivost

- Vsi šifranti so povezani s funkcionalnimi sklopi sistema:
 - GERK → subjekt (MID),
 - naslov → oseba/subjekt,
 - prostorske enote → filtriranje, prikaz,
 - merske enote → validacija podatkov v evidencah.

2.3.3.4.2 Predpogoji

- API povezave z uradnimi registri aktivne (GURS, MKGP, UVHVVR),
- notranja evidenca zadnjega časovnega žiga,
- aktivna cron konfiguracija ali ročni zagon,
- podpora za prostorske podatke (geometrija, referenčni sistem).

2.3.3.4.3 Merila uspešnosti

- Vsaka entiteta validirana ločeno,

- Transakcijski model prepreči delno osvežitev,
- Čas prenosa ≤ 10 s na 1000 zapisov,
- V UI je na voljo seznam zadnjih izvedb po sklopih,
- Šifranti so povezani z entitetami, kjer so uporabljeni.

2.3.3.4.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~5300	Sinhronizacija šifrantov	Periodični uvoz šifrantov za GERK, naslove, prostorske enote, merske enote in splošne klasifikacije.	Aktivne povezave do GURS/MKGP, časovni žig	Transakcijski uvoz, metapodatki, povezljivost z ostalimi entitetami.
~5301	Uvoz GERK s prostorsko komponento	Uvoz GERK enot s statusi, površinami, MID, rabo in geometrijo.	API aktiven, struktura znana	GERK podatki pravilno povezani z MID, validirane geometrije.
~5302	Sinhronizacija naslovnih podatkov	Uvoz in osvežitev naslovov, stavb, naselij, EŠD, katastrskih enot.	API (GURS ali MKGP) aktiven	Naslovi pravilno strukturirani in povezani z osebami/subjekti.
~5303	Uvoz vrst rabe in merskih enot	Osvežitev šifranta vrste rabe in merskih enot za potrebe podatkovne standardizacije.	API dosegljiv, struktura enot skladna	Vrednosti pravilno uporabljene pri uvozu GERK in v funkcionalnih modulih.
~5304	Beleženje rezultatov	Če je katerikoli zapis neveljaven, se celoten prenos zavrne in ne uveljavi. Rezultat se zabeleži.	Validacija podatkov pred zapisom	Vse ali nič model – brez delnih zapisov. Zapis rezultata sinhronizacije.

2.3.3.4.5 Diagram primerov uporabe

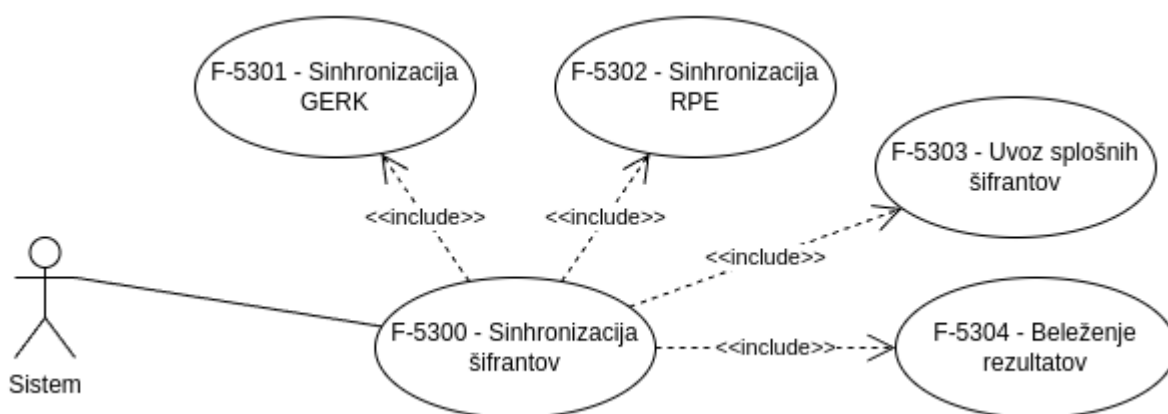


Diagram 31: Diagram primerov uporabe za SKL-KOM-F-5300.

2.3.3.5 SKL-KOM-F-5700: Sinhronizacija CRK, CRŽ, RKG za podan RKG (MID)

2.3.3.5.1 Opis funkcionalnosti

Funkcionalnost omogoča ciljno sinhronizacijo podatkov iz državnih evidenc **RKG, CRŽ in CRK** za konkretno **kmetijsko gospodarstvo (KMG)**, identificirano z MID.

Glavni namen je, da se ob **enrollmentu novega KMG** v sistem POKK takoj prenesejo vsi **ključni podatki** (subjekt, GERK, živali, nosilci, dogodki...), ne da bi bilo treba sprožiti polno sinhronizacijo vseh podatkov iz vseh registrov.

Ta funkcionalnost deluje kot **selektivna delta sinhronizacija na osnovi MID**, in je primerna tudi za:

- ponovno sinhronizacijo posameznega KMG (npr. ob napaki),
- izredne administrativne posege (npr. ročno sproženo).

Vsebinski obseg podatkov (za podan MID):

1. RKG

- Subjekt (MID, naziv, status, naslov),
- GERK (enote, površine, vrsta rabe, geometrija),
- Nosilci in povezane osebe.

2. CRŽ / CRK

- Živali registrirane na podanem MID,
- Premiki, dogodki, statusi (registracija, pogin, prodaja),
- Identifikatorji živali (ušesne številke, čipi...).

Potek sinhronizacije:

1. Sprožitev

- Samodejno ob uspešnem enrollmentu KMG (glej F-5000),
- Ročno (prek UI ali administrativnega orodja).

2. Izvedba ciljanih klicev

- Sistem izvede sinhroni dostop do:
 - RKG API → subjekt + GERK za MID,
 - CRŽ / CRK API → živali in dogodki vezani na MID.

3. Obdelava in zapis

- Preneseni podatki se validirajo,
- Uporablja se transakcijski zapis (vse ali nič),
- Podatki se označijo z virom in časom prenosa.

4. Beležba izvedbe

- Vsaka takšna sinhronizacija se zabeleži z:
 - MID,
 - tipom sinhronizacije (ciljna / ob enrollmentu),
 - številom entitet,

- statusom (OK / FAILED).

5. Ločitev od sistemske sinhronizacije

- Ciljna sinhronizacija **ne vpliva na globalni časovni žig** sinhronizacije RKG, CRŽ ali CRK,
- Namenjena je le za **vklučitev posameznega KMG** v trenutno podatkovno stanje.

2.3.3.5.2 Predpogoji

- Subjekt mora imeti znan in preverjen MID.
- API-ji RKG, CRŽ, CRK morajo omogočati iskanje po MID.
- Sistem mora imeti omogočeno administrativno ali sistemsko sprožanje selektivnih sinhronizacij.
- Revizijsko sledenje mora biti aktivno.

2.3.3.5.3 Merila uspešnosti

- Podatki se prenesejo v ≤ 10 s po entiteti (pod pogojem razpoložljivosti).
- V primeru napake se prenos zavrne in ne spremeni stanja sistema.
- Sinhronizacija je zabeležena z MID, časom in rezultatom.
- Izvedba ne vpliva na ostale sinhronizacijske tokove.

2.3.3.5.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~5700	Sinhronizacija podatkov za podan RKG	Ciljna sinhronizacija podatkov iz RKG, CRŽ in CRK za izbrano kmetijo (MID), ob enrollmentu ali po potrebi.	MID znan, API dostopen	Selektivni prenos, podatki validirani in označeni, revizijsko sledljivi.
~5701	Samodejna sinhronizacija	Ob uspešnem vpisu KMG v sistem se sproži avtomatski prenos vseh pripadajočih podatkov iz CRŽ, CRK, RKG.	F-5000 izveden	Podatki na voljo takoj za uporabo znotraj sistema POKK.
~5702	Ročna sinhronizacija	Administrator lahko sproži ciljni prenos po MID, npr. za preverjanje ali ob napaki.	UI ali orodje omogoča vnos MID	Izvedba zabeležena in ločena od globalnega žiga sinhronizacij.

2.3.3.5.5 Diagram primerov uporabe

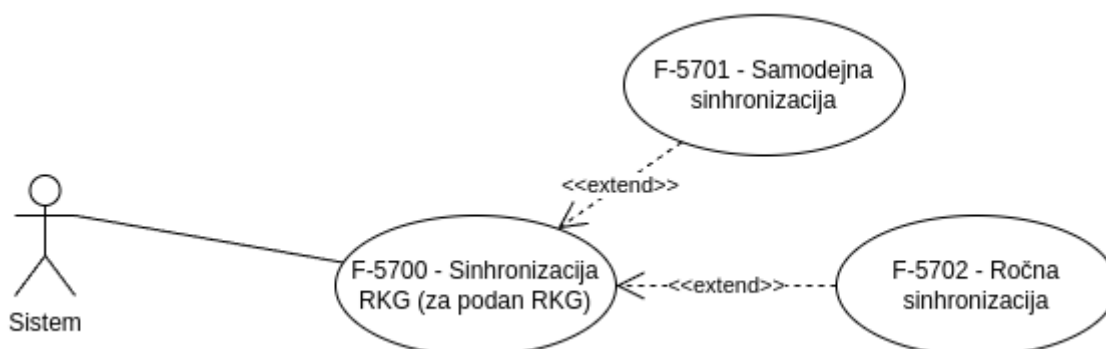


Diagram 32: Diagram primerov uporabe za SKL-KOM-F-5700.

2.3.4 SKL-DW: Sklop podatkovno skladišče

2.3.4.1 SKL-DW-F-7000: Polni zajem (Full Snapshot Extract)

2.3.4.1.1 Opis funkcionalnosti

Funkcionalnost omogoča **periodično ali ročno izvedbo popolnega zajema** podatkov iz **operativnih podatkovnih shem sistema POKK** v namensko **Extract shemo**, ki je osnova za nadaljnjo obdelavo, analitiko in poročanje v okviru podatkovnega skladišča (DW).

Polni zajem (Snapshot Extract) pomeni, da se:

- **počistijo vsi relevantni objekti** v Extract shemi (staging area),
- **zajame popolno stanje** vseh podatkovnih tabel v sistemu v trenutku izvajanja,
- **ustvari točka stanja ("snapshot")** vseh evidenc na določen datum/čas.

Glavne značilnosti:

1. Proženje zajema

- **Periodično** glede na konfiguracijo (npr. vsak X dni, tednov, mesecev),
- **Ročno (on-demand)** na zahtevo administratorja.

2. Čiščenje pred zajemom

- Pred vsakim polnim zajemom se **počistijo vsi podatki** v ciljni Extract shemi, razen kontrolnih tabel,
- Zagotovljena konsistentnost – ni mešanja starih in novih podatkov.

3. Zajem podatkov

- Vse entitete sistema POKK, relevantne za DW:
 - Subjekti, osebe, uporabniki,
 - Živali, GERK, prostorske enote,
 - Zastopništva, soglasja, premiki, zgodovina,
 - Sinhronizacije, revizije, šifranti,
- Zajem se izvaja ločeno za vsako entiteto,
- Transakcijska konsistenca na ravni posamezne entitete.

4. Upravljanje prenosa

- Ob vsakem zajemu se:
 - **kreira zapis v kontrolni tabeli**,
 - shrani datum/čas zajema,
 - shrani število prenesenih zapisov po tabelah,
 - evidentira status prenosa (OK / FAILED).

5. Revizijska sled

- Vsak zajem ima popolno revizijsko evidenco:
 - čas začetka, čas konca,
 - status vsake tabele (uspeh/neuspeh),
 - število zapisov.

6. Napake in obnovitev

- Če pride do napake med prenosom posamezne entitete:
 - se prenos prekine,
 - Extract shema ostane prazna,
 - status prenosa se označi kot FAILED.

Namen funkcionalnosti:

- Omogoča stabilne **analitične izvlečke** stanja sistema na določen dan,
- Podpira dolgoročno **sledljivost populacijskih, prostorskih in operativnih sprememb**,
- Uporablja se kot **osnova za poročanje, BI analitiko, raziskave**.

2.3.4.1.2 Predpogoji

- Extract shema mora biti pripravljena in dostopna,
- Na voljo morajo biti pravice za brisanje in vpis podatkov v Extract shemo,
- Konfiguracija časovnega proženja mora biti definirana,
- Aktivno revizijsko sledenje za prenose.

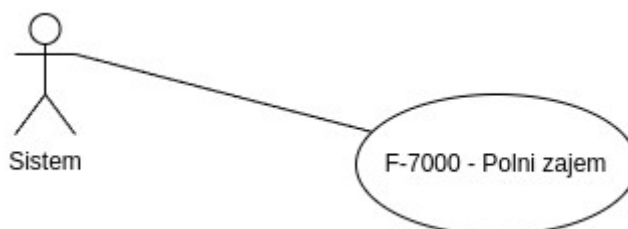
2.3.4.1.3 Merila uspešnosti

- Celotni zajem vseh evidenc zaključen v ≤ 60 minut za povprečno količino podatkov (nastavljivo glede na obseg),
- Revizijsko beleženje popolno za vsak zajem,
- V primeru napake Extract shema ostane prazna,
- Ni mešanja stare in nove vsebine.

2.3.4.1.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~7000	Polni zajem (Full Snapshot Extract)	Periodično ali on-demand čiščenje in popoln zajem vseh podatkovnih entitet v Extract shemo za potrebe podatkovnega skladišča.	Extract shema pripravljena, pravice, aktivno revizijsko sledenje	Podatki konsistentni, status prenosa zabeležen, napake preprečijo delno uvožene podatke.

2.3.4.1.5 Diagram primerov uporabe



2.3.4.2 SKL-DW-F-7100: Incremental zajem (Delta Extract)

2.3.4.2.1 Opis funkcionalnosti

Funkcionalnost omogoča **pogostejšo, optimizirano osvežitev podatkov** v Extract shemi podatkovnega skladišča POKK s prenosom **samo novih ali spremenjenih zapisov** od zadnjega uspešnega prenosa dalje.

Incremental zajem (Delta Extract) deluje brez brisanja obstoječih podatkov – **obstoječa Extract shema se ohranja**, in k njej se dodajajo samo **novi ali spremenjeni podatki**.

Delta zajem je namenjen za:

- pogostejšo sinhronizacijo (npr. urni, dnevni prenos),
- zmanjšanje obremenitve na sistemih,
- hitrejšo pripravo podatkov za analitiko.

Glavne značilnosti:

1. Proženje zajema

- **Periodično** (npr. vsako uro, vsak dan, odvisno od nastavitev),
- **Ročno** (on-demand) v primeru nujnih osvežitev.

2. Identifikacija novih/spremenjenih podatkov

- Na podlagi sistemskih časovnih oznak (posodobljeno_od, updated_at),
- Uporabi se **čas zadnjega uspešnega prenosa**, zapisan v kontrolni tabeli.

3. Pridobivanje in zapis podatkov

- Poizvedbe so filtrirane glede na posodobljeno_od,
- Novi podatki se **dodajo v Extract shemo**,
- Možno je nadgraditi zapis, če obstaja že isti ključ (npr. UPDATE logika na osnovi primarnega ključa).

4. Upravljanje prenosa

- Vsak zajem ustvari:
 - zapis v kontrolni tabeli,
 - status prenosa (OK / FAILED),
 - število obdelanih zapisov po entitetah,
 - časovni žig novega prenosa.

5. Revizijska sled

- Zabeležijo se:
 - podatki o obdelanih tabelah,
 - število novih in spremenjenih zapisov,

- napake, če se pojavijo.

6. Napake in obnovitev

- Če pride do napake pri zajemu določene entitete:
 - se prenos za to entiteto prekine,
 - preostale entitete se lahko obdelajo (če to konfiguracija dovoljuje),
 - status prenosa označen kot PARTIAL ali FAILED.

Namen funkcionalnosti:

- Omogoča **sprotno osveževanje** podatkovnega skladišča,
- Podpira analitiko, kjer je potrebno **hitro odzivanje** na spremembe,
- Zmanjšuje sistemsko obremenitev v primerjavi s polnimi zajemi.

2.3.4.2.2 Predpogoji

- Extract shema mora biti inicializirana s predhodnim polnim zajemom,
- Kontrolna tabela za beleženje prenosa mora biti vzpostavljena,
- Časovna oznaka zadnjega uspešnega prenosa mora biti dostopna,
- Aktivno revizijsko sledenje za ETL proces.

2.3.4.2.3 Merila uspešnosti

- Vsak prenos podatkov dopolnjuje Extract shemo brez izgube obstoječih podatkov,
- V primeru neuspeha se status prenosa jasno zabeleži,
- Novi podatki so v Extract shemi na voljo v ≤ 15 min po zajemu,
- Kontrolna tabela se konsistentno osveži po vsaki izvedbi.

2.3.4.2.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~7100	Incremental zajem (Delta Extract)	Periodična ali ročna osvežitev podatkov v Extract shemi s prenosom samo novih ali spremenjenih zapisov od zadnjega prenosa.	Extract shema vzpostavljena, čas zadnjega prenosa shranjen	Novi podatki dodani brez brisanja, status prenosa beležen, revizijsko sledenje popolno.

2.3.4.2.5 Diagram primerov uporabe



Diagram 34: Diagram primerov uporabe za SKL-DW-F-7100.

2.3.4.3 SKL-DW-F-7200: Upravljanje prenosov in zgodovine (Control Management)

2.3.4.3.1 Opis funkcionalnosti

Funkcionalnost omogoča **centralizirano upravljanje in beleženje vseh prenosov** podatkov v Extract shemo podatkovnega skladišča POKK, tako za **polne** kot **incrementalne** zajeme.

Sistem beleži ključne metapodatke vsakega prenosa, omogoča nadzor nad stanjem, ponavljanje neuspešnih prenosov, analizo učinkovitosti in sledenje zgodovini prenosa podatkov.

Glavne značilnosti:

1. Kontrolna tabela za prenose

- Za vsak prenos se shrani:
 - **ID prenosa** (enolični identifikator),
 - **tip prenosa** (FULL / DELTA),
 - **čas začetka** in **čas konca** prenosa,
 - **status prenosa** (OK, FAILED, PARTIAL),
 - **število obdelanih zapisov** po entitetah,
 - **časovni žig** zajema (`extract_timestamp`).

2. Beležba entitetnih statistik

- Za vsako entiteto v prenosu (tabela) se zabeleži:
 - ime entitete,
 - število uspešno prenesenih zapisov,
 - število napak (če pride do delnih težav),
 - čas obdelave posamezne entitete.

3. Statusni nadzor

- Omogočen pregled trenutnega stanja vseh prenosov:
 - seznam zadnjih X prenosov,
 - filtriranje po tipu, statusu, času.

4. Upravljanje neuspešnih prenosov

- Sistem omogoča:
 - ročni **retry** neuspešnega prenosa,
 - avtomatsko obveščanje o neuspehu (če konfigurirano).

5. Analitika in poročanje

- Omogočena izdelava osnovnih analitičnih poročil:
 - povprečen čas prenosa,

- delež uspešnih/neuspešnih prenosov,
- trend števila spremenjenih zapisov.

6. Omejevanje zgodovine

- Sistem vzdržuje omejeno zgodovino:
 - npr. zadnjih 365 dni prenosov,
 - starejši podatki se lahko arhivirajo ali brišejo po pravilniku.

7. Revizijska sled

- Vsi administrativni posegi v kontrolo prenosov (retry, brisanje) se revizijsko beležijo.

Namen funkcionalnosti:

- Zagotavlja **sledljivost** vseh ETL operacij v DW,
- Omogoča **hiter nadzor** nad prenosnim stanjem,
- Povečuje **zanesljivost podatkovnega skladišča** s popolnim beleženjem operacij.

2.3.4.3.2 Predpogoji

- Kontrolna shema ali tabela mora biti vzpostavljena,
- Sistem mora podpirati branje in zapis metapodatkov o prenosih,
- Aktivno revizijsko sledenje.

2.3.4.3.3 Merila uspešnosti

- Vsak prenos je zapisan s popolnim setom metapodatkov,
- Stanje prenosov je vedno ažurno dostopno,
- Sistem omogoča hitro identifikacijo neuspešnih prenosov,
- Revizijska sled za administrativne operacije je popolna.

2.3.4.3.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~7200	Upravljanje prenosov in zgodovine	Centralno beleženje in nadzor nad vsemi polnimi in inkrementalnimi prenosi v Extract shemo.	Kontrolna shema/tabela vzpostavljena	Vsak prenos popolnoma zabeležen, statusno spremljanje omogočeno, revizijsko sledljiv.
~7201	Upravljanje neuspešnih prenosov	Omogoča ročno ali avtomatsko ponovno sprožitev neuspešnih prenosov.	UI ali orodje omogoča retry funkcijo	Rezultat ponovljenega prenosa ločeno zabeležen.
~7202	Analitika prenosov	Omogoča osnovna poročila o trajanju, uspešnosti in obsegu prenosov.	Podatki o prenosih na voljo	Pregled povprečnega časa, trenda sprememb, deleža uspešnih/neuspešnih prenosov.

2.3.4.3.5 Diagram primerov uporabe

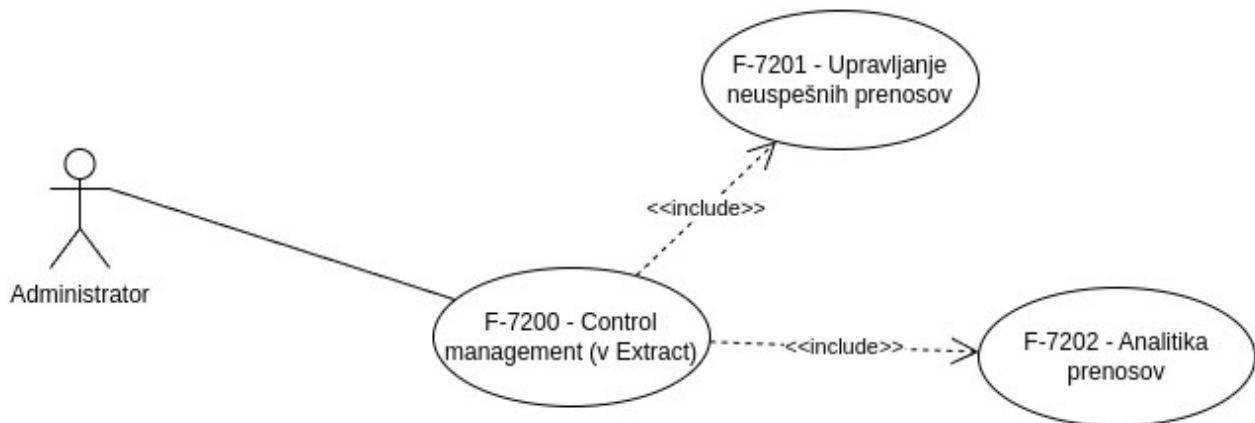


Diagram 35: Diagram primerov uporabe za SKL-DW-F-7200.

2.3.4.4 SKL-DW-F-7300: Čiščenje in standardizacija podatkov

2.3.4.4.1 Opis funkcionalnosti

Funkcionalnost omogoča **čiščenje in standardizacijo vseh podatkov**, ki so bili preneseni v Extract shemo podatkovnega skladišča POKK v sklopu polnih ali incremental zajemov.

Primarni cilj čiščenja in standardizacije je:

- zagotavljanje **konsistence** podatkov,
- odprava **nepravilnih, manjkajočih ali neoblikovanih vrednosti**,
- priprava podatkov na nadaljnjo fazo transformacije in modeliranja (dimenzijsko oblikovanje, agregacije).

Glavne značilnosti:

1. Identifikacija nepravilnih podatkov

- Prepoznavanje in označevanje zapisov z:
 - manjkajočimi ključnimi vrednostmi (npr. MID, DŠ),
 - nepravilnimi formati datumov, številke,
 - neveljavnimi referencami (npr. povezava na neobstoječ subjekt),
 - praznimi ali nekonzistentnimi besedilnimi polji (npr. dvojnimi presledki, nedovoljenimi znaki).

2. Popravljanje podatkov

- Samodejno:
 - nastavitev privzetih vrednosti za določena polja (npr. "neznano", 0),
 - obrezovanje presledkov, enotna velikost črk (npr. uppercase za šifre),
 - konverzija formatov (npr. standardizacija datuma v YYYY-MM-DD),
 - konsolidacija podobnih vrednosti (npr. »govedo« ↔ »Govedo«).

3. Standardizacija kod in klasifikacij

- Usmeritev vrednosti v uradne šifrante in klasifikacije:
 - vrst rabe (GERK),
 - vrst živali (CRŽ/CRK),
 - pravnih statusov subjektov.

4. Beležba izvedenih popravkov

- Vsaka sprememba se evidentira v kontrolni tabeli čiščenja:
 - entiteta in ID zapisa,
 - vrsta popravka (čiščenje / standardizacija),
 - prejšnja in nova vrednost (če relevantno),
 - čas izvedbe.

5. Pravila čiščenja

- Čiščenje temelji na vnaprej definiranih pravilih, ki jih je mogoče:
 - **konfigurirati**,
 - **vzdrževati** neodvisno od kode,
 - različna pravila za različne entitete.

6. Napake in opozorila

- Če čiščenje odkrije zapise, ki jih ni mogoče samodejno popraviti:
 - se ti zapisi posebej zabeležijo,
 - pripravi se poročilo za ročno pregledovanje.

Namen funkcionalnosti:

- Zagotavlja **kakovost podatkov** pred nadaljnjo transformacijo,
- Omogoča **standardizirano analitiko**,
- Povečuje zanesljivost rezultatov v podatkovnem skladišču.

2.3.4.4.2 Predpogoji

- Extract shema mora vsebovati zadnji uspešen zajem podatkov,
- Definirana morajo biti pravila čiščenja in standardizacije,
- Kontrolna tabela za beleženje čiščenja mora biti vzpostavljena.

2.3.4.4.3 Merila uspešnosti

- 100 % obdelava vseh zapisov v Extract shemi,
- Popravljenе napake ustrezajo definiranimu standardu,
- Nepopravljive napake so ločeno zabeležene za ročno obravnavo,

- Vsako čiščenje je revizijsko sledljivo.

2.3.4.4.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~7300	Čiščenje in standardizacija podatkov	Priprava podatkov z odstranitvijo napak, standardizacijo formatov in konsolidacijo vrednosti za nadaljnjo obdelavo v DW.	Extract shema in pravila čiščenja pripravljena	Vsi podatki obdelani, napake ločeno evidentirane, vsaka sprememba revizijsko zabeležena.

2.3.4.4.5 Diagram primerov uporabe

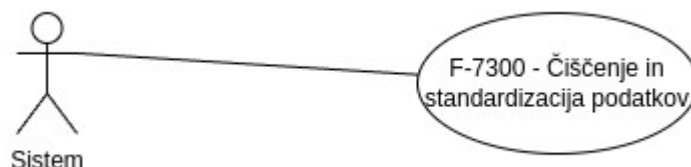


Diagram 36: Diagram primerov uporabe za SKL-DW-F-7300.

2.3.4.5 SKL-DW-F-7400: Generiranje nadomestnih ključev (Surrogate Keys)

2.3.4.5.1 Opis funkcionalnosti

Funkcionalnost omogoča **generiranje nadomestnih (surrogate) ključev** za vse glavne entitete v podatkovnem skladišču POKK, ki se bodo uporabljale v dimenzijskem modeliranju.

Nadomestni ključi nadomeščajo naravne ključe (npr. MID, DŠ), ki:

- **niso stabilni** skozi čas (npr. spremembe pri subjektih, osebah),
- **niso optimalni** za relacijsko povezovanje v podatkovnem skladišču,
- **zmanjšujejo izpostavljenost občutljivih podatkov** v analitičnih procesih.

Glavne značilnosti:

1. Ustvarjanje nadomestnih ID-jev

- Za vsako entiteto (npr. subjekt, oseba, žival, GERK, naslov, prostorska enota) se ustvari:
 - **enoličen nadomestni ID** (npr. INTEGER, BIGINT, GUID),
 - ki se uporablja kot primarni ključ v podatkovnem skladišču (DW).

2. Preslikava naravnih ključev

- Sistem vzdržuje **mapping tabele**:
 - naravni ključ → surrogate ključ,
 - omogoča rekonstrukcijo povezav, če je potrebno.

3. Obvladovanje sprememb

- Če se naravni ključ spremeni (npr. sprememba DŠ), se:
 - ohrani isti surrogate ključ (če entiteta ostaja ista),

- kreira nova verzija (če gre za spremembo entitete – podprto z SCD v F-7600).

4. Format in generacija ključev

- Surrogate ključi so:
 - **sekvenčno generirani** (ID generatorji na ravni entitete),
 - **enolični znotraj sistema**,
 - ne vsebujejo semantične informacije (ni informacij o vrsti entitete v ključu).

5. Beležba generacije

- Vsaka ustvarjena preslikava se zabeleži z:
 - naravnim ključem,
 - surrogate ključem,
 - datumom kreacije,
 - entiteto, kjer se uporablja.

6. Revizijska sled

- Vsi posegi v mappinge so sledljivi,
- sistem beleži čas ustvarjanja in morebitne spremembe.

Namen funkcionalnosti:

- Zagotavlja **stabilne relacijske povezave** v podatkovnem skladišču,
- Povečuje **varnost** podatkov (odmik od uporabe občutljivih naravnih ključev),
- Omogoča **dimenzijsko modeliranje** skladno z BI standardi.

2.3.4.5.2 Predpogoji

- Extract shema mora vsebovati konsistentne naravne ključe,
- Vzpostavljen mora biti mehanizem za generacijo unikatnih ID-jev na entiteto.

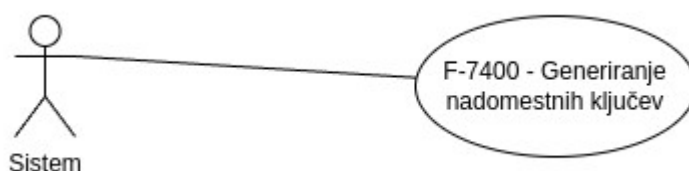
2.3.4.5.3 Merila uspešnosti

- Vsak zapis v Extract shemi ima enoličen surrogate ključ,
- Mapping naravni ključ → surrogate ključ je popoln in brez konfliktov,
- V primeru napak so spremembe zavrnjene (transakcijsko).

2.3.4.5.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~7400	Generiranje nadomestnih ključev	Ustvarjanje surrogate ključev za vse glavne entitete v podatkovnem skladišču POKK, za potrebe stabilnih povezav in varnosti.	Extract shema, ID generatorji na entitete	Vsaka entiteta ima stabilen, enoličen surrogate ključ, mapping je revizijsko sledljiv.

2.3.4.5.5 Diagram primerov uporabe



2.3.4.6 SKL-DW-F-7500: Dimenzijsko modeliranje podatkov

2.3.4.6.1 Opis funkcionalnosti

Funkcionalnost omogoča **strukturiranje podatkov v dimenzijsko obliko** za potrebe analitičnega obdelovanja v podatkovnem skladišču POKK.

Dimenzijsko modeliranje omogoča:

- **učinkovito agregacijo, filtriranje in analizo** podatkov,
- oblikovanje **analitično prijaznih struktur** (star schema, snowflake schema),
- **povezovanje dejstev (facts)** preko **stabilnih dimenzij (surrogate keys)**.

Modeliranje temelji na vzpostavljenih **surrogate ključih** (glej F-7400) in standardnih principih BI sistemov.

Glavne značilnosti:

1. Vzpostavitev minimalnih dimenzij

Dimenzija	Opis
Živali	Vrsta, pasma, identifikatorji, status
Lokacije (Kmetije)	MID, naslov, regija, velikost GERK
Statistične enote	Upravne enote, občine, regije, NUTS2
GERK	Površina, vrsta rabe, status GERK
Pasme	Šifra pasme, naziv pasme, vrsta
Živalske vrste	Klasifikacija vrste (govedo, ovca, čebela, prašič, konj)

2. Dodatne dimenzije

Po dogovoru z naročnikom glede na potrebe analize, npr.:

- uporabniške vloge,
- soglasja,
- časovne dimenzije (letnik, sezona).

3. Struktura modela

- **Dimenzije (dimension tables):** stabilne tabele z minimalnimi spremembami (Slowly Changing Dimensions - SCD podprto).
- **Dejstva (fact tables):**
 - štetje živali po vrsti, pasmi, lokaciji,
 - število zastopništev po regiji,
 - premiki živali po regijah,
 - populacije po GERK rabi.

4. Povezave

- Dimenzije so povezane s fact tabelami prek surrogate ključev,

- Uporabljajo se standardne relacije (one-to-many, many-to-one).

5. Aktualnost in veljavnost

- Vsaka dimenzija vključuje:
 - atribut "velja_od" in "velja_do" (časovna veljavnost),
 - status ("aktivno", "neaktivno") za podporo počasno spreminjajočih se dimenzij.

6. Dograditev modela

- Dodatne dimenzije ali prilagoditve modela se izvajajo v skladu s potrebami naročnika,
- Vse spremembe se dokumentirajo in verzionirajo.

Namen funkcionalnosti:

- Omogoča učinkovito izvajanje **poizvedb, analiz, poročil in vizualizacij**,
- Povečuje **učinkovitost poizvedb** preko optimiziranih struktur,
- Pripravi sistem za **BI analitiko** in **raziskovalne potrebe**.

2.3.4.6.2 Predpogoji

- Surrogate ključi morajo biti generirani (glej F-7400),
- Extract shema mora biti čista in standardizirana (glej F-7300),
- Dogovorjen minimalni nabor dimenzij potrjen z naročnikom.

2.3.4.6.3 Merila uspešnosti

- Vsaka glavna entiteta je vključena v vsaj eno dimenzijo,
- Fact tabele so pravilno povezane s surrogate ključi do dimenzij,
- Model omogoča standardne agregacije (COUNT, SUM, AVG),
- Časovna veljavnost dimenzij je zagotovljena.

2.3.4.6.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~7500	Dimenzijsko modeliranje podatkov	Oblikovanje minimalnega nabora dimenzij in fact tabel za analitično obdelavo podatkov v podatkovnem skladišču POKK.	Surrogate ključi, čist Extract, potrjen nabor dimenzij	Dimenzije pokrivajo vse ključne entitete, povezave pravilno vzpostavljene, podpora agregacijam omogočena.

2.3.4.6.5 Diagram primerov uporabe

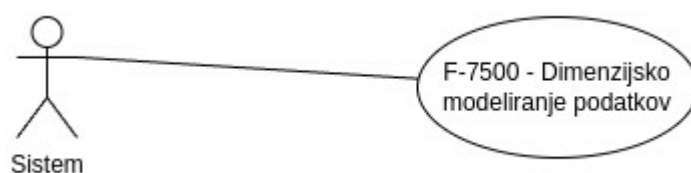


Diagram 38: Diagram primerov uporabe za SKL-DW-F-7500.

2.3.4.7 SKL-DW-F-7600: Upravljanje zgodovinskih podatkov (Slowly Changing Dimensions – SCD)

2.3.4.7.1 Opis funkcionalnosti

Funkcionalnost omogoča **upravljanje sprememb v dimenzijskih podatkih skozi čas** po modelu **Slowly Changing Dimensions (SCD)**, kar omogoča **ohranjanje zgodovine sprememb** brez izgube preteklih stanj v podatkovnem skladišču POKK.

Z uvedbo SCD modeliranja POKK zagotavlja, da:

- so **vse spremembe v entitetah** (npr. sprememba naslova subjekta, nove vloge osebe, sprememba GERK statusa) ustrezno zabeležene,
- analitični sistemi lahko izvajajo **časovno odvisne analize** (npr. stanje na določen datum, primerjava preteklih in sedanjih stanj).

Glavne značilnosti:

1. Uporabljen model SCD Type 2

- Ob spremembi podatkov:
 - se **ne prepíše** obstoječ zapis,
 - **ustvari se nova verzija** zapisa z:
 - novim surrogate ključem,
 - oznako obdobja veljavnosti (`velja_od`, `velja_do`),
 - statusom ("aktivno" / "neaktivno").

2. Atributi za časovno sledljivost

- Vsak zapis v dimenziji vsebuje:
 - `velja_od` (datum začetka veljavnosti),
 - `velja_do` (datum konca veljavnosti ali NULL za trenutno veljavne zapise),
 - `status` (aktivno/neaktivno).

3. Mehanizem zaznavanja sprememb

- Sistem primerja relevantna polja dimenzije med novimi in obstoječimi podatki:
 - če se podatki razlikujejo → sproži se kreacija nove verzije,
 - če se ne razlikujejo → nič ne ukrene.

4. Ohranjanje zgodovine

- Prejšnji zapisi niso brisani niti prepisani,
- Omogočen vpogled v stanje na poljuben datum.

5. Revizijska sled sprememb

- Vsaka nova verzija zapisa je revizijsko zabeležena z:
 - atributom spremembe,

- razlogom spremembe (če podprt),
- datumom in časom nastanka nove verzije.

6. Povezave na fact tabele

- Fact tabele v DW so vedno povezane z **veljavno verzijo dimenzije** glede na čas dogodka,
- S tem je omogočena časovno konsistentna analitika.

Namen funkcionalnosti:

- Omogoča **časovno primerjalne analize** (stanje nekoč vs. danes),
- Ohranja **popolno zgodovino poslovnih entitet** v podatkovnem skladišču,
- Podpira **sproten in zanesljiv nadzor sprememb** v sistemu POKK.

2.3.4.7.2 Predpogoji

- Vzpostavljene stabilne dimenzijske tabele z surrogate ključi,
- Omogočeno revizijsko sledenje sprememb v ETL procesih.

2.3.4.7.3 Merila uspešnosti

- Vsaka sprememba v relevantnem polju ustvari novo verzijo zapisa,
- Aktivni zapis vedno vsebuje `velja_do IS NULL`,
- Vse spremembe so revizijsko zabeležene,
- Fact tabele uporabljajo pravilno verzijo dimenzije glede na čas dogodka.

2.3.4.7.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~7600	Upravljanje zgodovinskih podatkov (SCD)	Upravljanje sprememb v dimenzijah po modelu Slowly Changing Dimensions (Type 2), z ohranitvijo popolne časovne zgodovine.	Stabilne dimenzije, surrogate ključi, revizijsko sledenje	Nova verzija ob vsaki spremembi, pravilna časovna sledljivost, povezave fact tabel časovno konsistentne.

2.3.4.7.5 Diagram primerov uporabe

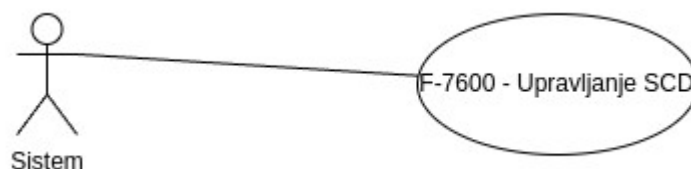


Diagram 39: Diagram primerov uporabe za SKL-DW-F-7600.

2.3.4.8 SKL-DW-F-7700: Validacija in integriteta transformiranih podatkov

2.3.4.8.1 Opis funkcionalnosti

Funkcionalnost zagotavlja **preverjanje kakovosti, popolnosti in referenčne integritete** podatkov, ki so bili transformirani in preneseni v podatkovno skladišče (DW) sistema POKK.

Glavni cilj je zagotoviti, da so vsi podatki, vključeni v dimenzijske in dejstvene strukture:

- popolni,
- konsistentni,
- pravilno povezani med sabo.

Validacija se izvaja **samodejno** po vsaki transformaciji in pred končno osvežitvijo podatkovnega skladišča.

2.3.4.8.2 Glavne značilnosti:

1. Preverjanje popolnosti podatkov

- Vsaj polja, ki so označena kot **obvezna** (npr. surrogate ključ, naziv, klasifikacija), morajo biti prisotna in izpolnjena.
- Preverjanje polj:
 - NULL vrednosti,
 - neveljavnih vrednosti (npr. negativna površina GERK, nedovoljena kategorija živali).

2. Preverjanje referenčne integritete

- Vsak zapis v **fact tabeli** mora imeti:
 - veljavno povezavo na ustrezne **dimenzije** (npr. subjekt, žival, GERK, pasma, statistična enota),
- Preverjanje ustreznosti tujih ključev (FK → PK):
 - v primeru manjkajočih povezav prenos zavrne.

3. Konsistentnost podatkov

Podatki morajo biti:

- v pravih formatih (npr. datumi v YYYY-MM-DD, števila v decimalnem formatu),
- usklajeni s šifranti (npr. vrste rabe, vrste živali).

4. Beležba rezultatov validacije

Vsa preverjanja so dokumentirana v **tabeli validacijskih rezultatov**, ki vključuje:

- datum in čas validacije,
- obseg preverjenih zapisov,
- število napak po entitetah,
- podrobnosti napak.

5. Upravljanje napak

- Kritične napake (manjkajoči ključ, napačne povezave) preprečijo končno osvežitev skladišča.

- Manj kritične napake (npr. opozorila o formatih) se zabeležijo in poročajo, a ne blokirajo prenosa.

6. Revizijska sled

- Vsaka validacija se revizijsko zabeleži,
- Omogočen vpogled v zgodovino validacij in rezultate preteklih preverjanj.

Namen funkcionalnosti:

- Zagotoviti **visoko kakovost in konsistenco podatkov** v DW,
- Preprečiti **analitične napake** zaradi napačnih ali manjkajočih povezav,
- Omogočiti **sproten nadzor nad kakovostjo** podatkovnega skladišča.

2.3.4.8.3 Predpogoji

- Dimenzije in fact tabele morajo biti transformirane in pripravljene,
- Pravila popolnosti in integritete morajo biti definirana,
- Extract shema mora biti dostopna za validacijo.

2.3.4.8.4 Merila uspešnosti

- 100 % preverba vseh relevantnih tujih ključev in obveznih polj,
- Vsaka validacija revizijsko zabeležena,
- V primeru kritičnih napak se postopek osvežitve ustavi,
- Napake so kategorizirane (kritične / opozorilne) in poročane.

2.3.4.8.5 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~7700	Validacija in integriteta transformiranih podatkov	Preverjanje popolnosti, konsistence in referenčne integritete vseh transformiranih podatkov pred osvežitvijo podatkovnega skladišča.	Transformirane dimenzije in fact tabele pripravljene	Kritične napake preprečijo prenos, validacije popolno zabeležene in dostopne v zgodovini.

2.3.4.8.6 Diagram primerov uporabe

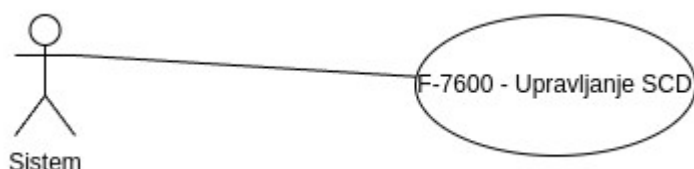


Diagram 40: Diagram primerov uporabe za SKL-DW-F-7700.

2.3.4.9 SKL-DW-F-7800: Nalaganje podatkov v DW shemo (Load)

2.3.4.9.1 Opis funkcionalnosti

Funkcionalnost omogoča **nadzorovan in transakcijsko konsistenten prenos** validiranih, transformiranih podatkov iz **Extract/Transform sloja v končno produkcijsko podatkovno shemo** podatkovnega skladišča POKK.

Faza Load je zadnja faza v ETL procesu in zagotavlja, da so vsi podatki pripravljeni za:

- analitične poizvedbe,
- poročanje,
- BI orodja in podporne sisteme.

Glavne značilnosti:

1. Vrste nalaganja

- **Polni prenos (Full Load):**
 - uporablja se za polne zajeme (snapshot),
 - predhodno čiščenje produkcijske DW sheme (dimenzije in facts),
 - popolno nadomestitev obstoječih podatkov.
- **Incremental Load:**
 - uporablja se za delta zajeme,
 - dodajanje novih zapisov,
 - posodabljanje obstoječih zapisov glede na surrogate ključe.

2. Nalaganje dimenzijskih tabel

- Vstavljanje ali posodabljanje dimenzij:
 - upoštevanje `velja_od`, `velja_do`,
 - podpora za Slowly Changing Dimensions (SCD) model,
 - ustrezno zapiranje in odpiranje verzij dimenzijskih zapisov.

3. Nalaganje dejstev (fact tables)

- Povezovanje facts z ustreznimi dimenzijami prek surrogate ključev,
- Polnjenje fact tabel s časovno konsistentnimi podatki (upoštevanje datuma dogodka vs. veljavnosti dimenzije).

4. Integriteta in validacija nalaganja

- Pred končno commit operacijo se:
 - preveri veljavnost povezav (foreign key constraints),
 - preveri usklajenost dimenzij in dejstev.

5. Evidenca prenosa

- Za vsak prenos se zabeleži:
 - vrsta prenosa (Full/Incremental),
 - število prenesenih zapisov po tabelah,
 - status (OK/FAILED),
 - čas začetka in konca operacije.

6. Upravljanje napak

- V primeru kritične napake:
 - se transakcija zavrne (rollback),
 - stanje DW sheme ostane nespremenjeno,
 - napaka se zabeleži za nadaljnjo analizo.

Namen funkcionalnosti:

- Omogočiti **zaključen, konsistenten in nadzorovan prenos** podatkov v produkcijsko podatkovno skladišče,
- Podpreti **analitične procese** na točnih in pravočasnih podatkih,
- Preprečiti **vnos nekonsistentnih ali delnih podatkov** v DW.

2.3.4.9.2 Predpogoji

- Validirani in transformirani podatki v Extract shemi,
- Definiran režim nalaganja (Full/Incremental),
- Omogočeno transakcijsko izvajanje operacij.

2.3.4.9.3 Merila uspešnosti

- 100 % konsistenten prenos podatkov glede na validacijske pogoje,
- Vsaka sprememba v DW shemi revizijsko zabeležena,
- Napake so kategorizirane in jasno poročane,
- Pri polnem prenosu: ni ostankov starih podatkov,
- Pri inkrementalnem prenosu: pravilno dodajanje ali posodabljanje.

2.3.4.9.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~7800	Nalaganje podatkov v DW shemo (Load)	Nadzorovan prenos validiranih podatkov iz Extract/Transform sloja v produkcijsko podatkovno skladišče POKK.	Validirani in transformirani podatki pripravljeni	Pravilno prenesene dimenzije in dejstva, napake preprečijo delne prenose, vse operacije revizijsko zabeležene.

2.3.4.9.5 Diagram primerov uporabe

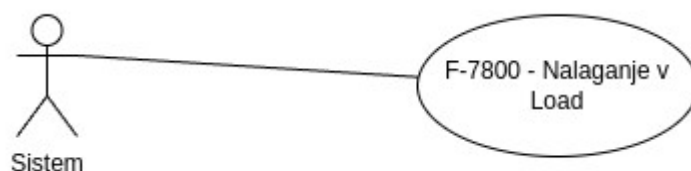


Diagram 41: Diagram primerov uporabe za SKL-DW-F-7800.

2.3.5 SKL-Ziv: Sklop žival

2.3.5.1 SKL-ZIV-F-10000: Pregledovanje živali

2.3.5.1.1 Opis funkcionalnosti

Funkcionalnost omogoča pregledovanje in iskanje živali v sistemu POKK preko spletnega uporabniškega vmesnika in API vmesnika.

Uporabnik nastavi filtre (preproste ali napredne) in pridobi seznam živali, ki ustrezajo zahtevanim kriterijem.

Seznam je strukturiran, ostranjen in omogoča nadaljnjo uporabo v drugih sklopih (premiki, dogodki, pasme, dokumenti itd.).

Podatki so dostopni tako preko grafičnega uporabniškega vmesnika kot preko standardiziranega API.

Glavne značilnosti:

- Podpora spletnemu uporabniškemu vmesniku (GUI) in API (REST).
- Možnost iskanja in filtriranja po ključnih atributih (identifikatorji, spol, status, species, pasma, datum rojstva).
- Stranjenje rezultatov (paginacija) in sortiranje po poljubnih atributih.
- Ostranjen seznam: prikaz samo bistvenih podatkov (ID živali, status, ključni identifikatorji).
- Standardiziran izpis v strukturirani obliki (JSON za API, dinamična tabela za GUI).
- Zagotovljena skladnost s performančnimi zahtevami tudi pri velikem številu živali.

2.3.5.1.2 Predpogoji

- Uporabnik je avtentificiran in avtoriziran za dostop do modula Žival.
- Veljavna izbira zastopništva (kadar sistem deluje v zastopniškem načinu).
- Podatkovne sheme in komunikacijski modul so operativni.
- Sistem za revizijsko beleženje dogodkov je aktiven.

2.3.5.1.3 Merila uspešnosti

- Rezultati za do 100 živali se naložijo v ≤ 1.5 s.
- Pravilnost prikaza rezultatov glede na filtre $\geq 99,9$ %.
- Stranjenje in sortiranje podpirata vsaj 1.000.000 živali brez degradacije hitrosti.
- Vsak klic API ali iskanje v GUI se revizijsko beleži z vsemi parametri.

2.3.5.1.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~10001	Nastavitev filtrov	Uporabnik nastavi enostavne ali napredne filtre za iskanje živali.	Avtentikacija in avtorizacija uspešna.	Filtri pravilno omejijo rezultate v ≤ 1.5 s.

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~10002	Prikaz seznama živali	Sistem prikaže strukturiran seznam živali glede na nastavljene filtre.	Veljavni filtri, dostop do baze.	Rezultat vsebuje samo predpisane attribute.
~10003	Ostranjevanje rezultatov	Podpora za prikaz strani rezultatov (npr. 20, 50, 100 na stran).	Parametri strani so nastavljeni.	Brezhibno delovanje do 1M zapisov.
~10004	Sortiranje rezultatov	Uporabnik lahko sortira rezultate po izbranem atributu.	Podpora za sortiranje v vmesniku.	Razvrščanje izvedeno v ≤ 500 ms.
~10005	API dostop do seznama	Omogočen API za iskanje in pridobivanje seznama živali.	Aktivacija API ključa in avtorizacija uspešna.	JSON struktura natančno definirana in skladna.

2.3.5.1.5 Diagram primerov uporabe

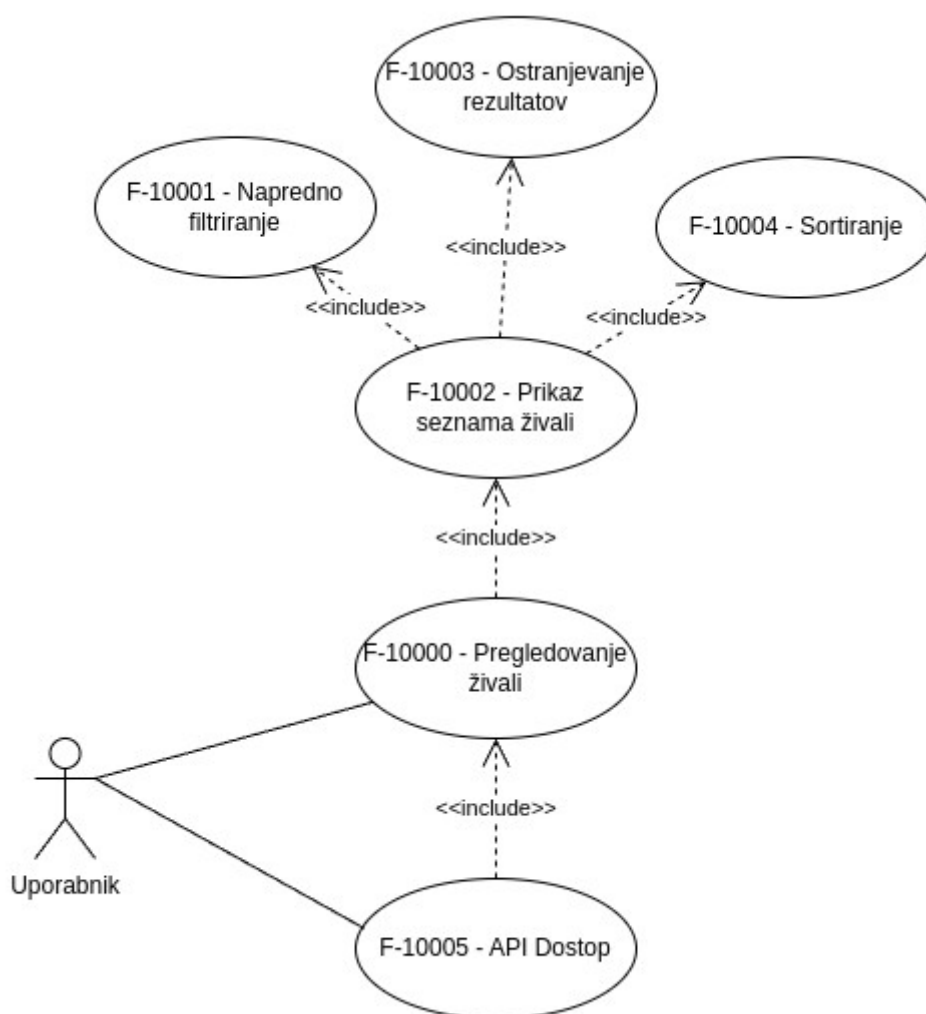


Diagram 42: Diagram primerov uporabe za SKL-ZIV-F-10000.

