

Univerzitet u Beogradu
Fakultet organizacionih nauka
Laboratorija za softversko inženjerstvo

Seminarski rad iz predmeta

Projektovanje softvera

Tema: Softverski sistem za evidenciju iznajmljenih
nekretnina

Profesor:
dr Siniša Vlajić

Student:
Karalić Irena 418/13

Beograd, 2018.

Sadržaj

1.	Korisnički zahtevi.....	5
1.1.	Verbalni opis	5
1.2.	Slučajevi korišćenja	6
1.2.1.	SK1: Slučaj korišćenja – Unos nekretnine	7
1.2.2.	SK2: Slučaj korišćenja – Izmena nekretnine	8
1.2.3.	SK3: Slučaj korišćenja – Pretraga nekretnina	9
1.2.4.	SK4: Slučaj korišćenja – Unos vlasnika.....	10
1.2.5.	SK5: Slučaj korišćenja – Unos klijenta	11
1.2.6.	SK6: Slučaj korišćenja – Izmena klijenta.....	12
1.2.7.	SK7: Slučaj korišćenja – Pretraga klijenta	13
1.2.8.	SK8: Slučaj korišćenja – Unos zaduženja	14
1.2.9.	SK9: Slučaj korišćenja – Izmena zaduženja	15
1.2.10.	SK10: Slučaj korišćenja – Pretraga zaduženje	16
1.2.11.	SK11: Slučaj korišćenja – Pretraga vlasnika	17
1.2.12.	SK12: Slučaj korišćenja – Prijava korisnika	18
2.	Analiza	19
2.1	Ponašanje softverskog sistema – sistemski dijagrami sekvenci.....	19
2.1.1.	DS1: Dijagram sekvenci slučaja korišćenja – Unos nekretnine	19
2.1.2.	DS2: Dijagram sekvenci slučaja korišćenja – Izmena nekretnine	21
2.1.3.	DS3: Dijagram sekvenci slučaja korišćenja – Pretraga nekretnina.....	24
2.1.4.	DS4: Dijagram sekvenci slučaja korišćenja – Unos vlasnika.....	25
2.1.5.	DS5: Dijagram sekvenci slučaja korišćenja – Unos klijenta	26
2.1.6.	DS6: Dijagram sekvenci slučaja korišćenja – Izmena klijenta.....	27
2.1.7.	DS7: Dijagram sekvenci slučaja korišćenja – Pretraga klijenata.....	30
2.1.8.	DS8: Dijagram sekvenci slučaja korišćenja – Unos zaduženja	31
2.1.9.	DS9: Dijagram sekvenci slučaja korišćenja – Izmena zaduženja	34
2.1.10.	DS10: Dijagram sekvenci slučaja korišćenja – Pretraga zaduženja	37
2.1.11.	DS11: Dijagram sekvenci slučaja korišćenja – Pretraga vlasnika	38
2.1.12.	DS12: Dijagram sekvenci slučaja korišćenja – Prijava korisnika	39
2.2	Definisanje ugovora o sistemskim operacijama.....	40
2.2.1.	Ugovor UG1: VratiListuGradova	40
2.2.2.	Ugovor UG2: PretraziVlasnike	40
2.2.3.	Ugovor UG3: ZapamtiNekretninu	40
2.2.4.	Ugovor UG4: PretraziNekretnine	40
2.2.5.	Ugovor UG5: VratiPodatkeONekretnini.....	40
2.2.6.	Ugovor UG6: IzmeniNekretninu	40
2.2.7.	Ugovor UG7: ZapamtiVlasnika	41

2.2.8. Ugovor UG8: ZapamtiKlijenta.....	41
2.2.9. Ugovor UG9: PretraziKlijente	41
2.2.10. Ugovor UG10: VратиPodatkeOKlijentu(Klijent)	41
2.2.11. Ugovor UG11: IzmeniKlijenta	41
2.2.12. Ugovor UG12: ZapamtiZaduzenje.....	41
2.2.13. Ugovor UG13: PretraziZaduzenja	42
2.2.14. Ugovor UG14: VратиPodatkeOZaduzenju	42
2.2.15. Ugovor UG15: IzmeniZaduzenje	42
2.2.16. Ugovor UG18: PronadjiKorisnika	42
2.3 Struktura softverskog sistema – Konceptualni (domenski) model	43
2.4 Struktura softverskog sistema – Relacioni model.....	45
3. Projektovanje.....	48
3.1 Arhitektura softverskog sistema	48
3.2 Projektovanje korisničkog interfejsa	50
3.2.1. Projektovanje ekranskih formi	51
SK1: Slučaj korišćenja – Unos nekretnine.....	52
SK2: Slučaj korišćenja – Izmena nekretnine	55
SK3:Slučaj korišćenja – Pretraga nekretnina	59
SK4: Slučaj korišćenja – Unos vlasnika	61
SK5:Slučaj korišćenja – Unos klijenta.....	63
SK6: Slučaj korišćenja – Izmena klijenta	65
SK8: Slučaj korišćenja – Unos zaduženja	71
SK9: Slučaj korišćenja – Izmena zaduženja.....	76
SK10: Slučaj korišćenja – Pretraga zaduženje.....	79
SK11: Slučaj korišćenja – Pretraga vlasnika	81
SK12: Slučaj korišćenja – Prijava korisnika	83
3.3 Projektovanje aplikacione logike	84
3.3.1 Komunikacija sa klijentima	84
3.3.2 Kontroler aplikacione logike.....	87
3.4 Projektovanje strukture softverskog sistema – Domenske klase	88
3.5 Projektovanje strukture softverskog sistema – Sistemske operacije	89
Ugovor UG1: VратиListuGradova	89
Ugovor UG2: PretraziVlasnike.....	89
Ugovor UG3: ZapamtiNekretninu	90
Ugovor UG4: PretraziNekretnine	90
Ugovor UG5: VратиPodatkeONekretnini.....	90
Ugovor UG6: IzmeniNekretninu	91
Ugovor UG7: ZapamtiVlasnika	91
Ugovor UG8: ZapamtiKlijenta.....	91

Ugovor UG9: PretraziKlijente	92
Ugovor UG10: VratiPodatkeOKlijentu(Klijent).....	92
Ugovor UG11: IzmeniKlijenta.....	92
Ugovor UG12: ZapamtiZaduzenje	93
Ugovor UG13: PretraziZaduzenja	93
Ugovor UG14: VratiPodatkeOZaduzenju.....	93
Ugovor UG15: IzmeniZaduzenje.....	94
Ugovor UG18: PronadjiKorisnika.....	94
3.6 Projektovanje aplikacione logike – Broker baze podataka.....	95
3.7 Projektovanje skladišta podataka.....	98
5.Testiranje.....	102
6.Literatura.....	102

1. Korisnički zahtevi

1.1. Verbalni opis

Potrebno je napraviti softverski sistem za agenciju koji obezbeđuje evidenciju iznajmljenih nekretnina, kao i njihovo zaduživanje od strane klijenata agencije.

Sistem treba da omogući unos novih nekretnina, izmenu izabranih nekretnina i pretragu nekretnina po određenim kriterijumima pretraživanja.

Moguće je izvršiti unos novih klijenata agencije, izmenu postojećih i pretragu klijenata po određenom kriterijumu. Svaki klijent može zadužiti više nekretnina, pa je o tome potrebno voditi detaljniju evidenciju.

Neophodno je obezbediti unos i pretragu vlasnika odgovarajućih nekretnina.

Da bi korisnik mogao da vrši izmene podataka u sistemu, potrebno je da bude prijavljen na sistem, pa sistem treba da obezbedi logovanje korisnika. Da bi se prijavio, korisnik mora da unese tačne podatke kako bi se izvršila autentifikacija, tj. sistem mora da prepozna korisnika.

1.2. Slučajevi korišćenja

U konkretnom slučaju identifikovani su sledeći slučajevi korišćenja:

1. Unos nekretnike
2. Izmena nekretnine
3. Pretraga nekretnine
4. Unos vlasnika nekretnine
5. Unos klijenta
6. Izmena klijenta
7. Pretraga klijenta
8. Unos zaduženja (složen)
9. Izmena zaduženja
10. Pretraga zaduženja
11. Pretraga vlasnika
12. Prijava korisnika



Slika 1. Dijagram slučajeva korišćenja

1.2.1. SK1: Slučaj korišćenja – Unos nekretnine

Naziv SK

Unos nekretnine

Aktori SK

Korisnik

Učesnici SK

Korisnik i sistem (program)

Preduslov: **Sistem** je uključen i **korisnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa nekretninama. Učitana je lista gradova.

Osnovni scenario SK

1. **Korisnik** unosi vrednost po kojoj pretražuje vlasnike. (APUSO)
2. **Korisnik** poziva **sistem** da nađe vlasnike po zadatoj vrednosti. (APSO)
3. **Sistem** traži vlasnike po zadatoj vrednosti. (SO)
4. **Sistem** prikazuje **korisniku** pronađene vlasnike i poruku: „**Sistem** je našao vlasnike po zadatoj vrednosti”. (IA)
5. **Korisnik** bira vlasnika i unosi podatke o nekretnini. (APUSO)
6. **Korisnik** kontroliše da li je korektno uneo podatke o nekretnini. (ANSO)
7. **Korisnik** poziva **sistem** da zapamti podatke o novoj nekretnini. (APSO)
8. **Sistem** pamti podatke o nekretnini. (SO)
9. **Sistem** prikazuje **korisniku** poruku: „Sistem je zapamtio nekretninu”. (IA)

Alternativna scenarija

- 4.1 Ukoliko **sistem** ne može da nađe vlasnike on prikazuje **korisniku** poruku: „**Sistem** ne može da nađe vlasnike po zadatoj vrednosti“. Prekida se izvršenje scenarija. (IA)
- 9.1 Ukoliko **sistem** ne može da zapamti podatke o novoj nekretnini on prikazuje **korisniku** poruku: „**Sistem** ne može da zapamti nekretninu“. (IA)

1.2.2. SK2: Slučaj korišćenja – Izmena nekretnine

Naziv SK

Izmena nekretnine

Aktori SK

Korisnik

Učesnici SK

Korisnik i sistem (program)

Preduslov: **Sistem** je uključen i **korisnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa nekretninama.

Osnovni scenario SK

1. **Korisnik** unosi vrednost po kojoj pretražuje nekretnine. (APUSO)
2. **Korisnik** poziva **sistem** da nađe nekretnine po zadatoj vrednosti. (APSO)
3. **Sistem** traži nekretnine po zadatoj vrednosti. (SO)
4. **Sistem** prikazuje **korisniku** pronađene nekretnine i poruku: „**Sistem** je našao nekretnine po zadatoj vrednosti”. (IA)
5. **Korisnik** bira nekretninu. (APUSO)
6. **Korisnik** poziva **sistem** da vrati podatke o nekretnini. (APSO)
7. **Sistem** traži podatke o nekretnini. (SO)
8. **Sistem** prikazuje **korisniku** podatke o nekretnini i poruku: „**Sistem** je pronašao podatke o izabranoj nekretnini”. (IA)
9. **Korisnik** menja podatke o nekretnini. (APUSO)
10. **Korisnik** kontroliše da li je korektno uneo podatke o nekretnini. (ANSO)
11. **Korisnik** poziva **sistem** da zapamti podatke o nekretnini. (APSO)
12. **Sistem** pamti podatke o nekretnini. (SO)
13. **Sistem** prikazuje **korisniku** poruku: „Sistem je izmenio nekretninu”. (IA)

Alternativna scenarija

- 4.1 Ukoliko **sistem** ne može da nađe nekretninu on prikazuje **korisniku** poruku: „Sistem ne može da nađe nekretnine po zadatoj vrednosti”. Prekida se izvršenje scenarija. (IA)
- 8.1 Ukoliko **sistem** ne može da nađe podatke o nekretnini, obaveštava **korisnika** o tome: „**Sistem** ne može da nađe podatke o izabranoj nekretnini”. Prekida se izvršenje scenarija. (IA)
- 13.1 Ukoliko **sistem** ne može da zapamti izmenjene podatke o nekretnini on prikazuje **korisniku** poruku: „**Sistem** ne može da izmeni nekretninu”. (IA)

1.2.3. SK3:Slučaj korišćenja – Pretraga nekretnina

Naziv SK

Pretraga nekretnina

Aktori SK

Korisnik

Učesnici SK

Korisnik i sistem (program)

Preduslov: **Sistem** je uključen i **korisnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa nekretninama.

Osnovni scenario SK

1. **Korisnik** unosi vrednost po kojoj pretražuje nekretnine. (APUSO)
2. **Korisnik** poziva **sistem** da nađe nekretnine po zadatoj vrednosti. (APSO)
3. **Sistem** traži nekretnine po zadatoj vrednosti. (SO)
4. **Sistem** prikazuje **korisniku** podatke o nekretninama i poruku: „**Sistem** je našao nekretnine po zadatoj vrednosti“. (IA)

Alternativna scenarija

- 4.1 Ukoliko **sistem** ne može da nađe nekretnine on prikazuje **korisniku** poruku: „**Sistem** ne može da nađe nekretnine po zadatoj vrednosti“. (IA)

1.2.4. SK4: Slučaj korišćenja – Unos vlasnika

Naziv SK

Unos vlasnika

Aktori SK

Korisnik

Učesnici SK

Korisnik i sistem (program)

Preduslov: **Sistem** je uključen i **korisnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa vlasnicima.

Osnovni scenario SK

1. **Korisnik** unosi podatke o vlasniku. (APUSO)
2. **Korisnik** kontroliše da li je korektno uneo podatke o vlasniku. (ANSO)
3. **Korisnik** poziva **sistem** da zapamti podatke o vlasnika. (APSO)
4. **Sistem** pamti podatke o vlasniku. (SO)
5. **Sistem** prikazuje **korisniku** poruku: „**Sistem** je zapamtio vlasnika“. (IA)

Alternativna scenarija

- 5.1 Ukoliko **sistem** ne može da zapamti podatke o novom vlasniku on prikazuje **korisniku** poruku: „**Sistem** ne može da zapamti novog vlasnika“. (IA)

1.2.5. SK5:Slučaj korišćenja – Unos klijenta

Naziv SK

Unos klijenta

Aktori SK

Korisnik

Učesnici SK

Korisnik i sistem (program)

Preduslov: **Sistem** je uključen i **korisnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa klijentima.

Osnovni scenario SK

1. **Korisnik** unosi podatke o klijentu. (APUSO)
2. **Korisnik** kontroliše da li je korektno uneo podatke o klijentu. (ANSO)
3. **Korisnik** poziva **sistem** da zapamti podatke o klijentu. (APSO)
4. **Sistem** pamti podatke o klijentu. (SO)
5. **Sistem** prikazuje **korisniku** poruku: „Sistem je zapamtio klijenta“. (IA)

Alternativna scenarija

- 5.1 Ukoliko **sistem** ne može da zapamti podatke o novom klijentu on prikazuje **korisniku** poruku: „**Sistem** ne može da zapamti klijenta“. (IA)

1.2.6. SK6: Slučaj korišćenja – Izmena klijenta

Naziv SK

Izmenaklijenta

Aktori SK

Korisnik

Učesnici SK

Korisnik i sistem (program)

Preduslov: **Sistem** je uključen i **korisnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa klijentom. Učitana je lista gradova.

Osnovni scenario SK

1. **Korisnik** unosi vrednost po kojoj pretražuje klijente. (APUSO)
2. **Korisnik** poziva **sistem** da nađe klijente po zadatoj vrednosti. (APSO)
3. **Sistem** traži klijente po zadatoj vrednosti. (SO)
4. **Sistem** prikazuje **korisniku** pronađene klijente i poruku: „**Sistem** je našao klijente po zadatoj vrednosti“. (IA)
5. **Korisnik** bira klijenta. (APUSO)
6. **Korisnik** poziva **sistem** da vrati podatke o klijentu. (APSO)
7. **Sistem** traži podatke o klijentu. (SO)
8. **Sistem** prikazuje **korisniku** podatke o klijentu i poruku: „**Sistem** je pronašao podatke o izabranom klijentu“. (IA)
9. **Korisnik** unosi (menja) podatke o klijentu. (APUSO)
10. **Korisnik** kontroliše da li je korektno uneo podatke o klijentu. (ANSO)
11. **Korisnik** poziva **sistem** da zapamti podatke o klijentu. (APSO)
12. **Sistem** pamti podatke o klijentu. (SO)
13. **Sistem** prikazuje **korisniku** poruku: „Sistem je zapamtio klijenta“. (IA)

Alternativna scenarija

- 4.1 Ukoliko **sistem** ne može da nađe klijente prikazuje **korisniku** poruku: „Sistem ne može da nađe klijente po zadatoj vrednosti“. Prekida se izvršenje scenarija. (IA)
- 8.1 Ukoliko **sistem** ne može da nađe podatke o klijentu, obaveštava **korisnika** o tome: „**Sistem** ne može da nađe podatke o izabranom klijentu“. Prekida se izvršenje scenarija. (IA)
- 13.1 Ukoliko **sistem** ne može da zapamti izmenjene podatke o klijentu on prikazuje **korisniku** poruku: „**Sistem** ne može da izmeni klijenta“. (IA)

1.2.7. SK7: Slučaj korišćenja– Pretraga klijenta

Naziv SK

Pretraga klijenta

Aktori SK

Korisnik

Učesnici SK

Korisnik i sistem (program)

Preduslov: **Sistem** je uključen i **korisnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa klijentima.

Osnovni scenario SK

1. **Korisnik** unosi vrednost po kojoj pretražuje klijente. (APUSO)
2. **Korisnik** poziva **sistem** da nađe klijente po zadatoj vrednosti. (APSO)
3. **Sistem** traži klijente po zadatoj vrednosti. (SO)
4. **Sistem** prikazuje **korisniku** podatke o klijentima i poruku: „**Sistem** je našao klijente po zadatoj vrednosti“. (IA)

Alternativna scenarija

- 4.1 Ukoliko **sistem** ne može da nađe klijente on prikazuje **korisniku** poruku: „**Sistem** ne može da nađe klijente po zadatoj vrednosti“. (IA)

1.2.8. SK8: Slučaj korišćenja – Unos zaduženja

Naziv SK

Unos zaduženja

Aktori SK

Korisnik

Učesnici SK

Korisnik i sistem (program)

Preduslov: **Sistem** je uključen i **korisnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa zaduženjima.

Osnovni scenario SK

1. **Korisnik** unosi vrednost po kojoj pretražuje klijente. (APUSO)
2. **Korisnik** poziva **sistem** da nađe klijente po zadatoj vrednosti. (APSO)
3. **Sistem** traži klijente po zadatoj vrednosti. (SO)
4. **Sistem** prikazuje **korisniku** podatke o klijentima i poruku: „**Sistem** je našao klijente po zadatoj vrednosti“. (IA)
5. **Korisnik** unosi vrednost po kojoj pretražuje nekretnine. (APUSO)
6. **Korisnik** poziva **sistem** da nađe nekretnine po zadatoj vrednosti. (APSO)
7. **Sistem** traži nekretnine po zadatoj vrednosti. (SO)
8. **Sistem** prikazuje **korisniku** podatke o nekretninama i poruku: „**Sistem** je našao nekretnine po zadatoj vrednosti“. (IA)
9. **Korisnik** vrši izbor vlasnika, nekretnine i unosi ostale podatke o zaduženju. (APUSO)
10. **Korisnik** kontroliše da li je uneo zaduženja za izabrane nekretnine. (ANSO)
11. **Korisnik** poziva **sistem** da zapamti podatke o zaduženju. (APSO)
12. **Sistem** pamti podatke o zaduženjima. (SO)
13. **Sistem** prikazuje **korisniku** poruku: „**Sistem** je zapamtio zaduženje“. (IA)

Alternativna scenarija

- 4.1 Ukoliko **sistem** ne može da nađe klijente on prikazuje **korisniku** poruku: „**Sistem** ne može da nađe klijente po zadatoj vrednosti“. (IA)
- 8.1 Ukoliko **sistem** ne može da nađe nekretnine on prikazuje **korisniku** poruku: „**Sistem** ne može da nađe nekretnine po zadatoj vrednosti“. (IA)
- 13.1 Ukoliko **sistem** ne može da zapamti podatke o zaduženjima, on prikazuje **korisniku** poruku „**Sistem** ne može da zapamti zaduženje“. (IA)

1.2.9. SK9: Slučaj korišćenja – Izmena zaduženja

Naziv SK

Izmena zaduženja

Aktori SK

Korisnik

Učesnici SK

Korisnik i sistem (program)

Preduslov: **Sistem** je uključen i **korisnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa zaduženjem.

Osnovni scenario SK

1. **Korisnik** unosi vrednost po kojoj pretražuje zaduženja. (APUSO)
2. **Korisnik** poziva **sistem** da nađe zaduženja po zadatoj vrednosti. (APSO)
3. **Sistem** traži zaduženja po zadatoj vrednosti. (SO)
4. **Sistem** prikazuje **korisniku** pronađena zaduženja i poruku: „**Sistem** je našao zaduženja po zadatoj vrednosti”. (IA)
5. **Korisnik** bira zaduženje. (APUSO)
6. **Korisnik** poziva **sistem** da vrati podatke o zaduženju. (APSO)
7. **Sistem** traži podatke o zaduženju. (SO)
8. **Sistem** prikazuje **korisniku** podatke o zaduženju i poruku: „**Sistem** je pronašao podatke o izabranom zaduženju”. (IA)
9. **Korisnik** menja podatke o zaduženju. (APUSO)
10. **Korisnik** kontroliše da li je korektno uneo podatke o zaduženju. (ANSO)
11. **Korisnik** poziva **sistem** da zapamti podatke o zaduženju. (APSO)
12. **Sistem** pamti podatke o zaduženju. (SO)
13. **Sistem** prikazuje **korisniku** poruku: „Sistem je izmenio zaduženje”. (IA)

Alternativna scenarija

4.1 Ukoliko **sistem** ne može da nađe zaduženja on prikazuje **korisniku** poruku: „Sistem ne može da nađe zaduženja po zadatoj vrednosti”. Prekida se izvršenje scenarija. (IA)

8.1 Ukoliko **sistem** ne može da nađe podatke o zaduženju, obaveštava **korisnika** o tome: „**Sistem** ne može da nađe podatke o izabranom zaduženju”. Prekida se izvršenje scenarija. (IA)

13.1 Ukoliko **sistem** ne može da zapamti izmenjene podatke o zaduženju on prikazuje **korisniku** poruku: „**Sistem** ne može da izmeni zaduženje”. (IA)

1.2.10. SK10: Slučaj korišćenja – Pretraga zaduženja

Naziv SK

Pretraga zaduženja

Aktori SK

Korisnik

Učesnici SK

Korisnik i sistem (program)

Preduslov: **Sistem** je uključen i **korisnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa zaduženjima.

Osnovni scenario SK

1. **Korisnik** unosi vrednost po kojoj pretražuje zaduženja. (APUSO)
2. **Korisnik** poziva **sistem** da nađe zaduženja po zadatoj vrednosti. (APSO)
3. **Sistem** traži zaduženja po zadatoj vrednosti. (SO)
4. **Sistem** prikazuje **korisniku** podatke o zaduženjima i poruku: „**Sistem** je našao zaduženja po zadatoj vrednosti“. (IA)

Alternativna scenarija

- 4.1 Ukoliko **sistem** ne može da nađe zaduženja on prikazuje **korisniku** poruku: „**Sistem** ne može da nađe zaduženja po zadatoj vrednosti“. (IA)

1.2.11. SK11: Slučaj korišćenja – Pretraga vlasnika

Naziv SK

Pretraga vlasnika

Aktori SK

Korisnik

Učesnici SK

Korisnik i sistem (program)

Preduslov: **Sistem** je uključen i **korisnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa vlasnicima.

Osnovni scenario SK

1. **Korisnik** unosi vrednost po kojoj pretražuje vlasnike. (APUSO)
2. **Korisnik** poziva **sistem** da nađe vlasnike po zadatoj vrednosti. (APSO)
3. **Sistem** traži vlasnike po zadatoj vrednosti. (SO)
4. **Sistem** prikazuje **korisniku** podatke o vlasnicima i poruku: „**Sistem** je našao vlasnike po zadatoj vrednosti“. (IA)

Alternativna scenarija

- 4.1 Ukoliko **sistem** ne može da nađe vlasnike on prikazuje **korisniku** poruku: „**Sistem** ne može da nađe vlasnike po zadatoj vrednosti“. (IA)

1.2.12. SK12: Slučaj korišćenja – Prijava korisnika

Naziv SK

Prijava korisnika

Aktori SK

Korisnik

Učesnici SK

Korisnik i sistem (program)

Preduslov: Sistem je uključen i prikazuje formu za prijavljivanje **korisnika**.

Osnovni scenario SK

1. **Korisnik** unosi podatke za autentifikaciju **korisnika**. (APUSO)
2. **Korisnik** poziva sistem da pronade **korisnika** sa zadatim podacima. (APSO)
3. **Sistem** pretražuje **korisnika**. (SO)
4. **Sistem** prikazuje poruku: "Uspešno ste se prijavili na sistem". (IA)

Alternativna scenarija

- 4.1. Ukoliko **sistem** ne može da nađe **korisnika** prikazuje poruku "**Sistem** ne može da vas prijavi". (IA)

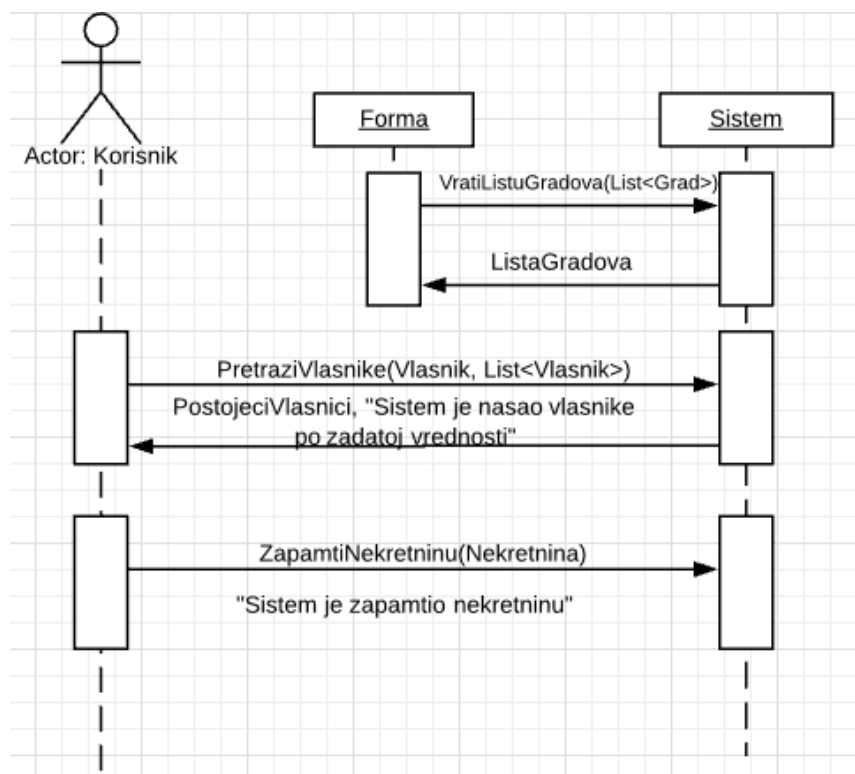
2. Analiza

2.1 Ponašanje softverskog sistema – sistemski dijagrami sekvenci

2.1.1. DS1: Dijagram sekvenci slučaja korišćenja – Unos nekretnine

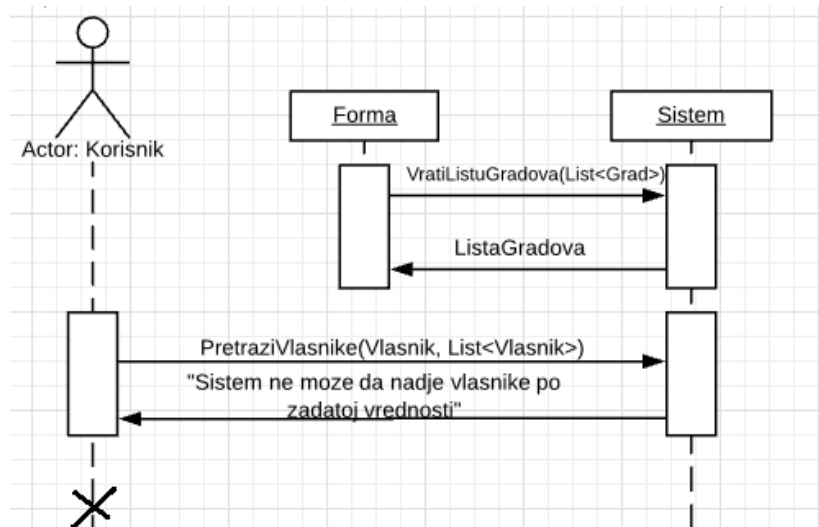
Osnovni scenario SK

1. **Korisnik** poziva **sistem** da nađe vlasnike po zadatoj vrednosti. (APSO)
2. **Sistem** prikazuje **korisniku** pronađene vlasnike i poruku: „**Sistem** je našao vlasnike po zadatoj vrednosti“. (IA)
3. **Korisnik** poziva **sistem** da zapamti podatke o novoj nekretnini. (APSO)
4. **Sistem** prikazuje **korisniku** poruku: „Sistem je zapamtio nekretninu“. (IA)

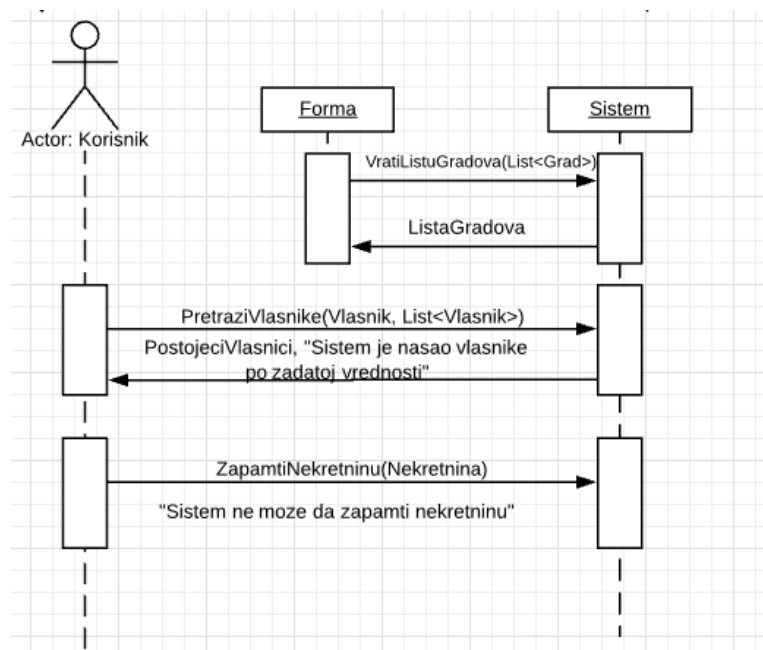


Alternativna scenarija

- 2.1 Ukoliko **sistem** ne može da nađe vlasnike on prikazuje **korisniku** poruku: „**Sistem** ne može da nađe vlasnike po zadatoj vrednosti“. (IA)



4.1 Ukoliko **sistem** ne može da zapamti podatke o nekretnini on prikazuje **korisniku** poruku: „**Sistem** ne može da zapamti nekretninu“. (IA)



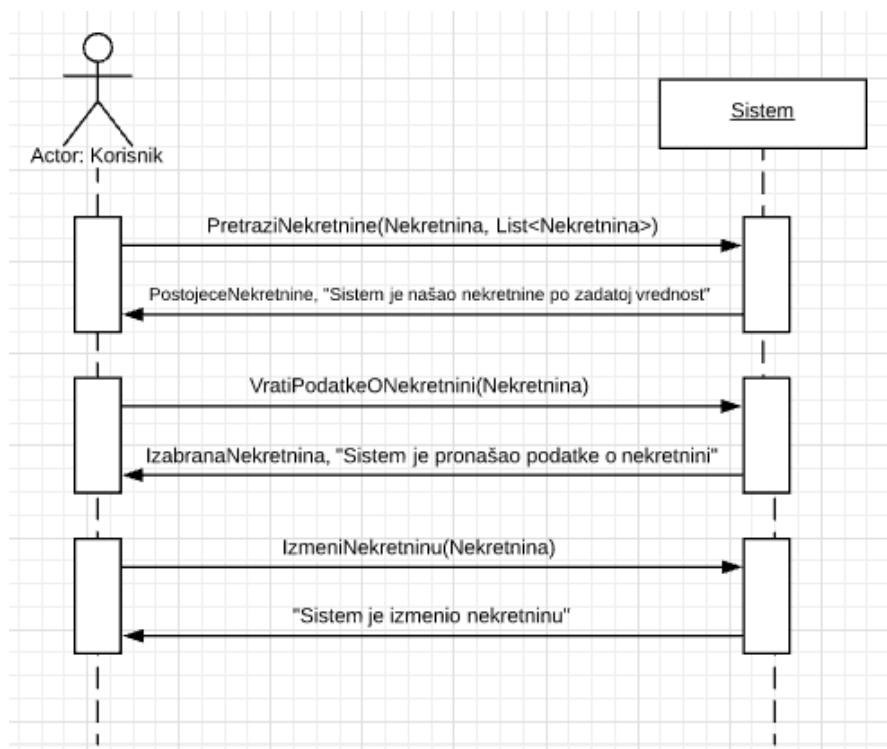
Iz dijagrama sekvenci uočavaju se sledeće sistemske operacije:

1. signal **VratiListuGradova**(List<Grad>)
2. signal **PretraziVlasnike**(Vlasnik, List<Vlasnik>);
3. signal **ZapamtiNekretninu**(Nekretnina);

