

**SEMINARSKI RAD**

Predmet: Projektovanje softvera

Tema: Softverski sistem za praćenje rada salona lepote

Profesor: Student:

Dr Siniša Vlajić Isidora Ivanović 0156/2017

Beograd, 2021.

**Sadržaj**

[**1.** **Korisnički zahtevi** 1](#_Toc81424529)

[**1.1. Verbalni opis** 1](#_Toc81424530)

[**1.2. Specifikacija zahteva pomoću modela slučajeva korišćenja** 1](#_Toc81424531)

[**1.2.1. Slučaj korišćenja – Prijavljivanje zaposlenog** 2](#_Toc81424532)

[**1.2.2. Slučaj korišćenja – Unos novog korisnika** 3](#_Toc81424533)

[**1.2.3. Slučaj korišćenja – Pretraga korisnika** 3](#_Toc81424534)

[**1.2.4. Slučaj korišćenja – Izmena postojećeg korisnika** 4](#_Toc81424535)

[**1.2.5. Slučaj korišćenja – Dodeljivanje usluga** 5](#_Toc81424536)

[**1.2.6. Slučaj korišćenja – Unos nove usluge** 6](#_Toc81424537)

[**1.2.7. Slučaj korišćenja – Pretraga usluge** 7](#_Toc81424538)

[**1.2.8. Slučaj korišćenja – Brisanje usluge** 8](#_Toc81424539)

[**1.2.9. Slučaj korišćenja – Zakazivanje termina (Složen SK)** 8](#_Toc81424540)

[**1.2.10. Slučaj korišćenja – Pretraživanje termina** 9](#_Toc81424541)

[**2.** **Analiza** 10](#_Toc81424542)

[**2.1.Ponašanje softverskog sistema – sistemski dijagrami sekvenci** 10](#_Toc81424543)

[**DS1: Dijagram sekvenci slučaja korišćenja - Prijavljivanje zaposlenog** 10](#_Toc81424544)

[**DS2: Dijagram sekvenci slučaja korišćenja - Unos novog korisnika** 11](#_Toc81424545)

[**DS3: Dijagram sekvenci slučaja korišćenja – Pretraga korisnika** 14](#_Toc81424546)

[**DS4: Dijagram sekvenci slučaja korišćenja – Izmena postojećeg korisnika** 16](#_Toc81424547)

[**DS5: Dijagram sekvenci slučaja korišćenja – Dodeljivanje usluga** 18](#_Toc81424548)

[**DS6: Dijagram sekvenci slučaja korišćenja – Unos nove usluge** 20](#_Toc81424549)

[**DS7: Dijagram sekvenci slučaja korišćenja – Pretraga usluge** 21](#_Toc81424550)

[**DS8: Dijagram sekvenci slučaja korišćenja – Brisanje usluge** 23](#_Toc81424551)

[**DS9: Dijagram sekvenci slučaja korišćenja – Zakazivanje termina (Složen SK)** 24](#_Toc81424552)

[**DS10: Dijagram sekvenci slučaja korišćenja – Pretraživanje termina** 25](#_Toc81424553)

[**2.2. Ponašanjesoftverskog sistema –Definisanje ugovora o sistemskim operacijama** 27](#_Toc81424554)

[**2.3. Struktura softverskog sistema- Konceptualni(domenski model)** 32](#_Toc81424555)

[**2.4. Struktura softverskog sistema – relacioni model** 33](#_Toc81424556)

[**3.Projektovanje** 37](#_Toc81424557)

[**3.1. Arhitektura softverskog sistema** 37](#_Toc81424558)

[**3.2. Projektovanje korisničkog interfejsa** 39](#_Toc81424559)

[**3.2.1. Projektovanje ekranskih formi** 39](#_Toc81424560)

[**3.2.1.1. Slučaj korišćenja - Prijavljivanje zaposlenog** 40](#_Toc81424561)

[**3.2.1.2. Slučaj korišćenja - Unos novog korisnika** 42](#_Toc81424562)

[**3.2.1.3. Slučaj korišćenja – Pretraga korisnika** 44](#_Toc81424563)

[**3.2.1.4.Slučaj korišćenja – Izmena postojećeg korisnika** 46](#_Toc81424564)

[**3.2.1.5. Slučaj korišćenja – Dodeljivanje usluga** 49](#_Toc81424565)

[**3.2.1.6. Slučaj korišćenja – Unos nove usluge** 53](#_Toc81424566)

[**3.2.1.7. Slučaj korišćenja – Pretraga usluge** 56](#_Toc81424567)

[**3.2.1.8. Slučaj korišćenja – Brisanje usluge** 59](#_Toc81424568)

[**3.2.1.9. Slučaj korišćenja – Zakazivanje termina (Složen SK)** 63](#_Toc81424569)

[**3.2.1.10. Slučaj korišćenja – Pretraživanje termina** 67](#_Toc81424570)

[**3.2.2. Projektovanje kontrolera korisničkog interfejsa** 70](#_Toc81424571)

[**3.3. Projektovanje aplikacione logike** 70](#_Toc81424572)

[**3.3.1. Kontroler aplikacione logike** 70](#_Toc81424573)

[**3.3.2. Poslovna logika** 71](#_Toc81424574)

[**3.3.3. Broker baze podataka** 90](#_Toc81424575)

[**3.4. Projektovanje skladišta podataka** 92](#_Toc81424576)

[**4.Implementacija** 94](#_Toc81424577)

[**5.Testiranje** 96](#_Toc81424578)

[**6.Literatura** 96](#_Toc81424579)

# **Korisnički zahtevi**

## **1.1. Verbalni opis**

Potrebno je kreirati softverski sistem koji će biti korišćen za vođenje pružanja usluga u kozmetičkom salonu (beauty salonu).

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.

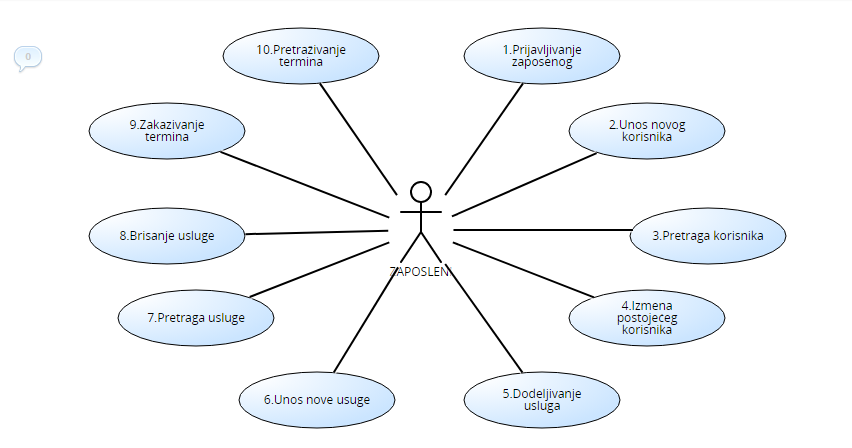
Potrebno je voditi evidenciju o uslugama koji postoje u salonu i omogućiti jednostavnu i brzu pretragu tih usluga po zadatim vrednostima. Korisnik usluga može u bilo kom trenutku da promeni vrstu usluge.

Korisnik sistema je zaposleni u salonu i on se mora prijaviti na sistem 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. Dodeljivanje usluga
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 – Dodeljivanje usluga**

**Naziv SK**

Dodeljivanje usluga

**Aktori SK**

Zaposleni

**Učesnici SK**

Zaposleni i sistem (program)

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

**Osnovni scenario SK**

1. Zaposleni **unosi** podatke o izvršenju usluga. (APUSO)
2. Zaposleni **kontroliše** da li je koretktno uneo podatke o izvršenju usluga.(ANSO)
3. Zaposleni **poziva** sistem da zapamti podatke o izvršenju usluga. (APSO)
4. Sistem **pamti** podatke o izvršenju usluga. (SO)
5. Sistem **prikazuje** zaposlenom zapamćeno izvršenje usluga i poruku : „Sistem je zapamtio izvršenje usluga“. (IA)

**Alternativna scenarija**

5.1. Ukoliko sistem ne može da zapamti podatke o izvršenju usluga on prikazuje

zaposlenom poruku “Sistem ne može da zapamti izvršenje usluga”. (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 zapamćenitermin i poruku: “Sistem je zapamtio termin”. (IA)

**Alternativna scenarija**

5.1 Ukoliko zaposleni unese istu uslugu za isti termin, sistem ne može da zapamti podatke o usluzi koja je uneta, on prikazuje zaposlenom poruku “Usluga je vec dodata u 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.

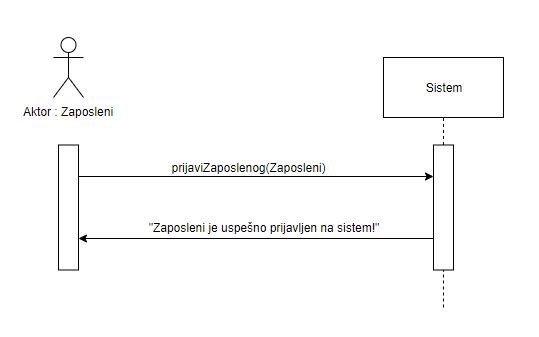
Ponašanje opisujemo pomoću sistemskih dijagrama sekvenci i ugovora o sistemskim operacijama.

Strukturu opisujemo pomoću konceptualnog modela i relacionog modela.

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

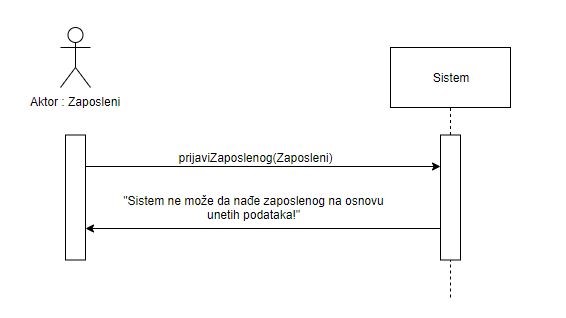
### **DS1: Dijagram sekvenci slučaja korišćenja - Prijavljivanje zaposlenog**

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)

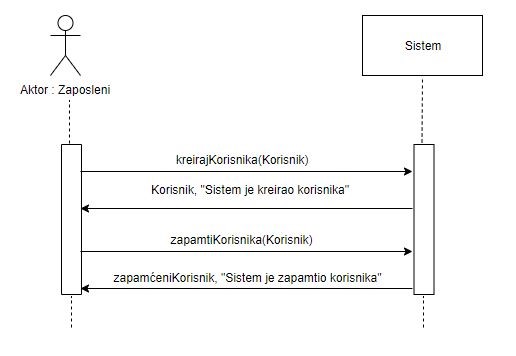


Sa navedenih sekvenci dijagrama se uočava 1 sistemska operacija:

1. Signal **prijaviZaposlenog(Zaposleni)**

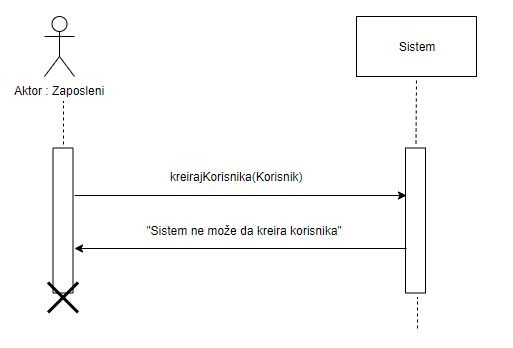
### **DS2: Dijagram sekvenci slučaja korišćenja - Unos novog korisnika**

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)

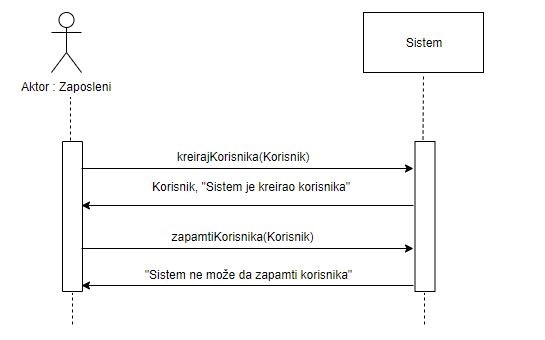


**Alternativna scenarija**

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“.

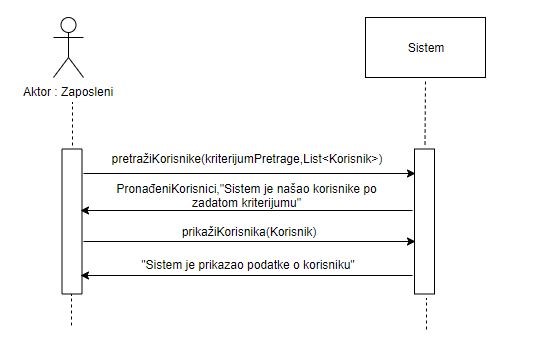


Sa navedenih sekvenci dijagrama se uočavaju 2 sistemske operacije:

1. Signal **kreirajKorisnika(Korisnik)**
2. Signal **zapamtiKorisnika(Korisnik)**

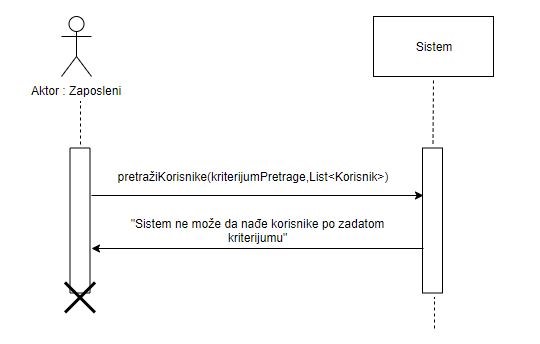
### **DS3: Dijagram sekvenci slučaja korišćenja – Pretraga korisnika**

1. Zaposleni poziva sistem da pronađe 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)

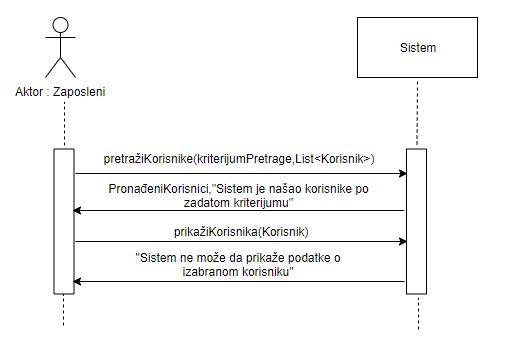


**Alternativna scenarija**

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

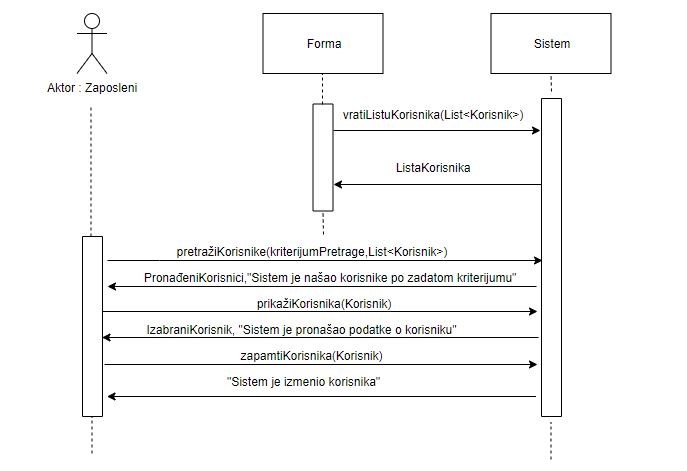


Sa navedenih sekvenci dijagrama se uočavaju 2 sistemske operacije:

1. Signal **pretražiKorisnike(zadatiKriterijum, List<Korisnik>)**
2. Signal **prikaziKorisnika(Korisnik)**

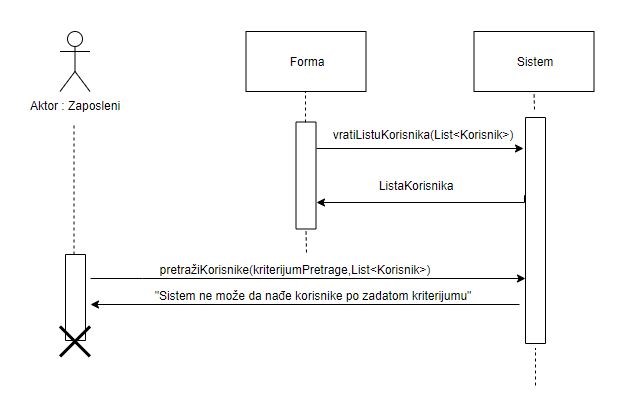
### **DS4: Dijagram sekvenci slučaja korišćenja – Izmena postojećeg korisnika**

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)

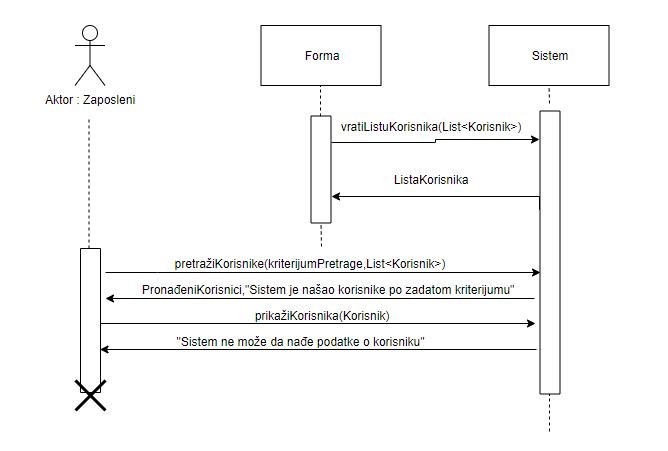


**Alternativna scenarija**

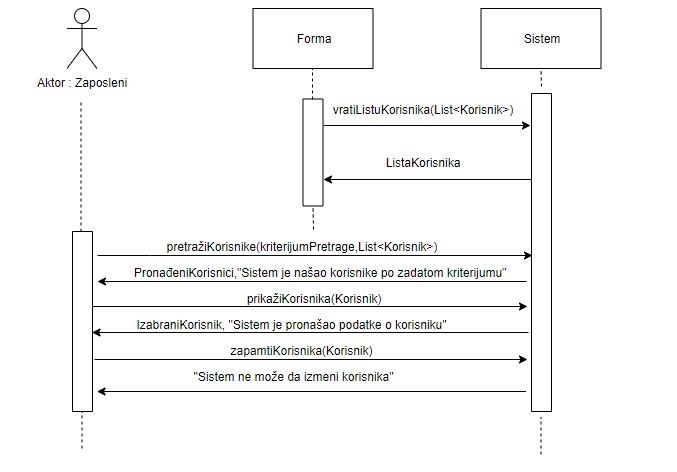
4.1. Ukoliko sistem ne može da nađe korisnike on prikazuje zaposlenom poruku: „Sistem ne može da nađe korisnike po zadatom kriterijumu“. Prekida se izvršenje scenarija. (IA)