2.3.5.2 SKL-ZIV-F-10100: Pregled podatkov o živali

2.3.5.2.1 Opis funkcionalnosti

Funkcionalnost omogoča podroben pregled vseh podatkov o posamezni živali v sistemu POKK. Uporabnik dostopa do razširjene kartice živali, kjer so prikazani vsi ključni atributi, status, povezani identifikatorji ter osnovne povezave na druge evidence.

Spletni vmesnik in API podpirata enotno strukturirano vračilo podatkov.

Glavne značilnosti:

- Podpora spletnemu uporabniškemu vmesniku (GUI) in API (REST).
- Prikaz vseh osnovnih atributov živali (spol, datum rojstva, species, pasma, status).
- Prikaz vseh identifikatorjev živali (ušesna številka, alternativni ID-ji, imena).
- Povezave na sorodne entitete: rejec, lokacija, premiki, dogodki, dokumenti.
- Vidnost statusnih informacij (potrjeno z uradno evidenco, status življenjskega cikla).
- Prikaz obstoječih verzij osnovnih podatkov z veljavnostjo.
- Standardizirana oblika prikaza (JSON za API, kartica v GUI).

2.3.5.2.2 Predpogoji

- Uporabnik je avtentificiran in avtoriziran za dostop do modula Žival.
- Izbrano je aktivno zastopništvo (če je potrebno).
- Sistem za revizijsko beleženje dogodkov je aktiven.

2.3.5.2.3 Merila uspešnosti

- Podatki o živali se naložijo v ≤ 1.5 s po zahtevi.
- Vsi osnovni podatki in identifikatorji so vidni v enem klicu.
- Struktura podatkov ustreza vnaprej definiranemu standardu.
- Povezave na sorodne entitete (premiki, dogodki, pasme) so funkcionalne in konsistentne.
- Vsak prikaz podatkov se revizijsko beleži.

2.3.5.2.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~10101	Pridobitev osnovnih atributov živali	Prikaz osnovnih atributov (spol, rojstvo, species, status) v kartici ali JSON odgovoru.	Avtentikacija in avtorizacija uspešna.	Vse osnovne informacije so prikazane v ≤ 1.5 s.
~10102	Prikaz vseh identifikatorjev živali	Prikaz vseh vezanih identifikatorjev (ušesna številka, alternativni ID-ji, imena).	Ustrezna avtorizacija uporabnika.	Vsi identifikatorji so dostopni brez napak.
~10103	Povezava na sorodne entitete	Omogočen dostop do sorodnih entitet (rejec, lokacija, dogodki, premiki, dokumenti).	Aktivne povezave v sistemu.	Sorodne entitete se pravilno naložijo in prikažejo.
~10104	Prikaz statusnih informacij	Prikaz statusa življenjskega cikla živali in povezave z uradnimi evidencami.	Sinhronizacijski podatki so na voljo.	Statusi so prikazani pravilno v vsakem pogledu.

2.3.5.2.5 Diagram primerov uporabe

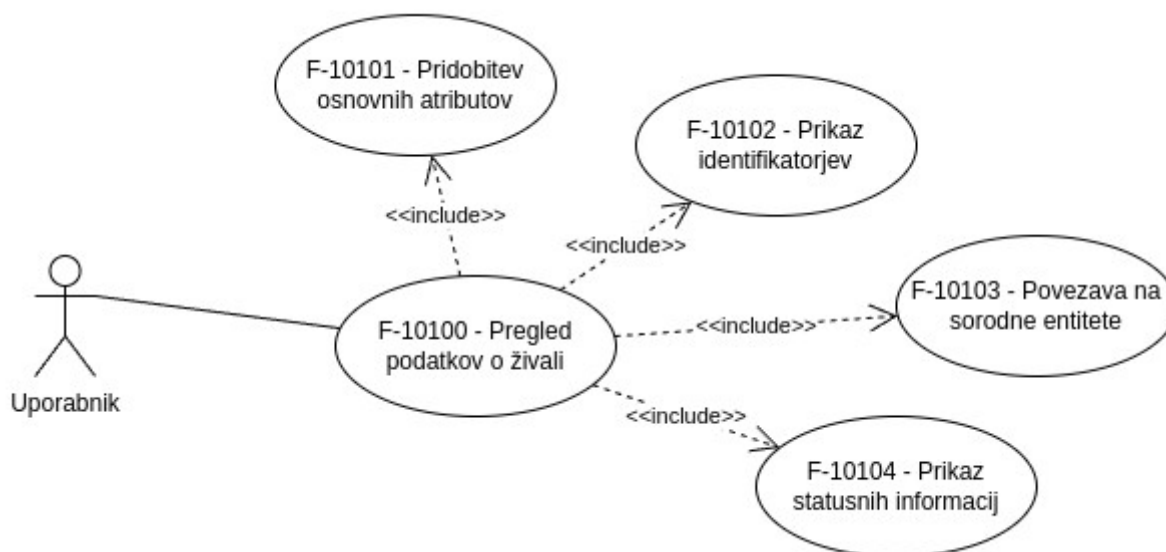


Diagram 43: Diagram primerov uporabe za SKL-ZIV-F-10100.

2.3.5.3 SKL-ZIV-F-10200: Pregled zgodovine podatkov o živali

2.3.5.3.1 Opis funkcionalnosti

Funkcionalnost omogoča uporabniku pregled zgodovine sprememb osnovnih podatkov živali v sistemu POKK.

Za vsako žival se vodi verzijska evidenca, kjer ima vsaka verzija določeno obdobje veljavnosti (velja od - velja do).

Uporabnik lahko izbere željeno verzijo in pridobi prikaz stanja podatkov živali na določen dan. Funkcionalnost je na voljo preko spletnega uporabniškega vmesnika in API.

Glavne značilnosti:

- Podpora spletnemu uporabniškemu vmesniku (GUI) in API (REST).
- Možnost izbire poljubne verzije osnovnih podatkov živali.
- Prikaz veljavnosti verzije (datum začetka in konca veljavnosti).
- Dostop do vseh atributov, kot so bili v času izbrane verzije.
- Standardiziran prikaz verzij v tabeli s filtri in možnostjo izbire datuma.
- Zagotovljena revizijska sled za dostop do zgodovinskih podatkov.

2.3.5.3.2 Predpogoji

- Uporabnik je avtentificiran in avtoriziran za dostop do modula Žival.
- Verzijsko sledenje sprememb osnovnih podatkov mora biti aktivno in konsistentno.
- Podatkovna shema mora podpirati hrambo podatkov po časovni veljavnosti.

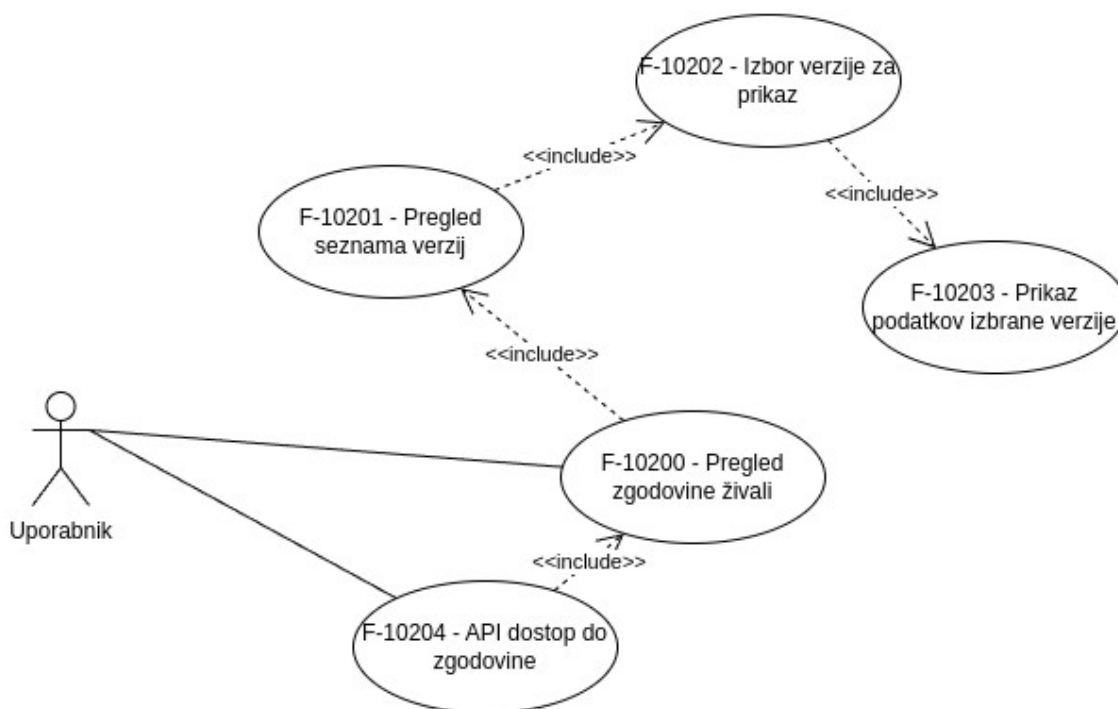
2.3.5.3.3 Merila uspešnosti

- Nalaganje seznama verzij za eno žival ≤ 1 s.
- Prikaz izbranega stanja živali v ≤ 1.5 s.
- Zagotovljena pravilna veljavnost datuma pri vsaki verziji.
- Revizijsko zabeležen dostop do verzijskih podatkov.

2.3.5.3.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~10201	Prikaz seznama verzij živali	Prikaz vseh verzij podatkov za izbrano žival, s prikazom obdobja veljavnosti.	Verzijsko sledenje aktivno.	Seznam verzij naložen v ≤ 1 s.
~10202	Izbor verzije za prikaz	Uporabnik izbere želeno verzijo (ali datum), sistem prikaže stanje podatkov na ta dan.	Veljavne verzije so na voljo.	Podatki ustrezajo stanju v izbranem obdobju.
~10203	Prikaz podatkov za izbrano verzijo	Sistem prikaže vse osnovne podatke živali, kot so veljali v času izbrane verzije.	Verzijska shema konsistentna.	Podatki pravilno predstavljeni v ≤ 1.5 s.
~10204	API dostop do verzij	API omogoča programskemu odjemalcu pridobitev verzij in stanj na podan dan.	Avtentikacija in avtorizacija uspešna.	API vrne standardizirano JSON strukturo.

2.3.5.3.5 Diagram primerov uporabe



Diag

ram 44: Diagram primerov uporabe za SKL-ZIV-F-10200.

2.3.5.4 SKL-ZIV-F-10300: Urejanje podatkov živali

2.3.5.4.1 Opis funkcionalnosti

Funkcionalnost omogoča urejanje osnovnih podatkov o živali v sistemu POKK.

Urejanje je omejeno izključno na tiste podatke, ki niso potrjeni s strani uradnih državnih evidenc. Vsaka sprememba osnovnih podatkov povzroči ustvarjanje nove verzije podatkovnega zapisa z novo veljavnostjo (časovno obdobje).

Funkcionalnost je dostopna prek spletnega uporabniškega vmesnika in API.

Glavne značilnosti:

- Podpora spletnemu uporabniškemu vmesniku (GUI) in API (REST).

- Omogočeno urejanje samo atributov, ki niso potrjeni iz državnih virov.
- Avtomatska kreacija nove verzije osnovnih podatkov ob vsaki spremembi.
- Validacija vnosov pred shranjevanjem (formati, obvezna polja, pravila).
- Sprožitev revizijskega zapisa o spremembi podatkov.
- Omejitev urejanja na uporabnike z ustreznimi pravicami.

2.3.5.4.2 Predpogoji

- Uporabnik je avtenticiran in ima avtorizacijo za urejanje podatkov živali.
- Sistem zna ločiti med atributi, ki so potrjeni iz uradnih evidenc in tistimi, ki niso.
- Verzijsko beleženje sprememb je aktivno in operativno.

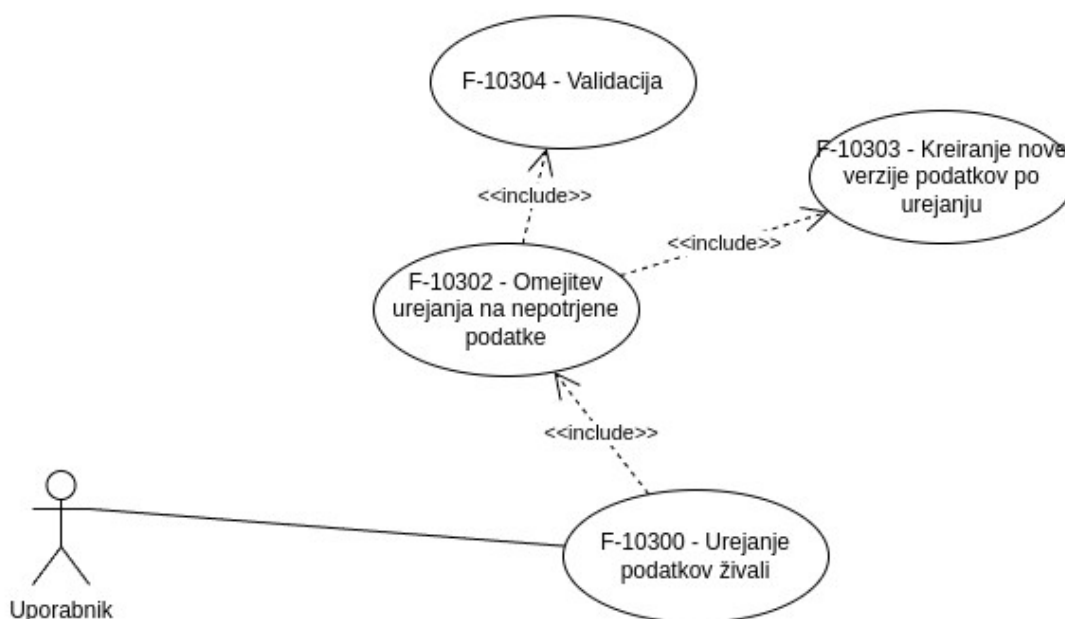
2.3.5.4.3 Merila uspešnosti

- Uspešna validacija in shranjevanje spremembe v ≤ 1.5 s.
- Nova verzija podatkov se kreira samodejno pri vsaki spremembi.
- Prepovedano urejanje atributov, potrjenih s strani uradnih evidenc.
- Revizijsko beleženje vsake izvedene spremembe.

2.3.5.4.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~10300	Urejanje atributov živali	Uporabnik ureja dovoljene (nepotrjene) podatke o živali.	Avtentikacija in avtorizacija uspešna.	Spremembe uspešno validirane in shranjene.
~10302	Omejitev na nepotrjene podatke	Sistem dovoljuje urejanje le podatkov, ki niso potrjeni iz uradnih evidenc.	Atributi ločeni na potrjene/nepotrjene.	Poskus spremembe potrjenih podatkov se zavrne.
~10303	Kreacija nove verzije podatkov	Ob vsaki spremembi sistema ustvari novo verzijo osnovnih podatkov živali.	Verzijsko sledenje aktivno.	Nova verzija ustvarjena ob vsaki spremembi.
~10304	Validacija vnosov	Sistem validira vse vnose pred shranjevanjem sprememb.	Pravila validacije aktivna.	Neveljavni vnosi so zavrnjeni s sporočilom.

2.3.5.4.5 Diagram primerov uporabe



Dia

2.3.5.5 SKL-ZIV-F-10400: Statusno vodenje živali

2.3.5.5.1 Opis funkcionalnosti

Funkcionalnost omogoča vodenje statusov živali v sistemu POKK.

Status živali odraža njen življenjski cikel in se posodablja glede na spremembe, zabeležene preko dogodkov (npr. potrdilo o zakolu spremeni status v "Mrtva", potrdilo o kotitvi spremeni status v "Živa" in nastavi datum rojstva).

Statusi določajo pravice za nadaljnje operacije nad živaljo in so osnova za konsistenco podatkov v sistemu.

Glavne značilnosti:

- Vodenje statusov: živa, mrtva, izločena, neaktivna.
- Avtomatsko posodabljanje statusa glede na prejete dogodke v modulu Dogodki.
- Ročna sprememba statusa le za uporabnike z ustreznimi pooblastili (npr. administratorji).
- Evidentiranje statusnih sprememb z datumom spremembe in virom spremembe (ročno, avtomatsko).
- Onemogočanje določenih operacij na podlagi statusa (npr. na mrtvi živali ni dovoljenih več premikov).
- Revizijsko beleženje vseh statusnih sprememb.

2.3.5.5.2 Predpogoji

- Uporabnik mora biti avtenticiran in avtoriziran za pregled ali urejanje statusov.
- Modul Dogodki mora biti aktiven in pravilno evidentira dogodke, ki vplivajo na status.
- Verzijsko in revizijsko beleženje sprememb mora biti omogočeno.

2.3.5.5.3 Merila uspešnosti

- Status se posodobi avtomatsko ob vnosu ustreznega dogodka v ≤ 2 s.
- Statusne spremembe so vidne takoj po potrditvi spremembe.
- Ročna sprememba statusa je dovoljena le upravičenim uporabnikom.
- Vsi statusi in spremembe so revizijsko zabeleženi z uporabnikom, virom in časom.

2.3.5.5.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~10401	Avtomatsko posodabljanje statusa	Sistem avtomatsko spremeni status na podlagi dogodkov (npr. zakol, rojstvo).	Modul Dogodki aktiven.	Sprememba izvedena v ≤ 2 s po zapisu dogodka.
~10402	Ročna sprememba statusa	Uporabnik z ustreznimi pravicami lahko ročno spremeni status.	Avtentikacija in avtorizacija uspešna.	Sprememba dovoljena samo z ustreznim pooblastilom.
~10403	Beleženje statusnih sprememb	Sistem evidentira vsako spremembo statusa z virom, datumom in uporabnikom.	Revizijsko beleženje aktivno.	Revizijski zapis dostopen v ≤ 2 s.
~10404	Omejitve operacij glede na status	Glede na status živali so omejene dovoljene operacije (npr. prepoved premikov za mrtve živali).	Veljavni status določene živali.	Neveljavne operacije se zavrnejo z jasnim obvestilom.

2.3.5.5.5 Diagram primerov uporabe

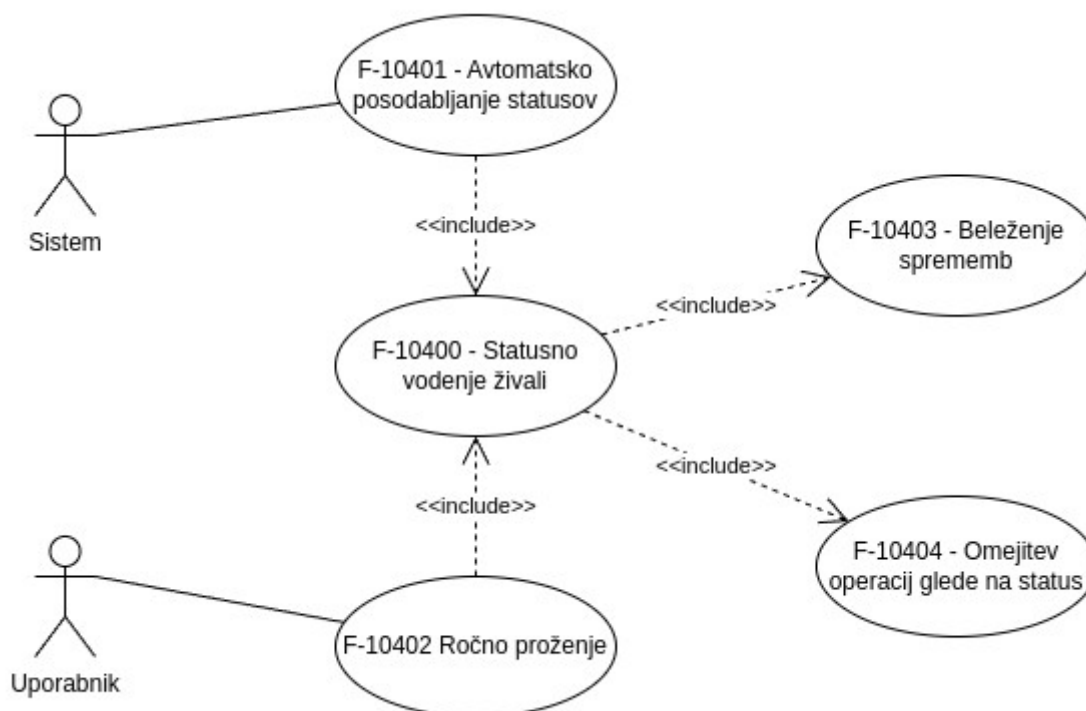


Diagram 46: Diagram primerov uporabe za SKL-ZIV-F-10400.

2.3.5.6 SKL-ZIV-F-10500: Upravljanje z identifikatorji živali

2.3.5.6.1 Opis funkcionalnosti

Funkcionalnost omogoča upravljanje z identifikatorji, vezanimi na živali v sistemu POKK. Uporabnik lahko pregleda, doda, preveri unikatnost in mehko izbriše identifikatorje živali. Identifikatorji vključujejo ušesne številke, alternativne ID-je, imena in druge identifikacijske oznake.

Za določene tipe identifikatorjev (npr. ušesna številka glede na species) velja obvezna unikatnost, ki jo sistem ob vnosu preveri.

Glavne značilnosti:

- Pregled vseh identifikatorjev, vezanih na izbrano žival.
- Filtriranje seznama identifikatorjev po tipu, vrednosti in veljavnosti.
- Dodajanje novih identifikatorjev s preverjanjem pravilnosti in unikatnosti (če je zahtevano).
- Mehko brisanje identifikatorjev (logično brisanje brez fizične izbriša iz baze).
- Povezava novega identifikatorja s časovno veljavnostjo (od–do).
- Revizijsko beleženje vseh operacij nad identifikatorji.

2.3.5.6.2 Predpogoji

- Uporabnik mora biti avtentificiran in avtoriziran za upravljanje identifikatorjev.
- Veljavna struktura shem identifikatorjev mora biti predhodno definirana.
- Pravila unikatnosti morajo biti pravilno konfigurirana za posamezne tipe identifikatorjev.

2.3.5.6.3 Merila uspešnosti

- Dodajanje novega identifikatorja mora biti izvedeno v ≤ 1.5 s.
- Preverjanje unikatnosti mora biti izvedeno pred potrditvijo vnosa.
- Brisanje (deaktivacija) mora označiti identifikator kot neveljavnega, brez fizičnega brisanja

zapisa.

- Vsaka operacija nad identifikatorji je revizijsko zabeležena.

2.3.5.6.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~10501	Pregled identifikatorjev	Prikaz vseh identifikatorjev živali s podporo filtriranju po tipu, vrednosti in statusu.	Avtentikacija in avtorizacija uspešna.	Prikaz rezultatov v ≤ 1 s.
~10502	Dodajanje novega identifikatorja	Uporabnik lahko doda nov identifikator živali.	Veljavna pravila identifikatorjev.	Dodajanje izvedeno v ≤ 1.5 s, validacija uspešna.
~10503	Preverjanje unikatnosti identifikatorja	Sistem preveri unikatnost identifikatorja glede na pravila (npr. species + ušesna številka).	Pravila unikatnosti definirana.	Preverjanje izvedeno pred vnosom.
~10504	Mehko brisanje identifikatorja	Uporabnik lahko mehko izbriše obstoječi identifikator (nastavi veljavnost konca).	Avtentikacija uspešna.	Identifikator označen kot neaktiven, brez fizičnega brisanja.

2.3.5.6.5 Diagram primerov uporabe

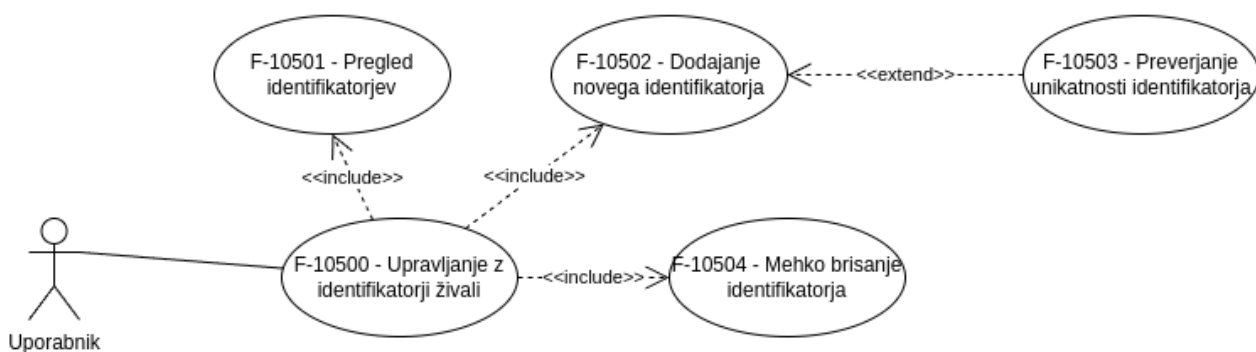


Diagram 47: Diagram primerov uporabe za SKL-ZIV-F-10500.

2.3.5.7 SKL-ZIV-F-10600: Združevanje podatkov dveh živali

2.3.5.7.1 Opis funkcionalnosti

Funkcionalnost omogoča združevanje dveh zapisov o živalih v sistemu POKK.

Združevanje je možno med dvema nepotrjenima živalima ali med eno nepotrjeno in eno potrjeno živaljo iz uradnih evidenc.

Cilj združevanja je konsolidacija podatkov v en veljaven zapis, ob hkratni deaktivaciji oziroma zaključku obstoječih starih zapisov.

Funkcionalnost podpira ročne in avtomatske postopke združevanja glede na vir potrjenosti.

Glavne značilnosti:

- Ročno združevanje dveh nepotrjenih živali v nov zapis.
- Ročno združevanje nepotrjene živali s potrjeno živaljo (podatki nepotrjene se prenesejo v potrjeno, kjer je to dovoljeno).
- Avtomatsko združevanje ob potrditvi živali iz uradnih evidenc.

- Deaktivacija (mehak zaključek) starih zapisov po uspešnem združevanju.
- Ohranitev verzijske zgodovine iz obeh združenih živali.
- Revizijsko beleženje vseh faz združevanja.

2.3.5.7.2 Predpogoji

- Obe živali morata biti dostopni in v aktivnem statusu pred združitvijo.
- Vsaj ena žival mora biti nepotrjena, če druga žival že obstaja kot potrjena.
- Verzijsko in revizijsko beleženje mora biti omogočeno.
- Pravila za prenos atributov morajo biti definirana (kateri podatki se lahko prenesejo).

2.3.5.7.3 Merila uspešnosti

- Postopek združitve mora biti izveden v ≤ 2 s po potrditvi uporabnika ali avtomatskem dogodku.
- Po združitvi mora biti jasno razvidna nova veljavna entiteta z ustrezno zgodovino.
- Noben potrjen uradni podatek se ne sme prepisati brez dovoljenja sistema.
- Vse spremembe morajo biti revizijsko evidentirane.

2.3.5.7.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~10601	Združevanje dveh nepotrjenih živali	Uporabnik ročno združi dve nepotrjeni živali v en nov zapis. Stari zapisi se deaktivirajo.	Dostopni dve nepotrjeni živali.	Nova žival ustvarjena, stari zapisi zaključeni, revizija izvedena.
~10602	Združevanje nepotrjene z potrjeno živaljo	Nepotrjeni žival priredimo potrjeni živali: prenesemo poljubne attribute (nepotrjene) in zaključimo staro žival.	Ena potrjena in ena nepotrjena žival dostopni.	Podatki pravilno preneseni, stari zapis deaktiviran.
~10603	Deaktivacija starih zapisov	Po uspešnem združevanju se izvorna zapisa označita kot neaktivna (veljavnost zaključena).	Aktivni stari zapisi pred združitvijo.	Veljavnost zaključena takoj po združitvi.

2.3.5.7.5 Diagram primerov uporabe

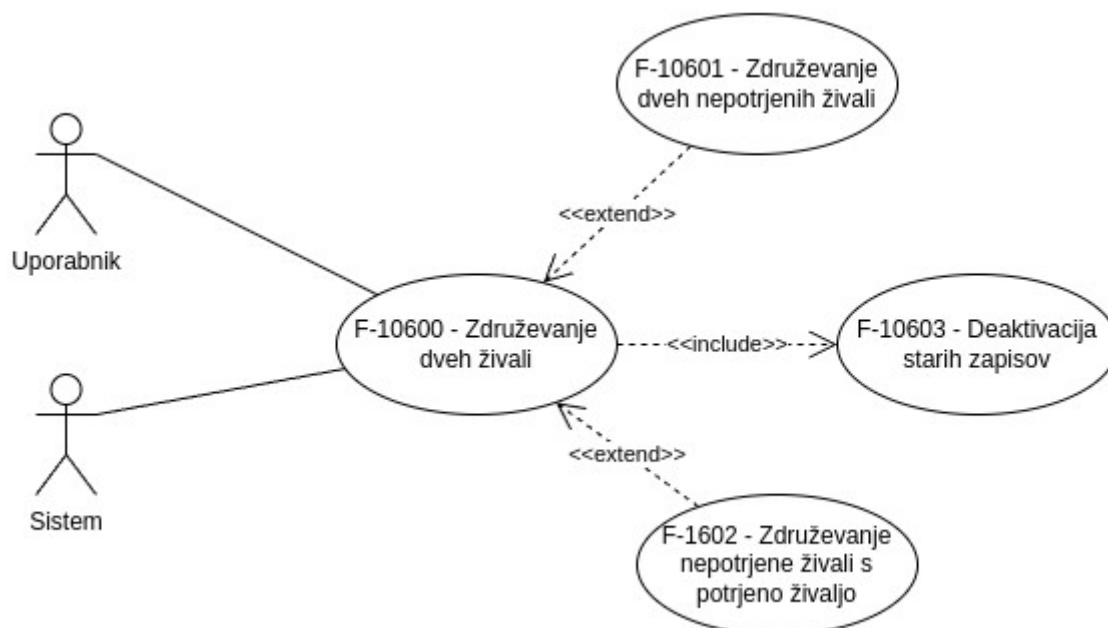


Diagram 48: Diagram primerov uporabe za SKL-ZIV-F-10600.

2.3.5.8 SKL-ZIV-F-10700: Povezovanje živali z uradno evidenco

2.3.5.8.1 Opis funkcionalnosti

Funkcionalnost omogoča povezovanje obstoječe živali v sistemu POKK z zapisom iz uradne evidence.

Ko pride do ujemanja identifikatorjev med lokalnim zapisom in podatki iz uradnih evidenc, se podatki iz uradne evidence priredijo obstoječi živali.

Sistem ob tem kreira novo verzijo osnovnih podatkov, kar omogoča popolno sledljivost sprememb. Funkcionalnost je lahko sprožena ročno ali avtomatsko preko sinhronizacijskih procesov.

Glavne značilnosti:

- Povezovanje obstoječe živali z zapisom iz uradne evidence na podlagi identifikatorjev.
- Pridobitev, validacija in prenos podatkov iz uradne evidence v obstoječo žival.
- Samodejna kreacija nove verzije osnovnih podatkov z označenim virom spremembe.
- Ohranjanje predhodnih podatkov v verzijski zgodovini.
- Revizijsko beleženje vseh operacij povezovanja in sprememb.

2.3.5.8.2 Predpogoji

- Nepotrjena žival obstaja v sistemu in ima aktivne identifikatorje.
- Ujemajoči podatki iz uradnih evidenc so na voljo prek komunikacijskega modula.
- Verzijsko sledenje sprememb in revizijsko beleženje sta aktivna.

2.3.5.8.3 Merila uspešnosti

- Povezovanje in kreacija nove verzije izvedena v ≤ 2 s po zaznavi ujemanja.
- Podatki iz uradne evidence so konsistentno prirejeni brez izgube predhodnih podatkov.
- Vse spremembe podatkov so verzijsko in revizijsko zabeležene.

2.3.5.8.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~10701	Povezava živali z zapisom iz uradne evidence	Na podlagi ujemanja identifikatorjev se obstoječa žival poveže z uradnim zapisom.	Aktivna lokalna žival in dostopni uradni podatki.	Ujemanje pravilno zaznano, povezava izvedena brez napak.
~10702	Prenos podatkov iz uradne evidence	Podatki iz uradne evidence se priredijo obstoječemu zapisu živali.	Validirani uradni podatki.	Podatki uspešno preneseni, skladni s pravili.
~10703	Kreacija nove verzije osnovnih podatkov	Ob povezavi se kreira nova verzija živalskega zapisa s prirejenimi podatki.	Verzijsko sledenje aktivno.	Nova verzija nastane takoj po prenosu podatkov.
~10704	Ohranjanje predhodnih podatkov	Prejšnja verzija ostane dostopna v verzijski zgodovini.	Verzijsko sledenje omogočeno.	Predhodne vrednosti niso izgubljene.

2.3.5.8.5 Diagram primerov uporabe

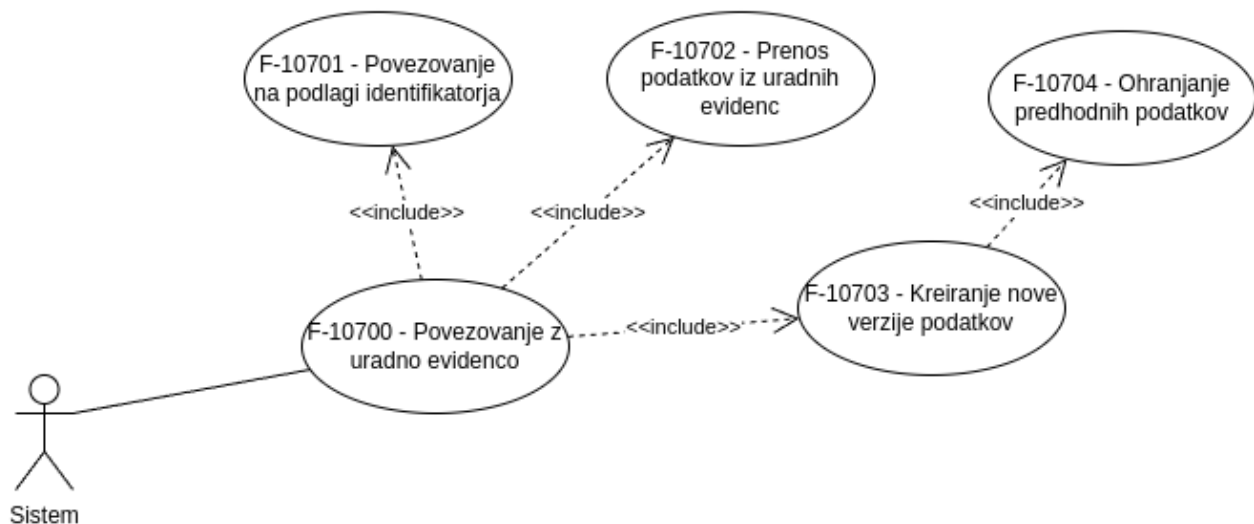


Diagram 49: Diagram primerov uporabe za SKL-ZIV-F-10700.

2.3.5.9 SKL-ZIV-F-10800: Sinhronizacija podatkov živali z uradno evidenco

2.3.5.9.1 Opis funkcionalnosti

Funkcionalnost omogoča sinhronizacijo podatkov živali med sistemom POKK in uradnimi evidencami.

Pri sinhronizaciji se podatki iz uradnih evidenc primerjajo z lokalnimi podatki in, v primeru sprememb, prepisujejo le tiste podatke, ki so označeni kot uradni.

Vsaka sprememba povzroči kreacijo nove verzije osnovnih podatkov živali, s čimer je zagotovljena popolna sledljivost sprememb.

Neuradni podatki in uporabniško vneseni podatki ostanejo nespremenjeni.

Glavne značilnosti:

- Pridobitev aktualnih podatkov o živali iz uradnih evidenc preko komunikacijskega modula.
- Primerjava lokalnih podatkov z uradnimi podatki.
- Prepis samo tistih atributov, ki so določeni kot uradni (npr. datum rojstva, spol, species).
- Samodejna kreacija nove verzije podatkov ob vsaki spremembi.
- Ohranjanje predhodnih vrednosti v verzijski zgodovini.
- Revizijsko beleženje vsake spremembe ob sinhronizaciji.

2.3.5.9.2 Predpogoji

- Aktivna povezava z uradnimi evidencami preko komunikacijskega modula.
- Lokalna žival obstaja in ima ustrezno povezavo na uradne identifikatorje.
- Verzijsko sledenje sprememb in revizijsko beleženje sta aktivna.

2.3.5.9.3 Merila uspešnosti

- Sinhronizacija za eno žival izvedena v ≤ 2 s po sprožitvi procesa.
- Samo uradni podatki so prepisani, neuradni podatki ostanejo nespremenjeni.
- Kreacija nove verzije podatkov pri vsaki spremembi.
- Vse spremembe so revizijsko zabeležene.

2.3.5.9.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~10801	Pridobitev podatkov iz uradnih evidenc	Sistem pridobi aktualne podatke o živali prek komunikacijskega modula.	Povezava z uradnimi evidencami operativna.	Podatki pridobljeni v ≤ 2 s.
~10802	Primerjava podatkov	Primerjava lokalnih in uradnih podatkov za zaznavo sprememb.	Aktivni lokalni podatki.	Vse spremembe pravilno zaznane.
~10804	Kreiranje nove verzije zapisa z posodobljenimi podatki	Ob vsaki spremembi se samodejno kreira nova verzija osnovnih podatkov.	Verzijsko sledenje aktivno.	Nova verzija ustvarjena takoj po spremembi, neuradni podatki ostanejo nespremenjeni.

2.3.5.9.5 Diagram primerov uporabe

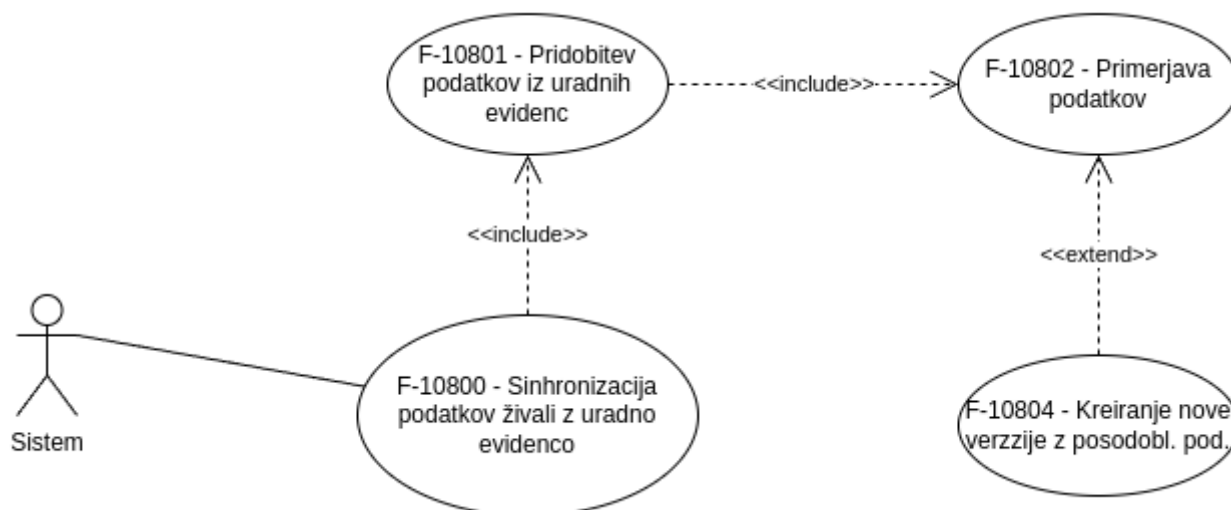


Diagram 50: Diagram primerov uporabe za SKL-ZIV-F-10800.

2.3.5.10 SKL-ZIV-F-10900: Spremljanje ključnih dejstev živali (Statusno spremljanje)

2.3.5.10.1 Opis funkcionalnosti

Funkcionalnost omogoča celovit pregled ključnih dejstev, povezanih z življenjskim ciklom posamezne živali v sistemu POKK.

Dejstva se **samodejno** prevzemajo iz vnosov v drugih sklopih sistema, kot so npr.:

- **SKL-GEN** (genealoški podatki, npr. datum rojstva, potrjena starša),
- **SKL-DOG** (dogodki, npr. odvzem vzorca, osemenitev, rojstvo, smrt),
- **SKL-PAS** (pasma živali, npr. klasifikacija, čistost).

Vsako dejstvo ima jasno opredeljen datum, kraj, vrsto dogodka in povezavo na izvorni podatek.

Dejstva morajo vedno odražati aktualno realno stanje — SUPB mora zagotoviti, da se ta podatkovna slika avtomatsko preračunava in osvežuje ob spremembah v teh sklopih (trigger ob potrditvah dogodkov, dokumentov, premikov).

Glavne značilnosti:

- Samodejni prevzem podatkov iz podležnih sklopov.
- Beleženje ključnih datumov, krajev in tipov dogodkov.
- Povezava vsakega dejstva z njegovim virom (dogodek, dokument, genealoški zapis, premik).
- Takojšnja posodobitev dejstev ob vsaki spremembi vira (trigger SUPB).
- Verzijsko beleženje vseh sprememb.

2.3.5.10.2 Predpogoji

- Uporabnik je avtenticiran in ima avtorizacijo za dostop do podatkov o živali.
- Podatki v podležnih sklopih so vnešeni in veljavni.
- Verzijsko in revizijsko sledenje mora biti omogočeno.
- SUPB mora zagotoviti avtomatsko osveževanje dejstev.

2.3.5.10.3 Merila uspešnosti

- Nov vnos ali sprememba dejstva izvedena v ≤ 2 s po spremembi vira.
- Vsa dejstva morajo biti skladna s trenutnim stanjem vseh povezanih sklopov.
- Verzijska sledljivost vseh sprememb mora biti zagotovljena.

2.3.5.10.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
10901	Prevzem ključnih dejstev	Samodejni prevzem datumov, krajev in tipov dogodkov iz sklopov.	Aktivna evidenca v vseh sklopih.	Prevzem izveden v ≤ 2 s po potrditvi spremembe v viru.
10902	Prikaz ključnih dejstev za žival	Strukturiran prikaz vseh pomembnih datumov, krajev in tipov za posamezno žival.	Izbrana žival.	Prikaz podatkov v ≤ 1 s.
10903	Povezava dejstev z izvorom	Vsako dejstvo povezano s svojim virom (dogodek, dokument, genealoški zapis, premik).	Aktivna povezava na vir.	Povezava ustrezno prikazana.
10904	Verzijsko beleženje sprememb dejstev	Vsaka sprememba dejstva verzijsko evidentirana.	Verzijsko sledenje aktivno.	Predhodne verzije dostopne.
10905	Samodejno preračunavanje dejstev	SUPB sproži avtomatsko posodobitev dejstev ob spremembi vira (trigger).	Potrjen dogodek, dokument, premik ali genealoški podatek.	Dejstva posodobljena avtomatsko in pravilno v ≤ 2 s.

2.3.5.10.5 Diagram primerov uporabe

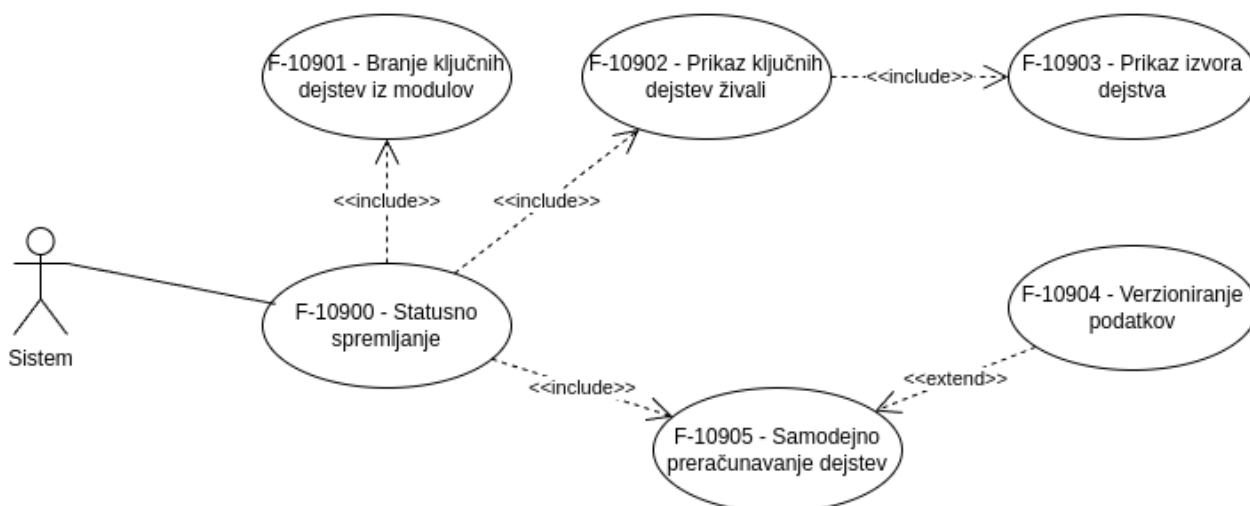


Diagram 51: Diagram primerov uporabe za SKL-ZIV-F-10900.

2.3.5.11 SKL-ZIV-F-11000: Vse živali pri rejcu

2.3.5.11.1 Opis funkcionalnosti

Funkcionalnost omogoča prikaz vseh živali, ki jih ima posamezen rejec v svoji lasti oziroma oskrbi. Podatki so prikazani po **živalskih vrstah** (govedo, prašiči, kopitarji, drobnica, čebele ipd.) in vključujejo:

- število živali po podvrsti ali kategoriji (npr. krave, teleta, svinje, ovce),
- pasmo,
- lokacijo (npr. pašnik, hlev),
- dodatne opise ali opombe (npr. kontrolor, status).

Za vsako skupino je na voljo gumb "**Uredi/dodaj**", ki uporabnika preusmeri v ustrezen modul aplikacije za nadaljnje delo z živalmi tiste vrste (npr. eGovedo, SKL-Pra, SKL-Drob).

Funkcionalnost predstavlja centralno izhodišče za rejce in upravne uporabnike, ki želijo hitro dostopati do pregleda in urejanja stanja črede, jate ali skupine.

2.3.5.11.2 Predpogoji

- Uporabnik mora imeti aktivno izbrano zastopništvo rejca.
- Uporabnik mora imeti pravice za dostop do podatkov o živalih.
- Podatki o vrstah, pasmah, lokacijah in kategorijah morajo biti sinhronizirani in validni.

2.3.5.11.3 Merila uspešnosti

- Seznam se prikaže v ≤ 1 s.
- Vsaka povezava do urejevalnega modula deluje brez napake.
- Število prikazanih skupin ustreza dejanskemu stanju v bazi.
- Klik na vrsto ne osveži celotne strani, temveč sproži ciljno preusmeritev.

2.3.5.11.4

Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~11000	Prikaz vseh živali pri rejcu	Prikaz vseh živali, razvrščenih po vrstah, pasmah in lokacijah, z gumbi za urejanje in možnostjo filtriranja.	Avtoriziran rejec z veljavnim kontekstom zastopanja	Prikaz v ≤ 1 s, vsi podatki skladni z bazo
~11001	Preusmeritev v modul po vrsti	Klik na "Uredi/dodaj" preusmeri v funkcionalni modul za upravljanje izbrane vrste.	Modul aktiviran, URL konfiguriran	Preusmeritev uspešna brez zamika
~11002	Prikaz dodatnih podatkov	Prikaz dodatnih opisov (npr. kontrolor, opombe, posebnosti).	Podatki obstajajo v evidenci	Prikazano ob vsaki skupini, konsistentno

2.3.5.11.5

Diagram primerov uporabe

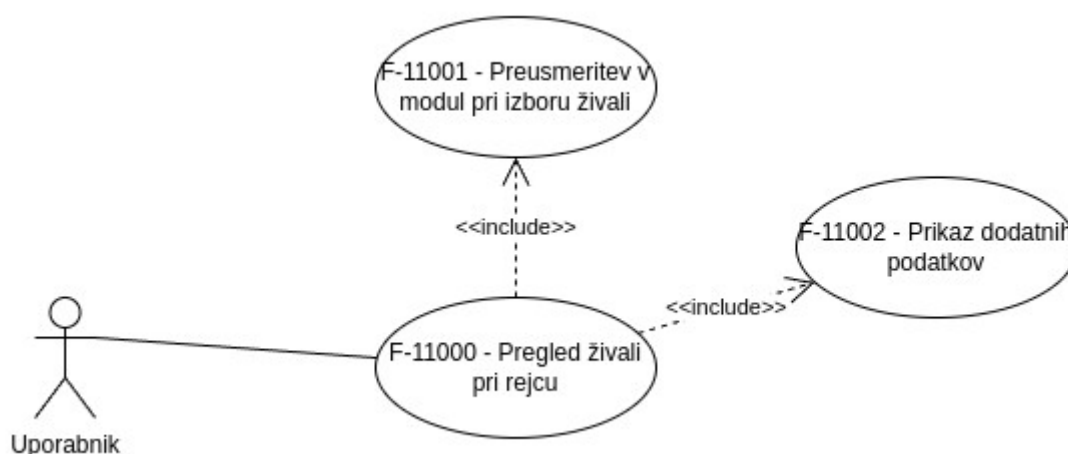


Diagram 52: Diagram primerov uporabe za SKL-ZIV-F-11000.

2.3.5.12

SKL-ZIV-F-11100: Vnos skupine živali

2.3.5.12.1

Opis funkcionalnosti

Funkcionalnost omogoča **vnos skupine živali**, kjer se podatki vnesejo hitro in učinkovito znotraj ene vnosne maske. Namenjena je situacijam, ko se hkrati registrira več živali s podobnimi podatki (npr. po kotitvi, po uvozu, po odvzemu vzorcev ali ob skupinskem preselitvenem dogodku).

Za vsako žival se vnesejo naslednji atributi:

- rejec, izvajalec, datum vnosa,
- datum rojstva in morebitni datum odvzema,
- spol, pasma, ime, identifikacijska številka (rodovni in tetovirani podatki),
- oče in mati (po možnosti iz šifrant ali neposredno),
- izvor, delna avtentikacija porekla (del. avt. p.),
- dodatne opombe.

Vnos se lahko izvaja zaporedno (navigacija med živalmi z gumboma < >) in vključuje možnost

razveljavitve ali potrditve posameznega vnosa.

2.3.5.12.2 Predpogoji

- Uporabnik mora biti avtoriziran za vnos v kontekstu danega rejca.
- Pasma, spol in datumi morajo biti skladni s sistemskimi šifranti.
- Vse identifikacijske številke morajo biti unikatne (če so podane).
- Vsaj osnovni obvezni atributi morajo biti vneseni za shranjevanje.

2.3.5.12.3 Merila uspešnosti

- Vnos ene živali se zaključi v ≤ 5 s.
- Navigacija med vnosi omogoča enostavno zaporedno dodajanje brez ponovnega nalaganja strani.
- Sistem preveri obvezna polja in prikaže jasna opozorila pri napakah.
- Vsaka žival je po vnosu samodejno vidna v pregledih posamezne vrste.
- Vsi vnosi so revizijsko sledeni.

2.3.5.12.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~11101	Vnos podatkov o živalih	Hiter vnos posameznih živali z identifikacijskimi in osnovnimi rejsko-genetskimi podatki.	Avtorizacija, aktivni rejec, konfigurirani šifranti	Vnos ≤ 5 s, validacija ob shranjevanju
~11102	Navigacija med vnosi	Gumbi za zaporedni vnos več živali brez prekinitev in ponovnega nalaganja maske.	Predhodno shranjena vsaj ena žival	Prehod v ≤ 300 ms
~11103	Razveljavitev vnosa	Možnost preklica vnosa pred shranjevanjem za poljubno žival ali celoto.	Dogodek še ni potrjen	Razveljavitev brez vpliva na druge zapise

2.3.5.12.5 Diagram primerov uporabe

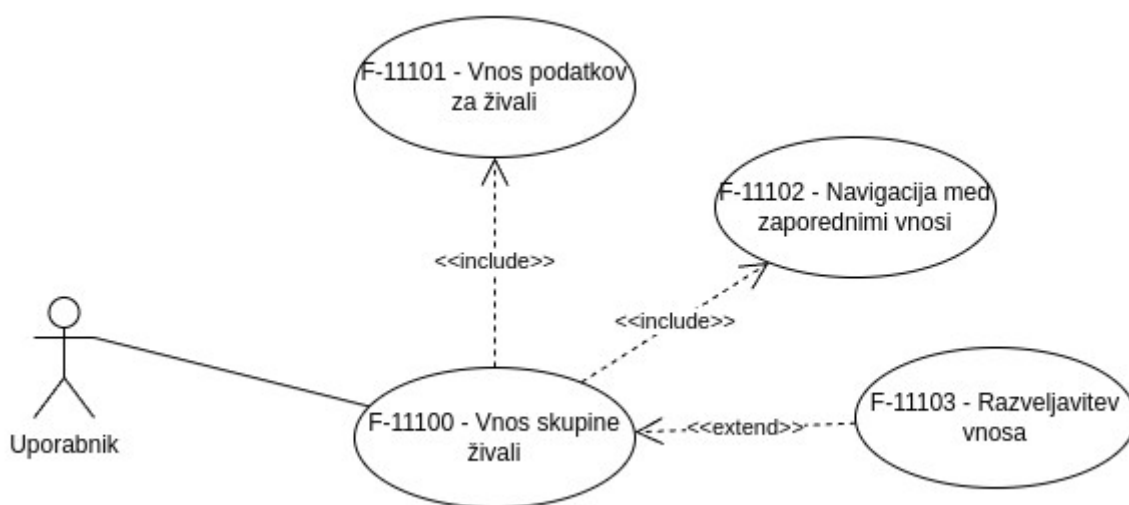


Diagram 53: Diagram primerov uporabe za SKL-ZIV-F-11100.