2.1.2. DS2: Dijagram sekvenci slučaja korišćenja – Izmena nekretnine

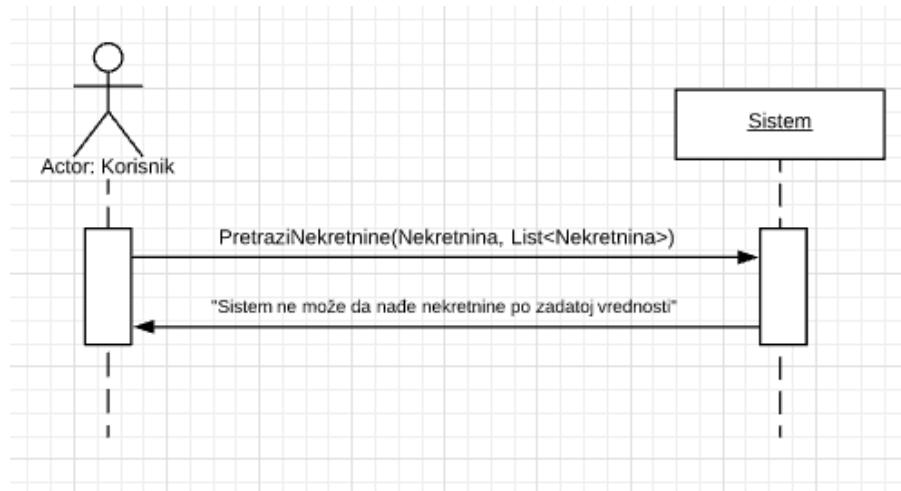
Osnovni scenario SK

1. **Korisnik** poziva **sistem** da nađe nekretnine po zadatoj vrednosti. (APSO)
2. **Sistem** prikazuje **korisniku** pronađene nekretnine i poruku: „**Sistem** je našao nekretnine po zadatoj vrednosti”. (IA)
3. **Korisnik** poziva **sistem** da vrati podatke o nekretnini. (APSO)
4. **Sistem** prikazuje **korisniku** podatke o nekretnini i poruku: „**Sistem** je pronašao podatke o izabranoj nekretnini”. (IA)
5. **Korisnik** poziva **sistem** da zapamti podatke o nekretnini. (APSO)
6. **Sistem** prikazuje **korisniku** poruku: „Sistem je izmenio nekretninu”. (IA)

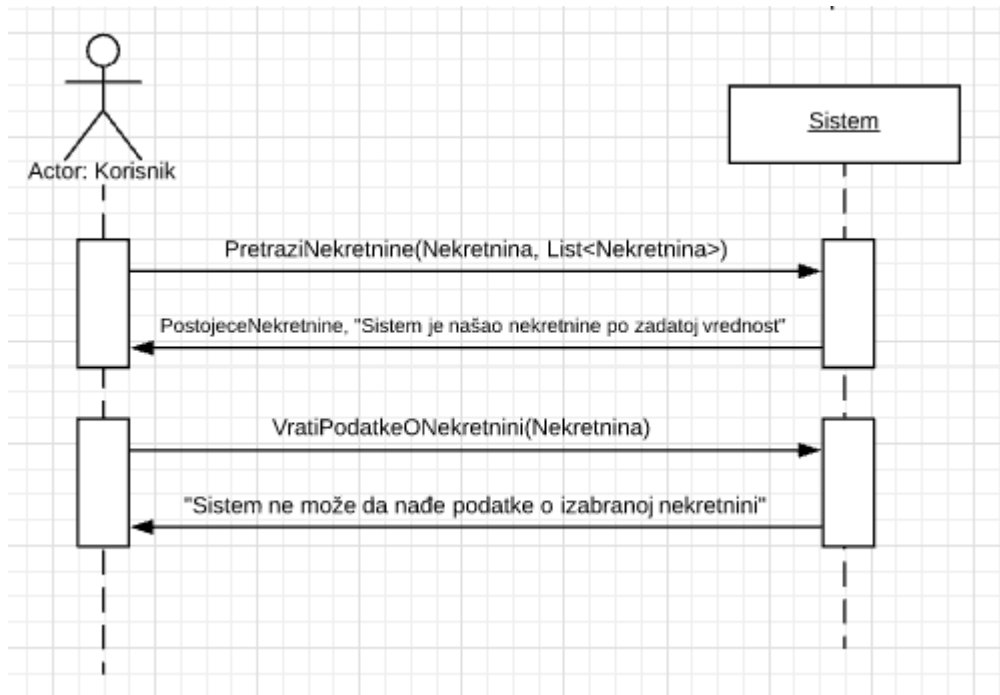


Alternativna scenarija

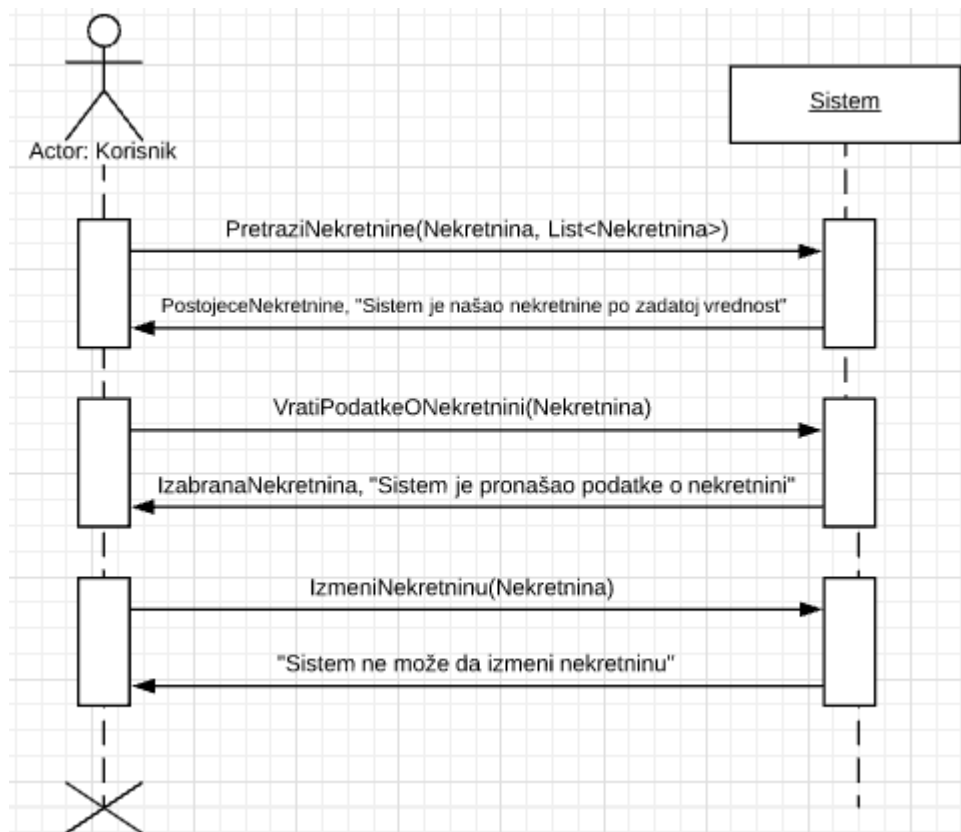
- 2.1 Ukoliko **sistem** ne može da nađe nekretnine on prikazuje **korisniku** poruku: „Sistem ne može da nađe nekretnine po zadatoj vrednosti”. Prekida se izvršenje scenarija. (IA)



- 4.1 Ukoliko **sistem** ne može da nađe podatke o nekretnini, obaveštava **korisnika** o tome: „**Sistem** ne može da nađe podatke o izabranoj nekretnini“. Prekida se izvršenje scenarija. (IA)



- 6.1 Ukoliko **sistem** ne može da zapamti izmenjene podatke o nekretnini on prikazuje **korisniku** poruku: „**Sistem** ne može da izmeni nekretninu“. (IA)



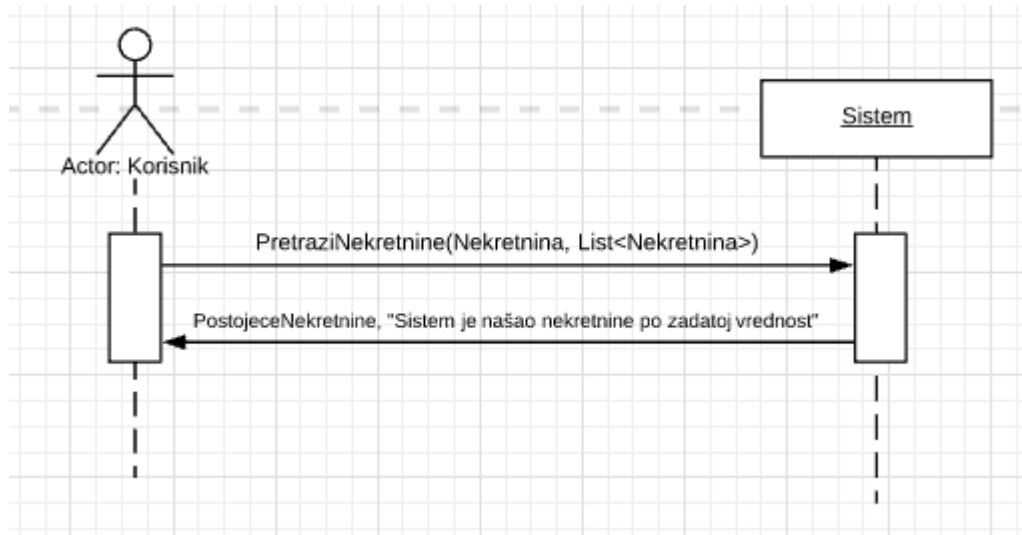
Iz dijagrama sekvenci uočavaju se sledeće sistemske operacije:

1. signal **PretraziNekretnine(Nekretnina, List<Nekretnina>)**;
2. signal **VратиPodatkeONekretnini(Nekretnina)**;
3. signal **IzmeniNekretninu(Nekretnina)**;

2.1.3. DS3: Dijagram sekvenci slučaja korišćenja – Pretraga nekretnina

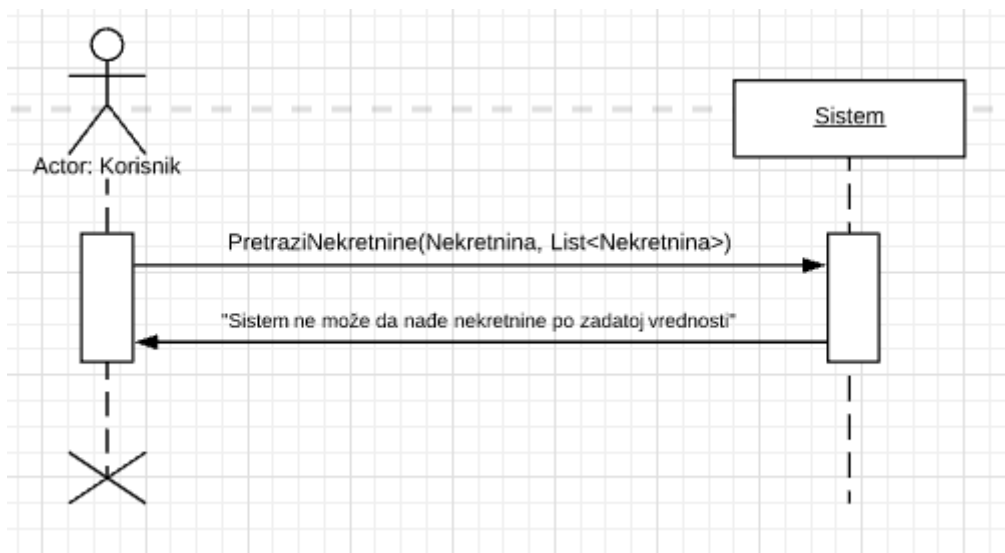
Osnovni scenario SK

1. **Korisnik** poziva **sistem** da nađe nekretnine po zadatoj vrednosti. (APSO)
2. **Sistem** prikazuje **korisniku** podatke o nekretninama i poruku: „**Sistem** je našao nekretnine po zadatoj vrednosti“. (IA)



Alternativna scenarija

- 2.1 Ukoliko **sistem** ne može da nađe nekretnine on prikazuje **korisniku** poruku: „**Sistem** ne može da nađe nekretnine po zadatoj vrednosti“. (IA)



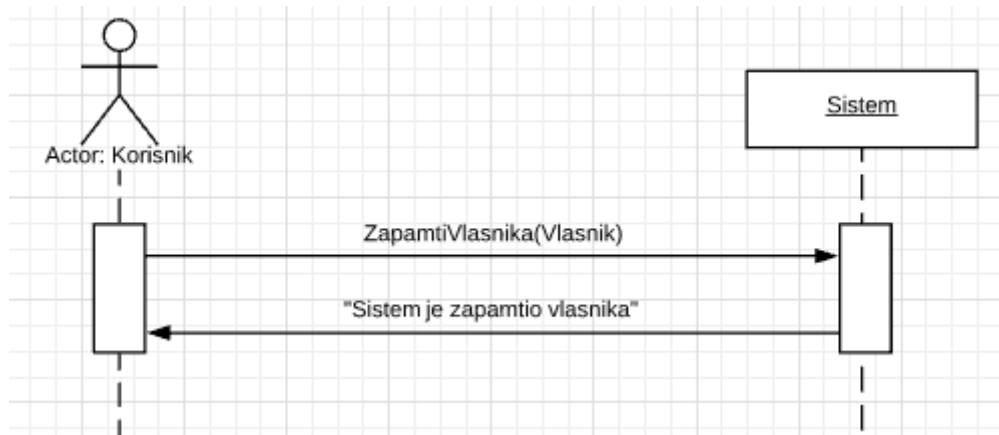
Iz dijagrama sekvenci uočavaju se sledeće systemske operacije:

1. signal **PretraziNekretnine**(Nekretnine, List<Nekretnina>);

2.1.4. DS4: Dijagram sekvenci slučaja korišćenja – Unos vlasnika

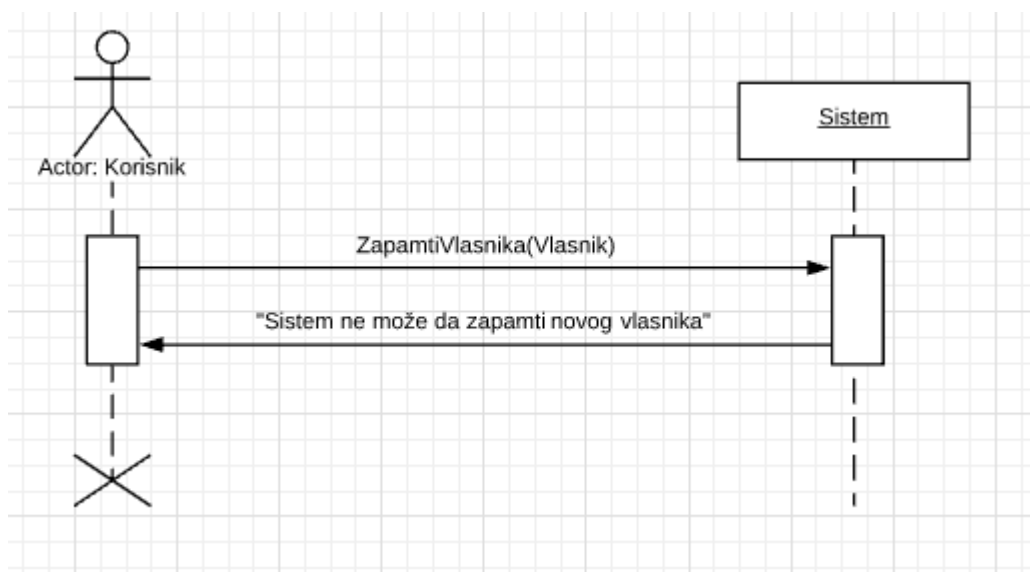
Osnovni scenario SK

1. **Korisnik** poziva **sistem** da zapamti podatke o vlasniku. (APSO)
2. **Sistem** prikazuje **korisniku** poruku: „Sistem je zapamtio vlasnika“. (IA)



Alternativna scenarija

- 2.1 Ukoliko **sistem** ne može da zapamti podatke o novom vlasniku on prikazuje **korisniku** poruku: „**Sistem** ne može da zapamti vlasnika“. (IA)



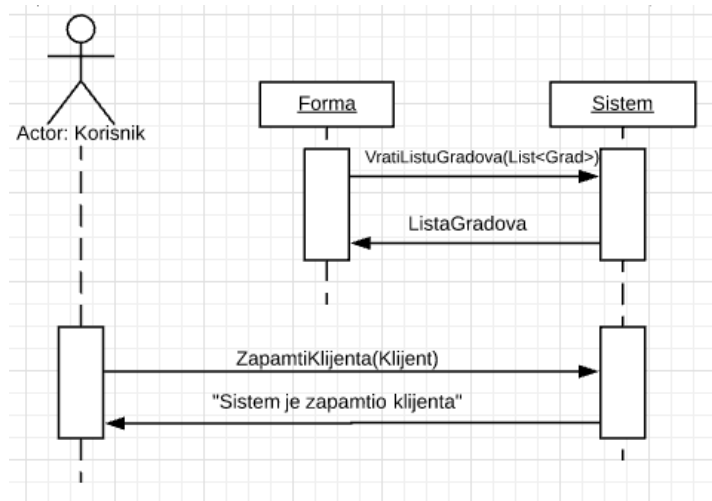
Iz dijagrama sekvenci uočavaju se sledeće sistemske operacije:

1. signal **ZapamtiVlasnika(Vlasnik)**;

2.1.5. DS5: Dijagram sekvenci slučaja korišćenja – Unos klijenta

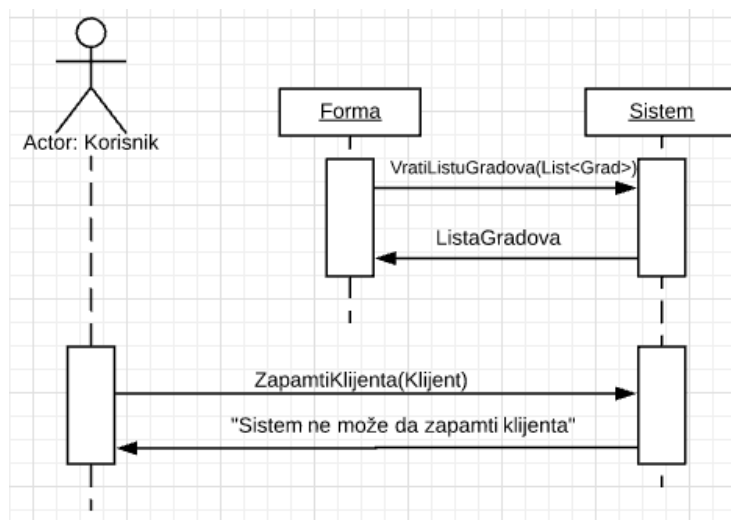
Osnovni scenario SK

1. **Forma** poziva **sistem** da vrati listu gradova. (APSO)
2. **Sistem** prikazuje **korisniku** listu gradova. (IA)
3. **Korisnik** poziva **sistem** da zapamti podatke o klijentu. (APSO)
4. **Sistem** prikazuje **korisniku** poruku: „Sistem je zapamtio klijenta“. (IA)



Alternativna scenarija

- 4.1 Ukoliko **sistem** ne može da zapamti podatke o novom klijentu on prikazuje **korisniku** poruku: „**Sistem** ne može da zapamti klijenta“. (IA)



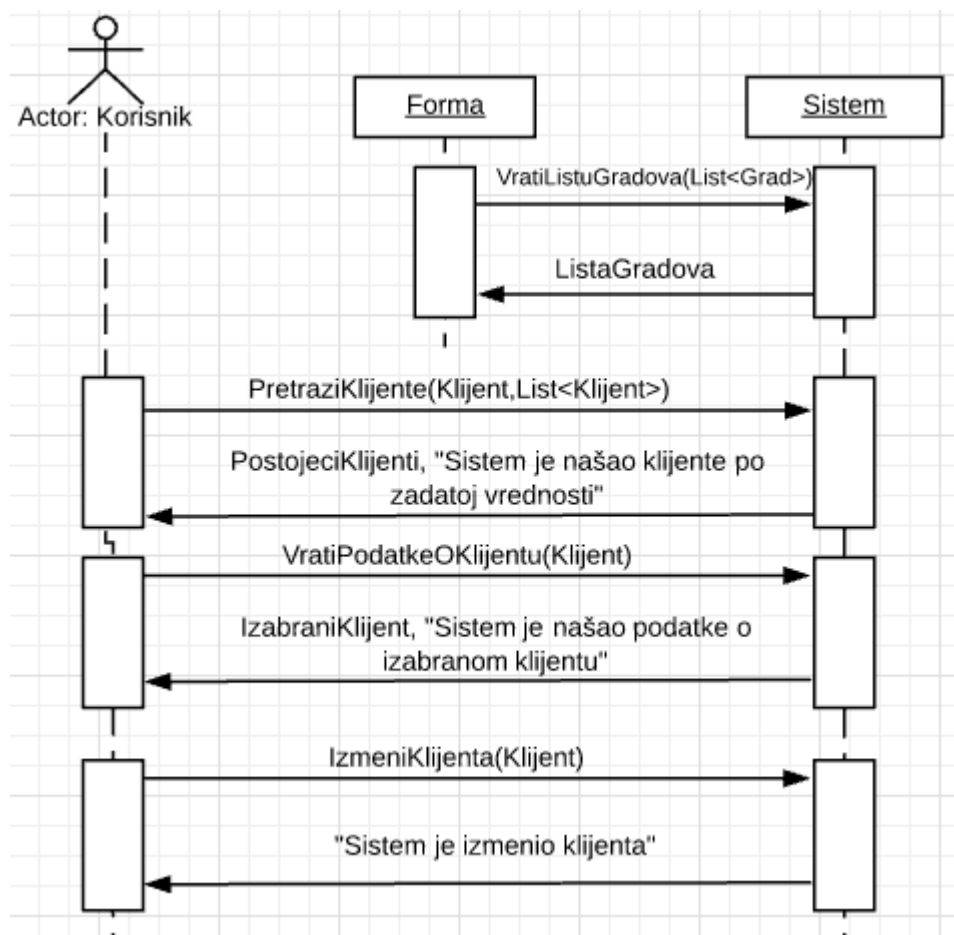
Iz dijagrama sekvenci uočavaju se sledeće sistemske operacije:

1. Signal **VratiListuGradova**(List<Grad>);
2. signal **ZapamtiKlijenta**(Klijent);

2.1.6. DS6: Dijagram sekvenci slučaja korišćenja – Izmena klijenta

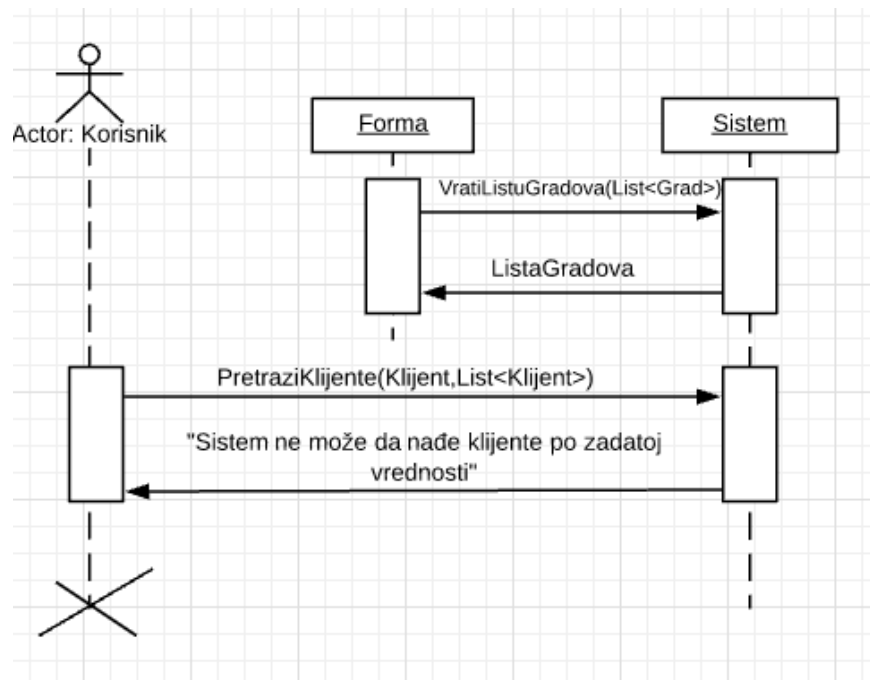
Osnovni scenario SK

1. **Forma** poziva **sistem** da vrati listu gradova. (APSO)
2. **Sistem** prikazuje **korisniku** listu gradova. (IA)
3. **Korisnik** poziva **sistem** da nađe klijente po zadatoj vrednosti. (APSO)
4. **Sistem** prikazuje **korisniku** pronađene klijente i poruku: „**Sistem** je našao klijente po zadatoj vrednosti”. (IA)
5. **Korisnik** poziva **sistem** da vrati podatke o klijentu. (APSO)
6. **Sistem** prikazuje **korisniku** podatke o klijentu i poruku: „**Sistem** je pronašao podatke o izabranom klijentu”. (IA)
7. **Korisnik** poziva **sistem** da izmeni podatke o klijentu. (APSO)
8. **Sistem** prikazuje **korisniku** poruku: „Sistem je izmenio klijenta”. (IA)

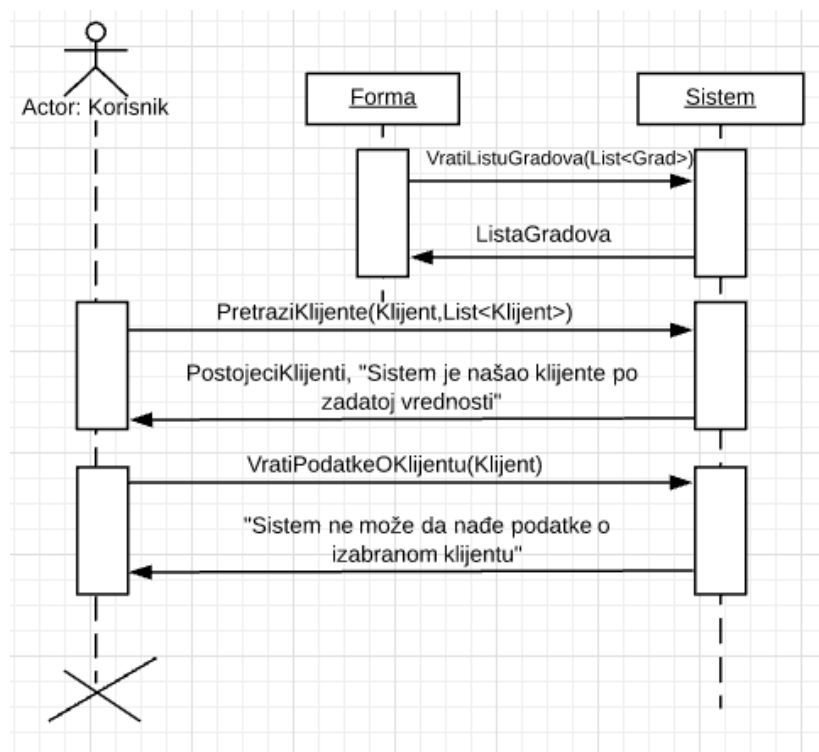


Alternativna scenarija

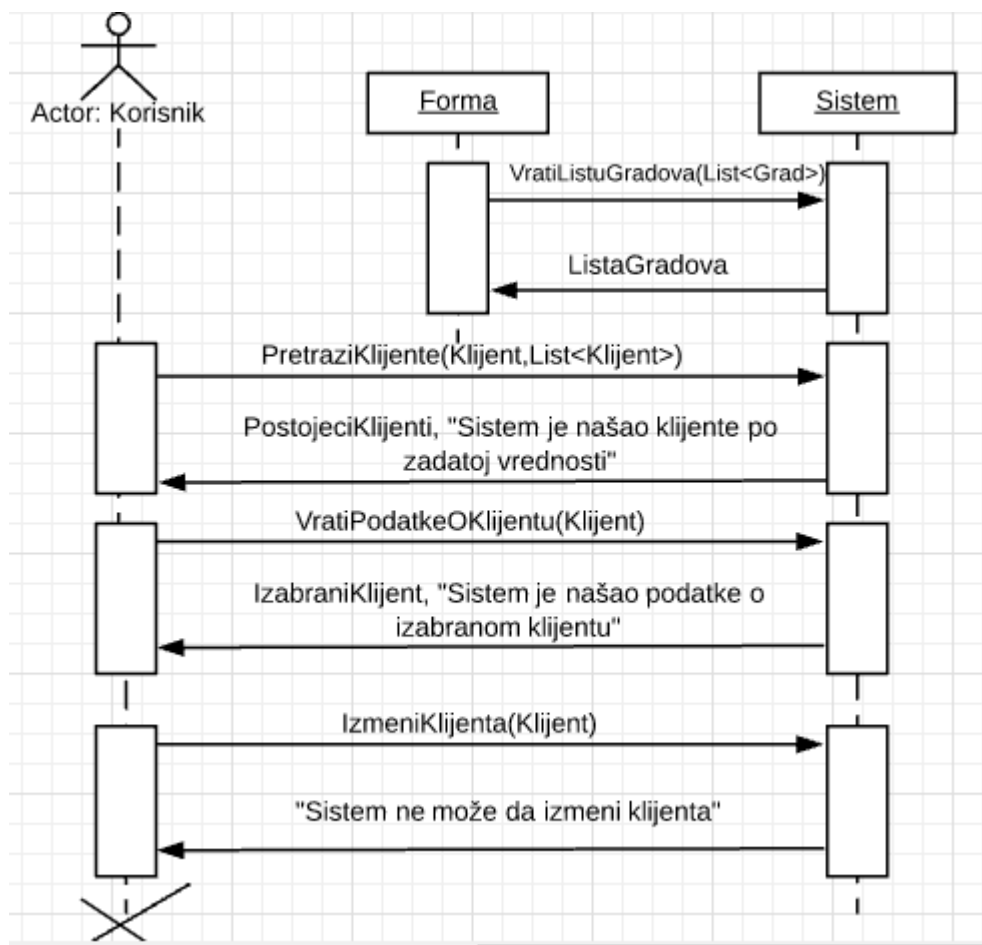
- 2.1 Ukoliko **sistem** ne može da nađe klijenta on prikazuje **korisniku** poruku: „Sistem ne može da nađe klijente po zadatoj vrednosti”. Prekida se izvršenje scenarija. (IA)



4.1 Ukoliko **sistem** ne može da nađe podatke o klijentu, obaveštava **korisnika** o tome: „**Sistem** ne može da nađe podatke o izabranom klijentu“. Prekida se izvršenje scenarija. (IA)



6.1 Ukoliko **sistem** ne može da zapamti izmenjene podatke o klijentu on prikazuje **korisniku** poruku: „**Sistem** ne može da izmeni klijenta“. (IA)



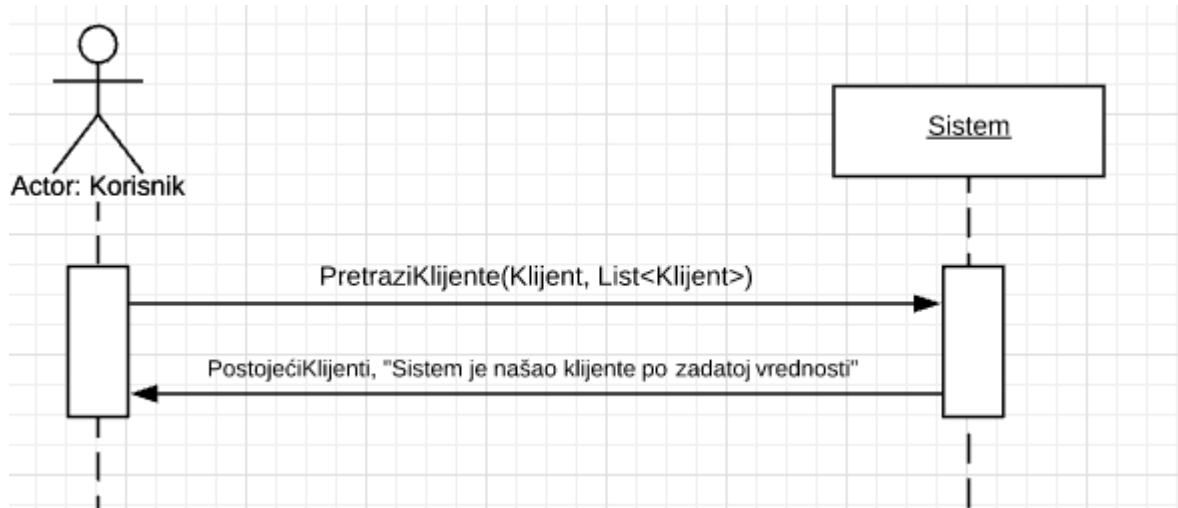
Iz dijagrama sekvenci uočavaju se sledeće sistemske operacije:

1. signal **VratiListuGradova**(List<Grad>);
2. signal **PretraziKlijente**(Klijent, List<Klijent>);
3. signal **VratiPodatkeOKlijentu**(Klijent);
4. signal **IzmeniKlijenta**(Klijent);

2.1.7. DS7: Dijagram sekvenci slučaja korišćenja – Pretraga klijenata

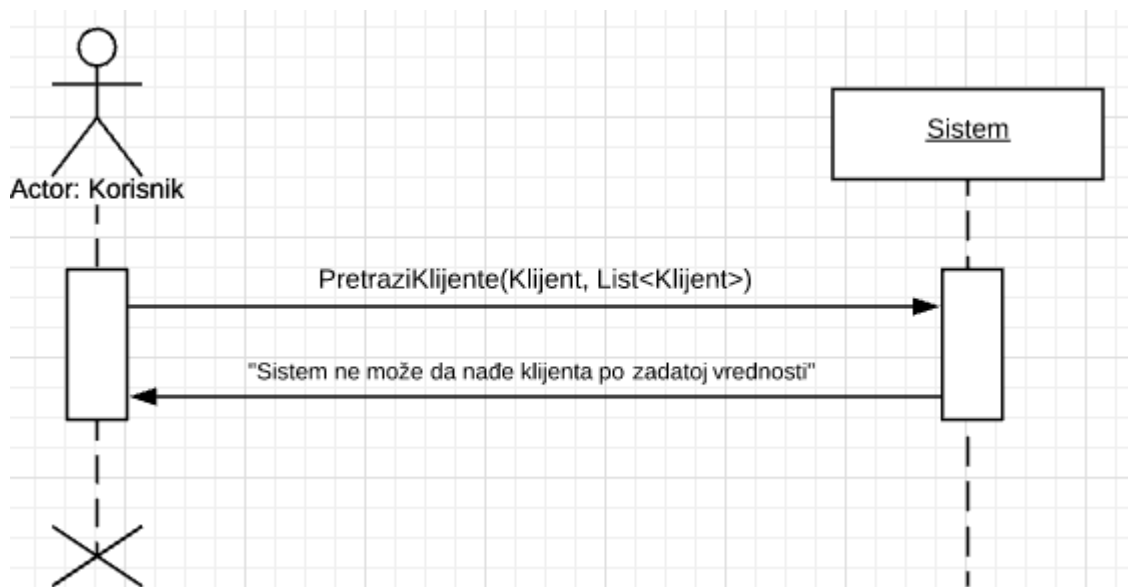
Osnovni scenario SK

1. **Korisnik** poziva **sistem** da nađe klijente po zadatoj vrednosti. (APSO)
2. **Sistem** prikazuje korisniku podatke o klijentima i poruku: „**Sistem** je našao klijente po zadatoj vrednosti“. (IA)