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)

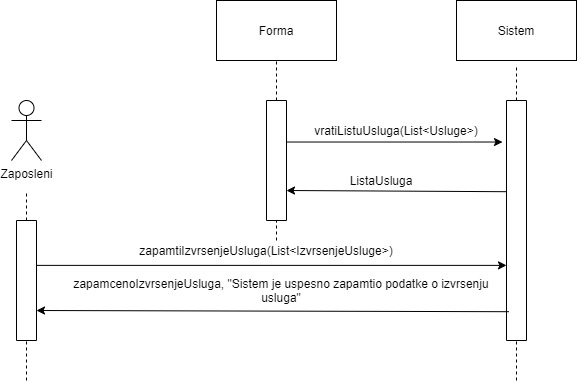


Sa navedenih sekvenci dijagrama se uočavaju 4 sistemske operacije:

1. Signal **vratiListuKorisnika(List<Korisnik>)**
2. Signal **pretražiKorisnike(kriterijumPretrage, List<Korisnik>)**
3. Signal **prikaziKorisnika(Korisnik)**
4. Signal **zapamtiKorisnika(Korisnik)**

### **DS5: Dijagram sekvenci slučaja korišćenja – Dodeljivanje usluga**

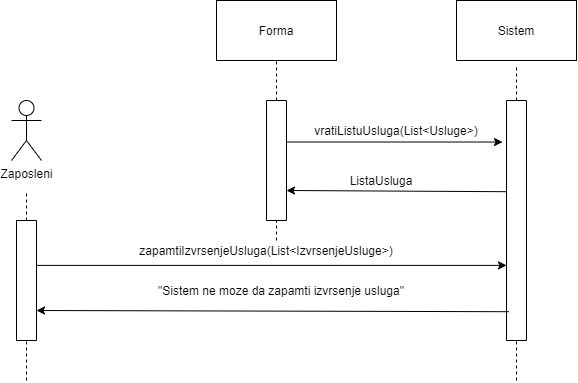
1. Forma poziva sistem da vrati listu usluga. (APSO)
2. Sistem vraća listu usluga. (IA)
3. Zaposleni poziva sistem da zapamti podatke o izvršavanju usluga. (APSO)
4. Sistem prikazuje zaposlenom zapamćeno izvršavanje usluga I poruku: “Sistem je uspešno zapamtio podatke o izvršenju usluga”.(IA)



**Alternativna scenarija**

2.1. Ukoliko sistem ne može da zapamti podatke o izvršenju usluge on prikazuje

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

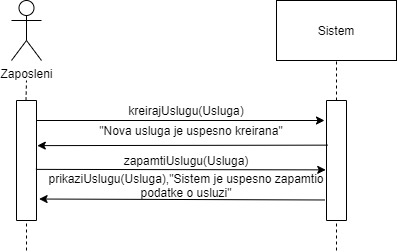


Sa nabedenih sekvencih dijagrama se uočava 1 sistemska operacija:

1. Signal **vratiListuUsluga(list<Usluge>)**
2. Signal **zapamtiIzvrsenjeUsluga(List<IzvrsenjeUsluge>)**

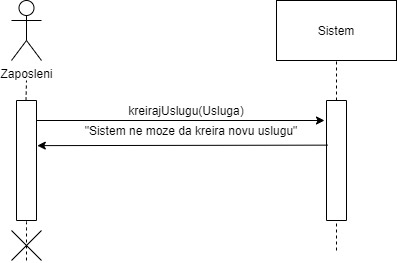
### **DS6: Dijagram sekvenci slučaja korišćenja – Unos nove usluge**

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)

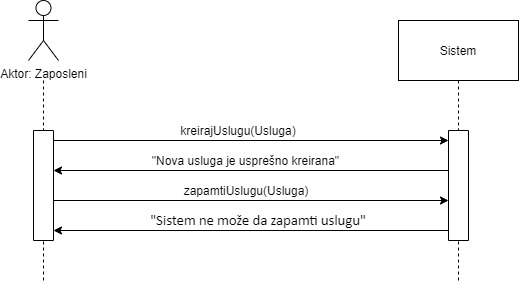


**Alternativna scenarija**

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“.

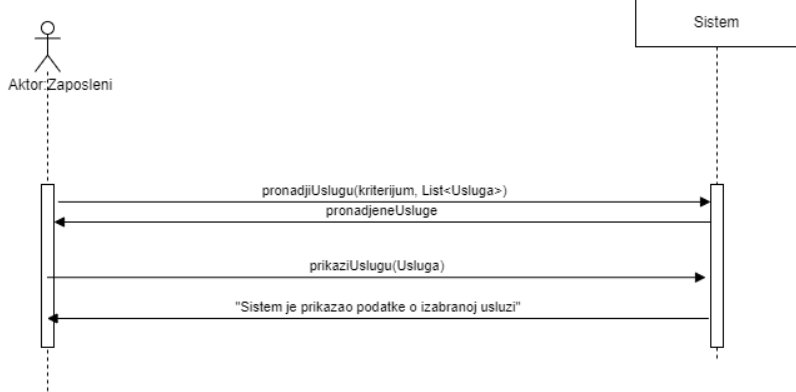


Sa navedenih sekvencnih dijagrama se uočavaju 2 sistemske operacije:

1. Signal **kreirajUslugu(Usluga)**
2. Signal **zapamtiUslugu(Usluga)**

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

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)

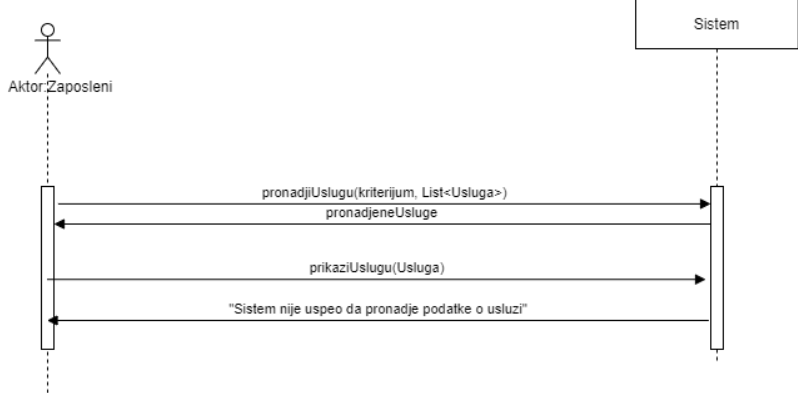


**Alternativna scenarija**

2.1.Ukoliko sistem ne može da nađe korisnike, on prikazuje zaposlenom poruku: „Sistem ne može da nađe korisnike po zadatom kriterijumu“. (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)

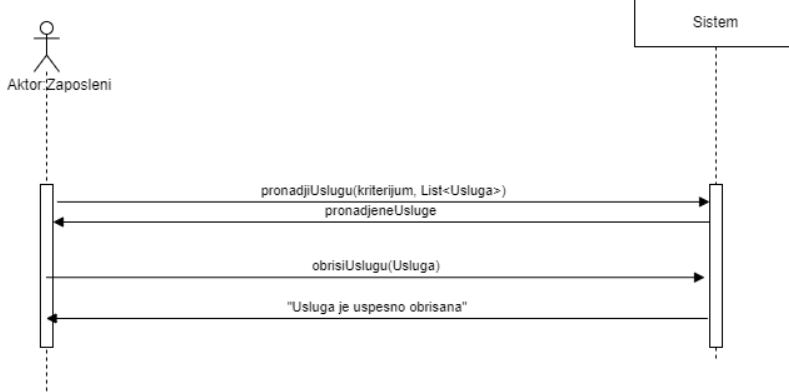


Sa navedenih sekvencnih dijagrama se uočavaju 2 sistemske operacije:

1. Signal **pronadjiUsluge(kriterijum, List<Usluga>)**
2. Signal **prikaziUslugu(Usluga)**

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

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

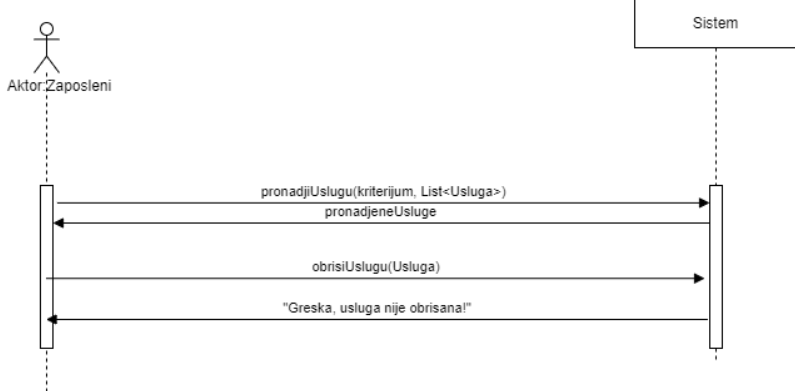


**Alternativna scenarija**

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



4.1. Ukoliko sistem ne može da izbriše uslugu on prikazuje zaposlenom poruku: „Greška, usluga nije obrisana!“. (IA)

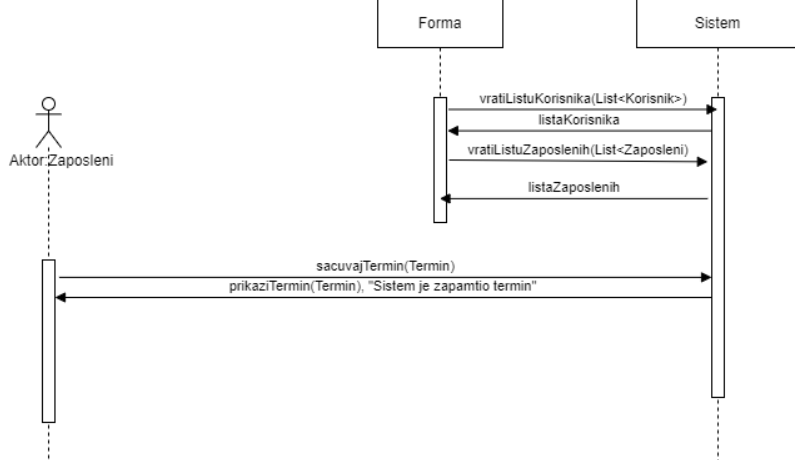


Sa navedenih sekvencnih dijagrama uočavaju se 4 sistemske operacije:

1. Signal **pronadjiUslugu(kriterijum, List<Usluga>)**
2. Signal **obrisiUslugu(Usluga)**

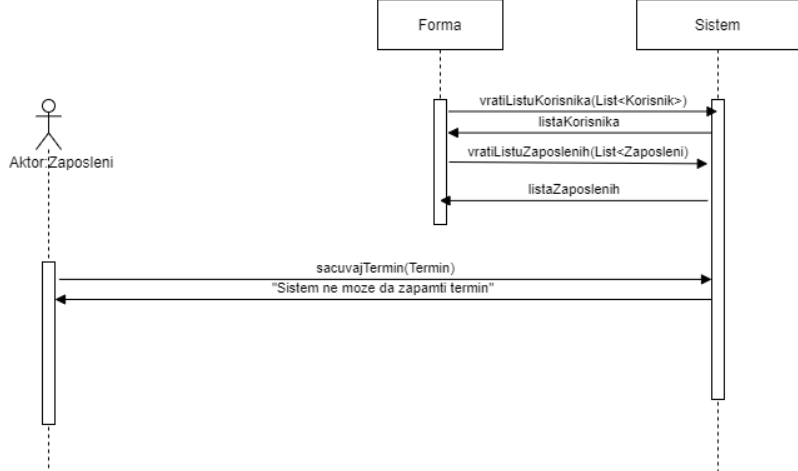
### **DS9: Dijagram sekvenci slučaja korišćenja – Zakazivanje termina (Složen SK)**

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 zapamćeni termin I poruku: “Sistem je zapamtio termin”.(IA)



**Alternativna scenarija**

6.1. Ukoliko sistem ne može da zapamti podatke o terminu on prikazuje zaposlenom poruku: “Sistem ne može da zapamti termin”. (IA)

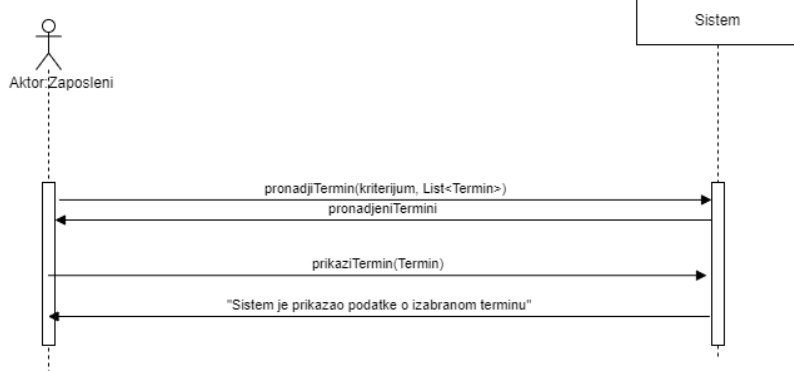


Sa navedenih sekvencnih dijagrama se uocavaju 3 sistemske operacije:

1. Signal **vratiListuKorisnika(List<Korisnik>)**
2. Signal **vratiListuZaposlenih(List<Zaposleni>)**
3. Signal **sacuvajTermin(Termin)**

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

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

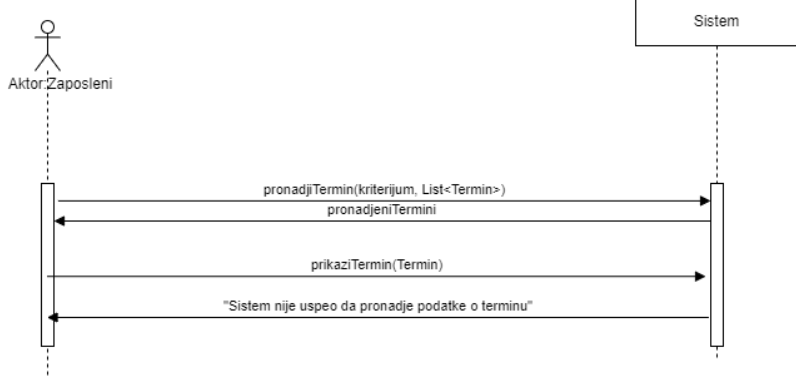


**Alternativna scenarija**

2.1.Ukoliko sistem ne može da nađe termine, on prikazuje zaposlenom poruku: „Sistem nije uspeo da pronađe termine na osnovu kriterijuma“. (IA)



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



Sa navedenih sekvencnih dijagrama se uočavaju 4 sistemske operacije:

1. Signal **pronadjiTermin(kriterijum, List<Termin>)**
2. Signal **prikaziTermin(Termin)**

Na osnovu analize scenarija dobijeno je 18 sistemskih operacija:

1. Signal **prijaviiZaposlenog(Zaposleni);**
2. Signal **kreirajKorisnika(Korisnik);**
3. Signal **pretraziKorisnike(zadatiKriterijum, List<Korisnik>);**
4. Signal **prikaziKorisnika(Korisnik);**
5. Signal **zapamtiKorisnika(Korisnik);**
6. Signal **pretraziTermine(kriterijum, List<Termin>);**
7. Signal **prikaziTermin(Termin);**
8. Signal **vratiListuKorisnika(List<Korisnik>);**
9. Signal **vratiListuUsluga(List<Usluga>)**
10. Signal **zapamtiTermin(Termin);**
11. Signal **kreirajUslugu(Usluga);**
12. Signal **obrisiUslugu(Usluga);**
13. Signal **pretraziUsluge(kriterijum, List<Usluga>);**
14. Signal **prikaziUslugu(Usluga);**
15. Signal **vratiListuKategorija(List<Kategorija>);**
16. Signal **vratiListuTipovaUsluga(List<TipUsluge>);**
17. Signal **vratiListuZaposlenih(List<Zaposleni>);**
18. Signal **zapamtiUslugu(Usluga);**

## **2.2. Ponašanjesoftverskog sistema –Definisanje ugovora o sistemskim operacijama**

Ugovor UG1: **prijaviZaposlenog(Zaposleni) Signal;**

Veza sa SK: SK1

Preduslovi:

Postuslovi:

Ugovor UG2: **kreirajKorisnika(Korisnik) Signal;**

Veza sa SK: SK2

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

Postuslovi: Podaci o korisniku su zapamćeni.

Ugovor UG3: **zapamtiKorisnika(Korisnik) Signal;**

Veza sa SK: SK4

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

Postuslovi: Podaci o korisniku su zapamćeni.

Ugovor UG4: **pretraziKorisnike(zadatiKriterijum, List<Korisnik>) Signal;**

Veza sa SK: SK3, SK4

Preduslovi:

Postuslovi:

Ugovor UG5: **prikaziKorisnika(Korisnik) Signal;**

Veza sa SK: SK3, SK4

Preduslovi:

Postuslovi:

Ugovor UG6: **vratiListuKorisnika(List<Korisnik>) Signal;**

Veza sa SK: SK4, SK9

Preduslovi:

Postuslovi:

Ugovor UG7: **vratiListuZaposlenih(List<Zaposleni>) Signal;**

Veza sa SK: SK9

Preduslovi:

Postuslovi:

Ugovor UG8: **vratiListuUsluga(List<Usluge>) Signal;**

Veza sa SK: SK5

Preduslovi:

Postuslovi:

Ugovor UG9: **vratiListuKategorija(List<Kategorija>) Signal;**

Veza sa SK: SK5

Preduslovi:

Postuslovi:

Ugovor UG10: **vratiListuTipovaUsluga(List<TipUsluge>) Signal;**

Veza sa SK: SK5

Preduslovi:

Postuslovi:

Ugovor UG11: **kreirajUslugu(Usluga) Signal;**

Veza sa SK: SK6

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

Postuslovi: Usluga je kreirana.

Ugovor UG12: **zapamtiUslugu(Usluga) Signal;**

Veza sa SK: SK5, SK6

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

Postuslovi: Usluga je sačuvana.

Ugovor UG13: **pretraziUsluge(kriterijum, List<Usluga>) Signal;**

Veza sa SK: SK7, SK8

Preduslovi:

Postuslovi:

Ugovor UG14: **prikaziUslugu(Usluga) Signal**;

Veza sa SK: SK7

Preduslovi:

Postuslovi:

Ugovor UG15: **obrisiUslugu(Usluga) Signal**;

Veza sa SK: SK8

Preduslovi: Strukturna ograničenja nad objektom Usluga moraju biti zadovoljena.

Postuslovi: Usluga je obrisana.

Ugovor UG16: **zapamtiTermin(Termin) Signal**;

Veza sa SK: SK9

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

Postuslovi: Termin je sačuvan.