2.3.6 SKL-Pas: Sklop pasma

2.3.6.1 SKL-PAS-F-12000: Pregled pasemskih klasifikacij (Pregled in filtriranje)

2.3.6.1.1 Opis funkcionalnosti

Funkcionalnost omogoča pregled in filtriranje vseh definiranih pasem živalskih vrst v sistemu POKK.

Uporabnik lahko išče pasme glede na vrsto živali, tip klasifikacije (izgled, genetska, administrativna), status veljavnosti in druge attribute.

Seznam omogoča enoten pregled vseh registriranih pasem in dostop do nadaljnjih operacij (urejanje, deaktivacija, kreiranje novih pasem).

Prikazani podatki vključujejo osnovne informacije o pasmi, tip klasifikacije, status obračuna in časovno veljavnost.

Glavne značilnosti:

- Pregled vseh pasemskih klasifikacij v sistemu.
- Filtriranje po vrsti živali, tipu klasifikacije, imenu pasme, statusu veljavnosti in obračunskem statusu.
- Iskanje po delnem imenu pasme (substring matching).
- Prikaz osnovnih atributov: naziv pasme, vrsta živali, tip klasifikacije, status, obračunski status, veljavnost.
- Stranjenje in sortiranje rezultatov.
- Povezava na podrobnosti posamezne pasme za nadaljnje operacije.

2.3.6.1.2 Predpogoji

- Uporabnik mora biti avtenticiran in avtoriziran za dostop do administrativnega dela pasemskih klasifikacij.
- Podatki o vrstah živali in tipih klasifikacij morajo biti predhodno definirani.

2.3.6.1.3 Merila uspešnosti

- Nalaganje seznama pasem (do 100 zapisov) mora biti izvedeno v ≤ 1.5 s.
- Filtriranje in iskanje morata vrniti pravilne rezultate v ≤ 1.5 s.
- Rezultati morajo biti točni glede na izbrane filtre in prikazani v strukturirani obliki.

2.3.6.1.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~12101	Prikaz seznama pasemskih klasifikacij	Prikaz vseh registriranih pasem z osnovnimi atributi.	Avtentikacija in avtorizacija uspešna.	Seznam naložen v ≤ 1.5 s.
~12102	Filtriranje pasemskih klasifikacij	Uporabnik filtrira seznam po vrsti živali, tipu klasifikacije, statusu veljavnosti in obračunskem statusu.	Veljavni filtri na voljo.	Filtriranje izvedeno v ≤ 1.5 s.
~12104	Ostranjevanje in sortiranje rezultatov	Možnost prikaza rezultatov po straneh in sortiranja po poljubnih stolpcih.	Veljavni podatki v bazi.	Stranjenje in sortiranje brezhibno do 1M zapisov.
~12105	Povezava na podrobnosti pasme	Možnost dostopa do podrobnosti posamezne pasme za nadaljnje urejanje.	Aktivna pasemska klasifikacija.	Povezava vzpostavljena brez napak.

2.3.6.1.5 Diagram primerov uporabe

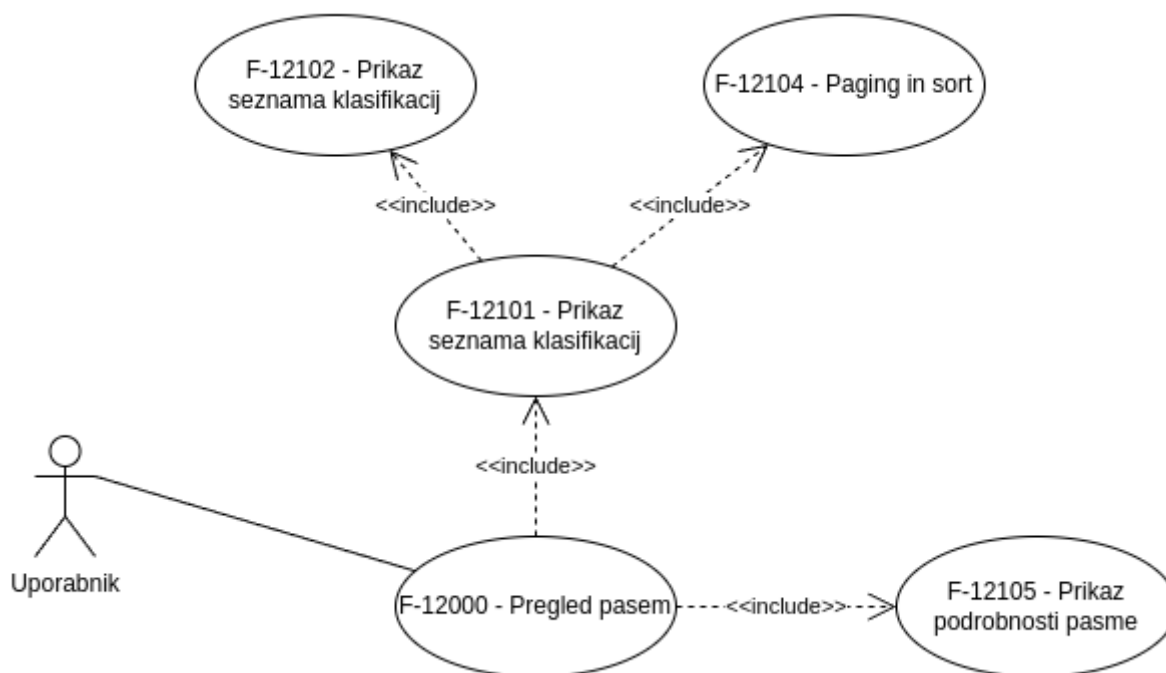


Diagram 54: Diagram primerov uporabe za SKL-PAS-F-12102.

2.3.6.2 SKL-PAS-F-12100: Kreiranje pasemske klasifikacije

2.3.6.2.1 Opis funkcionalnosti

Funkcionalnost omogoča vnos nove pasemske klasifikacije za izbrano vrsto živali v sistemu POKK. Uporabnik vnese osnovne podatke o pasmi, določi tip klasifikacije (npr. izgled, genetska, administrativna) in nastavi obračunski status pasme.

Pri kreiranju se lahko določi tudi povezava na starševske pasme (če gre za hibridno pasmo).

Če starševska pasma ni določena, se šteje, da gre za čistokrvno pasmo.

Sistem izvede validacijo vnesenih podatkov, preveri konsistentnost strukture pasme ter zagotovi osnovne kontrole nad vnosom.

Ob kreiranju pasme se določi tudi časovna veljavnost (začetek veljavnosti).

Glavne značilnosti:

- Vnos osnovnih podatkov: naziv pasme, vrsta živali, tip klasifikacije, obračunski status.
- Možnost določanja starševskih pasem (hibridne povezave).
- Samodejna nastavitve čistokrvnosti, če staršev ni izbranih.
- Definiranje začetnega datuma veljavnosti.
- Validacija obveznih polj, pravilnosti starševskih povezav in osnovne strukture.
- Zapis v sistem z možnostjo nadaljnje verzijske obdelave.

2.3.6.2.2 Predpogoji

- Uporabnik je avtentificiran in ima avtorizacijo za upravljanje pasemskih klasifikacij.
- Predhodno so definirane vrste živali in tipi klasifikacij.

2.3.6.2.3 Merila uspešnosti

- Vnos nove pasme mora biti izveden v ≤ 2 s po potrditvi vnosa.
- Vsi obvezni podatki morajo biti pravilno validirani pred shranjevanjem.
- Starševske povezave morajo biti logične in skladne (če določene).
- Vzpostavljena časovna veljavnost pasemske klasifikacije.

2.3.6.2.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~12101	Vnos osnovnih podatkov pasme	Uporabnik vnese naziv, vrsto živali, tip klasifikacije in obračunski status.	Avtentikacija in avtorizacija uspešna.	Vsi obvezni podatki prisotni in pravilno validirani.
~12102	Določitev starševskih pasem	Možnost določanja staršev za kreiranje hibridne pasme.	Predhodno vnesene starševske pasme.	Starševske povezave pravilno evidentirane.
~12103	Nastavitev začetnega datuma veljavnosti	Ob vnosu se določi datum začetka veljavnosti pasemske klasifikacije.	Veljavni datum vnosa.	Časovna veljavnost pravilno nastavljena.
~12104	Validacija vnosa	Sistem preveri pravilnost vseh podatkov pred shranjevanjem.	Aktivna pravila validacije.	Napačen vnos zavržen, uporabniku podana jasna napaka.
~12105	Shranjevanje nove pasme	Uspešno shranjevanje pasme v podatkovno bazo z inicialno verzijo.	Vsi prejšnji koraki uspešni.	Pasma dosegljiva za nadaljnje operacije v ≤ 2 s.

2.3.6.2.5 Diagram primerov uporabe

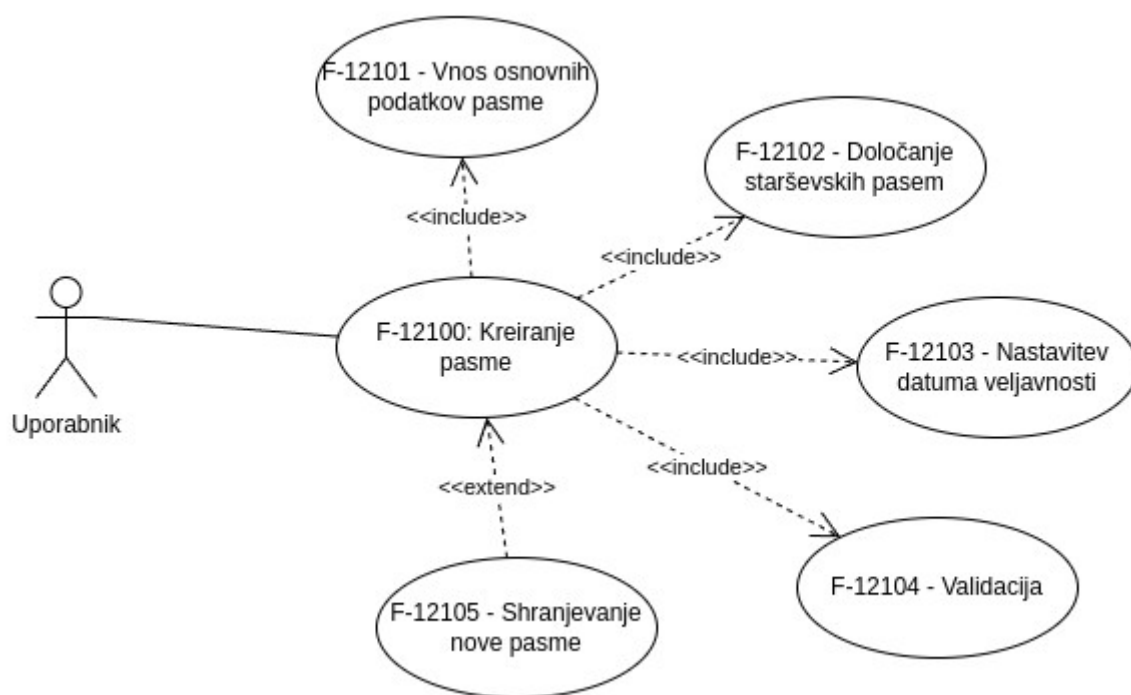


Diagram 55: Diagram primerov uporabe za SKL-PAS-F-12100.

2.3.6.3 SKL-PAS-F-12300: Posodabljanje pasemske klasifikacije

2.3.6.3.1 Opis funkcionalnosti

Funkcionalnost omogoča posodabljanje obstoječe pasemske klasifikacije v sistemu POKK. Posodobitve so razdeljene glede na vpliv na izračune:

- **Posodabljanje atributov, ki ne vplivajo na izračun** (npr. naziv pasme, opis):
Uredi se obstoječe vrednosti, pri čemer se kreira nova verzija zapisa s časovno veljavnostjo od datuma spremembe naprej.

- **Posodabljanje atributov, ki vplivajo na izračun** (npr. sprememba starševskih pasem, sprememba tipa klasifikacije):

Uporabnik mora ob spremembi določiti, ali želi:

- **Preračun deležev za nazaj** (rekalkulacija na celotni zgodovini podatkov), ali
- **Preračun deležev od novega datuma naprej** (sprememba vpliva le na prihodnje izračune).

V obeh primerih sistem kreira novo verzijo pasemske klasifikacije in sproži ustrezne preračune deležev pri vseh prizadetih živalih, skladno z določenim načinom rekalkulacije.

Glavne značilnosti:

- Ločeno posodabljanje atributov glede na vpliv na izračun (nazivi/opisi vs. starši/tip).
- Kreacija nove verzije pasemske klasifikacije ob vsaki spremembi.
- Uporabnik določi način rekalkulacije deležev (za nazaj ali od spremembe naprej).
- Samodejna sprožitev preračuna deležev pasem pri vseh prizadetih živalih.
- Zagotovljeno verzijsko sledenje vseh sprememb.

2.3.6.3.2 Predpogoji

- Uporabnik mora biti avtentificiran in avtoriziran za urejanje pasemskih klasifikacij.
- Sistem mora imeti aktivno verzijsko in revizijsko beleženje sprememb.
- Predhodno definirane pasme in tipi klasifikacij.

2.3.6.3.3 Merila uspešnosti

- Posodobitev atributov brez vpliva na izračun izvedena v ≤ 2 s.
- Pri spremembi atributov z vplivom na izračun mora biti sprožen rekalkulacijski proces.
- Vse spremembe so verzijsko zabeležene, predhodne verzije ostanejo dostopne.
- Rekalkulacija deležev pri vseh prizadetih živalih izvedena skladno z izbranim načinom.

2.3.6.3.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~12301	Posodobitev atributov brez vpliva na izračun	Uporabnik posodobi naziv ali opis pasme brez vpliva na obračune.	Avtentikacija in avtorizacija uspešna.	Nova verzija kreirana, sprememba vidna v ≤ 2 s.
~12302	Posodobitev atributov z vplivom na izračun	Uporabnik spremeni starševske pasme ali tip klasifikacije.	Aktivna verzijska shema.	Obvezno določanje načina rekalkulacije.
~12303	Določitev načina preračuna deležev	Uporabnik izbere: rekalkulacija za nazaj ali od novega datuma naprej.	Na voljo izbor načina rekalkulacije.	Izbrani način se pravilno aplicira.
~12304	Kreacija nove verzije pasme	Ob vsaki spremembi se ustvari nova verzija pasemske klasifikacije.	Verzijsko sledenje aktivno.	Časovna veljavnost pravilno nastavljena.
~12305	Sprožitev preračuna deležev pri živalih	Sistem sproži rekalkulacijo deležev pri vseh živalih te pasme.	Sprememba vpliva na genealoške povezave.	Preračuni izvedeni brez napak, upoštevajo verzijsko stanje.

2.3.6.3.5 Diagram primerov uporabe

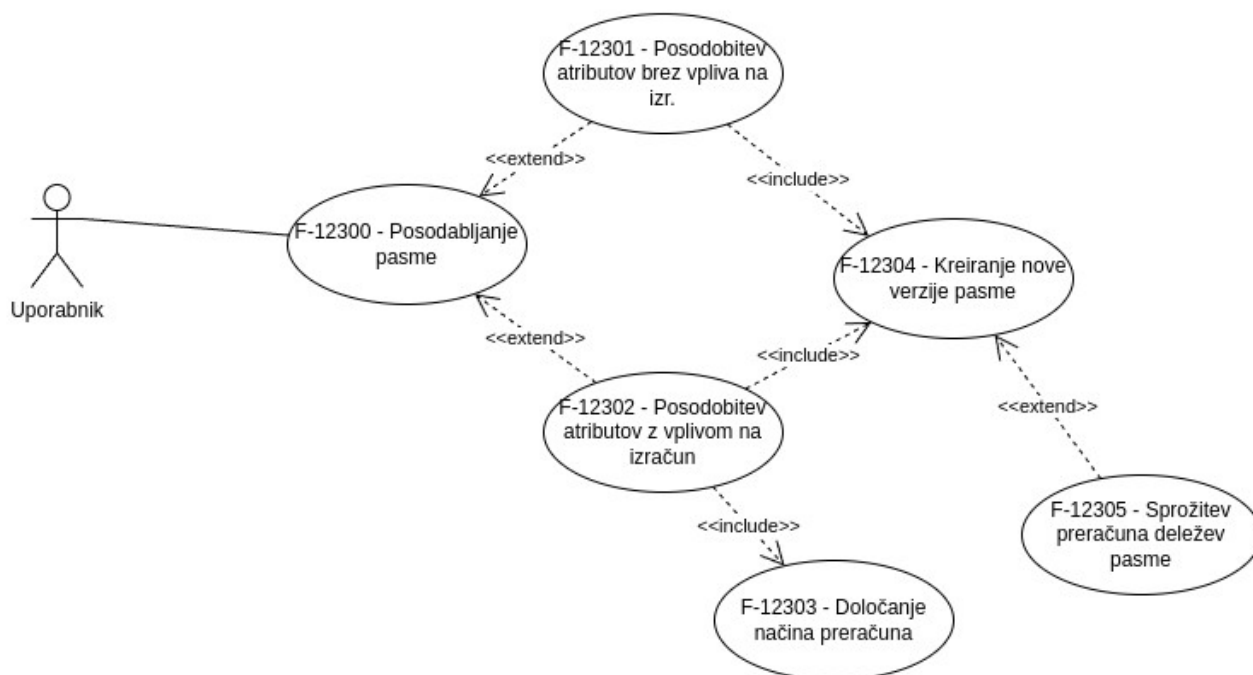


Diagram 56: Diagram primerov uporabe za SKL-PAS-F-12300.

2.3.6.4 SKL-PAS-F-12400: Deaktivacija pasemske klasifikacije

2.3.6.4.1 Opis funkcionalnosti

Funkcionalnost omogoča deaktivacijo (mehko brisanje) obstoječe pasemske klasifikacije v sistemu POKK.

Deaktivacija pomeni, da pasma postane neaktivna in se ne uporablja več v novih izračunih.

Ob deaktivaciji uporabnik izbere način ravnanja z že obstoječimi izračuni:

- **Popolna deaktivacija:**

Pasma se deaktivira in vsi obstoječi izračuni, kjer je ta pasma upoštevana, se zaključijo oziroma deaktivirajo.

- **Prenehanje uporabe v novih izračunih:**

Pasma ostane v starih izračunih nespremenjena, vendar se v prihodnjih izračunih več ne upošteva in je ni mogoče določati živalim.

Deaktivacija ne briše fizičnih podatkov iz baze, ampak nastavi končno veljavnost obstoječih zapisov.

Glavne značilnosti:

- Mehka deaktivacija pasemske klasifikacije (nastavitev končnega datuma veljavnosti).
- Izbira načina obravnave že izvedenih izračunov (popolna deaktivacija ali samo ustavitev za naprej).
- Ohranjanje predhodnih verzij v sistemu za revizijske potrebe.
- Pravilno upravljanje preračunov glede na izbrani način deaktivacije.

2.3.6.4.2 Predpogoji

- Uporabnik mora biti avtentificiran in avtoriziran za upravljanje pasemskih klasifikacij.
- Pasemska klasifikacija mora biti aktivna in brez napak v genealogskih povezavah.

2.3.6.4.3 Merila uspešnosti

- Deaktivacija pasemske klasifikacije izvedena v ≤ 2 s po potrditvi.
- Pravilno izvedena izbrana obravnava obstoječih izračunov (popolna deaktivacija ali samo za naprej).
- Vsi podatki ostanejo dostopni v verzijski zgodovini.

2.3.6.4.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
12401	Deaktivacija pasemske klasifikacije	Mehka deaktivacija obstoječe pasme z nastavitvijo končnega datuma veljavnosti.	Avtentikacija in avtorizacija uspešna.	Deaktivacija izvedena v ≤ 2 s.
12402	Izbira načina obravnave obstoječih izračunov	Uporabnik izbere: popolna deaktivacija vseh izračunov ali ustavitev le za prihodnost.	Aktivni izračuni obstajajo.	Izbrana strategija pravilno izvedena.
12403	Nastavitev statusa pasme	Pasma postane neaktivna v sistemu in se ne upošteva v novih izračunih.	Deaktivacija uspešna.	Status pasme pravilno nastavljen.

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
12404	Ohranjanje zgodovine pasme	Predhodne verzije in obstoječi podatki ostanejo dostopni v sistemu.	Verzijsko sledenje aktivno.	Dostopnost zgodovinskih podatkov zagotovljena.
12405	Deaktivacija izračunov pri živalih	Ob izboru deaktivacije pasme za nazaj se obstoječi izračuni deaktivirajo (mehko izbrišejo)	Izbran je način deaktivacije vseh obstoječih izračunov.	Izračuni se deaktivirajo.

2.3.6.4.5 Diagram primerov uporabe

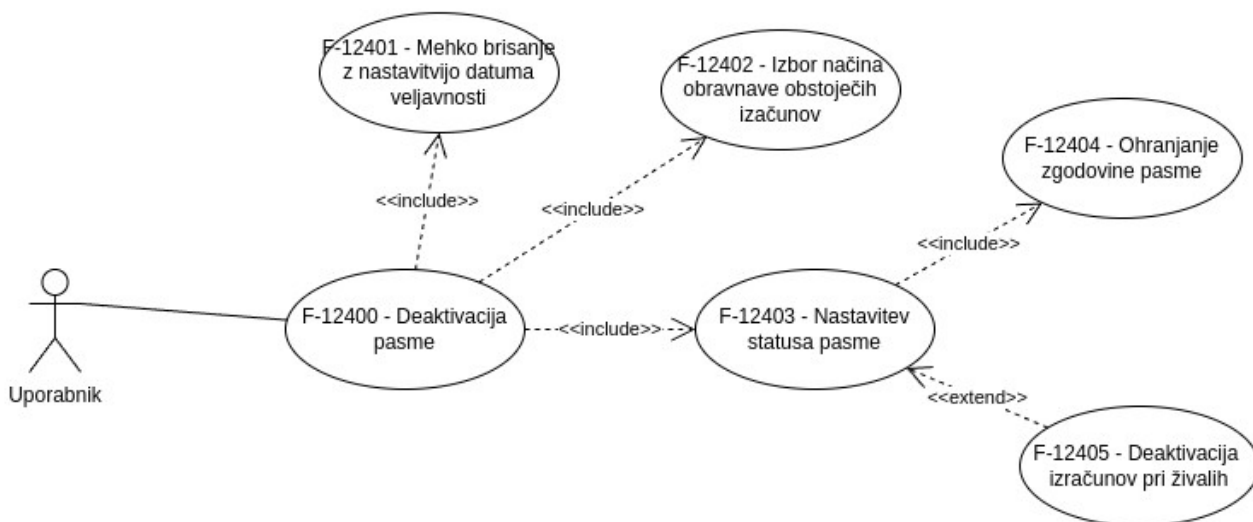


Diagram 57: Diagram primerov uporabe za SKL-PAS-F-12400.

2.3.6.5 SKL-PAS-F-12500: Upravljanje pasemskih podatkov pri živali

2.3.6.5.1 Opis funkcionalnosti

Funkcionalnost omogoča upravljanje vseh pasemskih klasifikacij, ki so povezane z določeno živaljo v sistemu POKK.

Uporabnik ima vpogled v seznam vseh pasemskih zapisov, ki so bili bodisi **avtomatsko izračunani** bodisi **vnešeni ročno**.

V sklopu funkcionalnosti je omogočeno:

- Dodajanje novih pasemskih zapisov, če je ročni vnos za določeno pasemsko klasifikacijo dovoljen.
- Brisanje (mehka deaktivacija) obstoječih pasemskih zapisov pri živali.
- Popravljanje obstoječih avtomatsko izračunanih pasemskih zapisov s spremembo deleža, kjer sistem ločeno vodi **izračunan delež** in **popravljen delež**.
- Pri popravljenih pasemskih zapisih pravila za ponovni izračun **ne veljajo več** — tak popravljen zapis je fiksiran.

Glavne značilnosti:

- Prikaz vseh aktivnih pasemskih zapisov živali z atributi (tip klasifikacije, delež, status vnosa).
- Možnost filtriranja pasem glede na tip klasifikacije, status (izračunan/ročno vnešen/popravljen) in veljavnost.
- Ročni vnos novega pasemskega zapisa, če je ročni vnos dovoljen.
- Brisanje (mehka deaktivacija) pasemskega zapisa pri živali.
- Popravljanje avtomatsko izračunanega deleža in ločeno shranjevanje izračunanega ter popravljenega deleža.
- Zagotovljeno verzijsko sledenje sprememb in jasno označevanje popravljenih podatkov.

2.3.6.5.2 Predpogoji

- Uporabnik mora biti avtenticiran in avtoriziran za upravljanje pasemskih podatkov.
- Aktivna povezava na veljavne pasemske klasifikacije in pravila dovoljenosti ročnih vnosov.
- Aktivno verzijsko in revizijsko sledenje sprememb.

2.3.6.5.3 Merila uspešnosti

- Nalaganje seznama pasemskih zapisov za žival $v \leq 1.5$ s.
- Dodajanje novega ročnega zapisa izvedeno $v \leq 2$ s.
- Popravljanje izračunanih zapisov zabeleženo pravilno, izračun ne prepisuje popravljenih vrednosti.
- Mehka deaktivacija pasemskih zapisov izvedena brez napake, zapis ostane v zgodovini.

2.3.6.5.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
12501	Pregled pasemskih zapisov pri živali	Prikaz vseh aktivnih pasemskih zapisov za izbrano žival.	Avtentikacija in avtorizacija uspešna.	Seznam naložen v ≤ 1.5 s.
12502	Dodajanje novega pasemskega	Ročni vnos novega zapisa, če pasemska klasifikacija to	Aktivna pasemska klasifikacija z	Dodajanje izvedeno v ≤ 2 s, podatki pravilno

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
	zapisa	dovoljuje.	ročnim vnosom.	evidentirani.
12503	Brisanje (deaktivacija) pasemskega zapisa	Mehka deaktivacija obstoječega zapisa pri živali.	Aktivni zapis obstaja.	Zapis označen kot neaktiven brez fizičnega brisanja.
12504	Popravljanje izračunanega deleža	Sprememba izračunanega deleža pasemske klasifikacije z ločenim beleženjem izračunanega in popravljenega deleža.	Obstoječi avtomatski zapis.	Popravljen delež evidentiran ločeno, preračuni za naprej ne prepišejo popravka.

2.3.6.5.5 Diagram primerov uporabe

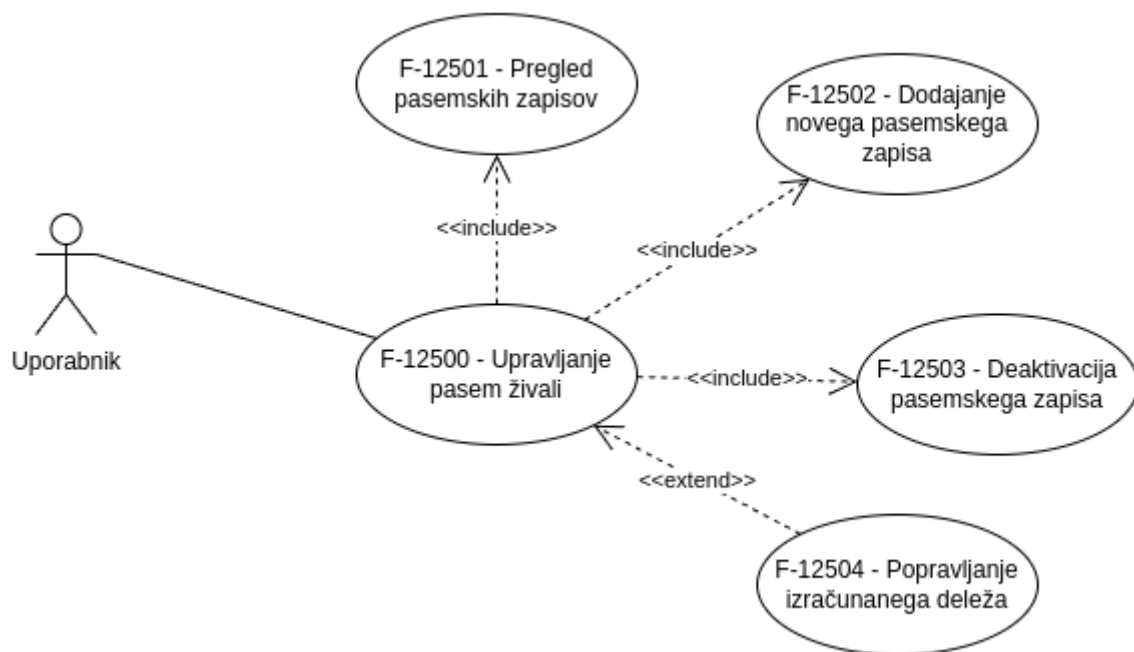


Diagram 58: Diagram primerov uporabe za SKL-PAS-F-12500.

2.3.6.6 SKL-PAS-F-12600: Izračun deležev pasem

2.3.6.6.1 Opis funkcionalnosti

Funkcionalnost omogoča izračun oziroma preračun deležev pasem pri živalih v sistemu POKK. Preračun lahko poteka na več nivojih, pri čemer se vedno upošteva verzijsko sledenje sprememb:

- **Ročno proženje preračuna na nivoju pasme:**
Za izbrano pasmo se sproži preračun deležev pri vseh živalih, ki imajo to pasmo v svoji genealogiji. Ustvari se nova verzija izračuna.
- **Ročno proženje preračuna na nivoju sistema:**
Sistem sproži preračun vseh pasem za vse živali (npr. ob večjih spremembah pasemskih struktur). Ustvari se nove verzije vseh prizadetih izračunov.
- **Ročno proženje preračuna na nivoju posamezne živali:**
Za izbrano žival se sproži preračun vseh pasemskih deležev na osnovi njene aktualne genealogije. Ustvari se nove verzije vseh zapisov pasem pri tej živali.
- **Ročno proženje preračuna na nivoju pasme pri posamezni živali:**
Za izbrano pasemsko klasifikacijo na določeni živali se sproži ciljni preračun in kreira nova verzija izračuna.

Glavne značilnosti:

- Možnost selektivnega ali celovitega preračuna glede na nivo (pasma, sistem, žival, pasma na živali).
- Samodejna kreacija nove verzije vsakega spremenjenega izračuna.
- Ohranjanje predhodnih vrednosti v verzijski zgodovini.
- Upoštevanje vseh sprememb genealogije in sprememb v definicijah pasem.
- Hitro izvajanje operacij z obvladovanjem velikega števila izračunov.

2.3.6.6.2 Predpogoji

- Aktivne povezave na veljavne pasemske klasifikacije in genealogijo.
- Aktivno verzijsko sledenje sprememb izračunov.
- Uporabnik mora biti avtentificiran in imeti pravice za sprožanje preračunov.

2.3.6.6.3 Merila uspešnosti

- Preračun na nivoju sistema zaključen brez izgub podatkov in z verzijsko sledljivostjo.
- Vsaka nova verzija vsebuje jasno razliko glede na prejšnjo verzijo.
- Popravki (ročno popravljeni deleži) se pri preračunih ne prepisujejo.

2.3.6.6.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
12601	Preračunavanje na nivoju pasme	Preračun deležev za vse živali, povezane z izbrano pasmo.	Avtentikacija in pravica za preračun.	Nova verzija za vsako spremembo kreirana pravilno.
12602	Preračunavanje na nivoju sistema	Preračun vseh pasem za vse živali v sistemu.	Aktivne pasemske in genealogijske povezave.	Izračuni izvedeni brez napak in v ustreznem času.
12603	Preračunavanje na	Preračun vseh pasem	Veljavna genealogija	Nova verzija vseh

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
	nivoju živali	pri eni izbrani živali.	in pasemski podatki.	deležev ustvarjena za žival.
12604	Preračunavanje na nivoju pasme na živali	Preračun določene pasemske klasifikacije na eni izbrani živali.	Aktivni zapis pasemske klasifikacije na živali.	Samo ta delež posodobljen, nova verzija shranjena.
12605	Ohranjanje popravkov pri preračunih	Pri preračunih se ročno popravljene deleži ne prepisujejo.	Označeni popravki obstajajo.	Popravljeni zapisi ostanejo nespremenjeni.

2.3.6.6.5 Diagram primerov uporabe

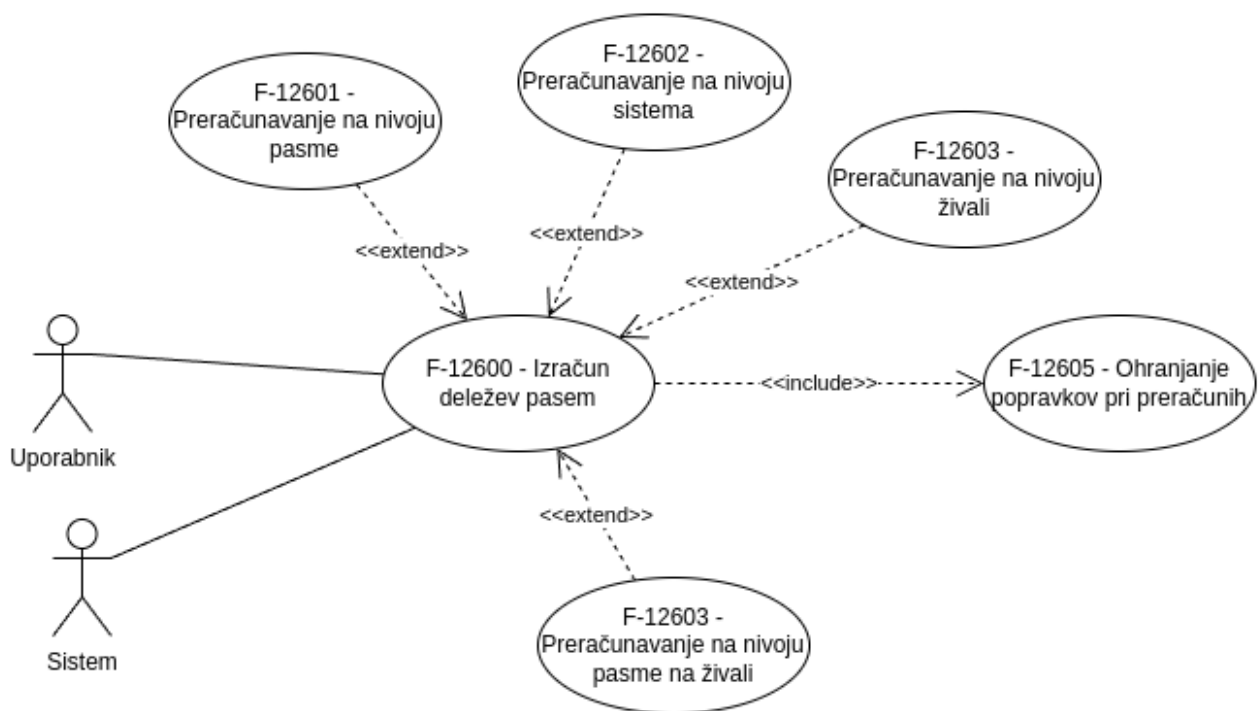


Diagram 59: Diagram primerov uporabe za SKL-PAS-F-12600.

2.3.7 SKL-Gen: Sklop genealogija

2.3.7.1 SKL-GEN-F-14000: Upravljanje z genealogijo živali

2.3.7.1.1 Opis funkcionalnosti

Funkcionalnost omogoča osnovno upravljanje genealogije posamezne živali v sistemu POKK.

Dodajanje, spreminjanje in brisanje genealoških povezav (oče, mati, klon, nadomestna mati) **poteka avtomatsko** na podlagi podatkov iz potrjenih dogodkov in dokumentov v **SKL-DOG: Sklop dogodki**.

Genealoške povezave se **ustvarjajo, spreminjajo ali brišejo samodejno** glede na vsebino ključnih dokumentov:

- **Rojstni list:** vsebuje podatek o materi in morebitne povezane podatke o očetu (prek

pripustnega dokumenta).

- **Pripustni dokument:** vsebuje podatek o očetu.
- **Kotitveni zapis:** vsebuje podatek o materi in mladiču.

SUPB mora zagotoviti, da ob vsakem vnosu ali spremembi teh dokumentov pride do:

- Avtomatskega kreiranja novih genealoških povezav.
- Avtomatskega spreminjanja obstoječih povezav.
- Avtomatskega brisanja povezav, če je sprememba dokumentov to zahtevala.

Vse spremembe v genealogiji so verzijsko sledene, predhodna stanja ostanejo dostopna. Uporabnik ima možnost pregledovanja genealogije živali (navigacija po prednikih in potomcih).

Glavne značilnosti:

- Samodejno ustvarjanje genealoških povezav iz dogodkov in dokumentov v SKL-DOG.
- Samodejno posodabljanje ali brisanje genealoških povezav ob spremembi dokumentov.
- Navigacija po genealogiji: prikaz prednikov in potomcev več generacij nazaj.
- Kreacija nove verzije genealoškega zapisa ob vsaki spremembi.
- Verzijsko in revizijsko beleženje vseh sprememb.
- **Implementacija s triggerji** ali primerljivim mehanizmom za samodejno izvajanje sprememb brez ročnega posega.

2.3.7.1.2 Predpogoji

- Uporabnik mora biti avtenticiran in imeti avtorizacijo za pregled genealogije živali.
- Relevantni dogodki in dokumenti v SKL-DOG morajo biti potrjeni in veljavni.
- SUPB mora zagotavljati avtomatsko vzdrževanje genealogije ob spremembah virov.
- Verzijsko in revizijsko beleženje mora biti aktivno.

2.3.7.1.3 Merila uspešnosti

- Kreacija, sprememba ali brisanje povezave izvedeno v ≤ 2 s po potrditvi dokumenta ali dogodka.
- Genealoške povezave so vedno konsistentne z vsebino dokumentov.
- Navigacija po genealogiji omogoča prikaz prednikov v ≤ 1 s.
- Predhodne verzije genealoških zapisov ostanejo dostopne.

2.3.7.1.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
14001	Samodejno ustvarjanje povezav	Kreiranje genealoških povezav na podlagi potrjenih dokumentov (rojstni list, pripustni dokument ipd.).	Potrjeni dokumenti in dogodki.	Povezava ustvarjena v ≤ 2 s po potrditvi dokumenta.
14002	Samodejno posodabljanje povezav	Spreminjanje obstoječih genealoških povezav ob spremembi dokumentov.	Aktivne povezave, veljavna sprememba dokumentov.	Povezava pravilno spremenjena v ≤ 2 s.
14003	Samodejno brisanje povezav	Odstranitev genealoške povezave ob preklicu ali spremembi dokumenta.	Potrjena sprememba dokumenta.	Povezava odstranjena, nova verzija ustvarjena.
14004	Navigacija po	Prikaz prednikov in	Aktivni genealoški	Prikaz prednikov v

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
	genealogiji	potomcev več generacij nazaj.	podatki.	≤ 1 s.
14006	Trigger-mehanizem za upravljanje genealogije	SUPB zagotovi avtomatsko izvajanje sprememb na podlagi dogodkov in dokumentov brez ročnih posegov.	Triggerji implementirani.	Spremembe izvedene avtomatsko in skladno z vsebino dokumentov.

2.3.7.1.5 Diagram primerov uporabe

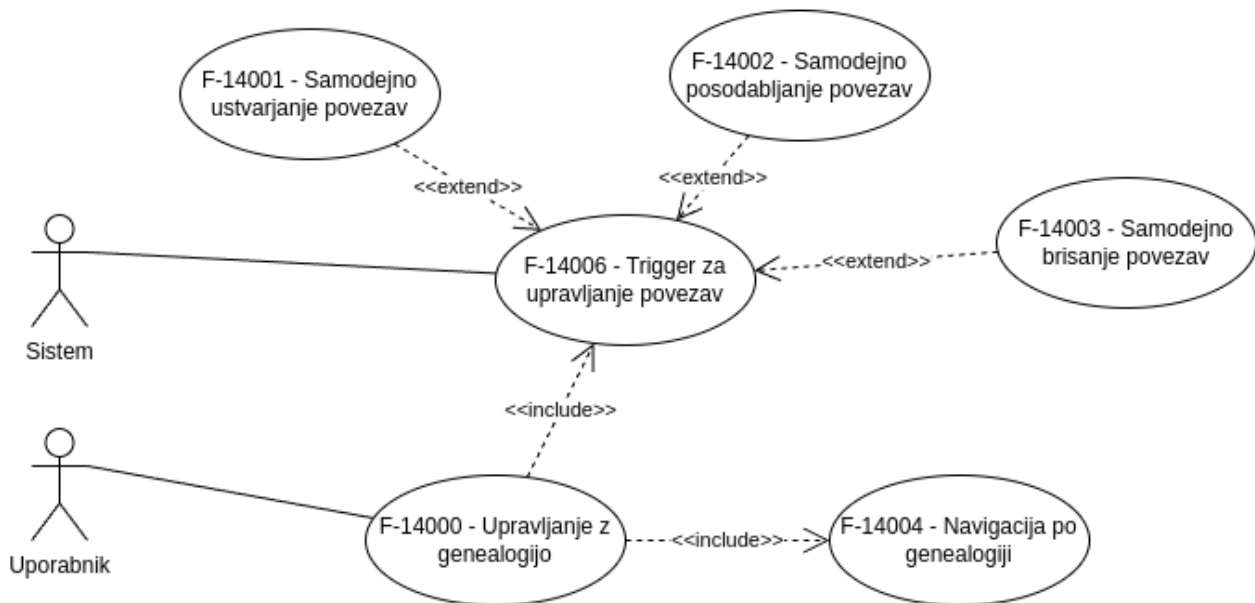


Diagram 60: Diagram primerov uporabe zs SKL-GEN-F-14000.

2.3.7.2 SKL-GEN-F-14100: Potrjevanje ali zavrnitev genealoških povezav

2.3.7.2.1 Opis funkcionalnosti

Funkcionalnost omogoča avtomatsko potrjevanje ali zavrnitev genealoških povezav za posamezno žival v sistemu POKK, na podlagi potrjenih dogodkov iz **SKL-DOG: Sklop dogodki**.

Vsak dogodek vsebuje podatke o:

- Na katero genealoško povezavo se nanaša (oče, mati, klon, nadomestna mati).
- Ali gre za **potrditev** ali **zavrnitev** povezave.

Ob potrditvi dogodka sistem samodejno sproži eno od dveh operacij:

- **Potrditev:** Povezava se označi kot dokončno potrjena. Povezava se dopolni z vrstico o tipu potrditve (genetska analiza, dokumenti ipd.) in povezavo na dokumente (prek SKL-Docs).
- **Zavrnitev:** Povezava se izbriše iz genealogije (npr. odstranitev očeta iz polja oče), ustvari se nova verzija genealoškega zapisa.

Glavne značilnosti:

- Samodejna zaznava dogodkov za potrjevanje ali zavrnitev povezav.

- Avtomatska sprememba stanja povezav brez ročnih intervencij.
- Vnos podatkov o tipu potrditve in povezava na dokumente za potrjene povezave.
- Odstranitev povezave in kreacija nove verzije zapisa pri zavrnitvi.
- Popolno vezanje sprememb na vir (dogodek + dokument).
- Verzijsko beleženje vseh sprememb.
- **Implementacija na ravni sistema mora biti izvedena s triggerji ali primerljivim mehanizmom, ki avtomatsko zagotavlja stanje.**

2.3.7.2.2 Predpogoji

- Uporabnik je avtenticiran in ima avtorizacijo za dostop do genealogije živali.
- Dogodki v SKL-DOG morajo biti potrjeni in vsebovati jasno oznako (potrditev ali zavrnitev povezave).
- Genealoške povezave morajo obstajati pred potrjevanjem ali zavrnitvijo.
- Aktivna povezava s SKL-Docs za prenos dokumentov.
- Verzijsko in revizijsko sledenje mora biti omogočeno.
- SUPB mora zagotoviti avtomatsko osveževanje genealoških podatkov ob potrditvi dogodkov (trigger mehanizem).

2.3.7.2.3 Merila uspešnosti

- Sprememba stanja povezave (potrditev ali zavrnitev) izvedena v ≤ 2 s po potrditvi dogodka.
- Pri potrditvi je tip potrditve pravilno zabeležen in dokument pravilno povezan.
- Pri zavrnitvi je povezava pravilno odstranjena, nova verzija zapisa ustvarjena.
- Vse spremembe so verzijsko in revizijsko beležene.

2.3.7.2.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
14101	Samodejna potrditev genealoške povezave	Povezava označena kot dokončno potrjena na podlagi dogodka.	Potrjen dogodek za potrditev.	Status povezave spremenjen, dokumenti povezani.
14102	Samodejna zavrnitev genealoške povezave	Povezava odstranjena iz genealogije na podlagi dogodka.	Potrjen dogodek za zavrnitev.	Povezava odstranjena, nova verzija ustvarjena.
14103	Povezava spremembe z virom dogodka	Vsaka sprememba (potrditev/zavrnitev) povezana s konkretnim dogodkom in dokumenti.	Aktivni viri povezav.	Povezava ustrezno vidna v sistemu.
14105	Implementacija trigger mehanizma	SUPB zagotovi sprožitev sprememb ob potrditvi dogodka brez ročnega posega.	Triggerji implementirani.	Samodejna posodobitev v ≤ 2 s po potrditvi dogodka.

2.3.7.2.5 Diagram primerov uporabe

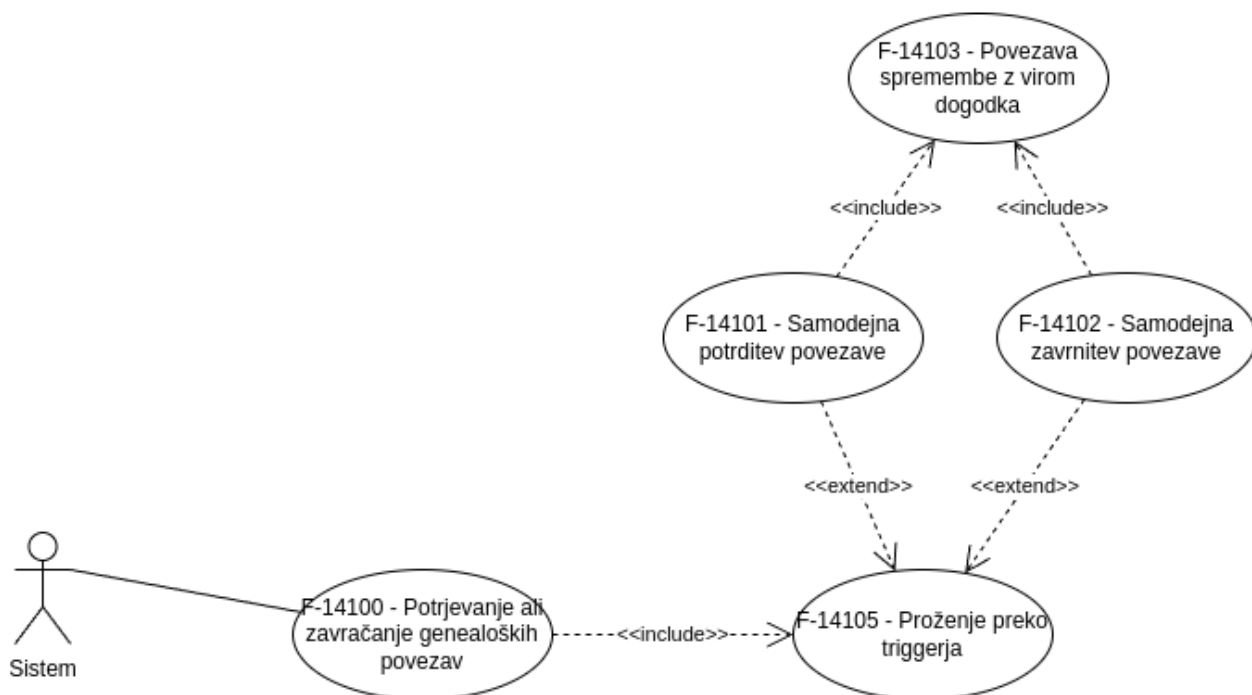


Diagram 61: Diagram primerov uporabe za SKL-GEN-F-14100.

2.3.8 SKL-Inf: Sklop splošne informacije

2.3.8.1 SKL-INF-F-15000: Upravljanje statičnih vsebin preko CMS

2.3.8.1.1 Opis funkcionalnosti

Sistem omogoča upravljanje **statičnih vsebin** z uporabo integriranega CMS (Content Management System) modula, namenjenega urejanju in prikazu vsebin, ki niso neposredno vezane na podatkovne entitete (npr. pomoč, navodila, obrazci, pogoji uporabe, obvestila, novosti, sheme postopkov).

S tem se omogoča, da lahko pooblaščen uporabniki (npr. administratorji, uredniki vsebin) neodvisno in brez razvoja:

- vnašajo in urejajo besedilne vsebine,
- prilagajajo jezikovne različice (SL, EN ipd.),
- vključujejo datoteke (PDF, DOCX, slike) ter oblikovane HTML-bloke,
- dinamično prilagajajo vidnost vsebin glede na vlogo ali tip uporabnika.

CMS je integriran znotraj sistema in uporablja sistem za pravice (RBAC) za ločevanje uredniških vlog od operativnih uporabnikov.

2.3.8.1.2 Predpogoji

- Uporabnik ima vlogo z uredniškimi pravicami.
- CMS modul mora biti ustrezno konfiguriran in povezan s sistemskim UI.
- Uporabniški vmesnik mora podpirati večjezično upravljanje vsebin.
- Predhodno definirane cone CMS morajo biti implementirane na mestih prikaza (npr. naslovi strani, pomoč, opisi postopkov).

2.3.8.1.3 Merila uspešnosti

- Dodajanje nove vsebine traja ≤ 1 min preko CMS vmesnika.
- Vsebine so dostopne končnim uporabnikom takoj po objavi.
- CMS podpira vsaj 2 jezikovni različici in možnost HTML obogatene formata.
- Sistem revizijsko beleži spremembe vsebin (čas, avtor, razlika).
- Vsebine so ločene od aplikacijske kode in se ne prevedejo skupaj s sistemom.

2.3.8.1.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~15000	Upravljanje vsebin s CMS	Uporabniki z vlogo "urednik" lahko urejajo vsebine preko CMS vmesnika.	Konfiguriran CMS, aktivna vloga	Dodajanje/urejanje v ≤ 1 min
~15001	Večjezičnost vsebin	Vsebino je mogoče vnašati in urejati ločeno po jezikih (npr. slovenščina, angleščina).	CMS mora podpirati lokalizacijo	Jezikovno preklapljanje brez napake
~15002	Prikaz statičnih strani	Uporabnik vidi vsebine, ki so povezane z delom aplikacije (npr. pomoč, pogoji).	CMS cona aktivna na strani	Vsebina prikazana dinamično

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~15003	Pripenjanje dokumentov	Možnost dodajanja priponk k vsebini (npr. PDF, DOC, JPG).	CMS podpira priponke	Dokumenti dostopni in varno servirani
~15004	Verzioriranje sprememb	Sistem vodi revizijsko zgodovino sprememb vsebin.	Vklopljen revizijski modul	Spremembe se beležijo in mogoč je vpogled »za nazaj«

2.3.8.1.5 Diagram primerov uporabe

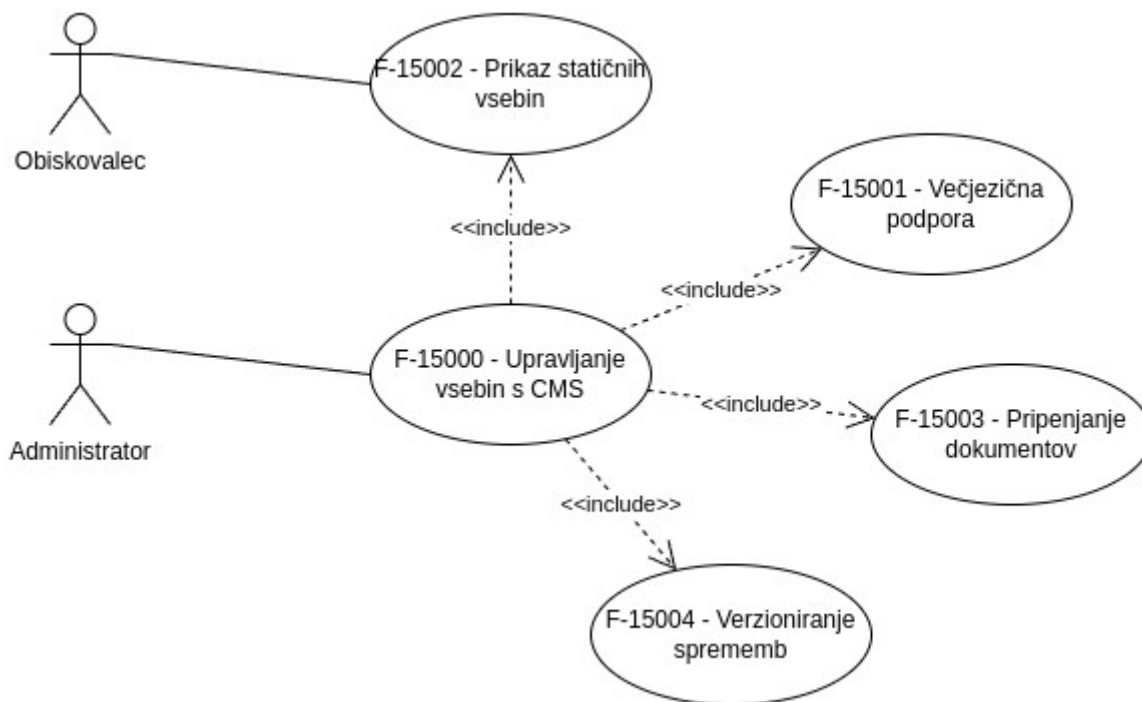


Diagram 62: Diagram primerov uporabe za SKL-INF-F-15000.

