**UNIVERZITET U BEOGRADU**

**FAKULTET ORGANIZACIONIH NAUKA**

**SEMINARSKI RAD**

Predmet: Projektovanje softvera

Tema: Softverski sistem za praćenje rada salona za kućne ljubimce

Profesor: Student:

Dr Siniša Vlajić Aleksandra Marjanović 1024/2018

Beograd, 2019.

**Sadržaj**

[**1.** **Korisnički zahtevi** 3](#_Toc23848309)

[**1.1.** **Verbalni opis** 3](#_Toc23848310)

[**1.2. Specifikacija zahteva pomoću modela slučajeva korišćenja** 3](#_Toc23848311)

[**1.2.1. Slučaj korišćenja – Prijavljivanje zaposlenog** 4](#_Toc23848312)

[**1.2.2. Slučaj korišćenja – Unos novog korisnika** 5](#_Toc23848313)

[1.2.3. **Slučaj korišćenja – Pretraga korisnika** 5](#_Toc23848314)

[**1.2.4. Slučaj korišćenja – Izmena postojećeg korisnika** 6](#_Toc23848315)

[**1.2.5. Slučaj korišćenja – Brisanje korisnika** 7](#_Toc23848316)

[**1.2.6. Slučaj korišćenja – Unos nove usluge** 8](#_Toc23848317)

[**1.2.7. Slučaj korišćenja – Pretraga usluge** 9](#_Toc23848318)

[**1.2.8. Slučaj korišćenja – Brisanje usluge** 10](#_Toc23848319)

[**1.2.9. Slučaj korišćenja – Zakazivanje termina (Složen SK)** 10](#_Toc23848320)

[**1.2.10. Slučaj korišćenja – Pretraživanje termina** 11](#_Toc23848321)

[**2.** **Analiza** 12](#_Toc23848322)

[**2.1.** **Ponašanje softverskog sistema – sistemski dijagrami sekvenci** 12](#_Toc23848323)

[**2.1.1.** **DS1: Dijagram sekvenci slučaja korišćenja - Prijavljivanje zaposlenog** 12](#_Toc23848324)

[**2.1.2.** **DS2: Dijagram sekvenci slučaja korišćenja - Unos novog korisnika** 13](#_Toc23848325)

[**2.1.3.** **DS3: Dijagram sekvenci slučaja korišćenja – Pretraga korisnika** 15](#_Toc23848326)

[**2.1.4.** **DS4: Dijagram sekvenci slučaja korišćenja – Izmena postojećeg korisnika** 17](#_Toc23848327)

[**2.1.5.** **DS5: Dijagram sekvenci slučaja korišćenja – Brisanje korisnika** 20](#_Toc23848328)

[**2.1.6.** **DS6: Dijagram sekvenci slučaja korišćenja – Unos nove usluge** 22](#_Toc23848329)

[**2.1.7.** **DS7: Dijagram sekvenci slučaja korišćenja – Pretraga usluge** 24](#_Toc23848330)

[**2.1.8.** **DS8: Dijagram sekvenci slučaja korišćenja – Brisanje usluge** 26](#_Toc23848331)

[**2.1.9.** **DS9: Dijagram sekvenci slučaja korišćenja – Zakazivanje termina** 28](#_Toc23848332)

[**2.1.10. DS10: Dijagram sekvenci slučaja korišćenja – Pretraživanje termina** 30](#_Toc23848333)

[**2.2.** **Ponašanje softverskog sistema – Definisanje ugovora o sistemskim operacijama** 32](#_Toc23848334)

[**2.3.** **Struktura softverskog sistema – Konceptualni (domenski) model** 35](#_Toc23848335)

[**2.4.** **Struktura softverskog sistema – Relacioni model** 36](#_Toc23848336)

[**3.** **Projektovanje** 38](#_Toc23848337)

[**4.** **Implementacija** 38](#_Toc23848338)

[**5.** **Testiranje** 38](#_Toc23848339)

[**6.** **Literatura** 38](#_Toc23848340)

# **Korisnički zahtevi**

## **Verbalni opis**

Potrebno je kreirati softverski sistem koji će biti korišćen za vođenje pružanja usluga u salonu za kućne ljubimce.

Softverski sistem mora da obezbedi vođenje evidencije o korisnicima, pretragu korisnika po određenim vrednostima, unos novih korisnika, ažuriranje podataka o postojećim korisnicima i brisanje istih. Zaposleni treba imati uvid u spisak svih korisnika usluga salona za kućne ljubimce.

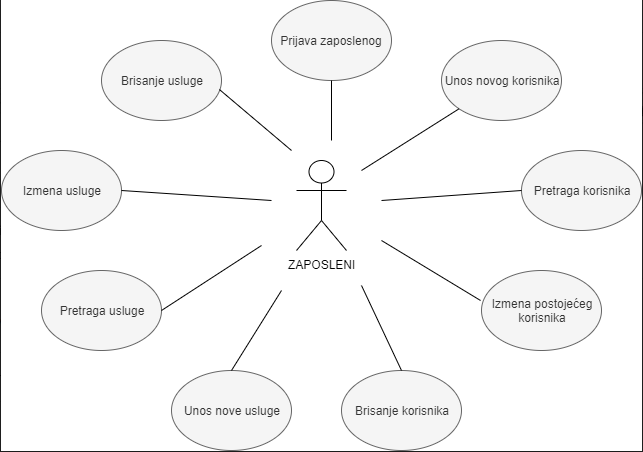
Potrebno je voditi evidenciju o uslugama koji postoje u salonu i omogućiti jednostavnu i brzu pretragu tih usluga po zadatim vrednostima. Korisnik usluga može u bilo kom trenutku da promeni vrstu usluge.

Korisnik sistema je zaposleni u salonu za kućne ljubimce i on se mora registrovati kako bi vršio unos i izmenu podataka.

## **1.2. Specifikacija zahteva pomoću modela slučajeva korišćenja**

Identifikovani su sledeći slučajevi korišćenja:

1. Prijavljivanje zaposlenog
2. Unos novog korisnika
3. Pretraga korisnika
4. Izmena postojećeg korisnika
5. Brisanje korisnika
6. Unos nove usluge
7. Pretraga usluge
8. Brisanje usluge
9. Zakazivanje termina (Složen SK)
10. Pretraživanje termina



Slika 1 - Dijagram slučajeva korišćenja

### **1.2.1. Slučaj korišćenja – Prijavljivanje zaposlenog**

**Naziv SK**

Prijavljivanje zaposlenog

**Aktori SK**

Zaposleni

**Učesnici SK**

Zaposleni i sistem (program)

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

**Osnovni scenario SK**

1. Zaposleni **unosi** podatke za prijavljivanje zaposlenog. (APUSO)

2. Zaposleni **kontroliše** da li je ispravno uneo podatke za registrovanje. (ANSO)

3. Zaposleni **poziva** sistem da ga pronađe na osnovu unetih podataka. (APSO)

4. Sistem **pretražuje** zaposlene. (SO)

5. Sistem **prikazuje** zaposlenom poruku na ekranu: „Zaposleni je uspešno registrovan u sistem“ i omogućava pristup sistemu. (IA)

**Alternativna scenarija**

5.1. Ukoliko sistem ne može da pronađe zaposlenog na osnovu unetih podataka, prikazuje poruku „Sistem nije uspeo da pronađe zaposlenog na osnovu unetih podataka“. (IA)

### **1.2.2. Slučaj korišćenja – Unos novog korisnika**

**Naziv SK**

Unos novog korisnika

**Aktori SK**

Zaposleni

**Učesnici SK**

Zaposleni i sistem (program)

**Preduslov:** Sistem je uključen i zaposleni je prijavljen. Sistem prikazuje formu za unos novog korisnika.

**Osnovni scenario SK**

1. Zaposleni **poziva** sistem da kreira novog korisnika. (APSO)
2. Sistem **kreira** korisnika. (SO)
3. Sistem **prikazuje** zaposlenom poruku: „Novi korisnik je uspešno kreiran.“ (IA)
4. Zaposleni **unosi** podatke za kreiranog korisnika. (ANSO)
5. Zaposleni **poziva** sistem da sačuva podatke o novom korisniku. (APSO)
6. Sistem **pamti** podatke o korisniku. (SO)
7. Sistem **prikazuje** zaposlenom sačuvane podatke o novom korisniku i ispisuje poruku: „Sistem je uspešno zapamtio podatke o korisniku.“ (IA)

**Alternativna scenarija**

3.1. Ukoliko sistem nije uspeo da kreira novog korisnika, on prikazuje zaposlenom poruku: „Sistem ne može da kreira novog korisnika.“ Prekida se izvršenje scenarija. (IA)

7.1. Ukoliko sistem nije uspeo da zapamti podatke o korisniku, prikazuje zaposlenom sledeću poruku: „Sistem nije uspeo da sačuva korisnika“. (IA)

### 1.2.3. **Slučaj korišćenja – Pretraga korisnika**

**Naziv SK**

Pretraga korisnika

**Aktori SK**

Zaposleni

**Učesnici SK**

Zaposleni i sistem (program)

**Preduslov:** Sistem je uključen i zaposleni je prijavljen. Sistem prikazuje formu za rad sa korisnikom.

**Osnovni scenario SK**

1. Zaposleni **unosi** kriterijum u formu za pretraživanje korisnika. (APUSO)
2. Zaposleni **poziva** sistem da pronađe korisnika na osnovu zadatog kriterijuma. (APSO)
3. Sistem **traži** korisnika. (SO)
4. Sistem **prikazuje** pronađene korisnike. (IA)
5. Zaposleni **bira** korisnika čije podatke želi da vidi. (APUSO)
6. Zaposleni **poziva** sistem da prikaže podatke o izabranom korisniku. (APSO)
7. Sistem **pronalazi** podatke o izabranom korisniku. (SO)
8. Sistem **prikazuje** podatke o izabranom korisniku i poruku: „Sistem je prikazao podatke o korisniku“. (IA)

**Alternativna scenarija**

4.1. Ukoliko sistem nije uspeo da pronađe korisnike, prikazuje se poruka zaposlenom: „Sistem nije uspeo da pronađe korisnike na osnovu zadatog kriterijuma.“ Prekida se izvršenje scenarija. (IA)

8.1. Ukoliko sistem nije uspeo da pronađe podatke o izabranom korisniku, prikazuje se odgovarajuća poruka zaposlenom: „Ne postoje podaci o izabranom korisniku“. (IA)

### **1.2.4. Slučaj korišćenja – Izmena postojećeg korisnika**

**Naziv SK**

Izmena postojećeg korisnika

**Aktori SK**

Zaposleni

**Učesnici SK**

Zaposleni i sistem (program)

**Preduslov:** Sistem je uključen i zaposleni je prijavljen. Učitana je lista korisnika. Sistem prikazuje formu za rad sa korisnikom.

**Osnovni scenario SK**

1. Zaposleni **unosi** kriterijum po kom pretražuje korisnike. (APUSO)
2. Zaposleni **poziva** sistem da pronađe korisnike na osnovu zadatog kriterijuma. (APSO)
3. Sistem **traži** korisnika. (SO)
4. Sistem **prikazuje** zaposlenom pronađene korisnike i poruku: “Sistem je našao korisnike prema zadatom kriterijumu“. (IA)
5. Zaposleni **bira** korisnika kog želi da izmeni. (APUSO)
6. Zaposleni **poziva** sistem da učita podatke o izabranom korisniku. (APSO)
7. Sistem **pronalazi** podatke o izabranom korisniku. (SO)
8. Sistem **prikazuje** zaposlenom podatke o izabranom korisniku i poruku: „Sistem je pronašao podatke o korisniku.“ (IA)
9. Zaposleni **unosi** nove vrednosti za odabranog korisnika. (APUSO)
10. Zaposleni **poziva** sistem da izmeni izabranog korisnika. (APSO)
11. Sistem **pamti** izmenjene podatke o korisniku. (SO)
12. Sistem **prikazuje** izmenjene podatke o korisniku i poruku: „Sistem je uspešno izmenio podatke o korisniku“. (IA)

**Alternativna scenarija**

4.1. Ukoliko sistem nije uspeo da pronađe korisnike koji odgovaraju zadatom kriterijumu, zaposlenom se ispisuje odgovarajuća poruka: „Sistem nije uspeo da pronađe korisnike na osnovu zadatog kriterijuma!“. Prekida se izvršenje scenarija. (IA)

8.1. Ukoliko sistem nije uspeo da pronađe podatke o odabranom korisniku, zaposlenom se ispisuje poruka: „Sistem nije uspeo da pronađe podatke o korisniku.“. Prekida se izvršenje scenarija. (IA)

12.1. Ukoliko sistem nije uspeo da sačuva izmenjene podatke o korisniku, zaposlenom se ispisuje poruka: „Sistem ne može da zapamti korisnika!“. (IA)