Alternativna scenarija

- 2.1 Ukoliko **sistem** ne može da nađe klijenta on prikazuje **korisniku** poruku: „**Sistem** ne može da nađe klijente po zadatoj vrednosti“. (IA)



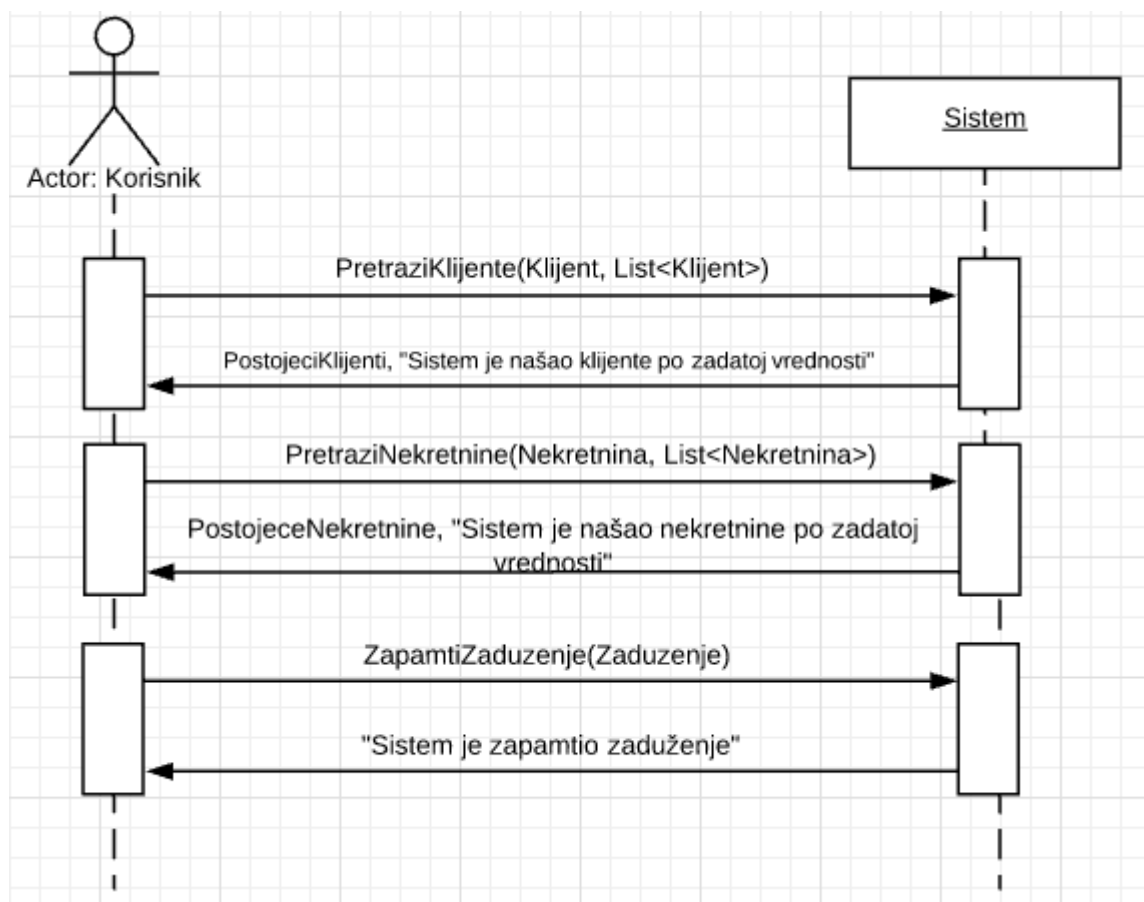
Iz dijagrama sekvenci uočavaju se sledeće systemske operacije:

1. signal **PretraziKlijente**(Klijent, List<Klijent>);

2.1.8. DS8: Dijagram sekvenci slučaja korišćenja – Unos zaduženja

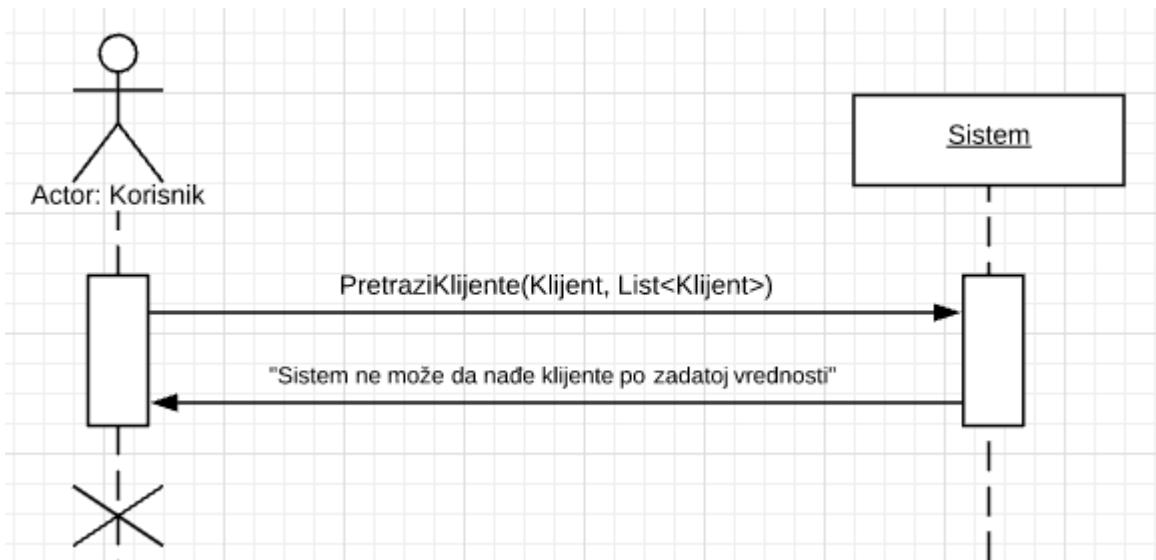
Osnovni scenario SK

1. **Korisnik** poziva **sistem** da nađe klijente po zadatoj vrednosti. (APSO)
2. **Sistem** prikazuje **korisniku** podatke o klijentima i poruku: „**Sistem** je našao klijente po zadatoj vrednosti“. (IA)
3. **Korisnik** poziva **sistem** da nađe nekretnine po zadatoj vrednosti. (APSO)
4. **Sistem** prikazuje **korisniku** podatke o nekretninama i poruku: „**Sistem** je našao nekretnine po zadatoj vrednosti“. (IA)
5. **Korisnik** poziva **sistem** da zapamti podatke o zaduženjima. (APSO)
6. **Sistem** prikazuje **korisniku** poruku: „**Sistem** je zapamtio zaduženje“. (IA)

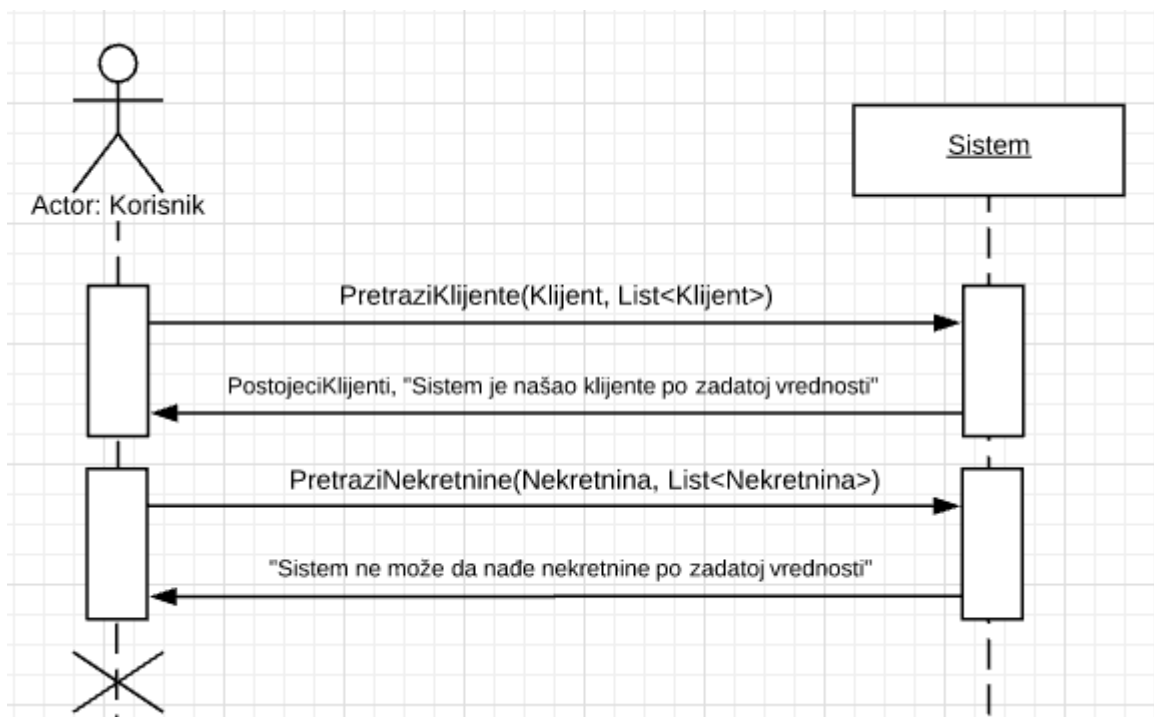


Alternativna scenarija

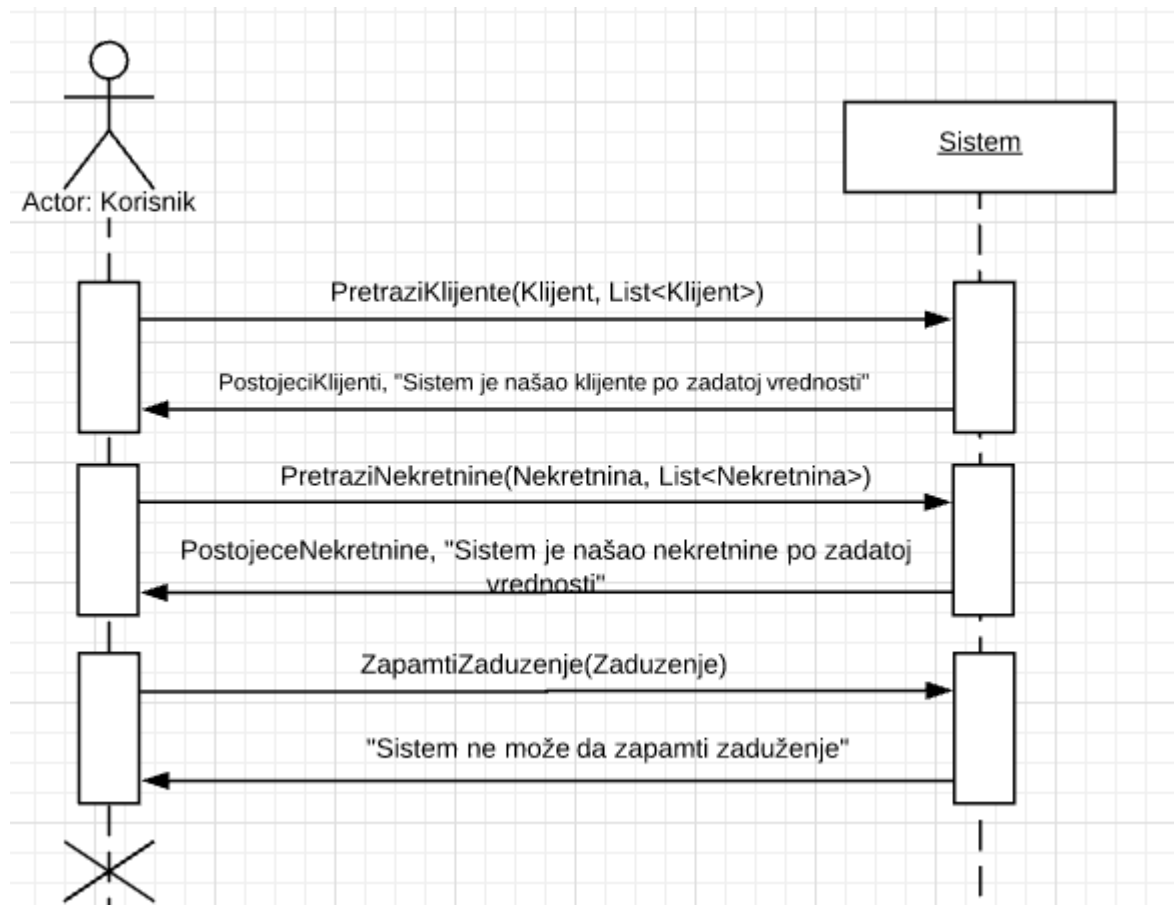
- 2.1 Ukoliko **sistem** ne može da nađe klijente on prikazuje **korisniku** poruku: „**Sistem** ne može da nađe klijente po zadatoj vrednosti“. (IA)



- 4.1. Ukoliko **sistem** ne može da nađe nekretnine on prikazuje **korisniku** poruku: „**Sistem** ne može da nađe nekretnine po zadatoj vrednosti“. (IA)



6.1 Ukoliko **sistem** ne može da zapamti podatke o zaduženjima, on prikazuje **korisniku** poruku „**Sistem** ne može da zapamti zaduženje“. (IA)



Iz dijagrama sekvenci uočavaju se sledeće sistemske operacije:

1. signal **PretraziKlijente**(Klijent, List<Klijent>);
2. signal **PretraziNekretnine** (Nekretnina, List<Nekretnina>);
3. signal **ZapamtiZaduzenje**(Zaduzenje);

2.1.9. DS9: Dijagram sekvenci slučaja korišćenja – Izmena zaduženja

Naziv SK

Izmena zaduženja

Aktori SK

Korisnik

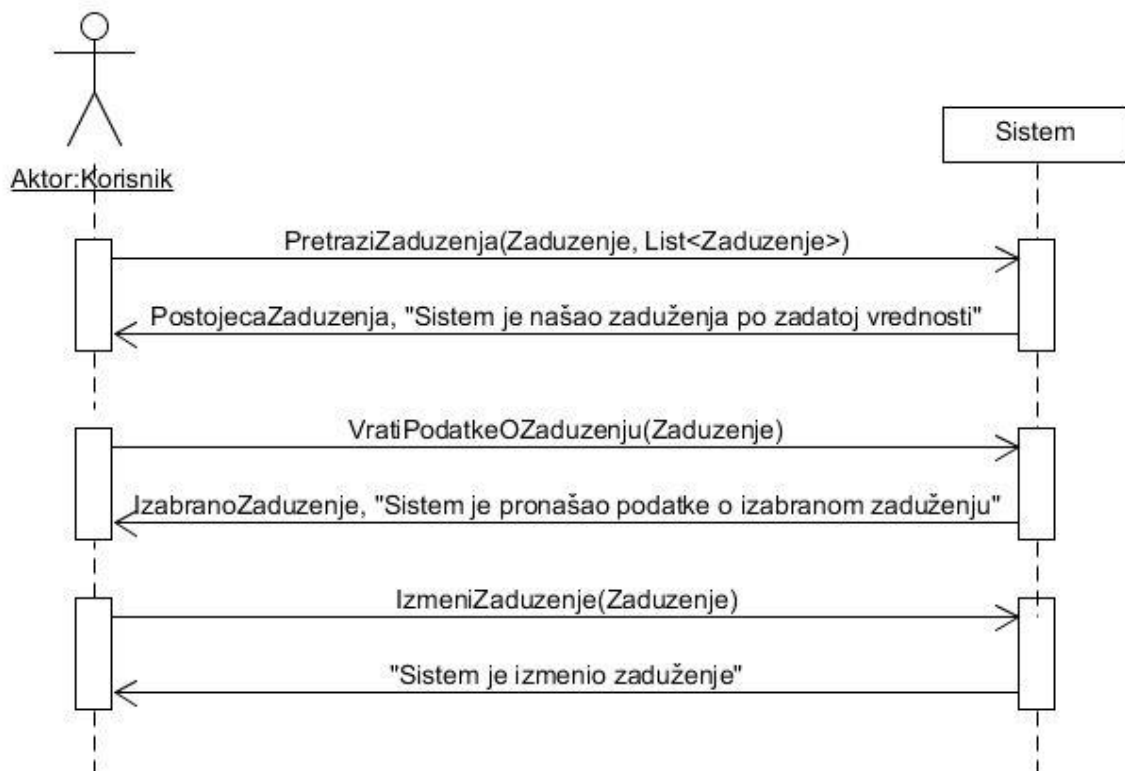
Učesnici SK

Korisnik i sistem (program)

Preduslov: **Sistem** je uključen i **korisnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa zaduženjem.

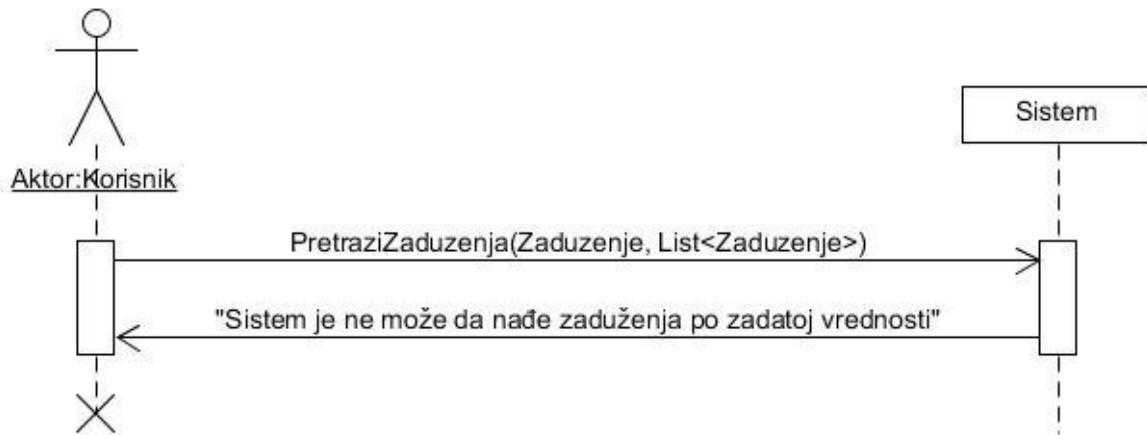
Osnovni scenario SK

1. **Korisnik poziva sistem** da nađe zaduženja po zadatoj vrednosti. (APSO)
2. **Sistem prikazuje korisniku** pronađena zaduženja i poruku: „**Sistem** je našao zaduženja po zadatoj vrednosti”. (IA)
3. **Korisnik poziva sistem** da vrati podatke o zaduženju. (APSO)
4. **Sistem prikazuje korisniku** podatke o zaduženju i poruku: „**Sistem** je pronašao podatke o izabranom zaduženju”. (IA)
5. **Korisnik poziva sistem** da zapamti podatke o zaduženju. (APSO)
6. **Sistem prikazuje korisniku** poruku: „Sistem je izmenio zaduženje”. (IA)

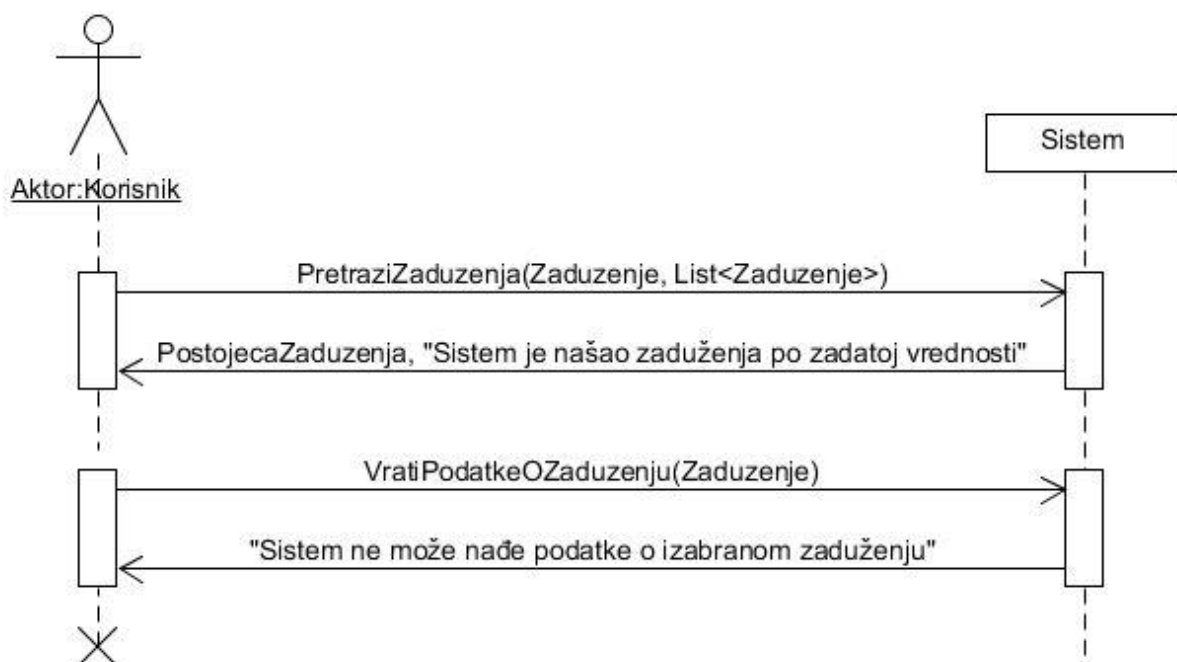


Alternativna scenarija

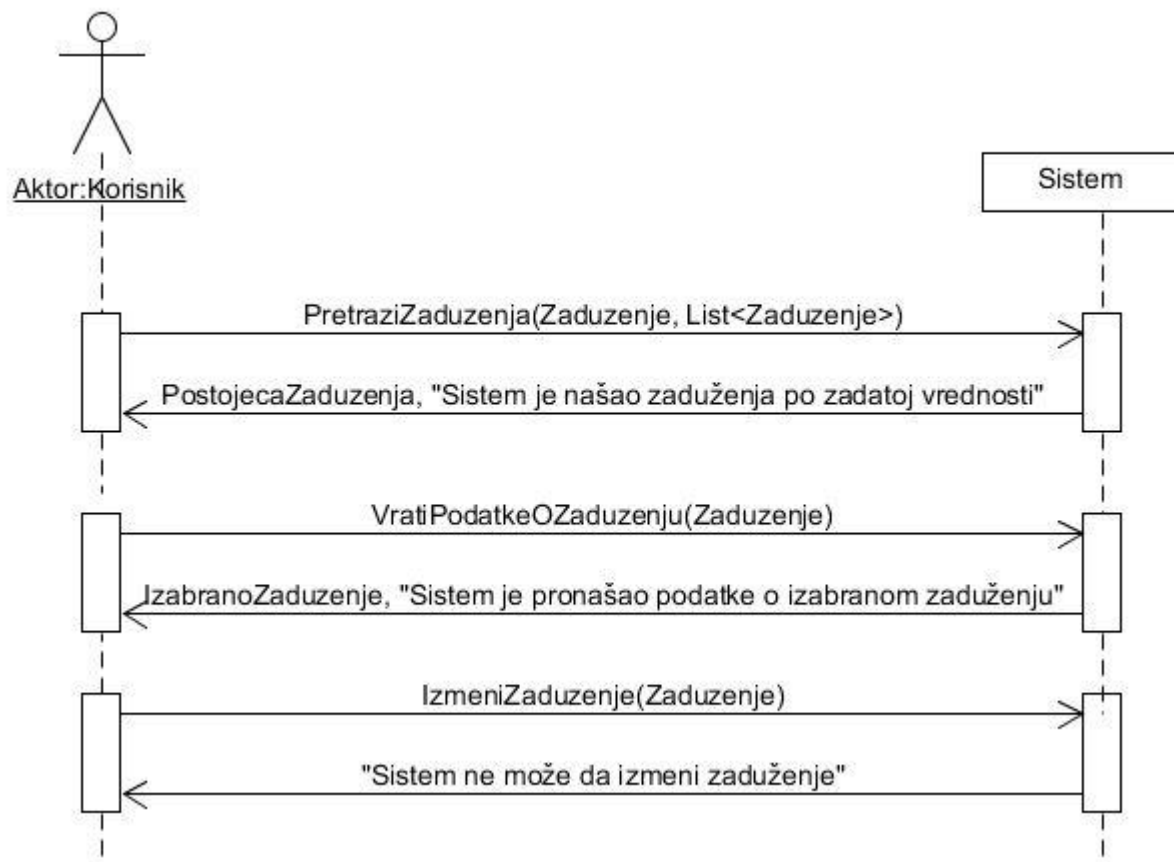
2.1 Ukoliko **sistem** ne može da nađe zaduženja on prikazuje **korisniku** poruku: „Sistem ne može da nađe zaduženja po zadatoj vrednosti“. Prekida se izvršenje scenarija. (IA)



4.1 Ukoliko **sistem** ne može da nađe podatke o zaduženju, obaveštava **korisnika** o tome: „**Sistem** ne može da nađe podatke o izabranom zaduženju“. Prekida se izvršenje scenarija. (IA)



6.1 Ukoliko **sistem** ne može da zapamti izmenjene podatke o zaduženju on prikazuje **korisniku** poruku: „**Sistem** ne može da izmeni zaduženje“. (IA)



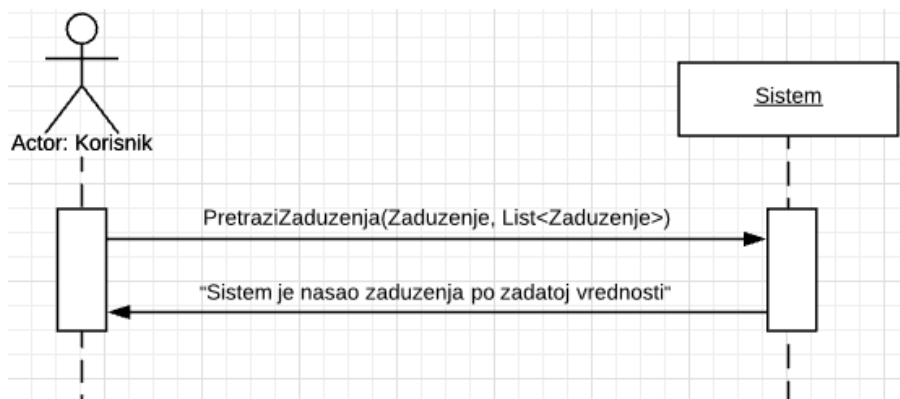
Iz dijagrama sekvenci uočavaju se sledeće sistemske operacije:

1. signal **PretraziZaduzenja**(Zaduzenje, List<Zaduzenje>);
2. signal **VратиPodatkeOZaduzenju** (Zaduzenje);
3. signal **IzmeniZaduzenje**(Zaduzenje);

2.1.10. DS10: Dijagram sekvenci slučaja korišćenja – Pretraga zaduženja

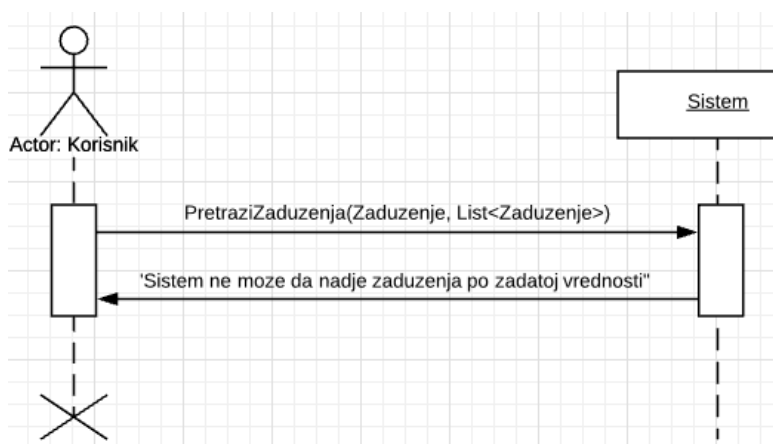
Osnovni scenario SK

1. **Korisnik** poziva **sistem** da nađe zaduženja po zadatoj vrednosti. (APSO)
2. **Sistem** prikazuje **korisniku** podatke o zaduženjima i poruku: „**Sistem** je našao zaduženja po zadatoj vrednosti“. (IA)



Alternativna scenarija

- 2.1 Ukoliko **sistem** ne može da nađe zaduženja on prikazuje **korisniku** poruku: „**Sistem** ne može da nađe zaduženja po zadatoj vrednosti“. (IA)



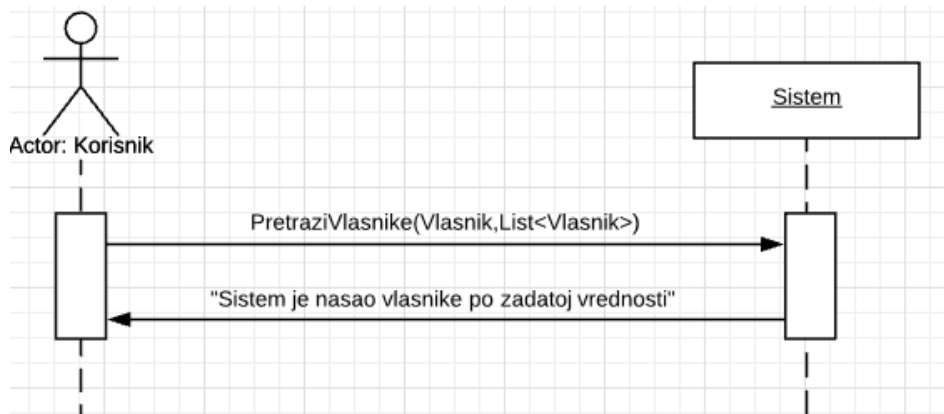
Iz dijagrama sekvenci uočavaju se sledeće systemske operacije:

1. signal **PretraziZaduzenja**(Zaduženje, List< Zaduženje >);

2.1.11. DS11: Dijagram sekvenci slučaja korišćenja – Pretraga vlasnika

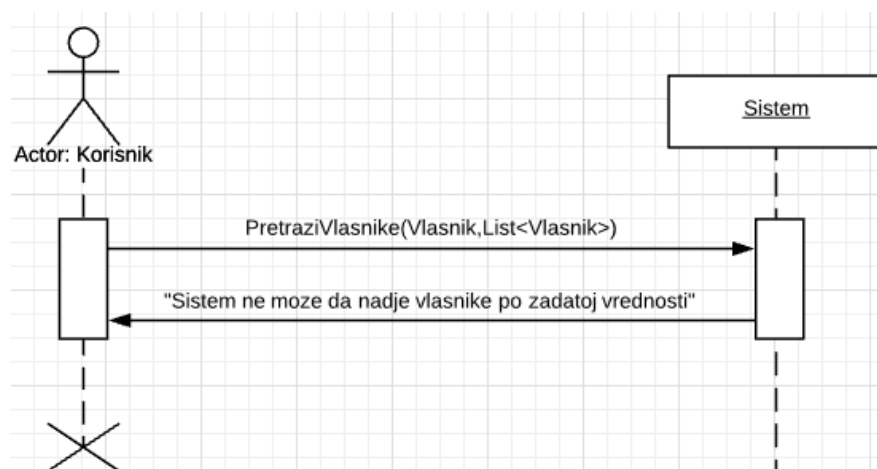
Osnovni scenario SK

1. **Korisnik** poziva **sistem** da nađe vlasnike po zadatoj vrednosti. (APSO)
2. **Sistem** prikazuje korisniku podatke o vlasnicima i poruku: „**Sistem** je našao vlasnike po zadatoj vrednosti“. (IA)



Alternativna scenarija

- 2.1 Ukoliko **sistem** ne može da nađe vlasnike on prikazuje **korisniku** poruku: „**Sistem** ne može da nađe vlasnike po zadatoj vrednosti“. (IA)



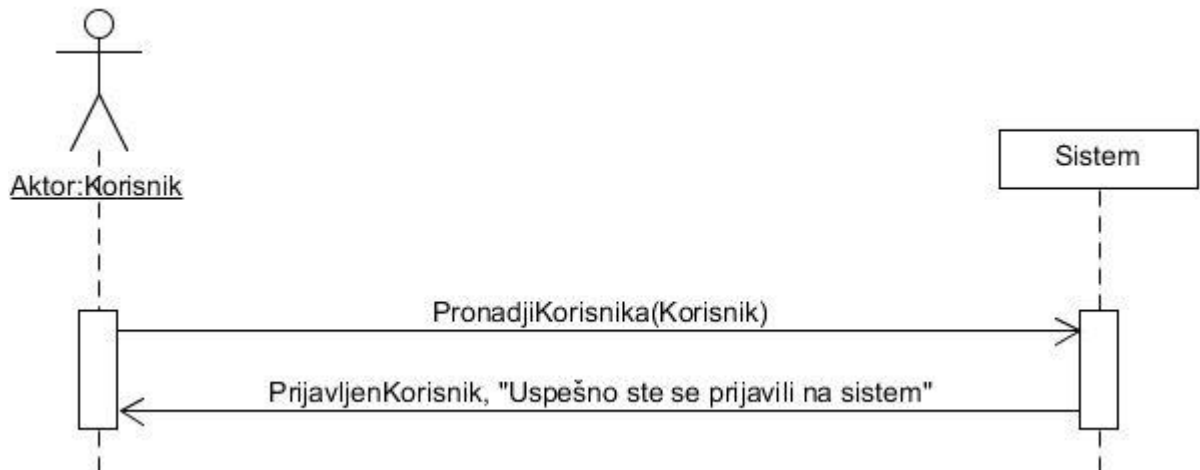
Iz dijagrama sekvenci uočavaju se sledeće systemske operacije:

1. signal **PretraziVlasnike**(Vlasnik, List<Vlasnik>);

2.1.12. DS12: Dijagram sekvenci slučaja korišćenja – Prijava korisnika

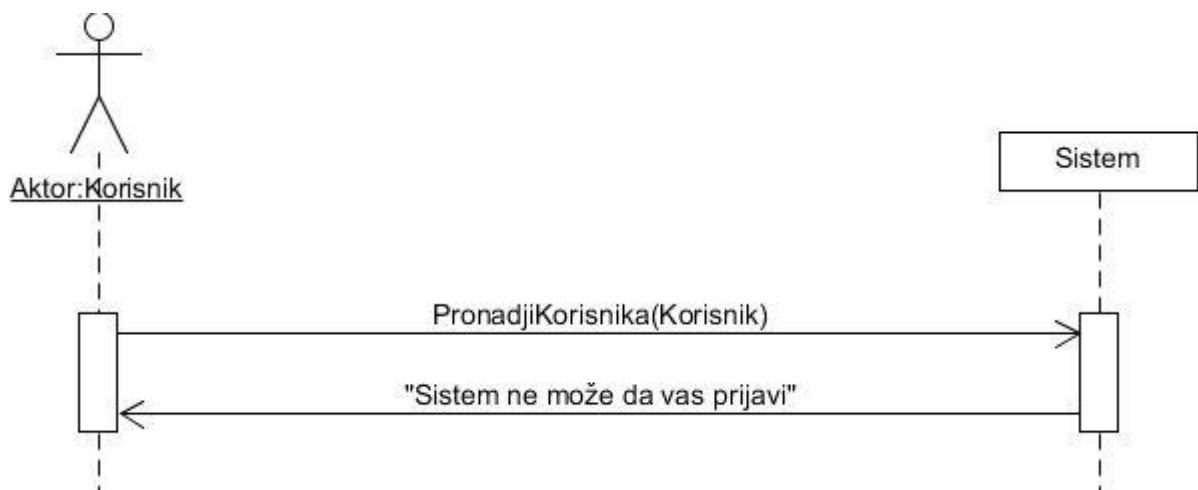
Osnovni scenario SK

1. **Korisnik** poziva sistem da pronade **korisnika** sa zadatim podacima. (APSO)
2. **Sistem** prikazuje poruku: "Uspešno ste se prijavili na sistem". (IA)



Alternativna scenarija

- 2.1 Ukoliko **sistem** ne može da nađe **korisnika** prikazuje poruku "**Sistem** ne može da vas prijavi". (IA)



Iz dijagrama sekvenci uočavaju se sledeće systemske operacije:

1. signal **PronadjiKorisnika(Korisnik)**;

2.2 Definisanje ugovora o sistemskim operacijama

2.2.1. Ugovor UG1: VратиListuGradova

Operacija: signal VратиListuGradova(List<Grad>):signal;

Veza sa SK: SK1, SK5, SK6,

Preduslovi: /

Postuslovi: /

2.2.2. Ugovor UG2: PretraziVlasnike

Operacija: signal PretraziVlasnike(Vlasnik, List<Vlasnik>): signal;

Veza sa SK: SK1, SK11

Preduslovi: /

Postuslovi: /

2.2.3. Ugovor UG3: ZapamtiNekretninu

Operacija: signal ZapamtiNekretninu(Nekretnina):signal;

Veza sa SK: SK1

Preduslovi: Vrednosna i strukturna ograničenja nad objektom Nekretnina moraju biti zadovoljena.