Ugovor UG17: **pretraziTermine(kriterijum, List<Termin>) Signal**;

Veza sa SK: SK10

Preduslovi:

Postuslovi:

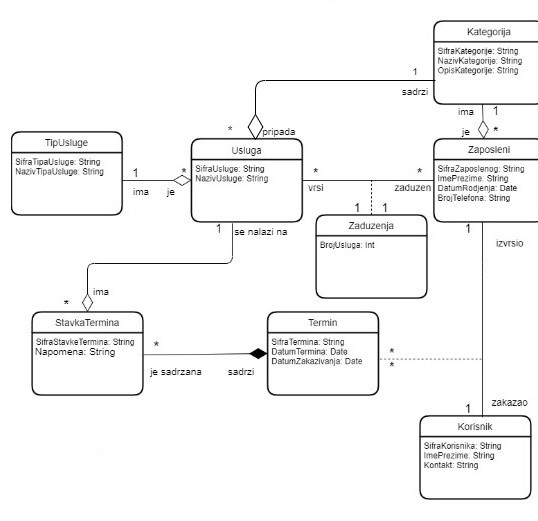
Ugovor UG18: **prikaziTermin(Termin) Signal;**

Veza sa SK: SK10

Preduslovi:

Postuslovi:

## **2.3. Struktura softverskog sistema- Konceptualni(domenski model)**



## **2.4. Struktura softverskog sistema – relacioni model**

**TipUsluge**(SifraTipaUsluge, NazivTipaUsluge)

**Usluga**(SifraUsluge, NazivUsluge, OpisUsluge, CenaUsluge, Sifra*TipaUsluge, sifraKategorije*)

**Korisnik**(SifraKorisnika, ImePrezime, Kontakt)

**Zaposleni**(SifraZaposlenog, ImePrezime, DatumRodjenja, BrojTelefona, *sifraKategorije*)

**Kategorija**(sifraKategorije, nazivKategorije, opisKategorije)

**Zaduzenja**(SifraUsluge, SifraZaposlenog, BrojUsluga)

**Termin**(SifraZaposlenog, SifraKorisnika, SifraTermina, DatumZakazivanja, DatumTermina*)*

**StavkaTermin**a(SifraTermina, SifraStavkeTermina, *sifraUsluge*)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela TipUsluge | | Prosto vrednosno  ogranicenje | | Slozeno vrednsono  ogranicenje | | Strukturno ogranicenje |
| Atributi | Ime | Tip  atributa | Vrednost atributa | Medju-zavisnost  atributa  jedne  tabele | Medju-zavisnost  atributa vise  tabela | INSERT /  UPDATE CASCADES Usluga  DELETE RESTRICTED  Usluga |
| SifraTipaUsluge | String | not null |  |  |
| NazivTipaUsluge | String | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela Usluga | | Prosto vrednosno ogranicenje | | Slozeno vrednsono ogranicenje | | Strukturno ogranicenje |
| Atributi | Ime | Tip atributa | Vrednost  atributa | Medju-zavisnost  atributa  jedne  tabele | Medju-zavisnost  atributa  vise tabela | INSERT  RESTRICTED TipUsluge,  Kategorija  UPDATE  RESTRICTED TipUsluge, Kategorija  CASCADES  StavkaTermina,  Zaduzenja  DELETE  RESTRICTED  Zaduzenja,  StavkaTermina |
| SifraUsluge | String | not null |  |  |
| NazivUsluge | String | not null |  |  |
| OpisUsluge | String | not null |  |  |
| CenaUsluge | Int | >0 |  |  |
| SifraTipUsluge | String | not null |  |  |
| SifraKategorije | String | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela Zaposleni | | Prosto vrednosno ogranicenje | | Slozeno vrednsono ogranicenje | | Strukturno ogranicenje |
| Atributi | Ime | Tip atributa | Vrednost  atributa | Medju-zavisnost  atributa  jedne  tabele | Medju-zavisnost  atributa  vise tabela | INSERT /  UPDATE CASCADES  Zaduzenja,  Termin  DELETE  RESTRICTED  Zaduzenja,  Termin |
| SifraZaposlenog | String | not null |  |  |
| ImePrezime | String | not null |  |  |
| BrojTelefona | String | not null |  |  |
| DatumRodjenja | Date |  |  |  |
| SifraKategorije | String |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela Korisnik | | Prosto vrednosno ogranicenje | | Slozeno vrednsono ogranicenje | | Strukturno ogranicenje |
| Atributi | Ime | Tip atributa | Vrednost  atributa | Medju-  Zavisnost  Atributa  jedne  tabele | Medju-zavisnost  atributa  vise tabela | INSERT /  UPDATE  RESTRICTED Termin  DELETE  RESTRICTED Termin |
| SifraKorisnika | String | not null |  |  |
| ImePrezime | String | not null |  |  |
| Kontakt | String | not null |  |  |

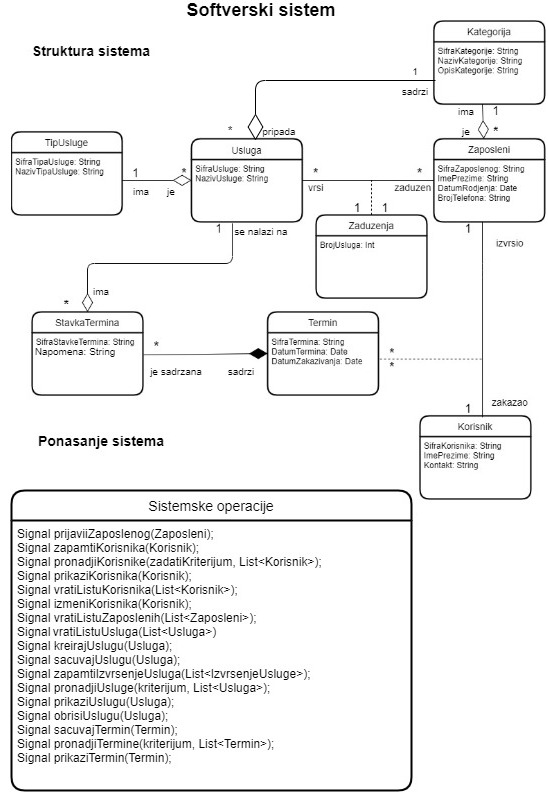
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela Zaduzenja | | Prosto vrednosno ogranicenje | | Slozeno vrednsono ogranicenje | | Strukturno ogranicenje |
| Atributi | Ime | Tip atributa | Vrednost atributa | Medju-  zavisnost  atributa  Jedne  tabele | Medju-  zavisnost  atributa  vise tabela | INSERT  RESTRICTED Usluga,  Zaposleni  UPDATE  RESTRICTED Usluga,  Zaposleni  DELETE  / |
| SifraUsluge | String | not null |  |  |
| SifraZaposlenog | String | not null |  |  |
| BrojUsluga | Int | >0  (default:0) |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela Termin | | Prosto vrednosno ogranicenje | | Slozeno vrednsono ogranicenje | | Strukturno ogranicenje |
| Atributi | Ime | Tip atributa | Vrednost  atributa | Medju-  zavisnost  atributa  jedne  tabele | Medju-zavisnost  atributa  vise tabela | INSERT  RESTRICTED Zaposleni,  Korisnik  UPDATE  RESTRICTED Zaposleni, Korisnik  CASCADES StavkaTermina  DELETE  RESTRICTED  StavkaTermina |
| SifraZaposlenog | String | not null |  |  |
| SifraKorisnika | String | not null |  |  |
| SifraTermina | String | not null |  |  |
| DatumZakazivanja | Date | not null |  |  |
| DatumTermina | Date | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela StavkaTermina | | Prosto vrednosno ogranicenje | | Slozeno vrednsono ogranicenje | | Strukturno ogranicenje |
| Atributi | Ime | Tip atributa | Vrednost  atributa | Medju-  zavisnost  Atributa  jedne  tabele | Medju-  zavisnost  atributa  vise tabela | INSERT  RESTRICTED Termin,  Usluga  UPDATE  RESTRICTED Termin, Usluga  DELETE  / |
| SifraTermina | String | not null |  |  |
| SifraStavkeTermina | String | not null |  |  |
| SifraUsluge | String | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela Kategorija | | Prosto vrednosno  ogranicenje | | Slozeno vrednsono  ogranicenje | | Strukturno ogranicenje |
| Atributi | Ime | Tip  atributa | Vrednost atributa | Medju-zavisnost  atributa  jedne  tabele | Medju-zavisnost  atributa vise  tabela | INSERT /  UPDATE CASCADES Zaposleni,  Usluga  DELETE RESTRICTED  Zaposleni,  Usluga |
| SifraKategorije | String | not null |  |  |
| NazivKategorije | String | not null |  |  |
| OpisKategorije | String | not null |  |  |

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



# **3.Projektovanje**

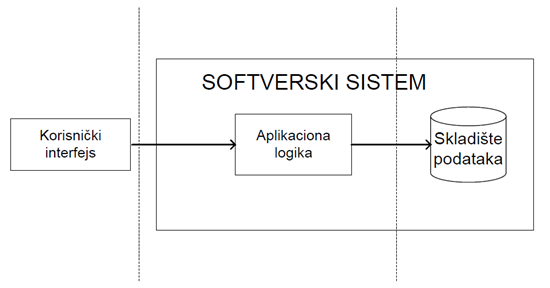
Faza projektovanja opisuje fizičku strukturu i ponašanje softverskog sistema (arhitekturu softverskog sistema).

## **3.1. Arhitektura softverskog sistema**

Arhitektura sistema je trinivojska i sastoji se od sledećih nivoa:

* korisnički interfejs
* aplikaciona logika
* skladište podataka

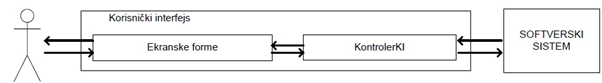
Nivo korisničkog interfejsa je na strani klijenta, dok su aplikaciona logika i skladište podataka na strani softvera.



Trinivojska arhitektura

## **3.2. Projektovanje korisničkog interfejsa**

Korisnički interfejs predstavlja realizaciju ulaza i,ili izlaza softverskog sistema I sastoji se od ekranske forme I kontrolera korisničkog interfejsa.

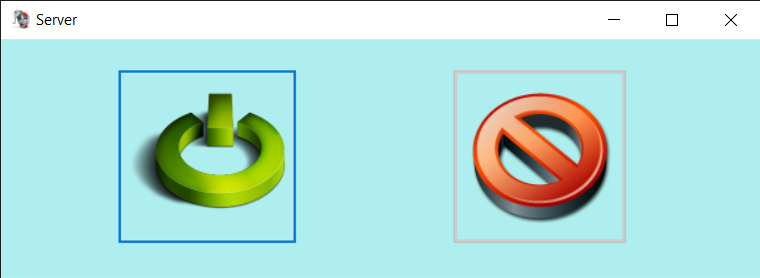


Struktura korisničkog interfejsa

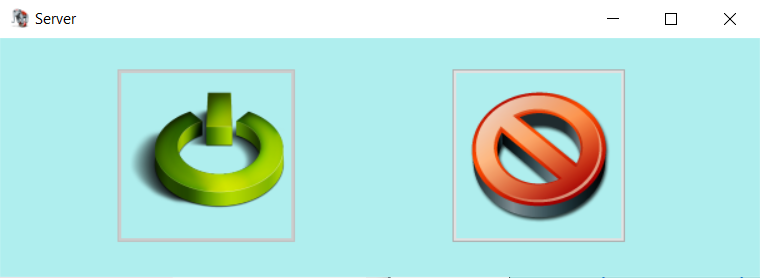
## **3.2.1. Projektovanje ekranskih formi**

Korisnički interfejs je definisan preko skupa ekranskih formi. Scenarija korišćenja ekranskih formi su direktno povezani sa scenarijima slučajeva korišćenja. Ekranska forma ima ulogu da prihvati podatke koje unosi aktor, prihvata događaje koje pravi aktor, poziva kontrolera korisničkog interfejsa kako bi mu prosledio te podatke i prikazuje podatke dobijene od kontrolera korisničkog interfejsa.

Na serverskoj strani programa projektovana je korisnička forma koja izgleda pre aktivacije ovako:



Nakon aktivacije, serverska forma izgleda ovako:



### **3.2.1.1. Slučaj korišćenja - Prijavljivanje zaposlenog**

Naziv SK

Prijavljivanje zaposlenog

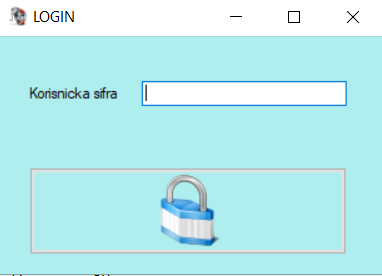
Aktori SK

Zaposleni

Učesnici SK

Zaposleni i sistem (program)

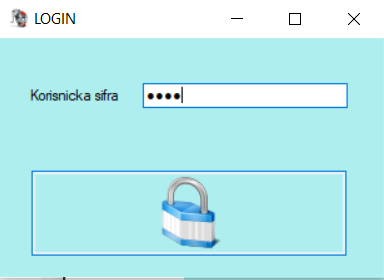
Preduslov: Sistem je uključen i prikazuje formu za prijavljivanje zaposlenog.



Osnovni scenario SK

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

*Opis akcije: Zaposleni unosi korisničku šifru u polje.*



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

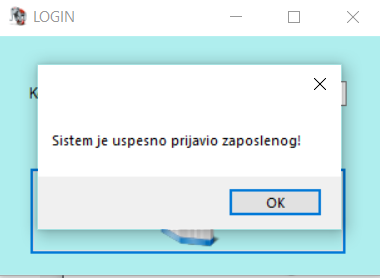
3. Zaposleni poziva sistem da ga pronađe na osnovu unetih podataka. (APSO)

*Opis akcije: Zaposleni klikom na dugme Uloguj se poziva sistemsku operaciju*

*PretražiZaposleneSO(Zaposleni) koja traži zaposlenog.*

4. Sistem pretražujezaposlene. (SO)

5.Sistem prikazuje zaposlenom poruku na ekranu: „Sistem je uspesno prijavio zaposlenog!“ 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)

### **3.2.1.2. Slučaj korišćenja - Unos novog korisnika**

**Naziv SK**

Unos novog korisnika

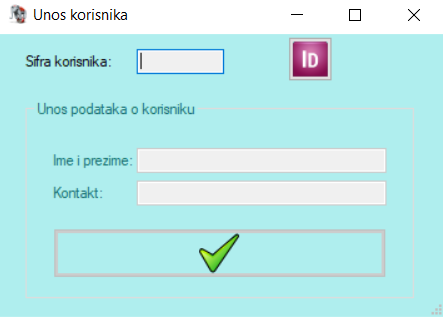
**Aktori SK**

Zaposleni

**Učesnici SK**

Zaposleni i sistem (program)

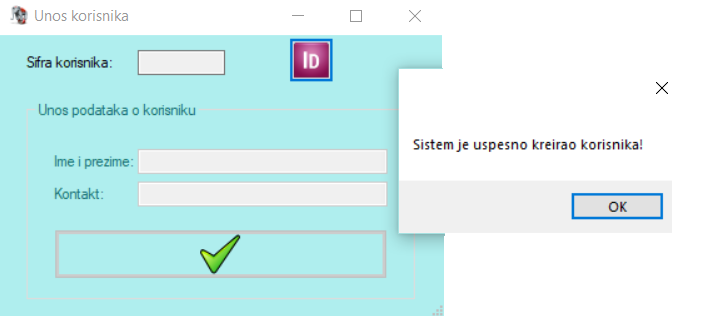
**Preduslov:** Sistem je uključen i zaposleni je prijavljen. Sistem prikazuje formu za unos novog korisnika.



**Osnovni scenario SK**

1. Zaposleni **poziva**sistem da kreira novog korisnika. (APSO)

*Opis akcije: Klikom na dugme ID sistem kreira novog korisnika.*



1. Sistem **kreira** korisnika. (SO)

*Opis akcije: Zaposleni klikom na dugme poziva sistemsku operacijy KreirajKorisnika(Korisnik) koja pamti novog korisnika u bazu korisnika.*

1. Sistem **prikazuje**zaposlenom poruku: „Sistem je uspesno kreirao korisnika“ (IA)
2. Zaposleni **unosi** podatke za kreiranog korisnika. (ANSO)
3. Zaposleni **poziva**sistem da sačuva podatke o novom korisniku. (APSO)
4. Sistem **pamti** podatke o korisniku. (SO)
5. 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)

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

**Naziv SK**

Pretraga korisnika

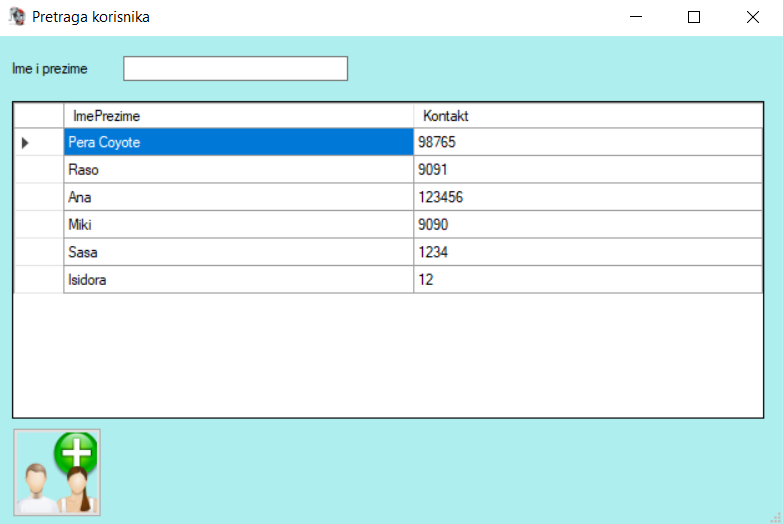
**Aktori SK**

Zaposleni

**Učesnici SK**

Zaposleni i sistem (program)

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



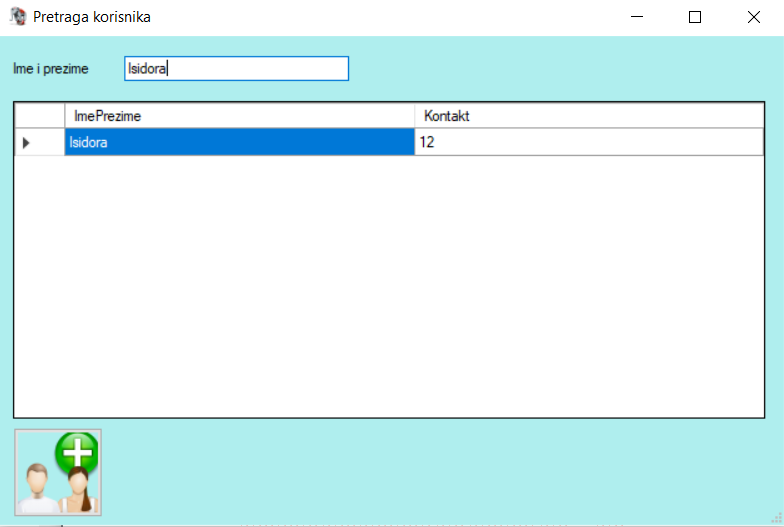
**Osnovni scenario SK**

1. Zaposleni **unosi** kriterijum u formu za pretraživanje korisnika. (APUSO)

*Opis akcije: Zaposleni unosi Ime I prezime korisnika kojeg želi da pronadje.*

1. Zaposleni **poziva**sistem da pronađe korisnika na osnovu zadatog kriterijuma. (APSO)
2. Sistem **traži** korisnika. (SO)
3. Sistem **prikazuje** pronađene korisnike. (IA)