### **1.2.5. Slučaj korišćenja – Brisanje korisnika**

**Naziv SK**

Brisanje korisnika

**Aktori SK**

Zaposleni

**Učesnici SK**

Zaposleni i sistem (program)

**Preduslov:** Sistem je uključen i zaposleni je prijavljen. Sistem prikazuje formu za rad sa korisnikom.

**Osnovni scenario SK**

1. Zaposleni **unosi** kriterijum po kom pretražuje korisnike. (APUSO)
2. Zaposleni **poziva** sistem da pronađe korisnike na osnovu zadatog kriterijuma. (APSO)
3. Sistem **traži** korisnika. (SO)
4. Sistem **prikazuje** zaposlenom pronađene korisnike. (IA)
5. Zaposleni **bira** korisnika kog želi da obriše. (APUSO)
6. Zaposleni **poziva** sistem da obriše izabranog korisnika. (APSO)
7. Sistem **briše** odabranog korisnika. (SO)
8. Sistem **prikazuje** zaposlenom odgovarajuću poruku: „Korisnik je uspešno obrisan“. (IA)

**Alternativna scenarija**

4.1. Ukoliko sistem nije uspeo da pronađe korisnike koji odgovaraju zadatom kriterijumu, zaposlenom se ispisuje odgovarajuća poruka: „Sistem nije uspeo da pronađe korisnike na osnovu kriterijuma!“. Prekida se izvršenje scenarija. (IA)

8.1. Ukoliko sistem nije uspeo da obriše odabranog korisnika, ispisuje se poruka zaposlenom: „Sistem nije uspeo da obriše korisnika.“. (IA)

### **1.2.6. Slučaj korišćenja – Unos nove usluge**

**Naziv SK**

Unos nove usluge

**Aktori SK**

Zaposleni

**Učesnici SK**

Zaposleni i sistem (program)

**Preduslov:** Sistem je uključen i zaposleni je prijavljen. Sistem prikazuje formu za unos nove usluge.

**Osnovni scenario SK**

1. Zaposleni **poziva** sistem da kreira novu uslugu. (APSO)
2. Sistem **kreira** novu uslugu. (SO)
3. Sistem zaposlenom **prikazuje** poruku: „Nova usluga je uspešno kreirana.“. (IA)
4. Zaposleni **unosi** podatke o usluzi. (APUSO)
5. Zaposleni **poziva** sistem da sačuva podatke o novoj usluzi. (APSO)
6. Sistem **pamti** podatke o usluzi. (SO)
7. Sistem **prikazuje** zaposlenom sačuvane podatke o novoj usluzi i ispisuje poruku: „Sistem je uspešno zapamtio podatke o usluzi!“ (IA)

**Alternativna scenarija**

3.1. Ukoliko sistem nije uspeo da kreira novu uslugu, zaposlenom se prikazuje sledeća poruka: „Sistem ne može da kreira novu uslugu.“ Prekida se izvršenje scenarija. (IA)

7.1. Ukoliko sistem nije uspeo da zapamti podatke o usluzi, prikazuje zaposlenom sledeću poruku: „Sistem nije uspeo da sačuva uslugu.“ (IA)

### **1.2.7. Slučaj korišćenja – Pretraga usluge**

**Naziv SK**

Pretraga usluge

**Aktori SK**

Zaposleni

**Učesnici SK**

Zaposleni i sistem (program)

**Preduslov:** Sistem je uključen i zaposleni je prijavljen. Sistem prikazuje formu za rad sa uslugama.

**Osnovni scenario SK**

1. Zaposleni **unosi** kriterijum po kom pretražuje usluge. (APUSO)
2. Zaposleni **poziva** sistem da pronađe usluge na osnovu kriterijuma. (APSO)
3. Sistem **traži** uslugu. (S0)
4. Sistem **prikazuje** zaposlenom pronađene usluge. (IA)
5. Zaposleni **bira** uslugu koju želi da pogleda. (APUSO)
6. Zaposleni **poziva** sistem da prikaže podatke o izabranoj usluzi. (APSO)
7. Sistem **traži** podatke o izabranoj usluzi. (SO)
8. Sistem **prikazuje** podatke o izabranoj usluzi i poruku: „Sistem je prikazao podatke o izabranoj usluzi“. (IA)

**Alternativna scenarija**

4.1. Ukoliko sistem nije uspeo da pronađe usluge koje odgovaraju kriterijumu, zaposlenom se ispisuje odgovarajuća poruka: „Sistem nije uspeo da pronađe usluge na osnovu kriterijuma!“. Prekida se izvršenje scenarija. (IA)

8.1. Ukoliko sistem nije uspeo da pronađe podatke o izabranoj usluzi, zaposlenom se ispisuje poruka: „Sistem nije uspeo da pronađe podatke o usluzi.“ (IA)

### **1.2.8. Slučaj korišćenja – Brisanje usluge**

**Naziv SK**

Brisanje usluge

**Aktori SK**

Zaposleni

**Učesnici SK**

Zaposleni i sistem (program)

**Preduslov:** Sistem je uključen i zaposleni je prijavljen. Sistem prikazuje formu za rad sa uslugom.

**Osnovni scenario SK**

1. Zaposleni **unosi** kriterijum po kom pretražuje usluge. (APUSO)
2. Zaposleni **poziva** sistem da pronađe usluge na osnovu kriterijuma. (APSO)
3. Sistem **traži** uslugu. (SO)
4. Sistem **prikazuje** zaposlenom pronađene usluge. (IA)
5. Zaposleni **bira** uslugu koju želi da obriše. (APUSO)
6. Zaposleni **poziva** sistem da obriše odabranu uslugu. (APSO)
7. Sistem **briše** odabranu uslugu. (SO)
8. Sistem **prikazuje** zaposlenom odgovarajuću poruku: „Usluga je uspešno obrisana.“. (IA)

**Alternativna scenarija**

4.1. Ukoliko sistem nije uspeo da pronađe uslugu sa odgovarajućim kriterijumom, prikazuje se poruka zaposlenom: „Sistem nije uspeo da pronađe usluge koje odgovaraju zadatom kriterijumu.“. Prekida se izvršenje scenarija. (IA)

8.1. Ukoliko sistem nije uspeo da obriše odabranu uslugu, sistem prikazuje poruku zaposlenom: „Greška! Usluga nije obrisana.“. (IA)

### **1.2.9. Slučaj korišćenja – Zakazivanje termina (Složen SK)**

**Naziv SK**

Zakazivanje termina

**Aktori SK**

Zaposleni

**Učesnici SK**

Zaposleni i sistem (program)

**Preduslov:** Sistem je uključen i zaposleni je prijavljen. Učitan je spisak korisnika i usluga. Sistem prikazuje formu za rad sa terminima.

**Osnovni scenario SK**

1. Zaposleni **unosi** podatke o terminu. (APUSO)
2. Zaposleni **kontroliše** da li je korektno uneo podatke o terminu. (ANSO)
3. Zaposleni **poziva** sistem da zapamti podatke o terminu. (APSO)
4. Sistem **pamti** podatke o terminu. (SO)
5. Sistem **prikazuje** zaposlenom zapamćeni termin i poruku: “Sistem je zapamtio termin”. (IA)

**Alternativna scenarija**

5.1 Ukoliko sistem ne može da zapamti podatke o terminu on prikazuje

zaposlenom poruku “Sistem ne može da zapamti termin”. (IA)

### **1.2.10. Slučaj korišćenja – Pretraživanje termina**

**Naziv SK**

Pretraživanje termina

**Aktori SK**

Zaposleni

**Učesnici SK**

Zaposleni i sistem (program)

**Preduslov:** Sistem je uključen i zaposleni je prijavljen. Sistem prikazuje formu za rad sa terminom.

**Osnovni scenario SK**

1. Zaposleni **unosi** kriterijume po kojima pretražuje termine. (APUSO)
2. Zaposleni **poziva** sistem da pronađe termin na osnovu kriterijuma. (APSO)
3. Sistem **traži** termin. (S0)
4. Sistem **prikazuje** zaposlenom pronađene termine. (IA)
5. Zaposleni **bira** termin koji želi da pogleda. (APUSO)
6. Zaposleni **poziva** sistem da prikaže podatke o terminu. (APSO)
7. Sistem **traži** podatke o terminu. (SO)
8. Sistem **prikazuje** podatke o terminu i poruku: „Sistem je prikazao podatke o izabranom terminu“. (IA)

**Alternativna scenarija**

4.1. Ukoliko sistem nije uspeo da pronađe termine koji odgovaraju kriterijumu, zaposlenom se ispisuje odgovarajuća poruka: „Sistem nije uspeo da pronađe termine na osnovu kriterijuma!“. Prekida se izvršenje scenarija. (IA)

8.1. Ukoliko sistem nije uspeo da pronađe podatke o izabranom terminu, zaposlenom se ispisuje poruka: „Sistem nije uspeo da pronađe podatke o terminu.“ (IA)

# **Analiza**

Na osnovu prethodno definisanih korisničkih zahteva pravi se analiza koja služi za opis logičke strukture i ponašanja sistema tj. poslovne logike sistema.

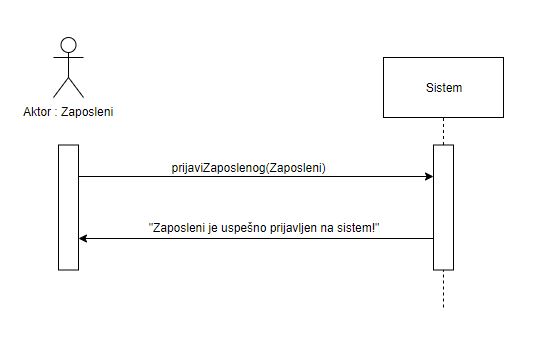
Ponašanje opisujemo pomoću sistemskih dijagrama sekvenci i ugovora o sistemskim operacijama.

Strukturu opisujemo pomoću konceptualnog modela i relacionog modela.

## **Ponašanje softverskog sistema – sistemski dijagrami sekvenci**

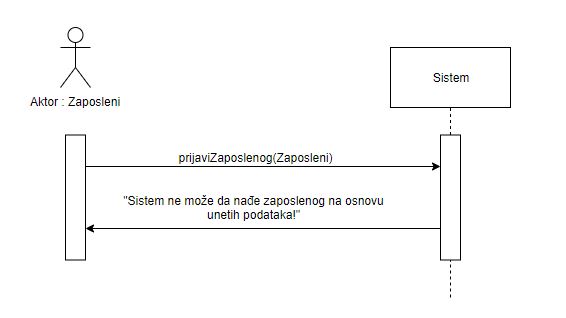
### **DS1: Dijagram sekvenci slučaja korišćenja - Prijavljivanje zaposlenog**

1. Zaposleni **poziva** sistem da ga prijavi na osnovu unetih podataka. (APSO)
2. Sistem **prikazuje** zaposlenom poruku : „Zaposleni je uspešno prijavljen na sistem!“. (IA)



**Alternativna scenarija**

* 1. Ukoliko sistem ne može da nađe radnika on prikazuje zaposlenom poruku: „Sistem ne može da nađe zaposlenog na osnovu unetih podataka!“. (IA)

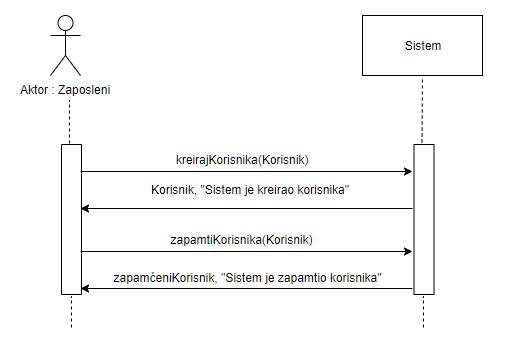


Sa navedenih sekvencnih dijagrama uočava se jedna sistemska operacija koju treba projektovati:

1.prijaviZaposlenog(Zaposleni);

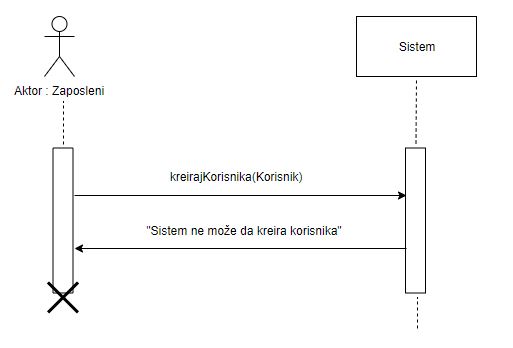
### **DS2: Dijagram sekvenci slučaja korišćenja - Unos novog korisnika**

1. Zaposleni **poziva** sistem da kreira korisnika. (APSO)
2. Sistem **prikazuje** zaposlenom korisnika i poruku: „Sistem je kreirao korisnika“. (IA)
3. Zaposleni **poziva** sistem da zapamti podatke o korisniku. (APSO)
4. Sistem **prikazuje** zaposlenom zapamćenog korisnika i poruku: „Sistem je zapamtio korisnika“. (IA)

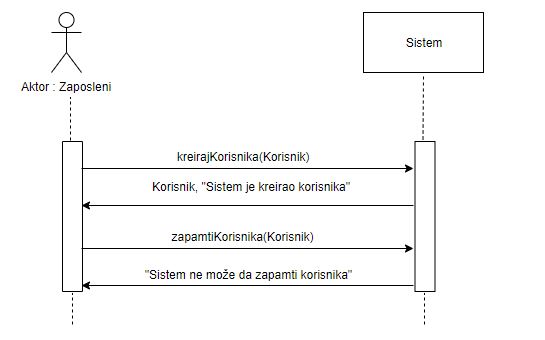


**Alternativna scenarija**

2.1. Ukoliko sistem ne može da kreira korisnika on prikazuje korisniku poruku: „Sistem ne može da kreira korisnika“. Prekida se izvršenje scenarija. (IA)



* 1. Ukoliko sistem ne može da zapamti podatke o korisniku on prikazuje zaposlenom poruku: „Sistem ne može da zapamti korisnika“.