Postuslovi: Nekretnina je zapamćena.

2.2.4. Ugovor UG4: PretraziNekretnine

Operacija: signal PretraziNekretnine(Nekretnina, List<Nekretnina>):signal;

Veza sa SK: SK2, SK3, SK8

Preduslovi: /

Postuslovi: /

2.2.5. Ugovor UG5: VратиPodatkeONekretnini

Operacija: signal VратиPodatkeONekretnini(Nekretnina):signal;

Veza sa SK: SK2

Preduslovi: /

Postuslovi: /

2.2.6. Ugovor UG6: IzmeniNekretninu

Operacija: signal IzmeniNekretninu(Nekretnina):signal;

Veza sa SK: SK2

Preduslovi: Vrednosna i strukturna ograničenja na objektom Nekretnina moraju biti zadovoljena.

Postuslovi: Nekretnina je izmenjena.

2.2.7. Ugovor UG7: ZapamtiVlasnika

Operacija: signal ZapamtiVlasnika(Vlasnik):signal;

Veza sa SK: SK4

Preduslovi: Vrednosna i strukturna ograničenja na objektom Vlasnik moraju biti zadovoljena.

Postuslovi: Vlasnik je zapamćen.

2.2.8. Ugovor UG8: ZapamtiKlijenta

Operacija: signal ZapamtiKlijenta(Klijent):signal;

Veza sa SK: SK5

Preduslovi: Vrednosna i strukturna ograničenja na objektom Klijent moraju biti zadovoljena.

Postuslovi: Podaci o klijentu su zapamćeni.

2.2.9. Ugovor UG9: PretraziKlijente

Operacija: signal PretraziKlijente (Klijent, List<Klijent>):signal;

Veza sa SK: SK6, SK7, SK8

Preduslovi: /

Postuslovi: /

2.2.10. Ugovor UG10: VратиPodatkeOKlijentu(Klijent)

Operacija: signal VратиPodatkeOKlijentu(Klijent):signal;

Veza sa SK: SK6

Preduslovi: /

Postuslovi: /

2.2.11. Ugovor UG11: IzmeniKlijenta

Operacija: signal IzmeniKlijenta(Klijent):signal;

Veza sa SK: SK6

Preduslovi: Vrednosna i strukturna ograničenja na objektom Klijent moraju biti zadovoljena.

Postuslovi: Klijent je izmenjen.

2.2.12. Ugovor UG12: ZapamtiZaduzenje

Operacija: signal ZapamtiZaduzenje(Zaduzenje):signal;

Veza sa SK: SK8

Preduslovi: Vrednosna i strukturna ograničenja nad objektom Zaduzenje moraju biti zadovoljena.

Postuslovi: Podaci o zaduženju su zapaćeni.

2.2.13. Ugovor UG13: PretraziZaduzenja

Operacija: signal PretraziZaduzenja(Zaduzenje, List<Zaduzenje>):signal;

Veza sa SK: SK9,SK10

Preduslovi: /

Postuslovi: /

2.2.14. Ugovor UG14: VratiPodatkeOZaduzenju

Operacija: signal VratiPodatkeOZaduzenju(Zaduzenje):signal;

Veza sa SK: SK9

Preduslovi: /

Postuslovi: /

2.2.15. Ugovor UG15: IzmeniZaduzenje

Operacija: signal IzmeniZaduzenje(Zaduzenje):signal;

Veza sa SK: SK9

Preduslovi: Vrednosna i strukturna ograničenja na objektom Zaduzenje moraju biti zadovoljena.

Postuslovi: Zaduzenje je izmenjeno.

2.2.16. Ugovor UG18: PronadjiKorisnika

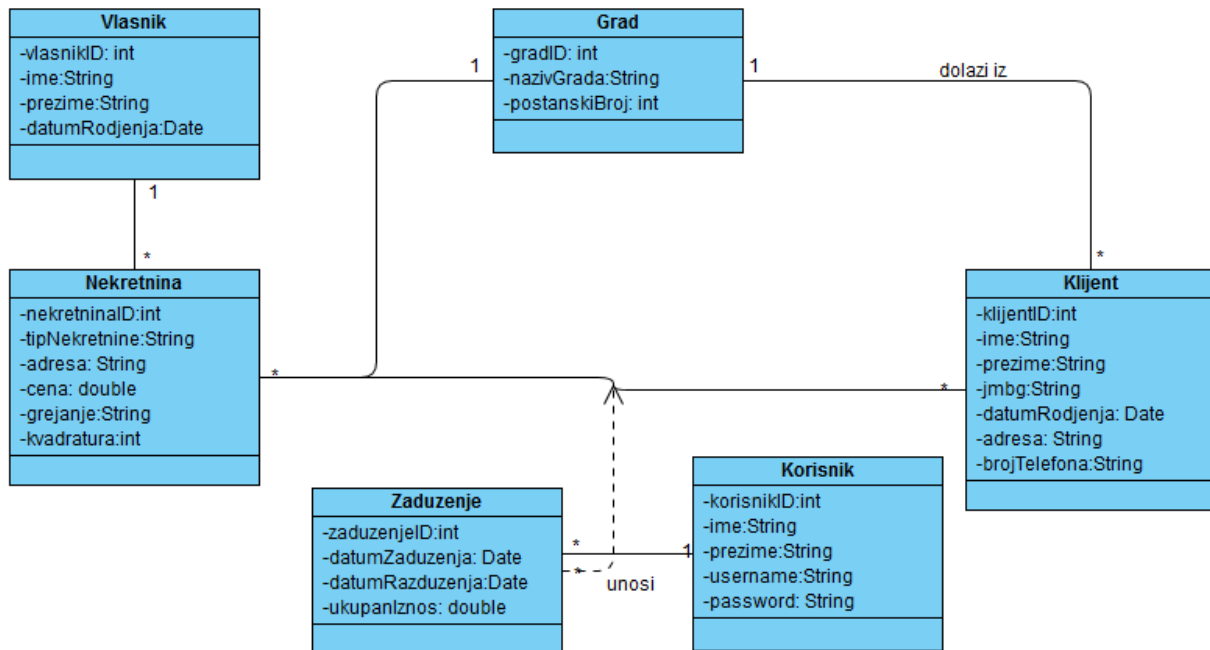
Operacija: signal PronadjiKorisnika(Korisnik):signal;

Veza sa SK: SK12

Preduslovi: /

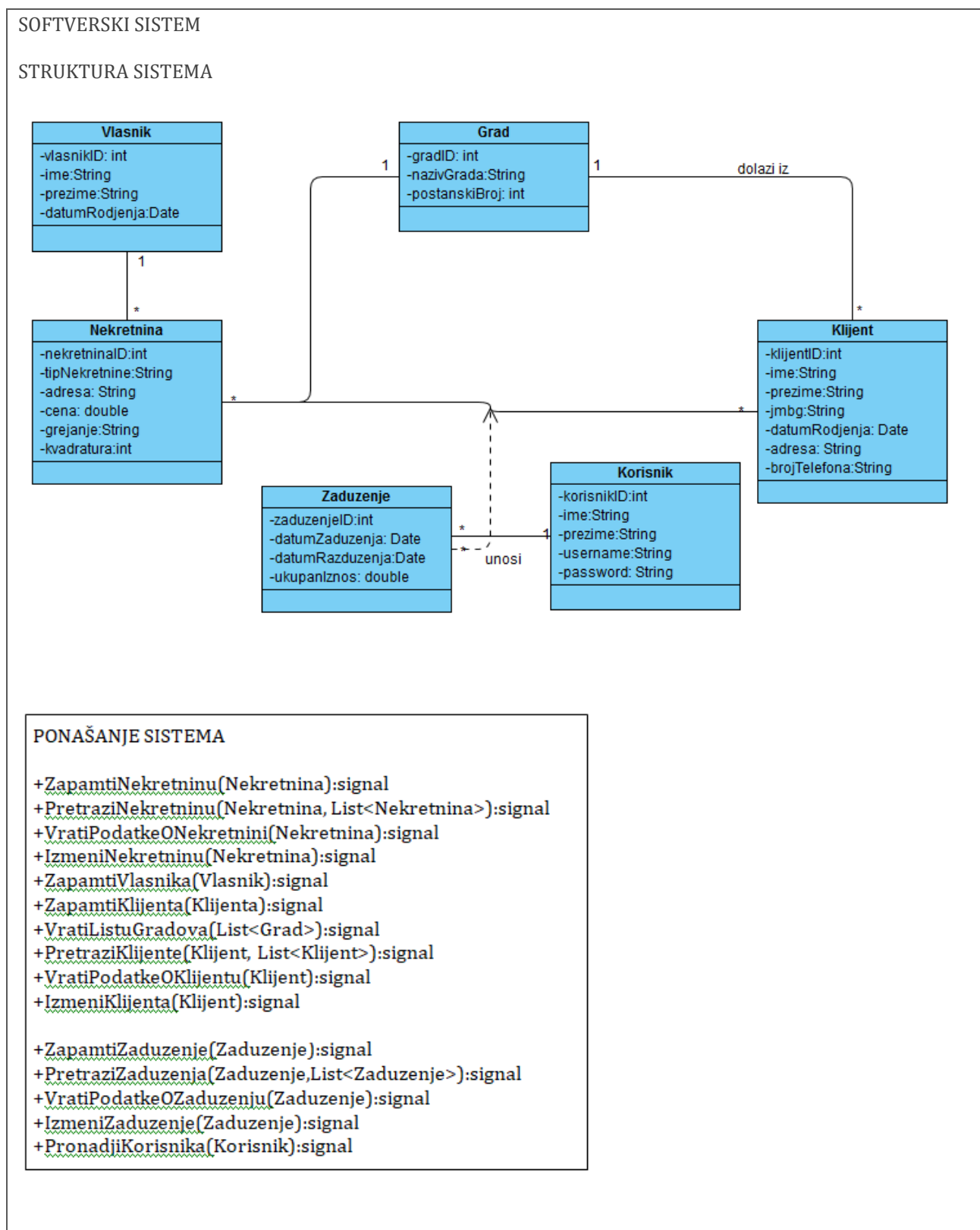
Postuslovi: /

2.3 Struktura softverskog sistema – Konceptualni (domenski) model



Slika 2. Konceptualni dijagram klasa

Kao rezultat analize scenarija SK i pravljenja konceptualnog modela dobija se logička struktura i ponašanje softverskog sistema:



Slika 3. Logička struktura i ponašanje softverskog sistema

2.4 Struktura softverskog sistema – Relacioni model

Vlasnik(vlasnikID, ime, prezime, datumRodjenja)

Nekretnina(nekretnaID, tipNekretnine, adresa, kvadratura, grejanje, cena, *gradID*, *vlasnikID*)

Grad(gradID, nazivGrada, postanskiBroj)

Klijent(klijentID, ime, prezime, JMBG, datumRodjenja, adresa, brojTelefona, *gradID*)

Zaduzenje(zaduzenjeID, datumZaduzenja, datumRazduzenja, ukupanIznos, *nekretnaID*, *klijentID*, *korisnikID*)

Korisnik(korisnikID, ime, prezime, username, password)

Tabela Nekretnina		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	
	nekretnaID	int	not null			INSERT RESTRICTED Vlasnik
	tipNekretnine	String				UPDATE RESTRICTED Vlasnik CASCADES Zaduzenje
	adresa	String				DELETE RESTRICTED Zaduzenje
	kvadratura	int				
	grejanje	String				
	cena	double				
	gradID	int	not null			
	vlasnikID	int	not null			

Tabela Vlasnik		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	INSERT / UPDATE CASCADES Nekretnina DELETE RESTRICTED Nekretnina
	vlasnikID	int	not null			
	ime	String				
	prezime	String				
	datumRodjenja	Date				

Tabela Grad		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	INSERT / UPDATE CASCADES Klijent DELETE RESTRICTED Klijent
	gradID	int	not null			
	postanskiBroj	int				
	nazivGrada	String				

Tabela Klijent		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	INSERT RESTRICTED Grad UPDATE RESTRICTED Grad CASCADES Zaduzenje DELETE RESTRICTED Zaduzenje
	klijentID	int	not null			
	ime	String				
	prezime	String				
	JMBG	String				
	datumRodjenja	Date				
	adresa	String				
	brojTelefona	String				
	gradID	int	not null			

Tabela Zaduzenje		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atri buti	Ime	Tip atribut a	Vrednost atributa	Međuzavisn ost atributa jedne tabele	Međuzavis nost atributa više tabela	INSERT RESTRICTED Nekretnina, Klijent UPDATE RESTRICTED Nekretnina, Klijent DELETE/
	zaduzenjeID	int	not null			
	datumZaduzenja	Date				
	datumRazduzenj a	Date				
	ukupanIznos	double				
	nekrentinaID	int	not null			
	klijentID	int	not null			
	korisnikID	int	not null			

Tabela Korisnik		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atribut a	Vrednost atributa	Međuzavisn ost atributa jedne tabele	Međuzavisn ost atributa više tabela	INSERT / UPDATE CASCADES Zaduzenje DELETE RESTRICTED Zaduzenje
	korisnikID	int	not null			
	ime	String				
	prezime	String				
	username	String				
	password	String				
	ulogovan	tinyint				

3. Projektovanje

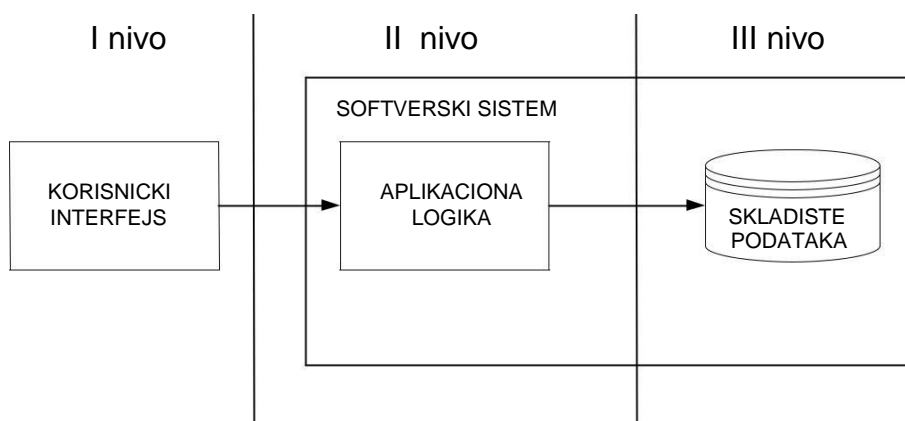
3.1 Arhitektura softverskog sistema

Faza projektovanja opisuje fizičku strukturu i ponašanje softverskog sistema (arhitekturu softverskog sistema).

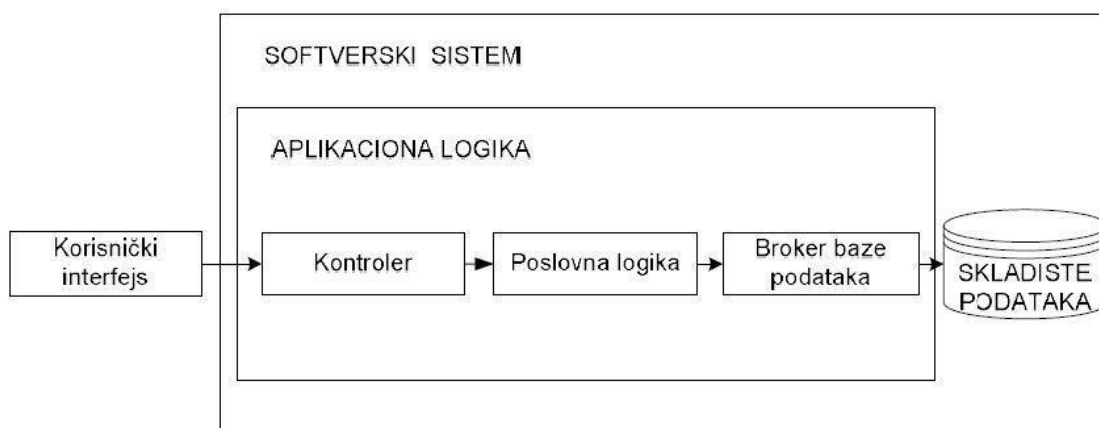
Arhitektura sistema je tronivojska i sastoji se od sledećih nivoa:

- korisnički interfejs
- aplikaciona logika
- skladište podataka

Nivo korisničkog interfejsa je na strani klijenta, dok su aplikaciona logika i skladište podataka na strani servera.



Slika 4. Tronivojska arhitektura

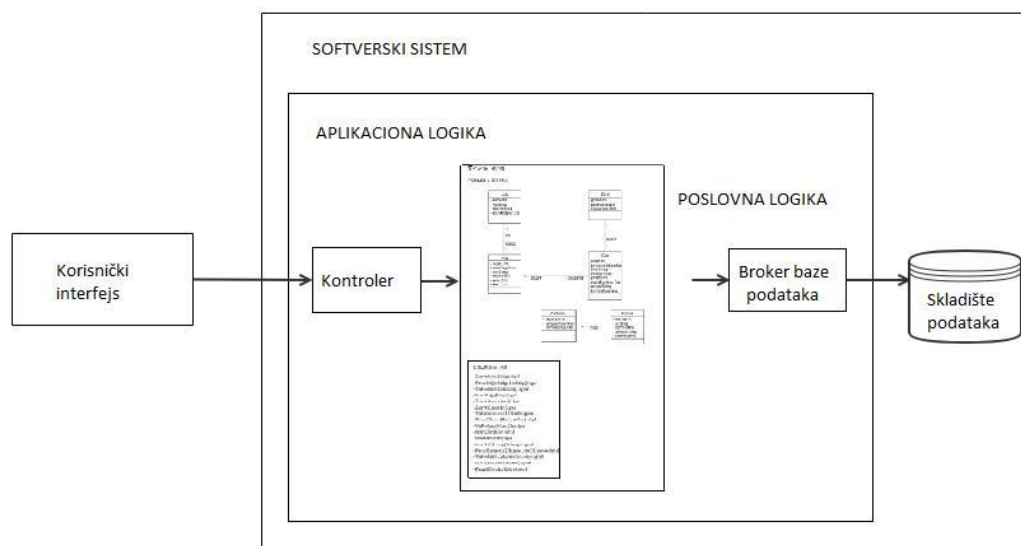


Slika 5. Softverski sistem

Da bi arhitektura softverskog sistema bila potpuna neophodno je isprojektovati:

1. Kontrolera
2. Poslovnu logiku - Domenske klase
3. Poslovnu logiku - Sistemske operacije
4. Brokera baze podataka
5. Skladište podataka
6. Korisnički interfejs

U fazi prikupljanja i analize već je data specifikacija strukture i ponašanja softverskog sistema, odnosno specifikacija poslovne logike softverskog sistema.



Slika 6. Arhitektura softverskog sistema

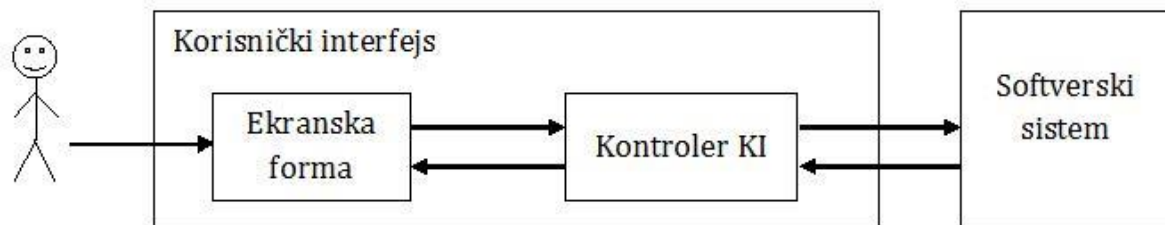
3.2 Projektovanje korisničkog interfejsa

Korisnički interfejs predstavlja realizaciju ulaza i/ili izlaza softverskog sistema. Njega čine ekranska forma i kontroler korisničkog interfejsa.

Korisnički interfejs definisan je preko skupa ekranskih formi. Scenarija korišćenja ekranskih formi su direktno povezani sa scenarijima slučajeva korišćenja.

Uloga ekranske forme je da:

- prihvati podatke koje korisnik unosi
- prihvata događaje koje pravi korisnik
- poziva kontroler korisničkog interfejsa kako bi mu prosledio podatke
- prikazuje podatke dobijene od kontrolera korisničkog interfejsa.

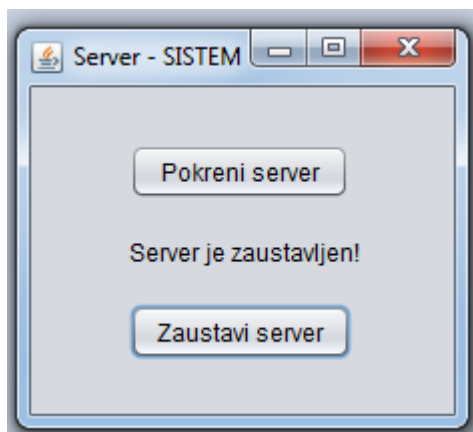


Slika 7. Arhitektura korisničkog interfejsa

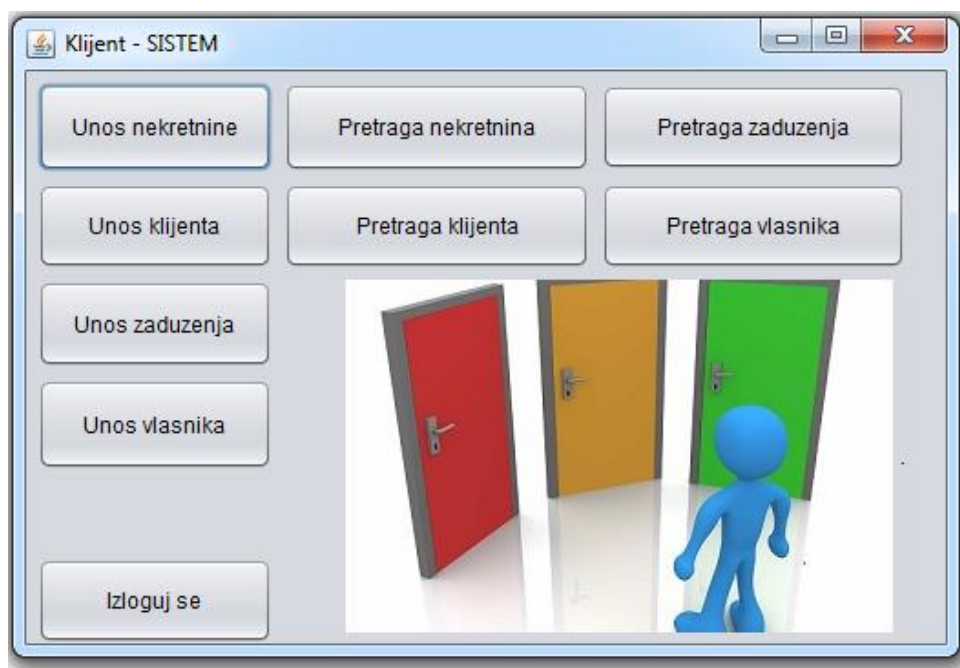
3.2.1. Projektovanje ekranskih formi

Korisnički interfejs definisan je preko skupa ekranskih formi. Scenarija korišćenja ekranskih formi su direktno povezani sa scenarijima slučajeva korišćenja.

Glavna ekranska forma serverskog dela aplikacija izgleda ovako:



Glavna ekranska forma klijentskog dela aplikacija izgleda ovako:



SK1: Slučaj korišćenja – Unos nekretnine

Naziv SK

Unos nekretnine

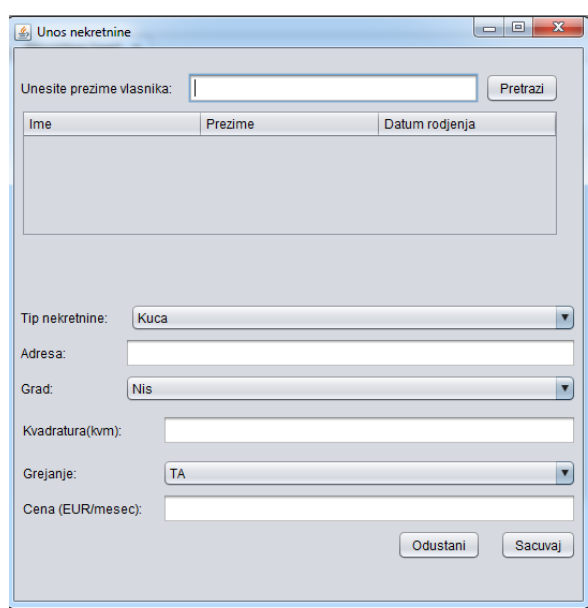
Aktori SK

Korisnik

Učesnici SK

Korisnik i sistem (program)

Preduslov: **Sistem** je uključen i **korisnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa nekretninama. Učitana je lista gradova.

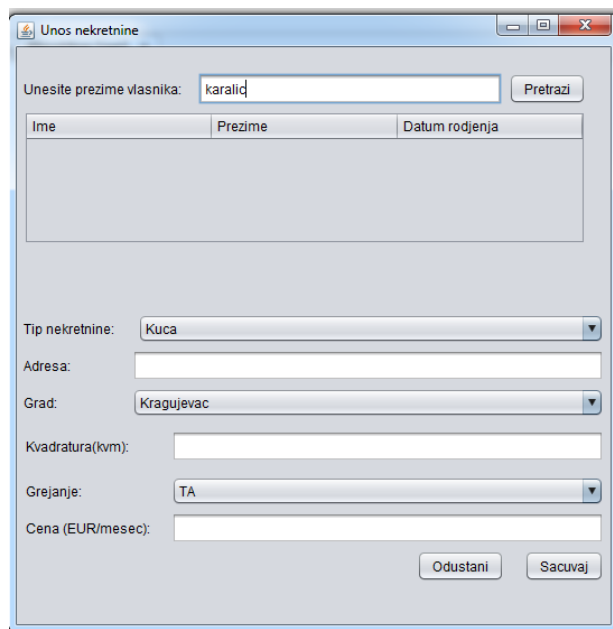


The screenshot shows a web application window titled "Unos nekretnine". It contains a search bar labeled "Unesite prezime vlasnika:" with a "Pretrazi" button. Below this is a table with columns "Ime", "Prezime", and "Datum rođenja". Further down are several input fields and dropdown menus: "Tip nekretnine:" (set to "Kuca"), "Adresa:", "Grad:" (set to "Nis"), "Kvadratura(kvm):", "Grejanje:" (set to "TA"), and "Cena (EUR/mesec):". At the bottom right are "Odustani" and "Sacuaj" buttons.

Osnovni scenario SK

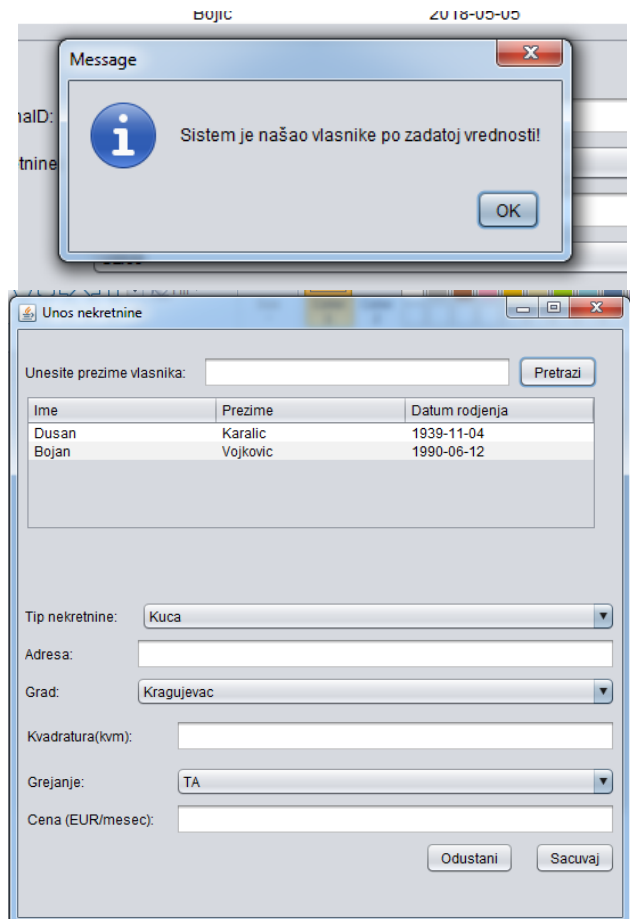
1. **Korisnik unosi** vrednost po kojoj pretražuje vlasnike. (APUSO)

Opis akcije: Korisnik unosi kriterijum pretrage (Prezime vlasnika) u polje za pretragu vlasnika. Ukoliko želi prikaz liste svih vlasnika korisnik ne unosi kriterijum ili unosi "*".

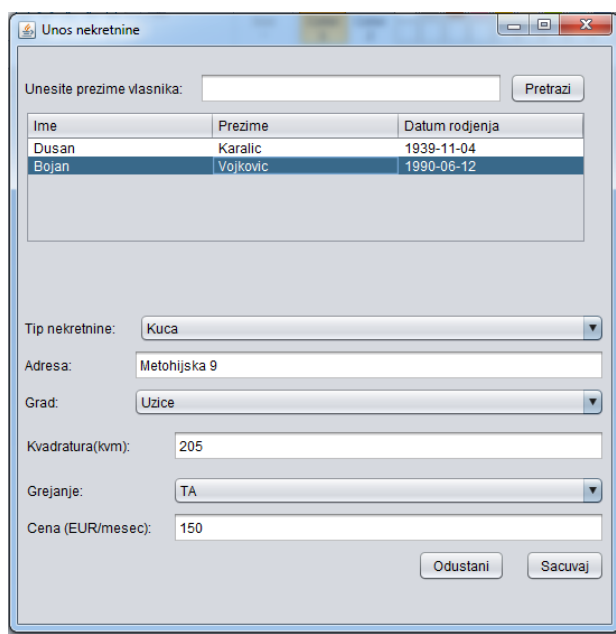


This screenshot is identical to the previous one, but the search bar "Unesite prezime vlasnika:" now contains the text "karalic". The rest of the form, including the table, dropdowns, and buttons, remains the same.

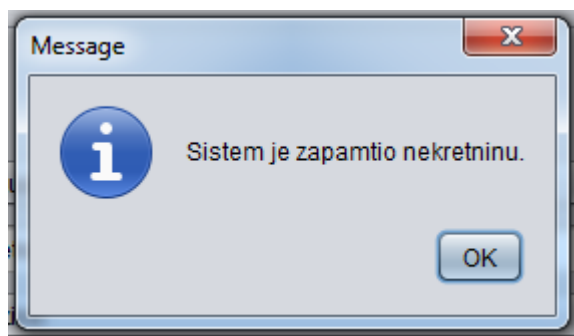
2. **Korisnik poziva sistem** da nađe vlasnike po zadatoj vrednosti. (APSO)
Korisnik klikom na dugme "Pretraži" poziva sistemsku operaciju PretraziVlasnike(Vlasnik, List< Vlasnik >).
3. **Sistem traži** vlasnike po zadatoj vrednosti. (SO)
4. **Sistem prikazuje korisniku** pronađene vlasnike i poruku: „**Sistem** je našao vlasnike po zadatoj vrednosti”. (IA)



5. **Korisnik bira** vlasnika i unosi podatke o nekretnini. (APUSO)
Opis akcije: Korisnik selektuje vlasnika iz tabele i unosi vrednosti u polja "Adresa", "Kvadratura", "Cena", i bira "Tip nekrenine", "Grejanje", "Grad".