*Opis akcije: Zaposleni klikom na dugme Pronadji korisnike prema zadatom kriterijumu poziva sistemsku operaciju PronadjiKorisnikeSO(zadatiKriterijum, List<Korisnik>) koja pronalazi korisnike po unetom kriterijumu.*



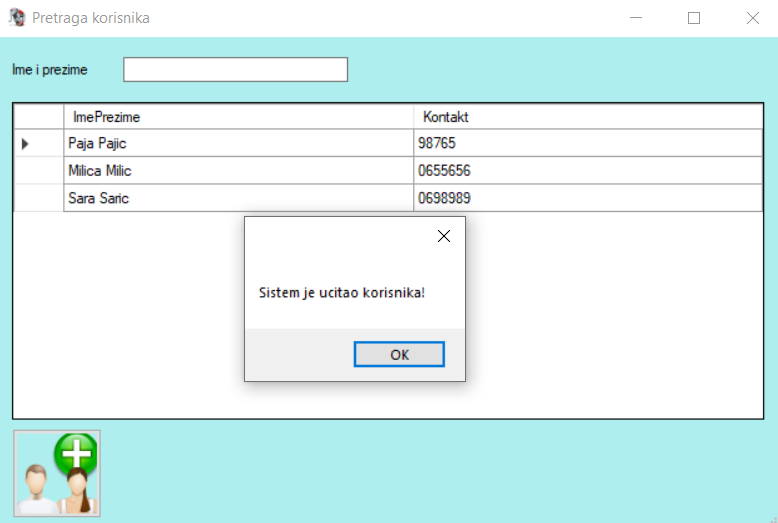
1. Zaposleni **bira** korisnika čije podatke želi da vidi. (APUSO)

*Opis akcije: Zaposleni bira iz tabele željenog korisnika.*

1. Zaposleni **poziva**sistem da prikaže podatke o izabranom korisniku. (APSO)

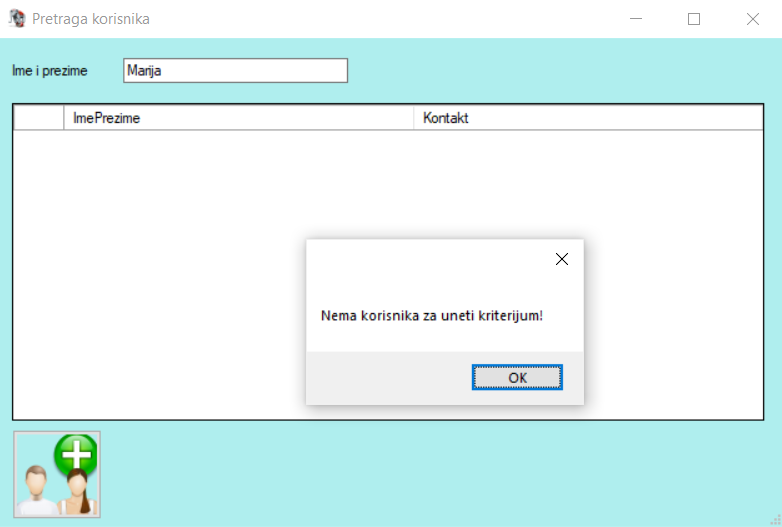
*Opis akcije: Zaposleni klikom na dugme poziva sistemsku operaciju PrikaziKorisnika(Korisnik ) koja prikazuje podatke o izabranom korisniku.*

1. Sistem **pronalazi** podatke o izabranom korisniku. (SO)
2. Sistem **prikazuje** podatke o izabranom korisniku i poruku: „Sistem je ucitao korisnika“. (IA)



**Alternativna scenarija**

4.1. Ukoliko sistem nije uspeo da pronađe korisnike, prikazuje se poruka zaposlenom: „Nema korisnika za uneti kriterijum .“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)

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

**Naziv SK**

Izmena postojećeg korisnika

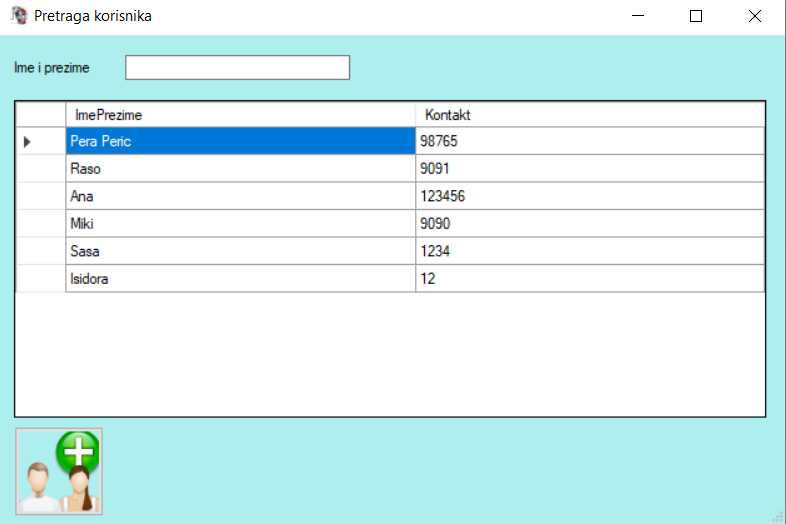
**Aktori SK**

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**

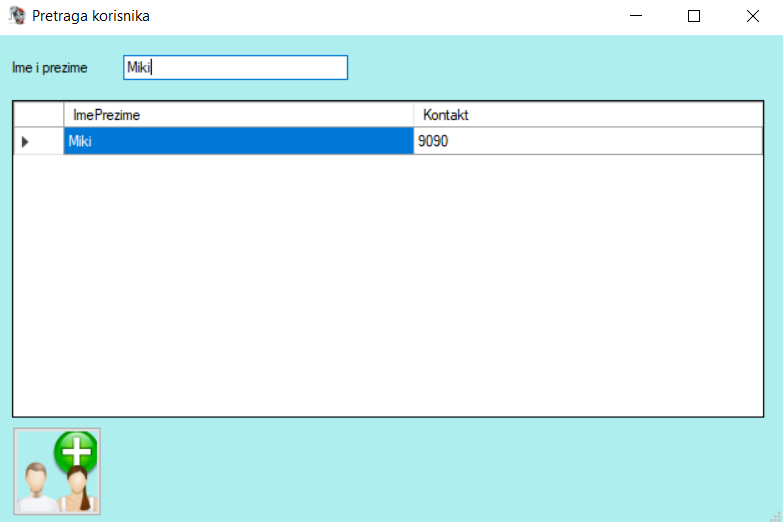
1. Zaposleni **unosi** kriterijum po kom pretražuje korisnike. (APUSO)

*Opis akcije: Zaposleni unosi u polje za pretragu ime i prezime korisnika.*

1. Zaposleni **poziva**sistem da pronađe korisnike na osnovu zadatog kriterijuma. (APSO)

*Opis akcije: Zaposleni klikom na dugme prema zadatom kriterijumu poziva sistemsku operaciju PronadjiKorisnike(zadatiKriterijum, List<Korisnik>) koja pronalazi korisnike po unetom kriterijumu.*

1. Sistem **traži** korisnika. (SO)
2. Sistem **prikazuje**zaposlenom pronađene korisnike i poruku: “Sistem je našao korisnike prema zadatom kriterijumu“. (IA)

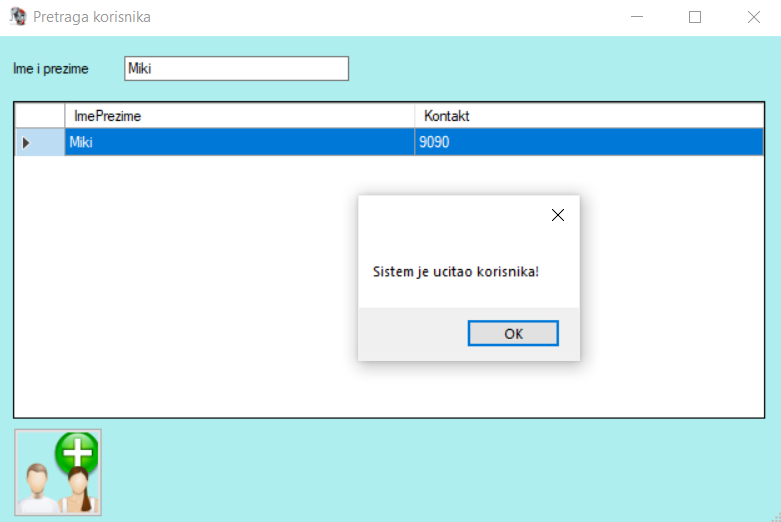


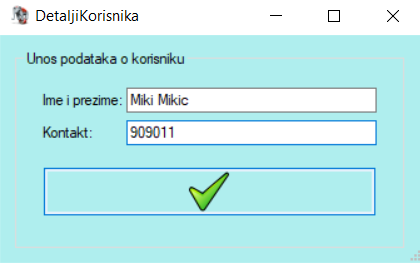
1. Zaposleni **bira** korisnika kog želi da izmeni. (APUSO)

*Opis akcije: Zaposleni bira željenog korisnika iz tabele.*

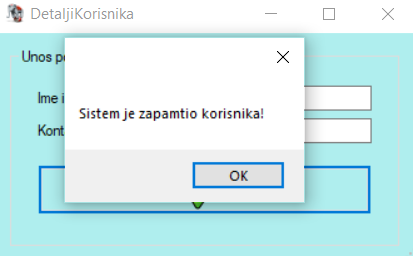
1. Zaposleni **poziva** sistem da učita podatke o izabranom korisniku. (APSO)

*Opis akcije: Zaposleni klikom na dugme poziva sistemsku operaciju PrikaziKorisnika(Korisnik ) koja prikazuje podatke o izabranom korisniku.*

1. Sistem **pronalazi** podatke o izabranom korisniku. (SO)
2. Sistem **prikazuje**zaposlenom podatke o izabranom korisniku i poruku: „Sistem je učitaokorisnika!.“(IA)
3. Zaposleni **unosi** nove vrednosti za odabranog korisnika. (APUSO)
4. Zaposleni **poziva**sistem da izmeni izabranog korisnika. (APSO)

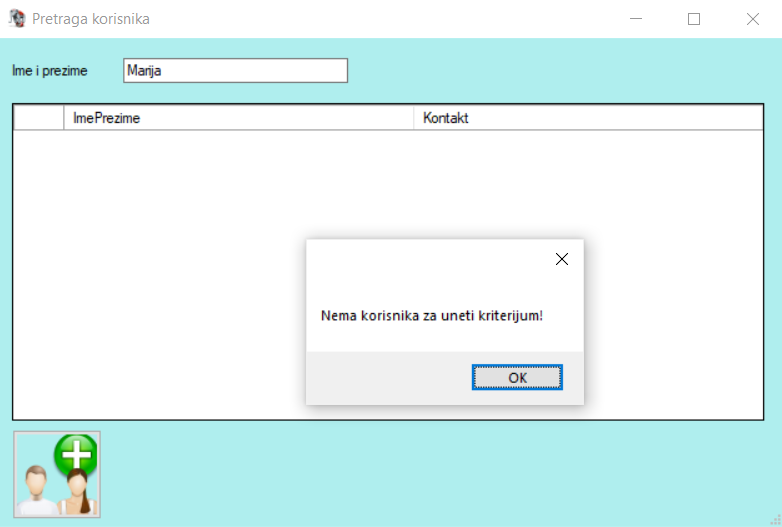


1. Sistem **pamti** izmenjene podatke o korisniku. (SO)
2. Sistem **prikazuje**izmenjene podatke o korisniku i poruku: „Sistem je uspešno izmenio podatke o korisniku“. (IA)



**Alternativna scenarija**

4.1. Ukoliko sistem nije uspeo da pronađe korisnike koji odgovaraju zadatom kriterijumu, zaposlenom se ispisuje odgovarajuća poruka: „Nema korisnika za uneti kriterujum“.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)

### **3.2.1.5. Slučaj korišćenja – Dodeljivanje usluga**

Naziv SK

Dodeljivanje usluga

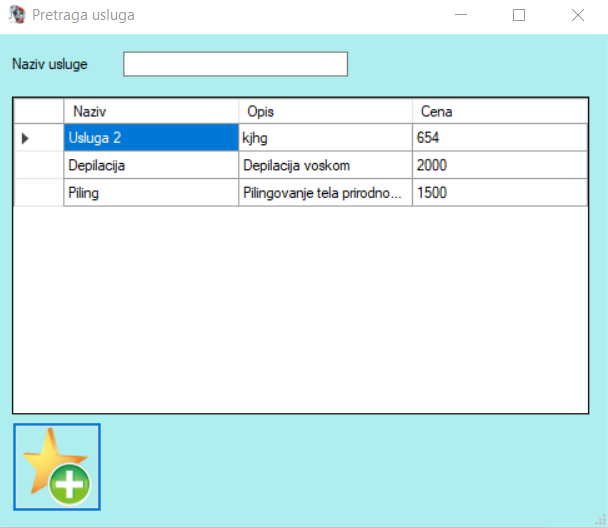
Aktori SK

Zaposleni

Učesnici SK

Zaposleni i sistem (program)

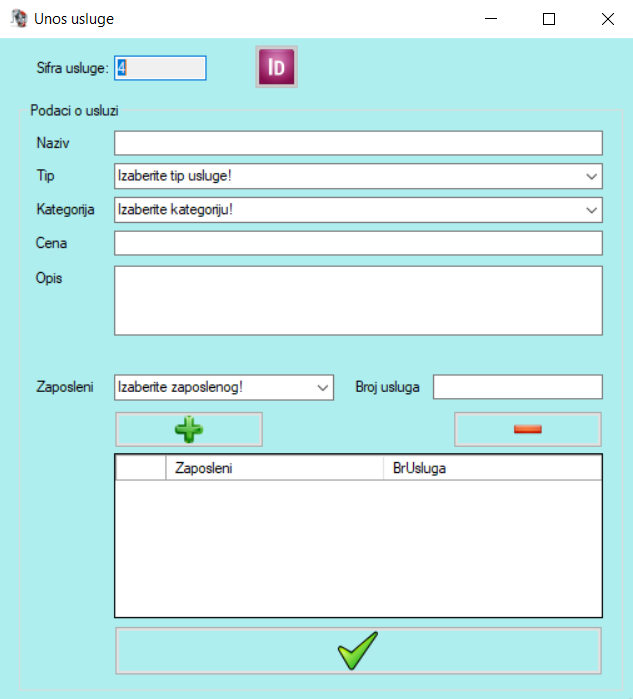
Preduslov: Sistem je uključen i zaposleni je prijavljen. Učitan je spisak usluga.Sistem prikazuje formu za rad sa uslugama.



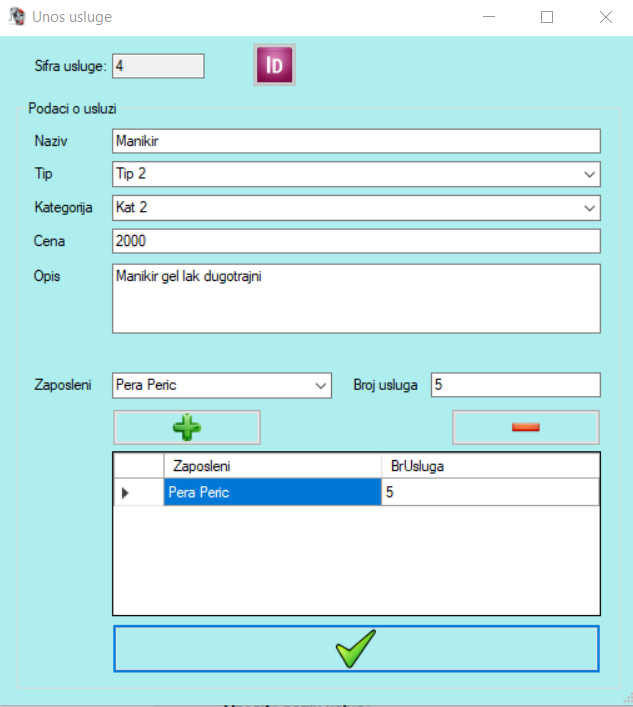
Osnovni scenario SK

1. Zaposleni unosi podatke o izvršenju usluga. (APUSO)

*Opis akcije: Zaposleni u prikazana polja unosi podatke o izvrsenju usluga koju želi da kreira i zamapti.*



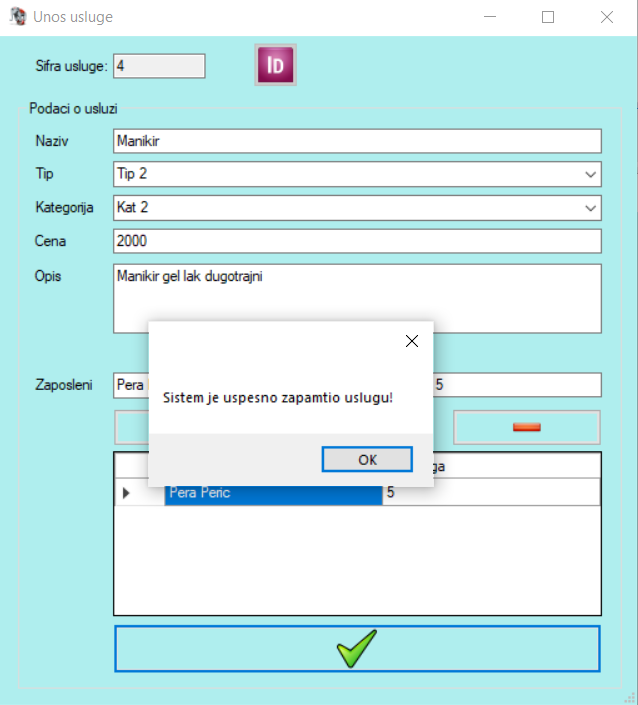
1. Zaposleni kontroliše da li je koretktno uneo podatke o izvršenju usluga.(ANSO)



1. Zaposleni poziva sistem da zapamti podatke o izvršenju usluga. (APSO)

*Opis akcije: Zaposleni klikom na dugme poziva sistemsku operaciju ZapamtiUslugu(Usluga) koja pamti izvrsenje usluga.*

1. Sistem pamti podatke o izvršenju usluga. (SO)
2. Sistem prikazuje zaposlenom zapamćeno izvršenje usluga i poruku : „Sistem je zapamtio izvršenje usluga“. (IA)