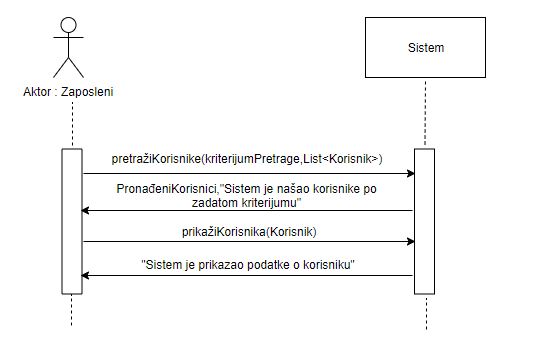
Sa navedenih sekvencnih dijagrama uočavaju se dve sistemske operacije koje treba projektovati:

1.signal kreirajKorisnika(Korisnik);

2.signal zapamtiKorisnika(Korisnik);

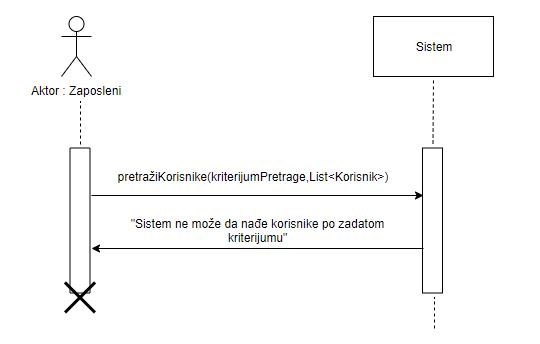
### **DS3: Dijagram sekvenci slučaja korišćenja – Pretraga korisnika**

1. Zaposleni **poziva** sistem da nađe korisnike po zadatom kriterijumu. (APSO)
2. Sistem **prikazuje** zaposlenom podatke o korisniku i poruku: „Sistem je našao korisnike po zadatom kriterijumu“. (IA)
3. Zaposleni **poziva** sistem da prikaže podatke o izabranom korisniku. (APSO)
4. Sistem **prikazuje** zaposlenom podatke o izabranom korisniku i poruku: „Sistem je prikazao podatke o korisniku“. (IA)

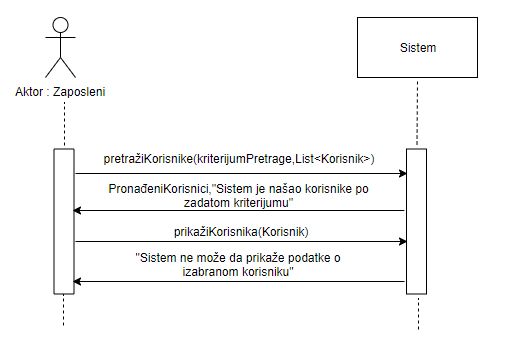


**Alternativna scenarija**

* 1. Ukoliko sistem ne može da nađe korisnike, on prikazuje zaposlenom poruku: „Sistem ne može da nađe korisnike po zadatom kriterijumu“. Prekida se izvršenje scenarija. (IA)



* 1. Ukoliko sistem ne može da nađe podatke o izabranom korisniku, on prikazuje poruku zaposlenom: „Sistem ne može da prikaže podatke o izabranom korisniku“. (IA)



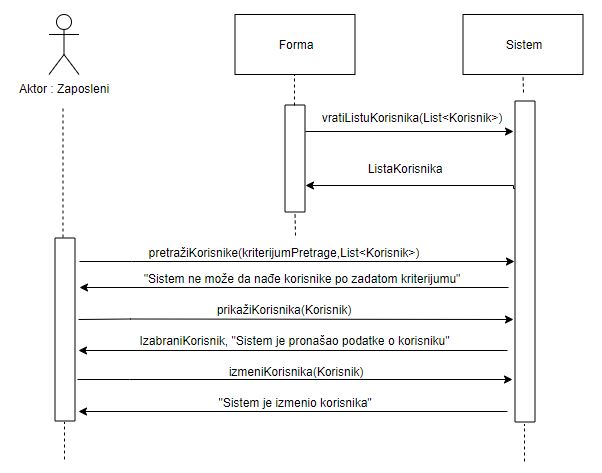
Sa navedenih sekvencnih dijagrama uočavaju se dve sistemske operacije koje treba projektovati:

1.signal pretražiKorisnike(kriterijumPretrage,List<Korisnik>);

2.signal prikažiKorisnika(Korisnik);

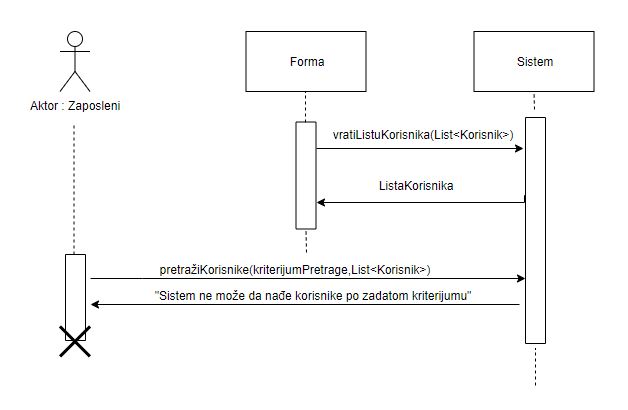
### **DS4: Dijagram sekvenci slučaja korišćenja – Izmena postojećeg korisnika**

1. Forma **poziva** sistem da vrati listu korisnika. (APSO)
2. Sistem **vraća** listu korisnika. (IA)
3. Zaposleni **poziva** sistem da nađe korisnika po zadatom kriterijumu. (APSO)
4. Sistem **prikazuje** zaposlenom korisnike i poruku: „Sistem je našao korisnike po zadatom kriterijumu“. (IA)
5. Zaposleni **poziva** sistem da učita podatke o izabranom korisniku. (APSO)
6. Sistem **prikazuje** zaposlenom podatke o izabranom korisniku i poruku: „Sistem je pronašao podatke o korisniku“. (IA)
7. Zaposleni **poziva** sistem da zapamti podatke o izmeni korisnika. (APSO)
8. Sistem **prikazuje** zaposlenom poruku: „Sistem je izmenio korisnika“. (IA)

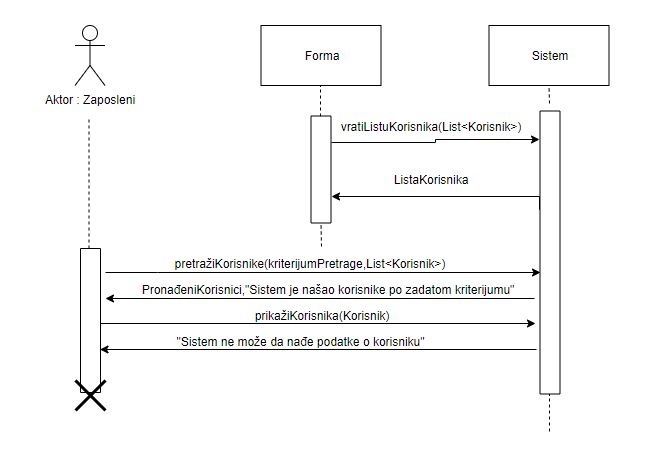


**Alternativna scenarija**

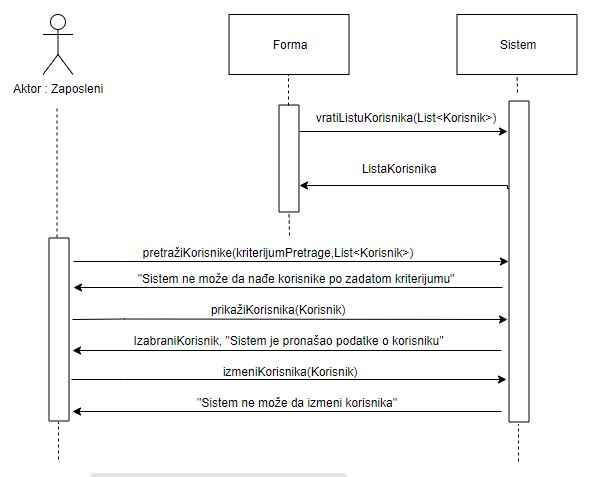
4.1. Ukoliko sistem ne može da nađe korisnike on prikazuje zaposlenom poruku: „Sistem ne može da nađe korisnike po zadatom kriterijumu“. Prekida se izvršenje scenarija. Prekida se izvršenje scenarija. (IA)



6.1. Ukoliko sistem ne može da nađe podatke o izabranom korisniku on prikazuje poruku zaposlenom: „Sistem ne može da nađe podatke o korisniku“. Prekida se izvršenje scenarija. (IA)



* 1. Ukoliko sistem ne može da zapamti podatke o izmeni korisnika on prikazuje zaposlenom poruku: „Sistem ne može da izmeni korisnika“. (IA)



Sa navedenih sekvencnih dijagrama uočava se pet sistemskih operacija koje treba projektovati:

1.signal vratiListuKorisnika(List<Korisnik>);

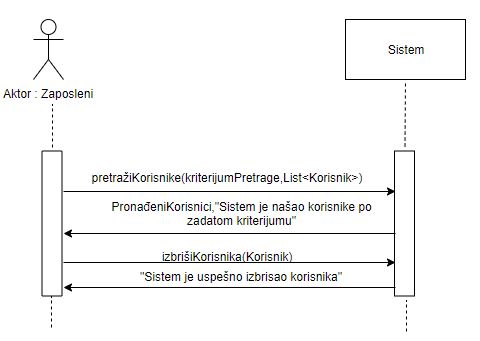
2.signal pretražiKorisnike(kriterijumPretrage,List<Korisnik>);

3.signal prikažiKorisnika(Korisnik);

4.signal izmeniKorisnika(Korisnik);

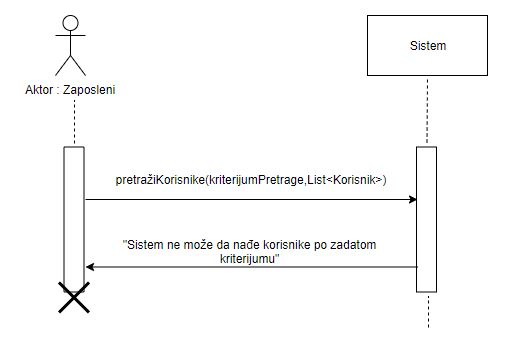
### **DS5: Dijagram sekvenci slučaja korišćenja – Brisanje korisnika**

1. Zaposleni **poziva** sistem da nađe korisnike po zadatom kriterijumu. (APSO)
2. Sistem **prikazuje** zaposlenom korisnike i poruku: „Sistem je pronašao korisnike po zadatom kriterijumu“. (IA)
3. Zaposleni **poziva** sistem da izbriše korisnika. (APSO)
4. Sistem **prikazuje** zaposlenom poruku: „Sistem je uspešno izbrisao korisnika“. (IA)

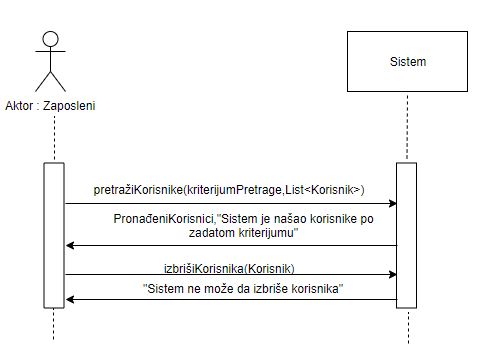


**Alternativna scenarija**

2.1. Ukoliko sistem ne može da nađe korisnike on prikazuje zaposlenom poruku: „Sistem ne može da nađe korisnike po zadatom kriterijumu“. Prekida se izvršenje scenarija. (IA)