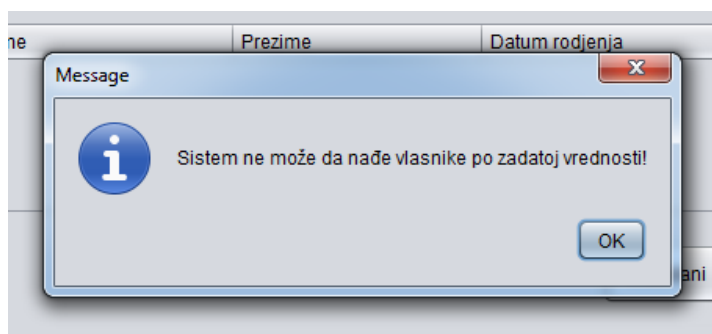


6. **Korisnik** kontrolira da li je korektno uneo podatke o nekretnini. (ANSO)
7. **Korisnik** poziva **sistem** da zapamti podatke o novoj nekretnini. (APSO)
Opis akcije: Korisnik klikom na dugme "Sačuvaj" poziva sistemsku operaciju ZapamtiNekretninu(Nekretnina) koja pamti novu nekretninu.
8. **Sistem** pamti podatke o nekretnini. (SO)
9. **Sistem** prikazuje **korisniku** poruku: „Sistem je zapamtio nekretninu“. (IA)

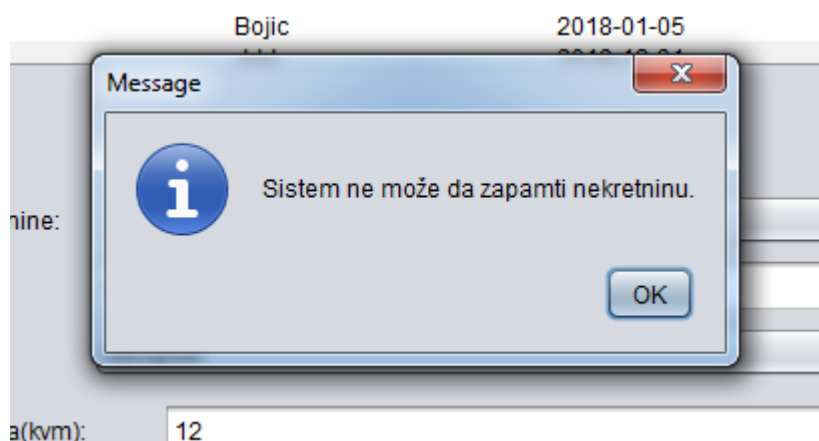


Alternativna scenarija

- 4.1 Ukoliko **sistem** ne može da nađe vlasnike on prikazuje **korisniku** poruku: „**Sistem** ne može da nađe vlasnike po zadatoj vrednosti“. Prekida se izvršenje scenarija. (IA)



- 9.1 Ukoliko **sistem** ne može da zapamti podatke o novoj nekretnini on prikazuje **korisniku** poruku: „**Sistem** ne može da zapamti nekretninu“. (IA)



SK2: Slučaj korišćenja – Izmena nekretnine

Naziv SK

Izmena nekretnine

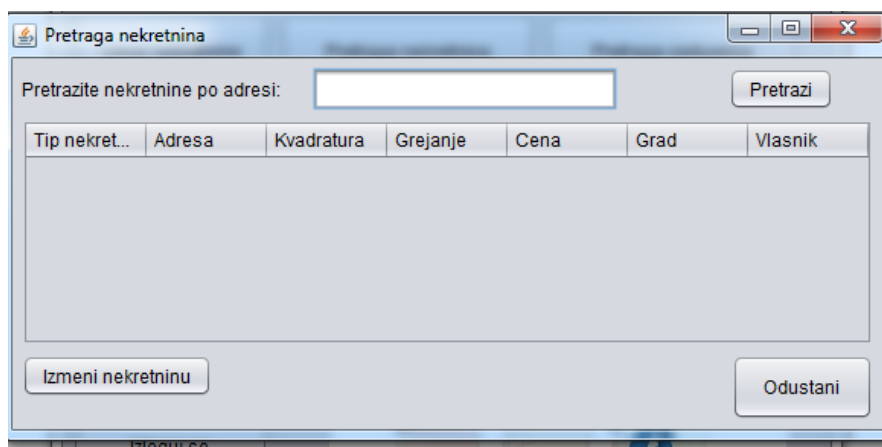
Aktori SK

Korisnik

Učesnici SK

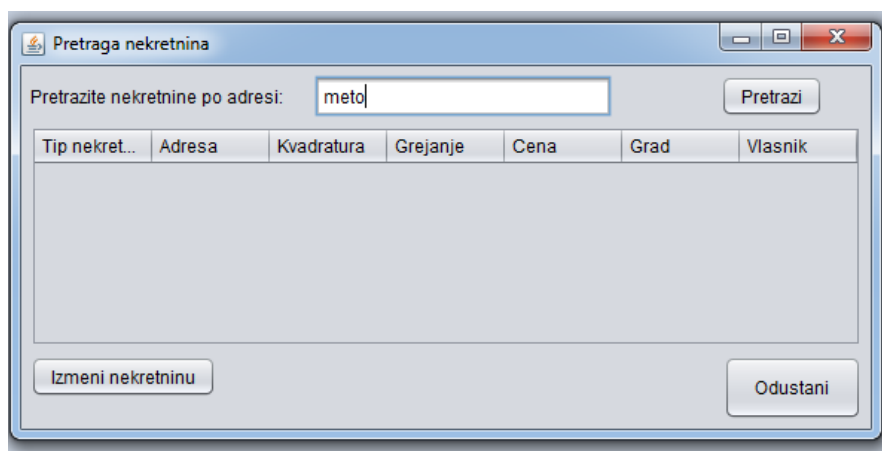
Korisnik i sistem (program)

Preduslov: **Sistem** je uključen i **korisnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa nekretninama.

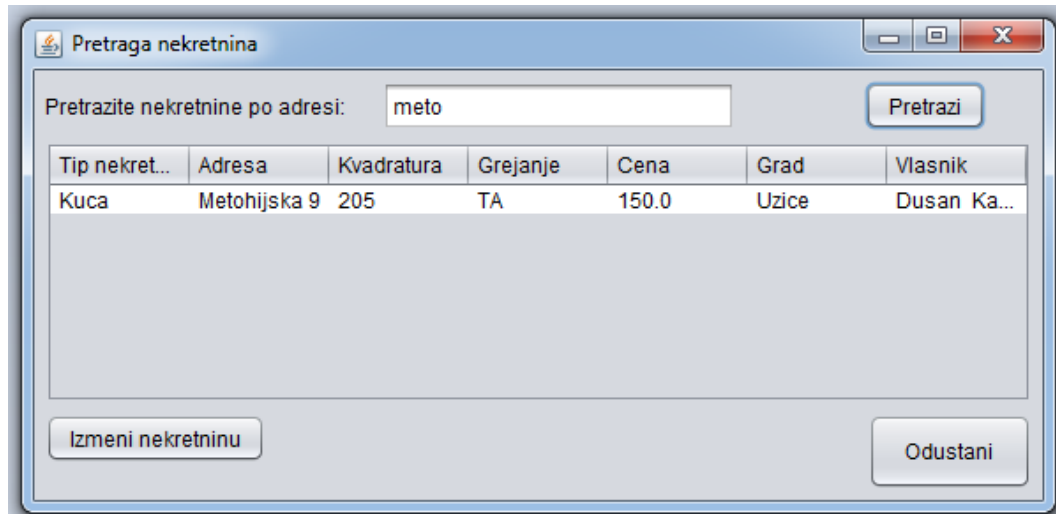
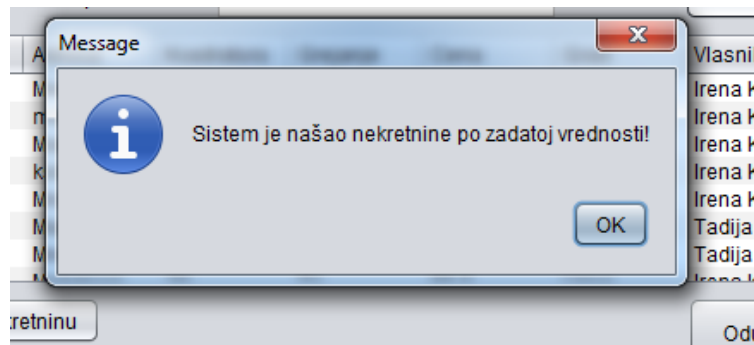


Osnovni scenario SK

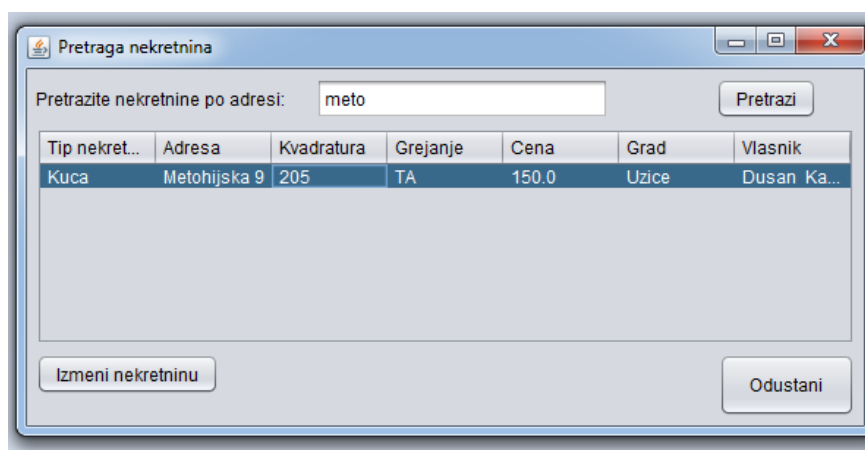
1. **Korisnik unosi** vrednost po kojoj pretražuje nekretnine. (APUSO)
Opis akcije: Korisnik unosi kriterijum pretrage (Adresa nekretnine) u polje za pretragu nekretnina. Ukoliko želi prikaz liste svih nekretnina korisnik ne unosi kriterijum ili unosi "*".



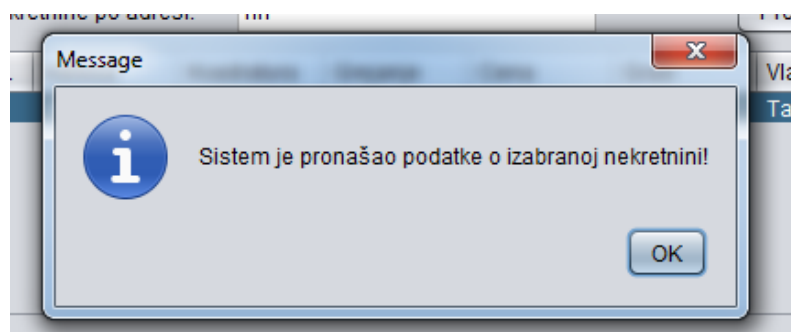
2. **Korisnik poziva sistem** da nađe nekretnine po zadatoj vrednosti. (APSO)
Opis akcije: Korisnik klikom na dugme "Pretraži" poziva sistemsku operaciju `PretraziNekretnine(Nekretnina, List< Nekretnina >)`.
3. **Sistem traži** nekretnine po zadatoj vrednosti. (SO)
4. **Sistem prikazuje korisniku** pronađene nekretnine poruku: „**Sistem** je našao nekretnine po zadatoj vrednosti”. (IA)



5. **Korisnik** bira nekretninu. (APUSO)
Opis akcije: Korisnik selektuje nekretninu u tabeli.



6. **Korisnik** poziva **sistem** da vrati podatke o nekretnini. (APSO)
Opis akcije: Korisnik klikom na dugme "Izmeni nekretninu" poziva sistemsku operaciju `VratiPodatkeONekretnini(Nekretnina)`
7. **Sistem** traži podatke o nekretnini. (SO)
8. **Sistem** prikazuje **korisniku** podatke o nekretnini i poruku: „**Sistem** je pronašao podatke o izabranoj nekretnini”. (IA)



Unos nekretnine

Unesite prezime vlasnika: Pretrazi

Ime	Prezime	Datum rođenja
Dusan	Karalic	1939-11-04

Tip nekretnine: Kuca

Adresa: Metohijska 9

Grad: Kragujevac

Kvadratura(kvm): 205

Grejanje: TA

Cena (EUR/mesec): 150.0

Izmeni Odustani

9. **Korisnik** menja podatke o nekretnini. (APUSO)

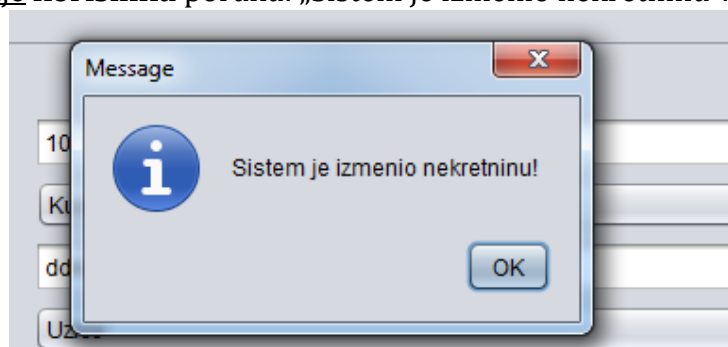
10. **Korisnik** kontrolira da li je korektno uneo podatke o nekretnini. (ANSO)

11. **Korisnik** poziva sistem da zapamti podatke o nekretnini. (APSO)

Opis akcije: Korisnik klikom na dugme "Izmeni" poziva sistemsku operaciju *IzmeniNekretninu(Nekretnina)* koja menja podatke o nekretnini.

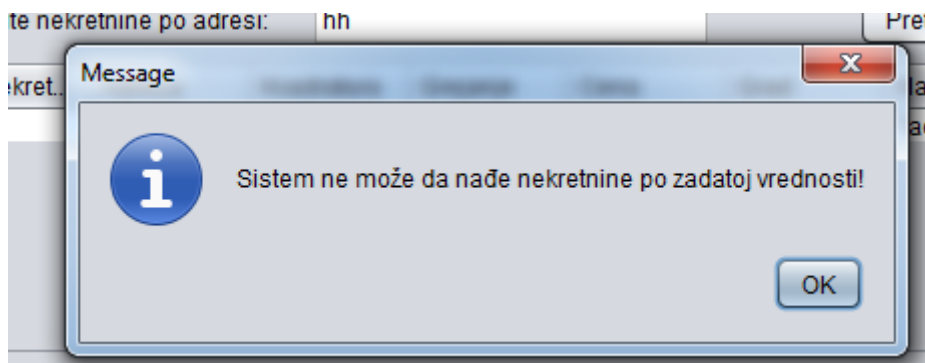
12. **Sistem** pamti podatke o nekretnini. (SO)

13. **Sistem** prikazuje korisniku poruku: „Sistem je izmenio nekretninu“. (IA)

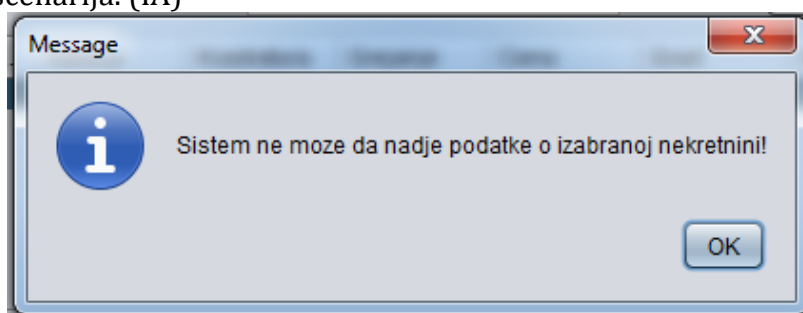


Alternativna scenarija

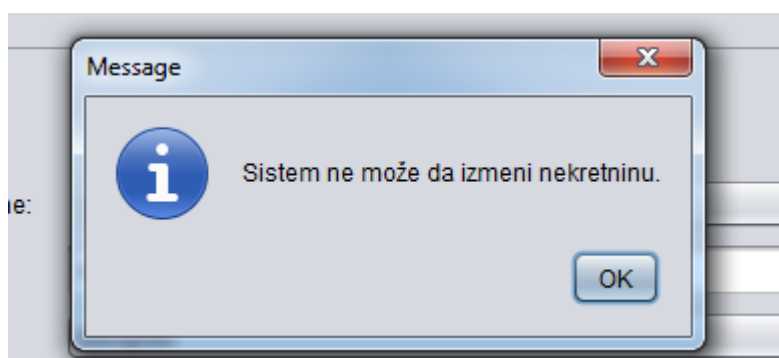
4.1 Ukoliko **sistem** ne može da nađe nekretninu on prikazuje **korisniku** poruku: „Sistem ne može da nađe nekretnine po zadatoj vrednosti“. Prekida se izvršenje scenarija. (IA)



8.1 Ukoliko **sistem** ne može da nađe podatke o nekretnini, obaveštava **korisnika** o tome: „**Sistem** ne može da nađe podatke o izabranoj nekretnini“. Prekida se izvršenje scenarija. (IA)



13.1. Ukoliko **sistem** ne može da zapamti izmenjene podatke o nekretnini on prikazuje **korisniku** poruku: „**Sistem** ne može da izmeni nekretninu“. (IA)



SK3:Slučaj korišćenja – Pretraga nekretnina

Naziv SK

Pretraga nekretnina

Aktori SK

Korisnik

Učesnici SK

Korisnik i sistem (program)

Preduslov: **Sistem** je uključen i **korisnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa nekretninama.

Tip neket...	Adresa	Kvadratura	Grejanje	Cena	Grad	Vlasnik
--------------	--------	------------	----------	------	------	---------

Osnovni scenario SK

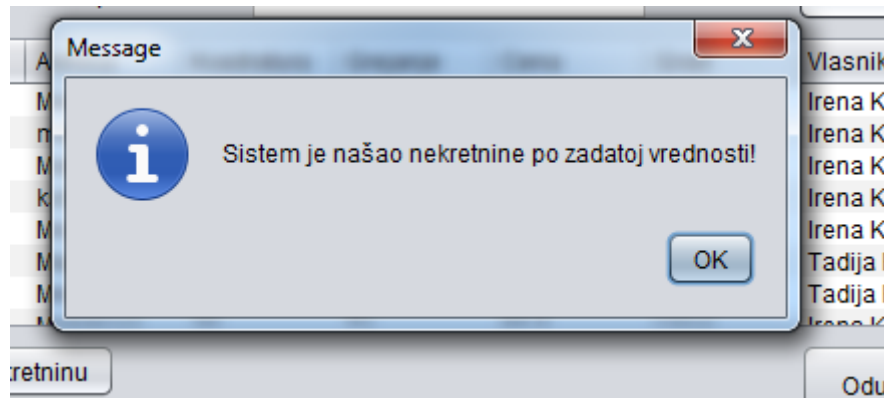
1. **Korisnik unosi** vrednost po kojoj pretražuje nekretnine. (APUS0)

Opis akcije: *Korisnik unosi kriterijum pretrage (Adresa nekretnine) u polje za pretragu nekretnina.*

Ukoliko želi prikaz liste svih nekretnina korisnik ne unosi kriterijum ili unosi "".*

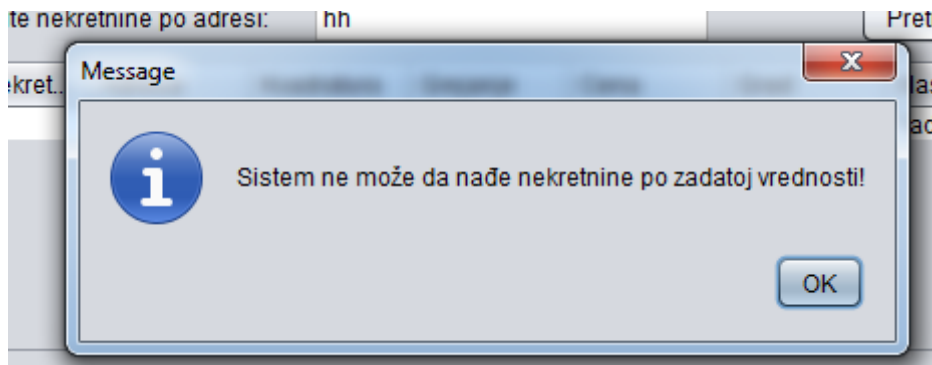
Tip neket...	Adresa	Kvadratura	Grejanje	Cena	Grad	Vlasnik
--------------	--------	------------	----------	------	------	---------

2. **Korisnik poziva sistem** da nađe nekretnine po zadatoj vrednosti. (APSO)
Opis akcije: Korisnik klikom na dugme "Pretraži" poziva sistemsku operaciju *PretraziNekretnine(Nekretnina, List< Nekretnina >)*.
3. **Sistem traži** nekretnine po zadatoj vrednosti. (SO)
4. **Sistem prikazuje korisniku** podatke o nekretninama i poruku: „**Sistem** je našao nekretnine po zadatoj vrednosti“. (IA)



Alternativna scenarija

- 4.1 Ukoliko **sistem** ne može da nađe nekretnine on prikazuje **korisniku** poruku: „**Sistem** ne može da nađe nekretnine po zadatoj vrednosti“. (IA)



SK4: Slučaj korišćenja – Unos vlasnika

Naziv SK

Unos vlasnika

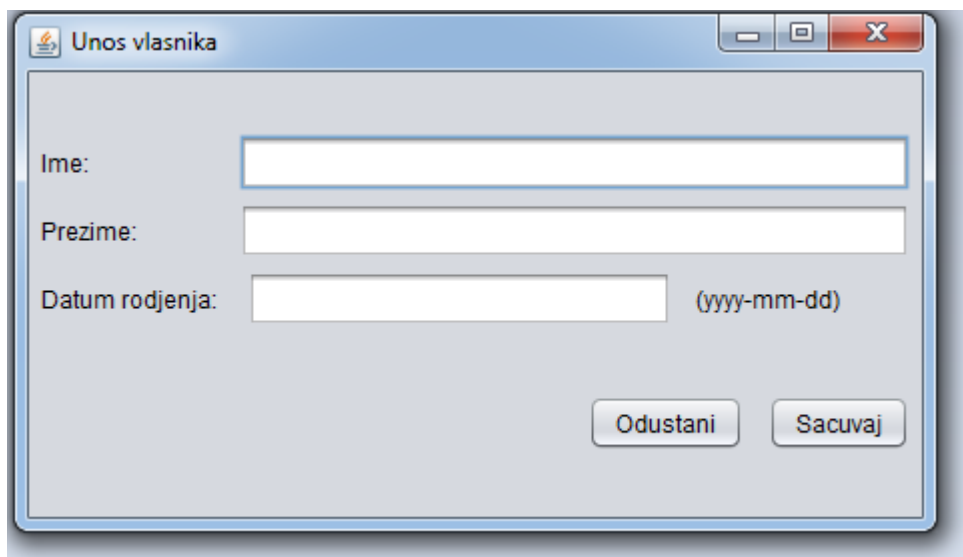
Aktori SK

Korisnik

Učesnici SK

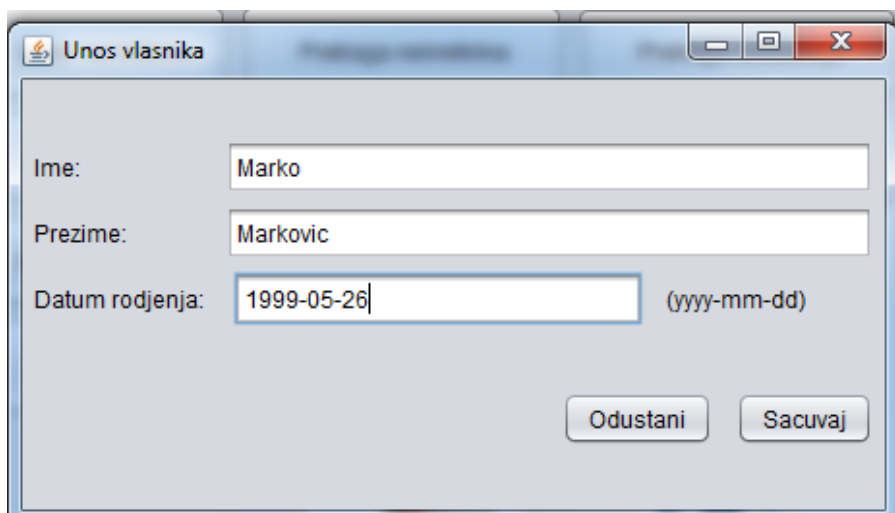
Korisnik i sistem (program)

Preduslov: **Sistem** je uključen i **korisnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa vlasnicima



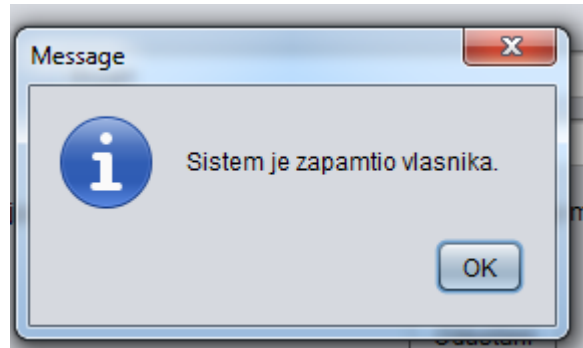
Osnovni scenario SK

1. **Korisnik** unosi podatke o vlasniku. (APUSO)
Opis akcije: *Korisnik unosi vrednosti u polja "Ime", "Prezime", "Datum rođenja".*



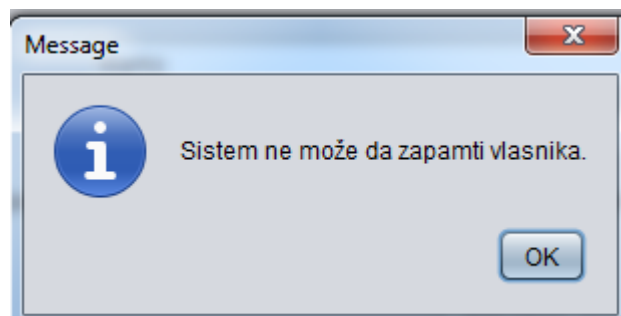
2. **Korisnik** kontroliše da li je korektno uneo podatke o vlasniku. (ANSO)

3. **Korisnik poziva sistem** da zapamti podatke o vlasniku. (APSO)
Opis akcije: Korisnik klikom na dugme "Sačuvaj" poziva sistemsku operaciju *ZapamtiVlasnika(Vlasnik)* koja pamti novog vlasnika.
4. **Sistem pamti** podatke o vlasniku. (SO)
5. **Sistem prikazuje korisniku** poruku: „**Sistem** je zapamtio vlasnika“. (IA)



Alternativna scenarija

- a. Ukoliko **sistem** ne može da zapamti podatke o novom vlasniku on prikazuje **korisniku** poruku: „**Sistem** ne može da zapamti vlasnika“. (IA)



SK5:Slučaj korišćenja – Unos klijenta

Naziv SK

Unos klijenta

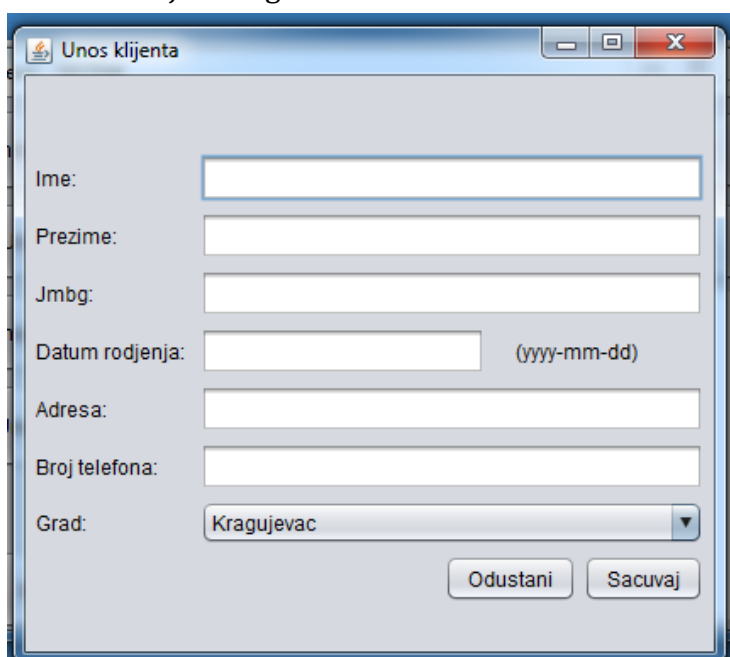
Aktori SK

Korisnik

Učesnici SK

Korisnik i sistem (program)

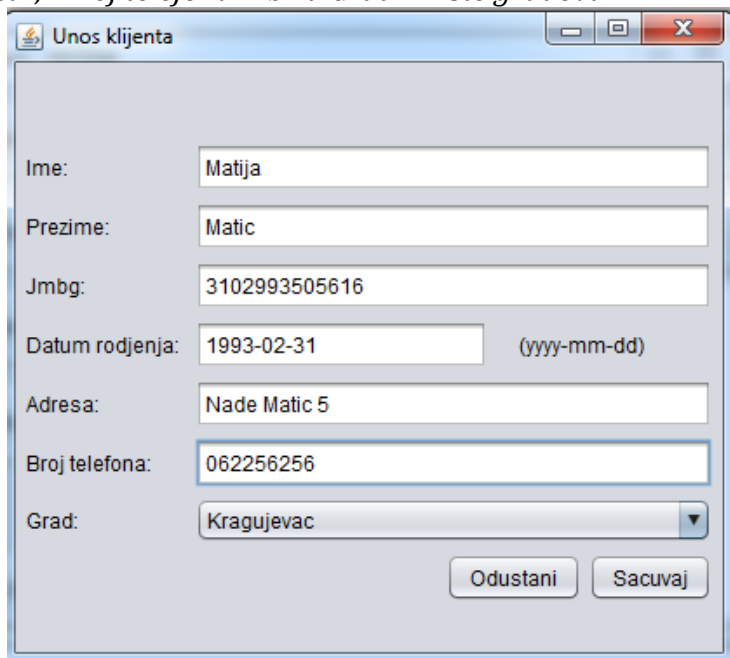
Preduslov: **Sistem** je uključen i **korisnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa klijentima. Učitana je lista gradova.



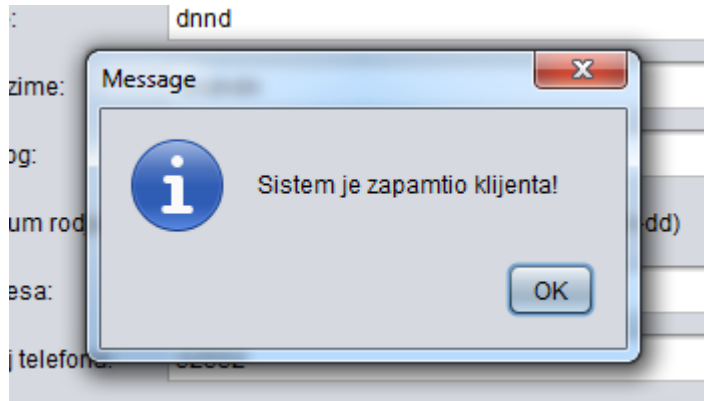
Osnovni scenario SK

1. **Korisnik unosi** podatke o klijentu. (APUSO)

Opis akcije: Korisnik unosi vrednosti u polja "Ime", "Prezime", "JMBG", "Datum rodjenja", "Adresa", "Broj telefona" i bira Grad iz liste gradova.

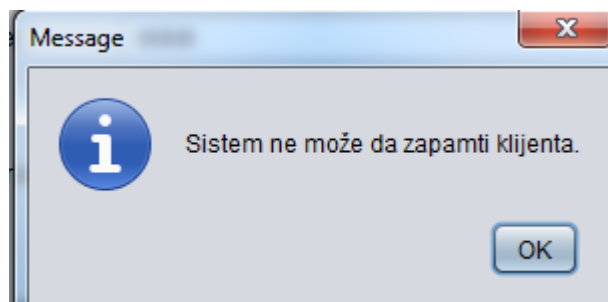


2. **Korisnik** kontrolira da li je korektno uneo podatke o klijentu. (ANSO)
3. **Korisnik** poziva **sistem** da zapamti podatke o klijentu. (APSO)
Opis akcije: Korisnik klikom na dugme "Sačuvaj" poziva sistemsku operaciju ZapamtiKlijenta(Klijent)
4. **Sistem** pamti podatke o klijentu. (SO)
5. **Sistem** prikazuje **korisniku** poruku: „Sistem je zapamtio klijenta“. (IA)



Alternativna scenarija

- 5.1 Ukoliko **sistem** ne može da zapamti podatke o novom klijentu on prikazuje **korisniku** poruku: „**Sistem** ne može da zapamti klijenta“. (IA)



SK6: Slučaj korišćenja – Izmena klijenta

Naziv SK

Izmenaklijenta

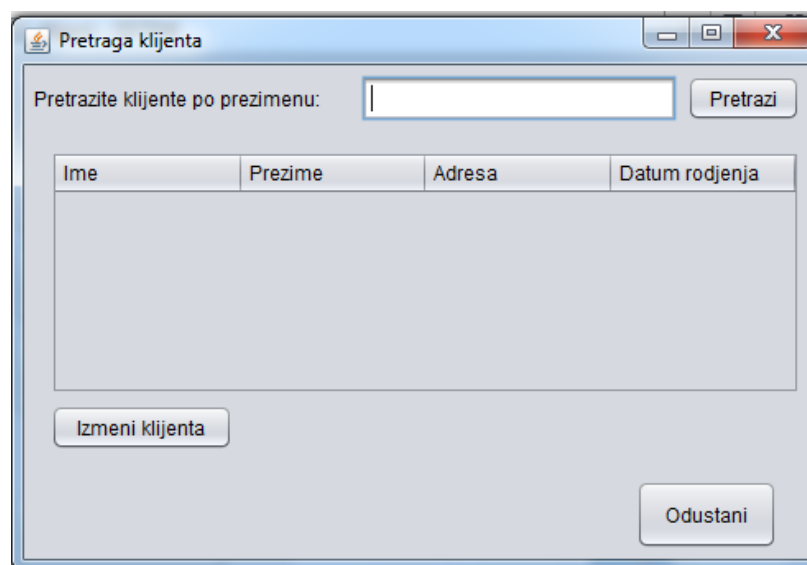
Aktori SK

Korisnik

Učesnici SK

Korisnik i sistem (program)

Preduslov: **Sistem** je uključen i **korisnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa klijentom. Učitana je lista gradova.