Alternativna scenarija

5.1. Ukoliko sistem ne može da zapamti podatke o izvršenju usluga on prikazuje

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

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

**Naziv SK**

Unos nove usluge

**Aktori SK**

Zaposleni

**Učesnici SK**

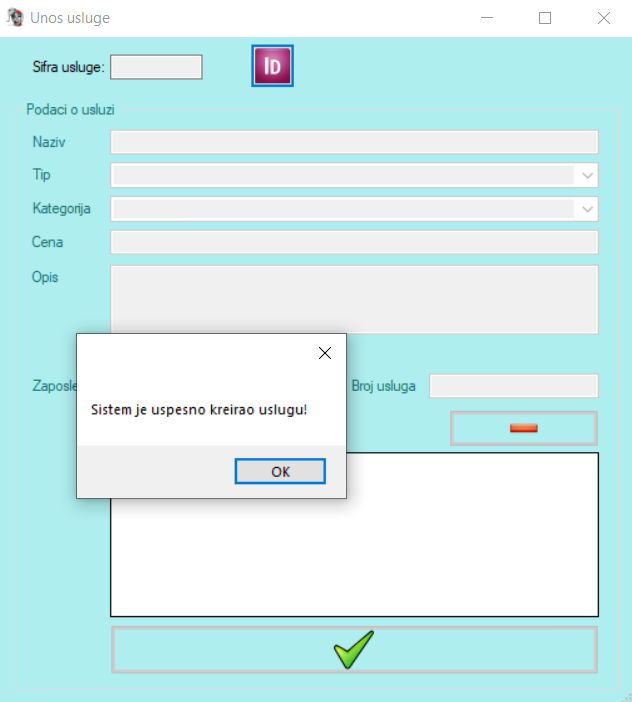
Zaposleni i sistem (program)

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

**Osnovni scenario SK**

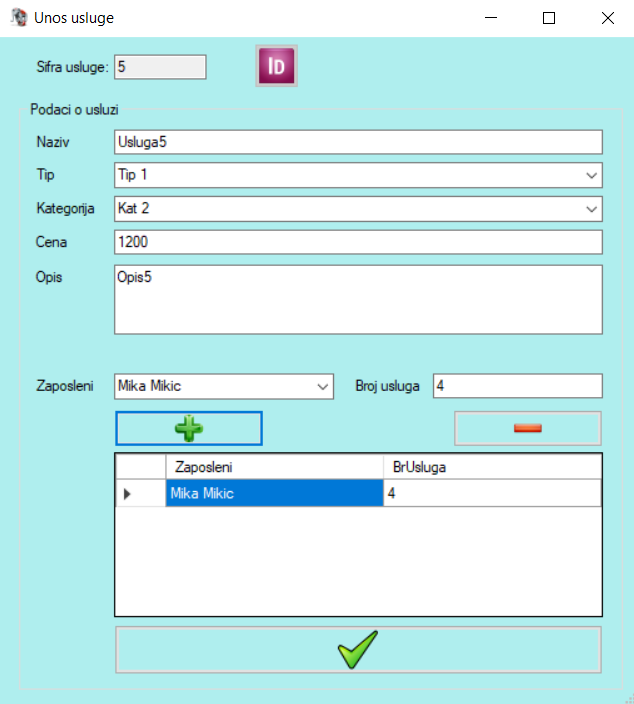
1. Zaposleni **poziva**sistem da kreira novu uslugu. (APSO)

*Opis akcije: Zaposleni u prikazana polja unosi podatke o novoj usluzi koju želi da kreira i zamapti.*

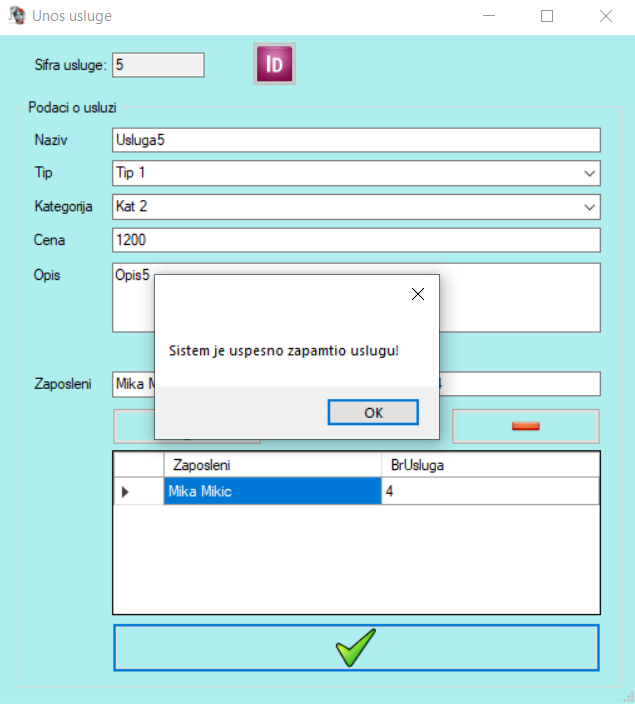


1. Sistem **kreira** novu uslugu. (SO)
2. Sistem zaposlenom **prikazuje** poruku: „Sistem je uspešno kreirao uslugu!“. (IA)
3. Zaposleni **unosi** podatke o usluzi. (APUSO)

*Opis akcije: Zaposleni klikom na dugme sistemsku operaciju ZapamtiUslugu(Usluga) koja pamti novu uslugu.*



1. Zaposleni **poziva** sistem da sačuva podatke o novoj usluzi. (APSO)
2. Sistem **pamti** podatke o usluzi. (SO)



1. Sistem **prikazuje** zaposlenom sačuvane podatke o novoj usluzi i ispisuje poruku: „Sistem je uspešno zapamtio uslugu!“ (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)

### **3.2.1.7. 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**

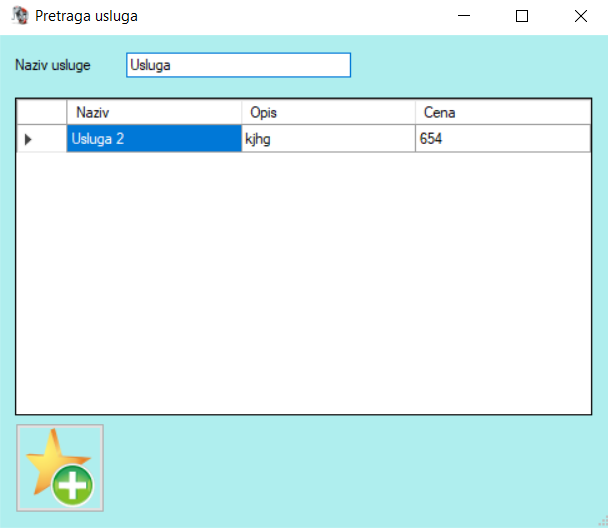
1. Zaposleni **unosi** kriterijum po kom pretražuje usluge. (APUSO)

*Opis akcije: Zaposleni u polje za kriterijum unosi naziv usluge.*

1. Zaposleni **poziva**sistem da pronađe usluge na osnovu kriterijuma. (APSO)

*Opis akcije: Zaposleni klikom na dugme enter prema zadatom kriterijumu poziva sistemsku operaciju PronadjiUsluge(zadatiKriterijum, List<Usluga>) koja pronalazi usluge po unetom kriterijumu.*

1. Sistem **traži** uslugu. (S0)
2. Sistem **prikazuje** zaposlenom pronađene usluge. (IA)

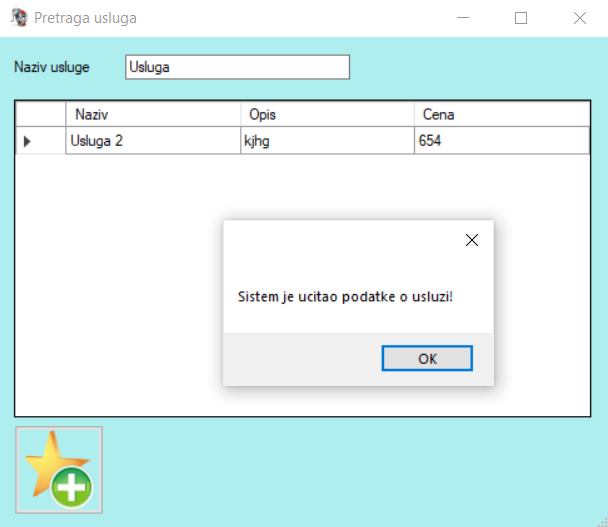


1. Zaposleni **bira** uslugu koju želi da pogleda. (APUSO)

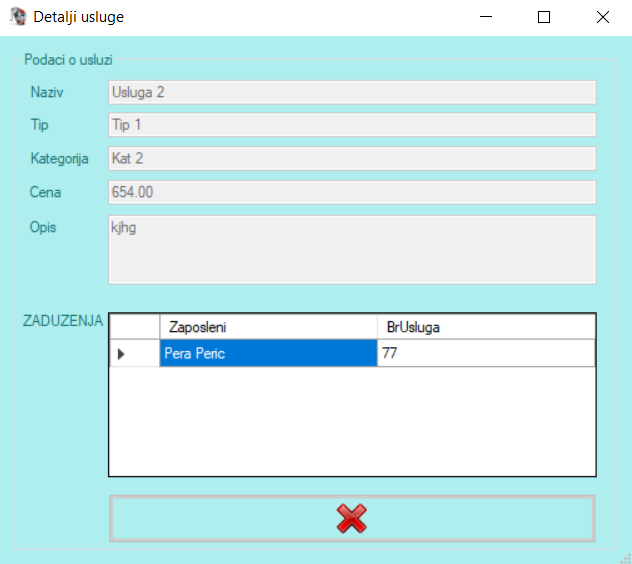
*Opis akcije: Zaposleni klikom na uslugu iz tabele vrši odabir.*

1. Zaposleni **poziva** sistem da prikaže podatke o izabranoj usluzi. (APSO)

*Opis akcije: Zaposleni klikom na dugme enter poziva sistemsku operaciju PrikaziUslugu(Usluga) koja prikazuje podatke o izabranoj usluzi.*



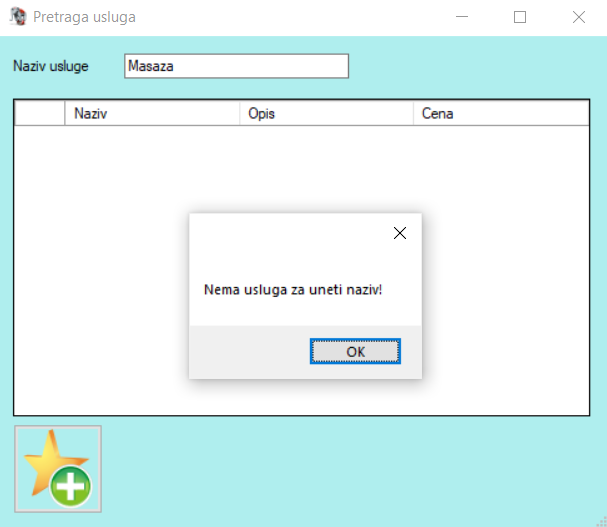
1. Sistem **traži** podatke o izabranoj usluzi. (SO)
2. Sistem **prikazuje** podatke o izabranoj usluzi i poruku: „Sistem je prikazao podatke o izabranoj usluzi“. (IA)



**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: „Nema usluga za uneti naziv.“ (IA)



### **3.2.1.8. 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.

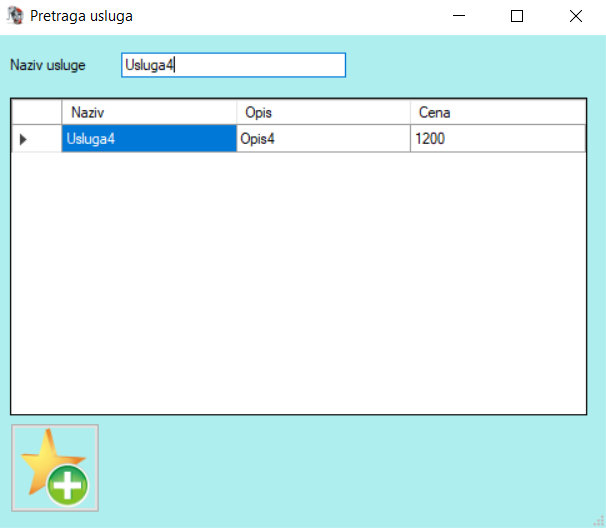
**Osnovni scenario SK**

1. Zaposleni**unosi** kriterijum po kom pretražuje usluge. (APUSO)

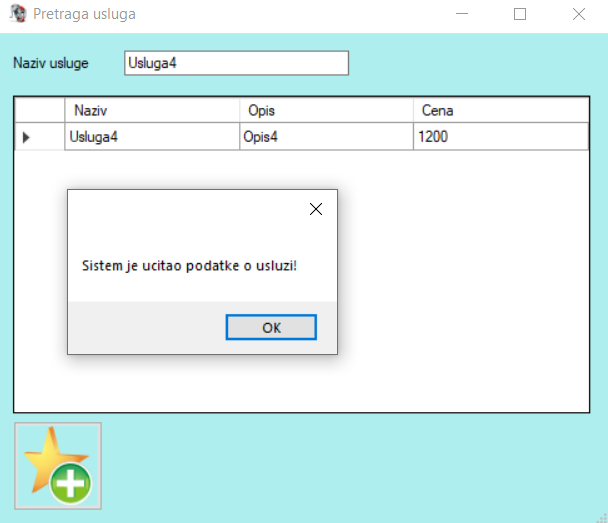
*Opis akcije: Zaposleni u polje za unos kriterijuma unosi naziv usluge.*

1. Zaposleni**poziva**sistem da pronađe usluge na osnovu kriterijuma. (APSO)

*Opis akcije: Zaposleni klikom na dugme enter prema zadatom kriterijumu poziva sistemsku operaciju PronadjiUsluge(zadatiKriterijum, List<Usluga>) koja pronalazi usluge po unetom kriterijumu.*

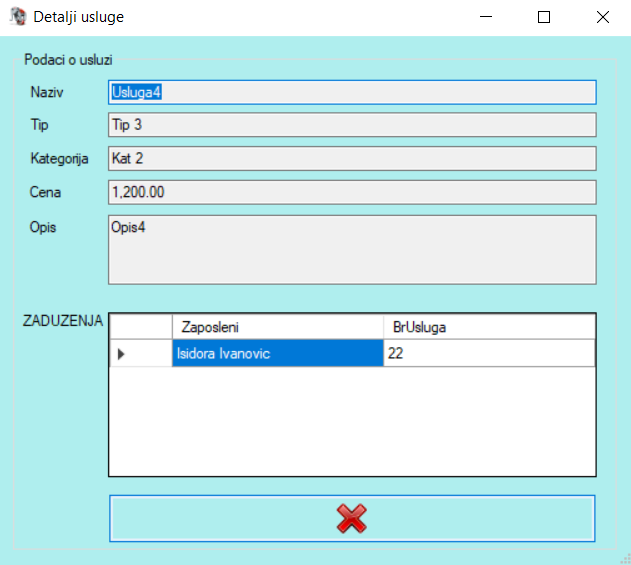
1. Sistem**traži** uslugu. (SO)
2. Sistem **prikazuje** zaposlenom pronađene usluge. (IA)
3. Zaposleni**bira** uslugu koju želi da obriše. (APUSO)

*Opis akcije: Zaposleni bira iz tabele usluga koje zadovoljavaju kriterijum klikom na određenu uslugu.*

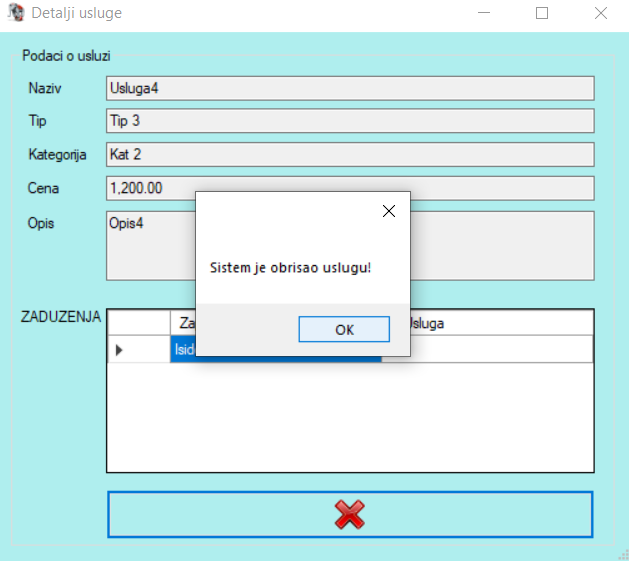


1. Zaposleni**poziva**sistem da obriše odabranu uslugu. (APSO)

*Opis akcije: Zaposleni klikom na dugme poziva sistemsku operaciju ObrisiUslugu(Usluga) koja briše zadatu uslugu.*

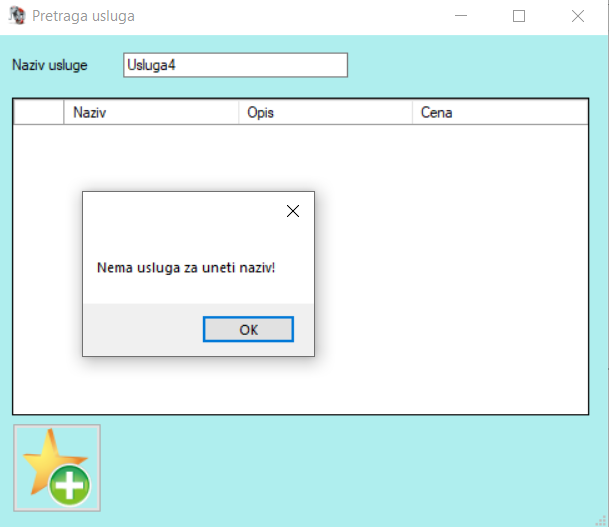


1. Sistem**briše** odabranu uslugu. (SO)
2. Sistem**prikazuje** zaposlenom odgovarajuću poruku: „Sistem je obrisao uslugu“. (IA)



**Alternativna scenarija**

4.1. Ukoliko sistem nije uspeo da pronađe uslugu sa odgovarajućim kriterijumom, prikazuje se poruka zaposlenom: „Nema usluga za uneti naziv“. Prekida se izvršenje scenarija. (IA)