4.1. Ukoliko sistem ne može da izbriše korisnika on prikazuje zaposlenom poruku: „Sistem ne može da izbriše korisnika“. (IA)



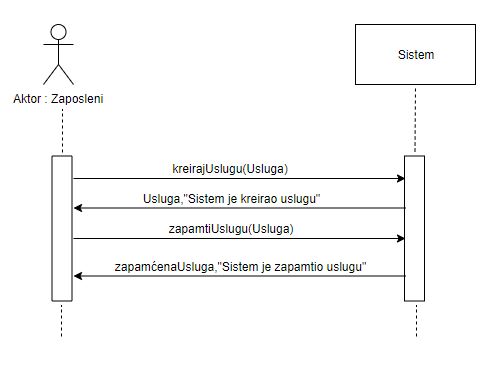
Sa navedenih sekvencnih dijagrama uočavaju se dve sistemske operacije koje treba projektovati:

1.signal pretražiKorisnike(kriterijumPretrage,List<Korisnik>);

2.signal izbrišiKorisnika(Korisnik);

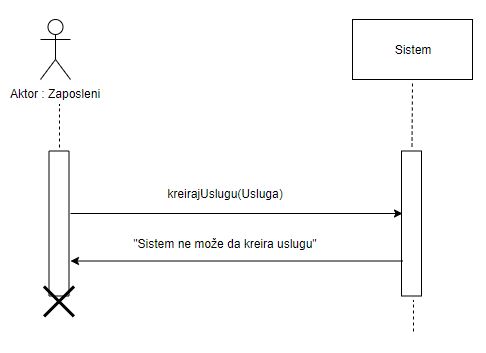
### **DS6: Dijagram sekvenci slučaja korišćenja – Unos nove usluge**

1. Zaposleni **poziva** sistem da kreira novu uslugu. (APSO)
2. Sistem **prikazuje** zaposlenom uslugu i poruku: „Sistem je kreirao uslugu“. (IA)
3. Zaposleni **poziva** sistem da zapamti podatke o usluzi. (APSO)
4. Sistem **prikazuje** zaposlenom zapamćenu uslugu i poruku: „Sistem je zapamtio uslugu“. (IA)

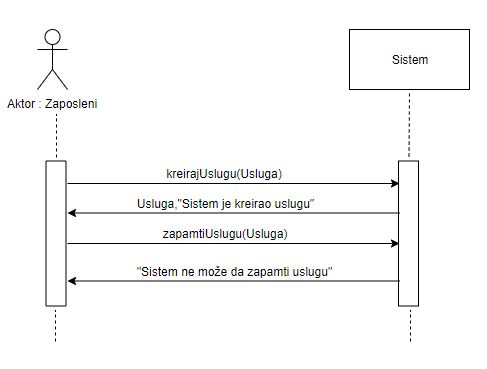


**Alternativna scenarija**

2.1. Ukoliko sistem ne može da kreira uslugu on prikazuje korisniku poruku: „Sistem ne može da kreira uslugu“. Prekida se izvršenje scenarija. (IA)



4.1. Ukoliko sistem ne može da zapamti podatke o usluzi on prikazuje zaposlenom poruku: „Sistem ne može da zapamti uslugu“.



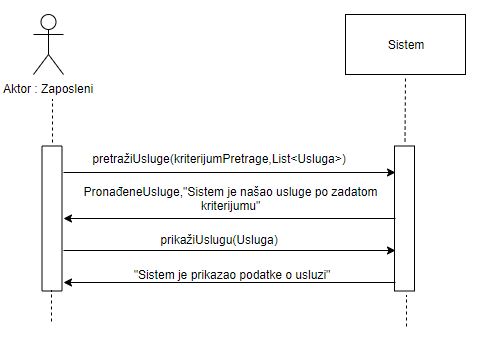
Sa navedenih sekvencnih dijagrama uočavaju se dve sistemske operacije koje treba projektovati:

1.signal kreirajUslugu(Usluga);

2.signal zapamtiUslugu(Usluga);

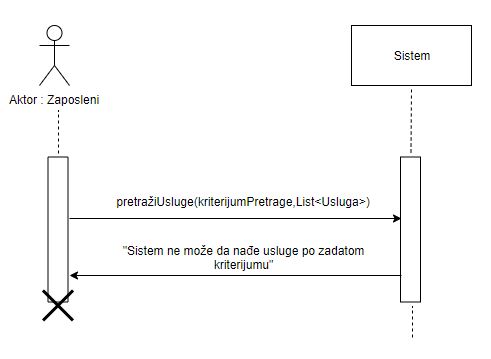
### **DS7: Dijagram sekvenci slučaja korišćenja – Pretraga usluge**

1. Zaposleni **poziva** sistem da nađe usluge po zadatom kriterijumu. (APSO)
2. Sistem **prikazuje** zaposlenom podatke o usluzi i poruku: „Sistem je našao usluge po zadatom kriterijumu“. (IA)
3. Zaposleni **poziva** sistem da prikaže podatke o izabranoj usluzi. (APSO)
4. Sistem **prikazuje** zaposlenom podatke o izabranoj usluzi i poruku: „Sistem je prikazao podatke o usluzi“. (IA)

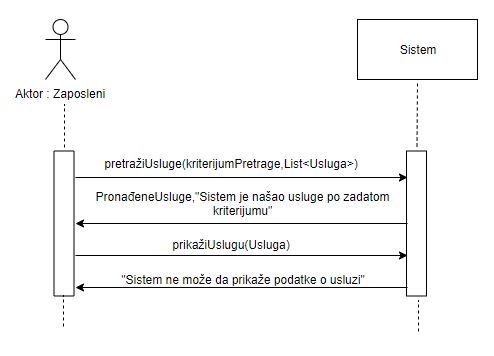


**Alternativna scenarija**

2.1.Ukoliko sistem ne može da nađe usluge, on prikazuje zaposlenom poruku: „Sistem ne može da nađe usluge po zadatom kriterijumu“. Prekida se izvršenje scenarija. (IA)



* 1. Ukoliko sistem ne može da nađe podatke o izabranoj usluzi, on prikazuje poruku zaposlenom: „Sistem ne može da prikaže podatke o usluzi“. (IA)



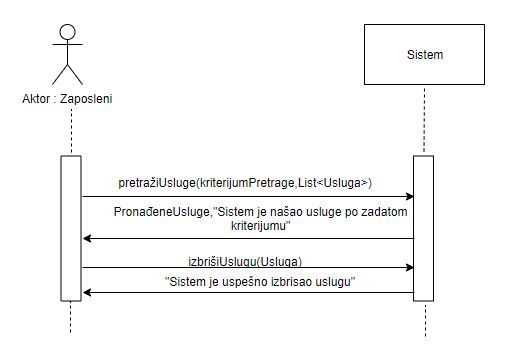
Sa navedenih sekvencnih dijagrama uočavaju se dve sistemske operacije koje treba projektovati:

1.signal pretražiUsluge(kriterijumPretrage,List<Usluga>);

2.signal prikažiUslugu(Usluga);

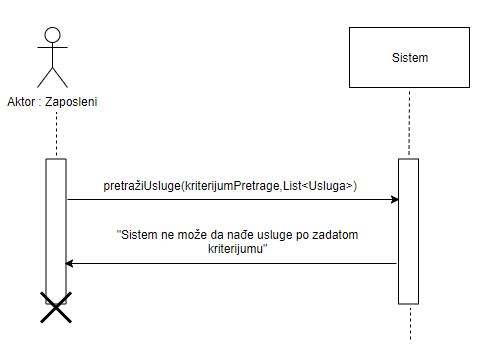
### **DS8: Dijagram sekvenci slučaja korišćenja – Brisanje usluge**

1. Zaposleni **poziva** sistem da nađe usluge po zadatom kriterijumu. (APSO)
2. Sistem **prikazuje** zaposlenom usluge i poruku: „Sistem je pronašao usluge po zadatom kriterijumu“. (IA)
3. Zaposleni **poziva** sistem da izbriše uslugu. (APSO)
4. Sistem **prikazuje** zaposlenom poruku: „Sistem je uspešno izbrisao uslugu“. (IA)

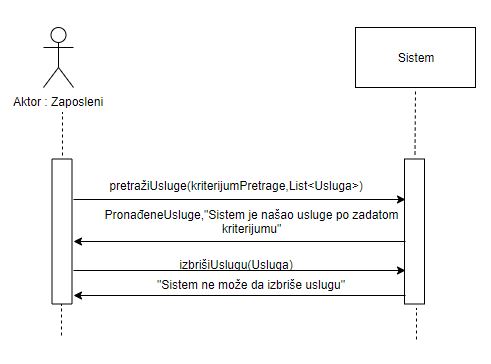


**Alternativna scenarija**

2.1. Ukoliko sistem ne može da nađe usluge on prikazuje zaposlenom poruku: „Sistem ne može da nađe usluge po zadatom kriterijumu“. Prekida se izvršenje scenarija. (IA)



4.1. Ukoliko sistem ne može da izbriše uslugu on prikazuje zaposlenom poruku: „Sistem ne može da izbriše uslugu“. (IA)



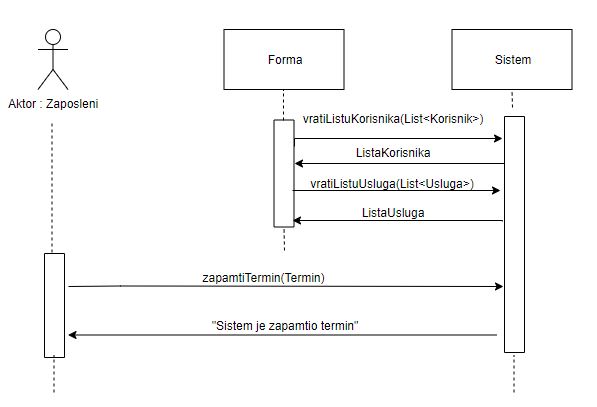
Sa navedenih sekvencnih dijagrama uočavaju se dve sistemske operacije koje treba projektovati:

1.signal pretražiUsluge(kriterijumPretrage,List<Usluga>);

2.signal izbrišiUslugu(Usluga);

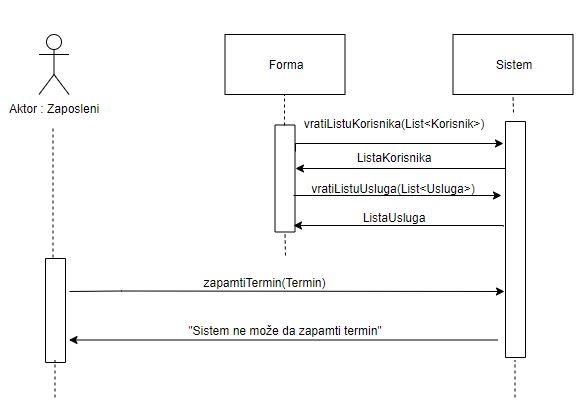
### **DS9: Dijagram sekvenci slučaja korišćenja – Zakazivanje termina**

1. Forma **poziva** sistem da vrati listu korisnika. (APSO)
2. Sistem **vraća** listu korisnika. (IA)
3. Forma **poziva** sistem da vrati listu usluga. (APSO)
4. Sistem **vraća** listu usluga. (IA)
5. Zaposleni **poziva** sistem da zapamti podatke o terminu. (APSO)
6. Sistem **prikazuje** zaposlenom termin i poruku: „Sistem je zapamtio termin“. (IA)



**Alternativna scenarija**

* 1. Ukoliko sistem ne može da zapamti podatke o terminu, on prikazuje zaposlenom poruku: „Sistem ne može da zapamti termin“.