2.3.9 SKL-Dog: Sklop dogodki

2.3.9.1 SKL-DOG-F-17000: Pregled dogodkov

2.3.9.1.1 Opis funkcionalnosti

Funkcionalnost omogoča uporabniku pregled vseh dogodkov, ki so bili evidentirani nad posamezno živaljo. Dogodki se prikazujejo v kronološkem vrstnem redu, z možnostjo filtriranja po tipu dogodka, datumu, izvajalcu ter drugih atributih. Sistem prikazuje podatke v tabelarični obliki in/ali kot časovnico (timeline), z možnostjo razširjenega ogleda podrobnosti posameznega dogodka.

Dogodki vključujejo:

- razmnoževalne dogodke: osemenitev, pripust, kotitev, zamrzovanje zarodkov, presajanje ipd.,
- zdravstvene dogodke: bolezni, pregledi, cepljenja, posegi,
- statusne spremembe: izločitve, pogini, spremembe lastništva ali lokacije,
- druge pomembne dogodke, definirane preko šifranta dogodkov.

Funkcionalnost vključuje povezave do drugih entitet (npr. potomcev, izvajalcev, dokumentacije) in omogoča hiter prehod na povezane podatke.

2.3.9.1.2 Predpogoji

- Uporabnik mora biti avtenticiran in avtoriziran za dostop do podatkov zadevne živali.
- Podatkovni modul DOGODEK mora biti dostopen in operativen.
- Žival mora imeti veljaven identifikator in vpisan vsaj en dogodek.

2.3.9.1.3 Merila uspešnosti

- Seznam dogodkov za posamezno žival se naloži v ≤ 1 s.
- Filtriranje in iskanje po dogodkih se izvede v ≤ 500 ms.
- Razširjen pogled dogodka se prikaže v ≤ 500 ms.
- Vsi dogodki so popolno revizijsko sledeni z uporabnikom, časom in spremembo.
- Sistemski odziv mora omogočati delovanje tudi pri ≥ 10.000 dogodkih na žival.

2.3.9.1.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~17000	Pregled dogodkov	Prikaz vseh dogodkov evidentiranih nad živaljo v kronološkem vrstnem redu.	Avtentikacija, obstoj živali	Prikaz v ≤ 1 s
~17001	Filtriranje dogodkov	Iskanje po tipu dogodka, datumu, izvajalcu, lokaciji in vsebini.	Dogodki obstajajo	Dinamično filtriranje v ≤ 500 ms
~17002	Prikaz podrobnosti dogodka	Razširjen pogled izbranega dogodka z vsemi atributi in povezanimi entitetami.	Izbran dogodek, podatki dostopni	Prikaz podrobnosti v ≤ 500 ms
~17003	Časovnica dogodkov	Grafični prikaz dogodkov na časovni premici z interaktivnim pregledom.	Časovni podatki so vpisani pri dogodkih	Brezhibno delovanje pri 100+ dogodkih
~17004	Izvoz dogodkov	Izvoz filtriranega seznama dogodkov v CSV ali PDF format.	Uporabnik ima pravico za izvoz	Datoteka generirana v ≤ 5 s
~17005	Povezava z drugimi moduli	Možnost prehoda iz dogodka v povezane funkcionalnosti (npr. osemenitev, laboratorij).	Relevantni ID-ji so na voljo	Preusmeritev deluje brez napak

2.3.9.1.5 Diagram primerov uporabe

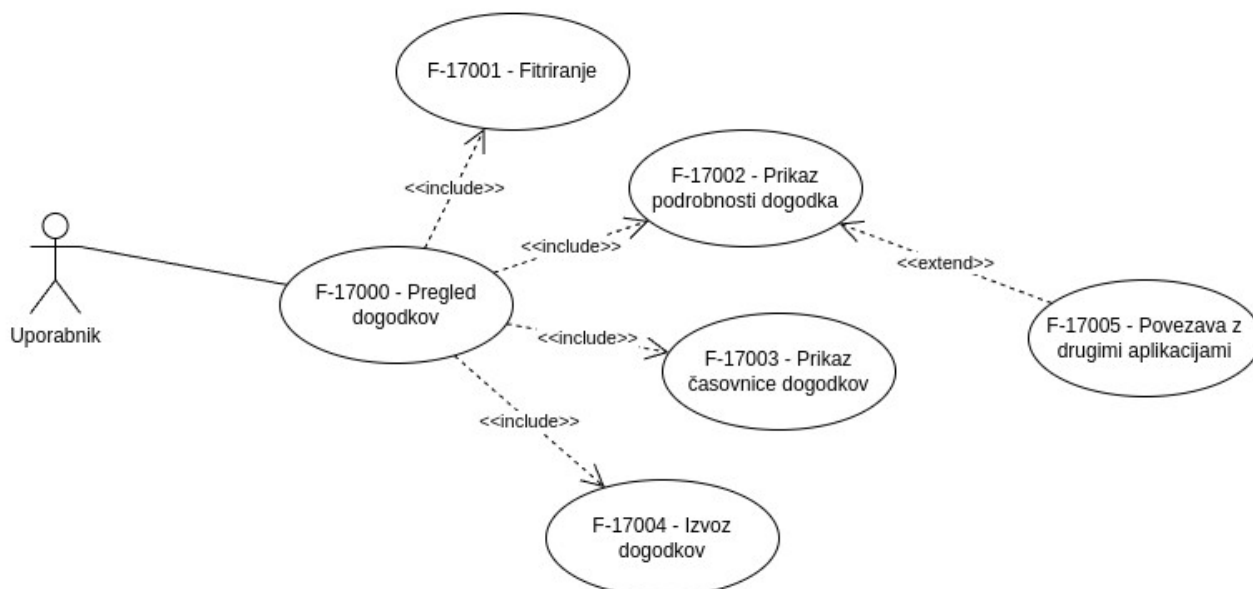


Diagram 63: Diagram primerov uporabe za SKL-DOG-F-17000.

2.3.9.2 SKL-DOG-F-17100: Vnos splošnega datumskega dogodka

2.3.9.2.1 Opis funkcionalnosti

Funkcionalnost omogoča vnos splošnega dogodka, ki se je zgodil na točno določen dan in ni vezan na kompleksno logiko. Uporabnik izbere tip dogodka iz šifranta (npr. rojstvo, smrt, zakol), vnese datum dogodka ter po potrebi dopiše opombo in doda priponke (dokazila, dokumentacija, poročila ipd.).

Če je izbran tip dogodka, ki ima namensko implementacijo (npr. tetoviranje, odbira, osemenitev), sistem uporabnika samodejno preusmeri na ustrezno masko za ta specifični dogodek, kjer se nadaljuje vnos z dodatnimi zahtevami. Tak prehod je izveden dinamično glede na konfiguracijo šifranta tipov dogodkov.

2.3.9.2.2 Predpogoji

- Uporabnik mora biti avtoriziran za vnos dogodkov nad izbrano živaljo.
- Mora biti izbran veljaven tip dogodka iz šifranta.
- Datum dogodka mora biti veljaven in ne sme biti v prihodnosti (razen če je dogodek označen kot planiran).

2.3.9.2.3 Merila uspešnosti

- Vnos dogodka se zaključi v ≤ 1 s po potrditvi.
- Sistem samodejno prepozna potrebo po preusmeritvi na podrobno masko (če velja).
- Vsi vnosi so revizijsko zabeleženi, vključno z vsemi atributi dogodka.
- Priloge se uspešno shranijo in vežejo na ID dogodka.

2.3.9.2.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~17100	Vnos splošnega	Vnos preprostega dogodka z	Avtorizacija,	Shrani se v ≤ 1 s

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
	datumskega dogodka	izbiro tipa, datumom, opombo in priponkami.	veljaven tip in datum	
~17101	Pripenjanje priponk	Uporabnik doda dokumentacijo, povezano z dogodkom.	Dogodek mora biti v fazi vnosa ali urejanja	Priponke se uspešno shranijo in povežejo
~17102	Preusmeritev na namenski dogodek	Ob izboru tipa dogodka s kompleksno obravnavo se uporabnik preusmeri na namensko masko.	Tip dogodka konfiguriran za namensko obravnavo	Preusmeritev izvede brez zamika

2.3.9.2.5 Diagram primerov uporabe

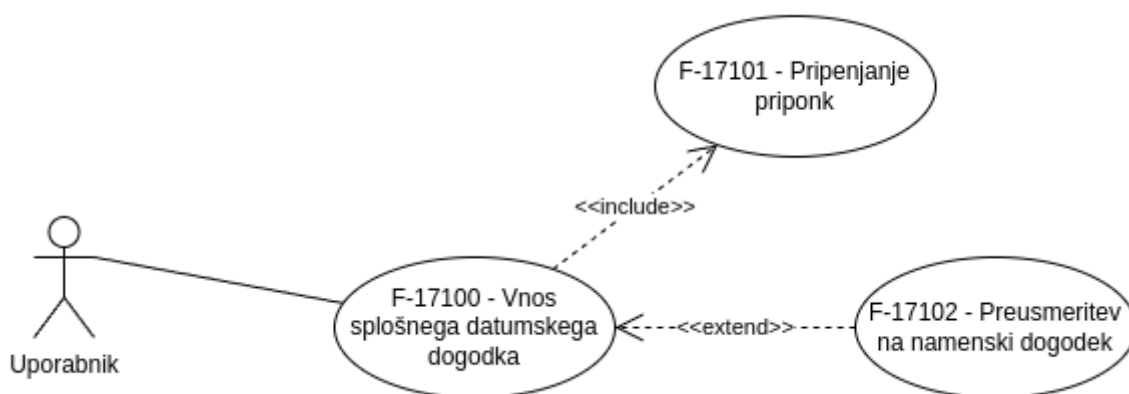


Diagram 64: Diagram primerov uporabe za SKL-DOG-F-17100.

2.3.9.3 SKL-DOG-F-17600: Seme

2.3.9.3.1 Opis funkcionalnosti

Funkcionalnost omogoča zajem podrobnih podatkov o odvzetem semenu. Vnosna maska je namenjena predvsem ovnom, pri katerih se beležijo lastnosti semena, podatki o odvzemu ter morfološke in giblјivostne karakteristike. Uporabnik vnese datum odvzema, identifikacijo ovna, ocenjevalca in zaporedno številko odvzema ter različne fizikalne in morfološke parametre.

Vnos vključuje naslednje vsebinske sklope:

- **Osnovni podatki:** datum, ocenjevalec, številka ovna, zaporedna št., volumen, obseg mod.
- **Giblјivost:** deleži giblјivih spermijev (%PG, %MG, %NG), gostota.
- **Morfometrija:** KSP, DSP, NR, RS.
- **Kategorizacija celic:** izbor iz šifranta in možnost ponovnega odvzema.
- **Splošni opis:** morfološke ocene (npr. glava, rep, PD-prox, PD-mid, PD-dist, multi).
- **Dokumentacija:** možnost pripenjanja dokumentov in priprave potnih listov ali poročil.

2.3.9.3.2 Predpogoji

- Uporabnik mora imeti avtorizacijo za delo s podatki o semenu in izbrano živaljo.
- Žival mora biti samec z ustrezno evidenco v sistemu (npr. oven z ID).
- Šifrant celic, morfoloških ocen in merskih enot mora biti aktiven in usklajen.

2.3.9.3.3 Merila uspešnosti

- Obrazec se naloži v ≤ 1 s.
- Vsi podatki se validirajo pred shranjevanjem; napačni vnosi se jasno označijo.
- Shramba podatkov se izvede v ≤ 1 s.
- Pripeti dokumenti so povezani s točno določenim odvzemom.
- Podatki so revizijsko sledeni z datumom, uporabnikom in rezultatom.

2.3.9.3.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~17600	Vnos podatkov o semenu	Omogoča zajem vseh potrebnih podatkov ob odvzemu semena za izbranega ovna.	Avtorizacija, veljavna žival, aktivni šifranti	Vnos in shranjevanje v ≤ 1 s
~17601	Giblјivost in gostota	Zajem parametrov giblјivosti (%PG, %MG, %NG) in gostote spermijev.	Vpisana meritev, veljavne vrednosti	Validacija uspešna, vrednosti shranjene
~17602	Morfološke lastnosti	Vnos podatkov o repu, glavi, deformacijah in drugih atributih morfologije.	Šifranti morfologije doseglјivi	Polja validirana, prikaz v UI
~17603	Dokumentacija in izpisi	Omogoča pripenjanje dnevnikov, pripravo izpisov in generiranje potnega lista.	Pravilno zaključen vnos odvzema	Dokument pripravljen v ≤ 3 s

2.3.9.3.5 Diagram primerov uporabe

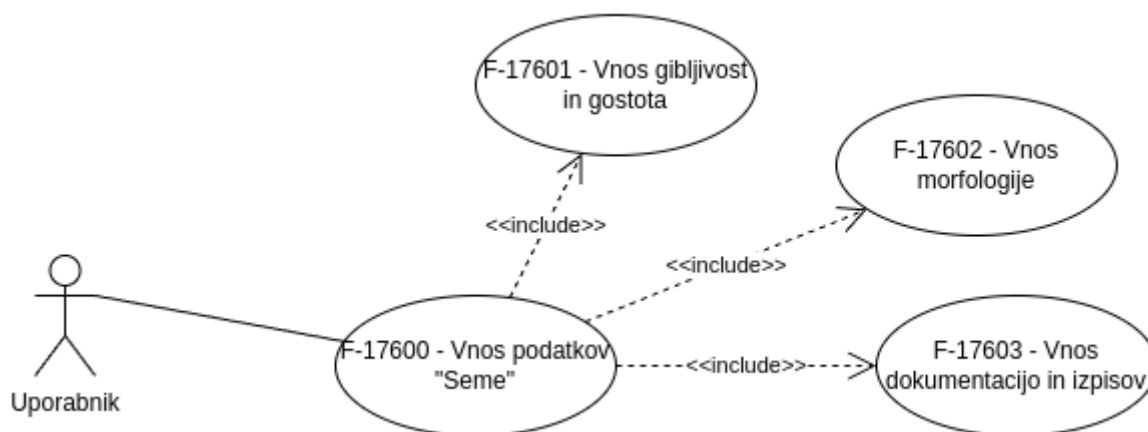


Diagram 65: Diagram primerov uporabe za SKL-DOG-F-17600.

2.3.9.4 SKL-DOG-F-17700: Pripust

2.3.9.4.1 Opis funkcionalnosti

Funkcionalnost omogoča zajem dogodka pripusta, vključno z vsemi potrebnimi informacijami o posameznem ali haremskem pripustu. Uporabniški vmesnik omogoča ločeno evidentiranje **individualnega pripusta** (z enim partnerjem) ali **haremskega pripusta** (z več partnerji), pri čemer so vključeni podatki o:

- obdobju pripusta (od–do),
- ID-ju partnerja ali partnerk,

- lokaciji pripusta,
- osemenjevalcu,
- obnašanju živali v času pripusta,
- načinu in aplikaciji semena (če je relevantno),
- pripadajoči dokumentaciji in opombah.

Uporabnik ima možnost dodajanja in odstranitve več partnerjev ter shranjevanja dogodka s pripetimi dokumenti (npr. dnevnik pripustov). Maski za individualni in haremski pripust se nahajata znotraj enotne vnosne maske.

2.3.9.4.2 Predpogoji

- Uporabnik mora imeti avtorizacijo za vnos dogodkov nad izbranimi živalmi.
- Vse živali vključene v pripust morajo biti registrirane in aktivne.
- Šifranti (načini aplikacije, obnašanje, seme ipd.) morajo biti konfigurirani in dostopni.
- Lokacija pripusta mora biti predhodno definirana v sistemu.

2.3.9.4.3 Merila uspešnosti

- Vnos posameznega pripusta (individualnega ali haremskega) se zaključi v ≤ 1 s.
- Vsak partner (pri haremskem pripustu) je validiran in shranjen v ločenem zapisu.
- Sistem omogoča natančno beleženje vseh parametrov brez izgube podatkov.
- Pripeti dokumenti se uspešno povežejo z ID dogodka.
- Vsi vnosi so revizijsko sledeni.

2.3.9.4.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~17700	Vnos pripusta	Zajem dogodka pripusta z vsemi podatki o partnerstvu, lokaciji, aplikaciji semena in spremljavi.	Avtorizacija, registrirane živali, aktivni šifranti	Shranjevanje v ≤ 1 s, validacija podatkov
~17701	Individualni pripust	Evidentiranje pripusta z enim partnerjem in spremljajočimi atributi.	Izbrana žival, datum, ID partnerja	Vnos zaključen brez napak, status "potrjeno"
~17702	Haremski pripust	Omogoča vnos pripusta z več partnerji in vnosom individualnih nastavitev za vsako partnerko.	Seznam partnerjev, ustrezne pravice	Uporabniški vmesnik stabilen, podatki ločeni
~17703	Vnos atributov (obnašanje, seme)	Vnos posebnih atributov pripusta: način aplikacije, obnašanje živali ipd.	Šifranti dostopni, vnosni pogoji ustrezni	Vrednosti izbrane, pravilno shranjene
~17704	Dokumentacija in izpis	Omogoča pripenjanje dnevnikov, pripravo poročil in potnih listov.	Zaključen vnos dogodka	Dokument generiran v ≤ 3 s, priponka povezana

2.3.9.4.5 Diagram primerov uporabe

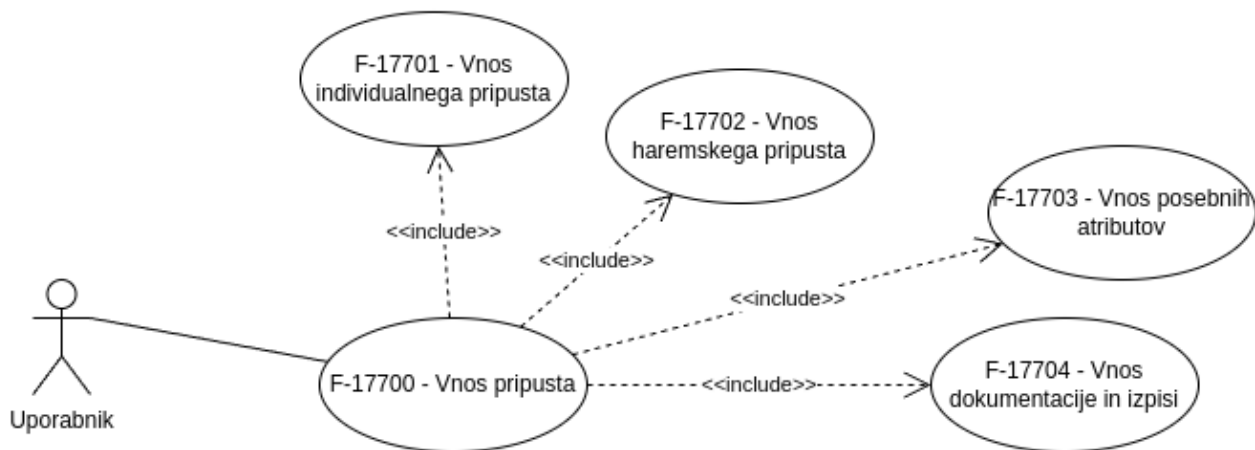


Diagram 66: Diagram primerov uporabe za SKL-DOG-F-17700.

2.3.9.5 SKL-DOG-F-17800: Tretiranje

2.3.9.5.1 Opis funkcionalnosti

Funkcionalnost omogoča vnos dogodka **tretiranja živali**, tj. uporabe veterinarsko odobrenih zdravil ali drugih sredstev za preprečevanje ali zdravljenje bolezni. Gre za pomemben del veterinarskega nadzora, ki je ključen zlasti za zagotavljanje sledljivosti v rejah, kjer je predpisana karenca.

Ob vnosu se zajemajo naslednji podatki:

- datum tretiranja,
- sredstvo za tretiranje (ime pripravka ali aktivne učinkovine),
- karenca (v dnevih), tj. čas, v katerem izdelki živali (mleko, meso) niso primerni za potrošnjo.

Vnos je lahko izveden za posamezno žival ali skupino živali (npr. čreda, jata), glede na predhodno izbrani kontekst.

2.3.9.5.2 Predpogoji

- Žival mora biti v aktivnem statusu.
- Uporabnik mora biti avtoriziran za vnos veterinarskih posegov.
- Ime sredstva in datum tretiranja morata biti obvezno izpolnjena.
- Karenca mora biti pozitivno celo število ali prazno, če ni predpisana.

2.3.9.5.3 Merila uspešnosti

- Vnosna maska se naloži v ≤ 1 s.
- Obvezna polja se validirajo ob potrditvi vnosa.
- Shranjevanje podatkov se izvede v ≤ 1 s.
- Vsak dogodek je revizijsko zabeležen z uporabnikom in časom vnosa.
- Karenca se upošteva v vseh funkcijah povezanih s pridelavo in prometom proizvodov.

2.3.9.5.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~17800	Vnos dogodka tretiranja	Zajem podatkov o uporabi veterinarskega sredstva pri posamezni živali ali skupini.	Avtorizacija, aktivna žival	Vnos uspešen v ≤ 1 s
~17801	Evidenca karence	Zajem števila dni karence po tretiranju.	Vnos številke ali prazen vnos	Vrednost validirana, shranjena
~17802	Obvezna dokumentacija	Vpis sredstva in datuma tretiranja kot obvezna elementa.	Polji ne smeta biti prazni	Validacija ob potrditvi
~17803	Priprava izpisa	Generiranje dokumenta (npr. potrdilo o tretiranju, izpis za rejca ali inšpekcijo).	Dogodek uspešno zaključen	Dokument pripravljen in ustrezno indeksiran

2.3.9.5.5 Diagram primerov uporabe

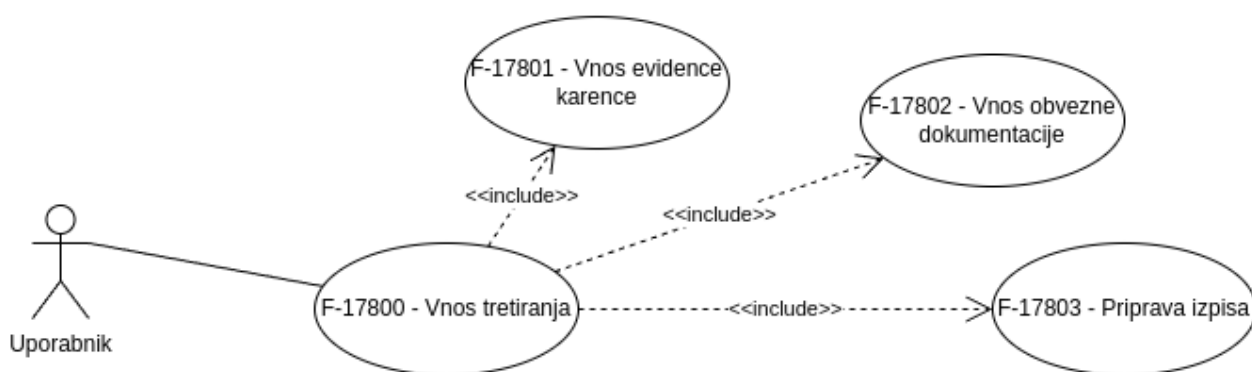


Diagram 67: Diagram primerov uporabe za SKL-DOG-F-17800.

2.3.9.6 SKL-DOG-F-17900: Izločitev

2.3.9.6.1 Opis funkcionalnosti

Funkcionalnost omogoča evidentiranje dogodka izločitve živali iz reje. Uporabnik vpiše datum izločitve, izbere ustrezen razlog izločitve iz šifranta in lahko doda dodatna pojasnila v obliki opombe ter pripne dokumentacijo (npr. zakolno poročilo, interni zapisnik, izvid). Vnosna maska je enostavna in omogoča hiter zajem osnovnih informacij.

Vneseni podatki so ključni za zaključek življenjskega cikla posamezne živali in posledično vplivajo na druge poslovne procese (npr. onemogočanje pripustov, prestavitev v status arhivirano ipd.).

2.3.9.6.2 Predpogoji

- Uporabnik mora biti avtoriziran za delo z izbrano živaljo.
- Žival mora biti v aktivnem stanju (ni že izločena).
- Šifrant razlogov za izločitev mora biti na voljo.
- Datum izločitve mora biti enak ali po datumu rojstva živali.

2.3.9.6.3 Merila uspešnosti

- Maska se naloži v ≤ 500 ms.
- Vnos dogodka se izvede v ≤ 1 s.

- Sistem onemogoči dvojno izločitev iste živali.
- Vsi podatki se revizijsko beležijo, vključno z uporabnikom, časom in razlogom.
- Pripeti dokumenti se uspešno povežejo z dogodkom.

2.3.9.6.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~17900	Vnos dogodka izločitve	Omogoča vnos datuma in razloga izločitve ter dodatnih opomb in priponk.	Avtorizacija, aktivna žival, dosegljiv šifrant	Shranjevanje v ≤ 1 s
~17901	Validacija pogojev	Preverjanje pravilnosti datuma, enkratnosti vnosa ter ujemanja statusa živali.	Vsi podatki morajo biti veljavni in časovno usklajeni	Napake so jasno označene
~17902	Pripenjanje dokumentacije	Omogoča dodajanje poročil, obrazcev, odločb in drugih spremljajočih dokumentov.	Vnos v fazi shranjevanja	Dokument uspešno pripet

2.3.9.6.5 Diagram primerov uporabe

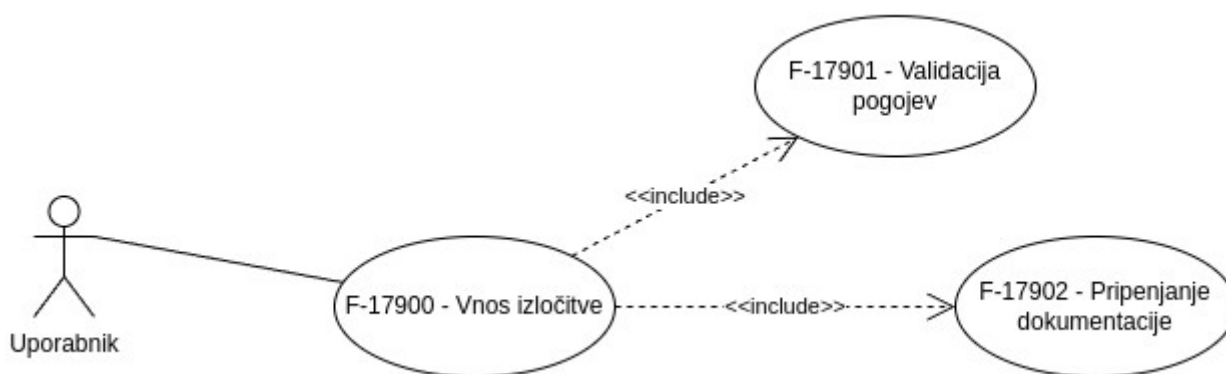


Diagram 68: Diagram primerov uporabe za SKL-DOG-F-17900.

2.3.10 SKL-Drob: Sklop drobnica

2.3.10.1 SKL-DOG-F-17300: Meritve drobnica

2.3.10.1.1 Opis funkcionalnosti

Funkcionalnost omogoča **celovit zajem merilnih podatkov pri drobnici**, ki so potrebni za genetsko vrednotenje, rejski nadzor in poročanje. Obseg podatkov pokriva parametre kakovosti volne, telesne in mišične lastnosti ter genetske označevalce.

Uporabnik izbere dogodek, vnašalca in datum meritve, nato pa vnese naslednje sklope podatkov:

- **Volnarski parametri** (npr. mic_avg, mic_sd, mic_cv, cem, cf, sf, sl, pod15, deb_min, deb_max, deb_sd, crv, crv_sd),
- **Telesne in mišične mere** (telesna masa, masa volne, hrbtna mišica),
- **Genetski podatki** (TSE alel1 in alel2),
- **Rodovni podatki** (ID matere in očeta, analiza, dokumenti),

- **Datumi posameznih meritev**, če so različni.

Vključen je tudi prostor za **opombe, pripenjanje dokumentacije** (PDF, laboratorijski izvidi, slike) ter možnost generiranja **izpisa** (npr. potni list, rejski list, poročilo).

2.3.10.1.2 Predpogoji

- Žival mora biti evidentirana kot drobnica (ovca/oven).
- Vsi šifranti (dogodki, aleli, vnašalci) morajo biti sinhronizirani in aktivni.
- Vsaka vrednost mora biti validirana po tipu (številčno polje, datumsko polje).
- V primeru vnosa genetskih ali rodovniških podatkov mora biti znana povezava z živalskimi evidencami.

2.3.10.1.3 Merila uspešnosti

- Obrazec se naloži v ≤ 1 s.
- Shranjevanje celotnega vnosa (vključno z dokumenti) se izvede v ≤ 1 s.
- Vse spremembe so revizijsko sledene.
- Vnos podpira **deljene datume** meritve (posamezni atributi lahko imajo različne datume).
- Sistem avtomatsko preverja povezave do staršev, če so ID-ji vnešeni.

2.3.10.1.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~17300	Vnos meritev pri drobnici	Zajem telesnih, volnarskih in genetskih podatkov ter povezava s starši in dokumenti.	Avtorizacija, žival drobnica	Shranjevanje v ≤ 1 s
~17301	Parametri kakovosti volne	Vnos podatkov: mic_avg, cem, cf, sf, debelina, crv itd.	Veljavne številčne vrednosti	Validacija brez napak
~17302	Telesne in mišične mere	Vnos mase, hrbtna mišice, mase volne	Vrednosti v sprejemljivih mejah	Shranjeno z datumom
~17303	Genetski podatki (TSE aleli)	Vnos alela 1 in 2 s pripadajočim datumom meritve	Vnos obvezen za določen tip dogodka	Preverjanje šifrantov
~17304	Prikaz podatkov staršev	Možnost vpogleda v podatke staršev (redirect).	ID morajo obstajati ali biti vpisani	Povezano z živaljo, shranjeno
~17305	Pripenjanje dokumentov	Povezava laboratorijskih in drugih dokazil z meritvijo	Vnos v fazi zaključka	Dokument indeksiran in revizijsko sleden

2.3.10.1.5 Diagram primerov uporabe

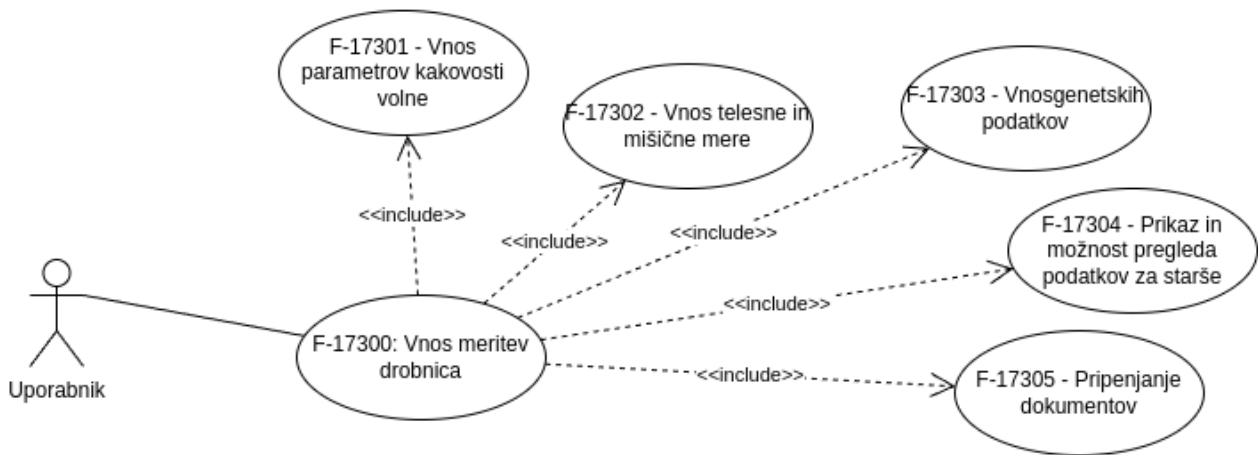


Diagram 69: Diagram primerov uporabe za SKL-DOG-F-17300.

2.3.10.2 SKL-DOG-F-18000: Izločitev TP Logatec

2.3.10.2.1 Opis funkcionalnosti

Funkcionalnost omogoča vnos dogodka izločitve živali iz reje s prenosom na **Transportno-prodajno (TP) lokacijo Logatec**, pri čemer je obravnava razširjena glede na klasično izločitev. V maski se evidentirajo dodatni podatki, povezani s selitvijo živali, spremno dokumentacijo, ocenjevanjem ter logistiko premika. Funkcionalnost je namenjena predvsem **drobnici**, kjer so TP postopki del standardiziranega upravljanja rejskih presežkov.

Zajeti podatki vključujejo:

- osnovne attribute izločitve (datum, vzrok, izvajalec),
- podatke o premiku: datum premika, nova in stara lokacija, KMG-MID, registrska številka vozila,
- prodajno in komisijsko oceno,
- spremljajoče dokumente: spremni list, izhodna dokumentacija.

2.3.10.2.2 Predpogoji

- Žival mora biti označena kot drobnica.
- Uporabnik mora biti avtoriziran za izvoz in izločanje rejskih živali.
- Šifranti za vzroke, lokacije in ocene morajo biti dosegljivi.
- Vsi atributi selitve morajo biti vneseni, če je TP aktiviran.

2.3.10.2.3 Merila uspešnosti

- Obrazec se naloži v ≤ 1 s.
- Vsi zahtevani atributi (datum, lokacija, ocena, spremni list) se validirajo pred potrditvijo.
- Shranjevanje se izvede v ≤ 1 s, vnos se evidentira z vsemi metapodatki.

- Dokumentacija se uspešno pripne in indeksira z ID izločitve.
- Podatki so revizijsko sledeni in pripravljeni za poročila in izvoze.

2.3.10.2.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~18000	Izločitev TP Logatec	Evidentiranje izločitve živali s selitvijo na TP Logatec in zajemom razširjenih podatkov.	Žival je drobnica, uporabnik ima ustrezne pravice	Vnos zaključen v ≤ 1 s
~18001	Podatki o selitvi	Zajem nove in stare lokacije, datuma premika, registrske številke, KMG-MID.	Vsi atributi obvezni	Validirani, napake so jasno prikazane
~18002	Ocena živali	Evidentiranje prodajne in komisijske ocene.	Aktivni šifranti in vloga ocenjevalca	Ocena shranjena z metapodatki
~18003	Pripenjanje spremnih dokumentov	Vnos številke spremnega lista in pripenjanje transportne dokumentacije.	Vnos v fazi shranjevanja	Dokumenti povezani z izločitvijo

2.3.10.2.5 Diagram primerov uporabe

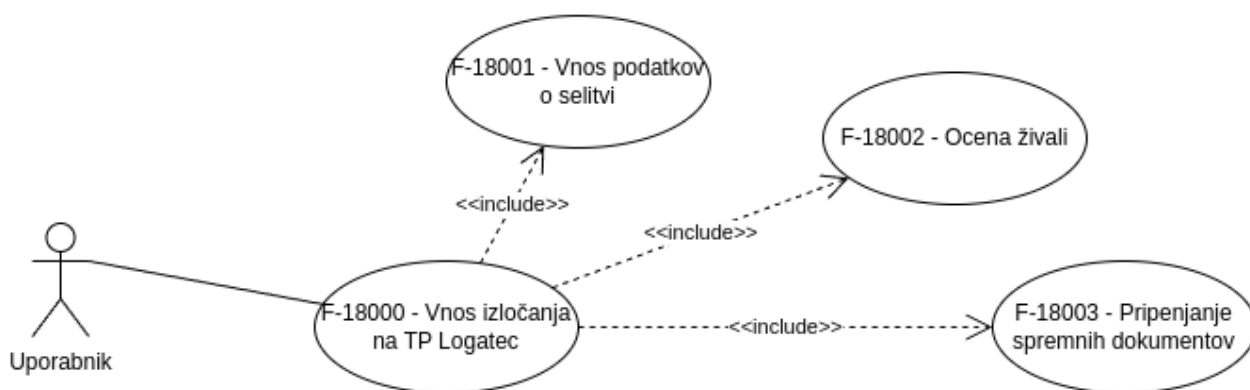


Diagram 70: Diagram primerov uporabe za SKL-DOG-F-18000.

2.3.10.3 SKL-DOG-F-18200: Jagnjitev

2.3.10.3.1 Opis funkcionalnosti

Funkcionalnost omogoča vnos dogodka **jagnjitve** pri drobnici, natančneje ovac. Uporabnik vnese datum jagnjitve, izvajalca, število rojenih in živorojenih jagnjet ter potek poroda (npr. normalen, otežen, s pomočjo). Vnos omogoča tudi dodajanje opombe in pripenjanje dokumentacije (npr. veterinarsko poročilo, zapisnik o jagnjitvi).

Dogodek jagnjitve je ključni mejnik v reprodukcijskem ciklu živali in vpliva na nadaljnje procese, kot so registracija potomcev, prirast, kasnejša odbira.

2.3.10.3.2 Predpogoji

- Žival mora biti registrirana ovca v aktivnem statusu.
- Uporabnik mora imeti pravico za vnos dogodkov v okviru danega rejca.
- Šifranti za izvajalca in potek poroda morajo biti aktivni in usklajeni.
- Datum jagnjitve ne sme biti v prihodnosti.

2.3.10.3.3 Merila uspešnosti

- Maska se naloži v ≤ 500 ms.
- Validacija vseh številskih vnosov in šifrantnih vrednosti se izvede pred potrditvijo.
- Vnos se shrani v ≤ 1 s in postane takoj viden v zgodovini dogodkov.
- Priponka se uspešno poveže z dogodkom.
- Vse spremembe so revizijsko sledene.

2.3.10.3.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~18200	Vnos dogodka jagnjitve	Zajem osnovnih podatkov o jagnjitvi: datum, izvajalec, št. rojenih in živih, porod.	Ovca z veljavnim ID, aktivni šifranti	Shranjevanje v ≤ 1 s
~18201	Evidenca poteka poroda	Uporabnik izbere potek poroda iz šifranta.	Vrednosti v šifrantu	Validacija vrednosti pred shranjevanjem
~18202	Pripenjanje dokumentacije	Možnost nalaganja pripadajočih dokumentov (zapisniki, listi).	Dogodek v fazi vnosa ali potrditve	Priponka vezana na dogodek

2.3.10.3.5 Diagram primerov uporabe

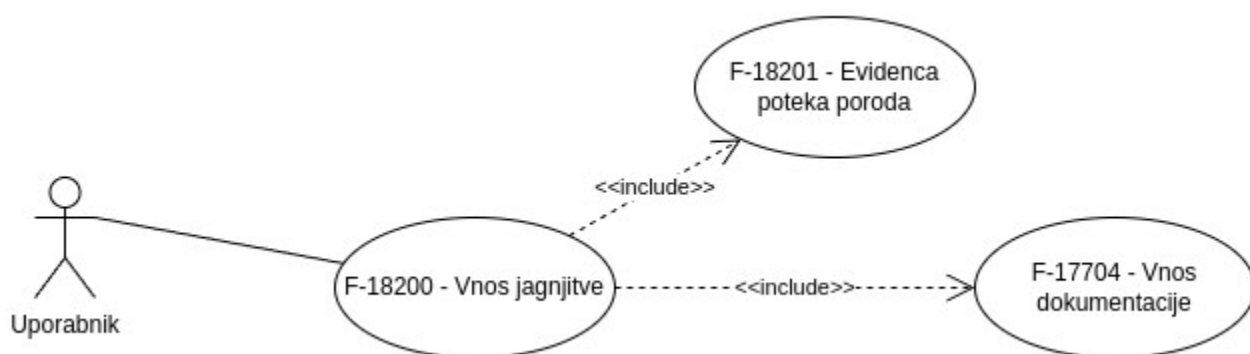


Diagram 71: Diagram primerov uporabe za SKL-DOG-F-18200.

2.3.10.4 SKL-DOG-F-18300: Označitev drobnice

2.3.10.4.1 Opis funkcionalnosti

Funkcionalnost omogoča vnos dogodka **označitve mlade drobnice**, pri čemer se zajamejo podatki o posameznih živalih, ki še niso bile individualno označene. V maski se evidentirajo naslednji podatki za vsakega posameznega mladiča:

- identifikacijska številka,
- oznaka (ušesna številka ali druga identifikacija),
- spol, barva, pasma,
- zaporedna številka v leglu (T. roj),
- podatki o usodi (ostane, izločen ipd.),

- ocena “dobra mati” in rogatost.

Poleg osnovnih atributov uporabnik vnese datum označitve, izvajalca, morebitno opombo ter lahko pripne dokumentacijo (npr. evidenčni list, potrdila o označitvi). Izračunana pasma mladičev je na voljo kot referenca na podlagi starševskih podatkov.

2.3.10.4.2 Predpogoji

- Mlada žival mora biti predhodno registrirana brez identifikacijske oznake.
- Uporabnik mora imeti avtorizacijo za označevanje živali.
- Šifranti (oznake, pasme, barve, spol, usoda) morajo biti dosegljivi.
- Datum označitve mora biti enak ali kasnejši od datuma rojstva.

2.3.10.4.3 Merila uspešnosti

- Obrazec se naloži v ≤ 1 s.
- Vnos do 10 živali hkrati mora biti omogočen brez napake.
- Vsaka nova oznaka mora biti unikatna v okviru evidence.
- Vsi vnosi se shranijo v ≤ 1 s in revizijsko beležijo.
- Pripeti dokumenti se uspešno povežejo z dogodkom.

2.3.10.4.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~18300	Označitev drobnice	Vnos podatkov o prvi individualni označitvi mlade živali.	Mladiči brez identifikacije, avtorizacija	Vnos do 10 enot v ≤ 1 s
~18301	Vnos osnovnih atributov	Vnos: spol, pasma, barva, T. roj, dobra mati, rogatost, usoda.	Šifranti konfigurirani	Polja validirana, napake označene
~18302	Izračun pasme	Samodejni izračun predlagane pasme na podlagi staršev.	Pasma staršev znana	Izračun prikazan uporabniku
~18303	Pripenjanje dokumentacije	Možnost pripenjanja dokumentov (npr. potrdila o označitvi, evidenčni listi).	Dogodek v fazi vnosa	Dokument pripet, indeksiran z ID dogodka

2.3.10.4.5 Diagram primerov uporabe

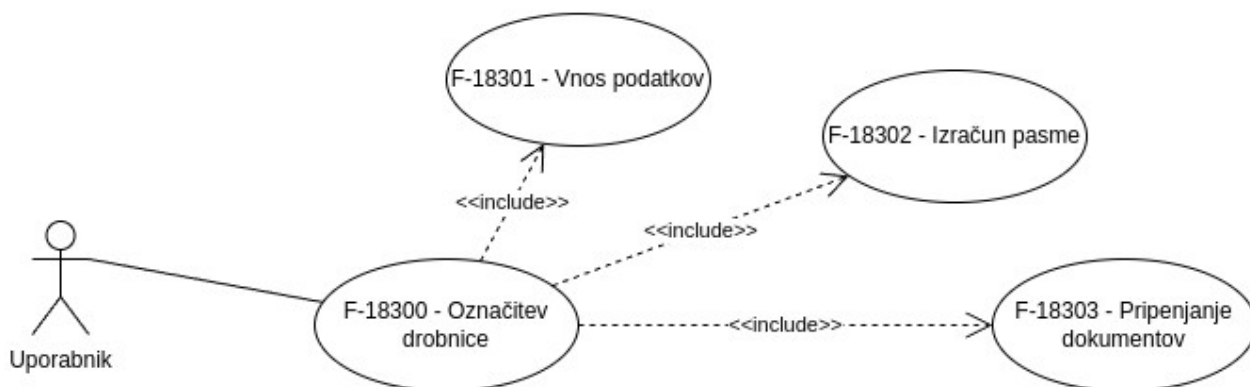


Diagram 72: Diagram primerov uporabe za SKL-DOG-F-18300.

2.3.10.5 SKL-DOG-F-18400: Odstavitev drobnice

2.3.10.5.1 Opis funkcionalnosti

Funkcionalnost omogoča evidentiranje dogodka **odstavitve jagnjet ali jarčkov**. Podatki se zajamejo za vsako posamezno žival v leglu, s poudarkom na **datumu odstavitve** in **teži** ob odstavitvi. Funkcionalnost je ključna za nadaljnje spremljanje rasti in selekcijske presoje pri drobnici.

Maska omogoča:

- skupinski ali individualni vnos datuma odstavitve,
- vnos identifikacijske številke živali (ID),
- določitev spola živali (m/ž),
- vnos mase ob odstavitvi (v kg),
- beleženje opomb (npr. zdravstveno stanje, posebnosti),
- pripenjanje dokumentov (npr. evidenčni listi, rejski pregledi),
- generiranje dokumentov (potni list, skupinsko poročilo).

2.3.10.5.2 Predpogoji

- Jagnje/jarec mora biti predhodno registriran z veljavnim ID.
- Uporabnik mora imeti pravice za vnos dogodkov za izbranega rejca.
- Datum odstavitve mora biti \geq datumu rojstva in \leq današnji datum.
- Masa mora biti številčna in > 0 .

2.3.10.5.3 Merila uspešnosti

- Vnos podatkov za do 10 živali hkrati mora biti možen brez napake.
- Validacija vnosov se izvede ob potrditvi (masa, datumi, ID).
- Shranjevanje se izvede v ≤ 1 s.
- Priponke so povezane z dogodkom in so revizijsko zabeležene.
- Vsi vnosi so takoj na voljo za nadaljnje postopke (npr. odbira, selekcija).

2.3.10.5.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~18400	Vnos odstavitve	Zajem datuma in mase ob odstavitvi za jagnjeta/jarce.	Veljavni ID, avtorizacija	Shranjevanje v ≤ 1 s
~18401	Teža ob odstavitvi	Vnos telesne mase v trenutku odstavitve.	Masa > 0	Validacija uspešna
~18402	Skupinski vnos	Možnost vnosa enakega datuma odstavitve za vse jagnje v leglu.	Jagnjeta iz istega legla	Časovno učinkovit, konsistenten vnos
~18403	Pripenjanje dokumentacije	Dodajanje priponk (PDF, Excel, slike).	Dokumenti do 10 MB	Povezano z dogodkom, varno

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~18404	Izpis dogodka	Generiranje potnih listov ali izpisa o odstavitvi.	Dogodek zaključen	hranjeno Dokument v ≤ 3 s, podatki usklajeni

2.3.10.5.5 Diagram primerov uporabe

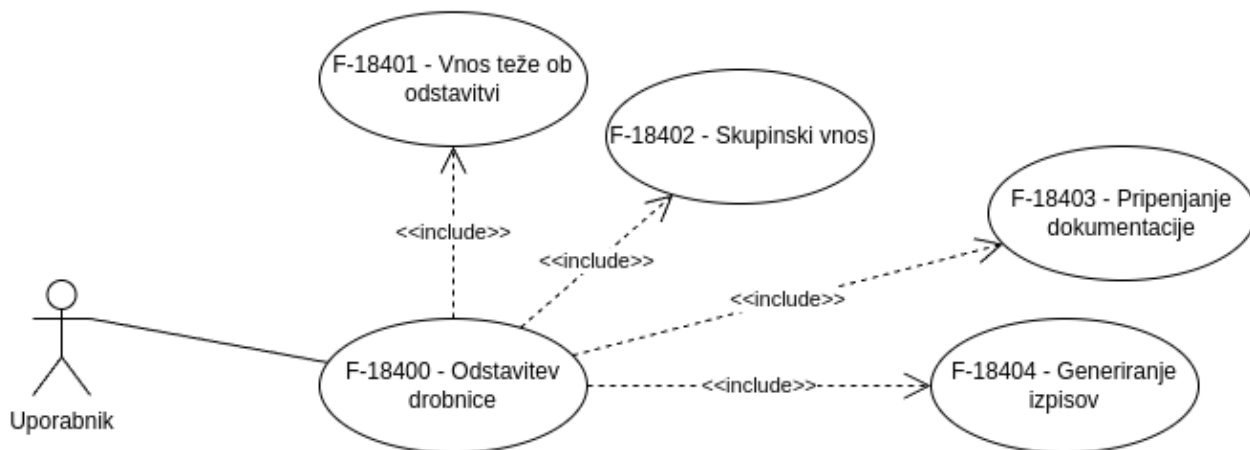


Diagram 73: Diagram primerov uporabe za SKL-DOG-F-18400.

2.3.10.6 SKL-DOG-F-18600: Mlečnost

2.3.10.6.1 Opis funkcionalnosti

Funkcionalnost omogoča vnos **kontrole mlečnosti** pri drobnici (npr. ovce), kjer se spremljajo kvantitativni in kvalitativni parametri mleka posamezne živali. Gre za specializirano meritev, ki se navezuje na predhodne dogodke, kot je jagnjitev/jaritev, in je lahko ena izmed več zaporednih kontrol v času laktacije.

Zajemajo se naslednji podatki:

- zaporedna številka kontrole,
- datum kontrole, začetek in konec molže,
- jutranja količina mleka,
- sestava mleka: maščoba, beljakovine, laktoza, suha snov, sečnina (urea), somatske celice,
- opombe in pripadajoča dokumentacija (npr. poročila laboratorija).

Podatki se lahko uporabijo za rejsko vrednotenje, odločanje o selekciji ali izločitvah ter poročanje rejskih organizacij.

2.3.10.6.2 Predpogoji

- Žival mora biti ovca z evidentiranim dogodkom jagnjitve ali jaritve.
- Žival ne sme imeti zabeleženega konca laktacije v času kontrole.
- Uporabnik mora biti avtoriziran za vnos podatkov.
- Šifranti za merske enote in lastnosti mleka morajo biti aktivni.

2.3.10.6.3 Merila uspešnosti

- Obrazec se naloži v ≤ 1 s.
- Vsi podatki se validirajo glede na pričakovane vrednosti in enote.
- Shranjevanje vnosa se izvede v ≤ 1 s.
- Kontrola je revizijsko sledena (čas, uporabnik, spremembe).
- Dokumentacija se uspešno pripne in indeksira z ID dogodka.

2.3.10.6.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~18600	Vnos podatkov o mlečnosti	Evidentiranje količine mleka in sestave v okviru kontrole mlečnosti.	Žival je ovca, po jagnjitvi, brez končane molže	Vnos podatkov v ≤ 1 s
~18601	Sestava mleka	Vnos maščobe, beljakovin, laktoze, suhe snovi, sečnine in somatskih celic.	Šifranti in enote konfigurirani	Validacija uspešna, vrednosti shranjene
~18602	Evidenca molže	Vnos začetka in konca molže ter zaporedne številke kontrole.	Laktacija ni zaključena	Časovna konsistenca dogodkov
~18603	Pripenjanje poročil	Dodajanje poročil o laboratorijski analizi ali kontroli mleka.	Dogodek v fazi potrditve	Priponka pripeta, indeksirana

2.3.10.6.5 Diagram primerov uporabe

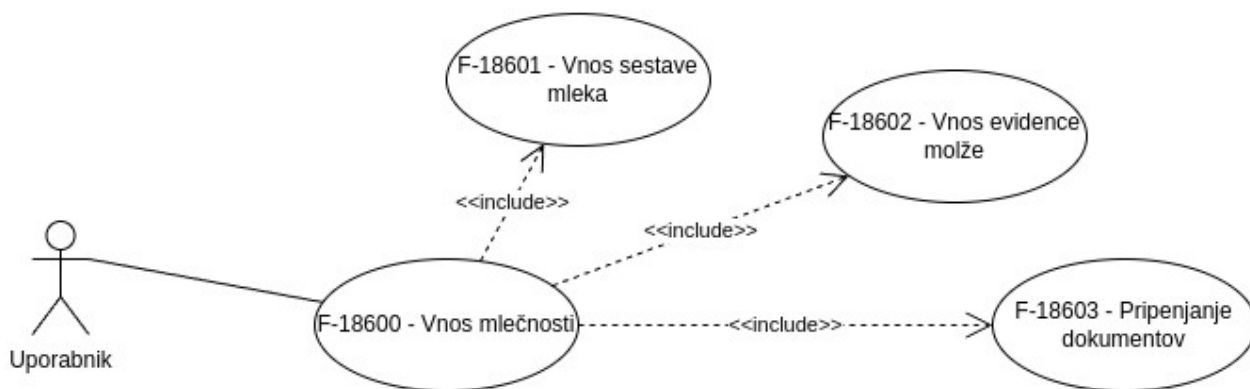


Diagram 74: Diagram primerov uporabe za SKL-DOG-F-18600.