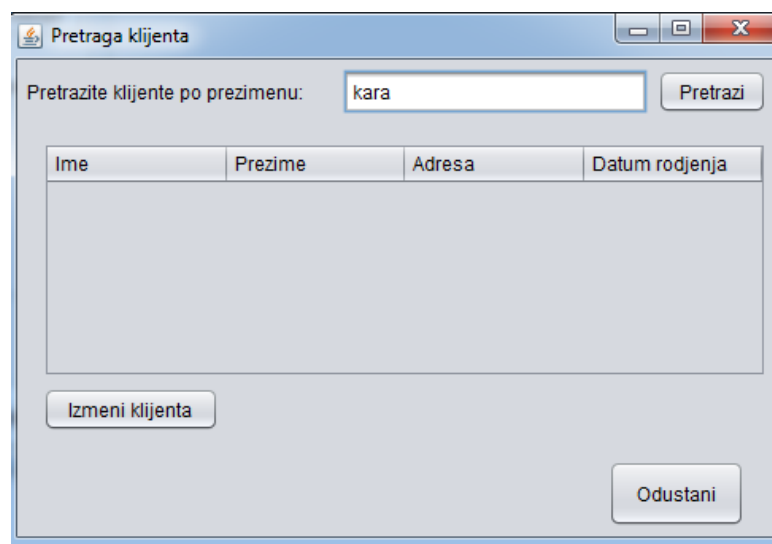


Osnovni scenario SK

1. **Korisnik** unosi vrednost po kojoj pretražuje klijente. (APUSO)

Opis akcije: *Korisnik unosi kriterijum pretrage (Prezime klijenta) u polje za pretragu klijenata. Ukoliko želi prikaz liste svih klijenata korisnik ne unosi kriterijum ili*

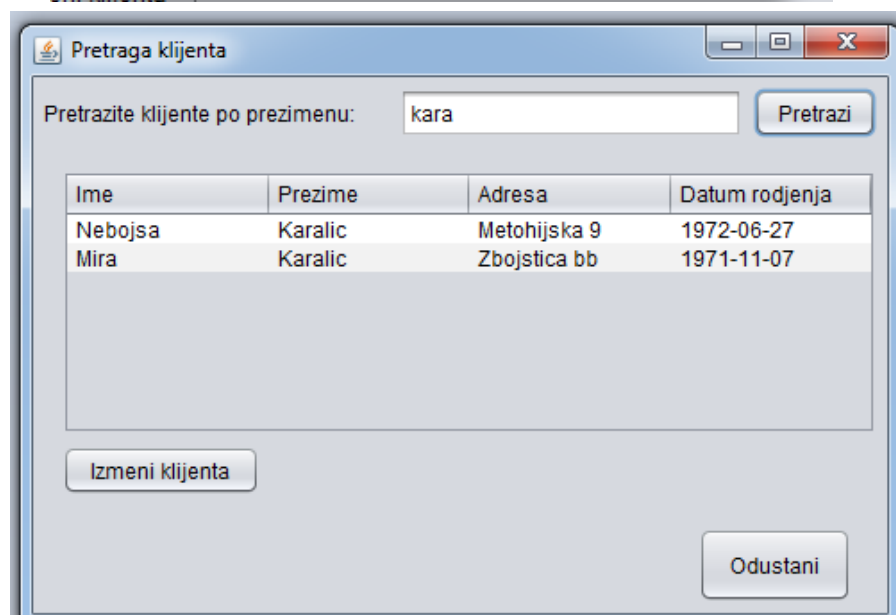
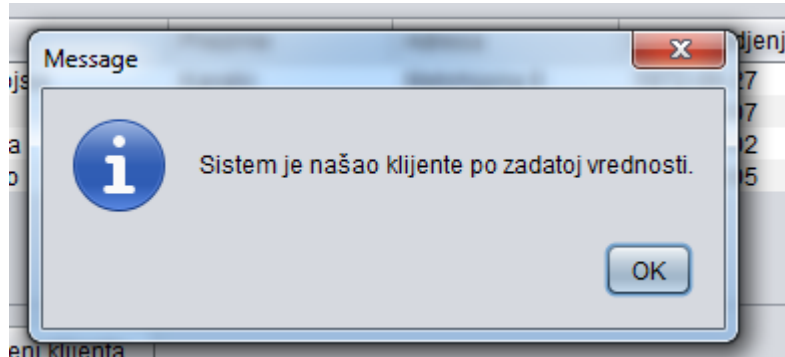
14. *unos*i““““.



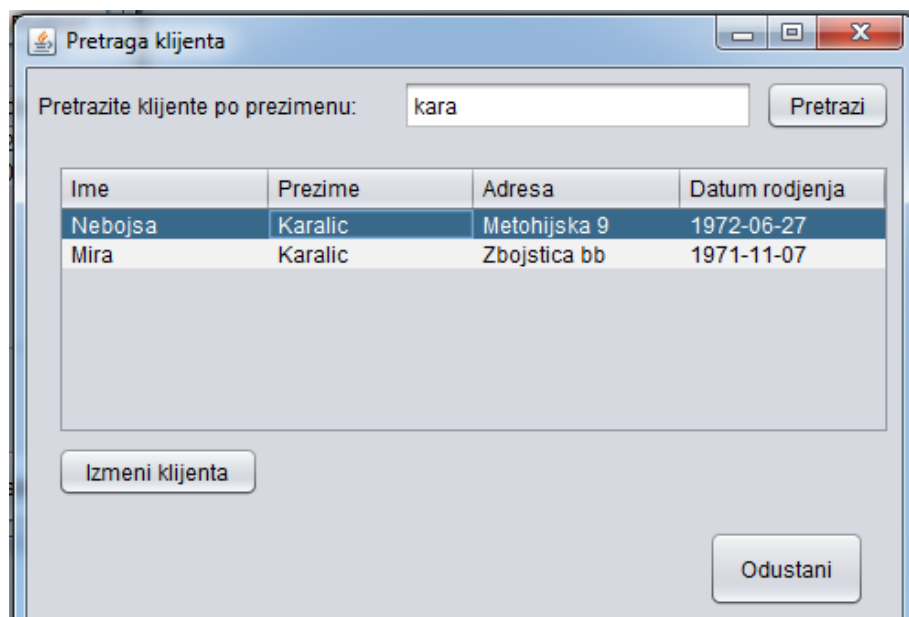
2. **Korisnik** poziva **sistem** da nađe klijente po zadatoj vrednosti. (APSO)

Opis akcije: Korisnik klikom na dugme "Pretraži" poziva sistemsku operaciju *PretraziKlijente(Klijent, List< Klijent >)*.

3. **Sistem traži** klijente po zadatoj vrednosti. (SO)
4. **Sistem prikazuje korisniku** pronađene klijente i poruku: „**Sistem** je našao klijente po zadatoj vrednosti”. (IA)

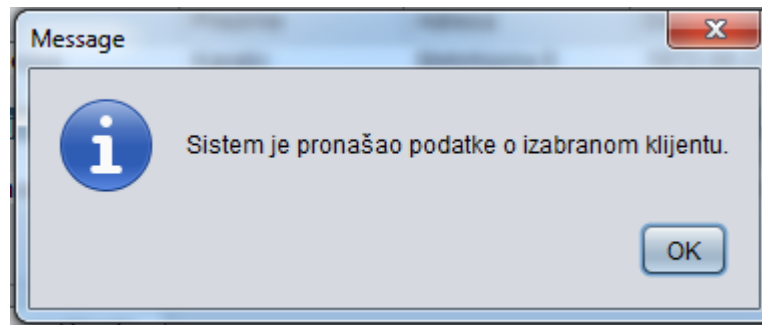


5. **Korisnik bira** klijenta. (APUSO)

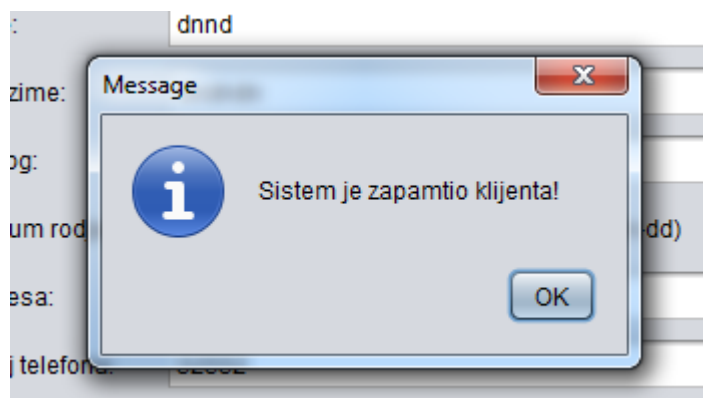


6. **Korisnik poziva sistem** da vrati podatke o klijentu. (APSO)
Opis akcija: Korisnik klikom na dugme "Izmeni klijenta" poziva sistemsku operaciju *VratiPodatkeOKlijentu(Klijent)*.

7. **Sistem** traži podatke o klijentu. (SO)
8. **Sistem** prikazuje korisniku podatke o klijentu i poruku: „**Sistem** je pronašao podatke o izabranom klijentu”. (IA)

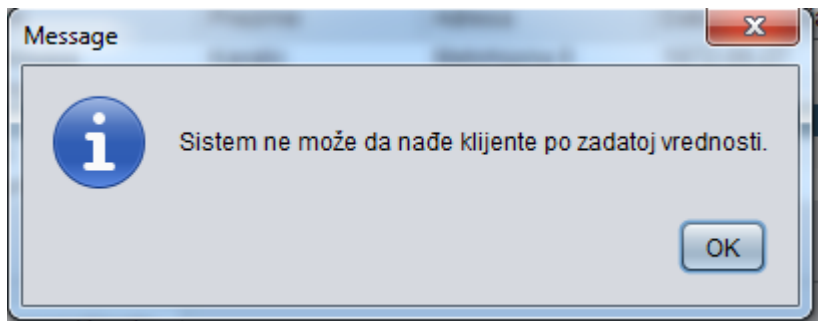


9. **Korisnik** unosi (menja) podatke o klijentu. (APUSO)
10. **Korisnik** kontrolira da li je korektno uneo podatke o klijentu. (ANSO)
11. **Korisnik** poziva sistem da zapamti podatke o klijentu. (APSO)
Opis akcije: Korisnik klikom na dugme "Izmeni" poziva sistemsku operaciju IzmeniKlijenta(Klijent)
12. **Sistem** pamti podatke o klijentu. (SO)
13. **Sistem** prikazuje korisniku poruku: „Sistem je zapamtio klijenta”. (IA)

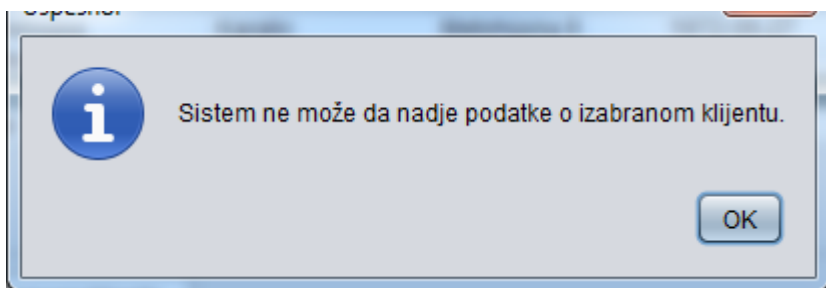


Alternativna scenarija

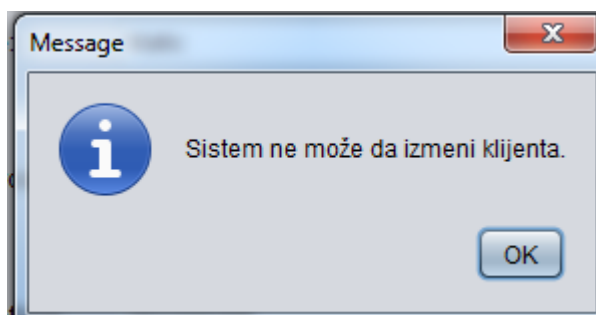
- 4.1 Ukoliko **sistem** ne može da nađe klijente prikazuje **korisniku** poruku: „Sistem ne može da nađe klijente po zadatoj vrednosti“. Prekida se izvršenje scenarija. (IA)



- 8.1 Ukoliko **sistem** ne može da nađe podatke o klijentu, obaveštava **korisnika** o tome: „**Sistem** ne može da nađe podatke o izabranom klijentu“. Prekida se izvršenje scenarija. (IA)



- 13.1 Ukoliko **sistem** ne može da zapamti izmenjene podatke o klijentu on prikazuje **korisniku** poruku: „**Sistem** ne može da izmeni klijenta“. (IA)



SK7: Slučaj korišćenja- Pretraga klijenta

Naziv SK

Pretraga klijenta

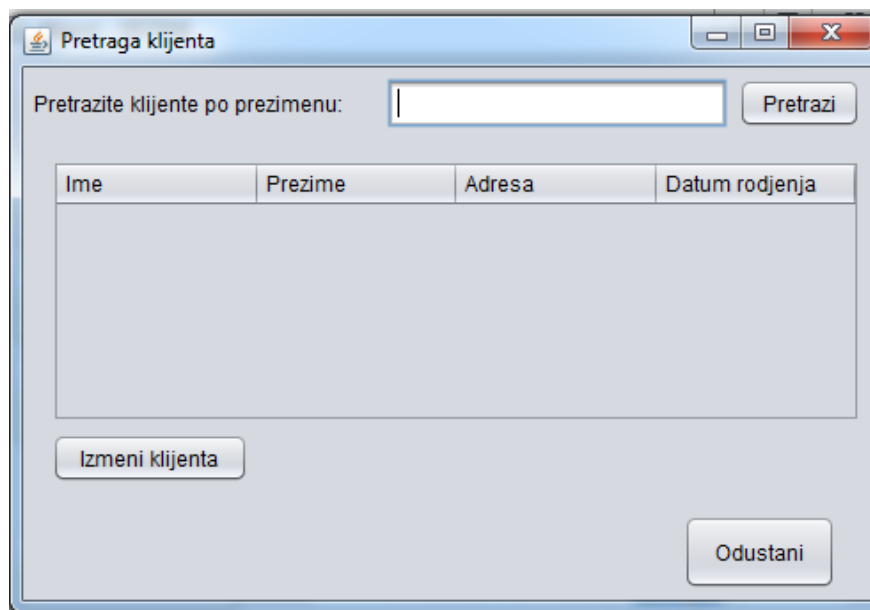
Aktori SK

Korisnik

Učesnici SK

Korisnik i sistem (program)

Preduslov: **Sistem** je uključen i **korisnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa klijentima.

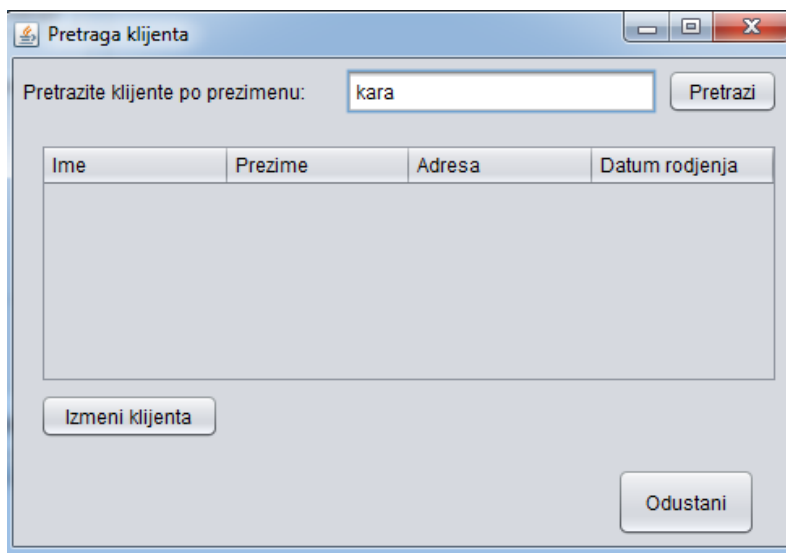


Osnovni scenario SK

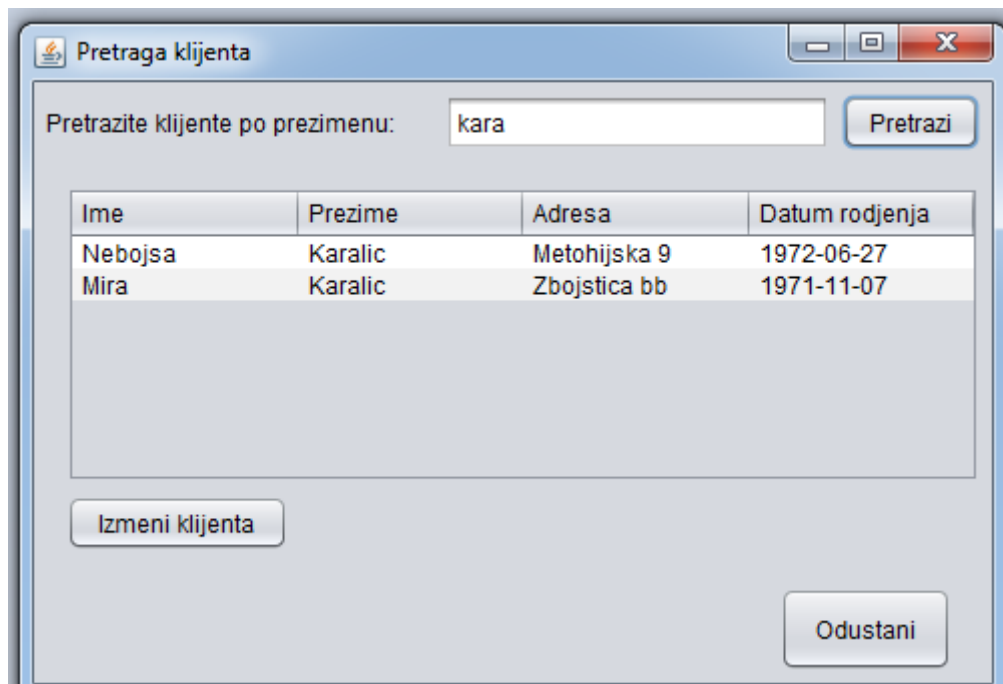
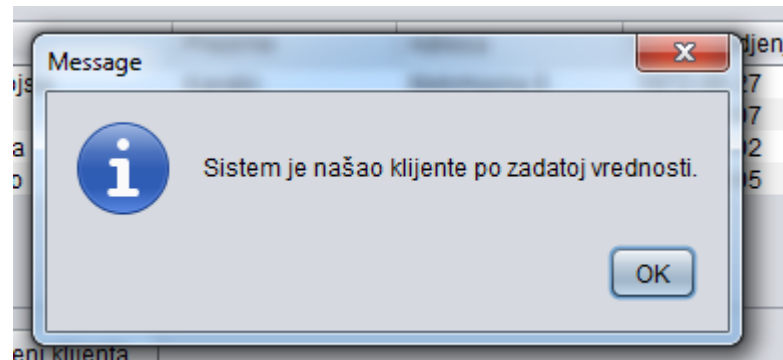
1. **Korisnik** unosi vrednost po kojoj pretražuje klijente. (APUS0)

Opis akcije: *Korisnik unosi kriterijum pretrage (Prezime klijenta) u polje za pretragu klijenata. Ukoliko želi prikaz liste svih klijenata korisnik ne unosi kriterijum ili unosi “*”.*

- 2.

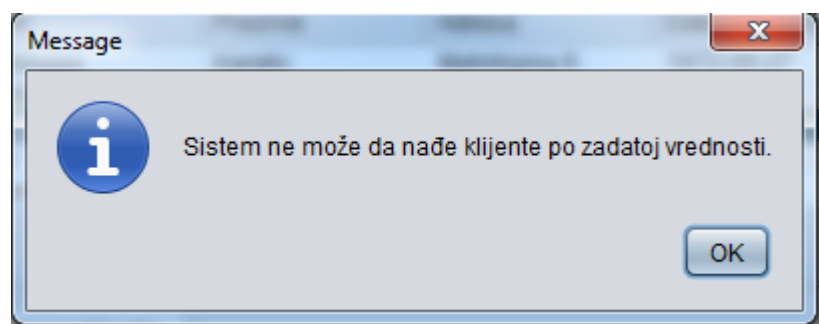


3. **Korisnik poziva sistem** da nađe klijente po zadatoj vrednosti. (APSO)
Opis akcije: Korisnik klikom na dugme "Pretraži" poziva sistemsku operaciju *PretraziKlijente(Klijent, List< Klijent >)*.
4. **Sistem traži** klijente po zadatoj vrednosti. (SO)
5. **Sistem prikazuje korisniku** podatke o klijentima i poruku: „**Sistem** je našao klijente po zadatoj vrednosti“. (IA)



Alternativna scenarija

- 4.1 Ukoliko **sistem** ne može da nađe klijente on prikazuje **korisniku** poruku: „**Sistem** ne može da nađe klijente po zadatoj vrednosti“. (IA)



SK8: Slučaj korišćenja – Unos zaduženja

Naziv SK

Unos zaduženja

Aktori SK

Korisnik

Učesnici SK

Korisnik i sistem (program)

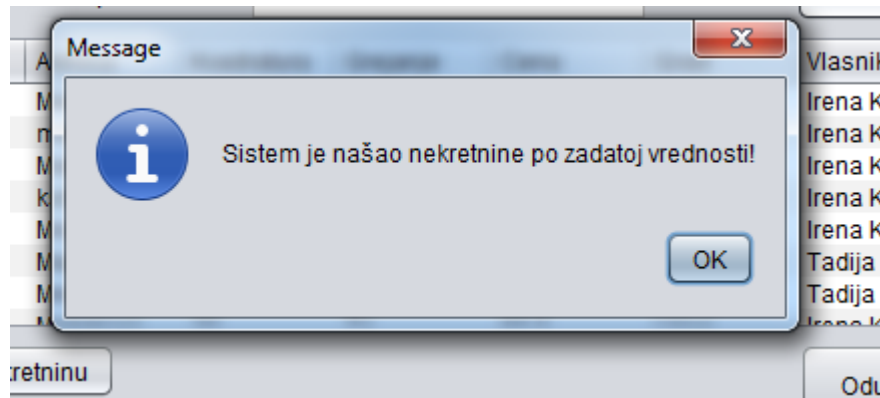
Preduslov: **Sistem** je uključen i **korisnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa zaduženjima.

Osnovni scenario SK

1. **Korisnik unosi** vrednost po kojoj pretražuje nekretnine. (APUSO)

Opis akcije: Korisnik unosi kriterijum pretrage (Adresa nekretnine) u polje za pretragu nekretnina. Ukoliko želi prikaz liste svih nekretnina korisnik ne unosi kriterijum ili unosi “*“.

2. **Korisnik poziva sistem** da nađe nekretnine po zadatoj vrednosti. (APSO)
Opis akcije: Korisnik klikom na dugme "Pretraži" poziva sistemsku operaciju *PretraziNekretnine(Nekretnina, List< Nekretnina >)*.
3. **Sistem traži** nekretnine po zadatoj vrednosti. (SO)
4. **Sistem prikazuje korisniku** podatke o nekretninama i poruku: „Sistem je našao nekretnine po zadatoj vrednosti“. (IA)



Unos zaduzenja

Pretražite nekretnine po adresi:

Tip nekretnine	Adresa	Kvadratura	Grejanje	Cena	Grad	Vlasnik
Kuca	Metohijska 9	205	TA	150.0	Uzice	Dusan Karalic

Pretražite klijente po prezimenu:

Ime	Prezime	Adresa	Datum rođenja

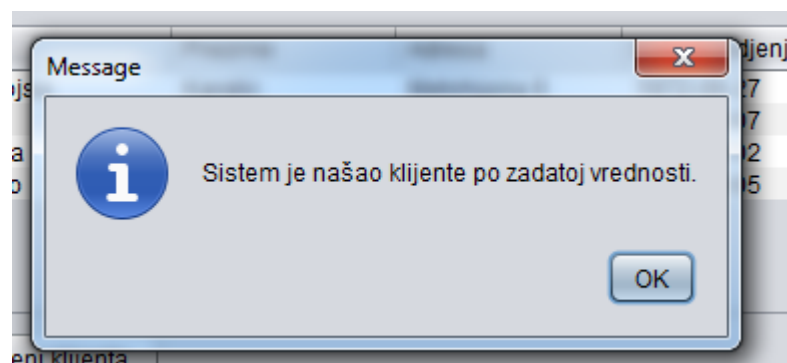
Datum zaduzenja:

Datum razduzenja:

5. **Korisnik unosi** vrednost po kojoj pretražuje klijente. (APUSO)

Opis akcije: Korisnik unosi kriterijum pretrage (Prezime klijenta) u polje za pretragu klijenata. Ukoliko želi prikaz liste svih klijenata korisnik ne unosi kriterijum ili unosi “*“.

6. **Korisnik poziva sistem** da nađe klijente po zadatoj vrednosti. (APSO)
Opis akcije: Korisnik klikom na dugme “Pretraži” poziva sistemsku operaciju *PretraziKlijente(Klijente, List< Klijente>)*.
7. **Sistem traži** klijente po zadatoj vrednosti. (SO)
8. **Sistem prikazuje korisniku** podatke o nekretninama i poruku: „Sistem je našao klijente po zadatoj vrednosti“. (IA)



9. **Korisnik** vrši izbor klijenta, nekretnine i unosi ostale podatke o zaduženju. (APUSO)
Opis akcije: *Korisnik selektuje nekretninu iz tabele, bira klijenta tako što ih selektuje iz tabele. Korisnik unosi vrednosti u polje "Datum zaduženja" i "Datum razduženja".*

Unos zaduzenja

Pretražite nekretnine po adresi:

Tip nekretnine	Adresa	Kvadratura	Grejanje	Cena	Grad	Vlasnik
Kuca	Metohijska 9	205	TA	150.0	Uzice	Dusan Karalic

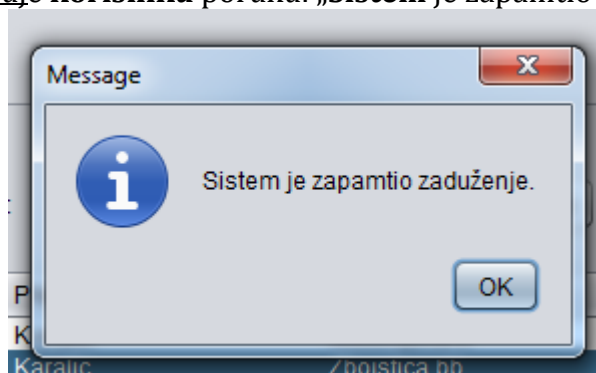
Pretražite klijente po prezimenu:

Ime	Prezime	Adresa	Datum rođenja
Nebojsa	Karalic	Metohijska 9	1972-06-27
Mira	Karalic	Zbojsica bb	1971-11-07
Matija	Matic	Nade Matic 5	1993-03-03
Iel	dididi	dddd	2018-02-02

Datum zaduzenja:

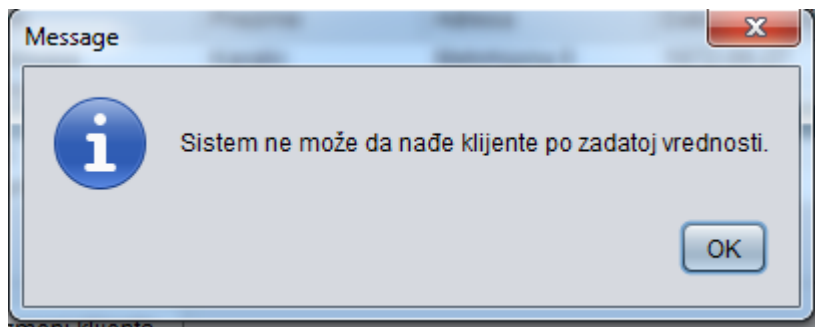
Datum razduzenja:

10. **Korisnik** kontrolise da li je uneo zaduženje. (ANSO)
 11. **Korisnik** poziva sistem da zapamti podatke o zaduženju. (APSO)
Opis akcije: *Korisnik klikom na dugme "Sačuvaj" poziva sistemsku operaciju SacuvajZaduzenje(Zaduzenje).*
 12. **Sistem** pamti podatke o zaduženjima. (SO)
 13. **Sistem** prikazuje korisniku poruku: „**Sistem** je zapamtio zaduženje“. (IA)

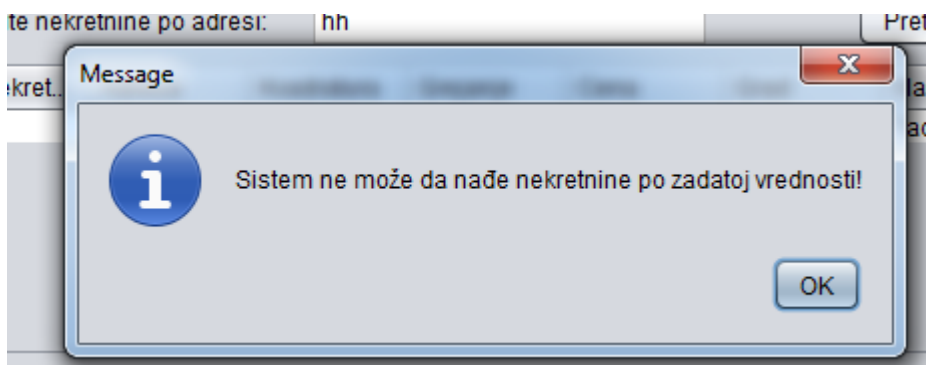


Alternativna scenarija

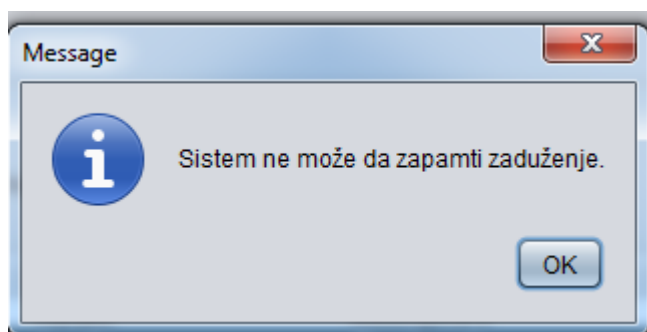
- 4.1 Ukoliko **sistem** ne može da nađe klijente on prikazuje **korisniku** poruku: „**Sistem** ne može da nađe klijente po zadatoj vrednosti“. (IA)



8.1 Ukoliko **sistem** ne može da nađe nekretnine on prikazuje **korisniku** poruku: „**Sistem** ne može da nađe nekretnine po zadatoj vrednosti“. (IA)



13.1 Ukoliko **sistem** ne može da zapamti podatke o zaduženjima, on prikazuje **korisniku** poruku „**Sistem** ne može da zapamti zaduženje“. (IA)



SK9: Slučaj korišćenja – Izmena zaduženja

Naziv SK

Izmena zaduženja

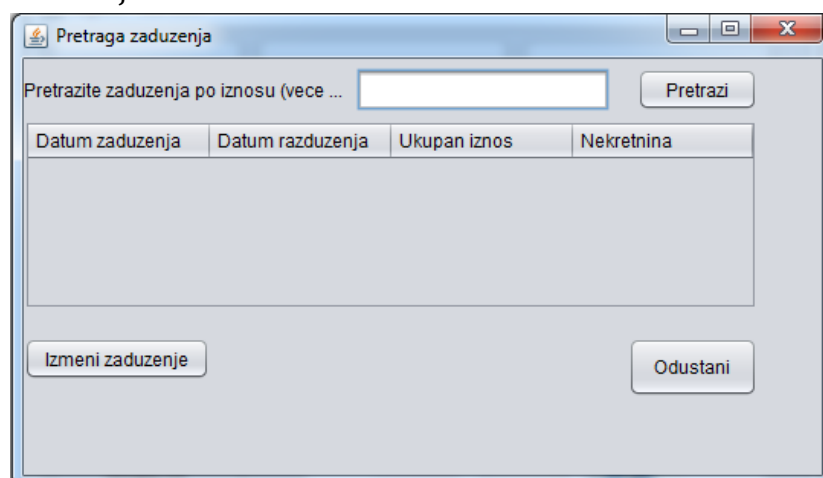
Aktori SK

Korisnik

Učesnici SK

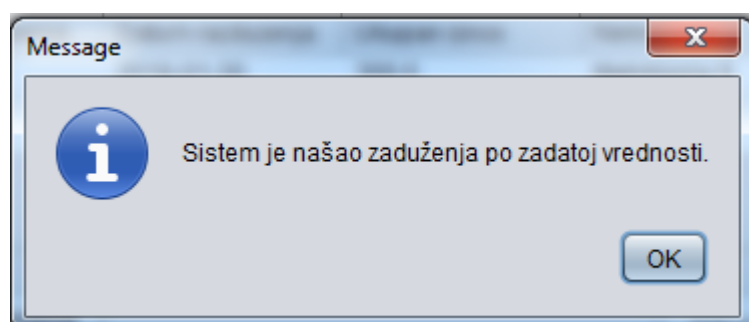
Korisnik i sistem (program)

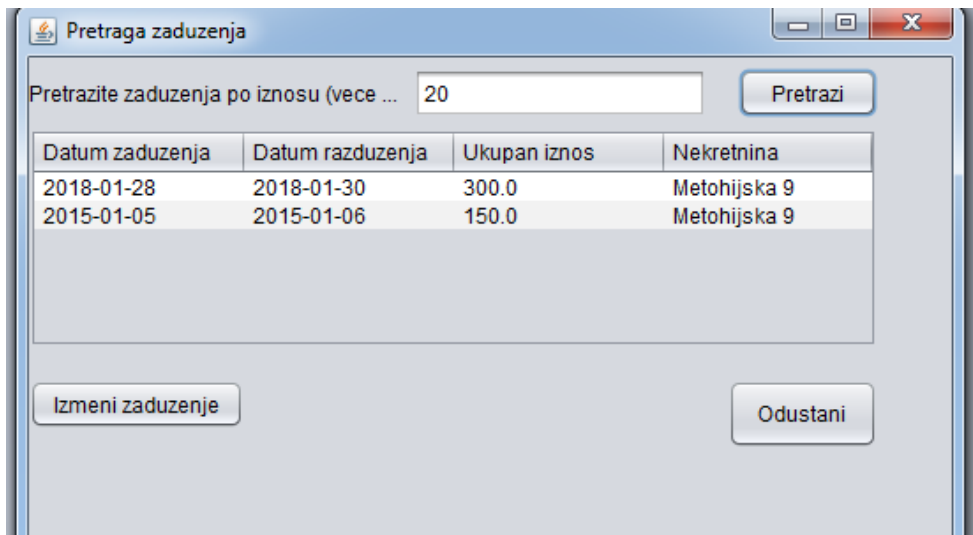
Preduslov: **Sistem** je uključen i **korisnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa zaduženjem.