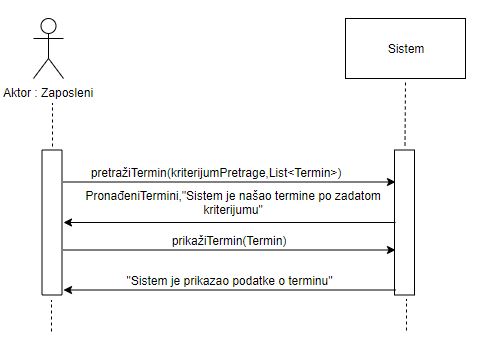


Sa navedenih sekvencnih dijagrama uočavaju se tri sistemske operacije koje treba projektovati:

1. Signal vratiListuKorisnika(List<Korisnik>););
2. Signal vratiListuUsluga(List<Usluga>);
3. Signal zapamtiTermin(Termin);

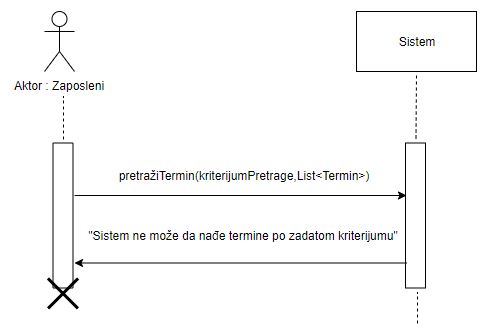
### **2.1.10. DS10: Dijagram sekvenci slučaja korišćenja – Pretraživanje termina**

1. Zaposleni **poziva** sistem da nađe termine po zadatom kriterijumu. (APSO)
2. Sistem **prikazuje** zaposlenom podatke o terminu i poruku: „Sistem je našao termine po zadatom kriterijumu“. (IA)
3. Zaposleni **poziva** sistem da prikaže podatke o izabranom terminu. (APSO)
4. Sistem **prikazuje** zaposlenom podatke o izabranom terminu i poruku: „Sistem je prikazao podatke o terminu“. (IA)

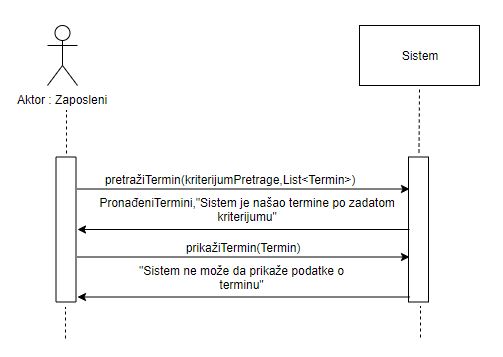


**Alternativna scenarija**

2.1. Ukoliko sistem ne može da nađe termine, on prikazuje zaposlenom poruku: „Sistem ne može da nađe termine po zadatom kriterijumu“. Prekida se izvršenje scenarija. (IA)



* 1. Ukoliko sistem ne može da nađe podatke o izabranom terminu, on prikazuje poruku zaposlenom: „Sistem ne može da prikaže podatke o terminu“. (IA)



Sa navedenih sekvencnih dijagrama uočavaju se dve sistemske operacije koje treba projektovati:

1.signal pretražiTermine(kriterijumPretrage,List<Termin>);

2.signal prikažiTermin(Termin);

Kao rezultat scenarija dobijeno je ukupno 17 sistemskih operacija koje treba projektovati:

1. prijaviZaposlenog(Zaposleni);
2. signal vratiListuKorisnika(List<Korisnik>);
3. signal vratiListuUsluga(List<Usluga>);
4. signal kreirajKorisnika(Korisnik);
5. signal zapamtiKorisnika(Korisnik);
6. signal pretražiKorisnike(kriterijumPretrage,List<Korisnik>);
7. signal prikažiKorisnika(Korisnik);
8. signal izmeniKorisnika(Korisnik);
9. signal izbrišiKorisnika(Korisnik);
10. signal kreirajUslugu(Usluga);
11. signal zapamtiUslugu(Usluga);
12. signal pretražiUsluge(kriterijumPretrage,List<Usluga>);
13. signal prikažiUslugu(Usluga);
14. signal izbrišiUslugu(Usluga);
15. signal zapamtiTermin(Termin);
16. signal pretražiTermine(kriterijumPretrage,List<Termin>);
17. signal prikažiTermin(Termin);

## **Ponašanje softverskog sistema – Definisanje ugovora o sistemskim operacijama**

**1.Ugovor UG1: prijaviZaposlenog**

Operacija: prijaviZaspolenog(Zaposleni)

Veza sa SK: SK1

Preduslovi: /

Postuslovi: /

**2.Ugovor UG2: vratiListuKorisnika**

Operacija: vratiListuKorisnika(List<Korisnik>):signal;

Veza sa SK: SK4, SK9

Preduslovi: /

Postuslovi: /

**3.Ugovor UG3: vratiListuUsluga**

Operacija: vratiListuUsluga(List<Usluga>):signal;

Veza sa SK: SK9

Preduslovi: /

Postuslovi: /

**4.Ugovor UG4: kreirajKorisnika**

Operacija: kreirajKorisnika(Korisnik):signal;

Veza sa SK: SK2

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

Postuslovi: Kreiran je novi objekat odgovarajuće domenske klase.

**5.Ugovor UG5: zapamtiKorisnika**

Operacija: zapamtiKorisnika(Korisnik):signal;

Veza sa SK: SK2

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

Postuslovi: Podaci o korisniku su zapamćeni.

**6.Ugovor UG6: pretražiKorisnike**

Operacija: pretražiKorisnike(kriterijumPretrage,List<Korisnik>):signal;

Veza sa SK: SK3, SK4, SK5

Preduslovi: /

Postuslovi: /

**7.Ugovor UG7: prikažiKorisnika**

Operacija: prikažiKorisnika(Korisnik):signal;

Veza sa SK: SK3, SK4, SK5

Preduslovi: /

Postuslovi: /

**8.Ugovor UG8: izmeniKorisnika**

Operacija: izmeniKorisnika(Korisnik):signal;

Veza sa SK: SK4

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

Postuslovi: Objekat Korisnik je izmenjen.

**9.Ugovor UG9: izbrišiKorisnika**

Operacija: izbrišiKorisnika(Korisnik):signal;

Veza sa SK: SK5

Preduslovi: Strukturna ograničenja nad objektom Korisnik moraju biti zadovoljena.

Postuslovi: Objekat Korisnik je izbrisan.

**10. Ugovor UG10: kreirajUslugu**

Operacija: kreirajUslugu(Usluga):signal;

Veza sa SK: SK6

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

Postuslovi: Kreiran je novi objekat odgovarajuće domenske klase.

**11.Ugovor UG11: zapamtiUslugu**

Operacija: zapamtiUslugu(Usluga):signal;

Veza sa SK: SK6

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

Postuslovi: Podaci o usluzi su zapamćeni.

**12.Ugovor UG12: pretražiUsluge**

Operacija: pretražiUsluge(kriterijumPretrage,List<Usluga>):signal;

Veza sa SK: SK7, SK8

Preduslovi: /

Postuslovi: /

**13.Ugovor UG13: prikažiUslugu**

Operacija: prikažiUslugu(Usluga):signal;

Veza sa SK: SK7, SK8

Preduslovi: /

Postuslovi: /

**14.Ugovor UG14: izbrišiUslugu**

Operacija: izbrišiUslugu(Usluga):signal;

Veza sa SK: SK8

Preduslovi: Strukturna ograničenja nad objektom Usluga moraju biti zadovoljena.

Postuslovi: Objekat Usluga je izbrisan.

**15.Ugovor UG15: zapamtiTermin**

Operacija: zapamtiTermin(Termin):signal;

Veza sa SK: SK9

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

Postuslovi: Podaci o terminu su zapamćeni.

**16.Ugovor UG16: pretražiTermine**

Operacija: pretražiTermine(kriterijumPretrage,List<Termin>):signal;

Veza sa SK: SK10

Preduslovi: /

Postuslovi: /

**17.Ugovor UG17: prikažiTermin**

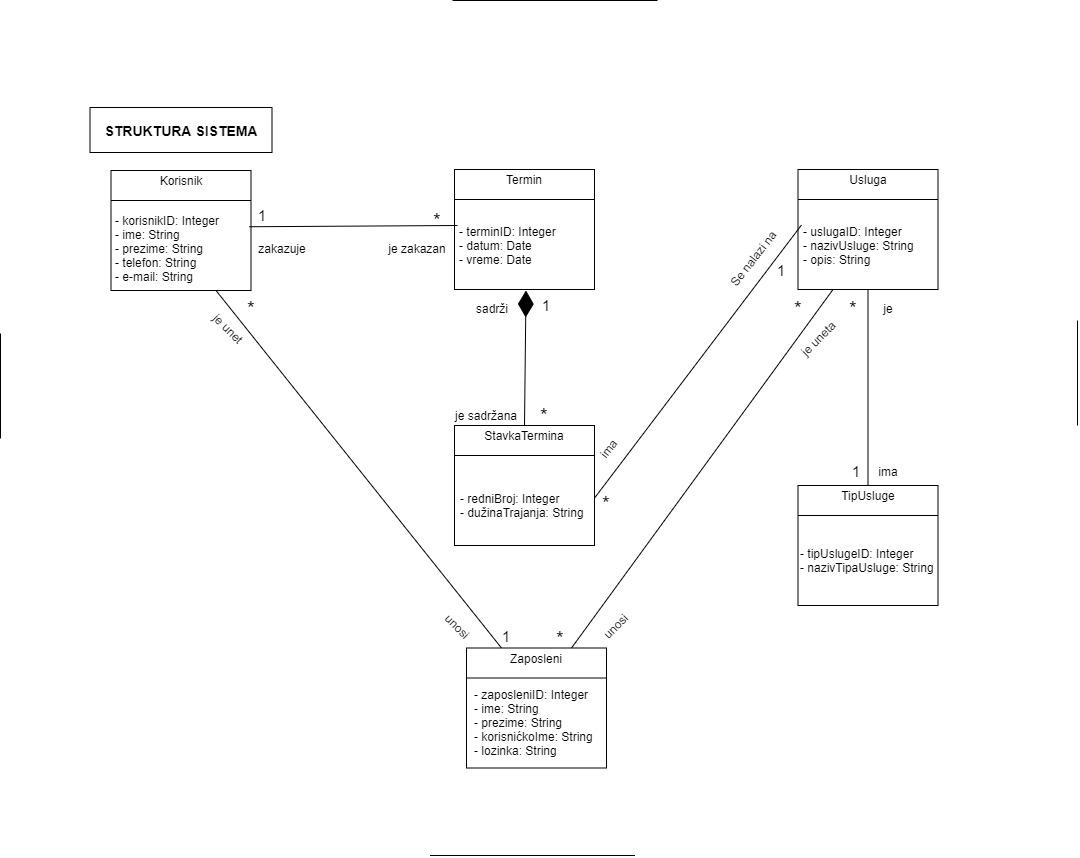
Operacija: prikažiTermin(Termin):signal;

Veza sa SK: SK10

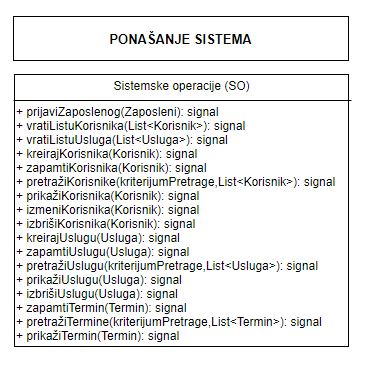
Preduslovi: /

Postuslovi: /

## **Struktura softverskog sistema – Konceptualni (domenski) model**



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



## **Struktura softverskog sistema – Relacioni model**

Na osnovu konceptualnog modela koji je predstavljen dijagramom klasa dobijen je sledeći relacioni model:

**Zaposleni**(ZaposleniID, Ime, Prezime, KorisničkoIme, Lozinka)

**Korisnik**(KorisnikID, Ime, Prezime, Telefon, E-mail, *ZaposleniID*)

**Usluga**(UslugaID, NazivUsluge, Opis, *TipUslugeID, ZaposleniID*)

**TipUsluge**(UslugaID, TipUslugeID, NazivTipaUsluge)

**Termin**(TerminID, Datum, Vreme, *KorisnikID*)

**StavkaTermina**(TerminID, RedniBroj, DužinaTrajanja, *UslugaID*)

**ZaposleniUsluga**(ZaposleniID, UslugaID)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela Zaposleni | | Prosto vrednosno ograničenje | | Složeno vrednosno ograničenje | | Strukturno ograničenje |
| Atributi | Ime | Tip atributa | Vrednost atributa | Međuzav. Atributa jedne tabele | Međuzav. Atributa više tabela | INSERT /  UPDATE  CASCADES Korisnik, Zaposleni  DELETE  RESTRICTED  Korisnik, Zaposleni |
| ZaposleniID | Integer | not null and >0 |  |  |
| Ime | String | not null |  |  |
| Prezime | String | not null |  |  |
| KorisničkoIme | String | not null |  |  |
| Lozinka | String | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela Korisnik | | Prosto vrednosno ograničenje | | Složeno vrednosno ograničenje | | Strukturno ograničenje |
| Atributi | Ime | Tip atributa | Vrednost atributa | Međuzav. Atributa jedne tabele | Međuzav. Atributa više tabela | INSERT /  UPDATE  CASCADES  Termin  DELETE  RESTRICTED  Termin |
| KorisnikID | Integer | not null and >0 |  |  |
| Ime | String | not null |  |  |
| Prezime | String | not null |  |  |
| Telefon | String | not null |  |  |
| E-mail | String |  |  |  |
| ZaposleniID | Integer | not null and >0 |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela Usluga | | Prosto vrednosno ograničenje | | Složeno vrednosno ograničenje | | Strukturno ograničenje |
| Atributi | Ime | Tip atributa | Vrednost atributa | Međuzav. Atributa jedne tabele | Međuzav. Atributa više tabela | INSERT RESTRICTED  TipUsluge  UPDATE  RESTRICTED  TipUsluge  CASCADES  StavkaTermina, ZaposleniUsluga  DELETE  RESTRICTED  StavkaTermina, ZaposleniUsluga |
| UslugaID | Integer | not null and >0 |  |  |
| NazivUsluge | String | not null |  |  |
| Opis | String | not null |  |  |
| TipUslugeID | Integer | not null and >0 |  |  |
| ZaposleniID | Integer | not null and >0 |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela TipUsluge | | Prosto vrednosno ograničenje | | Složeno vrednosno ograničenje | | Strukturno ograničenje |
| Atributi | Ime | Tip atributa | Vrednost atributa | Međuzav. Atributa jedne tabele | Međuzav. Atributa više tabela | INSERT /  UPDATE CASCADES  Usluga  DELETE RESTRICTED  Usluga |
| UslugaID | Integer | not null and >0 |  |  |
| TipUslugeID | Integer | not null and >0 |  |  |
| NazivTipaUsluge | String | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela Termin | | Prosto vrednosno ograničenje | | Složeno vrednosno ograničenje | | Strukturno ograničenje |
| Atributi | Ime | Tip atributa | Vrednost atributa | Međuzav. Atributa jedne tabele | Međuzav. Atributa više tabela | INSERT  RESTRICTED  Korisnik  UPDATE RESTRICTED Korisnik  CASCADES  StavkaTermina  DELETE  CASCADES  StavkaTermina |
| TerminID | Integer | not null and >0 |  |  |
| Datum | Date | not null |  |  |
| Vreme | Date | not null |  |  |
| KorisnikID | Integer | not null and >0 |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela StavkaTermina | | Prosto vrednosno ograničenje | | Složeno vrednosno ograničenje | | Strukturno ograničenje |
| Atributi | Ime | Tip atributa | Vrednost atributa | Međuzav. Atributa jedne tabele | Međuzav. Atributa više tabela | INSERT  RESTRICTED  Termin,  Usluga  UPDATE  RESTRICTED  Termin,  Usluga  DELETE / |
| TerminID | Integer | not null and >0 |  |  |
| RedniBroj | Integer | not null and >0 |  |  |
| DužinaTrajanja | String | not null |  |  |
| UslugaID | Integer | not null and >0 |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela ZaposleniUsluga | | Prosto vrednosno ograničenje | | Složeno vrednosno ograničenje | | Strukturno ograničenje |
| Atributi | Ime | Tip atributa | Vrednost atributa | Međuzav. Atributa jedne tabele | Međuzav. Atributa više tabela | INSERT RESTRICTED Zaposleni, Usluga  UPDATE RESTRICTED Zaposleni, Usluga  DELETE / |
| ZaposleniID | Integer | not null and >0 |  |  |
| UslugaID | Integer | not null and >0 |  |  |

# 3. PROJEKTOVANJE

Faza projektovanja opisuje fizičku strukturu i ponašanje softverskog sistema arhitekturu softverskog sistema). Projektovanje arhitekture softverskog sistema obuhvata projektovanje korisničkog interfejsa, aplikacione logike i skladišta podataka. Projektovanje korisničkog interfejsa obuhvata projektovanje ekranskih formi i kontrolera korisničkog interfejsa. U okviru aplikacione logike se projektuju kontroler aplikacione logike, poslovna logika I broker baze podataka. Projektovanje poslovne logika obuhvata projektovanje logičke structure i ponašanja softverskog sistema.

## 3.1 Arhitektura Softverskog Sistema

U okviru faze projektovanja opisuje se fizička struktura i ponašanje softverskog sistema tj. arhitektura softverskog sistema. U ovom seminarskom radu je korišćena klasična troslojna arhitektura, koja se sastoji od:

* Korisničkog interfejsa
* Aplikacione logike
* Skladišta podataka

Korisnički interfejs

Aplikaciona logika

Skladište podataka

**I sloj**

**II sloj**

**III sloj**

**Softverski sistem**

*Slika: Troslojna arhitektura*

Skladište podataka

**Korisnički interfejs**

Ekranska forma

KontrolerKI

**Aplikaciona logika**

KontrolerAL

Poslovna logika

Database broker

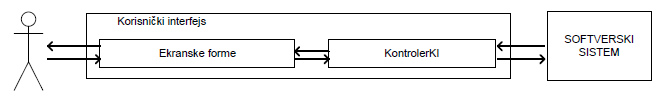
*Slika: Troslojna arhitektura – detaljniji prikaz*

## 3.2. Projektovanje korisničkog interfejsa

Kontroler korisničkog interfejsa i ekranske forme ne predstavljaju softverski sistem, već realizaciju ulaza i izlaza iz softverskog sistema. Kontroler korisničkog interfejsa ima uloga da konvertuje podatke u grafičke elemente ekranskih formi i da podatke ili pošalje u softverski sistem ili da primi podatke iz softverskog sistema i iste konvertuje u elemente ekranske forme.

Korisnički interfejs se sastoji iz:

* Ekranske forme,
* Kontrolera korisničkog interfejsa.



### Projektovanje ekranskih formi

### **Slučaj korišćenja – Prijavljivanje zaposlenog**

**Naziv SK**

Prijavljivanje zaposlenog

**Aktori SK**

Zaposleni

**Učesnici SK**

Zaposleni i sistem (program)

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

**Osnovni scenario SK**

1. Zaposleni **unosi** podatke za prijavljivanje zaposlenog. (APUSO)

2. Zaposleni **kontroliše** da li je ispravno uneo podatke za registrovanje. (ANSO)

3. Zaposleni **poziva** sistem da ga pronađe na osnovu unetih podataka. (APSO)

4. Sistem **pretražuje** zaposlene. (SO)

5. Sistem **prikazuje** zaposlenom poruku na ekranu: „Zaposleni je uspešno registrovan u sistem“ i omogućava pristup sistemu. (IA)

**Alternativna scenarija**

5.1. Ukoliko sistem ne može da pronađe zaposlenog na osnovu unetih podataka, prikazuje poruku „Sistem nije uspeo da pronađe zaposlenog na osnovu unetih podataka“. (IA)

### **Slučaj korišćenja – Unos novog korisnika**

**Naziv SK**

Unos novog korisnika

**Aktori SK**

Zaposleni

**Učesnici SK**

Zaposleni i sistem (program)

**Preduslov:** Sistem je uključen i zaposleni je prijavljen. Sistem prikazuje formu za unos novog korisnika.

**Osnovni scenario SK**

1. Zaposleni **poziva** sistem da kreira novog korisnika. (APSO)
2. Sistem **kreira** korisnika. (SO)
3. Sistem **prikazuje** zaposlenom poruku: „Novi korisnik je uspešno kreiran.“ (IA)
4. Zaposleni **unosi** podatke za kreiranog korisnika. (ANSO)
5. Zaposleni **poziva** sistem da sačuva podatke o novom korisniku. (APSO)
6. Sistem **pamti** podatke o korisniku. (SO)
7. Sistem **prikazuje** zaposlenom sačuvane podatke o novom korisniku i ispisuje poruku: „Sistem je uspešno zapamtio podatke o korisniku.“ (IA)

**Alternativna scenarija**

3.1. Ukoliko sistem nije uspeo da kreira novog korisnika, on prikazuje zaposlenom poruku: „Sistem ne može da kreira novog korisnika.“ Prekida se izvršenje scenarija. (IA)

7.1. Ukoliko sistem nije uspeo da zapamti podatke o korisniku, prikazuje zaposlenom sledeću poruku: „Sistem nije uspeo da sačuva korisnika“. (IA)

### **Slučaj korišćenja – Pretraga korisnika**

**Naziv SK**

Pretraga korisnika

**Aktori SK**

Zaposleni

**Učesnici SK**

Zaposleni i sistem (program)

**Preduslov:** Sistem je uključen i zaposleni je prijavljen. Sistem prikazuje formu za rad sa korisnikom.

**Osnovni scenario SK**

1. Zaposleni **unosi** kriterijum u formu za pretraživanje korisnika. (APUSO)
2. Zaposleni **poziva** sistem da pronađe korisnika na osnovu zadatog kriterijuma. (APSO)
3. Sistem **traži** korisnika. (SO)
4. Sistem **prikazuje** pronađene korisnike. (IA)
5. Zaposleni **bira** korisnika čije podatke želi da vidi. (APUSO)
6. Zaposleni **poziva** sistem da prikaže podatke o izabranom korisniku. (APSO)
7. Sistem **pronalazi** podatke o izabranom korisniku. (SO)
8. Sistem **prikazuje** podatke o izabranom korisniku i poruku: „Sistem je prikazao podatke o korisniku“. (IA)

**Alternativna scenarija**

4.1. Ukoliko sistem nije uspeo da pronađe korisnike, prikazuje se poruka zaposlenom: „Sistem nije uspeo da pronađe korisnike na osnovu zadatog kriterijuma.“ Prekida se izvršenje scenarija. (IA)

8.1. Ukoliko sistem nije uspeo da pronađe podatke o izabranom korisniku, prikazuje se odgovarajuća poruka zaposlenom: „Ne postoje podaci o izabranom korisniku“. (IA)

### **Slučaj korišćenja – Izmena postojećeg korisnika**

**Naziv SK**

Izmena postojećeg korisnika

**Aktori SK**

Zaposleni

**Učesnici SK**

Zaposleni i sistem (program)

**Preduslov:** Sistem je uključen i zaposleni je prijavljen. Učitana je lista korisnika. Sistem prikazuje formu za rad sa korisnikom.

**Osnovni scenario SK**

1. Zaposleni **unosi** kriterijum po kom pretražuje korisnike. (APUSO)
2. Zaposleni **poziva** sistem da pronađe korisnike na osnovu zadatog kriterijuma. (APSO)
3. Sistem **traži** korisnika. (SO)
4. Sistem **prikazuje** zaposlenom pronađene korisnike i poruku: “Sistem je našao korisnike prema zadatom kriterijumu“. (IA)
5. Zaposleni **bira** korisnika kog želi da izmeni. (APUSO)
6. Zaposleni **poziva** sistem da učita podatke o izabranom korisniku. (APSO)
7. Sistem **pronalazi** podatke o izabranom korisniku. (SO)
8. Sistem **prikazuje** zaposlenom podatke o izabranom korisniku i poruku: „Sistem je pronašao podatke o korisniku.“ (IA)
9. Zaposleni **unosi** nove vrednosti za odabranog korisnika. (APUSO)
10. Zaposleni **poziva** sistem da izmeni izabranog korisnika. (APSO)
11. Sistem **pamti** izmenjene podatke o korisniku. (SO)
12. Sistem **prikazuje** izmenjene podatke o korisniku i poruku: „Sistem je uspešno izmenio podatke o korisniku“. (IA)

**Alternativna scenarija**

4.1. Ukoliko sistem nije uspeo da pronađe korisnike koji odgovaraju zadatom kriterijumu, zaposlenom se ispisuje odgovarajuća poruka: „Sistem nije uspeo da pronađe korisnike na osnovu zadatog kriterijuma!“. Prekida se izvršenje scenarija. (IA)

8.1. Ukoliko sistem nije uspeo da pronađe podatke o odabranom korisniku, zaposlenom se ispisuje poruka: „Sistem nije uspeo da pronađe podatke o korisniku.“. Prekida se izvršenje scenarija. (IA)

12.1. Ukoliko sistem nije uspeo da sačuva izmenjene podatke o korisniku, zaposlenom se ispisuje poruka: „Sistem ne može da zapamti korisnika!“. (IA)

### **Slučaj korišćenja – Brisanje korisnika**

**Naziv SK**

Brisanje korisnika

**Aktori SK**

Zaposleni

**Učesnici SK**

Zaposleni i sistem (program)