2.3.10.7 SKL-DOG-F-18700: Telesne ocene

2.3.10.7.1 Opis funkcionalnosti

Funkcionalnost omogoča vnos dogodka **telesne ocene drobnice**, pri čemer se ocenjujejo morfološke, funkcionalne in pasemske lastnosti živali. Gre za specializirano obliko meritve, ki se izvaja z namenom selekcijskega vrednotenja živali v okviru rejske presoje.

Vnosna maska omogoča:

- kvantitativno ocenjevanje posameznih morfoloških parametrov (vrednosti 0–999),
- kvalitativno ocenjevanje lastnosti preko šifrantov (opisne ocene),

- pripravo skupne in končne ocene ter opisa živali.

Parametri so razdeljeni v vsebinske sklope:

- **OKVIR:** dolžina trupa, globina in širina prsi, širina križa,
- **OBLIKE:** hrbtina linija, noge, sklepi, biclji,
- **OMIŠIČENOST:** stegno, hrbet,
- **VIME/MODA:** vime, seski, moda, obseg mod,
- **OPISNE LASTNOSTI:** obarvanost, temperament, ostriženost, voljna, testiranost, glava, nosna linija, čeljust, rogatost, število pasesekov, izenačenost vime/moda.

Pripeta je lahko tudi dokumentacija (npr. kontrolni list, slikovni zapis), na podlagi katere se ocena utemeljuje.

2.3.10.7.2 Predpogoji

- Žival mora biti ovca ali oven v aktivnem rejsko-seleksijskem statusu.
- Uporabnik mora biti pooblaščen ocenjevalec oziroma pooblaščen rejska služba.
- Šifranti za opisne ocene morajo biti vnaprej konfigurirani in sinhronizirani.

2.3.10.7.3 Merila uspešnosti

- Obrazec se naloži v ≤ 1 s.
- Vsaka numerična in šifrantna vrednost je validirana pred shranjevanjem.
- Shranjevanje celotne ocene se izvede v ≤ 1 s.
- Dokumenti se pripnejo uspešno, skupaj z vsemi metapodatki.
- Ocena je revizijsko sledena in vidna v profilu živali.

2.3.10.7.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~18700	Vnos telesne ocene	Evidentiranje celovite telesne presoje živali z morfološkimi in opisnimi parametri.	Ovca ali oven v aktivnem rejskem statusu	Shranjevanje v ≤ 1 s
~18701	Morfološke kvantitativne	Vnos meritev v številčni obliki: trup, noge, vime, mišičnost ipd.	Polja validirana na območje 0–999	Napake jasno označene, vnos uspešen
~18702	Kvalitativne ocene	Izbira iz šifrantov za glavo, temperament, obarvanost, testiranost, rogatost ipd.	Aktivni in veljavni šifranti	Vrednosti ustrezno prikazane in zabeležene
~18703	Končna in opisna ocena	Določitev skupne ocene ter utemeljitveni opis v prostem besedilu.	Vnos predhodnih parametrov opravljen	Končna ocena shranjena
~18704	Pripenjanje dokumentacije	Nalaganje spremljajoče dokumentacije (slike, obrazci, kontrolni list).	Dogodek v fazi vnosa ali obravnave	Dokument pripet, indeksiran

2.3.10.7.5 Diagram primerov uporabe

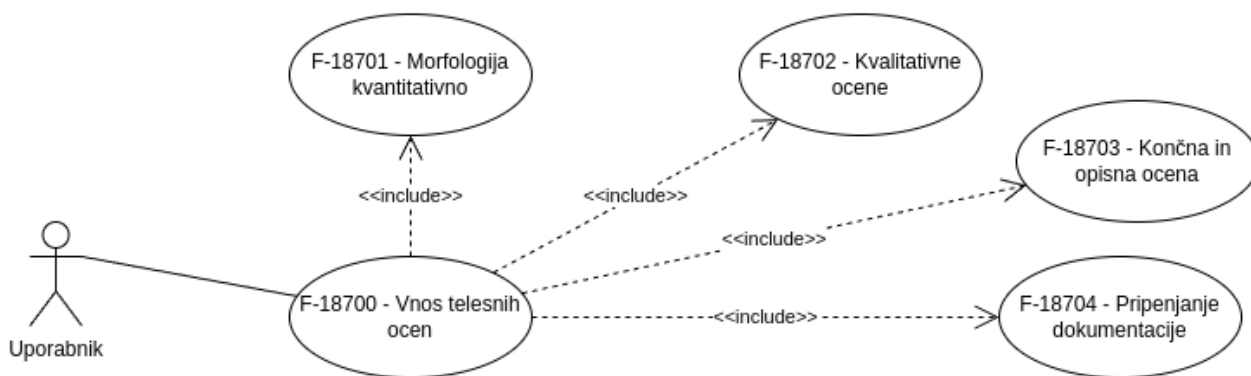


Diagram 75: Diagram primerov uporabe za SKL-DOG-F-18700.

2.3.10.8 SKL-DOG-F-18800: Krvna analiza

2.3.10.8.1 Opis funkcionalnosti

Funkcionalnost omogoča vnos podatkov o **krvnih analizah pri drobnici**, zlasti ob zaključku testiranj. Gre za specializirano evidenco zdravstvenih preverjanj, kjer se beležijo rezultati laboratorijskih preiskav za določene bolezni oziroma okužbe.

Zajeti podatki vključujejo:

- številko izvida krvi in datum izvida,
- rezultate testiranj na naslednje bolezni:
 - Ovčji epididimitis,
 - Enzootsko zvijanje ovac,
 - Paratuberkuloza,
 - Mrzlica Q,
 - Bruceloza ovc in koz,
 - Maedi/Visna,
 - Modri jezik,
- ločeno tudi rezultat TBC testa z lastnim izvidom in datumom.

Rezultati testov so zabeleženi v obliki:

- **pozitivno (+),**
- **negativno (-),**
- **nejasno (?),**
- **ni testa (Ø).**

Možen je tudi vnos za več živali hkrati (skupinski vnos). Obrazec podpira pripenjanje dokumentacije (npr. originalni laboratorijski izvid) in generiranje izpisov.

2.3.10.8.2 Predpogoji

- Žival mora biti drobnica z veljavnim ID v sistemu.
- Uporabnik mora imeti avtorizacijo za vnos veterinarskih podatkov.
- Datumi izvidov morajo biti pravilno oblikovani in časovno smiselni.
- Vsaj ena bolezen mora biti ocenjena pri posameznem vnosu.

2.3.10.8.3 Merila uspešnosti

- Vnos se shrani v ≤ 1 s.
- Vsi rezultati se validirajo po logiki (izključno ena izbira na bolezen).
- Pripeti dokument se pravilno poveže z dogodkom in evidentira.
- Vnos podpira izpis za posamezno ali več živali.
- Vsaka analiza je revizijsko sledena.

2.3.10.8.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~18800	Vnos krvne analize	Vnos rezultatov krvnih testiranj po boleznih ter podatkov o izvidu.	Aktivna žival, avtorizacija, vsaj ena ocena	Vnos shranjen v ≤ 1 s
~18801	Vnos TBC testa	Ločen vnos TBC testa z lastnim izvidom in datumom.	TBC test opravljen	Vnos validiran in shranjen
~18802	Validacija rezultatov	Vsaka bolezen ima natanko eno izbrano vrednost (izključujoče izbire).	UI omogoča samo eno izbiro na vrstico	Napačni vnosi zavrženi z opozorilom
~18803	Pripenjanje dokumentacije	Pripenjanje PDF-jev in skenov izvida.	Dokumentacija pripravljena	Priponka uspešno shranjena
~18804	Vnos za več živali	Možnost izvedbe dogodka krvne analize za več živali hkrati (bulk operacija).	Izbrana množica živali	Vnos izveden brez prekinitve, sledljiv

2.3.10.8.5 Diagram primerov uporabe

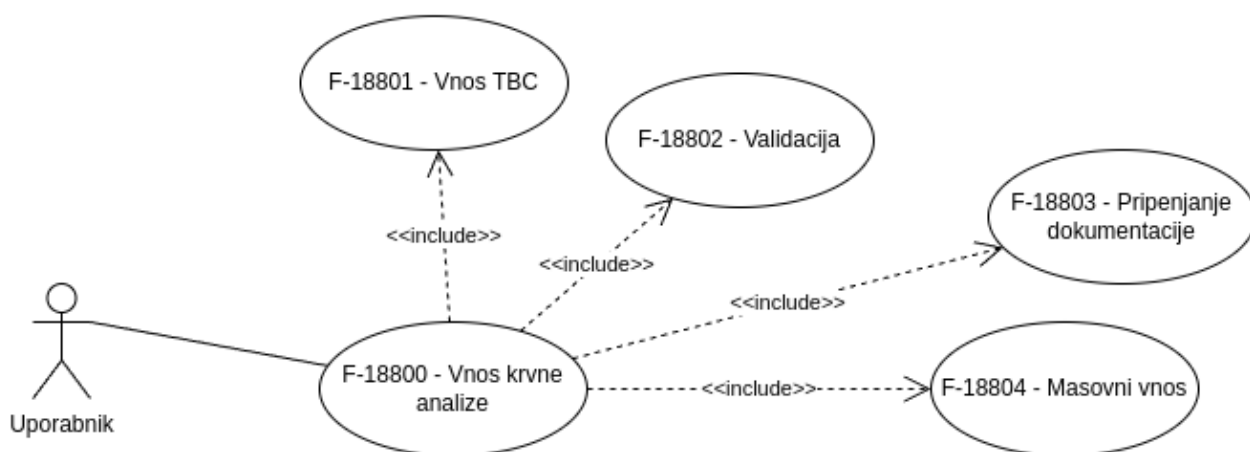


Diagram 76: Diagram primerov uporabe za SKL-DOG-F-18800.

2.3.10.9 SKL-DOG-F-18900: Odbira drobnice

2.3.10.9.1 Opis funkcionalnosti

Funkcionalnost omogoča vnos dogodka **odbire jagnjet ali jarčkov** pri drobnici, s ciljem klasifikacije živali glede na nadaljnjo rejo, izločitev ali vključitev v testiranje. Vnos vključuje osnovne fizične in statusne parametre posamezne živali ter dodatne podatke prek pop-up obrazca.

Za vsako žival se beležijo:

- ID številka (rodovna številka),
- spol,
- datum odstavitve,
- telesna masa,
- status izločitve (DA/NE),
- vzrok izločitve (iz šifranta),
- opomba.

Znotraj **pop-up obrazca** (ob kliku na rodovno številko) je možno dodatno vnesti:

- ime živali,
- številko tetoviranja,
- genotip,
- datum vstopa v čredo,
- izvor (rejec, tip izvora).

Možen je tudi prenos dokumentacije (dnevnik odbire, zapisnik komisije) in generiranje izpisa (npr. rejski list, odločba o izločitvi).

2.3.10.9.2 Predpogoji

- Žival mora biti predhodno vnešena kot del legla (jagnje ali jarec).
- Uporabnik mora biti avtoriziran za delo z rejcem.
- ID-ji živali in datum odstavitve morajo biti veljavni.
- Če je označena izločitev, mora biti izbran vzrok.

2.3.10.9.3 Merila uspešnosti

- Uporabniški vnos ≤ 30 s na žival.
- Vsi numerični vnosi (masa) in šifranti (vzrok izločitve, izvor) se validirajo.
- Shranjevanje se izvede v ≤ 1 s.
- Dokumenti se pripnejo in povežejo z dogodkom.
- Podatki so revizijsko zabeleženi in dostopni v kasnejših fazah (npr. analiza reje).

2.3.10.9.4

Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~18900	Odbira drobnice	Vnos osnovnih podatkov o živalih ob odbiri – status, masa, izločitev.	Registrirane živali, avtorizacija	Vnos ≤ 30 s na žival, validacija uspešna
~18901	Vnos vzroka izločitve	Vnos vzroka iz šifranta, če je označena izločitev.	Pogojno obvezen vnos	Napaka ob manjkajočem vnosu
~18902	Dodatni podatki v pop-upu	Ime, tetoviranje, datum vstopa, izvor – dostopni preko klikabilnega ID.	Žival mora obstajati ali biti v pripravi	Pop-up validiran, povezan z ID
~18903	Pripenjanje dokumentov	Dnevnik odbire, komisijski zapisnik ali obrazci.	Dokument v podprtem formatu	Dokument shranjen, povezan z dogodkom
~18904	Generiranje izpisa	Priprava izpisov (npr. odločba o izločitvi, selekcijska odločitev).	Vnos uspešno zaključen	Izpis generiran v ≤ 3 s

2.3.10.9.5

Diagram primerov uporabe

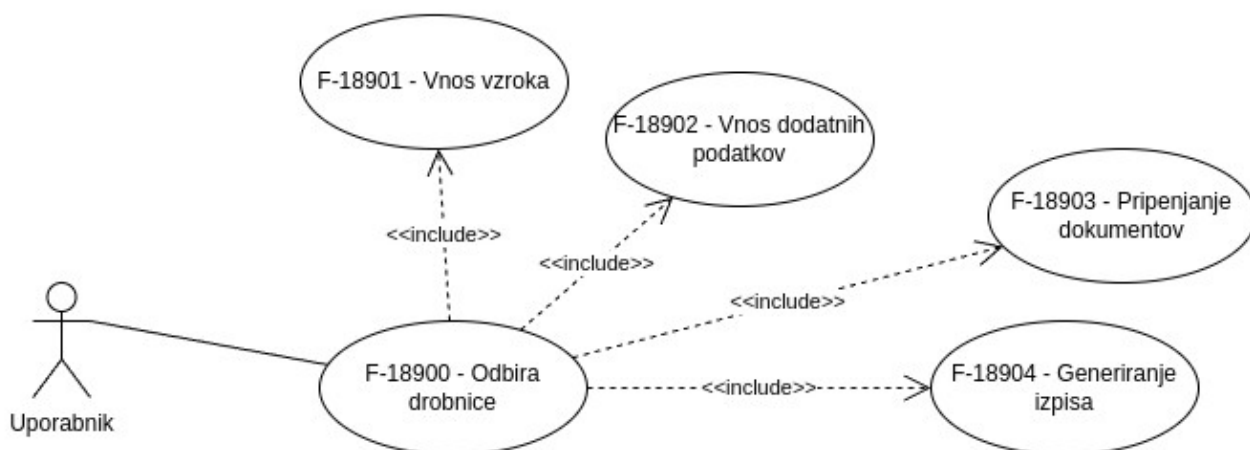


Diagram 77: Diagram primerov uporabe za SKL-DOG-F-18900.

2.3.10.10 SKL-DROB-F-22000: Molža

2.3.10.10.1 Opis funkcionalnosti

Funkcionalnost omogoča vnos podatkov o **začetku in koncu molže** pri drobnici, ki so bile vključene v molzni cikel. Sistem na osnovi predhodno zabeleženih dogodkov (zlasti **jagnjitev/jaritev** in **odstavitev**) omogoča filtriran in dinamičen prikaz vseh živali, pri katerih je mogoče evidentirati začetek ali konec molže.

Maska je razdeljena na dva dela:

- **Zajem začetka molže** za vse živali pri rejcu, kjer je mogoče filtrirati po obdobju jagnjitve in po različnih kriterijih (npr. živali brez začetka molže, z odstavljenimi mladiči ipd.).
- **Zajem konca molže** za vse živali, kjer prav tako obstajajo različni iskalni kriteriji (npr. brez konca molže, izločene živali, zaključen nadzor nad molžo ipd.).

Uporabnik lahko hkrati vnese začetek oz. konec molže za več živali, na podlagi izpisanega seznama.

2.3.10.10.2 Predpogoji

- Žival mora imeti vnesen dogodek jagnjitve/jaritve.
- Za konec molže: mora biti vnesen tudi začetek molže.
- Uporabnik mora biti avtoriziran za delo s podatki rejca.
- Filtri in iskalni kriteriji morajo biti ustrezno konfigurirani.

2.3.10.10.3 Merila uspešnosti

- Seznam živali se naloži v ≤ 1 s glede na iskalne pogoje.
- Vnos začetka ali konca molže se izvede za do 50 živali hkrati.
- Vsaka sprememba je revizijsko zabeležena (čas, uporabnik, ID živali).
- Validacije preprečijo vnose, kjer pogoji (npr. zaporedje datumov) niso izpolnjeni.
- Vnos za izločene živali je dovoljen le za zaključek molže, če ni bil predhodno vpisan.

2.3.10.10.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~22001	Vnos začetka molže	Evidentiranje začetka molže na podlagi dogodka jagnjitve oz. jaritve.	Žival ima vnešeno jagnjitev, avtorizacija	Vnos za več živali, validacija datuma
~22002	Vnos konca molže	Evidentiranje zaključka molže, če obstaja začetek molže ali so izpolnjeni drugi pogoji.	Žival ima začetek molže ali je izločena	Shranjevanje v ≤ 1 s, revizijska sled
~22003	Filtriranje po kriterijih	Možnost iskanja živali brez začetka, brez konca, po jagnjitvi ali odstavitvi.	Aktivna evidenca dogodkov	Seznam generiran v ≤ 1 s
~22004	Bulk vnos za več živali	Možnost vnosa za več živali hkrati z enotno ali individualno vrednostjo.	Uporabnik izbere seznam, vnese datum	Shranjeno za vse brez napake

2.3.10.10.5 Diagram primerov uporabe

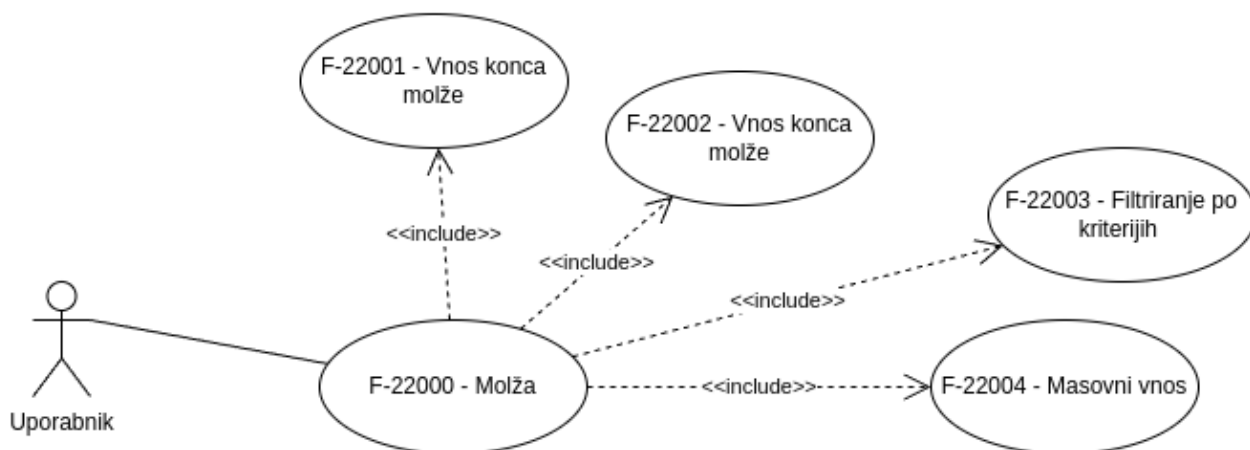


Diagram 78: Diagram primerov uporabe za SKL-DROB-F-22000.

2.3.10.11 SKL-DROB-F-22100: Odkup

2.3.10.11.1 Opis funkcionalnosti

Funkcionalnost omogoča **beleženje odkupa jagenjčkov** pri drobnici. Sistem omogoča vnos podatkov o posamezni živali, ki je bila odkupljena, ter beleženje osnovnih informacij o odkupu. Vsaka žival je lahko ob odkupu dodana v določeno skupino, kar omogoča lažje kasnejše sledenje in poročanje.

Zajeti podatki vključujejo:

- rodovno številko živali,
- težo ob odkupu,
- rejca (prodajalca),
- datum nakupa (odkupa),
- spremni list (številka dokumenta),
- lokacijo, kamor se žival preseli po odkupu (npr. izolatorij, testna postaja),
- skupino odkupa (izbrana obstoječa ali ustvarjena nova).

2.3.10.11.2 Predpogoji

- Žival mora imeti veljavno identifikacijo.
- Uporabnik mora biti avtoriziran za vnos odkupa.
- Datum odkupa ne sme biti v prihodnosti.
- Lokacija mora biti izbrana iz šifranta dovoljenih namestitvenih enot.

2.3.10.11.3 Merila uspešnosti

- Vnos podatkov se izvede v ≤ 1 s.
- Vsaka žival je ob odkupu povezana s točno eno skupino.
- Vse spremembe so revizijsko zabeležene (uporabnik, čas, ID živali).

- Skupine se lahko ustvarjajo dinamično ob vnosu.
- Podatki so na voljo za nadaljnje poročanje in selekcijsko obdelavo.

2.3.10.11.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~22100	Vnos odkupa jagenjčka	Zajem podatkov o odkupu posamezne živali: identifikacija, teža, rejec, datum, spremni list	Avtorizacija, aktivna žival, veljavna lokacija	Vnos $v \leq 1$ s
~22101	Dodelitev v skupino	Možnost izbire ali ustvarjanja skupine za odkupljene živali.	Vnos vsaj ene živali	Vsaka žival povezana s skupino
~22102	Selitev na lokacijo	Ob odkupu se žival premesti na izbrano ciljno lokacijo (npr. izolatorij).	Lokacija iz šifranta	Selitev zabeležena in sledljiva
~22103	Pripenjanje spremnega lista	Možnost vnosa številke spremnega lista in pripenjanja dokumenta.	Dokumentacija pripravljena	Podatek shranjen in revizijsko označen

2.3.10.11.5 Diagram primerov uporabe

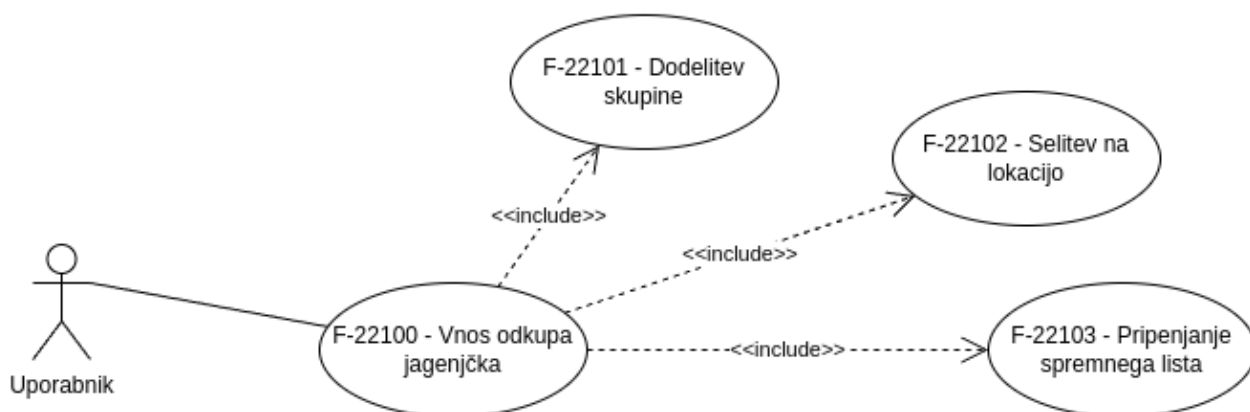


Diagram 79: Diagram primerov uporabe za SKL-DROB-F-22100.

2.3.10.12 SKL-DROB-F-22200: Pregled drobnice pri rejcu

2.3.10.12.1 Opis funkcionalnosti

Funkcionalnost omogoča **pregled vseh živali vrste drobnica**, ki so registrirane pri izbranem rejcu. Pregledna tabela vključuje ključne attribute posamezne živali ter omogoča masovne operacije, kot je sprememba lokacije za več izbranih živali hkrati.

V pregled so vključeni naslednji podatki:

- identifikacijska številka (ID) živali,
- spol,
- pasma,
- genotip,
- datum rojstva,

- ID matere in očeta,
- koeficient inbridinga,
- NSP (neznani starševski prispevek),
- opombe,
- datum spremembe lokacije,
- indikator za trenutno lokacijo.

Poleg pregleda je omogočena funkcionalnost za **masovno selitev** živali:

- izbor živali prek potrditvenih polj,
- določitev nove lokacije (iz šifrantu),
- določitev datuma selitve.

2.3.10.12.2 Predpogoji

- Uporabnik mora biti avtoriziran za dostop do podatkov rejca.
- Živali morajo imeti veljavno identifikacijo in status.
- Lokacije morajo biti predhodno registrirane v šifrantu.
- Datum selitve mora biti veljaven in ne sme biti v prihodnosti.

2.3.10.12.3 Merila uspešnosti

- Prikaz seznama živali se naloži v ≤ 1 s.
- Iskanje in filtriranje po stolpcih je odzivno in omogoča hitro selekcijo.
- Selitev več živali (do 100) se izvede brez prekinitve.
- Spremembe lokacij so revizijsko sledene in vezane na uporabnika.
- Vsi podatki so usklajeni z matično evidenco.

2.3.10.12.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~22200	Pregled živali drobnice	Prikaz seznama vseh živali vrste drobnica, registriranih pri rejcu.	Avtorizacija, registriran rejec	Seznam naložen v ≤ 1 s
~22201	Podatki o živali	Prikaz ključnih atributov živali (spol, pasma, starši, NSP, opombe).	Veljavni podatki v bazi	Podatki skladni z evidenco
~22202	Masovni prenos živali	Omogočen prenos označenih živali na novo lokacijo z izbranim datumom.	Lokacija izbrana, datum veljaven	Selitev ≤ 1 s na vnos

2.3.10.12.5 Diagram primerov uporabe

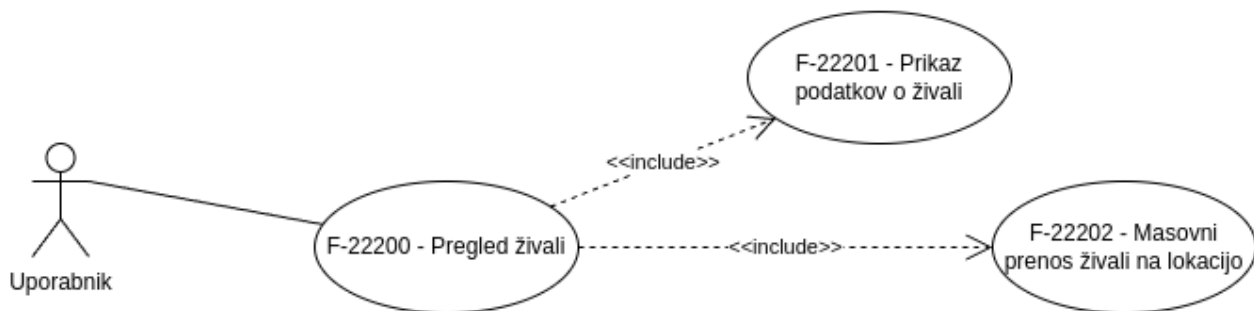


Diagram 80: Diagram primerov uporabe za SKL-DROB-F-22200.

2.3.10.13 SKL-DROB-F-22300: Masovno nalaganje podatkov za drobnico

2.3.10.13.1 Opis funkcionalnosti

Funkcionalnost omogoča **masovni uvoz podatkov** za drobnico preko različnih vrst strukturiranih dokumentov (npr. CSV, DOCX, XML, XLSX). Uporabnik izbere vrsto podatkov, izvorno datoteko ter konfigurira preslikave med podatkovnimi stolpci in ciljnim atributi v podatkovni bazi.

Podprti so naslednji tipi nalaganja:

- **plemenske vrednosti (PV),**
- **vrednosti tehtanja, mlečnosti in volne,**
- **laboratorijski podatki (npr. TSE, volna),**
- **rodovni podatki (oče, mati),**
- **meritve hrbtne mišice,**
- **odkup mladice.**

Funkcionalnost omogoča:

- izbor datoteke iz lokalnega sistema (.csv, .txt),
- določitev ločil (podpora za ; in ,),
- preslikavo stolpcev (Data/Table Mapping),
- pregled napredka uvoza po fazah (naloži → preslikaj → validiraj → shrani),
- zapis rezultatov z validacijskimi napakami.

Navodila za posamezni tip nalaganja so na voljo preko povezav do ustreznih **procesnih dokumentov** (npr. *proces_LDR_tehtanje_logatec.docx*, *proces_LDR_TSE.docx*).

2.3.10.13.2 Predpogoji

- Uporabnik mora imeti ustrezno vlogo (npr. rejska služba, laboratorij).
- Podatki v datoteki morajo ustrezati zahtevani strukturi.
- Za vsak tip nalaganja mora obstajati konfiguracijska definicija (mapping).

- Dovoljeni so samo preverjeni tipi podatkovnih zapisov (s checksum ali validacijo glave).

2.3.10.13.3 Merila uspešnosti

- Branje datoteke in preslikava stolpcev se zaključi v ≤ 5 s.
- Uvoz do 10.000 zapisov se izvede v ≤ 30 s.
- Neuspešni vnosi so jasno izpisani z razlogom napake.
- Sistem beleži uspešno in neuspešno obdelane vrstice.
- Vsak uvoz je revizijsko sledljiv (čas, uporabnik, tip nalaganja, datoteka).

2.3.10.13.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~22300	Masovni uvoz podatkov	Nalaganje PV podatkov preko .csv datotek, možnost merge ali nadomestitve.	Konfiguriran mapping, datoteka ustrezne oblike	≤ 5 s do validacije, 1 s/1000 zapisov
~22301	Izbor tipa uvoza	Uporabnik naloži datoteko in izbere tip uvoza (kaj uvaža)	Validna struktura, ločena polja	Validacija in zapis z ločenim izpisom napak
~22302	Konfiguracija uvoza	Data/table mapping z možnostjo ignoriranja določenih stolpcev.	Polja pravilno preslikana	Vsaj 1 ciljna tabela, vsi obvezni atributi
~22303	Pregled napredka	Vizualni prikaz faz uvoza z možnostjo prekinitve ali ponovnega zagona.	Uvoz v teku	Napredek se osvežuje sproti

2.3.10.13.5 Diagram primerov uporabe

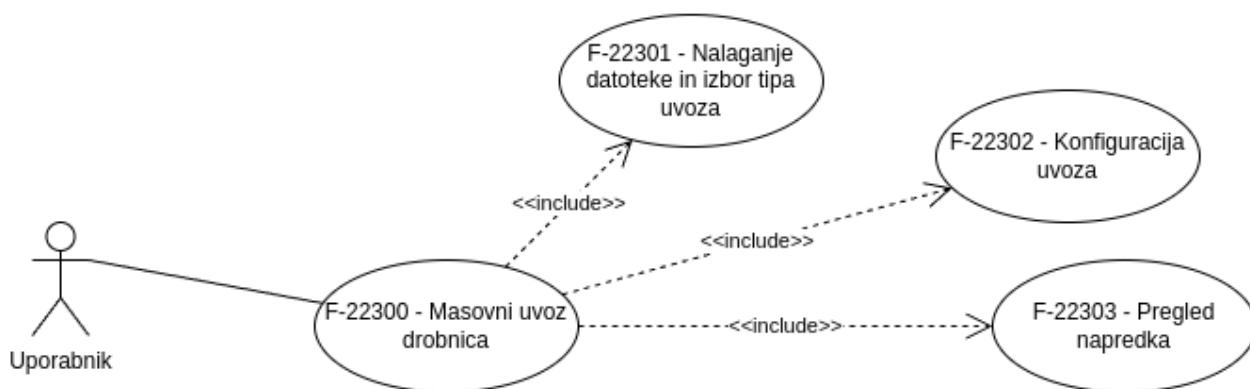


Diagram 81: Diagram primerov uporabe za SKL-DROB-F-22300.

2.3.11 SKL-Pras: Sklop prašiči

2.3.11.1 SKL-DOG-F-17200: Tetoviranje

2.3.11.1.1 Opis funkcionalnosti

Funkcionalnost omogoča vnos dogodka tetoviranja za živali, pri katerih je tak dogodek predviden glede na njihovo živalsko vrsto (npr. pri prašičih). Vnosna maska omogoča sočasen vnos več

tetoviranih živali, določitev tetovirne številke, spola, ušesne lege ter seska (levo/desno), kamor je številka vtisnjena. Uporabnik lahko izbere vnašalca, vnese datum tetoviranja, opombo in pripne dokumentacijo. Pri kraškopoljcu še obarvanost in namen posameznega pujska.

Če tetoviranje za določeno vrsto ni predvideno, se ta funkcionalnost uporabniku ne ponudi.

2.3.11.1.2 Predpogoji

- Vrsta živali mora imeti konfigurirano možnost tetoviranja.
- Uporabnik mora imeti pravice za vnos dogodkov nad izbranimi živalmi.
- Podatki o posameznih živalih morajo biti predhodno vnešeni ali izbrani iz konteksta.

2.3.11.1.3 Merila uspešnosti

- Obrazec se naloži v ≤ 500 ms, če je tetoviranje dovoljeno.
- Vnos do 20 tetoviranih živali v eni operaciji mora biti izveden brez napake.
- Vsi podatki se uspešno shranijo, priponke pa se vežejo na ID dogodka.
- Uporabnik je jasno obveščen, če tetoviranje za vrsto ni podprto (masko onemogočeno).

2.3.11.1.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~17200	Vnos dogodka tetoviranja	Vnos tetovirnih števil, ušesne lege, seskov in pripadnosti spolu za več živali hkrati.	Vrsta omogoča tetoviranje, uporabnik ima pravice	Vnos uspešen za do 20 živali, zabeležen dogodek
~17201	Pripenjanje dokumentacije	Možnost nalaganja dokumentacije o tetoviranju (dnevnik, listi, potrdila).	Dogodek v fazi vnosa ali urejanja	Priponke uspešno shranjene
~17202	Priprava dokumentov	Samodejna priprava dokumentov (npr. potni list) po zaključenem tetoviranju.	Dogodek uspešno zaključen	Dokument ustrezno pripravljen

2.3.11.1.5 Diagram primerov uporabe

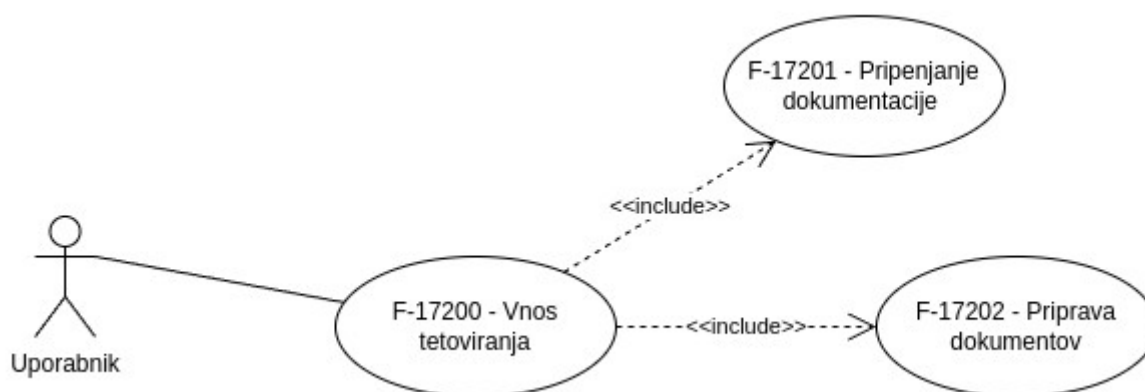


Diagram 82: Diagram primerov uporabe za SKL-DOG-F-17200.

2.3.11.2 SKL-DOG-F-17400: Meritve prašičev

2.3.11.2.1 Opis funkcionalnosti

Funkcionalnost omogoča zajem **standardnih morfoloških meritev** za posameznega prašiča, ki so

osnova za selekcijsko presojo, rejski nadzor in spremljanje rasti v različnih fazah reje.

Zajemajo se naslednji parametri:

- **telesna masa,**
- **DHS meritve** (DHS1, DHS2, DHS3),
- **debelina hrbtna mišice A,**
- **debelina hrbtna slanina A,**
- **število seskov levo in desno.**

Dodatno so omogočeni:

- izbor **vnašalca** in **tipa dogodka** (npr. selekcijska meritev, testiranje ipd.),
- vnos **datuma meritve,**
- polje za **opombo,**
- možnost **pripenjanja dokumentacije** (npr. evidenčni list, ročni zapisnik),
- priprava izpisov (npr. potni list, kontrolna tabela).

2.3.11.2.2 Predpogoji

- Žival mora biti prašič z veljavnim identifikatorjem in aktivnim statusom.
- Vsi šifranti (vnašalec, dogodek) morajo biti pravilno konfigurirani in sinhronizirani.
- Vnosne vrednosti morajo biti validirane glede na mejne vrednosti (npr. debelina slanina, masa).
- Datum meritve mora biti enak ali manjši od trenutnega datuma.

2.3.11.2.3 Merila uspešnosti

- Maska se naloži v ≤ 1 s.
- Validacija vseh numeričnih vnosov je izvedena ob shranjevanju.
- Vnos posameznega dogodka traja ≤ 1 s.
- Dokumenti se uspešno pripnejo in indeksirajo ob shranjevanju.
- Vsi podatki so revizijsko zabeleženi.

2.3.11.2.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~17400	Vnos standardnih meritev	Evidentiranje telesne mase, DHS, debeline mišice in slanina ter seskov.	Avtorizacija, aktivna žival	Vnos uspešen v ≤ 1 s
~17401	Validacija numeričnih vnosov	Vrednosti morajo biti v pričakovanih območjih (npr. masa > 0 , debeline < 100 mm).	Veljavne številke	Napake jasno označene, vnos zavržen
~17402	Pripenjanje dokumentacije	Možnost nalaganja dokumentov ob meritvi (slike, PDF).	Dokument v sprejemljivi	Priponka zabeležena z ID

ID	Naziv	Opis	Predpogoji obliki	Merila uspešnosti meritve
----	-------	------	----------------------	------------------------------

2.3.11.2.5 *Diagram primerov uporabe*

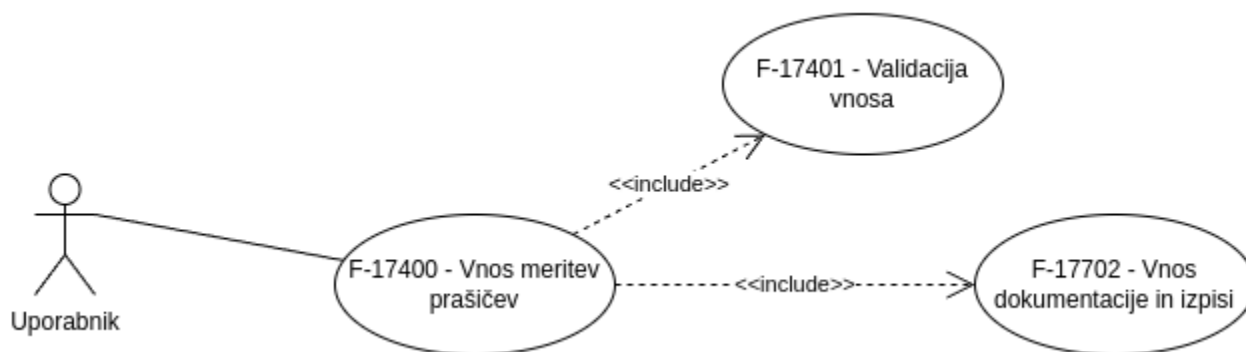


Diagram 83: Diagram primerov uporabe za SKL-DOG-F-17400.

2.3.11.3 *SKL-DOG-F-17500: Odstavitev prašičkov*

2.3.11.3.1 **Opis funkcionalnosti**

Funkcionalnost omogoča evidentiranje dogodka **odstavitve pujskov** pri svinji. Zajema se seznam vseh pujskov iz gnezda, skupaj z njihovimi osnovnimi atributi, datumom odstavitve, seski in stanjem svinje po odstavitvi.

Vnosna maska omogoča:

- **vnos datuma odstavitve** za vse pujske ali posamezno,
- pregled tetovirne številke, spola, števila seskov levo in desno za vsakega pujska,
- določitev **kondicije svinje** po odstavitvi (iz šifranta),
- vnos opombe (npr. opazovanja po odstavitvi, zdravstveno stanje),
- pripenjanje dokumentacije (dnevnik pripustov, poročila o gnezdu),
- pripravo izpisa (npr. poročilo o odstavitvi).

Podatki so ključni za rejski nadzor, sledenje rasti, zdravja in nadaljnje selekcije pujskov.

2.3.11.3.2 **Predpogoji**

- Vsi pujski morajo biti predhodno evidentirani kot del gnezda (povezani z mamom).
- Uporabnik mora imeti pravico za vnos dogodkov za izbranega rejca.
- Šifranti (vnašalec, kondicija, spol) morajo biti aktivni in usklajeni.
- Datum odstavitve mora biti enak ali večji od datuma prasiatve.

2.3.11.3.3 **Merila uspešnosti**

- Vnos vseh podatkov za do 10 pujskov hkrati mora biti zaključen v ≤ 1 min.
- Shranjevanje se izvede v ≤ 1 s.
- Dokumenti se pripnejo k dogodku in indeksirajo z ID gnezda.

- Vsi podatki so revizijsko zabeleženi.
- Vsak pujs po odstavitvi preide v ustrezno nadaljnjo obdelavo (npr. odbira, tetoviranje).

2.3.11.3.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~17500	Vnos dogodka odstavitve	Evidentiranje datuma in parametrov odstavitve za celotno gnezdo.	Aktivna žival, povezana gnezda	Vnos ≤ 1 min, shranjevanje ≤ 1 s
~17501	Atributi pujskov	Spol, seski levo/desno, tetovirna številka za vsakega pujska.	Predhodno vnešeni pujski	Polja validirana in pravilno shranjena
~17502	Kondicija svinje	Vnos kondicije svinje po odstavitvi iz šifrant.	Šifrant aktiven	Uporabniku prikazan izbor, validiran vnos
~17503	Pripenjanje dokumentov	Priloženi PDF ali druge datoteke (npr. dnevnik, list).	Veljavna datoteka, dogodek v fazi vnosa	Dokument priložen in indeksiran
~17504	Generiranje izpisa	Priprava dokumenta (npr. rejski list, evidenčni zapis) o odstavitvi.	Vnos zaključen	Dokument pripravljen v ≤ 3 s

2.3.11.3.5 Diagram primerov uporabe

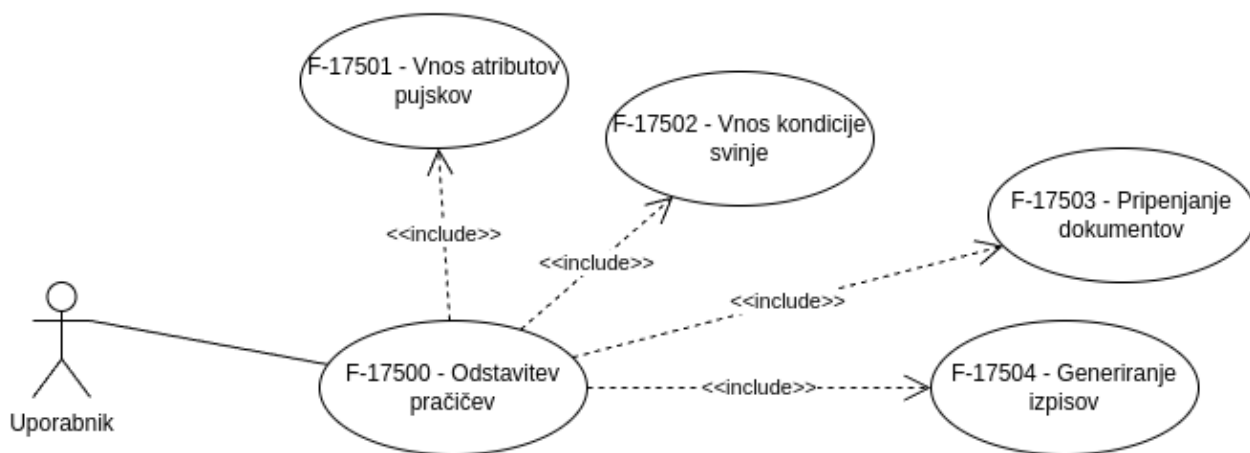


Diagram 84: Diagram primerov uporabe za SKL-DOG-F-17500.

2.3.11.4 SKL-DOG-F-18100: Prasitev

2.3.11.4.1 Opis funkcionalnosti

Funkcionalnost omogoča vnos dogodka **prasitev** pri prašičih. Uporabnik vpiše podatke o datumu prasiatve, lokaciji (hlev, boks), številu živorojenih, mrtvorojenih in mumificiranih pujskov, maso gnezda, dodatek ali odvzem pujskov ter stanje (kondicijo) svinje ob prasiatvi.

Dodatno je omogočeno:

- vnos opombe,
- pripenjanje dokumentov (npr. evidenčni list, dnevnik),
- generiranje izvoznega dokumenta (npr. potni list).

Podatki se ob potrditvi zapišejo kot časovno pomemben dogodek v življenjski cikel živali in so osnova za kasnejše dogodke (odstavitev, tetoviranje, odbira).

2.3.11.4.2 Predpogoji

- Žival mora biti registrirana svinja in odbrana.
- Uporabnik mora biti avtoriziran za vnos dogodkov.
- Šifranti (kondicija, lokacija, boks) morajo biti konfigurirani.
- Datum prasiatve ne sme biti v prihodnosti.

2.3.11.4.3 Merila uspešnosti

- Maska se naloži v ≤ 500 ms.
- Vsi ključni podatki se validirajo pred shranjevanjem.
- Vnos dogodka se zaključi v ≤ 1 s.
- Napake se jasno označijo v uporabniškem vmesniku.
- Vsi podatki so revizijsko sledeni (čas, uporabnik, vir vnosa).

2.3.11.4.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~18100	Vnos dogodka prasitev	Zajem podatkov o prasiatvi, številu pujskov, masi, pogojih in pripadajoči dokumentaciji.	Svinja z aktivnim statusom, ustrezna vloga	Vnos uspešen v ≤ 1 s
~18101	Beleženje števila in mase	Vnos števila živorojenih, mrtvorojenih, mumificiranih, mase gnezda, dodajanja/odvzema.	Vrednosti ≥ 0 , validirane	Podatki pravilno shranjeni
~18102	Evidenca lokacije	Izbira lokacije (hlev, boks) iz šifrant in zapis v dogodek.	Konfigurirani šifranti	Vnos brez napake, prikaz v UI
~18103	Pripenjanje dokumentov	Dodajanje spremljajočih dokumentov (npr. dnevnik, potrdilo).	Dokumentacija pripravljena	Datoteka pripeta in indeksirana

2.3.11.4.5

Diagram primerov uporabe

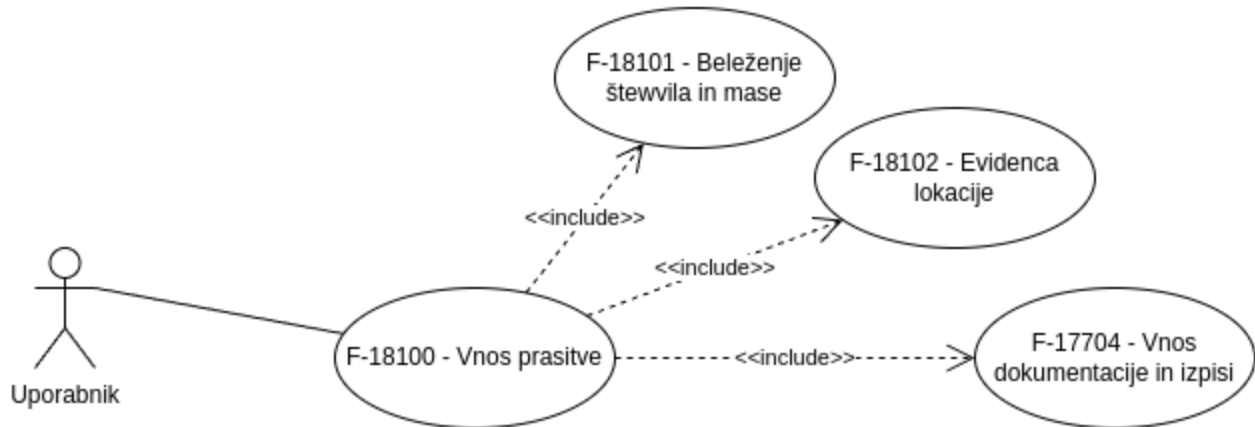


Diagram 85: Diagram primerov uporabe za SKL-DOG-F-18100.

2.3.11.5

SKL-DOG-F-18500: Odbira plemenskega podmladka

2.3.11.5.1

Opis funkcionalnosti

Funkcionalnost omogoča izvedbo dogodka **odbiranja pujskov** za nadaljnjo rejo ali testiranje. Pri odbiru se ocenjuje fizični status pujska, beležijo se osnovne telesne in genetske lastnosti ter tip preizkusa, v katerega je žival vključena.

Maska omogoča vnos naslednjih podatkov:

- tetovirna številka pujska (povezana s pop-up obrazcem za dodatne podatke),
- spol,
- število seskov (levo/desno),
- datum odstavitve,
- meritve (DHS1, DHS2, DHS3, telesna masa),
- tip preizkusa (npr. selekcija, kontrolni zakol, testiranje v kontrolni reji),
- opomba.

V pop-up obrazcu za dodatne podatke se beležijo:

- ušesna številka,
- genotip,
- datum vstopa v čredo,
- izvor,
- lokacija namestitve (hlev, boks).

Dogodku je možno priložiti dokumentacijo (npr. rejski list, obrazec za vnos v testno postajo) ter generirati izpis.

2.3.11.5.2 Predpogoji

- Pujski morajo biti predhodno registrirani in povezani z mamom (prasitev).
- Uporabnik mora biti avtoriziran za vnos dogodkov.
- Šifranti (tip preizkusa, lokacije, genotip, izvori) morajo biti aktivni.
- Vnosi (DHS, teža, seski) morajo biti validirani po numeričnem formatu.

2.3.11.5.3 Merila uspešnosti

- Odbira se izvede za ≥ 1 in ≤ 10 pujskov hkrati.
- Shranjevanje in priprava dogodka se izvede v ≤ 1 s.
- Vsi vnosi so revizijsko sledeni in takoj na voljo za nadaljnjo obdelavo.
- Sistem podpira samodejni prenos v register testnih živali (če aktivirano).

2.3.11.5.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~18500	Vnos podatkov o odbiri	Vnos tetovirke, seskov, meritve, odstavitve in tipa preizkusa za pujske.	Aktivna prasitev, avtorizacija	Shranjevanje v ≤ 1 s
~18501	Pop-up vnos dodatnih podatkov	Ob kliku na tetovirko se odpre obrazec za dodatne podatke (ušesna št., lokacija ipd.).	Žival že obstaja ali je v pripravi	Vnos brez napak, povezan s pujskom
~18502	Tip preizkusa	Izbira iz šifranta (selekcija, testiranje ipd.)	Šifrant konfiguriran	Validacija uspešna
~18503	Dokumentacija	Prilaganje priponk (npr. testna odločba, zapisnik).	Dokument v podprti obliki	Priponka indeksirana
~18504	Priprava izpisa	Izpis za izvoz v PDF (npr. potrdilo o vključitvi, rejski list).	Dogodek uspešno shranjen	Izpis v ≤ 3 s

2.3.11.5.5 Diagram primerov uporabe

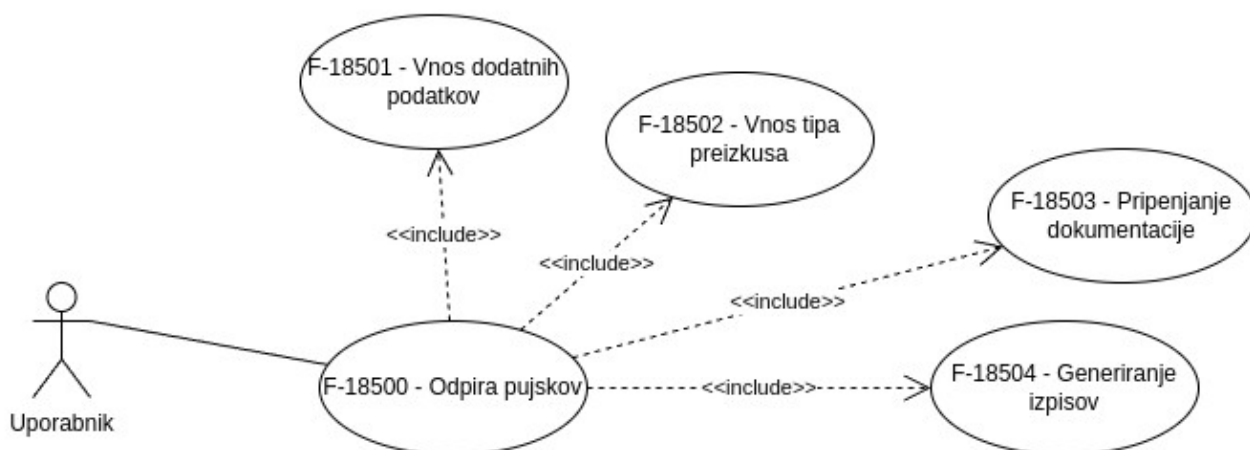


Diagram 86: Diagram primerov uporabe za SKL-DOG-F-18500.