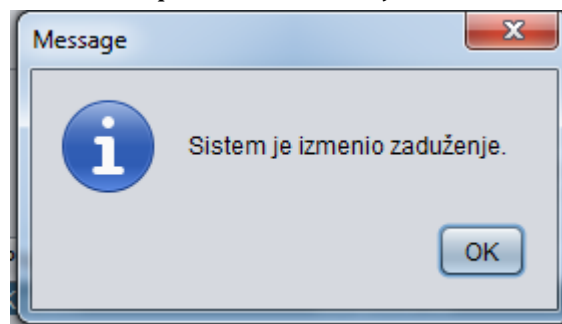
Osnovni scenario SK

1. **Korisnik** unosi vrednost po kojoj pretražuje zaduženja. (APUSO)
Opis akcije: Korisnik unosi kriterijum pretrage (iznos zaduzenja) u polje za pretragu zaduzenja. Ukoliko želi prikaz liste svih zaduzenja korisnik ne unosi kriterijum ili unosi "*".
- 2.
3. **Korisnik** poziva **sistem** da nađe zaduženja po zadatoj vrednosti. (APSO)
Opis akcije: Korisnik klikom na dugme "Pretraži" poziva sistemsku operaciju PretraziZaduzenja(Zaduzenje, List<Zaduzenje>).
4. **Sistem** traži zaduženja po zadatoj vrednosti. (SO)
5. **Sistem** prikazuje **korisniku** pronađena zaduženja i poruku: „**Sistem** je našao zaduženja po zadatoj vrednosti”. (IA)



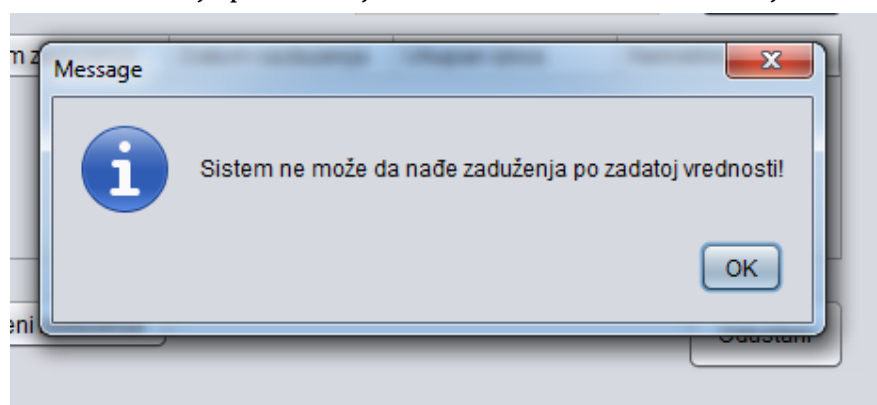


6. **Korisnik** bira zaduženje. (APUSO)
7. **Korisnik** poziva **sistem** da vrati podatke o zaduženju. (APSO)
Opis akcije: Korisnik klikom na dugme "Izmeni zaduženje" poziva sistemsku operaciju VratiPodatkeOZaduzenju(Zaduzenje)
8. **Sistem** traži podatke o zaduženju. (SO)
9. **Sistem** prikazuje **korisniku** podatke o zaduženju i poruku: „**Sistem** je pronašao podatke o izabranom zaduženju”. (IA)
10. **Korisnik** menja podatke o zaduženju. (APUSO)
11. **Korisnik** kontrolira da li je korektno uneo podatke o zaduženju. (ANSO)
12. **Korisnik** poziva **sistem** da zapamti podatke o zaduženju. (APSO)
Opis akcije: Korisnik klikom na dugme "Izmeni" poziva sistemsku operaciju IzmeniZaduzenje(Zaduzenje)
13. **Sistem** pamti podatke o zaduženju. (SO)
14. **Sistem** prikazuje **korisniku** poruku: „Sistem je izmenio zaduženje”. (IA)



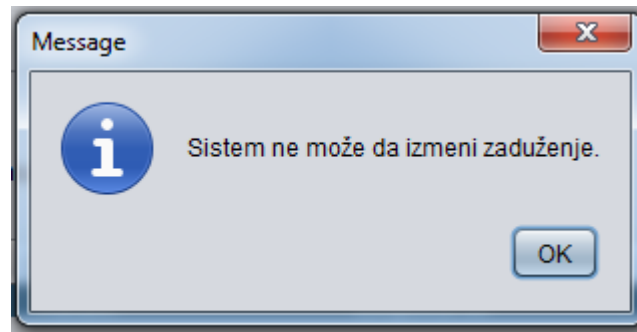
Alternativna scenarija

- 4.1 Ukoliko **sistem** ne može da nađe zaduženja on prikazuje **korisniku** poruku: „Sistem ne može da nađe zaduženja po zadatoj vrednosti”. Prekida se izvršenje scenarija. (IA)



8.1 Ukoliko **sistem** ne može da nađe podatke o zaduženju, obaveštava **korisnika** o tome: „**Sistem** ne može da nađe podatke o izabranom zaduženju“. Prekida se izvršenje scenarija. (IA)

14.1 Ukoliko **sistem** ne može da zapamti izmenjene podatke o zaduženju on prikazuje **korisniku** poruku: „**Sistem** ne može da izmeni zaduženje“. (IA)



SK10: Slučaj korišćenja – Pretraga zaduženje

Naziv SK

Pretraga zaduženja

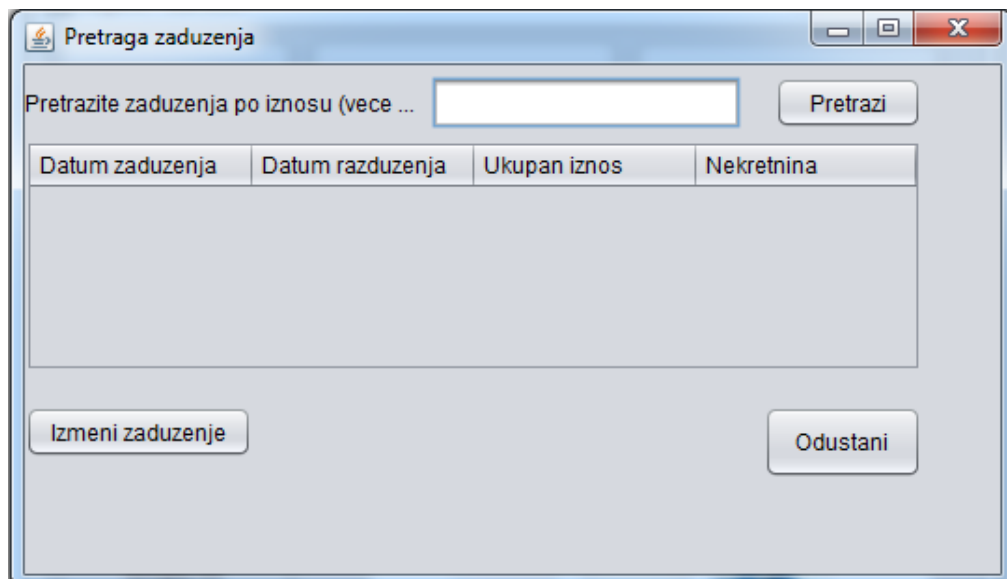
Aktori SK

Korisnik

Učesnici SK

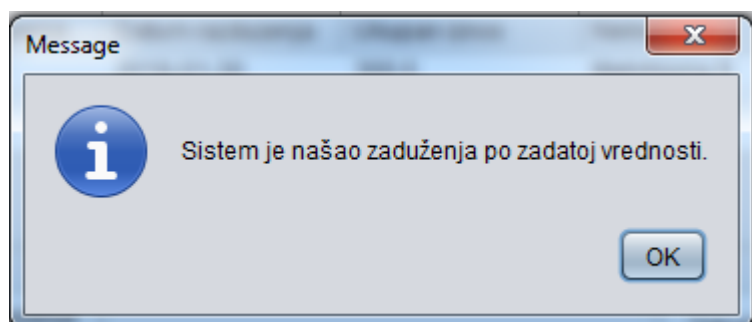
Korisnik i sistem (program)

Preduslov: **Sistem** je uključen i **korisnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa zaduženjima.



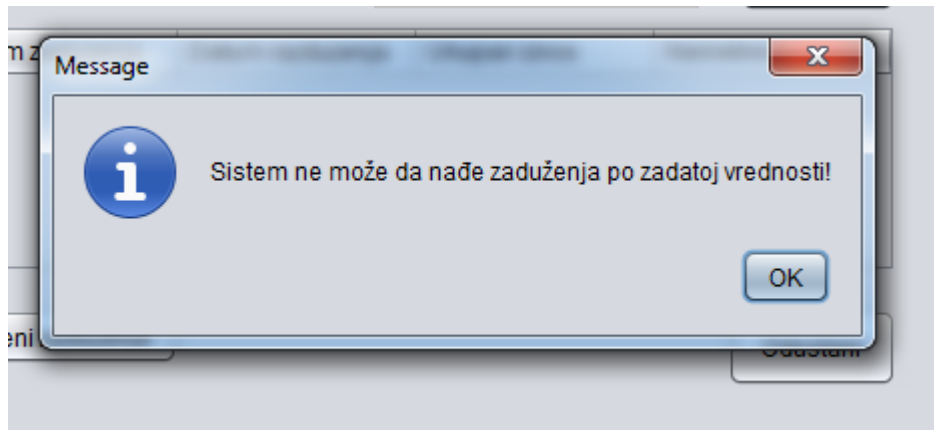
Osnovni scenario SK

5. **Korisnik** unosi vrednost po kojoj pretražuje zaduženja. (APUSO)
6. **Korisnik** poziva **sistem** da nađe zaduženja po zadatoj vrednosti. (APSO)
Opis akcije: Korisnik klikom na dugme "Pretraži" poziva sistemsku operaciju *PretraziZaduženja*(Zaduženje, List<Zaduženje>).
7. **Sistem** traži zaduženja po zadatoj vrednosti. (SO)
8. **Sistem** prikazuje **korisniku** podatke o zaduženjima i poruku: „**Sistem** je našao zaduženja po zadatoj vrednosti“. (IA)



Alternativna scenarija

4.1 Ukoliko **sistem** ne može da nađe zaduženja on prikazuje **korisniku** poruku: „**Sistem** ne može da nađe zaduženja po zadatoj vrednosti“. (IA)



SK11: Slučaj korišćenja – Pretraga vlasnika

Naziv SK

Pretraga vlasnika

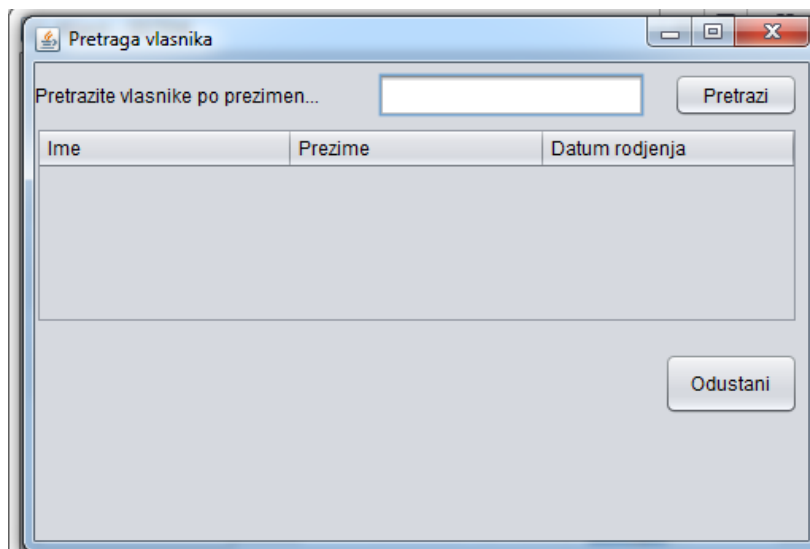
Aktori SK

Korisnik

Učesnici SK

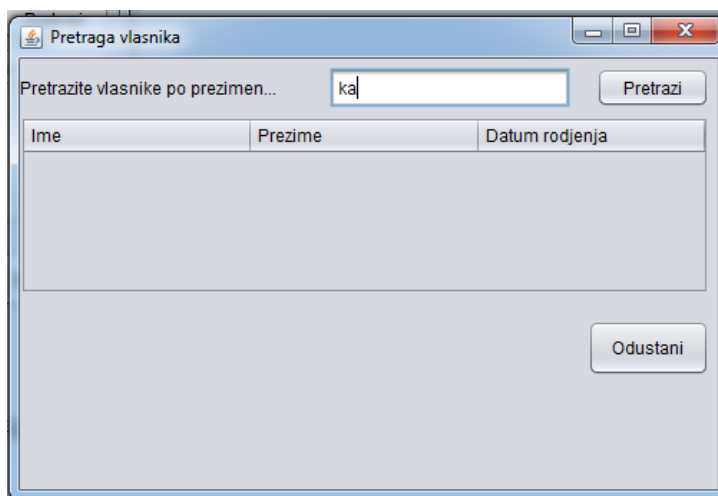
Korisnik i sistem (program)

Preduslov: **Sistem** je uključen i **korisnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa vlasnicima.



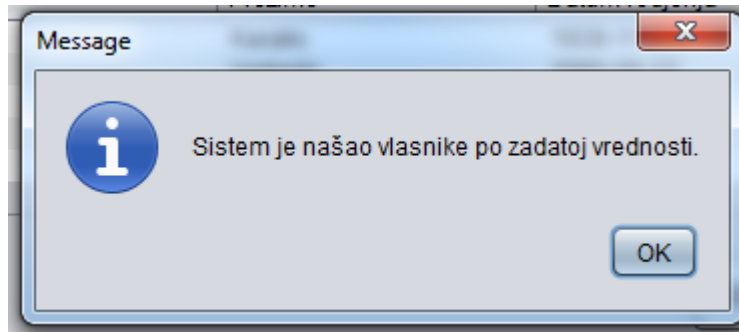
Osnovni scenario SK

1. **Korisnik** unosi vrednost po kojoj pretražuje vlasnike. (APUSO)



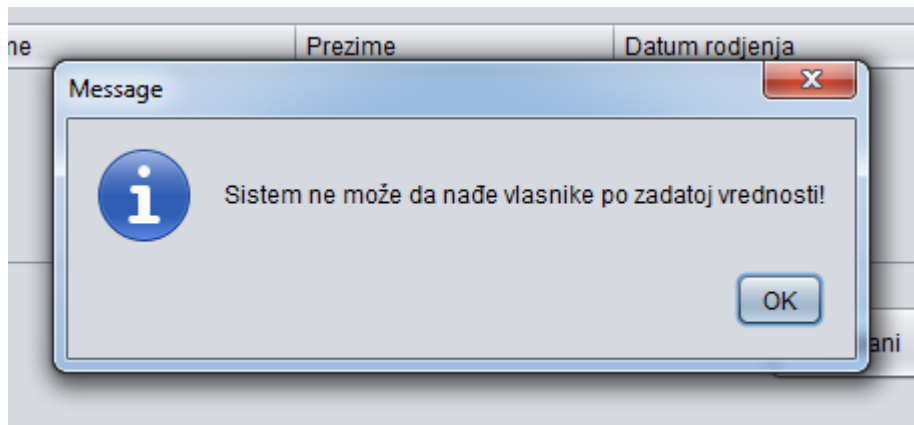
2. **Korisnik** poziva **sistem** da nađe vlasnike po zadatoj vrednosti. (APSO)
Opis akcije: Korisnik klikom na dugme "Pretraži" poziva sistemsku operaciju *PretraziVlasnike(Vlasnik, List< Vlasnik >)*.
3. **Sistem** traži vlasnike po zadatoj vrednosti. (SO)

4. **Sistem prikazuje korisniku** podatke o vlasnicima i poruku: „**Sistem** je našao vlasnike po zadatoj vrednosti“. (IA)



Alternativna scenarija

- 4.1 Ukoliko **sistem** ne može da nađe vlasnike on prikazuje **korisniku** poruku: „**Sistem** ne može da nađe vlasnike po zadatoj vrednosti“. (IA)



SK12: Slučaj korišćenja – Prijava korisnika

Naziv SK

Prijava korisnika

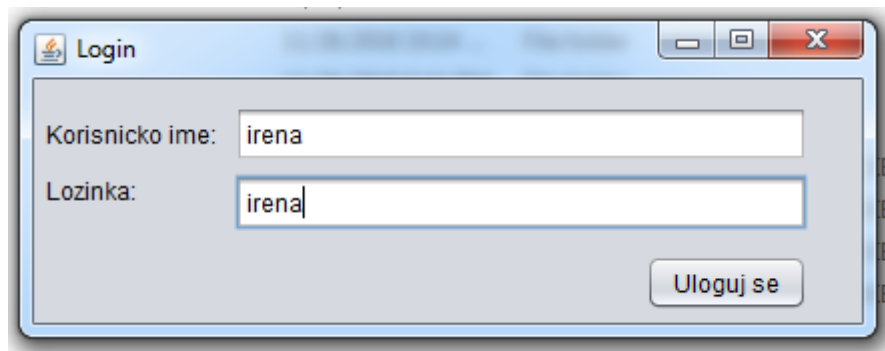
Aktori SK

Korisnik

Učesnici SK

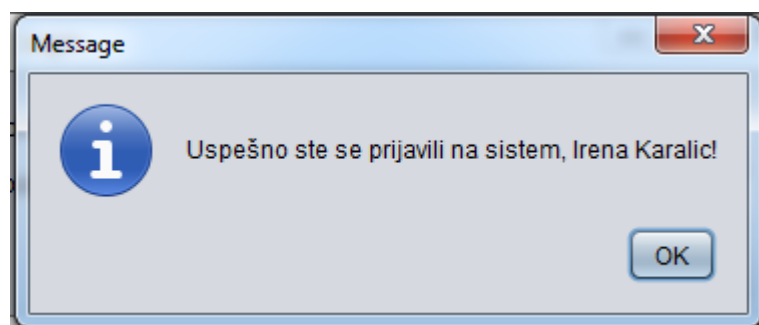
Korisnik i sistem (program)

Preduslov: **Sistem** je uključen i prikazuje formu za prijavljivanje **korisnika**.



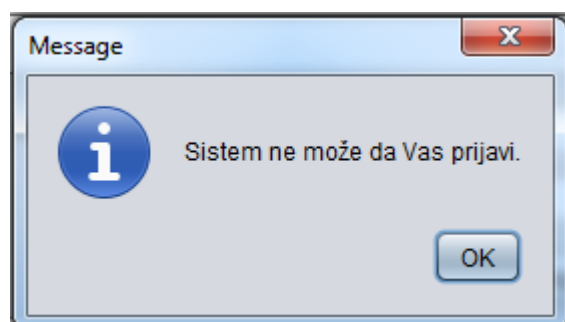
Osnovni scenario SK

1. **Korisnik** unosi podatke za autentifikaciju **korisnika**. (APUSO)
2. **Korisnik** poziva sistem da pronade **korisnika** sa zadatim podacima. (APSO)
3. **Sistem** pretražuje **korisnika**. (SO)
4. **Sistem** prikazuje poruku: "Uspešno ste se prijavili na sistem". (IA)



Alternativna scenarija

- 4.1. Ukoliko **sistem** ne može da nađe **korisnika** prikazuje poruku "**Sistem** ne može da vas prijavi". (IA)



3.3 Projektovanje aplikacione logike

Aplikaciona logika služi za opisivanje strukture i ponašanja softverskog sistema i projektuje se nezavisno od korisničkog interfejsa i obratno. Drugim rečima, aplikaciona logika (koja predstavlja Model u MVC paternu) nema znanje o tome gde se nalazi korisnički interfejs (koji predstavlja View u MVC paternu).

Aplikacioni serveri treba da obezbede servise koji će omogućiti realizaciju aplikacione logike softverskog sistema. Projektovanni aplikacioni server sadrži:

- deo za komunikaciju sa klijentom
- kontrolera aplikacione logike
- deo za komunikaciju sa skladištem podataka (broker baze podataka)
- deo koji sadrži poslovnu logiku

3.3.1 Komunikacija sa klijentima

Deo za komunikaciju podiže serverski soket koji dalje osluškuje mrežu. Kada klijent uspostavi konekciju, server generiše nit koja će biti odgovorna za dvosmernu vezu sa klijentom.

Klijent šalje zahtev za izvršenje neke od sistemskih operacija, dok odgovarajuća nit (dodeljena klijentu) prihvata zahtev i prosleđuje ga do kontrolera aplikacione logike. Nakon izvršenja sistemske operacija, kontroler vraća rezultat izvršenja operacije do niti koja je zadužena za tog klijenta. Rezultat zatim prosleđuje klijentu.

Komunikaciju između klijenta i server se obavlja razmenom objekta klase **“ServerskiOdgovor”** i klase **“KlijentskiZahtev”**.

```
public class KlijentskiZahtev implements Serializable {
```

```
    private int operacija;  
    private Object objekatOperacije;
```

```
    public KlijentskiZahtevObjekat() {  
    }
```

```
    public int getOperacija() {  
        return operacija;  
    }
```

```
    public void setOperacija(int operacija) {  
        this.operacija = operacija;  
    }
```

```

public Object getObjekatOperacije() {
    return objekatOperacije;
}

public void setObjekatOperacije(Object objekatOperacije)
    { this.objekatOperacije = objekatOperacije;
}

}

```

```

public class ServerskiOdgovor implements Serializable {

    private String poruka;

    private Object objekatOperacija;

    private boolean uspesno;

    private int operacija;

    public ServerskiOdgovorObjekat() {

    }

    public ServerskiOdgovorObjekat(String poruka, Object objekatOperacija, boolean
    uspesno) {

        this.poruka = poruka;

        this.objekatOperacija = objekatOperacija;

        this.uspesno = uspesno;

    }

    public String getPoruka() {

        return poruka;

    }

}

```

```

public void setPoruka(String poruka) {
    this.poruka = poruka;
}

public Object getObjekatOperacija() {
    return objekatOperacija;
}

public void setObjekatOperacija(Object
    objekatOperacija) { this.objekatOperacija =
    objekatOperacija;
}

public boolean isUspesno() {
    return uspesno;
}

public void setUspesno(boolean uspesno) {
    this.uspesno = uspesno;
}

public int getOperacija() {
    return operacija;
}

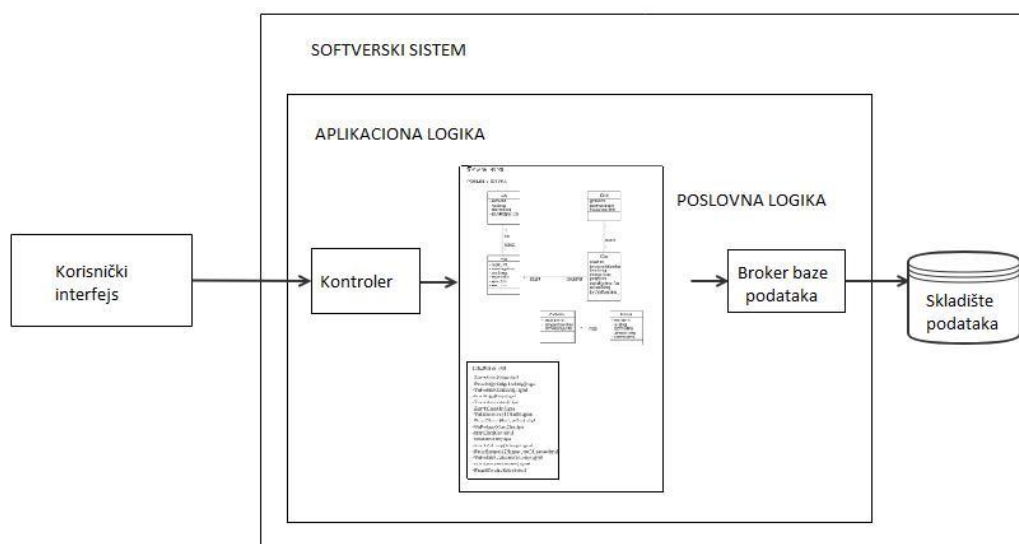
public void setOperacija(int operacija) {
    this.operacija = operacija;
}
}

```

3.3.2 Kontroler aplikacione logike

Kontroler aplikacione logike prihvata zahteve za izvršenje sistemskih operacija i iste prosleđuje do konkretne sistemske operacije. Nakon izvršenja sistemske operacije, kontroler prihvata odgovor i vraća nazad pozivaocu (niti klijenta).

Kako je u fazama prikupljanja zahteva i analiza data specifikacija strukture i ponašanja softverskog sistema, odnosno specifikacija poslovne logike softverskog sistema, sledeća slika daje opis sistema nakon faze projektovanja komunikacije sa klijentima i kontroler aplikacione logike.

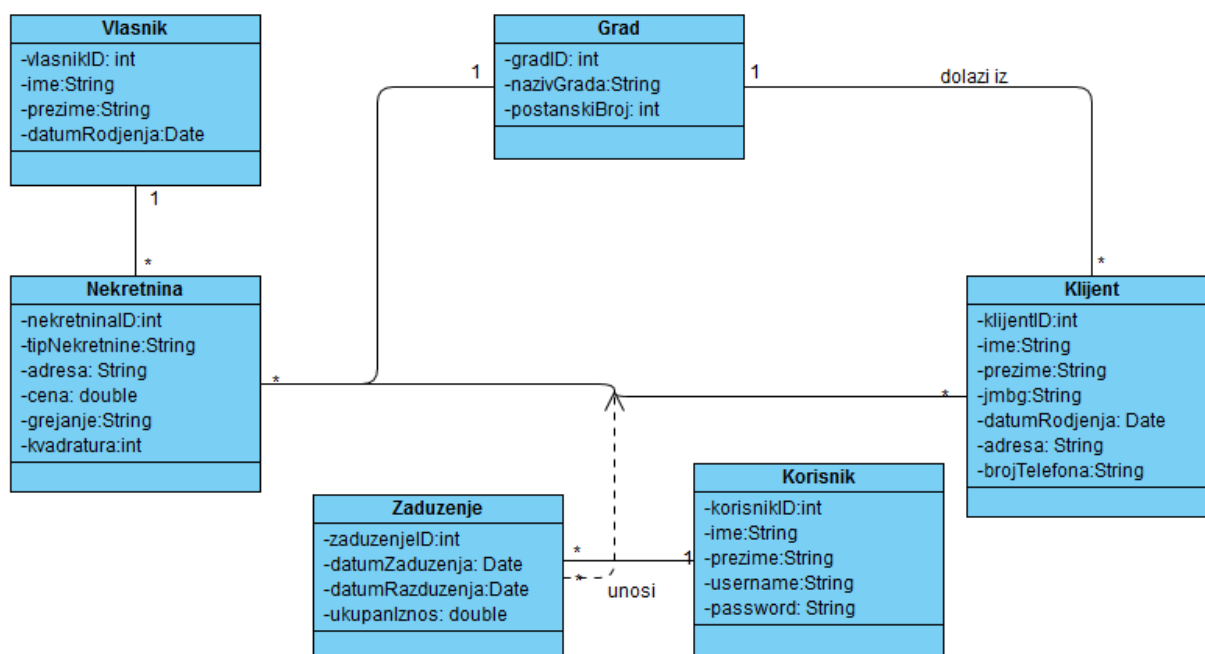


Slika 8. Prikaz arhitekture softverskog sistema

3.4 Projektovanje strukture softverskog sistema – Domenske klase

Na osnovu konceptualnih klasa prave se softverske klase strukture. Svaka klasa sadrži privatna polja atributa, modifikator pristupa za te attribute, konstruktore (neparametarski I parametarski).

Domenske klase implementiraju interfejsj OpstiDomenskiObjekat, da bi se omogućilo lakše implementiranje metode Brokera baze. Broker prima interfejs umesto samih klasa i olakšava kreiranje generičkih upita.



Slika 9. Konceptualni model

3.5 Projektovanje strukture softverskog sistema – Systemske operacije

Preporuka je da se na samom početku projektovanja sistemskih operacije naprave konceptualne realizacije (rešenja) za svaku sistemsku operaciju tako da ona bude direktno povezana sa logikom problema. Aspekti realizacije koji se odnose na konekciju sa bazom, perzistentnost i transakcije treba izbeći kako bi se logika rešavanja problema nezavisno razvijala.

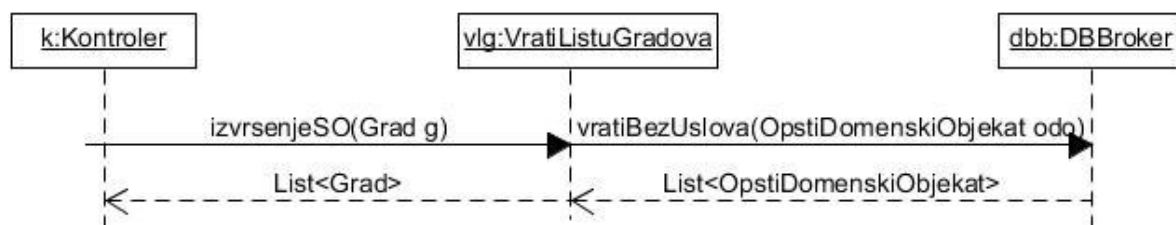
Ugovor UG1: VratiListuGradova

Operacija: signal VratiListuGradova(List<Grad>):signal;

Veza sa SK: SK1, SK5, SK6

Preduslovi: /

Postuslovi: /



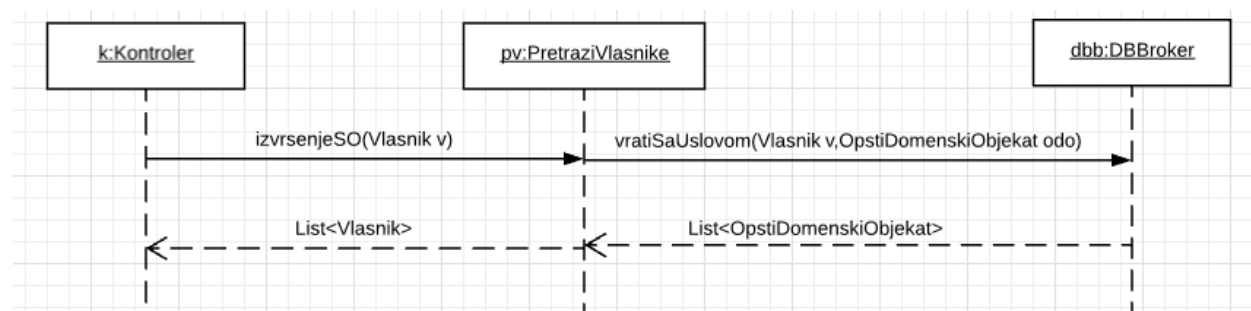
Ugovor UG2: PretraziVlasnike

Operacija: signal PretraziVlasnike(Vlasnik, List<Vlasnik>): signal;

Veza sa SK: SK1, SK11

Preduslovi: /

Postuslovi: /



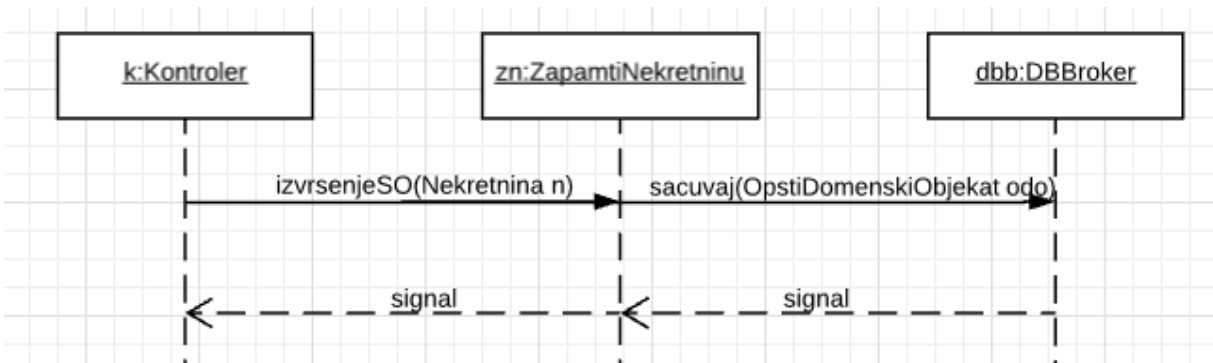
Ugovor UG3: ZapamtiNekretninu

Operacija: signal ZapamtiNekretninu(Nekretnina):signal;

Veza sa SK: SK1

Preduslovi: Vrednosna i strukturna ograničenja nad objektom nekretnina moraju biti zadovoljena.

Postuslovi: Nekretnina je zapamćena.