**Preduslov:** Sistem je uključen i zaposleni je prijavljen. Sistem prikazuje formu za rad sa korisnikom.

**Osnovni scenario SK**

1. Zaposleni **unosi** kriterijum po kom pretražuje korisnike. (APUSO)
2. Zaposleni **poziva** sistem da pronađe korisnike na osnovu zadatog kriterijuma. (APSO)
3. Sistem **traži** korisnika. (SO)
4. Sistem **prikazuje** zaposlenom pronađene korisnike. (IA)
5. Zaposleni **bira** korisnika kog želi da obriše. (APUSO)
6. Zaposleni **poziva** sistem da obriše izabranog korisnika. (APSO)
7. Sistem **briše** odabranog korisnika. (SO)
8. Sistem **prikazuje** zaposlenom odgovarajuću poruku: „Korisnik je uspešno obrisan“. (IA)

**Alternativna scenarija**

4.1. Ukoliko sistem nije uspeo da pronađe korisnike koji odgovaraju zadatom kriterijumu, zaposlenom se ispisuje odgovarajuća poruka: „Sistem nije uspeo da pronađe korisnike na osnovu kriterijuma!“. Prekida se izvršenje scenarija. (IA)

8.1. Ukoliko sistem nije uspeo da obriše odabranog korisnika, ispisuje se poruka zaposlenom: „Sistem nije uspeo da obriše korisnika.“. (IA)

### **Slučaj korišćenja – Unos nove usluge**

**Naziv SK**

Unos nove usluge

**Aktori SK**

Zaposleni

**Učesnici SK**

Zaposleni i sistem (program)

**Preduslov:** Sistem je uključen i zaposleni je prijavljen. Sistem prikazuje formu za unos nove usluge.

**Osnovni scenario SK**

1. Zaposleni **poziva** sistem da kreira novu uslugu. (APSO)
2. Sistem **kreira** novu uslugu. (SO)
3. Sistem zaposlenom **prikazuje** poruku: „Nova usluga je uspešno kreirana.“. (IA)
4. Zaposleni **unosi** podatke o usluzi. (APUSO)
5. Zaposleni **poziva** sistem da sačuva podatke o novoj usluzi. (APSO)
6. Sistem **pamti** podatke o usluzi. (SO)
7. Sistem **prikazuje** zaposlenom sačuvane podatke o novoj usluzi i ispisuje poruku: „Sistem je uspešno zapamtio podatke o usluzi!“ (IA)

**Alternativna scenarija**

3.1. Ukoliko sistem nije uspeo da kreira novu uslugu, zaposlenom se prikazuje sledeća poruka: „Sistem ne može da kreira novu uslugu.“ Prekida se izvršenje scenarija. (IA)

7.1. Ukoliko sistem nije uspeo da zapamti podatke o usluzi, prikazuje zaposlenom sledeću poruku: „Sistem nije uspeo da sačuva uslugu.“ (IA)

### **Slučaj korišćenja – Pretraga usluge**

**Naziv SK**

Pretraga usluge

**Aktori SK**

Zaposleni

**Učesnici SK**

Zaposleni i sistem (program)

**Preduslov:** Sistem je uključen i zaposleni je prijavljen. Sistem prikazuje formu za rad sa uslugama.

**Osnovni scenario SK**

1. Zaposleni **unosi** kriterijum po kom pretražuje usluge. (APUSO)
2. Zaposleni **poziva** sistem da pronađe usluge na osnovu kriterijuma. (APSO)
3. Sistem **traži** uslugu. (S0)
4. Sistem **prikazuje** zaposlenom pronađene usluge. (IA)
5. Zaposleni **bira** uslugu koju želi da pogleda. (APUSO)
6. Zaposleni **poziva** sistem da prikaže podatke o izabranoj usluzi. (APSO)
7. Sistem **traži** podatke o izabranoj usluzi. (SO)
8. Sistem **prikazuje** podatke o izabranoj usluzi i poruku: „Sistem je prikazao podatke o izabranoj usluzi“. (IA)

**Alternativna scenarija**

4.1. Ukoliko sistem nije uspeo da pronađe usluge koje odgovaraju kriterijumu, zaposlenom se ispisuje odgovarajuća poruka: „Sistem nije uspeo da pronađe usluge na osnovu kriterijuma!“. Prekida se izvršenje scenarija. (IA)

8.1. Ukoliko sistem nije uspeo da pronađe podatke o izabranoj usluzi, zaposlenom se ispisuje poruka: „Sistem nije uspeo da pronađe podatke o usluzi.“ (IA)

### **Slučaj korišćenja – Brisanje usluge**

**Naziv SK**

Brisanje usluge

**Aktori SK**

Zaposleni

**Učesnici SK**

Zaposleni i sistem (program)

**Preduslov:** Sistem je uključen i zaposleni je prijavljen. Sistem prikazuje formu za rad sa uslugom.

**Osnovni scenario SK**

1. Zaposleni **unosi** kriterijum po kom pretražuje usluge. (APUSO)
2. Zaposleni **poziva** sistem da pronađe usluge na osnovu kriterijuma. (APSO)
3. Sistem **traži** uslugu. (SO)
4. Sistem **prikazuje** zaposlenom pronađene usluge. (IA)
5. Zaposleni **bira** uslugu koju želi da obriše. (APUSO)
6. Zaposleni **poziva** sistem da obriše odabranu uslugu. (APSO)
7. Sistem **briše** odabranu uslugu. (SO)
8. Sistem **prikazuje** zaposlenom odgovarajuću poruku: „Usluga je uspešno obrisana.“. (IA)

**Alternativna scenarija**

4.1. Ukoliko sistem nije uspeo da pronađe uslugu sa odgovarajućim kriterijumom, prikazuje se poruka zaposlenom: „Sistem nije uspeo da pronađe usluge koje odgovaraju zadatom kriterijumu.“. Prekida se izvršenje scenarija. (IA)

8.1. Ukoliko sistem nije uspeo da obriše odabranu uslugu, sistem prikazuje poruku zaposlenom: „Greška! Usluga nije obrisana.“. (IA)

### **Slučaj korišćenja – Zakazivanje termina (Složen SK)**

**Naziv SK**

Zakazivanje termina

**Aktori SK**

Zaposleni

**Učesnici SK**

Zaposleni i sistem (program)

**Preduslov:** Sistem je uključen i zaposleni je prijavljen. Učitan je spisak korisnika i usluga. Sistem prikazuje formu za rad sa terminima.

**Osnovni scenario SK**

1. Zaposleni **unosi** podatke o terminu. (APUSO)
2. Zaposleni **kontroliše** da li je korektno uneo podatke o terminu. (ANSO)
3. Zaposleni **poziva** sistem da zapamti podatke o terminu. (APSO)
4. Sistem **pamti** podatke o terminu. (SO)
5. Sistem **prikazuje** zaposlenom zapamćeni termin i poruku: “Sistem je zapamtio termin”. (IA)

**Alternativna scenarija**

5.1 Ukoliko sistem ne može da zapamti podatke o terminu on prikazuje

zaposlenom poruku “Sistem ne može da zapamti termin”. (IA)

### **Slučaj korišćenja – Pretraživanje termina**

**Naziv SK**

Pretraživanje termina

**Aktori SK**

Zaposleni

**Učesnici SK**

Zaposleni i sistem (program)

**Preduslov:** Sistem je uključen i zaposleni je prijavljen. Sistem prikazuje formu za rad sa terminom.

**Osnovni scenario SK**

1. Zaposleni **unosi** kriterijume po kojima pretražuje termine. (APUSO)
2. Zaposleni **poziva** sistem da pronađe termin na osnovu kriterijuma. (APSO)
3. Sistem **traži** termin. (S0)
4. Sistem **prikazuje** zaposlenom pronađene termine. (IA)
5. Zaposleni **bira** termin koji želi da pogleda. (APUSO)
6. Zaposleni **poziva** sistem da prikaže podatke o terminu. (APSO)
7. Sistem **traži** podatke o terminu. (SO)
8. Sistem **prikazuje** podatke o terminu i poruku: „Sistem je prikazao podatke o izabranom terminu“. (IA)

**Alternativna scenarija**

4.1. Ukoliko sistem nije uspeo da pronađe termine koji odgovaraju kriterijumu, zaposlenom se ispisuje odgovarajuća poruka: „Sistem nije uspeo da pronađe termine na osnovu kriterijuma!“. Prekida se izvršenje scenarija. (IA)

8.1. Ukoliko sistem nije uspeo da pronađe podatke o izabranom terminu, zaposlenom se ispisuje poruka: „Sistem nije uspeo da pronađe podatke o terminu.“ (IA)

### 3.2.3. Projektovanje kontrolera korisničkog interfejsa

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

Kontroler je odgovoran da prihvati zahtev za izvršenje sistemske operacije od klijenta i da ga prosledi do poslovne logike koja je odgovorna za izvršenje sistemske operacije.



Slika KKI Korisnički interfejs u kontekstu arhitekture softverskog sistema

## 3.3. Projektovanje aplikacione logike

### 3.3.1. Kontroler aplikacione logike

Kontroler aplikacione logike treba da podigne serverski soket koji će da osluškuje mrežu. Kada klijent (klijentski soket) uspostavi konekciju sa kontrolerom (serverskim soketom), tada kontroler treba da generiše nit koja će uspostaviti dvosmernu vezu sa klijentom (ulaznu i izlaznu). Slanje i primanje podataka od klijenta se ostvaruje preko soketa.

Klijent šalje zahtev za izvršenje neke od SO do odgovarajuće niti (koju smo nazvali “Obrada”), koja je povezana sa tim klijentom. “Obrada” prima zahtev i dalje ga preusmerava do klasa koje su odgovorne za izvršenje SO. Nakon izvršenja SO rezultat se vraća do aplikacione logike, odnosno do “Obrade”, koja taj rezultat šalje nazad do klijenta.

**Primer specificnog kontrolera aplikacione logike**

using System;

using Domen;

using System.Net.Sockets;

using System.Runtime.Serialization.Formatters.Binary;

using System.Threading;

using Server.SO.KorisnikSO;

using Server.SO.UslugaSO;

using Server.SO.ZaposleniSO;

using Server.SO.TerminSO;

namespace Server

{

class NitKlijenta

{

private NetworkStream tok;

BinaryFormatter formater;

public NitKlijenta(NetworkStream tok)

{

this.tok = tok;

formater = new BinaryFormatter();

ThreadStart ts = obradi;

new Thread(ts).Start();

}

void obradi()

{

try

{

int operacija = 0;

while (operacija != (int)Operacije.Kraj)

{

TransferKlasa transfer = formater.Deserialize(tok) as TransferKlasa;

switch (transfer.Operacija)

{

case Operacije.Login:

Login log = new Login();

transfer.Rezultat = log.IzvrsiSO(transfer.TransferObjekat as OpstiDomenskiObjekat);

formater.Serialize(tok, transfer);

break;

case Operacije.KreirajKorisnika:

KreirajKorisnika knd = new KreirajKorisnika();

transfer.Rezultat = knd.IzvrsiSO(transfer.TransferObjekat as OpstiDomenskiObjekat);

formater.Serialize(tok, transfer);

break;

case Operacije.ZapamtiKorisnika:

ZapamtiKorisnika zd = new ZapamtiKorisnika();

transfer.Rezultat = zd.IzvrsiSO(transfer.TransferObjekat as OpstiDomenskiObjekat);

formater.Serialize(tok, transfer);

break;

case Operacije.PretraziKorisnike:

PretraziKorisnike pke = new PretraziKorisnike();

transfer.Rezultat = pke.IzvrsiSO(transfer.TransferObjekat as OpstiDomenskiObjekat);

formater.Serialize(tok, transfer);

break;

case Operacije.PrikaziKorisnika:

PrikaziKorisnika pk = new PrikaziKorisnika();

transfer.Rezultat = pk.IzvrsiSO(transfer.TransferObjekat as OpstiDomenskiObjekat);

formater.Serialize(tok, transfer);

break;

case Operacije.ObrisiKorisnika:

ObrisiKorisnika ok = new ObrisiKorisnika();

transfer.Rezultat = ok.IzvrsiSO(transfer.TransferObjekat as OpstiDomenskiObjekat);

formater.Serialize(tok, transfer);

break;

case Operacije.KreirajUslugu:

KreirajUslugu knk = new KreirajUslugu();

transfer.Rezultat = knk.IzvrsiSO(transfer.TransferObjekat as OpstiDomenskiObjekat);

formater.Serialize(tok, transfer);

break;

case Operacije.ZapamtiUslugu:

ZapamtiUslugu zk = new ZapamtiUslugu();

transfer.Rezultat = zk.IzvrsiSO(transfer.TransferObjekat as OpstiDomenskiObjekat);

formater.Serialize(tok, transfer);

break;

case Operacije.PretraziUsluge:

PretraziUsluge pu = new PretraziUsluge();

transfer.Rezultat = pu.IzvrsiSO(transfer.TransferObjekat as OpstiDomenskiObjekat);

formater.Serialize(tok, transfer);

break;

case Operacije.PrikaziUslugu:

PrikaziUslugu uk = new PrikaziUslugu();

transfer.Rezultat = uk.IzvrsiSO(transfer.TransferObjekat as OpstiDomenskiObjekat);

formater.Serialize(tok, transfer);

break;

case Operacije.ObrisiUslugu:

ObrisiUslugu ou = new ObrisiUslugu();

transfer.Rezultat = ou.IzvrsiSO(transfer.TransferObjekat as OpstiDomenskiObjekat);

formater.Serialize(tok, transfer);

break;

case Operacije.VratiSveTipove:

VratiListuTipovaUsluga vsa = new VratiListuTipovaUsluga();

transfer.Rezultat = vsa.IzvrsiSO(transfer.TransferObjekat as OpstiDomenskiObjekat);

formater.Serialize(tok, transfer);

break;

case Operacije.VratiSveKorisnike:

VratiLIstuKorisnika vsd = new VratiLIstuKorisnika();

transfer.Rezultat = vsd.IzvrsiSO(transfer.TransferObjekat as OpstiDomenskiObjekat);

formater.Serialize(tok, transfer);

break;

case Operacije.VratiSveUsluge:

VratiLIstuUsluga vlu = new VratiLIstuUsluga();

transfer.Rezultat = vlu.IzvrsiSO(transfer.TransferObjekat as OpstiDomenskiObjekat);

formater.Serialize(tok, transfer);

break;

case Operacije.ZapamtiTermin:

ZapamtiTermin zr = new ZapamtiTermin();

transfer.Rezultat = zr.IzvrsiSO(transfer.TransferObjekat as OpstiDomenskiObjekat);

formater.Serialize(tok, transfer);

break;

case Operacije.PretraziTermine:

PretraziTermine pr = new PretraziTermine();

transfer.Rezultat = pr.IzvrsiSO(transfer.TransferObjekat as OpstiDomenskiObjekat);

formater.Serialize(tok, transfer);

break;

case Operacije.PrikaziTermin:

PrikaziTermin ur = new PrikaziTermin();

transfer.Rezultat = ur.IzvrsiSO(transfer.TransferObjekat as OpstiDomenskiObjekat);

formater.Serialize(tok, transfer);

break;

case Operacije.Kraj:

operacija = 1;

Server.listaTokova.Remove(tok);

break;

default:

break;

}

}

}

catch (Exception)

{

Server.listaTokova.Remove(tok);

}

}

}

}

### 3.3.2. Sistemske operacije

U prethodnoj fazi, fazi analize, odredili smo ugovore o sistemskim operacijama, pri čemu smo rekli da jedan ugovor opisuje ponašanje jedne sistemske operacije, tako što opisuje ŠTA operacija treba da radi, ali ne i kako. Sada se u fazi projektovanja za svaki od ugovora projektuje konceptualno rešenje (realizacija) SO. To znači da ćemo za svaku klasu odgovornu za izvršenje SO definisati KAKO će se sistemska operacija izvršiti.

Na samom početku projektovanja SO izbeći ćemo aspekte realizacije koji su vezani za konekciju sa bazom, perzistentnost i transakcije.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using Biblioteka;

using Sesija;

namespace SistemskeOperacije

{

public abstract class OpstaSO

{

public Object izvrsiSO(OpstiDomenskiObjekat odo)

{

Object rezultat = null;

Broker.dajSesiju().otvoriKonekciju();

Broker.dajSesiju().zapocniTransakciju();

try

{

rezultat = Izvrsi(odo);

Broker.dajSesiju().potvrdiTransakciju();

}

catch (Exception)

{

Broker.dajSesiju().ponistiTransakciju();

}

finally

{

Broker.dajSesiju().zatvoriKonekciju();

}

return rezultat;

}

public abstract object Izvrsi(OpstiDomenskiObjekat odo);

}

}

**1.Ugovor UG1: prijaviZaposlenog**

Operacija: prijaviZaspolenog(Zaposleni)

Veza sa SK: SK1

Preduslovi: /

Postuslovi: /



**2.Ugovor UG2: vratiListuKorisnika**

Operacija: vratiListuKorisnika(List<Korisnik>):signal;

Veza sa SK: SK4, SK9

Preduslovi: /

Postuslovi: /



**3.Ugovor UG3: vratiListuUsluga**

Operacija: vratiListuUsluga(List<Usluga>):signal;

Veza sa SK: SK9

Preduslovi: /

Postuslovi: /



**4.Ugovor UG4: kreirajKorisnika**

Operacija: kreirajKorisnika(Korisnik):signal;

Veza sa SK: SK2

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

Postuslovi: Kreiran je novi objekat odgovarajuće domenske klase.



**5.Ugovor UG5: zapamtiKorisnika**

Operacija: zapamtiKorisnika(Korisnik):signal;

Veza sa SK: SK2

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

Postuslovi: Podaci o korisniku su zapamćeni.



**6.Ugovor UG6: pretražiKorisnike**

Operacija: pretražiKorisnike(kriterijumPretrage,List<Korisnik>):signal;

Veza sa SK: SK3, SK4, SK5

Preduslovi: /

Postuslovi: /



**7.Ugovor UG7: prikažiKorisnika**

Operacija: prikažiKorisnika(Korisnik):signal;

Veza sa SK: SK3, SK4, SK5

Preduslovi: /

Postuslovi: /



**8.Ugovor UG8: izmeniKorisnika**

Operacija: izmeniKorisnika(Korisnik):signal;

Veza sa SK: SK4

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

Postuslovi: Objekat Korisnik je izmenjen.



**9.Ugovor UG9: izbrišiKorisnika**

Operacija: izbrišiKorisnika(Korisnik):signal;

Veza sa SK: SK5

Preduslovi: Strukturna ograničenja nad objektom Korisnik moraju biti zadovoljena.

Postuslovi: Objekat Korisnik je izbrisan.



**10. Ugovor UG10: kreirajUslugu**

Operacija: kreirajUslugu(Usluga):signal;

Veza sa SK: SK6

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

Postuslovi: Kreiran je novi objekat odgovarajuće domenske klase.



**11.Ugovor UG11: zapamtiUslugu**

Operacija: zapamtiUslugu(Usluga):signal;

Veza sa SK: SK6

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

Postuslovi: Podaci o usluzi su zapamćeni.



**12.Ugovor UG12: pretražiUsluge**

Operacija: pretražiUsluge(kriterijumPretrage,List<Usluga>):signal;

Veza sa SK: SK7, SK8

Preduslovi: /

Postuslovi: /



**13.Ugovor UG13: prikažiUslugu**

Operacija: prikažiUslugu(Usluga):signal;

Veza sa SK: SK7, SK8

Preduslovi: /

Postuslovi: /



**14.Ugovor UG14: izbrišiUslugu**

Operacija: izbrišiUslugu(Usluga):signal;

Veza sa SK: SK8

Preduslovi: Strukturna ograničenja nad objektom Usluga moraju biti zadovoljena.

Postuslovi: Objekat Usluga je izbrisan.



**15.Ugovor UG15: zapamtiTermin**

Operacija: zapamtiTermin(Termin):signal;

Veza sa SK: SK9

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

Postuslovi: Podaci o terminu su zapamćeni.



**16.Ugovor UG16: pretražiTermine**

Operacija: pretražiTermine(kriterijumPretrage,List<Termin>):signal;

Veza sa SK: SK10

Preduslovi: /

Postuslovi: /



**17.Ugovor UG17: prikažiTermin**

Operacija: prikažiTermin(Termin):signal;

Veza sa SK: SK10

Preduslovi: /

Postuslovi: /





Slika OSO: Klase koje su odgovorne za SO nasleđuju klasu OpstaSO

### 3.3.3. Domenske klase

Na osnovu konceptualnih klasa prave se softverske klase strukture. Svaka klasa sadrži privatna polja atributa, getter-e, setter-e za iste, konstruktore (bilo bezparametrizovani bilo parametrizovani).

Domenske klase implementiraju interfejs *BazniObjekat* kako bi omogućili lakše implementiranje metoda *DatabaseBroker*-a. Na taj način *DatabaseBroker* prima interfejs umesto samih klasa i olakšava kreiranje generičkih upita.

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Domen

{

public interface OpstiDomenskiObjekat

{

string nazivTabele { get; }

string primarniKljuc { get; }

string uslovPrimarni { get; }

string uslovOstalo { get; }

string izmena { get; }

string unos { get; }

OpstiDomenskiObjekat procitaj(DataRow red);

}

}

### 3.3.4 Broker baze podataka

Broker baze podataka je softverska klasa odgovorna za komunikacju između poslovne logike i skladišta podataka. Drugim rečima, projektuje se kako bi obezbedio perzistentni servis objektima domenskih klasa koji se čuvaju u bazi podataka. Klasa *DatabaseBroker* predstavlja perzistentni okvir koji posreduje u svim operacijama nad bazom podataka i realizuje sledeće metode:

public void OtvoriKonekciju();

public void ZatvoriKonekciju();

public void ZapocniTransakciju();

public void PonistiTransakciju();

public void PotvrdiTransakciju();

public List<OpstiDomenskiObjekat> dajSve(OpstiDomenskiObjekat odo);

public List<OpstiDomenskiObjekat> dajSveZaUslov(OpstiDomenskiObjekat odo);

public List<OpstiDomenskiObjekat> dajSveZaUslovVise(OpstiDomenskiObjekat odo);

public OpstiDomenskiObjekat dajZaUslov (OpstiDomenskiObjekat odo);

public OpstiDomenskiObjekat dajZaUslovVise(OpstiDomenskiObjekat odo);

public int azuriraj(OpstiDomenskiObjekat odo);

public int ubaci(OpstiDomenskiObjekat odo);

public int obrisi(OpstiDomenskiObjekat odo);

public int obrisiSveZaUslov(OpstiDomenskiObjekat odo);

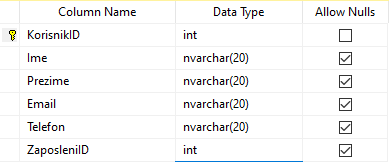
public int dajSifru(OpstiDomenskiObjekat odo);



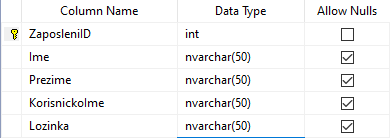
Slika DBBR: Database broker klasa se povezuje sa klasom OpstiDomenskiObjekat

### 3.3.5. Projektovanje skladišta podataka

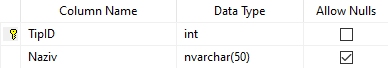
Na osnovu strukture softverskih klasa projektovane su tabele (skladišta podataka) relacionog sistema za upravljanje bazom podataka. U ovom radu je korišćen MS SQL.



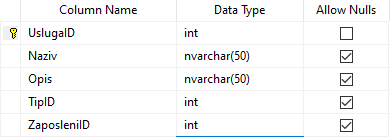
**KORISNIK**



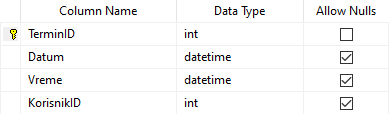
**ZAPOSLENI**



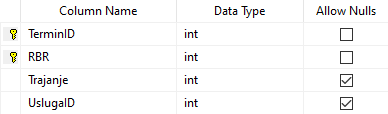
**TIP USLUGE**



**USLUGA**



**TERMIN**

 **STAVKA TERMINA**

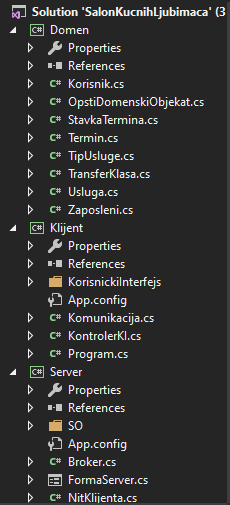


Slika ASSTBP: Arhitektura soft. sistema nakon projektovanja tabela baze podataka

# 4. IMPLEMENTACIJA

Softverski sistem, rezultat ovog rada, razvijen je u programskom jeziku C#, a izvorni kod je priložen na pratećem CD-u. Sistem je projektovan kao klijent-server. Kao razvojno okruženje korišćen je Microsoft Visual Studio 2017. Kao sistem za upravljanjem bazom podataka korišćen je MS SQL.

Na osnovu arhitekture softverskog sistema dobijene su sledeće softverske klase:



# 

# 5. TESTIRANJE

Svaki od implementiranih slučajeva korišćenja je testiran. Prilikom testiranja svakog slučaja korišćenja pored unetih pravilnih podataka unošeni su i nepravilni podaci da bi se utvrdilo kakav će rezultat izvršenja biti. Na osnovu izvršenih testiranja otklonjeni su uočeni nedostaci.

# 6. LITERATURA

Projektovanje softvera – Skripta, dr Siniša Vlajić, Beograd - 2020.