2.3.11.6 SKL-PRAS-F-24000: Pregled prašičev pri rejcu

2.3.11.6.1 Opis funkcionalnosti

Funkcionalnost omogoča pregled vseh prašičev, registriranih pri izbranem rejcu, z možnostjo urejanja lokacij živali. Pregled vključuje ključne identifikacijske, pasemske in lokacijske podatke ter omogoča skupinske operacije, kot je **masovna selitev** živali na novo lokacijo.

Zajeti podatki v tabeli vključujejo:

- identifikacijsko številko (ID) živali,
- spol,
- pasmo ali genotip,
- datum rojstva,
- rejca,
- trenutno lokacijo,
- datum zadnje spremembe lokacije,
- indikator prisotnosti v trenutni lokaciji.

Funkcionalnost spodnjega modula omogoča:

- izbor ene ali več živali,
- določitev nove ciljne lokacije prek iskalnika lokacij,
- vnos datuma selitve,
- potrditev spremembe lokacije.

2.3.11.6.2 Predpogoji

- Uporabnik mora biti avtoriziran za dostop do podatkov izbranega rejca.
- Živali morajo imeti veljavne ID-je in aktivni status.
- Lokacije morajo biti vnaprej vpisane v šifrant.
- Datum selitve mora biti pravilen in ne sme biti v prihodnosti.

2.3.11.6.3 Merila uspešnosti

- Seznam živali se naloži v ≤ 1 s.
- Sprememba lokacije za več živali se izvede v ≤ 2 s brez prekinitve.
- Vse spremembe so revizijsko sledene (čas, uporabnik, stara in nova lokacija).
- Vnos omogoča jasn pregled trenutne lokacije in ciljne spremembe.
- Vnos ne vpliva na druge aktivne dogodke, vezane na žival (npr. molžo, prasiatve).

2.3.11.6.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~24000	Pregled prašičev	Prikaz vseh prašičev pri rejcu z možnostjo razvrščanja in iskanja po atributih.	Avtorizacija, aktivne živali	Tabela naložena v ≤ 1 s

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~24001	Pregled podatkov o živalih	Prikaz spola, pasme, genotipa, rojstva, lokacije, rejca in sprememb.	Podatki ustrezno usklajeni z evidenco	Vse vrednosti konsistentne z bazo
~24002	Skupinska selitev živali	Možnost spreminjanja lokacije za več izbranih živali.	Nova lokacija iz šifranta, datum ustrezen	Shranjeno v ≤ 2 s, brez napak

2.3.11.6.5 Diagram primerov uporabe

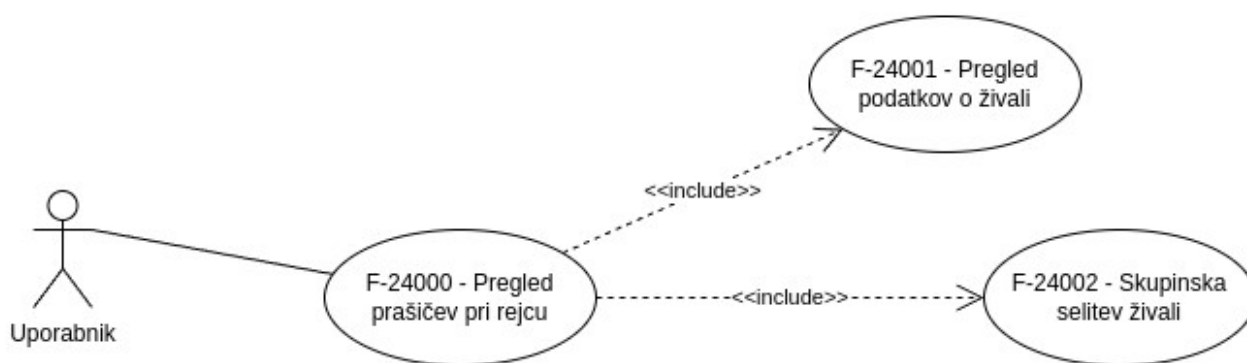


Diagram 87: Diagram primerov uporabe za SKL-PRAS-F-24000.

2.3.11.7 SKL-PRAS-F-24100: Masovni uvoz podatkov za prašiče

2.3.11.7.1 Opis funkcionalnosti

Funkcionalnost omogoča **centralizirano in nadzorovano nalaganje podatkov** za prašiče, prek strukturiranih datotek (.csv, .txt, .xlsx), s podporo za več podatkovnih tokov: merilni podatki, plemenske vrednosti, dogodki (npr. prasiatve, abortusi), rodovni podatki, parametri NPV (npr. mesnatost, dolžina, seski).

Modul vključuje podporo za:

- **izbor vrste nalaganja** (npr. meritve, NPV vrednosti, podatki o prasiatvah),
- **nalaganje in branje datotek** (z možnostjo določitve ločila),
- **preslikavo stolpcev** (data/table mapping),
- **sledenje napredku uvoza** (4 faze: izvor → mapping → validacija → zapis),
- **dokumentacijo nalagalnega postopka** (navodila, povezane .docx datoteke).

Na voljo je pregledni seznam vseh podprtih postopkov z neposredno povezavo do pripadajoče dokumentacije (npr. *proces_LOAD_seme_v1.docx*, *proces_LOAD_mensatost.docx*, *proces_LOAD_prasitev_v1.docx*, ipd.).

2.3.11.7.2 Predpogoji

- Uporabnik mora imeti vlogo z dostopom do nalagalnega modula.
- Izbrana datoteka mora ustrezati strukturi pričakovane vrste uvoza.
- Konfiguracija za mapping mora biti vnaprej definirana.

- Vsaj ena ciljna tabela mora biti vključena v mapping.

2.3.11.7.3 Merila uspešnosti

- Napake pri validaciji se jasno prikažejo po vrsticah in atributih.
- Uvoz večjega obsega (npr. 10.000 vrstic) se zaključi v ≤ 30 s.
- Vsak uvoz se revizijsko beleži (čas, uporabnik, tip datoteke, rezultat).

2.3.11.7.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
~24100	Masovni uvoz podatkov	Nalaganje PV podatkov preko .csv datotek, možnost merge ali nadomestitve.	Konfiguriran mapping, datoteka ustrezne oblike	≤ 5 s do validacije, 1 s/1000 zapisov
~24101	Izbor tipa uvoza	Uporabnik naloži datoteko in izbere tip uvoza (kaj uvaža).	Validna struktura, ločena polja	Validacija in zapis z ločenim izpisom napak
~24102	Konfiguracija uvoza	Data/table mapping z možnostjo ignoriranja določenih stolpcv.	Polja pravilno preslikana	Vsaj 1 ciljna tabela, vsi obvezni atributi
~24103	Pregled napredka	Vizualni prikaz faz uvoza z možnostjo prekinitve ali ponovnega zagona.	Uvoz v teku	Napredek se osvežuje sproti

2.3.11.7.5 Diagram primerov uporabe

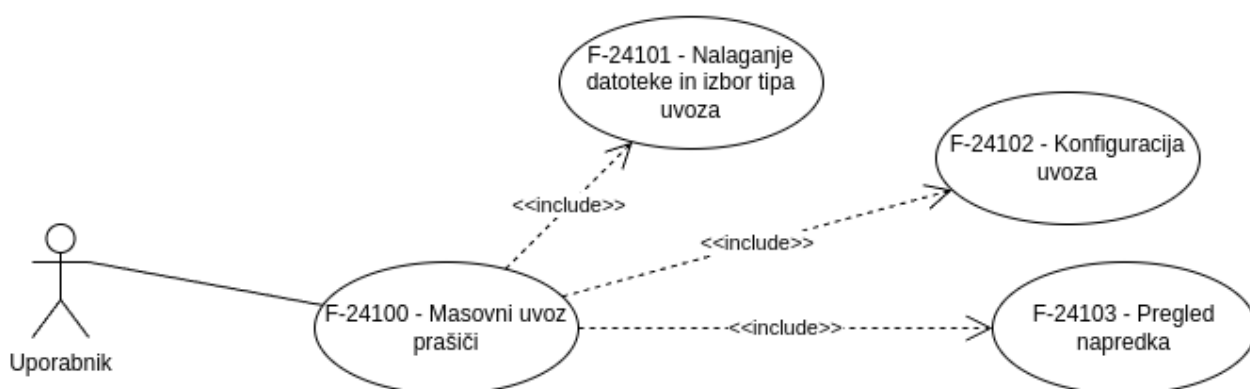


Diagram 88: Diagram primerov uporabe za SKL-PRAS-F-24100.

2.3.12 SKL-Kopit: Sklop kopitarji

Modul za področje kopitarjev (SKL-KOP) v okviru informacijskega sistema POKK pokriva funkcionalnosti v zvezi z identifikacijo, rodovniškim sledenjem, upravljanjem dogodkov in izdajo dokumentov za konje in druge kopitarje. Glede na obstoječe nacionalne evidence in zahteve mednarodnih organizacij mora modul zagotoviti podporo za izvajanje zakonsko predpisanih nalog ter strokovno upravljanje populacijskih podatkov.

Podrobna vsebinska in funkcionalna specifikacija tega modula je pripravljena v ločenem dokumentu, ki je vključen kot **priloga tehnični specifikaciji** (glej **poglavje 11 – Priloge in reference**). V tej specifikaciji so zajete funkcionalnosti v skladu z veljavno zakonodajo, zahtevami rejskih organizacij in vmesniki do obstoječih sistemov (npr. CRK/VOLOS).

Ne glede na dejstvo, da je tehnična vsebina ločeno dokumentirana, mora biti izvedba modula SKL-KOP **v celoti usklajena z arhitekturo sistema POKK**, zlasti v naslednjih vidikih:

- **integracija z osrednjimi horizontalnimi komponentami** (UAC, revizijska sled, šifranti, komunikacijski modul),
- **enotna uporabniška izkušnja** preko skupnega grafičnega in avtorizacijskega sloja,
- **samostojna razvojna in namestitvena pot**, skladno z načeli modularnosti iz metodologije GTZ 2.3,
- podpora za **CI/CD, vsebniško izvedbo (Docker/Kubernetes)** in gostovanje v infrastrukturi DRO.

Za koordinacijo vsebin, usklajevanje prioritet in strokovno validacijo modula je pristojna **Veterinarska fakulteta (UL VF)**, ki vodi razvojne aktivnosti v sodelovanju z ostalimi člani **skupnega projektnega odbora**. Vključitev v produkcijsko okolje je možna le ob predhodni validaciji usklajenosti z arhitekturnimi, podatkovnimi in varnostnimi zahtevami sistema POKK.

2.3.13 eGovedo

Modul za področje govedoreje (**eGovedo**) predstavlja samostojen funkcionalni sklop v okviru platforme **eŽivinoreja**, ki ga razvija KIS.

2.3.14 Ečebele

Modul za področje čebelarstva (eČebele) predstavlja samostojen funkcionalni sklop v okviru platforme **eŽivinoreja**, ki ga je razvil KIS.

2.3.15 SKL-GB: Sklop genska banka

Modul za gensko banko (SKL-GB) predstavlja specializiran funkcionalni sklop znotraj platforme **eŽivinoreja**, namenjen upravljanju podatkov o ohranjanju genskih virov domačih živali. Modul je razvit v okviru obstoječe platforme **eŽivinoreja**, zato je v tej fazi projekta ključno zagotoviti njegovo **celovito integracijo** v arhitekturo sistema **eŽivinoreja** ob **ohranitvi skladnosti, funkcionalnosti in uporabniške izkušnje**.

Sklop SKL-GB mora biti usklajen s skupnim tehnološkim in varnostnim modelom platforme, pri čemer se obstoječa vsebina **ne podvaja**, temveč se tehnično in podatkovno poveže z razvitimi horizontalnimi komponentami (npr. modul za prijavo in pooblastila, šifranti, komunikacijski sloj, revizijska sled). Uporabniki morajo do podatkov genske banke dostopati prek enotnega sistema, v okviru vnaprej določenih avtorizacijskih pravil in zastopniških razmerij.

Modul mora biti arhitekturno neodvisen in razvit na način, ki omogoča **samostojno CI/CD uvedbo, nadgradnjo in produkcijsko uvedbo**, brez neposrednega vpliva na druge funkcionalne sklope. Tako kot preostali deli sistema mora biti zasnovan za **vsebniško okolje (Docker/Kubernetes)** ter pripravljeno na gostovanje v infrastrukturi Državnega računalniškega oblaka (DRO).

Za vsebinsko ustreznost in nadaljnji razvoj sklopa je odgovorna **Univerza v Ljubljani** –

Biotehniška fakulteta (UL BF), ki sodeluje v okviru **skupnega projektnega odbora**. Vse tehnične rešitve morajo biti pred produkcijsko vključitvijo **usklajene s platformo eŽivinoreja**, validirane z vidika varnosti, revizijske sledljivosti in interoperabilnosti.

2.3.16 SKL-Docs: Sklop dokumenti

2.3.16.1 SKL-DOCS-F-30000: Upravljanje predlog

2.3.16.1.1 Opis funkcionalnosti

Funkcionalnost omogoča ustrezno avtoriziranemu uporabniku (npr. administratorju ali skrbniku sistema), da v sistem naloži predloge dokumentov v formatih DOCX ali PDF. Predloge so namenjene kasnejšemu generiranju dokumentov s pomočjo sistemskih podatkov. Ob nalaganju predloge uporabnik določi ustrezne metapodatke, kot so naslov, opis, veljavnost in status predloge.

Če uporabnik predlogo nadomesti z novo različico, se avtomatsko ustvari nova verzija, pri čemer sistem beleži celotno zgodovino sprememb.

V okviru celotnega sistema je predvidena uporaba približno **120** kompleksnejših dokumentov, ki vključujejo tudi **programske ključnike**. Ti zahtevajo namensko programsko implementacijo na ravni sistema (npr. vezava na podatkovne poizvedbe ali proceduro).

Uporabniški ključniki (npr. vnos kratkega ali dolgega besedila, tabele, konstante) so popolnoma dinamični in v celoti v domeni uporabniške konfiguracije — njihovo uporabo, imena, tipe in omejitve upravlja administrator preko uporabniškega vmesnika.

Glavne značilnosti:

- Naložljivost predlog v formatih DOCX in PDF
- Določitev metapodatkov: naziv, opis, datum veljavnosti, status
- Možnost zamenjave obstoječe predloge z novo verzijo
- Verzijsko beleženje vseh sprememb
- Avtorizacija: funkcionalnost je na voljo izključno za določene uporabniške vloge (npr. admin)

2.3.16.1.2 Predpogoji

- Uporabnik mora imeti ustrezno vlogo
- Vzpostavljen modul za upravljanje dokumentov
- Sistem za verzijsko sledenje mora biti omogočen

2.3.16.1.3 Merila uspešnosti

- Predloga je uspešno naložena in vidna v seznamu predlog
- Ob spremembi se ustvari nova verzija z ustreznimi metapodatki
- Zgodovina vseh sprememb ostaja sledljiva
- Nedovoljeni uporabniki ne morejo urejati predlog

2.3.16.1.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
30001	Naložitev predloge	Administrator naloži novo predlogo z vnosom metapodatkov	Avtoriziran uporabnik	Predloga je shranjena in vidna v seznamu
30002	Upravljanje metapodatkov	Uporabnik ureja naziv, opis in veljavnostne attribute	Uporabniški vmesnik omogoča	Spremembe so takoj vidne in shranjene

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
30003	Verzija predloge	predloge Ob spremembi sistema se shrani nova verzija predloge z zgodovino	vnos Verzijsko sledenje omogočeno	Nova verzija je zabeležena in prejšnje verzije ostanejo

2.3.16.1.5 Diagram primerov uporabe

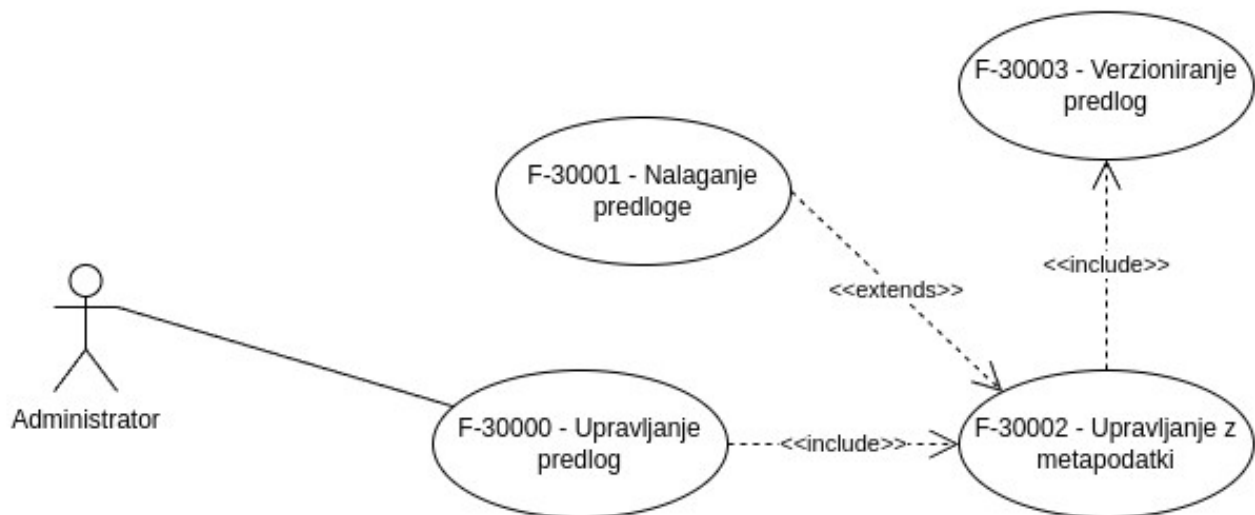


Diagram 89: Diagram primera uporabe SKL-DOCS-F-30000: Upravljanje predlog

2.3.16.2 SKL-DOCS-F-30100: Upravljanje ključnikov

2.3.16.2.1 Opis funkcionalnosti

Funkcionalnost omogoča ustrezno avtoriziranemu uporabniku urejanje nabora razpoložljivih ključnikov, ki jih sistem prepozna in nadomesti z vsebino pri generiranju dokumentov. Ključniki se uporabljajo znotraj predlog dokumentov (DOCX/PDF), in so klasificirani glede na izvor in način pridobivanja podatkov.

Ključniki so razdeljeni v dve skupini:

1. Ročno definirani ključniki

Ti se v celoti vzpostavijo in upravljajo znotraj uporabniškega vmesnika:

- KRATEK TEKST
- DOLG TEXT
- TABELA
- KONSTANTA

Za KRATEK TEKST, DOLG TEXT in TABELA uporabnik določi vrsto vsebine:

- besedilo, število, datum, datumski razpon, seznam ipd.

Za KONSTANTA uporabnik vnese fiksno vrednost, ki se vedno uporabi nespremenjena (npr. "Zavod za...", "Projekt št. XYZ").

2. Programski ključniki

Ti so vezani na sistemsko implementacijo (npr. shranjene procedure, SQL-pogledi):

- PROGRAMSKI KRATEK TEXT
- PROGRAMSKI DOLG TEXT
- PROGRAMSKA TABELA

Ti ključniki se **ne urejajo neposredno**, temveč se njihov izvor implementira na ravni SUPB. Uporabnik jih lahko:

- **doda** (z registracijo sistema imena)
- **odstrani**, če niso več v rabi
- ne more spreminjati njihove notranje logike

Glavne značilnosti:

- Podpora za vse tipe ključnikov, vključno s KONSTANTA
- Določitev vsebinskega tipa in opisa za ročne ključnike
- Seznam programskih ključnikov, vezanih na standardizirane sistemske poizvedbe
- Označevanje ključnikov kot aktivnih/neaktivnih
- Samodejna validacija imen in omejitev

2.3.16.2.2 Predpogoji

- Uporabnik mora imeti ustrezno vlogo
- Vzpostavljen mehanizem za hranjenje in interpretacijo ključnikov
- Sistem za validacijo duplikatov in rezerviranih imen

2.3.16.2.3 Merila uspešnosti

- Ključnik pravilno registriran in viden v administraciji
- Ključnik ni uporaben, če je označen kot neaktiven
- Programski ključnik zahteva aktivno vezavo na backend vir (implementacijsko preverjanje)
- Konstantna vrednost se prikaže enako v vseh primerih uporabe

2.3.16.2.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
30101	Dodajanje ročnega ključnika	Dodajanje novega ključnika s tipom vsebine (tekst, število, datum...)	Avtoriziran uporabnik	Ključnik uspešno shranjen
30102	Urejanje ročnega ključnika	Spreminjanje metapodatkov obstoječega ključnika	Ključnik ni programskega tipa	Spremembe so pravilno zabeležene
30103	Brisanje ročnega ključnika	Mehko brisanje (označevanje kot neaktiven)	Ključnik ni več v rabi	Ključnik neuporaben za nove predloge
30104	Dodajanje programskega ključnika	Vnos systemskega imena za programsko logiko	Backend podpora vzpostavljena	Ključnik registriran
30105	Brisanje programskega ključnika	Odstranitev definicije programskega ključnika (če ni več v rabi)	Ni aktivne vezave	Ključnik odstranjen
30106	Registracija konstantnega ključnika	Uporabnik vnese nespremenljivo vrednost za nadomestitev	Avtoriziran uporabnik	Vrednost se ustrezno uporabi ob generiranju
30107	Pregled vseh ključnikov	Vmesnik za pregled, iskanje, filtriranje in izvoz nabora ključnikov	Uporabnik s pravico do vpogleda	Podatki prikazani v ≤ 1 s

2.3.16.2.5

Diagram primerov uporabe

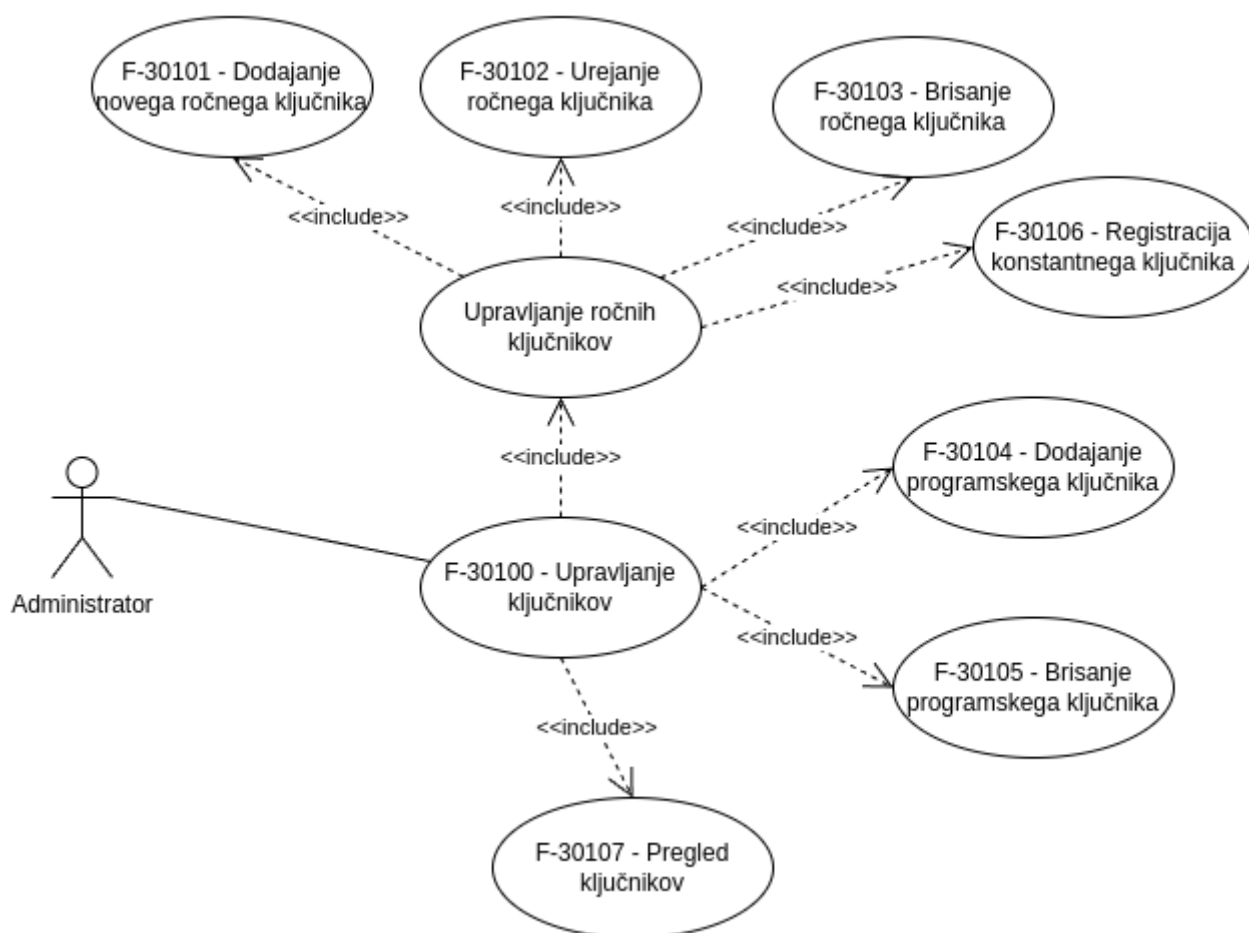


Diagram 90: Diagram primerov uporabe SKL-DOCS-F-30100: Upravljanje ključnikov

2.3.16.3 SKL-DOCS-F-30200: Generiranje dokumentov

2.3.16.3.1 Opis funkcionalnosti

Funkcionalnost omogoča prijavljenemu uporabniku generiranje dokumentov na osnovi predlog, ki so jih predhodno definirali skrbniki sistema. Dokumenti, ki jih je mogoče generirati, so vezani na določene kontekste in se filtrirajo glede na metapodatke (npr. tip postopka, tip subjekta, status postopka, identiteta uporabnika ipd.).

Uporabnik najprej izbere tip dokumenta, ki ga želi generirati. Sistem nato:

1. **Preišče izbrano predlogo** za vse prisotne ključnike.
2. **Za ročne ključnike** (KRATEK TEKST, DOLG TEXT, TABELA, KONSTANTA) prikaže ustrezna vnosna polja za interaktivni vnos.
3. **Za programske ključnike** (PROGRAMSKI *) sistem avtomatsko izvede vezane poizvedbe ali programske funkcije.
4. Po potrditvi uporabnika dokument generira – vse ključnike nadomesti z ustreznimi vrednostmi.

Dokument se ustvari kot nova instanca (DOCX in/ali PDF) in je uporabniku na voljo za prenos.

Glavne značilnosti:

- Kontekstna izbira razpoložljivih dokumentov na podlagi metapodatkov
- Samodejno razpoznavanje vseh ključnikov znotraj izbrane predloge
- Prikaz dinamičnih vnosnih polj za ključnike, ki zahtevajo ročni vnos
- Samodejna izvedba programskih ključnikov
- Zamenjava vseh ključnikov z ustreznimi vrednostmi (ročne in programske)
- Prenos končnega dokumenta s strani uporabnika
- Možnost shranjevanja generiranega dokumenta v sistem (če to zahteva proces)

2.3.16.3.2 Predpogoji

- Uporabnik mora biti prijavljen in imeti pravico do uporabe dokumentnega modula
- Na voljo mora biti vsaj ena ustrezna predloga z aktivnimi ključniki
- Programski ključniki morajo imeti aktivno vezavo na implementacijsko logiko
- Modul SKL-DOCS mora biti inicializiran z metapodatkovnim iskalnikom

2.3.16.3.3 Merila uspešnosti

- Seznam filtriranih predlog se prikaže v ≤ 2 s
- Ključniki se pravilno razpoznajo in ločijo po tipu
- Dokument je generiran v ≤ 3 s od potrditve vseh podatkov
- Generiran dokument vsebuje vse zahtevane podatke (ni neizpolnjenih ključnikov)
- Dokument je dostopen za prenos s strani uporabnika

2.3.16.3.4 Tabela funkcionalnosti

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
30201	Filtriranje razpoložljivih predlog	Uporabniku se prikažejo razpoložljive predloge glede na kontekst in metapodatke	Prijavljen uporabnik, aktivna seja	Prikaz ustreznega seznama v ≤ 2 s

ID	Naziv	Opis	Predpogoji	Merila uspešnosti
30202	Analiza ključnikov v predlogi	Sistem pregleda izbrano predlogo in razpozna vse prisotne ključnike	Aktivna predloga	Ključniki uspešno identificirani in razvrščeni
30203	Prikaz vnosnih polj za ročne ključnike	Za vsakega ročnega ključnika se prikaže primerno vnosno polje	Definirana struktura ključnikov	Vnosni obrazec ustrezno prikazan
30204	Zajem vrednosti iz programskih ključnikov	Sistem avtomatsko izvede vezane poizvedbe za programske ključnike	Programski ključniki aktivni	Vrednosti pravilno pridobljene iz sistema
30205	Generiranje dokumenta	Ustvarjen končni dokument z zamenjanimi vrednostmi ključnikov	Vsi ključniki uspešno izpolnjeni	Dokument ustvarjen v ≤ 3 s, brez neizpolnjenih ključnikov
30206	Prenos dokumenta	Uporabnik lahko prenese dokument (DOCX, PDF)	Dokument generiran	Prenos možen neposredno v vmesniku ali preko povezave

2.3.16.3.5 Diagram primerov uporabe

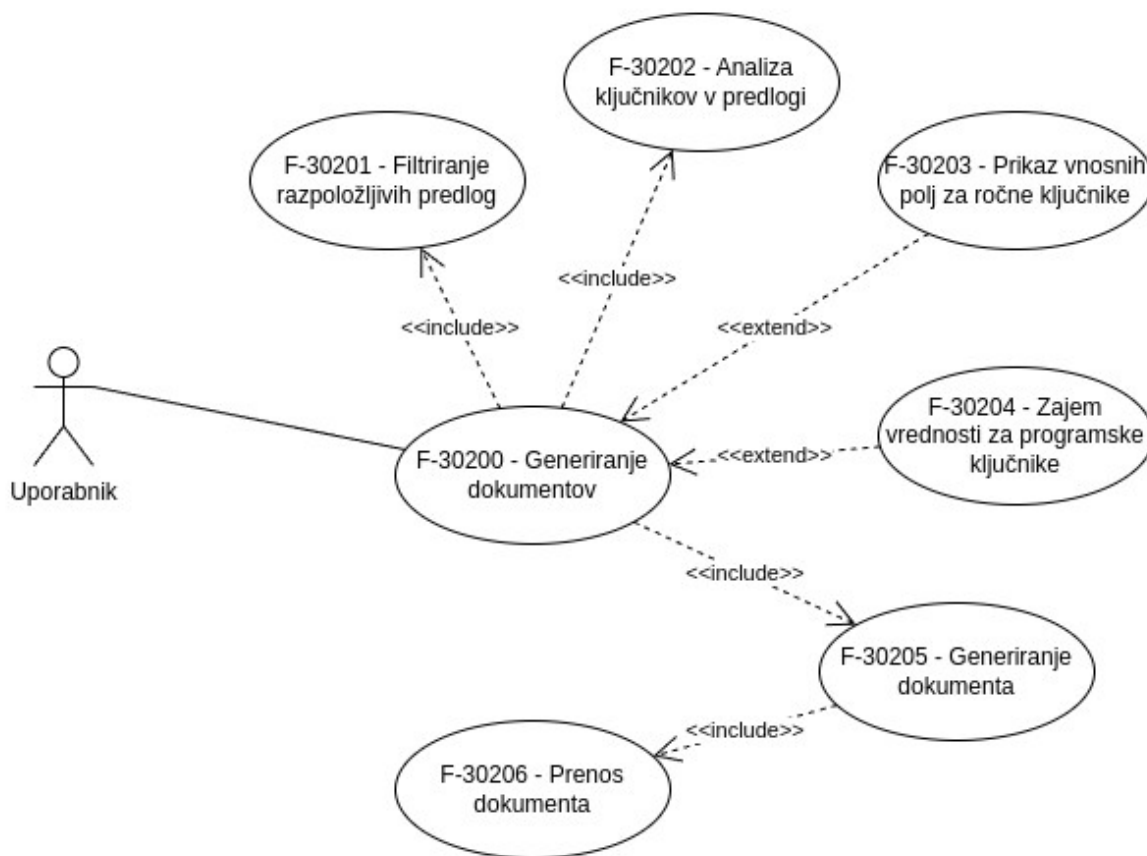


Diagram 91: Diagram primera uporabnosti SKL-DOCS-F-30200: Generiranje dokumentov

3 Arhitektura informacijske rešitve

3.1 Splošna arhitekturna zasnova

Informacijski sistem **POKK** temelji na **modularni, večslojni in dogodkovno vodeni arhitekturi**, ki je implementirana s pomočjo mikroservisnih principov in sodobnega CI/CD pristopa. Arhitektura zagotavlja **visoko ločljivost odgovornosti, neodvisnost modulov, enostavno horizontalno skaliranje** ter podporo za dolgoročno vzdrževanje in širitev sistema.

3.1.1 Logični sloji arhitekture

Uporabniški sloj (Presentation Layer) predstavlja spletno aplikacijo (POKK Spletno Aplikacijo), dostopno preko DMZ (reverse proxy) in spletno aplikacijo centralne avtentikacije (Federated Id). Vsebuje API končne točke, ki pokrivajo posamezne domene (ZivalAPI, PremikiAPI ipd.).

API sloj predstavlja vstopno točko v poslovno logiko sistema. Vsak API je ločen, verzioniran in neodvisen ter omogoča granularen nadzor nad dostopom ter validacijo vhodnih podatkov. API-ji so integrirani preko *OpenAPI* specifikacij. Sloj se dodatno loči na:

- **API sloj (API sloj)**, ki vsebuje metode, ki sprejemajo API klice.
- **Sloj poslovne logike (Business Logic Layer, BL)**, ki vsebuje poslovna pravila, mehanizme za usmerjanje podatkov, preverjanje pravic, pripravo vhodno/izhodnih podatkov, ipd.
- **Dostop do podatkov (Data Access Layer, DAL)**, ki je izoliran znotraj vsake domene in omogoča jasno sledljivost, omejen obseg delovanja ter enoten pristop k branju in pisanju. Podatkovni dostop poteka preko »*proxy shem*«, ki vsebujejo *sinonime in pravice do ustreznih baznih procedur ali pogledov*.

Vsi paketi uporabljajo skupen mehanizem za prijavo aktivnosti in napak (*BL Logging*).

Podatkovni sloj (Data Layer) temelji na relacijski ali NoSQL bazi, kjer so podatki organizirani v shemah: **POKK_VARNOSTNA_SHEMA** (Podatki o identitetah, pravicah, zastopništvih in nadzor nad uporabniškim računom (**User Account Control**)) in **POKK_PODATKOVNA_SHEMA** (podatki, ki jih sistem zbira). Vsa komunikacija poteka preko t.i. Proxy pogledov, z uporabo shranjenih procedur in pogledov. Vzporedno se vodi tudi **POKK_JOURNAL_SHEMA** za revizijsko sledenje sprememb nad tabelami (journal tabele, ki beležijo spremembe nad podatkovnimi tabelami).

3.1.2 Integracijska arhitektura in sinhronizacija

Sinhronizacijski mehanizem vključuje *sinhronizacijske servise*, ki skrbijo za asinhrono zajemanje podatkov (prek vnaprej definiranih API mehanizmov, ki delujejo po principu »PULL«). Vhodni podatki se pretvorijo v interni model in usklajujejo s poslovno logiko.

Pridobivanje podatkov iz zunanjih virov (npr. CRŽ– MKGP, UVHVVR) poteka centralizirano prek

podatkovnega skladišča (DW) na KIS, kamor se podatki najprej sinhronizirajo preko namenskih replikacijskih API točk. Posamezni podsistemi, vključno s POKK, dostopajo do teh podatkov iz DW, skladno s svojo pristojnostjo.

Dogodkovna arhitektura temelji na *Kafka* kanalih, ki omogočajo strukturiran zapis aktivnosti. Vsaka izmed komponent sistema na ustrezne *Kafka* kanale (topic-e) pošilja dnevniške zapise (loge).

3.1.3 Podatkovno skladišče in analitika

ETL mehanizem osvežuje **DW** (data warehouse), ki poganja **OLAP analitiko**. Rešitev omogoča poglobljene analize prek ločenih OLAP aplikacij.

Arhiviranje dokumentov se izvaja v datotečnem sistemu z možnostjo podvajanja v eksterni repozitorij (npr. Glacier). Dostop je omogočen preko *StorageAPI*, skupaj z vodenjem dnevnika aktivnosti.

OpenSearch skrbi za indeksiranje revizijskih in iskalnih podatkov, z možnostjo nadzora dostopa in kasnejše uporabe za poročila ali revizije. Sloj (surove) podatke pridobiva iz ustreznih *Kafka* kanalov (topic-ov) in jih prenaša v lokalni pomnilnik za obdelavo in hrambo.

3.1.4 Tehnološke osnove

Programski jezik: Jeziki skladni z Podprta-programaska-oprema-DRO_V1.0.

Baza podatkov: ACID oz. BASE compliant podatkovna baza.

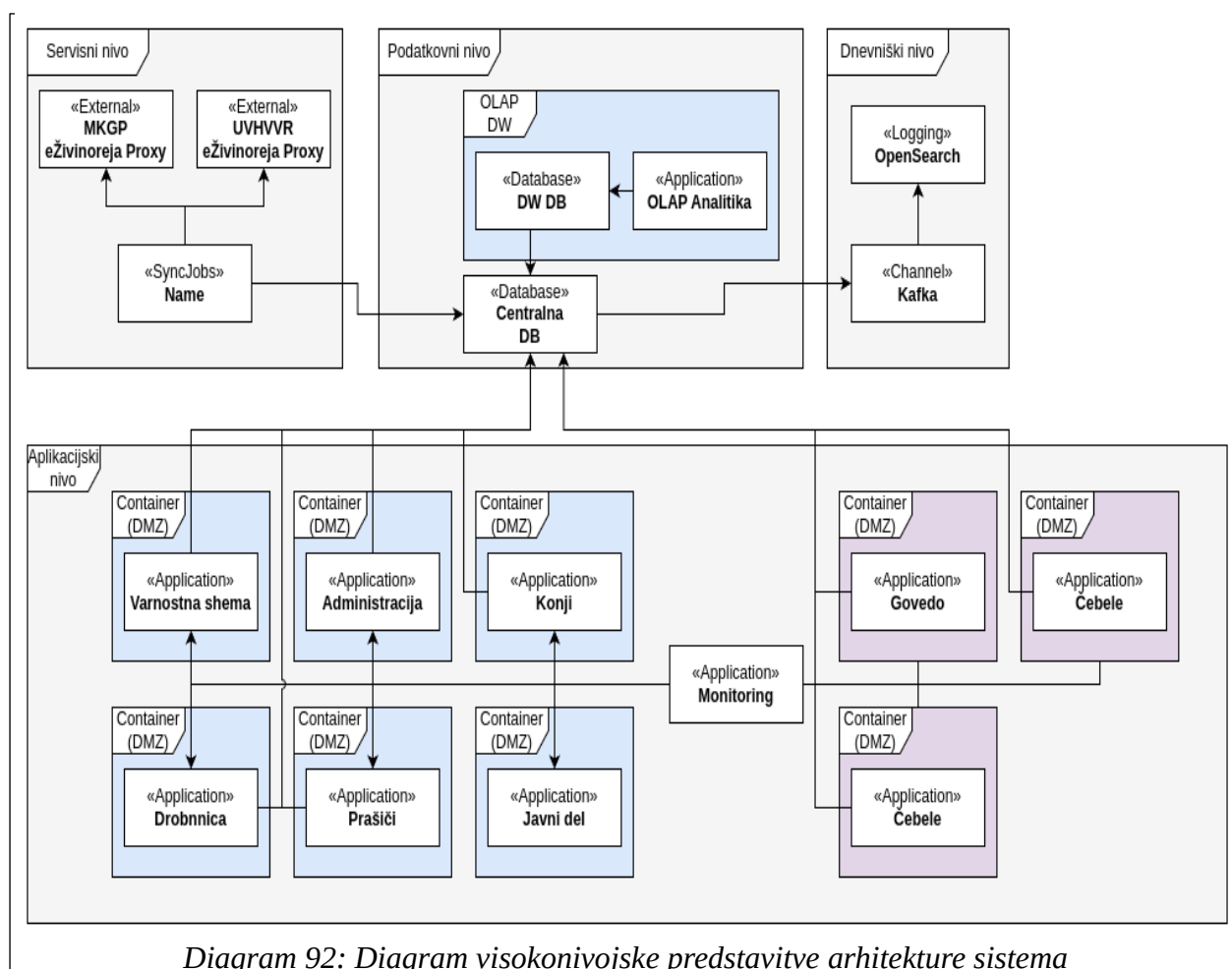
Komunikacija: REST API z **OpenAPI**, **Kafka** za dogodke.

CI/CD okolje: **GitLab**, **Jenkins** ali ekvivalentna orodja; **testna okolja z realističnimi obremenitvami**.

Monitoring: **Prometheus** ali **Grafana** oziroma ekvivalentno odprtokodno okolje.

Sledenje in analiza aktivnosti: **OpenSearch**.

3.2 Diagrami arhitekture



Informacijski sistem lahko v grobem razdelimo na ključne komponente, kot so prikazane na zgornjem diagramu. Aplikacijski API-ji so praviloma nameščeni v **Docker kontejnerjih**, ki delujejo v orkestriranem okolju.

Sinhronizacijski servisi se sprožajo časovno načrtovano (npr. s pomočjo *cron* mehanizmov) ter skrbijo za periodično polnjenje **OLAP podatkovne baze** za potrebe podatkovnega skladišča. Ta baza je ločena v lastno shemo, kar preprečuje t. i. *deadlock* pogoje v osrednji transakcijski podatkovni zbirki.

Blokiranje operacij je omejeno z uporabo asinhronega prenosa prek **Kafka kanalov**, medtem ko se za iskanje in analitiko uporablja odprtokodna rešitev **OpenSearch**, ki se polni neodvisno od osnovne podatkovne baze.

Datoteke (dokumentacija, poročila, izvozi) so shranjene tako v **lokalnem datotečnem sistemu** kot tudi v **oblačni shrambi** (npr. Amazon Glacier) za potrebe dolgoročnega arhiviranja.

Za **monitoring aplikacij** in aplikacijskih vmesnikov se uporablja namenska programska oprema (npr. *Prometheus*, *Grafana*), ki omogoča spremljanje tako **aplikacijskih končnih točk (endpoints)** kot tudi **namenskih aplikacijskih modulov** in njihovih obremenitev.

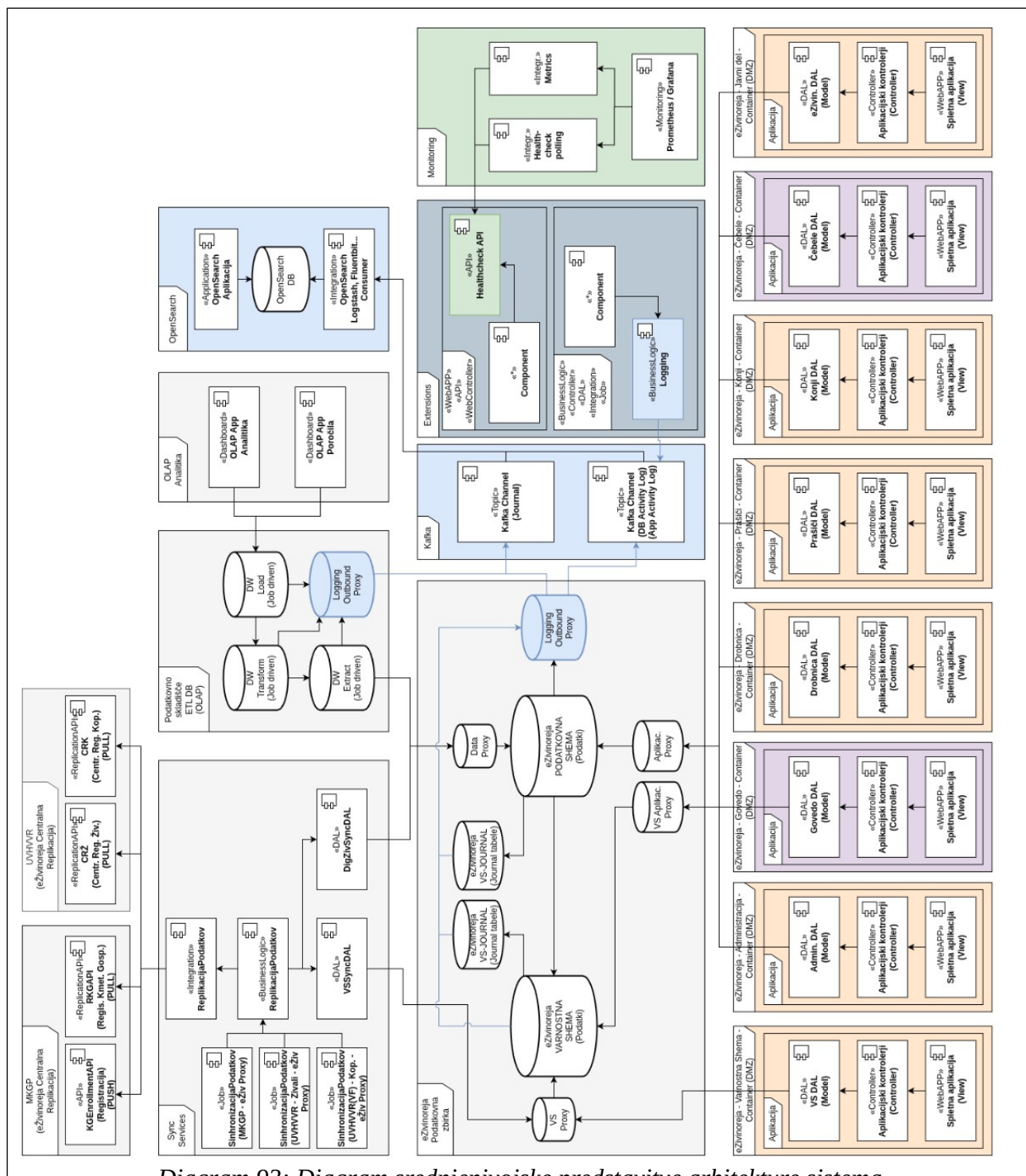


Diagram 93: Diagram srednjenivojske predstavitve arhitekture sistema

Srednjenivojska arhitektura informacijskega sistema **POKK** prikazuje razčlenjeno kompozicijo ključnih komponent sistema, vključno z logičnimi povezavami med moduli, sinhronizacijskimi mehanizmi, podatkovnimi sloji, analitiko, dokumentnim arhivom in zalednimi storitvami.

Jedro rešitve predstavlja **centralna podatkovna baza**, razdeljena na več logično ločenih shem, ki omogočajo strogo ločitev podatkov glede na namen in občutljivost:

- **VARNOSTNA shema** vsebuje osebne in varovane podatke ter je dostopna le omejenim komponentam preko definirane varnostne sheme. Ta shema se uporablja za potrebe avtentikacije in avtorizacije.
- **PODATKOVNA shema** hrani vsebinske podatke za posamezne funkcionalne module (npr. živali, premiki, plodnost, prireja ipd.) in deluje kot osrednja operativna zbirka.
- **JOURNAL shema** vsebuje podatke revizijskega sledenja in skrbi za zapis aktivnosti v žurnalne tabele.

Za dostop do podatkovnih zbirk se uporabljajo t.i. **proxy shema**, ki vsebujejo definirane sinonime do baznih pogledov (view) in paketov procedur (stored procedures), s čimer je zagotovljena transakcijska konsistentnost, nadzor nad spremembami in sledljivost operacij.

Na centralno zbirko se navezuje **podatkovno skladišče (DW)**, ki uporablja **ETL** pristop, razdeljen v tri funkcionalne sheme:

1. **EXTRACT** – izvaja izvleček podatkov iz operativnega sistema.
2. **TRANSFORM** – izvaja transformacije, filtriranje in agregacije.
3. **LOAD** – vsebuje končne strukturirane podatke za poročila in analitiko.

Te sheme omogočajo razbremenitev transakcijskega dela sistema in so zasnovane za asinhrono obdelavo, s čimer se preprečijo morebitni *deadlock* pogoji v osrednji podatkovni zbirki.

OLAP modul vključuje ločene aplikacije za podatkovno analitiko, pri čemer se shema **LOAD** uporablja kot vir za **BI poročila** (npr. *JasperReports*, *Crystal Reports*) in namenske analitične prikaze.

Vsi zapisi v dnevnik (revizijska sled) potekajo asinhrono in so spremljani z uporabo **Kafka kanalov**, ki omogočajo porazdeljeno obdelavo zapisov in kasnejše indeksiranje preko sistema **OpenSearch**.

Sinhronizacija z zalednimi sistemi poteka preko *ReplicationAPI*, ki omogoča integracijo s pomočjo standardiziranega **PULL mehanizma**, kjer institucije preko svojega *IntegrationAPI* periodično prevzemajo spremenjene zapise. Vsa sinhronizacija temelji na časovnih žigih sprememb, kar omogoča deterministično usklajevanje in sledenje.

POKK Sync Container, nameščen v DMZ, deluje kot replikacijski modul, ki z uporabo DAL komponent omogoča varen in nadzorovan dostop do ustreznih podatkov za replikacijo (sinhronizacijo) podatkov z preostalimi sistemi.

Uporabniški sloj strogo ločuje med spletnimi aplikacijami, ki podpirajo federativno prijavo (lahko preko SSO, piškotkov ali podobno), ki se izvrši preko namenske aplikacije v varnosnti shemi. Uporabnik ob prijavi izbere svojo **reprezentančno identiteto in vlogo**, kar je osnova za natančno dodeljevanje pravic in dostopov v sistemu.

Celotni sistem je pod nadzorom **monitoring modula**, ki z uporabo orodij (*Prometheus, Grafana,..*) periodično preverja:

- **zdravje aplikacijskih API točk (health-checks),**
- **zmogljivosti in metrike posameznih aplikacijskih komponent.**

Monitoring omogoča hitro odkrivanje anomalij, podporo za avtomatizirana opozorila in pripravo tehničnih poročil.

3.3 Povezljivost in integracije

Podsistemi POKK so zasnovani tako, da uporabljajo skupno podatkovno skladišče (DW) platforme eŽivinoreja kot osrednjo točko za integracijo zunanjih virov. Zunanji podatkovni sistemi (npr. veterinarske evidence, laboratoriji, registri) se najprej povežejo z DW prek varnih in standardiziranih integracijskih mehanizmov (npr. REST API, ETL procesi). Iz DW se nato ustrezni podatki distribuirajo v posamezne podsisteme, vključno s POKK, glede na funkcionalne potrebe, pravice dostopa in časovne zahteve. S tem je zagotovljena konsistentnost, sledljivost in ponovna uporabnost podatkov znotraj celotne platforme eŽivinoreja.

Vzpostavljene povezave vključujejo:

- **MKGP:**
 - **RGK (Register kmetijskih gospodarstev):** dostop preko REST API, asinhroni **PULL** mehanizem.
 - **KGenrollment (registracija kmetij v programe):** povezava preko REST API, asinhroni **PUSH** mehanizem (zunanji sistem potiska podatke v POKK).
- **UVHVVR:**
 - **CRŽ (Centralni register živali):** REST API, asinhroni **PULL** mehanizem.
 - **CRK (Centralni register kopitarjev):** REST API, asinhroni **PULL** mehanizem.

3.3.1 Tehnične značilnosti integracij

- **Protokol:** Vse povezave uporabljajo REST API z dokumentacijo skladno z OpenAPI specifikacijo (v. 3.0).
- **Varnost:** Vsi klici potekajo prek TLS 1.3. Avtentikacija se izvaja preko OAuth 2.0 ali podpisanih JWT žetonov.
- **Sledljivost:** Vse komunikacije in sinhronizacijski dogodki so zabeleženi prek Kafka kanalov ter sistemov za centralizirano beleženje logov (npr. OpenSearch).
- **Sinhronizacija:**
 - **PULL** povezave se izvajajo periodično (npr. vsakih 24 ur) na podlagi časovnih žigov sprememb.
 - **PUSH** povezave omogočajo neposredno obveščanje POKK sistema ob ustreznih dogodkih (registracija v program).

3.3.2 Formati izmenjave podatkov

Kot osnovni format se uporablja JSON oziroma XML, CSV za ročni uvoz/izvoz, PDF za generirana poročila.