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

### **3.2.1.9. 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**

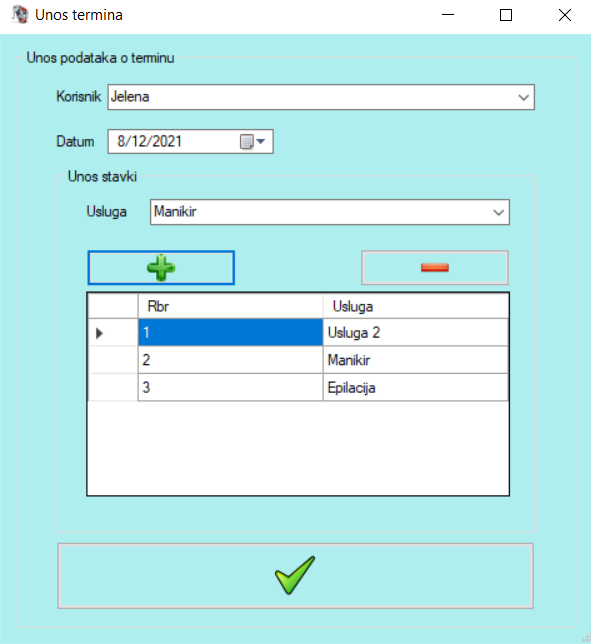
1. Zaposleni**unosi** podatke o terminu. (APUSO)

*Opis akcije: Zaposleni unosi podatke o novom terminu u odgovarajuća polja za unos.*

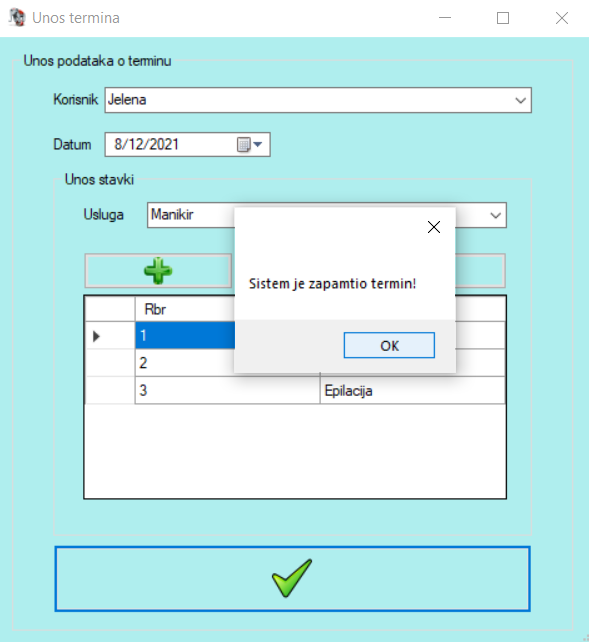


1. Zaposleni**kontroliše** da li je korektno uneo podatke o terminu. (ANSO)
2. Zaposleni**poziva**sistem da zapamti podatke o terminu. (APSO)

*Opis akcije: Zaposleni klikom na dugme poziva sistemsku operaciju ZapamtiTermin(Termin) koja pamti uneti termin.*

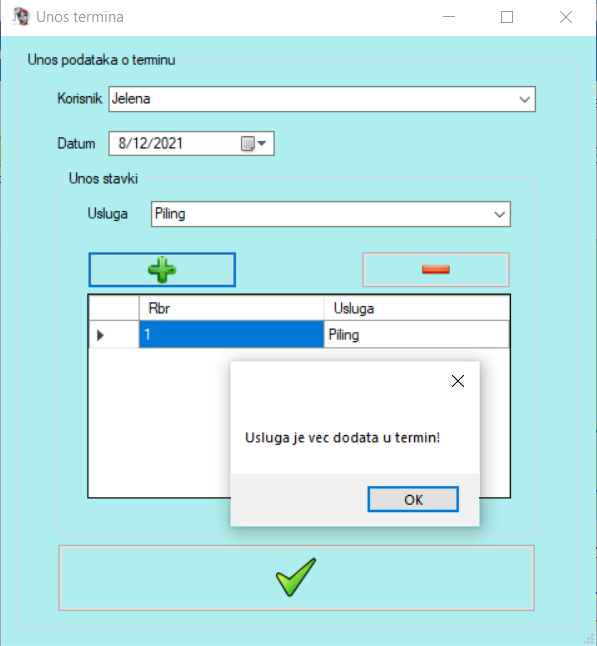


1. Sistem**pamti** podatke o terminu. (SO)
2. Sistem**prikazuje**zaposlenom zapamćenitermin i poruku: “Sistem je zapamtio termin”. (IA)



**Alternativna scenarija**

5.1 Ukoliko zaposleni unese istu uslugu za isti termin, sistem ne može da zapamti podatke o usluzi koja je uneta, on prikazuje zaposlenom poruku “Usluga je vec dodata u termin”. (IA)



### **3.2.1.10. 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.

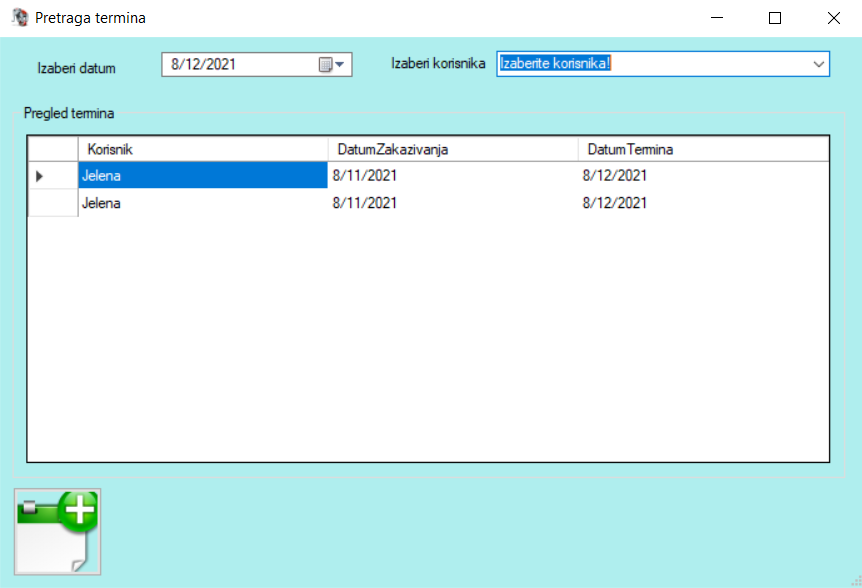
**Osnovni scenario SK**

1. Zaposleni**unosi** kriterijume po kojima pretražuje termine. (APUSO)

*Opis akcije: Zaposleni u polje za kriterijum unosi ime i prezime korisnika radi pretrage.*

1. Zaposleni **poziva**sistem da pronađe termin na osnovu kriterijuma. (APSO)

*Opis akcije: Sistem prema zadatom kriterijumu poziva sistemsku operaciju PronadjiTermine(zadatiKriterijum, List<Termin>) koja pronalazi termine po unetom kriterijumu*.



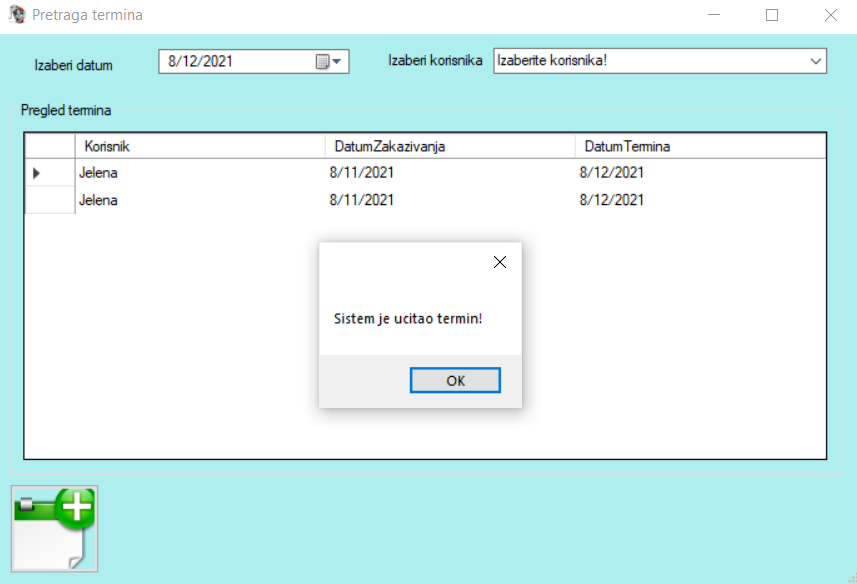
1. Sistem**traži** termin. (S0)
2. Sistem**prikazuje** zaposlenom pronađene termine. (IA)
3. Zaposleni**bira** termin koji želi da pogleda. (APUSO)

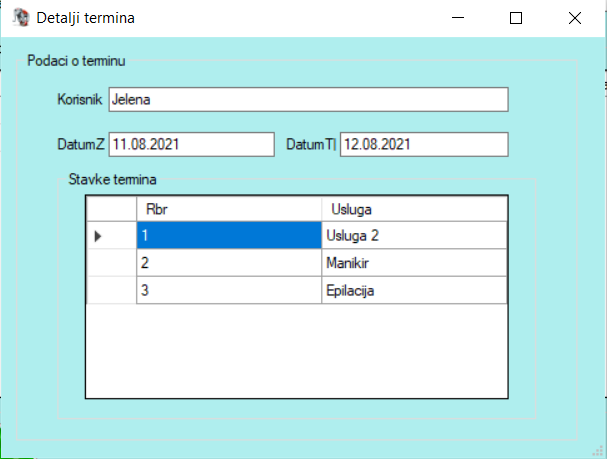
*Opis akcije: Zaposleni klikom na željeni termin u tabeli vrši odabir.*

1. Zaposleni**poziva** sistem da prikaže podatke o terminu. (APSO)

*Opis akcije: Zaposleni klikom na izabrani termin poziva sistemsku operaciju PrikaziTermin(Termin) koja prikazuje informacije o izabranom terminu.*

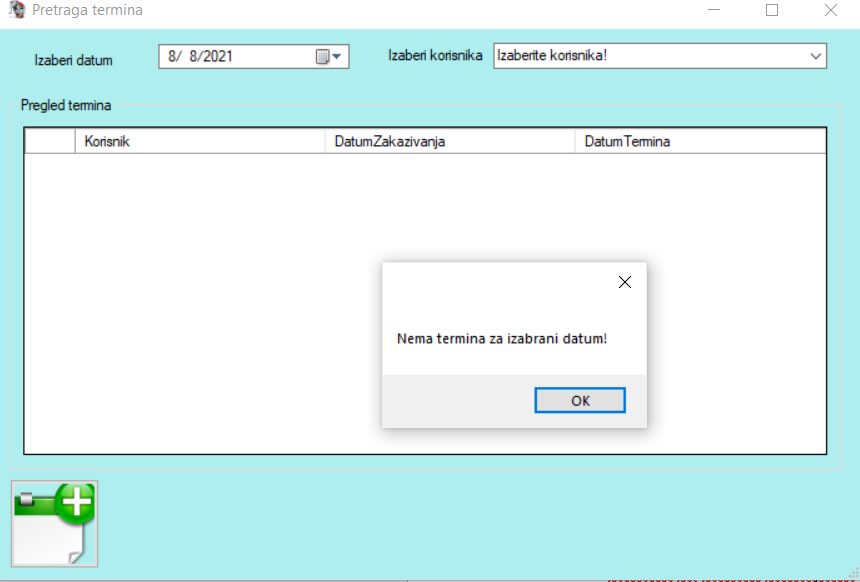
1. Sistem**traži** podatke o terminu. (SO)
2. Sistem **prikazuje** podatke o terminu i poruku: „Sistem je prikazao podatke o izabranom terminu“. (IA)





**Alternativna scenarija**

4.1. Ukoliko sistem nije uspeo da pronađe termine koji odgovaraju kriterijumu, zaposlenom se ispisuje odgovarajuća poruka: „Nema termina za izabrani datum!“. 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)

## **3.2.2. Projektovanje kontrolera korisničkog interfejsa**

Kontroler korisničkog interfejsa je odgovoran za:

1. Prihvatanje grafičkih objekata od ekranske forme,
2. Konvertovanje podataka koji se nalaze u grafičkim objektima u domenske objekte

koji će biti prosleđeni preko mreže do aplikacionog servera

1. Konvertovanje domenskih objekata u grafičke objekte i prosleđuje ih do ekranske

forme

## **3.3. Projektovanje aplikacione logike**

Aplikacioni serveri su odgovorni da obezbede servise koji će da omoguće realizaciju aplikacione logike softverskog sistema. Projektovani aplikacioni server sadrži:

* deo za komunikaciju sa klijentima
* kontroler aplikacione logike
* deo za komunikaciju sa skladištem podataka(Broker baze podataka)
* deo koji sadrži poslovnu logiku

Komunikacija sa klijentima

Deo za komunikaciju podiže serverski soket koji će da osluškuje mrežu. Kada klijentski soket uspostavi konekciju sa serverskim soketom, tada server generiše nit koja će uspostaviti dvosmernu vezu sa klijentom.

Slanje i primanje podataka od klijenta se obavlja razmenom objekata klase Transfer Objekat i ostvaruje se preko soketa.

Klijent šalje zahtev za izvršenje neke od sistemskih operacija do odgovarajuće niti koja je povezana sa tim klijentom. Ta nit prihvata zahtev, izvršava se odgovarajuća sistemska operacija, i rezultat se šalje nazad do klijenta.

### **3.3.1. Kontroler aplikacione logike**

Kontroler aplikacione logike prihvata zahtev za izvršenje sistemske operacije od niti klijenta i dalje ga preusmerava do klasa koje su odgovorne za izvršenje sistemskih operacija. Nakon izvršenja sistemske operacije kontroler aplikacione logike prihvata rezultat i prosleđuje ga pozivaocu (niti klijenta).

Preko klase Kontroler prihvataju se zahtevi o niti klijenta za izvršavanje sistemskih operacija, koji se zatim prosleđuje do odgovarajućih klasa koje su odgovorne za izvršenje sistemskih operacija. Za svaku sistemsku operaciju prave se softverske klase koje treba da realizuju sistemsku operaciju.

### **3.3.2. Poslovna logika**

Poslovna logika je opisana strukturom (domenskim klasama) i ponašanjem (sistemskim operacijama)

Projektovanje ponašanja softverskog sistema-sistemske operacije

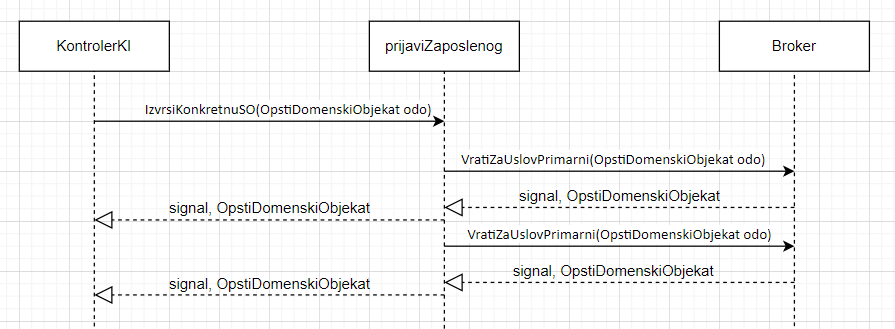
Tokom faze analize, odredili smo ugovore o sistemskim operacijama, pri čemu jedan ugovor opisuje ponašanje jedne sistemske operacije, i on opisuje šta operacija treba da radi, ali ne i kako. U fazi projektovanja se sada za svaki od ugovora projektuje konceptualno rešenje odnosno realizacija SO. Za svaku klasu se sada definiše KAKO će se sistemska operacija izvršiti. Za svaku sistemsku operaciju treba napraviti konceptualna rešenja koja su direktno povezana sa logikom problema. Za svaki od ugovora projektuje se konceptualno rešenje.

Ugovor UG1: **prijaviZaposlenog**(Zaposleni) Signal;

Veza sa SK: SK1

Preduslovi:

Postuslovi:



public class PrijaviZaposlenog : OpstaSO

{

public override object IzvrsiKonkretnuSO(OpstiDomenskiObjekat odo)

{

Zaposleni z= Broker.dajSesiju().VratiZaUslovPrimarni(odo) as Zaposleni;

z.Kategorija = Broker.dajSesiju().VratiZaUslovPrimarni(z.Kategorija) as Kategorija;

return z;

}

}

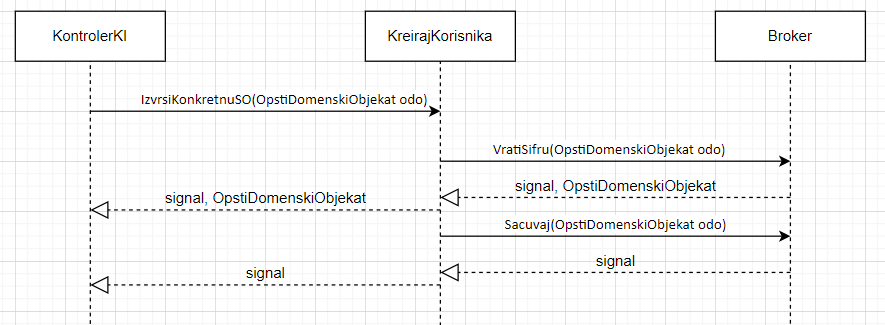
Ugovor UG2: **kreirajKorisnika**(Korisnik) Signal;

Veza sa SK: SK2

Preduslovi:Vrednosna i strukturna ograničenja nad objektom Korisnik moraju biti

zadovoljena.