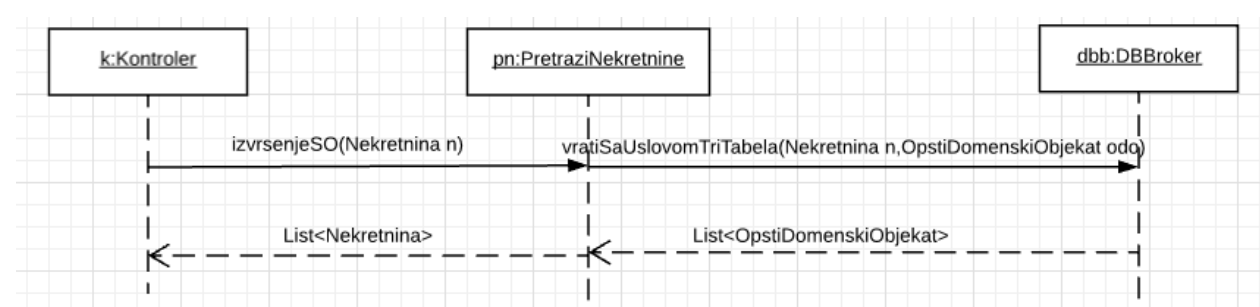
Ugovor UG4: PretraziNekretnine

Operacija: signal PretraziNekretnine(Nekretnina, List<Nekretnina>):signal;

Veza sa SK: SK2, SK3, SK8

Preduslovi: /

Postuslovi: /



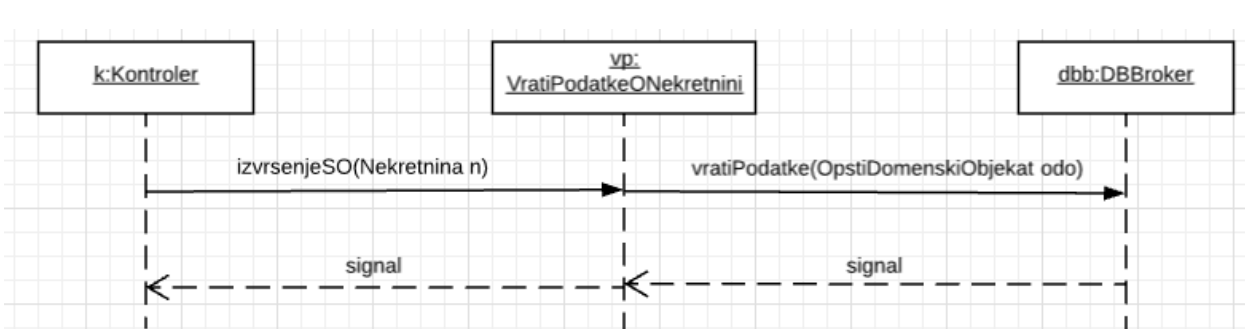
Ugovor UG5: VратиPodatkeONekretnini

Operacija: signal VратиPodatkeONekretnini(Nekretnina):signal;

Veza sa SK: SK2

Preduslovi: /

Postuslovi: /



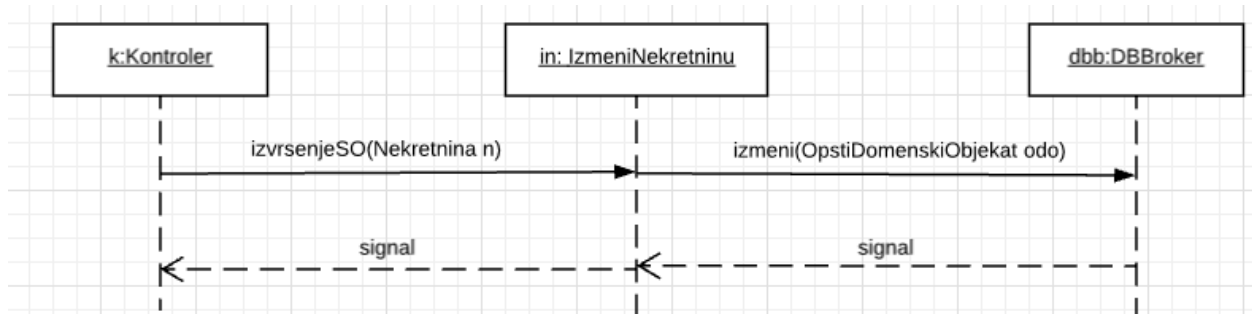
Ugovor UG6: IzmeniNekretninu

Operacija: signal IzmeniNekretninu(Nekretnina):signal;

Veza sa SK: SK2

Preduslovi: Vrednosna i strukturna ograničenja na objektom Nekretnina moraju biti zadovoljena.

Postuslovi: Nekretnina je izmenjena.



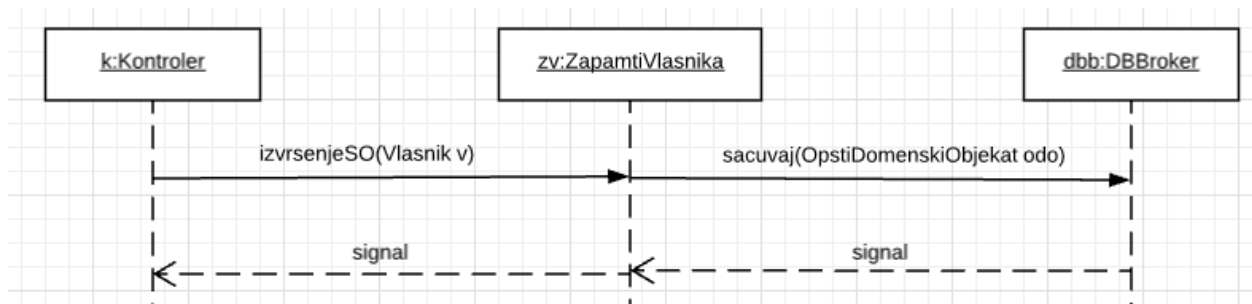
Ugovor UG7: ZapamtiVlasnika

Operacija: signal ZapamtiVlasnika(Vlasnik):signal;

Veza sa SK: SK4

Preduslovi: Vrednosna i strukturna ograničenja na objektom Vlasnik moraju biti zadovoljena.

Postuslovi: Vlasnik je zapamćen.



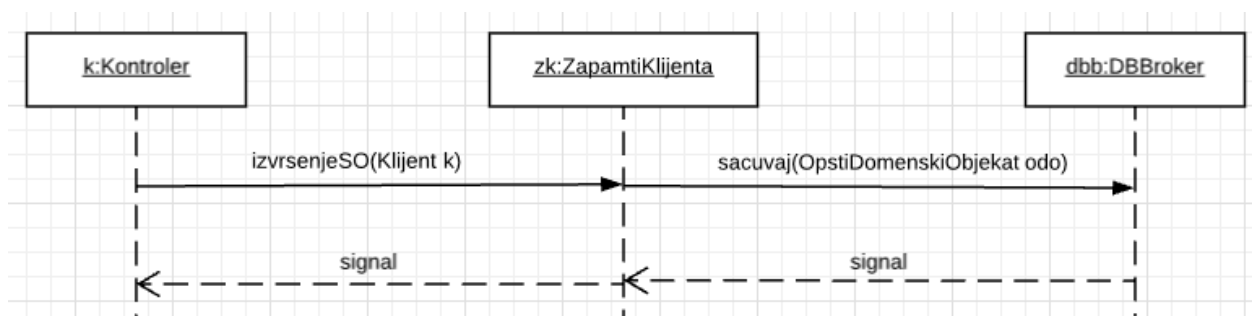
Ugovor UG8: ZapamtiKlijenta

Operacija: signal ZapamtiKlijenta(Klijent):signal;

Veza sa SK: SK5

Preduslovi: Vrednosna i strukturna ograničenja na objektom Klijent moraju biti zadovoljena.

Postuslovi: Podaci o klijentu su zapamćeni.



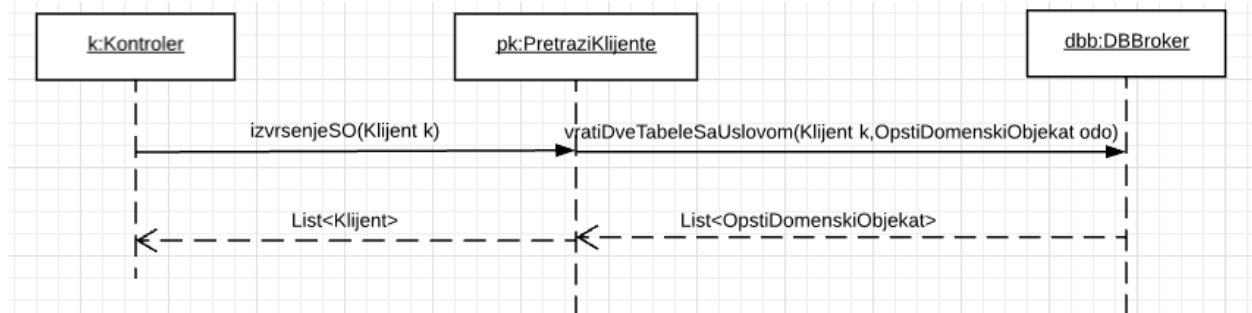
Ugovor UG9: PretraziKlijente

Operacija: signal PretraziKlijente (Klijent, List<Klijent>):signal;

Veza sa SK: SK6, SK7, SK8

Preduslovi: /

Postuslovi: /



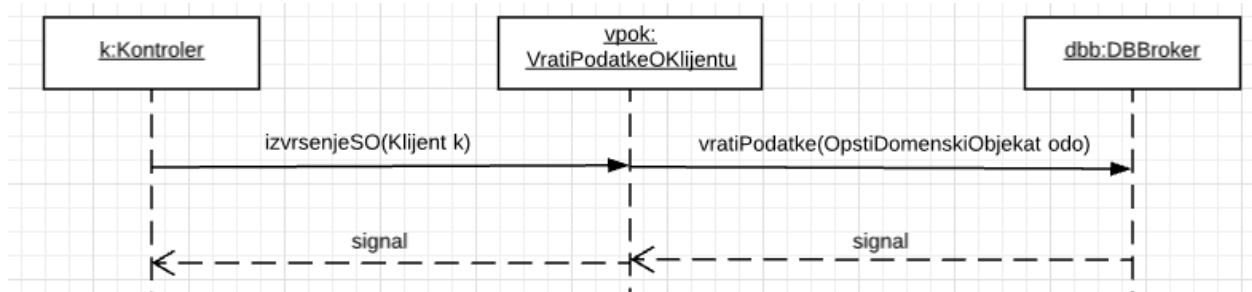
Ugovor UG10: VратиPodatkeOKlijentu(Klijent)

Operacija: signal VратиPodatkeOKlijentu(Klijent):signal;

Veza sa SK: SK6

Preduslovi: /

Postuslovi: /



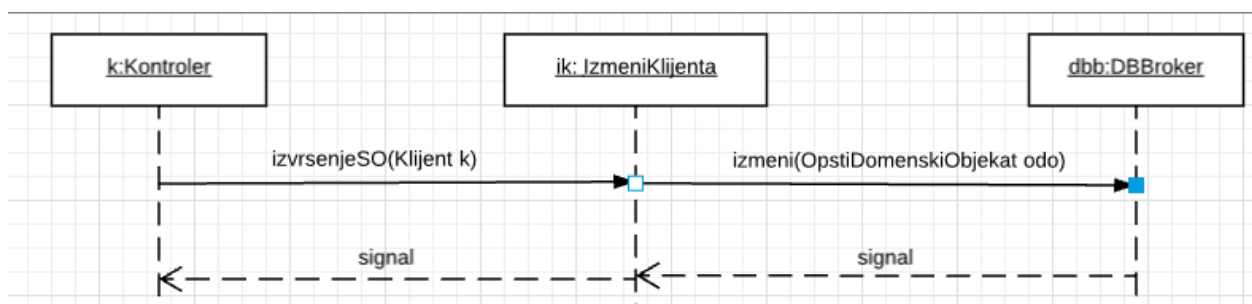
Ugovor UG11: IzmeniKlijenta

Operacija: signal IzmeniKlijenta(Klijent):signal;

Veza sa SK: SK6

Preduslovi: Vrednosna i strukturna ograničenja na objektom Klijent moraju biti zadovoljena.

Postuslovi: Klijent je izmenjen.



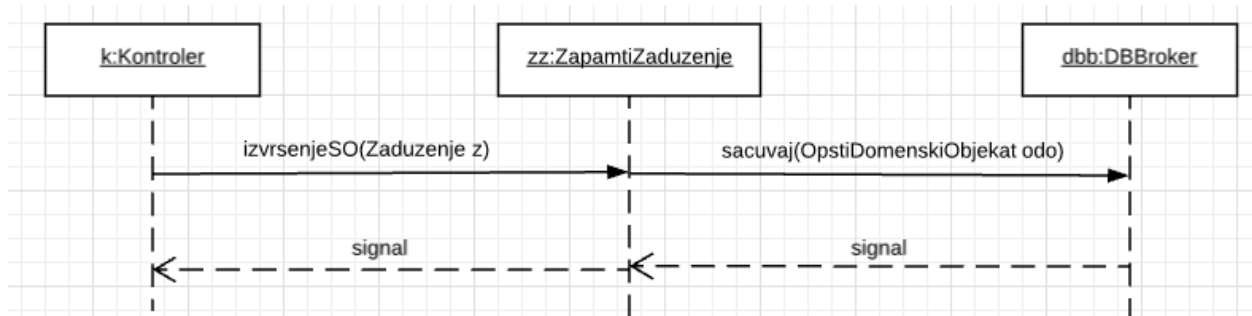
Ugovor UG12: ZapamtiZaduzenje

Operacija: signal ZapamtiZaduzenje(Zaduzenje):signal;

Veza sa SK: SK8

Preduslovi: Vrednosna i strukturna ograničenja nad objektom Zaduzenje moraju biti zadovoljena.

Postuslovi: Podaci o zaduženju su zapaćeni.



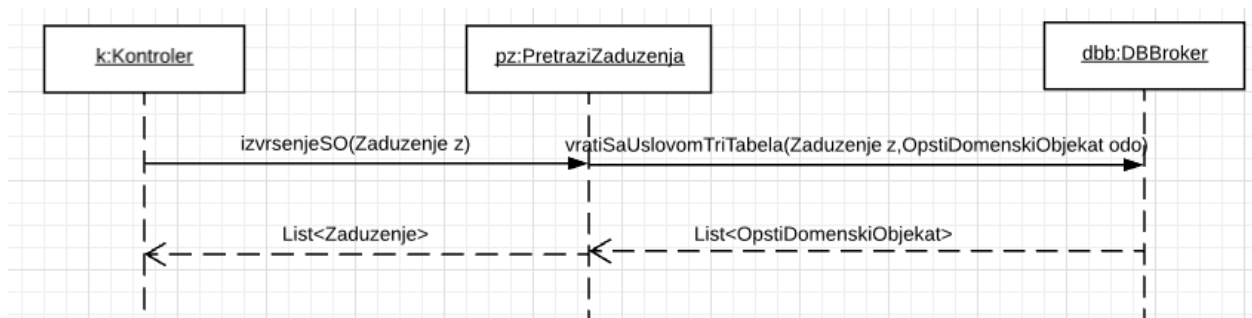
Ugovor UG13: PretraziZaduzenja

Operacija: signal PretraziZaduzenja(Zaduzenje, List<Zaduzenje>):signal;

Veza sa SK: SK9,SK10

Preduslovi: /

Postuslovi: /



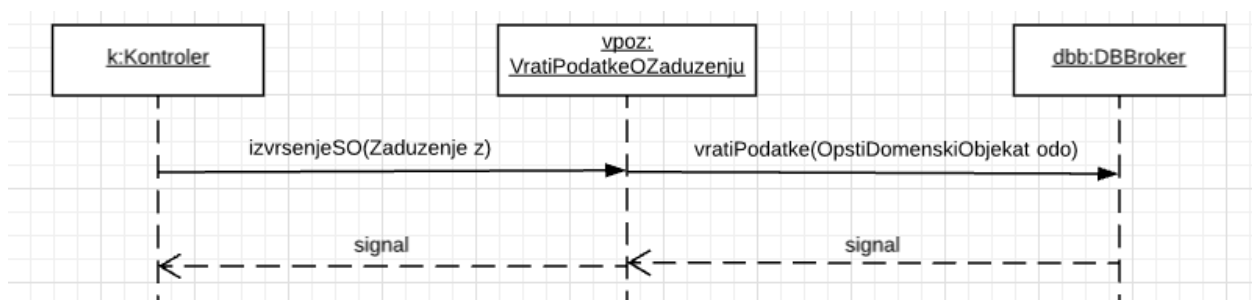
Ugovor UG14: VратиPodatkeOZaduzenju

Operacija: signal VратиPodatkeOZaduzenju(Zaduzenje):signal;

Veza sa SK: SK9

Preduslovi: /

Postuslovi: /



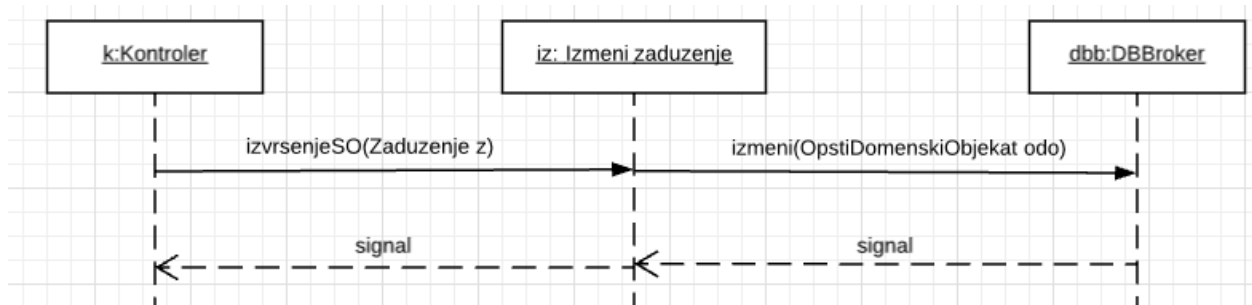
Ugovor UG15: IzmeniZaduzenje

Operacija: signal IzmeniZaduzenje(Zaduzenje):signal;

Veza sa SK: SK9

Preduslovi: Vrednosna i strukturna ograničenja na objektom Zaduzenje moraju biti zadovoljena.

Postuslovi: Zaduzenje je izmenjeno.



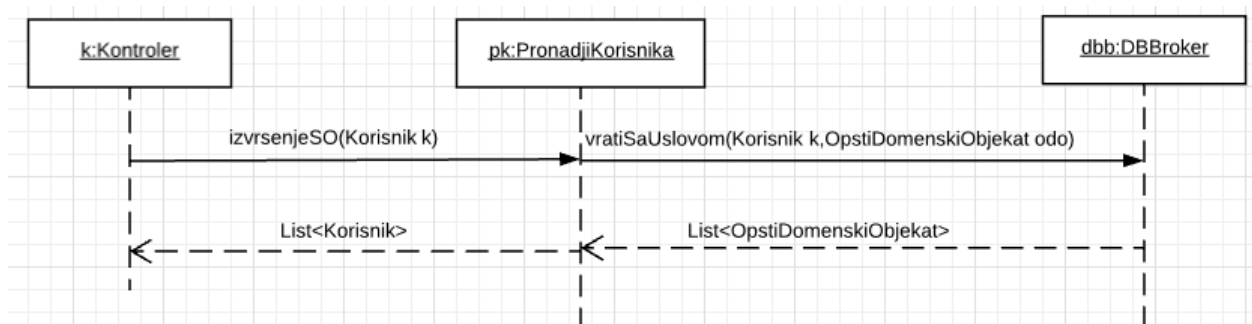
Ugovor UG18: PronadjiKorisnika

Operacija: signal PronadjiKorisnika(Korisnik):signal;

Veza sa SK: SK12

Preduslovi: /

Postuslovi: /



3.6 Projektovanje aplikacione logike – Broker baze podataka

Broker baze podataka je softverska klasa odgovorna za komunikaciju između poslovne logike i skladišta podataka.

Klasa DBBroker predstavlja perzistentni okvir koji posreduje u svim operacijama nad bazom podataka i realizuje sledeće metode:

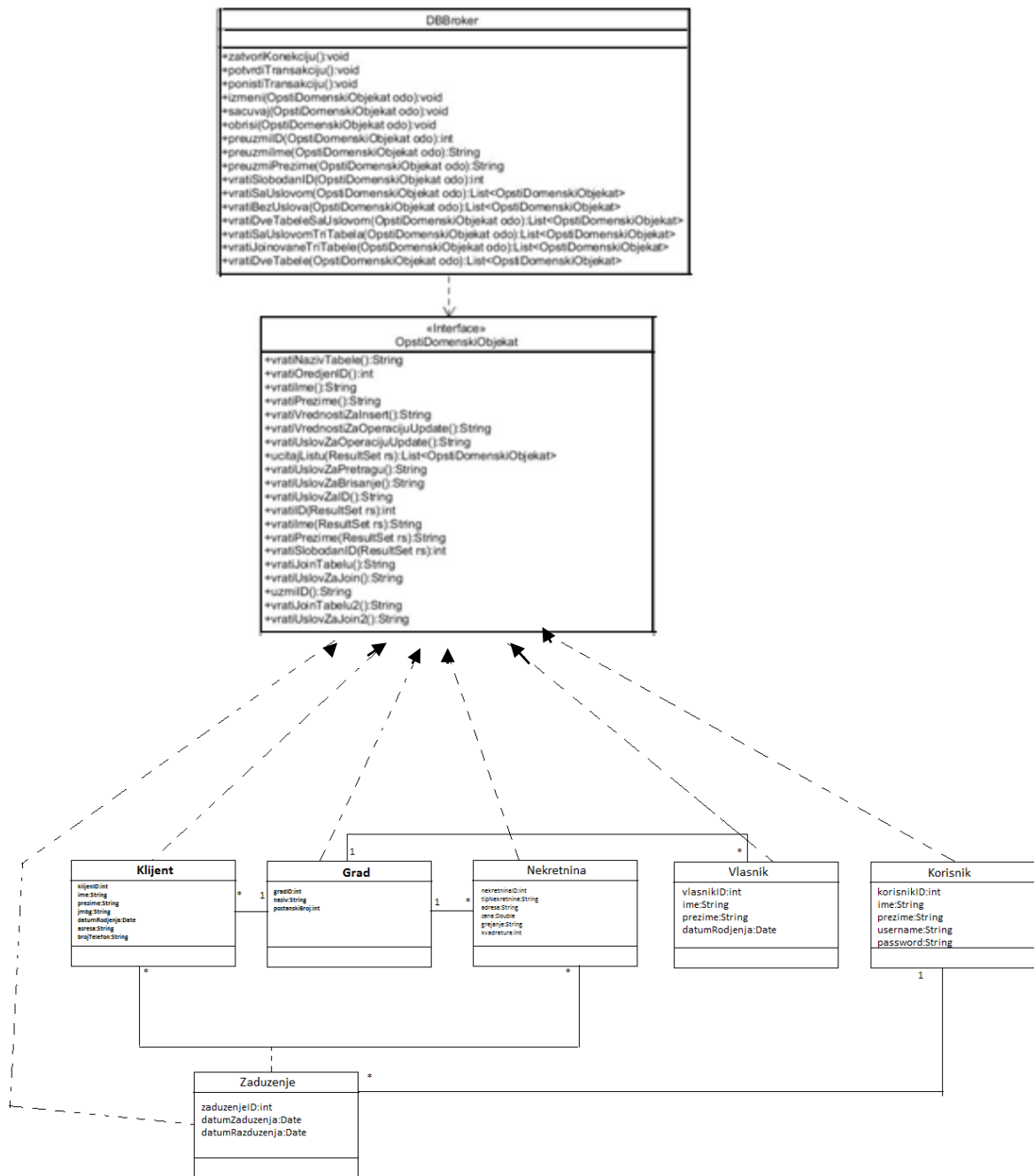
1. `public void zatvoriKonekciju();`
2. `public void potvrdiTransakciju();`
3. `public void ponistiTransakciju();`
4. `public void izmeni(OpstiDomenskiObjekat odo);`
5. `public void sacuvaj(OpstiDomenskiObjekat odo);`
6. `public int preuzmiID(OpstiDomenskiObjekat odo);`
7. `public String preuzmiIme(OpstiDomenskiObjekat odo);`
8. `public String preuzmiPrezime(OpstiDomenskiObjekat odo);`
9. `public int vratiSlobodanID(OpstiDomenskiObjekat odo);`
10. `public List<OpstiDomenskiObjekat> vratiSaUslovom(OpstiDomenskiObjekat odo);`
11. `public List<OpstiDomenskiObjekat> vratiBezUslova(OpstiDomenskiObjekat odo);`
12. `public List<OpstiDomenskiObjekat>`
 `vratiDveTabeleSaUslovom(OpstiDomenskiObjekat odo);`
13. `public List<OpstiDomenskiObjekat>`
 `vratiSaUslovomTriTabele(OpstiDomenskiObjekat odo);`
14. `public List<OpstiDomenskiObjekat>`
 `vratiJoinovaneTriTabele(OpstiDomenskiObjekat odo);`
15. `public List<OpstiDomenskiObjekat> vratiDveTabele(OpstiDomenskiObjekat odo);`

Sve metode rada su projektovane kao generičke, što znači da mogu da prihvate različite objekte i različite upite preko ulaznih parametara kako se ne bi za svaku sistemsku operaciju kreirala metoda u klasi DBBroker.

Rezultat je manja kompleksnost samog DBBroker-a. Da bi se to omogućilo korišćen je interfejs OpstiDomenskiObjekat:

```
public interface OpstiDomenskiObjekat extends Serializable {  
  
    public String vratiNazivTabele();  
    public int vratiOredjenID();  
    public String vratiIme();  
    public String vratiPrezime();  
    public String vratiVrednostiZaInsert();  
    public String vratiVrednostiZaOperacijuUpdate();  
    public String vratiUslovZaOperacijuUpdate();  
    public List<OpstiDomenskiObjekat> ucitajListu(ResultSet rs) throws SQLException;  
    public String vratiUslovZaPretragu();  
    public String vratiUslovZaBrisanje();  
    public String vratiUslovZaID();  
    public int vratiID(ResultSet rs);  
    public String vratiIme(ResultSet rs);  
    public String vratiPrezime(ResultSet rs);  
    public int vratiSlobodanID(ResultSet rs);  
    public String vratiJoinTabelu();  
    public String vratiUslovZaJoin();  
    public String uzmiID();  
    public String vratiJoinTabelu2();  
    public String vratiUslovZaJoin2();  
}
```

Svaka klasa iz domena implementira dati interfejs, i sve njegove metode. Na taj način je moguće da metode DBBroker-a budu generičke.



Slika 10. Klasa DBBroker se povezuje sa klasom OpstiDomenskiObjeka

3.7 Projektovanje skladišta podataka

Na osnovu strukture softverskih klasa projektovane su tabele (skladišta podataka) relacionog sistema za upravljanje bazom podataka. U ovom radu korišćen je SQLyog Community 64.

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update	Comment
<input type="checkbox"/>	vlasnikID	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	ime	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	prezime	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	datumRodjenja	date			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Slika 11. Struktura tabele Vlasnik u SQLyog-u

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update	Comment
<input type="checkbox"/>	nekretninaID	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	tipNekretnine	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	adresa	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	kvadratura	int	11		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	grejanje	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	cena	double			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	gradID	int	11		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	vlasnikID	int	11		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Slika 12. Struktura tabele Nekretnina u SQLyog-u

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update	Comment
<input type="checkbox"/>	gradID	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	nazivGrada	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	postanskiBroj	int	11		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Slika 13. Struktura tabele Grad u SQLyog-u

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update	Comment
<input type="checkbox"/>	klijentID	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	ime	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	prezime	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	jmbg	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	datumRodjenja	date			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	adresa	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	brojTelefona	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	gradID	int	11		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

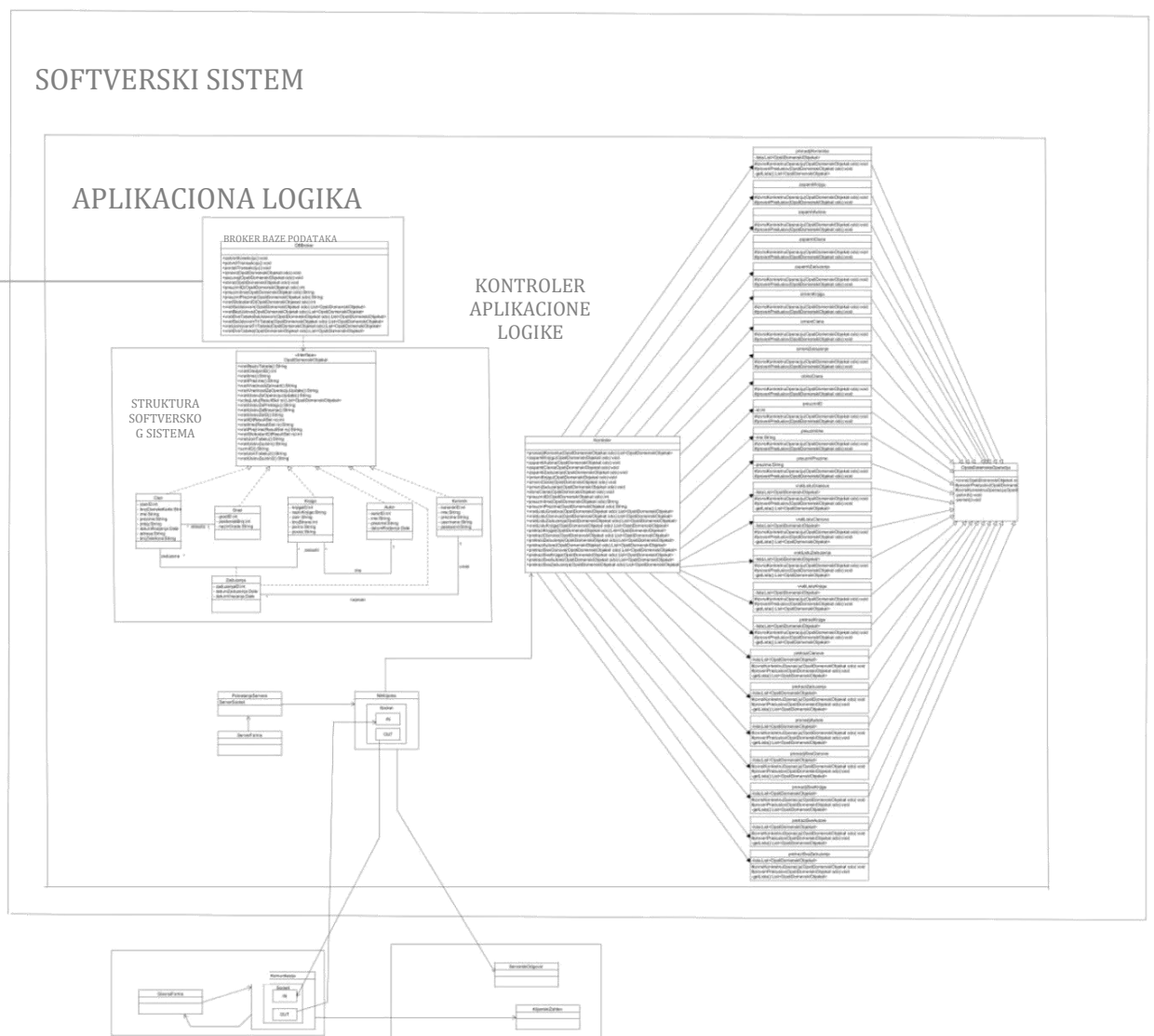
Slika 14. Struktura tabele Klijent u SQLyog-u

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update	Comment
<input type="checkbox"/>	zaduzenjeID	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	datumZaduzenja	date			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	datumRazduzenja	date			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	ukupanIznos	double			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	nekretninaID	int	11		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	klijentID	int	11		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	korisnikID	int	11		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Slika 15. Struktura tabele Zaduzenje u SQLyog-u

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update	Comment
<input type="checkbox"/>	korisnikID	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	ime	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	prezime	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	username	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	password	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	ulogovan	tinyint	4		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Slika 16. Struktura tabele Korisnik u SQLyog-u



Slika 17. Arhitektura softverskog sistema

4. Implementacija

Softverski sistem je razvijen u programskom jeziku Java. Sistem je projektovan kao klijent-server. Kao sistem za upravljanjem bazom podataka korišćen je SQLyog Community 64, dok je razvojno okruženje NetBeans IDE 8.2.

Sistem je realizovan u tri projekta: KlijentaApp Server i ZajednickiProjekat. Na osnovu arhitekture softverskog sistema dobijene su sledeće softverske klase:

1. Klijent:

- forme/GlavnaForma
- forme/Login
- forme/PretragaKlijentaForma
- forme/ PretragaNekretnineForma
- forme/ PretragaVlasnikaForma
- forme/ PretragaZaduzenjaForma
- forme/UnosKlijentaForma
- forme/ UnosNekretnineForma
- forme/ UnosVlasnikaForma
- forme/ UnosZaduzenjaForma
- kom/KomunikacijaSaServerom
- kontroler/Kontroler
- model/ModelTabelePretragaKlijenta
- model/ ModelTabelePretragaNekretnine
- model/ ModelTabelePretragaVlasnika
- model/ ModelTabelePretragaZaduzenja

2. Server

- db/DBBroker
- db/DBKonstante
- db/DBProperties
- forme/ServerskaForma
- kontroler/Kontroler
- server/NitKlijenta
- server/PokretanjeServera
- so/OpstaSistemskaOperacija
- so.vlasnik/SOPretraziVlasnike
- so.vlasnik/SOPretraziSveVlasnike
- so. vlasnik /SOZapamtiVlasnika
- so.klijent/SOIzmeniKlijenta

- so. klijent /SOPretraziKlijente
- so. klijent /SOPretraziSveKlijente
- so. klijent /SOZapamtiKlijenta
- so.grad/SOVratiListuGradova
- so.nekretnina/SOIzmeniNekretninu
- so. nekretnina /SOPretraziNekretnine
- so. nekretnina /SOPretraziSveNekretnine
- so. nekretnina /SOZapamtiNekretninu
- so.korisnik/SOIzmeniKorisnika
- so.korisnik/SOPreuzmiIDKorisnika
- so.korisnik/SOPreuzmiIme
- so.korisnik/SOPreuzmiPrezimeKorisnika
- so.korisnik/SOPronadjiKorisnika
- so.zaduzenje/SOIzmeniZaduzenje
- so.zaduzenje/SOPretraziZaduzenje
- so.zaduzenje/SOPretraziSvaZaduzenja
- so.zaduzenje/SOVratiListuZaduzenja
- so.zaduzenje/SOZapamtiZaduzenje

3. ZajedničkiProjekat

domen/Grad

domen/Klijent

domen/Korisnik

domen/Nekretnina

domen/OpstiDomenskiObjekat

domen/Zaduzenje

domen/Vlasnik

operacije/Operacije

transfer/KlijentskiZahtev

transfer/ServerskiOdgovor

5.Testiranje

Svaki od implementiranih slučajeva korišćenja je testiran. Prilikom testiranja svakog od njih, pored pravilno unetih podataka, unošeni su i nepravilni podaci da bi se uvideo rezultat izvršenja slučaja korišćenja. Na osnovu obavljenih testiranja uklonjeni su uočeni nedostaci.

6.Literatura

dr Siniša Vlajić, Projektovanje softvera