S tem se zagotavlja popolna interoperabilnost sistema POKK z vsemi ključnimi informacijskimi sistemi v živinoreji, skladno z nacionalnimi zakonodajnimi zahtevami in tehnološkimi smernicami

projekta.

3.4 Uporabljene tehnologije

Sistem omogoča uporabnikom z ustreznimi pravicami dostop do lastnih podatkov prek namenskih aplikacij, razvitih za posamezna funkcijska področja v živinoreji. Te aplikacije so zasnovane za gostovanje in izvajanje v ločenih vsebniških okoljih (Docker oziroma Kubernetes).

Format izmenjave podatkov: JSON je osnovni format, XML je omogočen tam, kjer je potreben (npr. izmenjava z EU sistemi), CSV za ročni uvoz/izvoz, PDF za generirana poročila.

S tem se zagotavlja popolna interoperabilnost sistema POKK z vsemi ključnimi informacijskimi sistemi v živinoreji, skladno z nacionalnimi zakonodajnimi zahtevami in tehnološkimi smernicami projekta.

3.4.1.1 3.4 Uporabljene tehnologije

Pri razvoju in vzdrževanju informacijskega sistema POKK se pričakuje dosledna skladnost z dokumentoma *Podprta-programska-oprema-DRO_V1.0* ter *Generične tehnološke zahteve GTZ 2.3*.

Primarna podatkovna zbirka mora temeljiti na eni izmed sprejetih podatkovnih baz.

Kjer je to smiselno in izvedljivo, naj se uporabljajo **odprtokodne tehnologije**. V kolikor se uporabijo lačljive tehnologije je skladno z GTZ 2.3 potrebno podati temeljito obrazložitev razlogov ter okoliščin, ki so botrovale odločitvi.

Za beleženje dogodkov in log zapisov se uporablja asinhron mehanizem prek **Kafka kanalov**, z nadaljnjo obdelavo in pregledovanjem teh zapisov preko **OpenSearch**.

Za nadzor delovanja sistema (t.i. health check) se uporablja ustrezna programska oprema, ki omogoča spremljanje stanja API-jev, obremenitev in drugih ključnih kazalnikov delovanja sistema.

V okviru navedenih omejitev glede skladnosti s tehnološkimi smernicami (GTZ 2.3) in predpisanimi tehnološkimi sklopi, je izvajalcu omogočen prost izbor konkretnih tehnologij, orodij in ogrodij, ki jih bo uporabil pri razvoju POKK. Izbor mora biti tehnično utemeljen in skladen z zahtevami za odprtost, vzdržljivost ter dolgoročno vzdrževanje ter povezljiv z že obstoječim informacijskim sistemom eŽivinoreja.

4 Tehnične zahteve in specifikacije

4.1 Tehnični standardi

Informacijski sistem POKK bo razvit in implementiran skladno z uveljavljenimi nacionalnimi in mednarodnimi standardi ter metodologijami, ki zagotavljajo visoko kakovost, varnost, interoperabilnost in dolgoročno vzdržnost rešitve. Osrednji okvir razvoja predstavlja metodologija **GTZ 2.3**, dopolnjena s tehničnimi in varnostnimi standardi, ki pokrivajo vse ključne vidike razvoja informacijskih sistemov.

Uporaba navedenih standardov omogoča strukturirano zasnovo, poenoteno komunikacijo med komponentami, transparentnost poslovnih pravil, revizijsko sledljivost in skladnost z zakonodajo.

4.1.1 Pregled uporabljenih standardov

ID	Standard/	Opis	Merila skladnosti
----	-----------	------	-------------------

	Smernica		
TS-01	GTZ 2.3 (Generične tehnološke zahteve)	Nacionalna metodologija za razvoj informacijskih rešitev. Določa arhitekturna pravila, strukturo dokumentacije, uporabo gradnikov, ločevanje slojev, CI/CD in testiranje.	Skladnost arhitekture, strukturiranost specifikacij, uvedba gradnikov, uporaba testnih okolij.
TS-02	Podprta programska oprema DRO V1.0	Določa dovoljena razvojna orodja, programske jezike, baze podatkov, CI/CD in spremljevalna orodja.	Uporaba zgolj podprtih tehnologij (npr. Java, PostgreSQL, Jenkins, GitLab).
TS-03	ISO/IEC 27002:2022	Mednarodni standard za upravljanje informacijske varnosti. Določa varnostne kontrole, politike, nadzore dostopa, šifriranje in odzivanje na incidente.	Uvedba varnostne politike, enkripcija podatkov, sledljivost dostopov, kontrolni postopki.
TS-04	OpenAPI 3.0	Standard za opis REST API-jev, omogoča avtomatizirano generiranje dokumentacije in validacijo API vmesnikov.	Vsi API-ji morajo biti opisani z OpenAPI specifikacijo in verzionirani.
TS-05	OWASP TOP 10	Priporočila za varno razvojno prakso in zaščito pred najpogostejšimi spletnimi ranljivostmi.	Pregled kode, izvajanje varnostnega testiranja, zaščita pred SQLi, XSS, CSRF itd.
TS-06	UML 2.x (Unified Modeling Language)	Standard za modeliranje poslovnih procesov in informacijskih arhitektur (use-case, sekvenčni, aktivnostni diagrami).	Uporaba UML notacije v tehnični dokumentaciji.
TS-07	JSON / XML / CSV	Standardi za strukturirano izmenjavo podatkov med komponentami in zunanjimi sistemi.	Validacija struktur, uporaba XSD / JSON schema, nadzor kompatibilnosti formatov.
TS-08	TLS 1.3	Kriptografski standard za varno komunikacijo preko interneta.	Šifriranje vseh prenosov s TLS 1.3, izključena podpora starejšim protokolom.
TS-09	ISO/IEC 25010:2011	Standard za model kakovosti programske opreme (funkcionalnost, zanesljivost, uporabnost, vzdržljivost, varnost...).	Definicija kvalitativnih meril, vključitev v strategijo testiranja in preverjanja kakovosti.
TS-10	ISO/IEC 12207:2017	Standard za življenjski cikel razvoja programske opreme – od analiz in specifikacij do vzdrževanja in nadgradnje.	Strukturiranje faz razvoja, uskladitev z dolgoročnimi zahtevami po modularnosti in nadgradnji.

4.1.2 Utemeljitev uporabe

Čeprav projekt POKK trenutno še ne spada pod neposredno upravljanje Direktorata za razvoj in digitalizacijo (DRO) ali enotno infrastrukturo državnih rešitev, se že v začetni fazi v celoti opira na metodologijo **GTZ 2.3** in **upoštevata priporočila iz smernice Podprta programska oprema DRO V1.0**.

Projekt se razvija kot del **javno financirane informacijske infrastrukture** z visokim družbenim pomenom, zato je dolgoročno pričakovano, da bo sistem bodisi:

- integriran v širšo državne digitalne storitve (npr. enotne identifikacije, e-uprave, podatkovna vozlišča),
- prenesen v vzdrževanje s strani DRO ali tehnično prevzet v eno od infrastrukturnih državnih platform.

Uporaba standardov GTZ in DRO zagotavlja:

- **skladnost z nacionalnimi razvojnimi politikami**, ki temeljijo na odprtosti, standardizaciji in transparentnosti,
- **prenosljivost rešitve** v druge institucionalne okvire (npr. DRO, MKGP, UVHVVR),
- **tehnično nevtralnost**, ki omogoča vključevanje več izvajalcev in fleksibilno modularno nadgradnjo sistema,
- **dolgoletno vzdržnost**, saj sistem sledi že uveljavljenim mehanizmom za testiranje, razmejevanje odgovornosti, sledljivost in informacijsko varnost.

Dopolnilni standardi (ISO/IEC 27002, 25010, 12207) nadgrajujejo GTZ metodologijo z vidika **varnosti, kakovosti in formalnega upravljanja življenjskega cikla**, medtem ko OpenAPI, UML in podatkovni formati (JSON, XML) omogočajo **interoperabilnost in standardizirano dokumentiranje**.

Uporaba teh standardov je temelj za:

- **enotno razumevanje specifikacij** med vsemi partnerji projekta (KIS, UL-BF, UL-VF),
- **konsistentno pripravo razpisne dokumentacije**,
- **zanesljivo oceno ponudb**,
- **in kontinuirano tehnično rast sistema**.

4.2 Zahteve glede izvajalnih okolij

Za podporo razvoju, testiranju in produkcijskemu delovanju informacijskega sistema POKK je potrebno zagotoviti ustrezno tehnično infrastrukturo in ločena izvajalna okolja, ki omogočajo varno, zanesljivo in nadzorovano delovanje. Okolja morajo biti med seboj ločena tako fizično kot logično, z jasno definiranimi pravili dostopa, nadzora različic, replikacije podatkov in spremljanja delovanja.

4.2.1 Osnovna okolja

Okolje	Namen	Zahteve in značilnosti
Razvojno (DEV)	Interno okolje za razvoj in integracijo posameznih gradnikov.	Vsebuje generatorje testnih podatkov, lokalne kopije baz, CI/CD pipeline za build/test, omogoča debugging, nima povezave do produkcijskih virov.
Testno (TEST)	Okolje za izvajanje testiranja s strani naročnikov in izvajalcev.	Polni se z anonimiziranimi realnimi zgodovinskimi podatki. Vključuje orodja za funkcionalno, obremenitveno in varnostno testiranje. Ločen dostop za naročnika.
Predprodukcijsko	Replika produkcijskega	Enaka konfiguracija kot produkcija, testne

Okolje	Namen	Zahteve in značilnosti
(RC – Release Candidate)	okolja za namene testiranja zadnjih verzij pred objavo.	verzije se namestijo ob vzpostavitvi pogojev za izločitev napak. Dostop strogo nadzorovan.
Produksijsko (PROD)	Dejansko delujoče okolje za končne uporabnike.	Visoka razpoložljivost (HA), vključuje varnostne preglede, redne revizije, sledljivost vseh sprememb. Možnost hot-fix popravkov pod pogoji spremembne kontrole.

4.2.2 Minimalne sistemske zahteve

Komponenta	Zahteve
Strežniki	Virtualizirano ali kontejnerizirano okolje (npr. Docker/Kubernetes), ločeno po okoljih, z možnostjo horizontalnega skaliranja.
Operacijski sistem	Linux distribucije, ki so podprte znotraj smernice Podprta-programska-oprema-DRO V1.0 (npr. Ubuntu LTS, CentOS Stream, RHEL, Debian).
Baze podatkov	Vsako okolje mora imeti lastno instanco baze.
Dostop in varnost	Vsako okolje mora imeti lastno shemo za avtorizacijo in ločen sistem revizijske sledi (OpenSearch/Kafka).
CI/CD	Vzpostavljen Git repository in orodja za CI/CD (npr. GitLab CI, Jenkins), s samodejnim nameščanjem v ustrezna okolja.
Monitoring in logiranje	Monitoring sistem (npr. Prometheus, Grafana), integriran z log strežnikom (OpenSearch); nadzoruje ključne metrike in API endpoint-e.

4.2.3 Posebne zahteve

Vsa testna okolja morajo biti opremljena z realističnimi zgodovinskimi podatki (vsaj 15 let oziroma več če obstajajo v digitalni obliki), ob ustrezni **anonimizaciji** in ločitvi osebnih podatkov.

Vsaka namestitev mora biti **popolnoma reproducibilna** z uporabo avtomatiziranih skript in konfiguracijskih profilov (npr. Helm charti, Ansible playbooks).

Dostop do posameznih okolij mora biti jasno določen glede na vloge: razvijalci, testna skupina, naročniki, administratorji.

Sistem mora podpirati **paralelno izvajanje modulov** različnih živalskih vrst brez konflikta med podatkovnimi shemami.

4.3 Zmogljivost in skalabilnost

Informacijski sistem POKK mora zagotavljati stabilno in odzivno delovanje tudi pri visokih obremenitvah ter omogočati enostavno prilagajanje povečanemu številu uporabnikov, obsegu podatkov in kompleksnosti obdelav. Sistem mora biti sposoben tako **horizontalnega** kot **vertikalnega skaliranja**, pri čemer mora biti arhitektura zasnovana modularno, z ločeno obdelavo po domenah (npr. vrste živali), komponentah in funkcionalnih enotah.

4.3.1 Ključni zmogljivostni kazalniki (KPI)

ID	Kazalnik	Ciljna vrednost	Merilna metoda
KP-01	Čas odziva za API klic (v 95. percentilu)	≤ 500 ms	APM (Application Performance Monitoring), npr. Prometheus, Grafana
KP-02	Največje število	≥ 2000	Obremenitveni test z uporabo

ID	Kazalnik	Ciljna vrednost	Merilna metoda
	vzporednih uporabnikov		JMeter, Gatling
KP-03	Obdelava podatkov (npr. ETL v DW)	< 1 ura za 30 let podatkov / dan	Poročila sistema in logi procesov
KP-04	Skalabilnost sinhronizacijskih mehanizmov	5x povečanje števila zapisov brez degradacije >20 %	Simulacija z umetnimi podatki
KP-05	Razpoložljivost sistema (uptime)	≥ 99.9 %	Monitoring uptime (Prometheus, UptimeRobot itd.)
KP-06	Čas restarta posameznega mikroservisa	< 5 sekund	Obnašanje v orkestriranem okolju (Docker/K8s health-check)

4.3.2 Načela skalabilnosti

Horizontalno skaliranje bo omogočeno z uporabo **kontejneriziranih mikroservisov** in orkestracije (npr. Kubernetes), kar omogoča dinamično prilagajanje števila instanc posamezne komponente.

Asinhrona obdelava prek Kafka kanalov preprečuje zastoje pri masovni obdelavi podatkov in ločuje kritične tokove (transakcijski vs. analitični del).

ETL procesi in OLAP baze bodo izvedeni ločeno od transakcijske baze, s čimer se preprečuje vpliv poročil na vsakodnevno uporabo sistema.

Podatki bodo razdeljeni po **shemah in domenah**, kar omogoča **ločeno skaliranje posameznih živalskih vrst** in postopno uvajanje novih podsistemov.

Caching mehanizmi (npr. Redis, ehcache) bodo uporabljeni za izboljšanje dostopa do pogosto uporabljenih šifrantov in poročil.

Vsak mikroservis bo vključeval **health-check endpoint**, kar omogoča avtomatizirano preusmerjanje prometa na zdrave instance.

4.3.3 Podpora obremenitvenemu testiranju

Sistem mora omogočati:

- Samodejno generiranje testnih podatkov za ključne entitete (živali, premiki, rejci, dogodki),
- Izvedbo **paralelnih testov** nad več podsistemi hkrati (govedo, drobnica, konji),
- Spremljanje ključnih metrik v realnem času,
- Pripravo **analitičnih poročil o zmogljivosti**, ki bodo vključeni v CI/CD pipeline.

4.4 Specifikacije integracij

Informacijski sistem POKK mora zagotoviti varno, zanesljivo in standardizirano integracijo z notranjimi in zunanji informacijskimi sistemi, s ciljem zagotavljanja interoperabilnosti, podatkovne konsistentnosti ter sprotne ali periodične sinhronizacije podatkov. Integracije morajo biti skladne s specifikacijami OpenAPI 3.0, uporabljati preverjene varnostne mehanizme (OAuth2, TLS 1.3) ter omogočati asinhrono ali sinhrono komunikacijo glede na naravo podatkovnega toka.

4.4.1 Vrste integracij

Tip integracije	Opis	Mehanizem
Zunanje sinhronizacije (PULL)	Periodični zajem podatkov iz zunanjih registrov (npr. CRŽ, CRK)	REST API + časovni žig sprememb
Zunanje sprožilne integracije (PUSH)	Sprejemanje podatkov s strani zunanjih sistemov ob dogodku (npr. registracija kmetije pri partnerju, vnos meritve, ...)	REST API z validacijo + podpisani JWT
Notranje medmodularne integracije	Komunikacija med mikroservisi znotraj sistema POKK (npr. Žival ↔ Drobnica)	Ločeni REST API-ji + Kafka topici za sledljivost
ETL integracije z DW	Periodični prenos podatkov v podatkovno skladišče	Batch ETL z logiranjem stanja in napak
Arhivski sistemi	Povezava z datotečnimi in oblačnimi shrambami za dolgoročno hrambo	StorageAPI (lokalno + Glacier/Artic Cold)
Poročilni in BI sistemi	Dostop do podatkov OLAP za potrebe poročanja	JDBC/REST + ločeni OLAP strežniki

4.4.2 Specifikacije komunikacije

Protokol	Uporaba	Podrobnosti
REST API (OpenAPI 3.0)	Osnovni protokol za komunikacijo med komponentami in z zunanjimi sistemi	Dokumentacija za vsako komponento, verzioniranje, podpora za JSON/XML
Kafka	Dogodkovna komunikacija in logiranje sprememb	Tematsko strukturirani "topic-i", asinhrono pošiljanje dogodkov
TLS 1.3	Šifriranje prenosa podatkov	Vključena v vse REST klice in komunikacijo z oblačnimi sistemi
OAuth 2.0 / JWT	Avtentikacija in avtorizacija API dostopov	Podpisani žetoni, kriptografsko potrjevanje identitete
JSON / XML / CSV / PDF	Formati za izmenjavo podatkov	JSON kot privzeti, XML za izmenjavo z EU sistemi, CSV, PDF

4.4.3 Primeri ključnih zunanjih integracij

Sistem	Nosilec	Funkcija	Način integracije
CRŽ Centralni register živali	UVHVVR	Podatki o identifikaciji in premikih živali	PULL (REST API + timestamp)
CRK / VOLOS Register kopitarjev	UVHVVR	Rodovniki, lastništvo, identifikacija	PULL (REST API + transformacija)
RGK Register kmetijskih gospodarstev	MKGP	Evidenca KG, podatki o subjektih	PULL (REST API)

4.4.4 Mehanizmi odpornosti in napak

Retry mehanizem pri napakah povezljivosti (nastavljivo število poskusov in časovni zamik),

Logiranje vseh neuspešnih zahtev v sistemu OpenSearch z možnostjo alarmiranja (Prometheus alerting),

Transakcijska zaščita vhodnih podatkov – podatki se obravnavajo šele po potrjeni validaciji.

4.4.5 Sinhronizacijski tokovi

Vse **PULL integracije** temeljijo na **časovnih žigih** in periodični sprožitvi (npr. vsakih 24 h),

PUSH integracije uporabljajo potrjene strukture in avtentikacijo preko **OAuth 2.0** z uporabo **JWT**,

Vsaka integracija vključuje **revizijsko sled** (Kafka log + OpenSearch indeks).

4.5 Tehnične omejitve

Razvoj in delovanje informacijskega sistema POKK mora upoštevati vrsto tehničnih omejitev, ki izhajajo iz organizacijskih, infrastrukturnih, varnostnih in zakonskih zahtev. Omejitve so pomemben del arhitekturne zasnove, saj določajo meje izbire tehnologij, način integracije, dostopnosti in izvajanja posameznih funkcionalnosti.

4.5.1 Tehnične omejitve po kategorijah

ID	Kategorija	Opis omejitve	Vpliv na sistem	Merilo skladnosti
TO-01	Programska oprema	Uporaba dovoljenih orodij in okolij iz dokumenta <i>Podprta-programska-oprema-DRO_V1.0</i> .	Omejen nabor programskih jezikov, CI/CD orodij, okolij za monitoring.	Uporaba izključno dovoljenih tehnologij (npr. Java, PostgreSQL, Jenkins, Grafana).
TO-02	Podatkovne baze	Primarna baza mora biti komercialno sprejeta.	Neuporaba eksperimentalnih rešitev za transakcijski sloj.	Sistem uporablja eno izmed dovoljenih baz; potrjeno v tehnični dokumentaciji.
TO-03	Varnost	Obvezna uporaba TLS 1.3 in OAuth 2.0 ali JWT za vse API-je in medsystemsko komunikacijo.	Izključena podpora starejšim ali manj varnim protokolom (npr. TLS 1.2, Basic Auth).	Vsi API-ji zaščiteni s TLS 1.3 in podpisanimi žetoni; preverjeno v testih.
TO-04	Strežniška infrastruktura	Vsi podsistemi morajo delovati v kontejneriziranem okolju (Docker/Kubernetes).	Ni dovoljena monolitna implementacija brez možnosti ločevanja in razširljivosti.	Arhitektura temelji na ločenih mikroservisih, z dokumentiranim orkestratorjem.
TO-06	Modularnost	Vsaka funkcionalna domena mora biti ločena v samostojen mikroservis z lastno podatkovno plastjo.	Izključena skupna implementacija več domen v eno komponento (npr. živali in premiki).	Vsak modul razvit kot samostojna komponenta z ustreznim API in

ID	Kategorija	Opis omejitve	Vpliv na sistem	Merilo skladnosti
				DAL.
TO-08	Jezikovna podpora	Podprta morata biti vsaj slovenščina in angleščina; vsebinske komponente in šifranti morajo biti lokalizirani.	Omejena uporaba trdih besedil v kodi.	Vse maske in strukture podpirajo i18n; preverljivo na UI nivoju.

4.5.2 Posebne omejitve vezane na razvoj in razpis

Uporaba odprtokodnih komponent je priporočena povsod, kjer je to mogoče, vendar mora biti skladna s politiko odprtih licenc (npr. Apache, MIT, EPL). Komercialne knjižnice so dovoljene le ob predhodni utemeljitvi.

Kode, knjižnice ali storitve, ki onemogočajo prenosljivost rešitve (vendor lock-in), niso dovoljene brez izrecne odobritve naročnika.

Avtomatizirano testiranje je obvezno in mora vključevati obremenitvene ter varnostne scenarije.

Dostop do sistema mora biti mogoč hkrati za več predstavništev uporabnika, z dinamično izbiro konteksta delovanja.

5 Podatkovna arhitektura in modeli

5.1 Opis podatkovnih zbirk

Informacijski sistem POKK uporablja večplastno podatkovno arhitekturo, ki temelji na centralni **podatkovni bazi** kot primarni podatkovni zbirki in vključuje podporne komponente za poročanje, iskanje, revizijsko sledenje in morebitno optimizacijo poizvedb.

5.1.1 Pregled podatkovnih zbirk

Zbirka	Tip	Opis in namen	Tehnologija
Operativna relacijska baza	Skladna z GTZ	Glavna transakcijska baza podatkov. Hrani vsebinske, referenčne in varnostne podatke sistema.	Skladna z GTZ
Journal baza	Skladna z GTZ (zgodovinska)	Hrani posnetke vseh sprememb nad entitetami glavnih shem. Vsaka sprememba se zajame s triggerji ob INSERT/UPDATE operacijah.	Skladna z GTZ
DW – podatkovno skladišče	Skladna z GTZ (OLAP)	Ločena zbirka za analitične in poročilne podatke, napolnjena s pomočjo ETL mehanizma.	Ločena instanca
OpenSearch indeks	Iskalna baza	Indeks za iskanje po entitetah, revizijskih sledovih in metapodatkih.	OpenSearch

5.1.2 Organizacija podatkov po shemah

Operativna baza je razdeljena na tri osnovne sheme:

- **POKK_PODATKOVNA_SCHEMA**

Vsebuje vsebinske podatke (živali, premiki, lastništva, dogodki ipd.). Vse entitete v tej shemi vsebujejo obvezne **revizijske in verzijske attribute**, ki zagotavljajo sledljivost ustvarjanja, spreminjanja in logičnega brisanja zapisov.

- **POKK_VARNOSTNA_SHEMA**

Vsebuje podatke o uporabniških računih, kontekstih prijave, pravicah in zastopanjih. Tudi entitete v tej shemi vključujejo **revizijske in verzijske attribute**, kar omogoča revizijo vseh sprememb v varnostnem podsistemu.

- **POKK_JOURNAL_SHEMA**

Vsebuje **ločene zgodovinske tabele**, ki zrcalijo strukturo entitet iz podatkovne in varnostne sheme. Vsaka sprememba nad entitetami (INSERT ali UPDATE) sproži **trigger**, ki zabeleži stanje zapisa ob spremembi, skupaj z uporabniškimi metapodatki in časovno oznako.

- Vpis v žurnalno tabelo vključuje celoten "snapshot" entitete v trenutku spremembe.
- Operacije logičnega brisanja (nastavitev revizijskega atributa za brisanje) se prav tako beležijo.

Fizično brisanje podatkov v operativni bazi **ni dovoljeno**; vsi podatki ostajajo dostopni prek zgodovinskih tabel ali s pomočjo revizijskih iskalnikov (OpenSearch).

5.2 Integracijski vmesniki

V informacijskem sistemu POKK je dostop do podatkov in funkcionalnosti podatkovne baze omogočen izključno preko **integracijskih vmesnikov**, ki so izvedeni s pomočjo **proxy shem**. Neposreden dostop aplikacijskih komponent do podatkovnih tabel je onemogočen. Ta arhitektura sledi načelom **varnostne enkapsulacije, sledljivosti, modularnosti in vzdržnosti**, kot jih predpisuje metodologija GTZ.

5.2.1 Proxy sheme

Proxy sheme so sheme znotraj podatkovne baze, ki **vsebujejo izključno sinonime** (SYNONYM) do dejanskih objektov (pogledov in paketov) v podatkovni in varnostni shemi. Te sheme:

- ne vsebujejo lastnih objektov (tabel, paketov, pogledov),
- ne vsebujejo poslovne logike,
- ne izvajajo nobene neposredne obdelave podatkov,
- **nimajo dostopa do žurnal sheme** (POKK_JOURNAL_SHEMA).

5.2.2 Mehanizem dodeljevanja dostopa in način delovanja

Dostop do objektov (pogledov in paketov) v glavni podatkovni in varnostni shemi se zagotovi izključno z uporabo **pravic, dodeljenih iz izvornih shem**, kot sledi:

V podatkovni in varnostni shemi se posameznim sinonimom v proxy shemi izvede ukaz GRANT <SELECT ON VIEW_NAME | EXECUTE ON PACKAGE_NAME> TO PROXY_SHEMA;

Branje podatkov poteka izključno preko vnaprej pripravljenih in varnostno omejenih **pogledov ali procedur**, ki so optimizirani za namen prikaza ali poročanja.

Vse aplikacijske interakcije z bazo (branje, pisanje, validacije) se izvajajo **prek paketov**, ki izvajajo logiko in izvajajo kontrolo nad spremembami.

Proxy sheme se obravnavajo kot **varnostni prehodi**, kjer je mogoče dosledno nadzirati vloge, slediti izvajanju operacij in zagotavljati celovitost podatkov. Aplikacije nato komunicirajo izključno preko teh sinonimov, brez možnosti dostopa do izvornih tabel ali zgodovinskih zapisov.

5.2.3 Prednosti arhitekture

Enostavno upravljanje pravic – vse aplikacijske pravice so centralizirane in nadzorovane preko GRANT ukazov znotraj proxy shem.

Boljša ločenost odgovornosti – aplikacija nikoli neposredno ne dostopa do struktur v notranjih shemah.

Možnost verzioniranja – spremembe v paketih in pogledih so izvedljive brez vpliva na aplikacijsko plast, če se ohrani enak vmesnik.

Nadgradljivost – proxy sheme omogočajo dodajanje novih funkcionalnosti brez prestrukturiranja baze.

6 Varnostne zahteve

6.1 Varnostna politika in mehanizmi

Informacijski sistem POKK mora zagotavljati visoko raven informacijske varnosti, zlasti zaradi občutljive narave podatkov, ki jih obdeluje (osebni podatki, podatki o živalih, podatki o premikih, genetski podatki ipd.). Sistem mora temeljiti na varnostni politiki, ki sledi načelom iz standardov **ISO/IEC 27002:2022**, **GTZ 2.3**, priporočil **OWASP TOP 10**, ter nacionalnih smernic za javne informacijske sisteme.

6.1.1 Osnovna varnostna načela

Varnostna arhitektura mora temeljiti na naslednjih načelih:

- **Najmanjših pravic (Principle of Least Privilege):** Vsak uporabnik ali proces ima dodeljene le tiste pravice, ki so nujne za izvajanje funkcij.
- **Razmejitev odgovornosti:** Vsaka komponenta (uporabniška, aplikacijska, podatkovna) ima jasno določene pristojnosti in ločen dostop.
- **Avtentikacija in avtorizacija:** Centralizirano upravljanje uporabnikov, vključno s podporo za federativno avtentikacijo (SSO), identifikacijo reprezentančnih vlog in avtentikacijo s podpisanimi žetoni (OAuth2 / JWT).
- **Revizijska sledljivost:** Vse operacije nad podatki se beležijo in shranjujejo ločeno v žurnalni shemi, indeksirane z OpenSearch.
- **Šifriranje:** Vsi prenosi podatkov med komponentami (API-ji, sinhronizacija) morajo biti zaščiteni s protokolom **TLS 1.3**.
- **Ločenost okolij in podatkovnih domen:** Fizična in logična ločitev varnostne, vsebinske in revizijske podatkovne sheme.

6.1.2 Mehanizmi in orodja

Upravljanje identitet in dostopov se izvaja preko varnostne sheme (UAC), ki skrbi za prijavo, dodeljevanje vlog, zastopništvo in sledljivost aktivnosti uporabnikov.

Varnost aplikacijskega sloja temelji na principih OWASP, z vključitvijo zaščite pred SQL Injection, XSS, CSRF in drugimi najpogostejšimi ranljivostmi.

Dostop do podatkov poteka izključno preko **proxy shem**, ki vsebujejo **sinonime** do shranjenih procedur in pogledov; dodeljevanje pravic poteka z `GRANT EXECUTE` in `GRANT SELECT` iz primarne sheme.

Omejitve dostopa: Proxy sheme nimajo dostopa do žurnal sheme, kar zagotavlja neodvisnost in zaščito revizijskih podatkov.

Monitoring in alerting: Sistem mora vključevati orodja za sprotno spremljanje dostopov, obremenitev, nenavadnih vzorcev obnašanja ter za proaktivno alarmiranje (npr. Prometheus + Alertmanager).

6.1.3 Politika varnostnega testiranja

Pred namestitvijo v produkcijsko okolje mora biti izvedeno varnostno testiranje sistema (penetracijski testi, varnostna analiza kode, preverjanje ranljivosti).

Vsa testiranja morajo biti dokumentirana, rezultati pa vključeni v CI/CD proces kot pogoji za release nove verzije sistema.

Avtomatizirano preverjanje ranljivosti mora biti vključeno v vsako CI/CD cevovod (npr. z uporabo SonarQube, Trivy).

6.1.4 Upravljanje z incidenti

V sistem mora biti vgrajen mehanizem za zaznavanje in odzivanje na incidente, vključno z beleženjem poskusov nedovoljenih dostopov, anomalij v vedenju uporabnikov in nenavadnih API vzorcev.

Dogodki se beležijo prek Kafka kanalov in obdelujejo v sistemu OpenSearch, kar omogoča hitro analizo in izdelavo poročil za varnostno ekipo.

6.2 Skladnost s standardi

Informacijski sistem POKK mora pri razvoju, delovanju in vzdrževanju dosegati skladnost s ključnimi nacionalnimi in mednarodnimi varnostnimi standardi, priporočili in smernicami.

Skladnost je pogoj za dolgoročno upravljanje sistema, njegovo vključevanje v državne informacijske arhitekture ter izpolnjevanje zakonodajnih obveznosti na področju varstva podatkov in informacijske varnosti.

6.2.1 Obvezni standardi

Standard	Vsebina	Uporaba v sistemu POKK
ISO/IEC 27002:2022	Kontrole informacijske varnosti – nadzor dostopa, zaščita komunikacij, revizija, upravljanje z	Osnova za varnostno politiko, tehnične in organizacijske ukrepe ter revizijske sledi.

Standard	Vsebina	Uporaba v sistemu POKK
	incidenti.	
OWASP TOP 10	Seznam najpogostejših ranljivosti spletnih aplikacij in priporočil za zaščito.	Uporabljen pri razvoju aplikacijskega sloja, preverjanje v CI/CD (SAST/DAST).
GTZ 2.3	Nacionalna metodologija za zasnovu informacijskih rešitev javnega sektorja.	Uporabljen za tehnično arhitekturo, modularnost, revizijsko sledljivost, ločenost slojev.
Podprta-programska-oprema-DRO V1.0	Seznam podprtih in dovoljenih tehnologij v sistemih javne uprave.	Uporaba potrjenih tehnologij (npr. PostgreSQL, Java, Jenkins, Grafana, OpenSearch).
TLS 1.3	Kriptografski protokol za zaščito komunikacije.	Uporabljen za šifriranje vseh komunikacij med komponentami in zunanjimi sistemi.
OAuth 2.0 / JWT	Standardi za varno avtentikacijo in avtorizacijo.	Uporaba za dostop do API-jev, prijavo v sistem in upravljanje s kontekstom zastopanja.
ZVOP-2	Nacionalni zakon o varstvu osebnih podatkov.	Uporabljen kot pravna podlaga za obdelavo osebnih podatkov, vključno z revizijo dostopov.

6.2.2 Skladnost v praksi

Vsi varnostni ukrepi bodo **preverjeni v sklopu CI/CD** (s statičnim in dinamičnim testiranjem kode ter avtomatiziranimi varnostnimi pregledi).

Ob vsakem večjem sproščanju produkcijske verzije bo izvedena **varnostna presoja in test penetracije**.

Aplikacijske komponente bodo pregledane na skladnost z **OWASP smernicami**.

Vzpostavljena bo dokumentacija o **politiki ravnanja z osebnimi podatki**, kot jo predpisuje **ZVOP-2**.

V primeru vključitve v enotno infrastrukturo DRO mora biti sistem pripravljen za **varnostno akreditacijo**.

6.3 Revizijske sledi

Revizijske sledi so ključni varnostni mehanizem za zagotavljanje **sledljivosti aktivnosti** znotraj informacijskega sistema POKK. Omogočajo nadzor nad spremembami podatkov, pregled operacij, identifikacijo zlorab in so temelj za forenzično analizo varnostnih incidentov. Vzpostavitev in dosledna uporaba revizijskega mehanizma je obvezna za vse podsisteme.

6.3.1 Journal shema

Sistem POKK uporablja namensko **JOURNAL_SHEMA**, ki deluje kot ločen sloj v podatkovni bazi, kamor se zapišejo vse pomembne spremembe nad entitetami glavnih shem (podatkovna in varnostna). Vsaka produkcijska entiteta je podprta z:

- **triggerjem na INSERT in UPDATE**, ki avtomatsko zapiše posnetek spremembe v ustrezno tabelo v **JOURNAL_SHEMA**,

- **zrcalno tabelo** v žurnalni shemi z dodatnimi polji:
 - ACTION (INSERT ali UPDATE)
 - ACTION_TS (čas izvedbe)
 - ACTION_BY (ID uporabnika, ki je sprožil spremembo)
 - TX_ID (ID transakcije za sledenje povezanim operacijam)
 - CONTEXT_ID (kontekst zastopanja, če obstaja)
- **roloidično** povezavo na ID entitete iz primarne sheme.

Neposreden dostop do journal sheme iz aplikacijskih proxy shem **ni dovoljen**. Podatke iz žurnala se lahko bere le preko sistemskih modulov (npr. revizijsko poročanje, nadzorni API-ji) z ustrezno utemeljitvijo.

6.3.2 Aplikacijsko logiranje

Poleg podatkovne revizije je obvezno tudi **aplikacijsko logiranje operacij**, ki mora biti **strukturirano** in **centralizirano**. Vsaka aplikacijska komponenta mora pisati v:

- **standardiziran log format** (JSON ali struktura s timestamp, ID seje, uporabnik, rezultat, entiteta),
- **Kafka kanal** za centralno zbiranje logov,
- **OpenSearch indeks**, kjer se izvaja iskanje, agregacija in vizualizacija (npr. z Grafano, Kibano).

6.3.3 Nivoji logiranja

Vse aplikacijske komponente morajo omogočati konfiguracijo logiranja z naslednjimi nivoji:

Nivo	Namen
TRACE	Sledenje podrobnostim o izvedbi znotraj ene operacije (npr. vhodni parametri, vmesne vrednosti).
INFO	Beleženje uspešnih operacij (npr. uspešna prijava, uspešen vnos).
WARN	Določene operacije so izvedene s posebnimi pogoji (npr. podatki niso popolni, vendar dovoljeni).
ERROR	Napake, ki so preprečile uspešno izvedbo operacije.
CRITICAL	Kritične napake , ki povzročijo sesutje procesa, neobvladane izjeme, izgubo storitve ali nujno potrebo po intervenciji (npr. Unhandled Exception, OutOfMemory, Broken pipe).
AUDIT	Posebej pomembne operacije, ki vplivajo na podatke, uporabnike ali sistem (npr. spreminjanje pravic, masovni vnosi).

6.3.4 Zahteve za vzdrževanje in arhiviranje

Dnevniki (logi) morajo biti hranjeni **vsaj 5 let** (razen če zakon določa drugače), z možnostjo iskanja, izvoza in rekonstrukcije dogodkov.

Dnevni promet se shranjuje v OpenSearch z rotacijo indeksov (npr. tedensko), z možnostjo arhiviranja na sekundarne sisteme (npr. Glacier).

Vse žurnalne tabele morajo imeti **neizbrisne in nespremenljive zgodovinske zapise**, ki se ne smejo popravljati ali nadomeščati.

6.4 Preverjanje in testiranje

Varnostne funkcionalnosti informacijskega sistema POKK morajo biti predmet sistematičnega preverjanja in testiranja v vseh fazah razvoja, vzdrževanja in posodabljanja. Cilj varnostnega preverjanja je pravočasno odkrivanje ranljivosti, validacija zaščitnih mehanizmov in preprečevanje vnosov nevarne kode ali konfiguracij.

6.4.1 Varnostno preverjanje v okviru CI/CD

Sistem CI/CD mora vključevati avtomatizirane varnostne kontrole:

- **SAST (Static Application Security Testing)**: analiza izvorne kode ob vsaki spremembi; identificira ranljivosti še pred izvajanjem.
- **DAST (Dynamic Application Security Testing)**: testiranje delujoče aplikacije z avtomatiziranimi orodji (npr. OWASP ZAP) v testnem okolju.
- **Dependency scanning**: preverjanje odvisnosti (knjižnic, paketov) na znane ranljivosti (npr. CVE baza).
- **Container image scanning**: varnostni pregled slik kontejnerjev (npr. Trivy, Clair), pred nameščanjem v okolja.

Vse kontrole morajo biti vključene kot **obvezen del CI/CD cevovoda**. Vsaka kritična ugotovitev (CRITICAL, HIGH) mora preprečiti napredovanje verzije v višje okolje.

6.4.2 Periodično varnostno testiranje

Ne glede na avtomatizacijo mora biti izvedeno tudi redno ročno ali nadzorovano varnostno testiranje:

- **Penetracijski testi (Pentest)**: vsaj 1× letno oziroma pred večjo objavo v produkcijo, s poročilom za naročnika.
- **Konfiguracijski pregledi**: preverjanje ustreznosti TLS konfiguracij, CORS, headerjev (HSTS, CSP).
- **Social engineering scenariji** (opcijsko): v primeru integracije s kritičnimi podatkovnimi viri ali občutljivimi podatki.

6.4.3 Notranja preverjanja revizijske konsistence

Redno preverjanje **ustreznosti zapisov v journal shemi** glede na dogajanje v aplikaciji.

Validacija pravilnega logiranja vseh zahtevanih nivojev (INFO, AUDIT, ERROR, CRITICAL).

Izdelava **poročil skladnosti** revizijskih podatkov za vsako večjo objavo.

6.4.4 Zahteve za izvajalce

Vsi izvajalci, ki sodelujejo v razvoju ali vzdrževanju, morajo:

- uporabljati certificirana orodja za preverjanje varnosti,
- upoštevati rezultate testov v lastnih delovnih ciklih,

- dokumentirati odpravo ugotovljenih ranljivosti z revizijsko sledjo (komentar, commit, issue tracking),
- zagotoviti **testne skripte in podatke**, ki omogočajo ponovljivost varnostnih preverjanj s strani naročnika.

7 Namestitev in konfiguracija

7.1 Postopek namestitve

Namestitev informacijskega sistema POKK temelji na avtomatiziranem procesu prek sistema za neprekinjeno integracijo in dostavo (CI/CD), pri čemer se dosledno ločujejo aplikacijska in podatkovna komponenta ter zagotavlja revizijska sledljivost nameščenih verzij.

7.1.1 Posopek namestitve nove verzije

1. Priprava SQL (posodobitvene) skripte

SQL posodobitveno skripto pripravi razvijalec. Skripta pokriva vse spremembe v strukturi in vsebini baze med obstoječo in novo različico.

SQL skripta mora biti idempotentna, sledljiva in dokumentirana, z možnostjo ponovitve brez neželenih učinkov.

2. Priprava namestitvenih artefaktov

Po uspešni validaciji funkcionalne in varnostne ustreznosti aplikacijske kode CI/CD sistem samodejno pripravi:

- **namestitvene datoteke aplikacije** (npr. kontejnerske slike, konfiguracije, statične komponente).

3. Prenos v ciljno okolje

Namestitveni paket (aplikacija + SQL) se prenese v ustrezno okolje (TEST, PREPROD, PROD), odvisno od faze izdaje in potrditvenih postopkov.

4. Izvedba nameščanja

Ročno ali samodejno se zažene **SQL posodobitvena skripta**, ki izvede potrebne spremembe nad bazo.

CI/CD sistem izvede **namestitev aplikacijskih komponent**, nadomesti kontejnerje ali datoteke, ter sproži inicializacijske postopke.

Aktivirajo se **health-check mehanizmi**, ki preverijo dostopnost in odzivnost ključnih storitev.

5. Potrditev in monitoring

Po namestitvi se izvedejo osnovni testi delovanja (smoke testi), preveri skladnost različice ter začne spremljanje z orodji za nadzor (Prometheus, Grafana, OpenSearch).

V primeru napake se sproži postopkovno definiran **rollback**, če je to predhodno omogočeno (npr. shranjena prejšnja verzija aplikacije in ustrezna SQL revert skripta).

V postopku namestitve v okolji TEST in PREPROD je že preverjeno delovanje SQL posodobitvene skripte in pripadajočih aplikacijskih artefaktov.

7.1.2 Sledljivost in arhiviranje

Vsaka izdaja mora biti **verzionirana**, opremljena z metapodatki (čas, izvajalec, spremembe, okolje).

CI/CD sistem ali repozitorij mora hraniti arhive:

- aplikacijskih paketov,
- SQL skript (vključno z revert varianto, kjer je smiselno),
- konfiguracij,
- rezultatov postopka nameščanja (log datoteke, poročila).

Na ta način je omogočen **nadzor nad vsako spremembo** ter izvedljiv sistemski **rollback** v primerih kritičnih napak.

7.1.3 Odgovornosti

Razvijalec pripravi, preizkusi in dokumentira SQL skripte za posodobitev baze ter zagotovi njihovo ujemanje z različico aplikacije.

CI/CD okolje avtomatizira pripravo aplikacijskih artefaktov, njihovo razporejanje in izvajanje namestitvenih postopkov.

Nadzorni izvajalec ali naročnik potrjuje prehod med okoljih in spremlja rezultate namestitve.

7.2 Konfiguracijski parametri

Delovanje informacijskega sistema POKK je odvisno od številnih konfiguracijskih parametrov, ki morajo biti ustrezno organizirani, zaščiteni in dokumentirani. Uporaba konfiguracij mora slediti načelom **ločitve podatkov od kode, varnostne zaščite občutljivih informacij in okoljskega profiliranja**.

7.2.1 Razvrstitev konfiguracijskih parametrov

1. Aplikacijski (funkcionalni) parametri

Parametri, ki določajo delovanje aplikacije v posameznem okolju (npr. naslov zunanjega sistema, število paralelnih niti, časovna okna za sinhronizacijo). Ti parametri se hranijo v:

- konfiguracijskih datotekah (npr. `application.yaml`, `.env`, `config.properties`),
- parametrih kontejnerja ali okolja (npr. `ENV` `VARS`),
- namensko razvitih upravljavcih konfiguracij (npr. konsolidiran config modul za vse mikroservise).

2. Varnostno občutljivi parametri

Sem spadajo:

- gesla in API ključi,
- enkripcijski ključi in žetoni (npr. JWT signing key),

- poverilnice za povezave do zunanjih sistemov (LDAP, SMTP, zaledne baze).

Ti parametri morajo biti:

- **shranjeni v varnih mehanizmi za tajne podatke** (npr. HashiCorp Vault, Kubernetes Secrets, Docker Secrets),
- **dostopni samo avtoriziranim komponentam**, z nadzorom dostopa (RBAC),
- **ločeni od konfiguracijskih datotek**, ki so dostopne širšemu krogu razvijalcev.

7.2.2 Zahteve

Vsi konfiguracijski parametri morajo biti zbrani, dokumentirani in verzionirani za vsako okolje posebej (DEV, TEST, PREPROD, PROD).

Konfiguracije morajo biti **v celoti ločene od izvirne kode** ter omogočati **samostojno upravljanje brez regradnje** aplikacije.

Občutljivi podatki se **ne smejo zapisovati v loge** ali vključevati v diagnostiko.

7.3 Namestitvena navodila

Vsaka izdaja informacijskega sistema POKK mora biti opremljena z natančno dokumentiranimi **namestitvenimi navodili**, ki omogočajo preverljivo, ponovljivo in sledljivo izvedbo postopka namestitve v ciljno okolje. Navodila so del obvezne tehnične dokumentacije vsake verzije.

7.3.1 Vsebina namestitvenih navodil

Namestitvena navodila morajo vsebovati najmanj naslednje sestavine:

1. Identifikacija izdaje:

- ime in številka verzije,
- datum izdaje,
- okolje, za katero je izdaja pripravljena (npr. TEST, PREPROD, PROD),
- odgovorna oseba oz. izvajalec izdaje.

2. Opis vsebine:

- seznam aplikacijskih komponent, ki se nadgrajujejo (mikrostoritve, moduli, uporabniški vmesniki),
- seznam sprememb v primerjavi s prejšnjo verzijo (changelog),
- seznam novih ali spremenjenih konfiguracijskih parametrov.

3. Navodila za namestitev:

- koraki za namestitev aplikacijskih artefaktov (npr. zamenjava kontejnerskih slik),
- natančna navodila za izvedbo SQL posodobitvene skripte,
- posebna opozorila (npr. potreba po ekskluzivnem dostopu, pričakovani časovni okvir).

4. Navodila za preverjanje delovanja po namestitvi:

- opis osnovnih testov (smoke test), ki jih mora izvajalec izvesti po namestitvi,
- kontrolna točka za validacijo uspešne nadgradnje (npr. verzija API, odzivnost, testni vnos).

5. Postopek za rollback (če je predviden):

- pogoji za sprožitev povrnitve na prejšnjo verzijo,
- lokacija in ime arhivirane prejšnje verzije,
- navodila za izvajanje revert SQL skripte (če obstaja),
- kontaktna oseba za eskalacijo v primeru napake.

7.3.2 Format in dostopnost

Navodila morajo biti dostavljena v obliki .md, .pdf ali kot datoteka znotraj CI/CD sistema (npr. INSTALL.md).

Vsaka verzija mora imeti lastna, z verzijo usklajena navodila; generična navodila niso dovoljena.

7.4 Okolja

Za razvoj, testiranje, potrjevanje in produkcijsko delovanje informacijskega sistema POKK se uporabljajo ločena izvajalna okolja. Ta pristop omogoča varno uvedbo sprememb, neodvisno testiranje funkcionalnosti in visoko razpoložljivost produkcijskih storitev.

7.4.1 Vrste okolij in njihov namen

Oznaka	Okolje	Namen
DEV	Razvojno okolje	Namenjeno samostojnemu razvoju in integraciji posameznih komponent. Omogoča lokalno testiranje, eksperimentiranje in CI/CD build procese.
TEST (UAT)	Testno / uporabniško potrditveno okolje	Namenjeno funkcionalnemu in varnostnemu testiranju. Vključuje zgodovinske podatke (anonimizirane) in je dostopno tako razvijalcem kot naročniku.
PREPROD	Predprodukcijsko okolje	Replika produkcijskega okolja za validacijo verzije tik pred objavo. Uporablja enako konfiguracijo kot produkcija, vendar z izoliranimi podatki in dostopi.
PROD	Produkcijsko okolje	Dejansko okolje za končne uporabnike. Vključuje vse varnostne, obremenitvene in visoko razpoložljive konfiguracije. Brez neposrednega dostopa za razvojne ekipe.

7.4.2 Ključne značilnosti izvajalnih okolij

Vsako okolje ima:

- **lastne baze podatkov** in sheme (ločene instance ali baze),
- **ločene konfiguracije** in parametre (URL-ji, ključi, poti),
- **ločene CI/CD cevovode**, kjer je to ustrezno.

Prenos med okolji (npr. TEST → PREPROD → PROD) je dovoljen le ob potrditvi izpolnjenih

pogojev:

- uspešna izvedba avtomatiziranih testov,
- potrjeni rezultati testnih scenarijev (ročno ali avtomatsko),
- uspešna namestitev v prejšnjem okolju.

7.4.3 Omejitve in varnost

Dostop do okolij je nadzorovan glede na vlogo:

- **DEV:** dostop za razvojne ekipe,
- **TEST (UAT):** dostop za razvojne ekipe, testne uporabnike in naročnika,
- **PREPROD:** dostop samo za validacijsko ekipo,
- **PROD:** omejen administrativni dostop za vzdrževalce, brez neposrednega dostopa za razvojne izvajalce.

PREPROD in PROD okolji morata imeti **enako verzijo infrastrukture, konfiguracije in orkestracije** (npr. enake Helm chart verzije, enaka shema CI/CD pipeline), kar omogoča ponovljivost rezultatov testiranja.

Vsa okolja morajo biti podprta z **monitoring in log mehanizmi**, ki omogočajo zgodnje zaznavanje napak in primerjavo obnašanja med okoljih.

8 Testiranje in validacija

8.1 Načrt testiranja

Cilj testiranja je preveriti funkcionalno pravilnost, varnost, zmogljivost in skladnost informacijskega sistema POKK z zahtevami naročnika, predpisi ter tehnično specifikacijo. Testiranje se izvaja sistematično v več fazah in okoljih, z uporabo ročnih in avtomatiziranih pristopov, pri čemer mora biti vsak rezultat **sledljiv, ponovljiv in dokumentiran**.

8.1.1 Vrste testiranja

Vrsta testa	Namen	Izvajaja
Enotski testi (unit tests)	Preverjanje pravilnosti posameznih metod, funkcij in modulov.	Razvijalec
Integracijski testi	Preverjanje pravilnosti komunikacije med komponentami in sistemi.	Razvijalec / avtomatizirano
Funkcionalni testi	Preverjanje skladnosti z zahtevanim vedenjem sistema (use-case testiranje).	Tester / naročnik
Obremenitveni testi (load & stress tests)	Preverjanje zmogljivosti in stabilnosti sistema pri visokih obremenitvah.	Tehnična ekipa / avtomatizirano
Varnostni testi	Preverjanje odpornosti proti ranljivostim in nedovoljenim dostopom.	Varnostna ekipa / zunanji izvajalec
Uporabniški potrditveni testi (UAT)	Preverjanje ustreznosti sistema z vidika uporabnikov pred prevzemom.	Naročnik / funkcionalni uporabniki

8.1.2 Faze testiranja

1. Testiranje v razvojnem okolju (DEV)

Vključuje enotske in osnovne integracijske teste kot del CI procesa.

Vsak commit sproži avtomatizirano testiranje (linting, build-check, test coverage).

2. Testiranje v testnem okolju (TEST/UAT)

Vključuje funkcionalne, obremenitvene in delno varnostne teste.

Vključeni so zgodovinski podatki (anonimizirani) za testiranje realnih scenarijev.

Uporabniški potrditveni testi potekajo pod nadzorom naročnika.

3. Validacija v predprodukcijskem okolju (PREPROD)

Namenjena preverjanju stabilnosti verzije pred vpeljavo v produkcijo.

Vključuje popolno ponovitev produkcijskega delovanja (enaka konfiguracija, obremenitev).

Uporablja se za test nameščanja, inicializacije ter končnih uporabniških tokov.

4. Testni protokoli v produkciji (PROD smoke tests)

Po namestitvi se izvede niz avtomatskih testov (npr. odzivnost API-jev, validacija osnovnih funkcij).

Namenjeni takojšnjemu odkrivanju kritičnih napak (t.i. smoke testi).

8.1.3 Upravljanje testnih primerov in rezultatov

Testni primeri se upravljajo v **sistemu za upravljanje testiranj** (npr. TestRail, Zephyr ali podobno), zvezani s CI/CD in issue tracking sistemom.

Vsak testni primer mora imeti:

- identifikator,
- vhodne podatke,
- pričakovani rezultat,
- stopnjo kritičnosti,
- informacijo o izvedbi in rezultat (pass/fail),
- sledljivost do zahteve (traceability matrix).

8.2 Testni scenariji in primeri uporabe

Testni scenariji za posamezne funkcionalnosti informacijskega sistema POKK so opisani znotraj poglavja **20 - Funkcionalne zahteve**, kjer so ob vsakem primeru uporabe navedeni tudi osnovni pogoji uspešnega izvajanja funkcionalnosti. Ti scenariji predstavljajo **izhodišče za podrobno validacijo v fazi PZI**, kjer bo naročnik pripravil celovite testne primere.