Postuslovi: Podaci o korisniku su zapamćeni.



public class KreirajKorisnika : OpstaSO

{

public override object IzvrsiKonkretnuSO(OpstiDomenskiObjekat odo)

{

Korisnik d = new Korisnik();

d.Id = Broker.dajSesiju().VratiSifru(d);

Broker.dajSesiju().Sacuvaj(d);

return d;

}

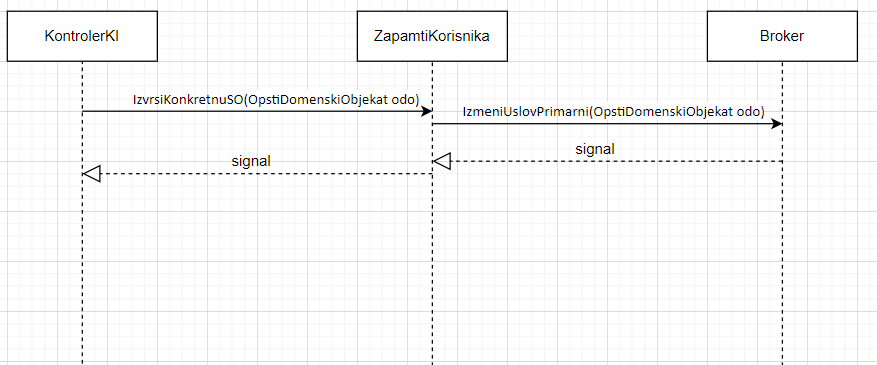
}

Ugovor UG3: **zapamtiKorisnika**(Korisnik) Signal;

Veza sa SK: SK2

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

Postuslovi: Podaci o korisniku su zapamćeni.



public class ZapamtiKorisnika : OpstaSO

{

public override object IzvrsiKonkretnuSO(OpstiDomenskiObjekat odo)

{

return Broker.dajSesiju().IzmeniUslovPrimarni(odo);

}

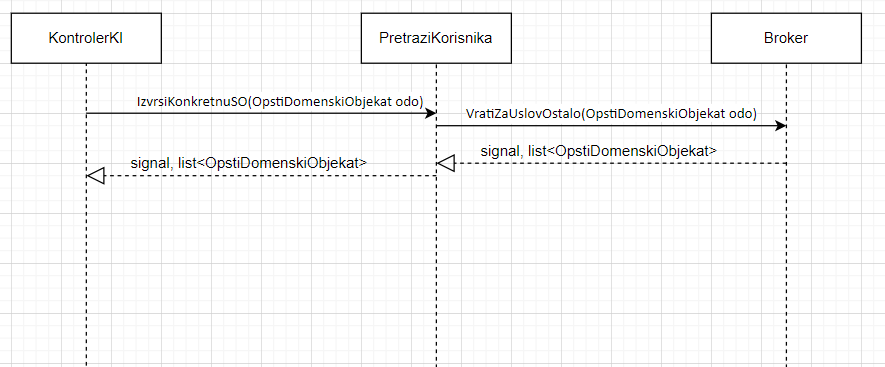
}

Ugovor UG4: **pretraziKorisnike**(zadatiKriterijum, List<Korisnik>) Signal;

Veza sa SK: SK3, SK4

Preduslovi:

Postuslovi:



public class PretraziKorisnike : OpstaSO

{

public override object IzvrsiKonkretnuSO(OpstiDomenskiObjekat odo)

{

return Broker.dajSesiju().vratiSveZaUslovOstalo(odo).OfType<Korisnik>().ToList();

}

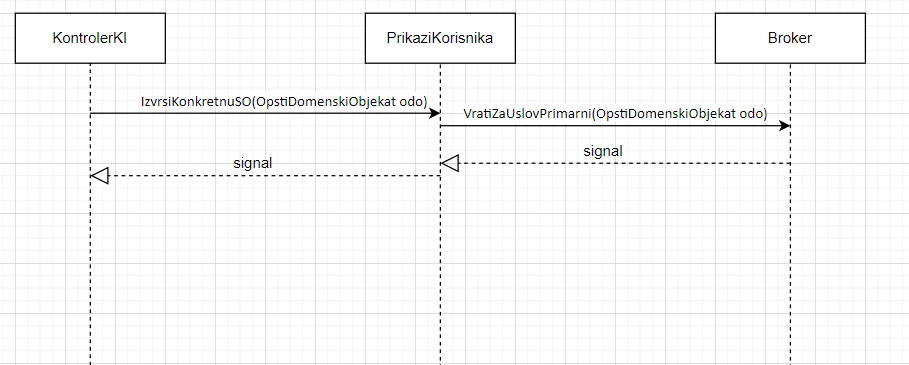
}

Ugovor UG5: **prikaziKorisnika**(Korisnik) Signal;

Veza sa SK: SK3, SK4

Preduslovi:

Postuslovi:



public class PrikaziKorisnika : OpstaSO

{

public override object IzvrsiKonkretnuSO(OpstiDomenskiObjekat odo)

{

return Broker.dajSesiju().VratiZaUslovPrimarni(odo) as Korisnik;

}

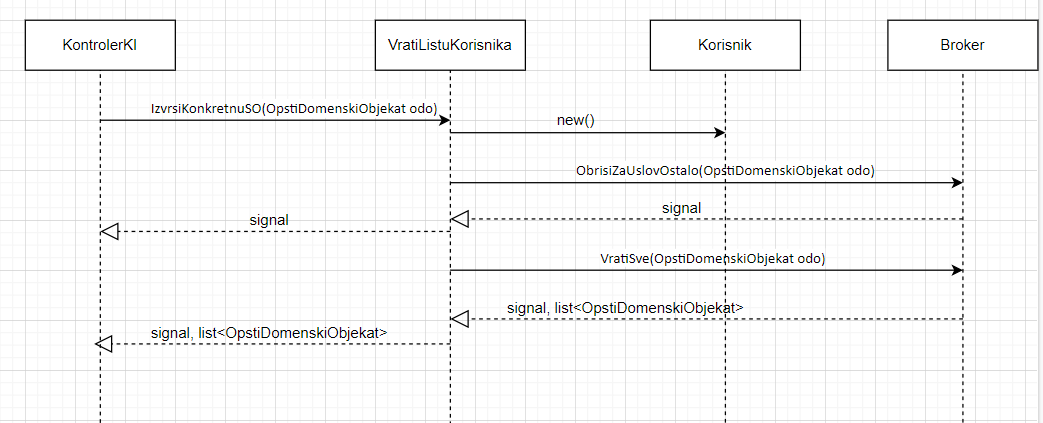
}

Ugovor UG6: **vratiListuKorisnika**(List<Korisnik>) Signal;

Veza sa SK: SK4, SK9

Preduslovi:

Postuslovi:



public class VratiLIstuKorisnika : OpstaSO

{

public override object IzvrsiKonkretnuSO(OpstiDomenskiObjekat odo)

{

Korisnik k = new Korisnik();

k.USLOV = "ImePrezime is null";

Broker.dajSesiju().ObrisiZaUslovOstalo(k);

return Broker.dajSesiju().vratiSve(odo).OfType<Korisnik>().ToList<Korisnik>();

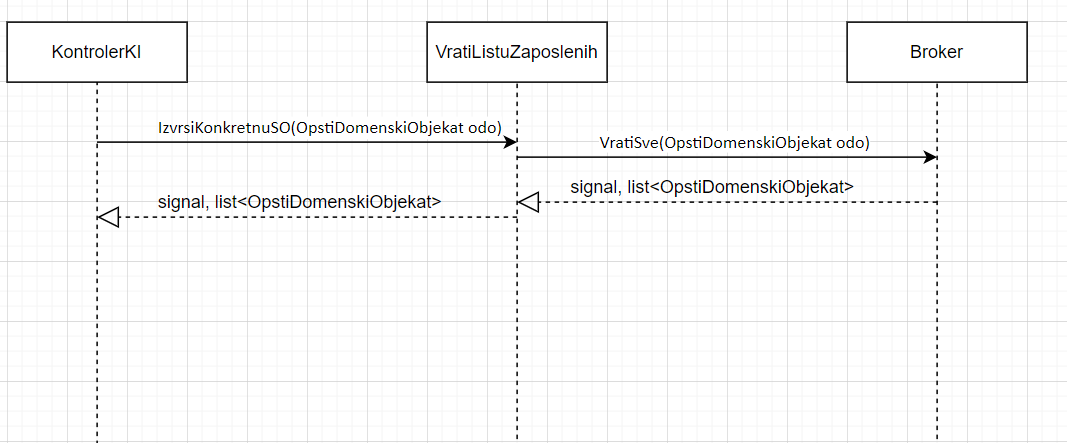
}

Ugovor UG7: **vratiListuZaposlenih**(List<Zaposleni>) Signal;

Veza sa SK: SK9

Preduslovi:

Postuslovi:



public class VratiListuZaposlenih : OpstaSO

{

public override object IzvrsiKonkretnuSO(OpstiDomenskiObjekat odo)

{

return Broker.dajSesiju().vratiSve(odo).OfType<Zaposleni>().ToList();

}

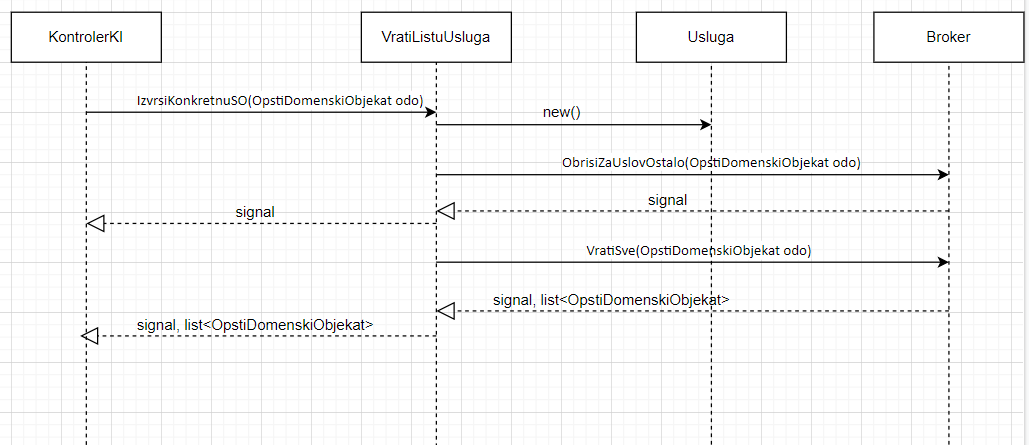
}

Ugovor UG8: **vratiListuUsluga**(List<Usluge>) Signal;

Veza sa SK: SK5

Preduslovi:

Postuslovi:



public class VratiLIstuUsluga : OpstaSO

{

public override object IzvrsiKonkretnuSO(OpstiDomenskiObjekat odo)

{

Usluga k = new Usluga();

k.Uslov = "naziv is null";

Broker.dajSesiju().ObrisiZaUslovOstalo(k);

return Broker.dajSesiju().vratiSve(odo).OfType<Usluga>().ToList<Usluga>();

}

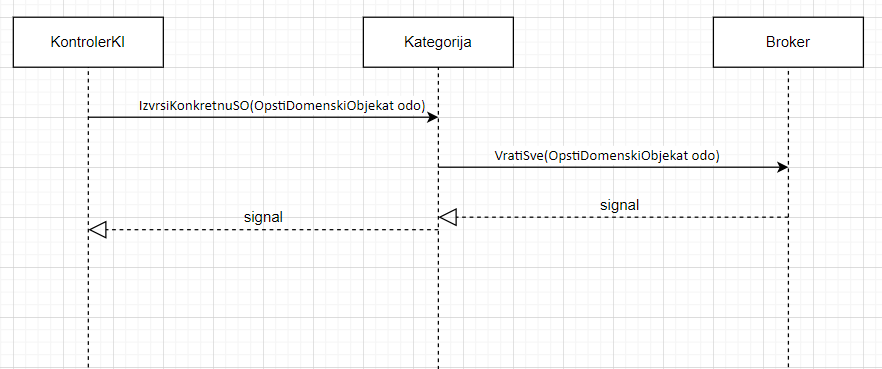
}

Ugovor UG9: **vratiListuKategorija**(List<Kategorija>) Signal;

Veza sa SK: SK5

Preduslovi:

Postuslovi:



public class VratiLIstuKategorija : OpstaSO

{

public override object IzvrsiKonkretnuSO(OpstiDomenskiObjekat odo)

{

return Broker.dajSesiju().vratiSve(odo).OfType<Kategorija>().ToList();

}

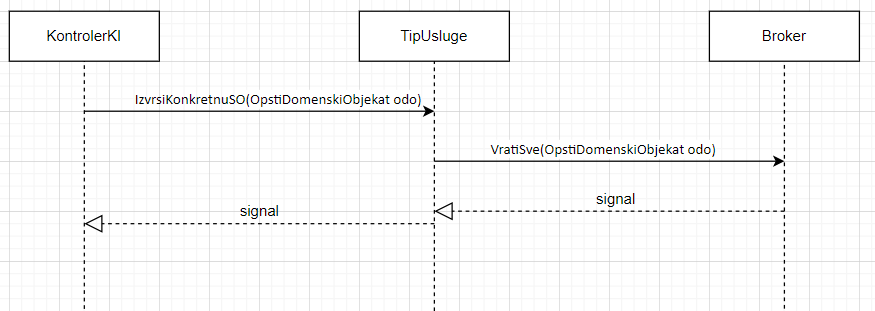
}

Ugovor UG10: **vratiListuTipovaUsluga(**List<TipUsluge>) Signal;

Veza sa SK: SK5

Preduslovi:

Postuslovi:



public class VratiListuTipovaUsluga : OpstaSO

{

public override object IzvrsiKonkretnuSO(OpstiDomenskiObjekat odo)

{

return Broker.dajSesiju().vratiSve(odo).OfType<TipUsluge>().ToList<TipUsluge>();

}

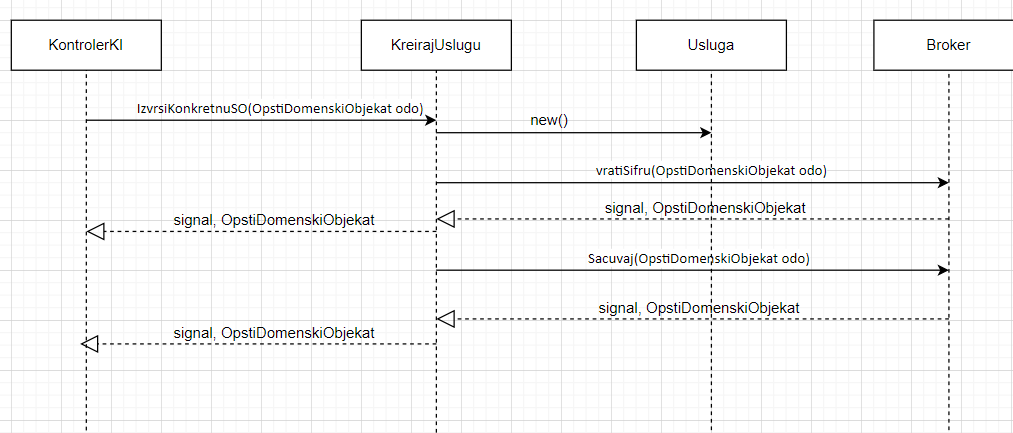
}

Ugovor UG11: **kreirajUslugu**(Usluga) Signal;

Veza sa SK: SK6

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

Postuslovi: Usluga je kreirana.



public class KreirajUslugu:OpstaSO

{

public override object IzvrsiKonkretnuSO(OpstiDomenskiObjekat odo)

{

Usluga u = new Usluga();

u.Id = Broker.dajSesiju().VratiSifru(u);

Broker.dajSesiju().Sacuvaj(u);

return u;

}

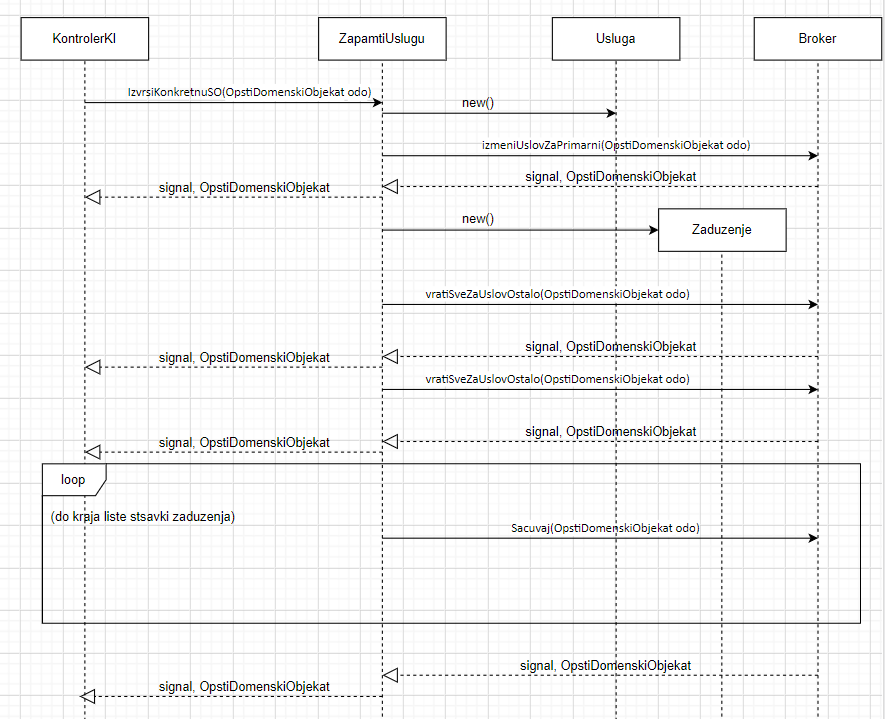
}

Ugovor UG12: **zapamtiUslugu**(Usluga) Signal;

Veza sa SK: SK6

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

Postuslovi: Usluga je sačuvana.



public class ZapamtiUslugu:OpstaSO

{

public override object IzvrsiKonkretnuSO(OpstiDomenskiObjekat odo)

{

Usluga u = odo as Usluga;

Broker.dajSesiju().IzmeniUslovPrimarni(odo);

Zaduzenje z = new Zaduzenje() { Usluga=u};

Broker.dajSesiju().ObrisiZaUslovOstalo(z);

foreach (Zaduzenje za in u.ListaZaduzenja)

{

Broker.dajSesiju().Sacuvaj(za);

}

return u;

}

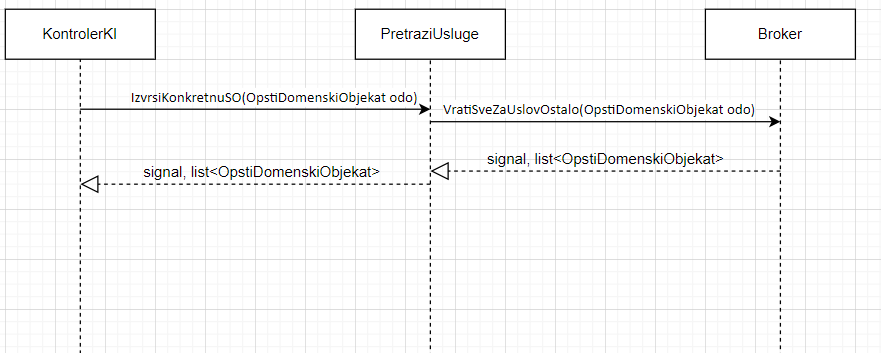
}

Ugovor UG13: **pretraziUsluge**(kriterijum, List<Usluga>) Signal;

Veza sa SK: SK7, SK8

Preduslovi:

Postuslovi:



public class PretraziUsluge : OpstaSO

{

public override object IzvrsiKonkretnuSO(OpstiDomenskiObjekat odo)

{

return Broker.dajSesiju().vratiSveZaUslovOstalo(odo).OfType<Usluga>().ToList<Usluga>();

}

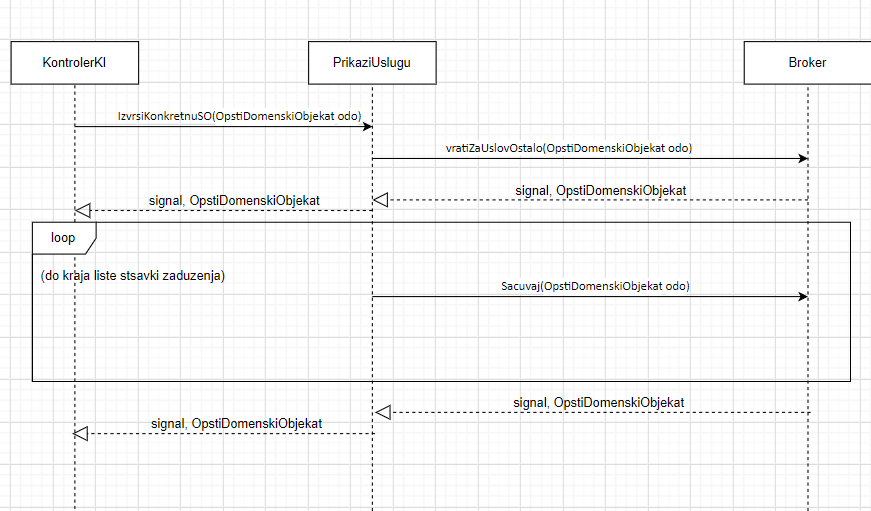
}

Ugovor UG14: **prikaziUslugu**(Usluga) Signal;

Veza sa SK: SK7

Preduslovi:

Postuslovi:



public class PrikaziUslugu : OpstaSO

{

public override object IzvrsiKonkretnuSO(OpstiDomenskiObjekat odo)

{

// nepotrebno

Usluga u = Broker.dajSesiju().VratiZaUslovPrimarni(odo) as Usluga;

u.Tip = Broker.dajSesiju().VratiZaUslovPrimarni(u.Tip) as TipUsluge;

u.Kategorija = Broker.dajSesiju().VratiZaUslovPrimarni(u.Kategorija) as Kategorija;

List<Zaduzenje> lista = Broker.dajSesiju().vratiSveZaUslovOstalo(new Zaduzenje() { Usluga=u}).OfType<Zaduzenje>().ToList();

foreach (Zaduzenje z in lista)

{

z.Usluga = u;

z.Zaposleni = Broker.dajSesiju().VratiZaUslovPrimarni(z.Zaposleni) as Zaposleni;

u.ListaZaduzenja.Add(z);

}

return u;

}

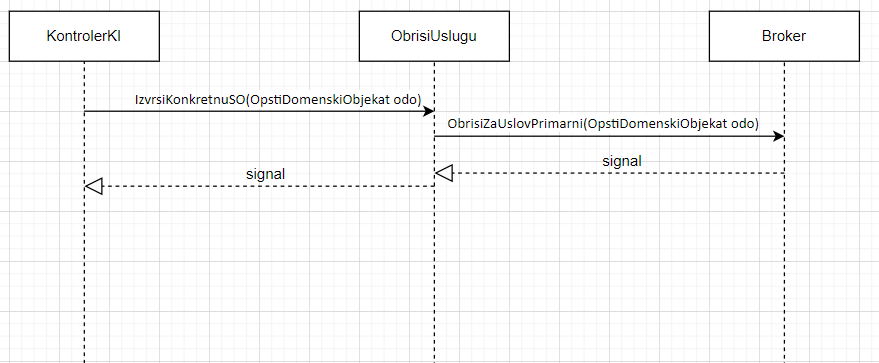
}

Ugovor UG15: **obrisiUslugu**(Usluga) Signal;

Veza sa SK: SK8

Preduslovi: Strukturna ograničenja nad objektom Usluga moraju biti zadovoljena.