V fazi PZI se bo na podlagi teh opisov pripravila **celovita testna matrika**, ki bo vključevala tudi

obravnavo robnih primerov in točk odpovedi.

8.3 Merila uspešnosti

Za vsako ključno funkcionalnost sistema POKK morajo biti v okviru prevzemnega postopka opredeljena merila uspešnosti, ki določajo pogoje, pod katerimi se funkcionalnost šteje kot pravilno implementirana in pripravljena za prevzem. V tej fazi specifikacije so opredeljena **splošna merila**, podrobni testni scenariji pa bodo izdelani v fazi **PZI** v sodelovanju z naročnikom in se priložijo k temu dokumentu.

8.3.1 Osnovna načela validacije

Vsaka funkcionalnost mora biti preverjena z vsaj enim uspešno izvedenim testom, ki pokriva osnovni tok uporabe (main flow).

Mejni pogoji (edge cases) in **točke odpovedi** morajo biti identificirane v PZI in ustrezno pokrite s testnimi primeri.

Funkcionalnosti, ki vključujejo **vnos, spremembo ali prikaz podatkov**, morajo dokazljivo:

- obravnavati neveljavne ali nepopolne vnose,
- preprečiti neželene spremembe (npr. zablokirati, opozoriti),
- zagotavljati odziv v sprejemljivem času (glej KPI iz poglavja 4.3).

8.3.2 Zahteve za testno pokritost

Vsak primer uporabe iz poglavja 20 - Funkcionalne zahteve mora imeti svoje **preverljive testne kriterije uspešnosti**.

Testni primeri morajo vključevati vsaj:

- **en osnovni primer uporabe** (standardni scenarij),
- **en mejni primer** (npr. nepopolni podatki, neveljavne kombinacije),
- **eno preverjanje nedostopnosti ali napačnih vlog** (avtorizacija).

8.3.3 Prevzemni kriterij

Funkcionalnost se šteje kot uspešno implementirana, če:

- vsi pripadajoči testi v okolju PREPROD uspešno prestanejo izvajanje,
- se funkcionalnost obnaša skladno z opisanimi poslovnimi pravili in pričakovanji uporabnikov,
- ne povzroča degradacije drugih funkcionalnosti (regresijsko testiranje),
- je sprememba ustrezno dokumentirana in vključena v prevzemno dokumentacijo.

8.4 Dokumentiranje rezultatov

Za zagotavljanje sledljivosti in dokazljivosti uspešne izvedbe testov mora biti vsak izveden test dokumentiran na standardiziran in strukturiran način. Dokumentiranje rezultatov testiranja je obvezno za vse faze preverjanja sistema in predstavlja ključen vhodni dokument za izvedbo prevzema.

8.4.1 Osnovna načela

Vsak testni primer mora imeti **identifikator**, **povezavo do funkcionalne zahteve** (primer uporabe), **datum izvedbe**, **izvajalca** ter **rezultat**.

Rezultati morajo biti vodeni v **centraliziranem sistemu za upravljanje testov**, ki omogoča izvoz poročil (npr. CSV, PDF, HTML) in povezovanje s CI/CD ter issue tracking sistemi.

Testi morajo biti ločeno vodeni glede na **okolje izvajanja** (TEST, PREPROD) in **vrsto testa** (funkcionalni, obremenitveni, varnostni, UAT).

8.4.2 Struktura dokumentacije

Za vsak zaključen testni cikel mora biti izdelano poročilo, ki vsebuje:

- osnovne informacije (verzija sistema, okolje, datum, izvajalec),
- seznam testnih primerov z rezultati (PASS, FAIL, BLOCKED, SKIPPED),
- opis ugotovljenih napak ali odstopanj (vključno z vezavo na zabeležene incidente),
- potrdilo naročnika o izvedbi testov (v primeru UAT ali prevzemnega testiranja),
- seznam odprtih vprašanj ali odprtih točk, ki niso kritične za prevzem, a zahtevajo spremljanje (open issues).

8.4.3 Shranjevanje in dostopnost

Dokumentacija o testiranju mora biti **shranjena v repozitoriju projekta**, z možnostjo revizije in zgodovinskega vpogleda.

Za vsako produkcijsko verzijo mora biti pripeta ustrezna testna dokumentacija kot **obvezen prilog izdanega paketa**.

Rezultati testiranja, izvedeni v CI/CD cevovodu, se samodejno beležijo in povezujejo z verzijo sistema.

8.4.4 Prevzemna osnova

Dokumentacija o uspešno izvedenem testiranju v okolju **PREPROD** in potrjena s strani naročnika predstavlja **formalno osnovo za izvedbo prevzema sistema oziroma njegove verzije v produkcijo**.

9 Upravljanje sprememb in vzdrževanje

9.1 Postopek upravljanja sprememb

Vsaka sprememba v informacijskem sistemu POKK mora biti obravnavana v skladu z vnaprej definiranim postopkom, ki omogoča:

- nadzorovano uvajanje sprememb,
- oceno vplivov na druge funkcionalnosti in sisteme,
- minimalizacijo tveganj za stabilnost produkcije,
- ter sledljivost odločitev in izvedenih aktivnosti.

9.1.1 Faze postopka

1. Zahteva za spremembo (RFC – Request for Change)

Sprememba se iniciira s formalizirano zahtevo, ki vključuje:

- opis spremembe in razlog zanjo,
- predviden vpliv na sistem in uporabnike,
- predlog roka izvedbe ter okolje(e), v katerih bo sprememba najprej testirana.

2. Analiza vplivov in presoje

Odgovorna tehnična oseba ali izvajalska ekipa pripravi analizo:

- vpliva na funkcionalnost, varnost, zmogljivost in povezane komponente,
- tveganj ob morebitnem neuspehu spremembe,
- potrebe po spremembah v konfiguraciji, podatkovnih strukturah ali integracijah.

3. Odobritev spremembe

Glede na naravo spremembe se uporabi ena izmed naslednjih stopenj odobritve:

- **manjše spremembe:** tehnična odobritev s strani naročnika ali pooblaščenega skrbnika,
- **večje spremembe ali spremembe poslovne logike:** formalna odobritev preko naročnikovega projektnega odbora ali pristojnega organa.

4. Izvedba spremembe

Sprememba se najprej izvede v okolju TEST ali PREPROD, kjer se preveri njena funkcionalna ustreznost in stabilnost.

Po potrditvi se sprememba prenese v produkcijo skladno s postopkom namestitve (glej poglavje 7.1).

Za vsako večjo spremembo mora biti na voljo **rollback načrt**, ki omogoča hitro povrnitev v prejšnje stanje.

5. Dokumentiranje spremembe

Vsaka sprememba mora biti:

- zabeležena v sistemu za upravljanje sprememb (npr. Git, Issue tracker, Change log),
- opremljena z identifikatorjem verzije in datumom izvedbe,
- povezana z morebitnimi spremembami v dokumentaciji (npr. konfiguracija, podatkovni model, navodila za uporabo).

6. Poizvedbeni in regresijski testi

Po vsaki spremembi se izvedejo preverjanja uspešnosti in osnovni regresijski testi, da se zagotovi:

- ohranitev obstoječe funkcionalnosti,
- ustrezno delovanje vseh odvisnih komponent.

9.1.2 Vloge in odgovornosti

Naročnik: potrjuje poslovne spremembe in usmerja prioritete.

Izvajalec: predlaga, dokumentira, implementira in testira spremembe.

Varnostni nadzornik / skrbnik CI/CD: skrbi za skladnost sprememb z varnostnimi politikami in objavo v pravilnem okolju.

Uporabnik / UAT predstavnik: potrdi pravilno delovanje spremembe v okolju PREPROD pred prenosom v produkcijo.

9.2 Načrt vzdrževanja

Vzdrževanje informacijskega sistema POKK je ključen del življenjskega cikla rešitve in zagotavlja njegovo stabilno, varno in funkcionalno delovanje po zaključku razvojne faze. Vzdrževanje mora biti organizirano tako, da omogoča pravočasno odpravo napak, uvajanje izboljšav, sledenje zakonskim spremembam in zagotavljanje odzivnosti na potrebe uporabnikov.

9.2.1 Vrste vzdrževanja

Vzdrževanje se deli na naslednje kategorije:

Vrsta vzdrževanja	Opis
Korektivno vzdrževanje	Odprava napak, ki vplivajo na pravilno delovanje sistema (npr. funkcionalne napake, varnostne ranljivosti, izpadi).
Adaptivno vzdrževanje	Prilagoditev sistema spremembam zakonodaje, okolja ali zunanjih povezav (npr. spremembe v API-jih povezanih sistemov).
Preventivno vzdrževanje	Posodobitve tehnologij, odprava tehničnega dolga, refaktoriranje in izboljšave zmogljivosti.
Evolutivno vzdrževanje	Uvajanje manjših funkcionalnih izboljšav ali dopolnitev, ki niso zajete v rednem razvojnem ciklu.

9.2.2 Organizacija vzdrževanja

Vzdrževanje mora biti organizirano kot del ločenega storitvenega sklopa z ločeno pogodbeno podlago (po potrebi).

Vzdrževalna ekipa mora imeti dostop do:

- orodij za spremljanje in analizo napak (OpenSearch, monitoring sistemi),
- okolij za preverjanje in testiranje popravkov (TEST, PREPROD),
- repozitorijev kode in konfiguracije (Git, CI/CD).

9.2.3 Odzivni časi in prioriteta obravnava

Ravnanje z napakami in zahtevki mora biti določeno po vnaprej dogovorjeni lestvici prioritet, npr.:

Rang	Opis napake	Maks. odzivni čas	Maks. čas odprave
P1 – Kritična	Onemogočeno delovanje ključnih funkcionalnosti / celoten izpad	2 uri	8 ur
P2 – Visoka	Napaka vpliva na uporabo pomembne funkcionalnosti brez obvoda	8 ur	2 delovna dneva
P3 – Srednja	Napaka z obhodno rešitvijo / mejna funkcionalnost	2 dni	5 delovnih dni
P4 – Nizka	Kozmetične napake, predlogi izboljšav	5 dni	po dogovoru / verzijsko

9.2.4 Trajanje vzdrževanja

Osnovno vzdrževanje sistema mora biti zagotavljano **najmanj 24 mesecev** po zaključku izvedbenega dela projekta oziroma produkcijskem prevzemu.

Naročnik lahko vzdrževanje razdeli med več izvajalcev, če je sistem ustrezno modulariziran in dokumentiran.

9.3 Revizijske sledi

V okviru vzdrževanja in upravljanja sprememb mora biti zagotovljena **polna revizijska sledljivost** nad vsemi aktivnostmi, ki vplivajo na stanje informacijskega sistema POKK. To vključuje tako spremembe v izvorni kodi, nameščene verzije sistema, spremembe konfiguracij kot tudi operacije vzdrževanja v produkcijskem in predprodukcijskem okolju.

9.3.1 Obvezne revizijske komponente

Revizijska sled mora zajemati:

1. Spremembe kode in konfiguracij:

- vse spremembe kode so evidentirane v **sistemu za nadzor različic (npr. Git)**,
- vsaka sprememba mora biti opremljena z ustreznim **komentarjem, identifikatorjem zahtevka (npr. issue ID)** in podatki o avtorju spremembe,
- spremembe konfiguracijskih parametrov (npr. v `.env`, `application.yaml`) morajo biti versionirane in arhivirane.

2. Spremembe v podatkovni strukturi (baza):

- SQL posodobitvene skripte so hranjene v repozitoriju in pripete k ustrezni verziji aplikacije,
- vsaka sprememba strukture sheme ali podatkov mora biti opremljena z metapodatki (datum, avtor, komentar).

3. Spremembe v konfiguraciji okolij:

- spremembe parametrov okolij (npr. spremenljivke okolja, secrets, konfiguracije infrastrukture) morajo biti dokumentirane,
- priporočljivo je centralno spremljanje konfiguracij (npr. GitOps, Helm release history).

4. Namestitve in nadgradnje:

- vsak CI/CD pipeline mora zapisovati revizijsko sled o izvedbi (čas, verzija, izvajalec, rezultat),
- izvedba SQL skript mora biti logirana (npr. preko `schema_version` tabele),
- arhivirani morajo biti vsi artefakti: binarne datoteke, konfiguracije, SQL skripte.

5. Vzdrževalne aktivnosti:

- vsaka ročna intervencija (npr. direktno na bazi, sistemski popravek, ročna konfiguracija) mora biti dokumentirana v sistemu za beleženje sprememb,
- mora vključevati: **kdo, kdaj, kaj, zakaj** in **potrjevalca spremembe**.

9.3.2 Dostopnost in hranjenje

Vsi revizijski zapisi morajo biti hranjeni **najmanj 5 let**.

Dostop do revizijskih zapisov mora biti omogočen le pooblaščenim osebam, in mora biti sam po sebi revizijsko sledljiv.

Revizijske evidence morajo biti na voljo tudi za potrebe **zunanjih revizij** in presoje skladnosti (npr. informacijska varnost, varstvo osebnih podatkov).

9.4 Komunikacija in podpora

Učinkovita in sledljiva komunikacija med naročnikom in izvajalcem je ključna za nemoteno vzdrževanje sistema, pravočasno odpravo napak ter varno uvajanje izboljšav in nadgradenj. Vzpostavljen mora biti formaliziran proces komunikacije in podpornih kanalov, ki omogoča:

- enoten vnos in spremljanje zahtevkov,
- določanje prioritete,
- ažurno obveščanje o stanju reševanja,
- arhiviranje komunikacije za potrebe revizije in spremljanja SLA.

9.4.1 Kanali komunikacije

Namen	Kanal / Orodje
Prijava napake ali težave	Sistem za podporo uporabnikom (npr. helpdesk portal, Jira Service Desk)
Tehnična eskalacija / urgentne težave	Namenski e-poštni naslov + telefonska številka za P1 incidente
Obveščanje o stanju sistema	Statusna stran ali sistem za obvestila (npr. statuspage, mailing list)
Vzdrževanje in spremembe	Issue tracking sistem s povezavo na verzije v CI/CD
Operativne komunikacije	Redni usklajevalni sestanki ali e-pošta (v naprej dogovorjeni termin)

9.4.2 Klasifikacija in odzivni časi

Vsi zahtevki morajo biti klasificirani glede na poslovni vpliv, kot je definirano v poglavju 9.2. Za vsako prioriteto mora biti določen maksimalen **odzivni čas** (čas do prvega odziva izvajalca) in **čas odprave**.

9.4.3 Dokumentiranje komunikacije

Vsa komunikacija o posameznem zahtevku mora biti vodena **centralizirano**, z zgodovino odgovorov, statusi, komentarji in spremembami.

Sporočila po elektronski pošti, klici ali druge oblike neformalnih komunikacij morajo biti, kjer je pomembno, zabeležene v sistem za spremljanje zahtevkov.

Po vsaki večji intervenciji ali zaključku zahtevka mora biti pripravljen kratek **zapisnik ali povzetek izvedenih aktivnosti**, ki je na voljo naročniku.

9.4.4 Podporna dokumentacija

Izvajalec mora vzdrževati ažurno dokumentacijo sistema (API-ji, konfiguracija, navodila), ki je dostopna preko centralnega repozitorija (npr. Confluence, Git, Wiki).

Vzpostavljen mora biti **nadzor nad verzijami dokumentacije**, z zabeležbo sprememb ob vsaki objavi nove verzije sistema.

9.4.5 Usklajevanje in pregled dela

V času vzdrževanja se izvajajo **redni operativni pregledi**, vsaj enkrat mesečno ali po dogovoru.

Ob večjih spremembah, incidentih ali nadgradnjah je treba pripraviti **poročilo o spremembi** z oceno vplivov, stanjem sistema in morebitnimi nadaljnjimi ukrepi.

10 Dokumentacija in navodila za uporabo

10.1 Predhodni načrt dokumentacije

Izvajalec mora v sklopu izvedbe informacijskega sistema POKK zagotoviti popolno tehnično in uporabniško dokumentacijo, ki mora biti dostavljena najpozneje ob zaključku razvoja posamezne funkcionalne celote oziroma produkcijskem prevzemu sistema.

10.1.1 Zahtevana dokumentacija

Dokumentacija mora pokrivati naslednje vsebinske sklope:

Vrsta dokumenta	Namen / vsebina	Uporabnik
Uporabniška navodila	Opis funkcionalnosti, navigacije, primerov uporabe, pogostih napak in rešitev	Končni uporabniki
Administratorska navodila	Upravljanje uporabnikov, vloge, konfiguracije, nadzorni vmesniki	Sistemske skrbniki
Tehnična dokumentacija	Arhitektura, API vmesniki, podatkovni modeli, konfiguracijski parametri	Tehnična ekipa naročnika
Navodila za namestitev	Namestitveni postopek, odvisnosti, skripte, primeri uporabe v CI/CD	DevOps / izvajalec
Dokument o prevzemu	Zapisniki testiranj, potrditve, verzije in podpisi	Naročnik / izvajalec

10.1.2 Oblika in dostava

Dokumenti morajo biti dostavljeni v **elektronski obliki (PDF in izvorni format)**.

Priročniki za končne uporabnike morajo biti na voljo tudi v **strukturirani spletni obliki (HTML/Markdown)**, če se zagotavlja spletna pomoč.

Vse dokumente mora izvajalec **verzionirati**, ob spremembah pa jasno označiti datum spremembe in verzijo sistema, na katero se nanašajo.

10.1.3 Končna priloga

V trenutni fazi dokumentacije bodo sezname in vsebine posameznih dokumentov **dopolnjeni v fazi PZI**, ko bodo znane konkretne implementacije, tehnologije in uporabniški profili.

10.2 Uporabniški priročniki

Poglavje se dopolni v fazi PZI in zaključi pred končnim prevzemom.

10.3 Tehnična dokumentacija

Poglavje se dopolni v fazi PZI in zaključi pred končnim prevzemom.

10.4 Navodila za namestitev

Poglavje se dopolni v fazi PZI in zaključi pred končnim prevzemom.

10.5 Vzorčni dokumenti

Poglavje se dopolni v fazi PZI in zaključi pred končnim prevzemom.

11 Priloge in reference

11.1 Seznam prilog

└─ Tehnična Specifikacija

├─ Analiza

│ └─ Drobnica

│ │ └─ CPZdrob_opis.docx

│ │ └─ CPZ_izpisi_o_tropu.docx

│ │ └─ drob_20250418(1)

│ │ │ └─ CPZ_drobnica

│ │ │ │ └─ CPZdrob_opis.docx

│ │ │ │ └─ CPZ_izpisi_o_tropu.docx

│ │ │ │ └─ razvoj.docx

│ │ │ │ └─ rejec_kmetija.pdf

│ │ │ │ └─ zival.pdf

│ │ │ └─ izpisi_html

│ │ │ │ └─ 1.2-web_aktivne_zival.pdf

│ │ │ │ └─ 1.6-kontr_Seznam_rejcev_v_obdobju.pdf

│ │ │ │ └─ 3.5-Plodnost_po_rejcih.pdf

│ │ │ │ └─ 4.12-kontr_Prirast-po_pasmah.pdf

│ │ │ │ └─ 4.13-kontr_Prirast-po_zavodih.pdf

│ │ │ │ └─ 4.7.1-Kontrola mle\237nosti.pdf

│ │ │ │ └─ 4.7.2-Kontrole pri rejcih.pdf

│ │ │ │ └─ 5.3-Poreklo_zival.pdf

│ │ │ │ └─ 6.1-Plemenjaki_z_brez_tlesnih_ocen.pdf

│ │ │ │ └─ 6.3-Povprecno_sorodstvo_s_tropom.pdf

│ │ │ │ └─ 6.4-Izracun_sorodstva.pdf

│ │ │ │ └─ 7.1-kontr_Izlocene_zivali.pdf

- | | | | └─ 8.1-Trop_PV_aktivne_zivali.pdf
- | | | | └─ izpisi_SQL.docx
- | | | └─ sifranti
- | | | └─ koze1_sif_ddl_all.sql
- | | | └─ ovce1_sif_ddl_all.sql
- | | └─ ERD_PowerDesigner2.zip
- | | └─ ERD_PowerDesigner.zip
- | | └─ izpisi
- | | | └─ 1.1-Seznam_zivali_za_rejca.pdf
- | | | └─ 1.2-web_aktivne_zival.pdf
- | | | └─ 1.3-Dopis-aktivni_rejci_aktivne_zivali.pdf
- | | | └─ 1.4-trop_podatki_o_zivalih.pdf
- | | | └─ 1.5-Potrdilo_o_reji_drobnice.pdf
- | | | └─ 1.6-kontr_Seznam_rejcev_v_obdobju.pdf
- | | | └─ 1.7-kontr_Stalez_na_dan.pdf
- | | | └─ 2.1-kontr_Ocenjevanje_zivali.pdf
- | | | └─ 3.1-Plodnost_dopis.pdf
- | | | └─ 3.2-Plodnost_po_zivalih_pri_rejcih.pdf
- | | | └─ 3.3-por_plodnost_po_letih_pri_rejcu.pdf
- | | | └─ 3.4-trop_plodnost_prirast_mladicev.pdf
- | | | └─ 3.5-Plodnost_po_rejcih.pdf
- | | | └─ 4.10-Tehtanja_ob_odstavitvi-interna_oznaka.pdf
- | | | └─ 4.11-Seznam_mladicev_v_tropu_prirast-interna_oznaka.pdf
- | | | └─ 4.12-kontr_Prirast-po_pasmah.pdf
- | | | └─ 4.13-kontr_Prirast-po_zavodih.pdf
- | | | └─ 4.1-Mlecnost_dopis.pdf
- | | | └─ 4.2-Mlecnost-delni_lakt_izracun.pdf
- | | | └─ 4.3-por_lakt_zak_cela_laktacija.pdf
- | | | └─ 4.4-Mlecnost-povp_slo.pdf
- | | | └─ 4.5-por_lakt_zak_STD_laktacija.pdf
- | | | └─ 4.6-Mlecnost-povp_slo_std_lakt.pdf
- | | | └─ 4.7.1-Kontrola mle\237nosti.pdf
- | | | └─ 4.7.2-Kontrole pri rejcih.pdf
- | | | └─ 4.7-Pregled_kontrol_pri_rejcih-posamezno.pdf
- | | | └─ 4.8-Pregled_nevnesenih_kontrol.pdf
- | | | └─ 4.9-Seznam_mladicev-tehtanja_interna_oznaka.pdf
- | | | └─ 5.1-Podatki_o_poreklu_zival.pdf
- | | | └─ 5.2-Podatki_o_poreklu_mladic.pdf
- | | | └─ 5.3-Poreklo_zival.pdf
- | | | └─ 6.1-Plemenjaki_z_brez_telesnih_ocen.pdf
- | | | └─ 6.2-Plemenjaki_za_trop.pdf

- | | | └─ 6.3-Povprecno_sorodstvo_s_tropom.pdf
- | | | └─ 6.4-Izracun_sorodstva.pdf
- | | | └─ 7.1-kontr_Izlocene_zivali.pdf
- | | | └─ 8.1-Trop_PV_aktivne_zivali.pdf
- | | | └─ 9.1-Zootehnski_dokument.pdf
- | | | └─ izpisi_SQL.docx
- | | └─ izpisi.zip
- | | └─ Ovni
- | | | └─ opis_TPP.docx
- | | | └─ TPP_izpisi.docx
- | | | └─ TPP_izracun.docx
- | | | └─ TPP_vnos_podatkov.docx
- | | └─ pripust.sql
- | | └─ procesi
- | | | └─ proces_F_delez_pasme
- | | | | └─ proces_F_delez_pasme.docx
- | | | └─ proces_F_INBR_mladic
- | | | | └─ proces_F_INBR_mladic.docx
- | | | └─ proces_F_INBR_zival
- | | | | └─ proces_F_INBR_zival.docx
- | | | └─ proces_F_lakt_izr
- | | | | └─ BdyLaktacija.sql
- | | | | └─ proces_F_lakt_izr.docx
- | | | └─ proces_F_plodnost
- | | | | └─ BdyPlodnost.sql
- | | | | └─ proces_F_plodnost.docx
- | | | └─ proces_F_povezava_mlad_zival
- | | | | └─ proces_F_povezava_mlad_zival.docx
- | | | └─ proces_F_RK_razdelek
- | | | | └─ proces_F_RK_razdelek.docx
- | | | └─ proces_F_rod_knjiga
- | | | | └─ proces_F_rod_knjiga.docx
- | | | └─ proces_LDR_hrbtna_misica
- | | | | └─ ldr_hrbtna_misica.docx
- | | | | └─ podatki_excel.xlsx
- | | | | └─ podatki_load.csv
- | | | | └─ proces_LDR_hrbtna_misica.docx
- | | | └─ proces_LDR_materinstvo
- | | | | └─ ldr_materinstvo.docx
- | | | | └─ podatki_excel.xlsx
- | | | | └─ podatki_load.csv

- | | | | └─ proces_LDR_materinstvo.docx
- | | | └─ proces_LDR_ocetovstvo
- | | | | └─ ldr_ocetovstvo.docx
- | | | | └─ podatki_excel.xlsx
- | | | | └─ podatki_load.csv
- | | | | └─ proces_LDR_ocetovstvo.docx
- | | | └─ proces_LDR_PV
- | | | | └─ podatki_load.csv
- | | | | └─ proces_LDR_PV.docx
- | | | └─ proces_LDR_tehtanja_Logatec
- | | | | └─ ldr_tehtanja.docx
- | | | | └─ podatki_excel.xlsx
- | | | | └─ podatki_load.csv
- | | | | └─ proces_LDR_tehtanja_Logatec.docx
- | | | └─ proces_LDR_TSE
- | | | | └─ ldr_tse.docx
- | | | | └─ podatki_excel.xlsx
- | | | | └─ podatki_load.csv
- | | | | └─ proces_LDR_TSE.docx
- | | | └─ proces_LDR_volna_LAB
- | | | | └─ ldr_volna_lab.docx
- | | | | └─ podatki_input.csv
- | | | | └─ podatki_load.csv
- | | | | └─ proces_LDR_volna_LAB.docx
- | | | └─ proces_LDR_volna_Logatec
- | | | | └─ ldr_volna_teza.docx
- | | | | └─ podatki_excel.xlsx
- | | | | └─ podatki_load.csv
- | | | | └─ proces_LDR_volna_Logatec.docx
- | | | └─ proces_TP
- | | | | └─ opis_TP.docx
- | | | | └─ proces_TP_IZLOCITEV.docx
- | | | | └─ proces_TP_KRI.docx
- | | | | └─ proces_TP_NAROCILNICA.docx
- | | | | └─ proces_TP_NASELITEV.docx
- | | | | └─ proces_TP_OCENE_TP.docx
- | | | | └─ proces_TP_ODKUP.docx
- | | | | └─ proces_TP_SEME.docx
- | | | | └─ proces_TP_TEHTANJE.docx
- | | | | └─ proces_TP_TRETIRANJE.docx
- | | | └─ proces_vpisi_izlocitev_mladic

- | | | | └─ proces_Vpis_izlocitve_mladica.docx
- | | | | └─ proces_vpis_izlocitev_zival
- | | | | └─ proces_Vpis_izlocitev_zival.docx
- | | | | └─ proces_vpis_jagnjitev
- | | | | └─ proces_Vpis_jagnjitev_jaritev.docx
- | | | | └─ proces_vpis_kontrola_mlecnosti
- | | | | └─ kontrole_pri_vnosu_analiz_mleka_2025 (002).pdf
- | | | | └─ mlacnost_112_apr.txt
- | | | | └─ mlacnost_format.txt
- | | | | └─ proces_Vpis_kontrole_mle\237nosti.docx
- | | | | └─ Upload_tools.sql
- | | | | └─ proces_vpis_KONTROLOR
- | | | | └─ proces_Vpis_kontrolorja.docx
- | | | | └─ proces_vpis_MOLZE
- | | | | └─ pogoji_zacetek_in_konec_molze_2025.pdf
- | | | | └─ proces_Vpis_konca_molze.docx
- | | | | └─ proces_Vpis_zacetka_molze.docx
- | | | | └─ proces_vpis_pripustov
- | | | | └─ pripust.sql
- | | | | └─ proces_Vpis_pripustov.docx
- | | | | └─ proces_vpis_REJEC
- | | | | └─ navodila_Vpis_novega_rejca.docx
- | | | | └─ proces_Vpis_rejca.docx
- | | | | └─ proces_vpis_selitev_zival
- | | | | └─ proces_vpis_selitev_zivali.docx
- | | | | └─ proces_vpis_tehtanje_mladic
- | | | | └─ proces_Vpis_tehtanje_mladicev.docx
- | | | | └─ proces_vpis_telesne_ocene
- | | | | └─ 1-proces_TP_TELESNE_OCENE_Gæ.docx
- | | | | └─ 2-proces_Vpis_telesne_ocene.docx
- | | | | └─ proces_Vpis_TELESNE_OCENE.docx
- | | | | └─ Telesne_ocene_koze.pdf
- | | | | └─ Telesne_ocene_ovce.pdf
- | | | | └─ proces_vpis_zival_IZVEN
- | | | | └─ proces_Vpis_zival_IZVEN.docx
- | | | | └─ proces_vpis_zival_ODBIRA
- | | | | └─ proces_Vpis_zival_ODBIRA.docx
- | | | | └─ proces_vpis_zival_POREKLO
- | | | | └─ proces_Vpis_zival_POREKLO.docx
- | | | | └─ proces_vpis_zival_SEME
- | | | | └─ proces_Vpis_zival_SEME.docx

- | | | └─ proces_vpis_zival_UVOZ
- | | | └─ proces_Vpis_zival_UVOZ.docx
- | | └─ procesi.zip
- | └─ GB
- | | └─ 2_GB_arhivske_tabele.pdf
- | | └─ 3_GB_monitoring_app.pdf
- | | └─ 4_GB_genetske_rezerve.pdf
- | | └─ 6.3.1_Spremljanje_kakovosti_semena.docx
- | | └─ DOPOLNITVE.docx
- | | └─ eGenskaBanka-Funkcionalna specifikacija.pdf
- | | └─ eŽiv_teh_specifikacija v2.pdf
- | | └─ GB_opisi_e_vsebin.jpg
- | | └─ protokol.docx
- | └─ KIS
- | | └─ KIS-ostalo
- | | | └─ Lokacije.docx
- | | | └─ Lokacije_vprašalnik.docx
- | | | └─ Prijava.docx
- | | | └─ Prijava_vprašalnik.docx
- | | └─ KIS - Označitev živali
- | | | └─ primer_tl_novi.docx
- | | | └─ PrimerVnosTL.bpm
- | | | └─ PrimerVnosTL.png
- | | | └─ shema_vnosa_tl_novo.drawio
- | | | └─ shema_vnosa_tl_novo.pdf
- | | └─ KIS - registracija telet
- | | | └─ primer_tl_novi.docx
- | | | └─ shema_vnosa_tl_novo.drawio
- | | | └─ shema_vnosa_tl_novo.pdf
- | └─ Kopitarji
- | | └─ crk.txt
- | | └─ Dig živ CRK za izdelavo specifikacije_ver4.doc
- | | └─ Nadgradnja_CRK_novi_moduli_rodovništva_verzija 1.docx
- | | └─ Nadgradnja modulov rodovništva_verzija 1.docx
- | | └─ Seznam izpisov.docx
- | | └─ šifrant_spustni_seznam_crk.xlsx
- | | └─ UPOR_TEH_DOK_REKO_3_2025
- | | | └─ REKO_TD_1_1.doc
- | | | └─ REKO_UN_IZPISI_1_3.doc
- | | | └─ REKO_UN_PREGLEDOVALNIKI_1_2.doc
- | | | └─ REKO_UN_PREMIKI_1_3.doc

- | | | |— REKO_UN_REGISTRACIJA_1_3.doc
- | | | |— REKO_UN_REPRODUKCIJA_1_2.doc
- | | | |— REKO_UN_RODOVNISKA_KNJIGA_1_2.doc
- | | | |— REKO_UN_STORITVE_1_1.doc
- | | | |— REKO_UN_UVOD_1_1.doc
- | | | |— REKO_UN_ZUNANJOST_1_1.doc
- | | |— UPOR_TEH_DOK_REKO_3_2025.zip
- | |— Podatkovno_skladisce
- | | |— Vprašalnik za popis poslovnega procesa v1(1)_DW_20250214(1).docx
- | |— Prašiči
- | | |— KGZS-2018-02-26 - Aplikacija za rejce KPP - Tehnicna specifikacija - V3.pdf
- | | |— prasici.txt
- | | |— proces_LOAD_abortus_v1.docx
- | | |— proces_LOAD_izločitev_v1.docx
- | | |— proces_LOAD_mesnatost_v1.docx
- | | |— proces_LOAD_NPV_AGE1F_v1.docx
- | | |— proces_LOAD_NPV_DMP_v1.docx
- | | |— proces_LOAD_NPV_IO_v1.docx
- | | |— proces_LOAD_NPV_LONG_v1.docx
- | | |— proces_LOAD_NPV_ME_v1.docx
- | | |— proces_LOAD_NPV_ML_v1.docx
- | | |— proces_LOAD_NPV_SES_v1.docx
- | | |— proces_LOAD_NPV_VG_v1.docx
- | | |— proces_LOAD_prasitev_v1.docx
- | | |— proces_LOAD_seme_v1.docx
- | | |— proces_Vnos_abortus_v1.docx
- | | |— proces_Vnos_izločitev_v1.docx
- | | |— proces_Vnos_merj_meritve_v1.docx
- | | |— proces_Vnos_odstavitev_v1.docx
- | | |— proces_Vnos_prasitev_v1.docx
- | | |— proces_Vnos_priprava_uvoz_živali_v1.docx
- | | |— proces_Vnos_priprava_uvoz_živali_v1.pdf
- | | |— proces_Vnos_pripust_v1.docx
- | | |— proces_Vnos_seme_v1.docx
- | | |— proces_Vnos_tetoviranje_v1.docx
- | | |— proces_Vnos_živali_brez_preizkusa_v1.docx
- | | |— proces_Vnos_živali_popis_črede_v1.docx
- | | |— proces_Vnos_živali_s_preizkusom_v1.docx
- | | |— proces_Vpis_NPV_ani_pest_v1.docx
- | | |— proces_Vpis_pripustov.docx
- | | |— tajnistvo@kgzs-zavodnm.si_20250221_092831.pdf

└─ Diagrami

| └─ Arhitektura - eZivin.drawio.png

| └─ eZivin - Akterji.drawio

| └─ SKL-DOG

| | └─ F-17000

| | | └─ SKL-DOG-F-17000.drawio

| | | └─ SKL-DOG-F-17000.drawio.png

| | └─ F-17100

| | | └─ SKL-DOG-F-17100.drawio

| | | └─ SKL-DOG-F-17100.drawio.png

| | └─ F-17200

| | | └─ SKL-DOG-F-17200.drawio

| | | └─ SKL-DOG-F-17200.drawio.png

| | └─ F-17300

| | | └─ SKL-DOG-F-17300.drawio

| | | └─ SKL-DOG-F-17300.drawio.png

| | └─ F-17400

| | | └─ SKL-DOG-F-17400.drawio

| | | └─ SKL-DOG-F-17400.drawio.png

| | └─ F-17500

| | | └─ SKL-DOG-F-17500.drawio

| | | └─ SKL-DOG-F-17500.drawio.png

| | └─ F-17600

| | | └─ SKL-DOG-F-17600.drawio

| | | └─ SKL-DOG-F-17600.drawio.png

| | └─ F-17700

| | | └─ SKL-DOG-F-17700.drawio

| | | └─ SKL-DOG-F-17700.drawio.png

| | └─ F-17800

| | | └─ SKL-DOG-F-17800.drawio

| | | └─ SKL-DOG-F-17800.drawio.png

| | └─ F-17900

| | | └─ SKL-DOG-F-17900.drawio

| | | └─ SKL-DOG-F-17900.drawio.png

| | └─ F-18000

| | | └─ SKL-DOG-F-18000.drawio

| | | └─ SKL-DOG-F-18000.drawio.png

| | └─ F-18100

| | | └─ SKL-DOG-F-18100.drawio

| | | └─ SKL-DOG-F-18100.drawio.png

| | └─ F-18200

| | | └─ SKL-DOG-F-18200.drawio
| | | └─ SKL-DOG-F-18200.drawio.png
| | └─ F-18300
| | | └─ SKL-DOG-F-18300.drawio
| | | └─ SKL-DOG-F-18300.drawio.png
| | └─ F-18400
| | | └─ SKL-DOG-F-18400.drawio
| | | └─ SKL-DOG-F-18400.drawio.png
| | └─ F-18500
| | | └─ SKL-DOG-F-18500.drawio
| | | └─ SKL-DOG-F-18500.drawio.png
| | └─ F-18600
| | | └─ SKL-DOG-F-18600.drawio
| | | └─ SKL-DOG-F-18600.drawio.png
| | └─ F-18700
| | | └─ SKL-DOG-F-18700.drawio
| | | └─ SKL-DOG-F-18700.drawio.png
| | └─ F-18800
| | | └─ SKL-DOG-F-18800.drawio
| | | └─ SKL-DOG-F-18800.drawio.png
| | └─ F-18900
| | └─ SKL-DOG-F-18900.drawio
| | └─ SKL-DOG-F-18900.drawio.png
| └─ SKL-DROB
| | └─ F-22000
| | | └─ SKL-DROB-F-22000.drawio
| | | └─ SKL-DROB-F-22000.drawio.png
| | └─ F-22100
| | | └─ SKL-DROB-F-22100.drawio
| | | └─ SKL-DROB-F-22100.drawio.png
| | └─ F-22200
| | | └─ SKL-DROB-F-22200.drawio
| | | └─ SKL-DROB-F-22200.drawio.png
| | └─ F-22300
| | └─ SKL-DROB-F-22300.drawio
| | └─ SKL-DROB-F-22300.drawio.png
| └─ SKL-DW
| | └─ F-7000
| | | └─ SKL-DW-F-7000.drawio
| | | └─ SKL-DW-F-7000.drawio.png
| | └─ F-7100

- | | | └─ SKL-DW-F-7000.drawio
- | | | └─ SKL-DW-F-7000.drawio.png
- | | | └─ SKL-DW-F-7100.drawio
- | | └─ F-7200
- | | | └─ SKL-DW-F-7200.drawio
- | | | └─ SKL-DW-F-7200.drawio.png
- | | └─ F-7300
- | | | └─ SKL-DW-F-7300.drawio
- | | | └─ SKL-DW-F-7300.drawio.png
- | | └─ F-7400
- | | | └─ SKL-DW-F-7400.drawio
- | | | └─ SKL-DW-F-7400.drawio.png
- | | └─ F-7500
- | | | └─ SKL-DW-F-7500.drawio
- | | | └─ SKL-DW-F-7500.drawio.png
- | | └─ F-7600
- | | | └─ SKL-DW-F-7600.drawio
- | | | └─ SKL-DW-F-7600.drawio.png
- | | └─ F-7700
- | | | └─ SKL-DW-F-7700.drawio
- | | | └─ SKL-DW-F-7700.drawio.png
- | | └─ F-7800
- | | └─ SKL-DW-F-7800.drawio
- | | └─ SKL-DW-F-7800.drawio.png
- | └─ SKL-GEN
- | | └─ F-14000
- | | | └─ SKL-GEN-F-14000.drawio
- | | | └─ SKL-GEN-F-14000.drawio.png
- | | └─ F-14100
- | | └─ SKL-GEN-F-14100.drawio
- | | └─ SKL-GEN-F-14100.drawio.png
- | └─ SKL-INF
- | | └─ F-15000
- | | └─ SKL-INF-F-15000.drawio
- | | └─ SKL-INF-F-15000.drawio.png
- | └─ SKL-KOM
- | | └─ F-5000
- | | | └─ SKL-Kom-F-5000.drawio
- | | | └─ SKL-Kom-F-5000.drawio.png
- | | └─ F-5100
- | | | └─ SKL-KOM-F-5100.drawio

- | | | └─ SKL-KOM-F-5100.drawio.png
- | | └─ F-5200
- | | | └─ SKL-KOM-F-5200.drawio
- | | | └─ SKL-KOM-F-5200.drawio.png
- | | └─ F-5300
- | | | └─ SKL-KOM-F-5300.drawio
- | | | └─ SKL-KOM-F-5300.drawio.png
- | | └─ F-5700
- | | └─ SKL-KOM-F-5700.drawio
- | | └─ SKL-KOM-F-5700.drawio.png
- | └─ SKL-PAS
- | | └─ F-12000
- | | | └─ SKL-PAS-F-12000.drawio
- | | | └─ SKL-PAS-F-12000.drawio.png
- | | └─ F-12100
- | | | └─ SKL-PAS-F-12100.drawio
- | | | └─ SKL-PAS-F-12100.drawio.png
- | | └─ F-12300
- | | | └─ SKL-PAS-F-12300.drawio
- | | | └─ SKL-PAS-F-12300.drawio.png
- | | └─ F-12400
- | | | └─ SKL-PAS-F-12400.drawio
- | | | └─ SKL-PAS-F-12400.drawio.png
- | | └─ F-12500
- | | | └─ SKL-PAS-F-12500.drawio
- | | | └─ SKL-PAS-F-12500.drawio.png
- | | └─ F-12600
- | | └─ SKL-PAS-F-12600.drawio
- | | └─ SKL-PAS-F-12600.drawio.png
- | └─ SKL-PRAS
- | | └─ F-24000
- | | | └─ SKL-PRAS-F-24000.drawio
- | | | └─ SKL-PRAS-F-24000.drawio.png
- | | └─ F-24100
- | | └─ SKL-PRAS-F-24100.drawio
- | | └─ SKL-PRAS-F-24100.drawio.png
- | └─ SKL-VS
- | | └─ F-000
- | | | └─ SKL-VS-F-000.drawio
- | | | └─ SKL-VS-F-000.drawio.png
- | | | └─ SKL-VS-F-000 Flowchart.drawio

| | | └─ SKL-VS-F-000 Flowchart.drawio.png
| | └─ F-100
| | | └─ SKL-VS-F-100.drawio
| | | └─ SKL-VS-F-100.drawio.png
| | | └─ SKL-VS-F-100 Flowchart.drawio
| | | └─ SKL-VS-F-100 Flowchart.drawio.png
| | └─ F-1100
| | | └─ SKL-VS-F-1100.drawio
| | | └─ SKL-VS-F-1100.drawio.png
| | └─ F-1200
| | | └─ SKL-VS-F-1200.drawio
| | | └─ SKL-VS-F-1200.drawio.png
| | └─ F-1300
| | | └─ SKL-VS-F-1300.drawio
| | | └─ SKL-VS-F-1300.drawio.png
| | └─ F-1400
| | | └─ SKL-VS-F-1400.drawio
| | | └─ SKL-VS-F-1400.drawio.png
| | └─ F-1500
| | | └─ SKL-VS-F-1500.drawio
| | | └─ SKL-VS-F-1500.drawio.png
| | └─ F-1600
| | | └─ SKL-VS-F-1600.drawio
| | | └─ SKL-VS-F-1600.drawio.png
| | └─ F-1700
| | | └─ SKL-VS-F-1700.drawio
| | | └─ SKL-VS-F-1700.drawio.png
| | └─ F-1800
| | | └─ SKL-VS-F-1800.drawio
| | | └─ SKL-VS-F-1800.drawio.png
| | └─ F-200
| | | └─ SKL-VS-F-200.drawio
| | | └─ SKL-VS-F-200.drawio.png
| | └─ F-2000
| | | └─ SKL-VS-F-2000.drawio
| | | └─ SKL-VS-F-2000.drawio.png
| | └─ F-2200
| | | └─ SKL-VS-F-2200.drawio
| | | └─ SKL-VS-F-2200.drawio.png
| | └─ F-2300
| | | └─ SKL-VS-F-2300.drawio

| | | └─ SKL-VS-F-2300.drawio.png
| | └─ F-2400
| | | └─ SKL-VS-F-2400.drawio
| | | └─ SKL-VS-F-2400.drawio.png
| | └─ F-2500
| | | └─ SKL-VS-F-2500.drawio
| | | └─ SKL-VS-F-2500.drawio.png
| | └─ F-2700
| | | └─ SKL-VS-F-2700.drawio
| | | └─ SKL-VS-F-2700.drawio.png
| | └─ F-2800
| | | └─ SKL-VS-F-2800.drawio
| | | └─ SKL-VS-F-2800.drawio.png
| | └─ F-2900
| | | └─ SKL-VS-F-2900.drawio
| | | └─ SKL-VS-F-2900.drawio.png
| | └─ F-300
| | | └─ SKL-VS-F-300.drawio
| | | └─ SKL-VS-F-300.drawio.png
| | └─ F-400
| | | └─ SKL-VS-F-400.drawio
| | | └─ SKL-VS-F-400.drawio.png
| | └─ F-500
| | | └─ SKL-VS-F-500.drawio
| | | └─ SKL-VS-F-500.drawio.png
| | └─ F-600
| | | └─ SKL-VS-F-600.drawio
| | | └─ SKL-VS-F-600.drawio.png
| | └─ F-700
| | | └─ SKL-VS-F-600.drawio
| | | └─ SKL-VS-F-700.drawio.png
| | └─ F-900
| | └─ SKL-VS-F-900.drawio
| | └─ SKL-VS-F-900.drawio.png
| └─ SKL-ZIV
| | └─ F-10000
| | | └─ SKL-ZIV-F-10000.drawio
| | | └─ SKL-ZIV-F-10000.drawio.png
| | └─ F-10100
| | | └─ SKL-ZIV-F-10100.drawio
| | | └─ SKL-ZIV-F-10100.drawio.png

- | | └─ F-10200
- | | | └─ SKL-ZIV-F-10200.drawio
- | | | └─ SKL-ZIV-F-10200.drawio.png
- | | └─ F-10300
- | | | └─ SKL-ZIV-F-10300.drawio
- | | | └─ SKL-ZIV-F-10300.drawio.png
- | | └─ F-10400
- | | | └─ SKL-ZIV-F-10400.drawio
- | | | └─ SKL-ZIV-F-10400.drawio.png
- | | └─ F-10500
- | | | └─ SKL-ZIV-F-10500.drawio
- | | | └─ SKL-ZIV-F-10500.drawio.png
- | | └─ F-10600
- | | | └─ SKL-ZIV-F-10600.drawio
- | | | └─ SKL-ZIV-F-10600.drawio.png
- | | └─ F-10700
- | | | └─ SKL-ZIV-F-10700.drawio
- | | | └─ SKL-ZIV-F-10700.drawio.png
- | | └─ F-10800
- | | | └─ SKL-ZIV-F-10800.drawio
- | | | └─ SKL-ZIV-F-10800.drawio.png
- | | └─ F-10900
- | | | └─ SKL-ZIV-F-10900.drawio
- | | | └─ SKL-ZIV-F-10900.drawio.png
- | | └─ F-11000
- | | | └─ SKL-ZIV-F-11000.drawio
- | | | └─ SKL-ZIV-F-11000.drawio.png
- | | └─ F-11100
- | | | └─ SKL-ZIV-F-11100.drawio
- | | | └─ SKL-ZIV-F-11100.drawio.png
- | └─ Slike
- | | └─ Arhitektura - eZivin.drawio
- | | └─ Arhitektura - eZivin.drawio.png
- | | └─ Arhitektura - HighLevel.drawio
- | | └─ Arhitektura - HighLevel.drawio.png
- | | └─ eZivin - Akterji.drawio.png
- └─ POKK_Tehnicna_Specifikacija.odt
- └─ eČebele-Funkcionalna specifikacija v4.pdf
- └─ eGenskaBanka-Funkcionalna specifikacija.pdf
- └─ eŽiv_teh_specifikacija v2.pdf
- └─ Maske

- | └─ POKKadm01.pdf
- | └─ POKKdrobnica03.pdf
- | └─ POKKprasici03.pdf
- | └─ SiteMap_POKK01.ods
- └─ Nadgradnja_CRK_novi_moduli_rodovništva_verzija 1.docx
- └─ SiteMap_POKK01.ods
- └─ _Smernice
 - └─ Genericne_teholoske_zhteve_GTZ_2.3.pdf
 - └─ Podprta-programska-oprema-DRO_V1.0.pdf
 - └─ Smernice_JN_IT.pdf

152 directories, 417 files

Poglavje se še dopolni v fazi PZI in zaključi pred končnim prevzemom.

11.2 Reference

Poglavje se še dopolni v fazi PZI in zaključi pred končnim prevzemom.

Ta tehnična specifikacija se pri določanju zahtev in oblikovanju rešitev sklicuje na naslednje izvirne dokumente in standarde:

11.2.1 Dokumenti

Oznaka	Naziv dokumenta	Opis / namen
[R1]	403-3-2025-4 Tehnične specifikacije.docx	Osnovni funkcionalni in tehnični opis zahtev za javno naročilo.
[R2]	Genericne_teholoske_zhteve_GTZ_2.3.pdf	Metodološki okvir za tehnično načrtovanje rešitev v javnem sektorju.
[R3]	Smernice_JN_IT.pdf	Smernice za pripravo tehničnih specifikacij v okviru postopkov javnega naročanja.
[R4]	Podprta-programska-oprema-DRO_V1.0.pdf	Seznam dovoljenih tehnologij in orodij v informacijskih rešitvah DRO.

11.2.2 Standardi

Oznaka	Standard	Uporaba
[S1]	ISO/IEC 27002:2022	Okvir za upravljanje informacijske varnosti in določanje tehničnih kontrol.
[S2]	OWASP TOP 10	Priporočila za zaščito spletnih aplikacij pred najpogostejšimi varnostnimi ranljivostmi.
[S3]	TLS 1.3	Kriptografski standard za zaščito komunikacij v tranzitu.
[S4]	OAuth 2.0 / JWT	Standard za varno avtentikacijo in avtorizacijo uporabnikov in sistemov.
[S5]	ZVOP-2	Zakon o varstvu osebnih podatkov – skladnost s pravnim okvirom obdelave osebnih podatkov.

11.3 Normativne povezave

V nadaljevanju so navedeni dokumenti, standardi in pravni akti, ki predstavljajo obvezno normativno podlago za načrtovanje, izvedbo, vzdrževanje in presojanje skladnosti informacijskega sistema POKK. Navedene povezave so za izvajalca zavezujoče, razen če je v besedilu tehnične specifikacije izrecno določeno drugače.

Oznaka	Naziv	Vrsta zahteve	Področje uporabe
[N1]	<i>Genericne_tehnoske_zhteve_GTZ_2.3.pdf</i>	Obvezna metodologija	Tehnična arhitektura, modularnost, revizijska sledljivost
[N2]	<i>Podprta-programska-oprema-DRO_V1.0.pdf</i>	Omejitev dovoljenih tehnologij	Izbor programskih orodij in komponent
[N3]	<i>ISO/IEC 27002:2022</i>	Standard informacijske varnosti	Upravljanje varnostnih kontrol in postopkov
[N4]	<i>ZVOP-2 (Zakon o varstvu osebnih podatkov)</i>	Zakonska obveznost	Obdelava in zaščita osebnih podatkov
[N5]	<i>TLS 1.3</i>	Obvezen protokol	Varnost komunikacijskih povezav
[N6]	<i>OAuth 2.0 / JWT</i>	Obvezen mehanizem	Avtentikacija in avtorizacija uporabnikov in sistemov
[N7]	<i>OWASP TOP 10</i>	Priporočilo z obvezno uporabo	Varnost spletnih aplikacij in API-jev

11.4 Tehnični in vizualni materiali

Poglavje se dopolni v fazi PZI in zaključi pred končnim prevzemom.

12 Ocena del potrebnih za izvedbo

12.1 Izhodišča za ocenjevanje

Pri oceni predvidenih del smo pri vseh postavkah upoštevali celoten postopek izdelave aplikacije in sicer po fazah:

1. Faza analize (del analize in rešitve predstavlja ta dokument, dodatno se izvede prilagoditev glede na tehnologijo in končno arhitekturo).
2. Faza oblikovanja rešitve –design arhitekture (del analize in rešitve predstavlja ta dokument, dodatno se izvede prilagoditev glede na tehnologijo in končno arhitekturo).
3. Razvojna faza.
4. Testiranje
5. Dokumentiranje - tehnična dokumentacija z nadgrajenimi dokumenti o podatkovnem modelu,

opisu tabel, podatkovnih diagramih, namestitvena in uporabniška dokumentacija.

6. Prehod v testno okolje.

7. Projektno vodenje – skozi celoten projekt.

12.2 Vrstni red izvedbe razvoja

Glede na izbrano rešitev in tehnologijo se razvije podatkovni model. Na podlagi podatkovnega modela se razvija aplikativni del, integracije in podatkovno skladišče. Iz razvitega podatkovnega skladišča se pripravijo poročila in dokumenti. Poročila se umesti v CMS (na portal).

12.3 Ocena del

Sklop razvoja rešitve
Varnostna shema
Aplikacije in integracije
Mobilna aplikacija z off-line načinom dela
Poročila in dokumenti (cca 100 poročil in 100 dokumentov)

Izvajalec pri oceni del upošteva zgoraj omenjene sklope in ocenjeno število poročil ter dokumentov.

Konec dokumenta