Postuslovi: Usluga je obrisana.



public class ObrisiUslugu : OpstaSO

{

public override object IzvrsiKonkretnuSO(OpstiDomenskiObjekat odo)

{

return Broker.dajSesiju().ObrisiZaUslovPrimarni(odo);

}

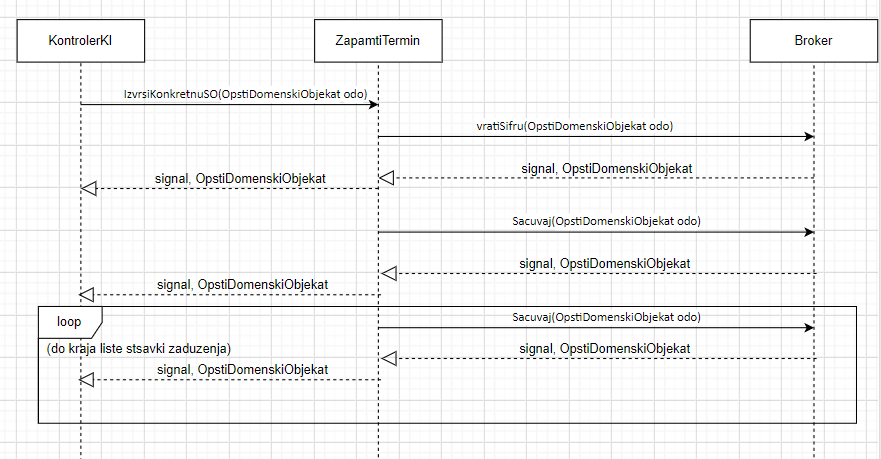
}

Ugovor UG16: **zapamtiTermin**(Termin) Signal;

Veza sa SK: SK9

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

Postuslovi: Termin je sačuvan.



public class ZapamtiTermin : OpstaSO

{

public override object IzvrsiKonkretnuSO(OpstiDomenskiObjekat odo)

{

Termin t = odo as Termin;

t.TerminId = Broker.dajSesiju().VratiSifru(t);

Broker.dajSesiju().Sacuvaj(t);

foreach (StavkaTermina st in t.StavkeTermina)

{

st.TerminID = t.TerminId;

Broker.dajSesiju().Sacuvaj(st);

}

return t;

}

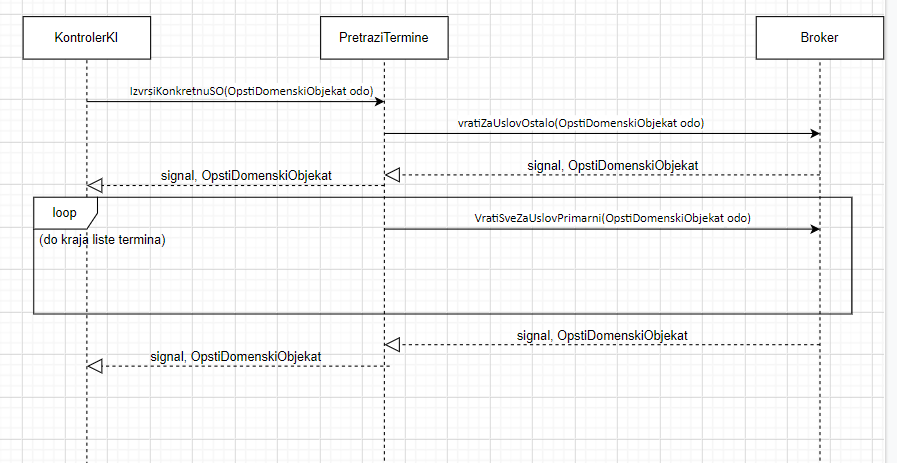
}

Ugovor UG16: **pretraziTermine**(kriterijum, List<Termin>) Signal;

Veza sa SK: SK10

Preduslovi:

Postuslovi:



public class PretraziTermine : OpstaSO

{

public override object IzvrsiKonkretnuSO(OpstiDomenskiObjekat odo)

{

List<Termin> lista= Broker.dajSesiju().vratiSveZaUslovOstalo(odo).OfType<Termin>().ToList<Termin>();

foreach (Termin t in lista)

{

t.Korisnik = Broker.dajSesiju().VratiZaUslovPrimarni(t.Korisnik) as Korisnik;

}

return lista;

}

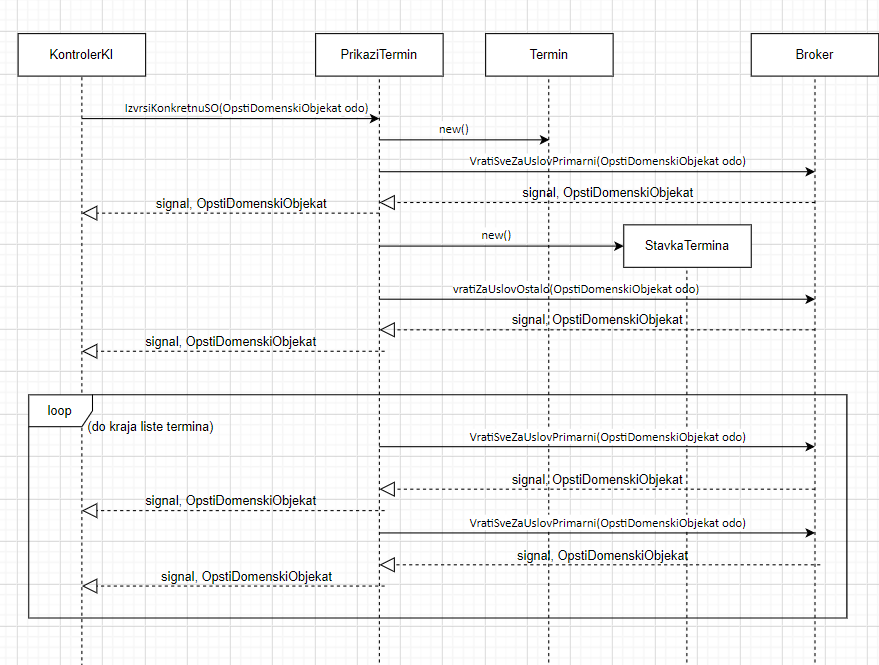
}

Ugovor UG18: **prikaziTermin**(Termin) Signal;

Veza sa SK: SK10

Preduslovi:

Postuslovi:



public class PrikaziTermin : OpstaSO

{

public override object IzvrsiKonkretnuSO(OpstiDomenskiObjekat odo)

{

Termin t = odo as Termin;

t.Korisnik = Broker.dajSesiju().VratiZaUslovPrimarni(t.Korisnik) as Korisnik;

StavkaTermina sta = new StavkaTermina();

sta.Uslov = "TerminID=" + t.TerminId;

List<StavkaTermina> lista = Broker.dajSesiju().vratiSveZaUslovOstalo(sta).OfType<StavkaTermina>().ToList();

foreach (StavkaTermina st in lista)

{

st.Usluga = Broker.dajSesiju().VratiZaUslovPrimarni(st.Usluga) as Usluga;

st.Usluga.Tip= Broker.dajSesiju().VratiZaUslovPrimarni(st.Usluga.Tip) as TipUsluge;

t.StavkeTermina.Add(st);

}

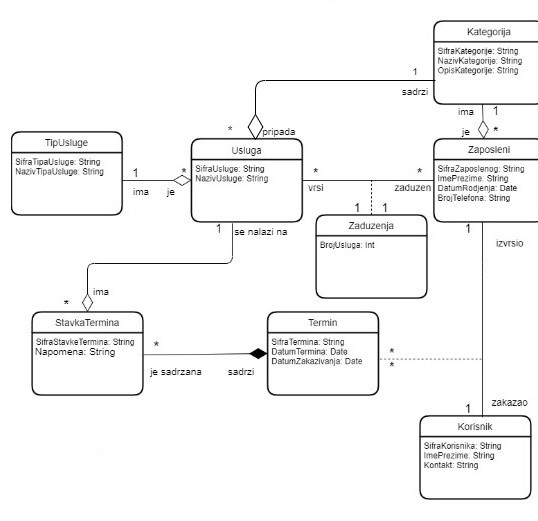
return t;

}

Klase koje su odgovorne za izvršenje sistemskih operacija nasleđuju klasu OpstaSO kako bi mogle da se povežu sa bazom i kako bi se njihovo izvršenje pratilo kao transakcija.OpstaSO predstavlja apstraktnu klasu čija glavna metoda (IzvrsiKonkretnuSO(OpstiDomenskiObjekat)) koja u sebi sadrži izvršenje operacije, potvrdu u bazu ukoliko je izvršenje uspešno, poništavanje ukoliko izvršenje nije bilo uspešno, zatvaranje konekcije.Svaka od sistemskih operacija daje svoju implementaciju metode za izvršenje konkretne sistemske operacije.

**Projektovanje strukture softverskog sistema**

Na osnovu konceptualnih klasa prave se softverske klase strukture.



*Softverske klase strukture:*

*[Serializable]*

*public class Korisnik:OpstiDomenskiObjekat{*

*int id;*

*string imePrezime;*

*string kontakt;*

*}*

*[Serializable]*

*public class Kategorija:OpstiDomenskiObjekat{*

*int id;*

*string naziv;*

*string opis;*

*}*

*[Serializable]*

*public class StavkaTermina : OpstiDomenskiObjekat{*

*int terminID;*

*int rbr;*

*Usluga usluga;*

*}*

*[Serializable]*

*public class Termin:OpstiDomenskiObjekat{*

*Zaposleni zaposleni;*

*Korisnik korisnik;*

*int terminID;*

*DateTime datumTermina;*

*DateTime datumZakazivanja;*

*BindingList<StavkaTermina> stavkeTermina;*

*}*

*[Serializable]*

*public class TipUsluge:OpstiDomenskiObjekat{*

*int id;*

*string naziv;*

*}*

*[Serializable]*

*public class Usluga:OpstiDomenskiObjekat{*

*int id;*

*string naziv;*

*string opis;*

*int cena;*

*TipUsluge tip;*

*Kategorija kategorija;*

*BindingList<Zaduzenje> listaZaduzenja;*

*}*

*[Serializable]*

*public class Zaduzenje:OpstiDomenskiObjekat{*

*Usluga usluga;*

*Zaposleni zaposleni;*

*int brUsluga;*

*}*

*[Serializable]*

*public class Zaposleni:OpstiDomenskiObjekat{*

*string id;*

*string imePrezime;*

*DateTime datumRodjenja;*

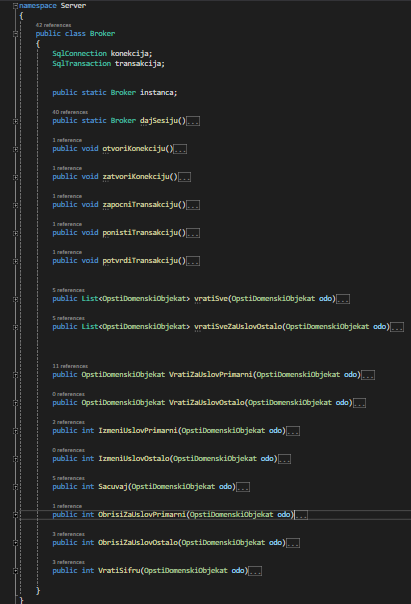
*string brTelefona;*

*Kategorija kategorija;*

*}*

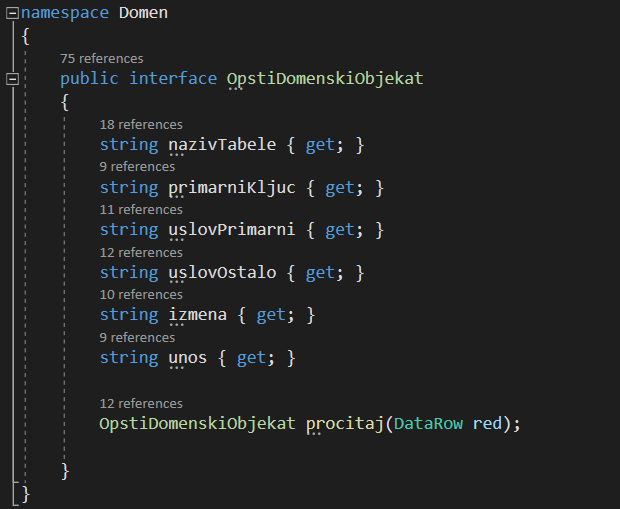
### **3.3.3. Broker baze podataka**

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



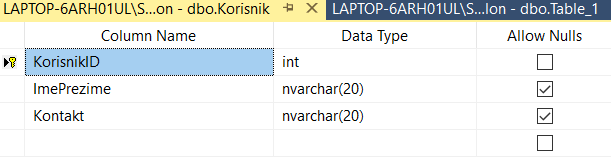
Sve metode Broker klase su projektovane kao generičke, što znači da mogu da prihvate različite domenske objekte preko parametara.Na taj način postižemo da u Broker klasi nemamo implementaciju pojedinačnih metoda za svaku domensku klasu.

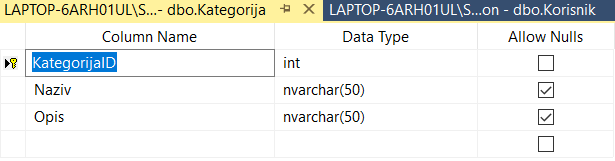
U procesu pravljenja Broker klase dobili smo metode apstraktne klase OpstiDomenskiObjekat. Svaka klasa iz domena nasleđuje datu apstraktnu klasu, i sve njene metode. Na taj način je omogućeno da metode klase Broker budu generičke i da se Broker klasa povezuje sa apstraktnom klasom.

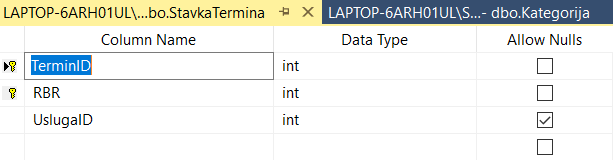


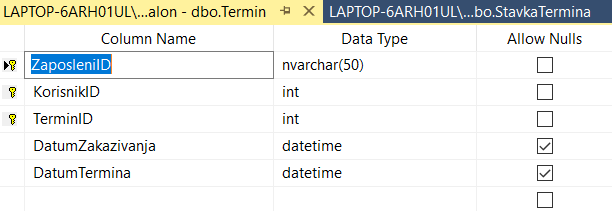
## **3.4. Projektovanje skladišta podataka**

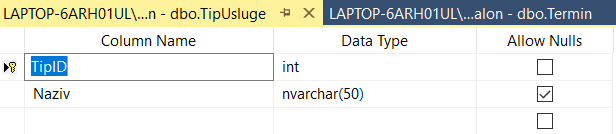
Na osnovu domenskih klasa softvera projektovane su tabele (skladišta podataka) relacionog sistema za upravljanje bazom podataka. Korišćen je Microsoft SQL Server Management Studio.

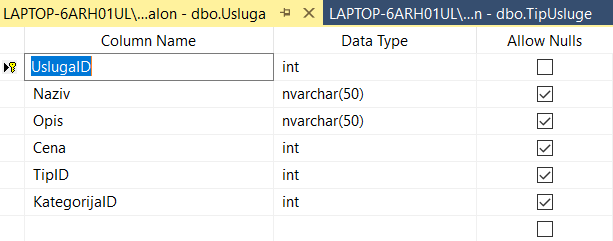


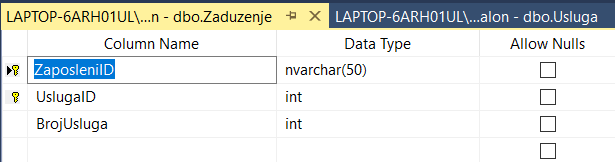


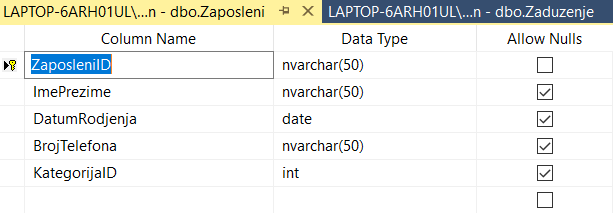






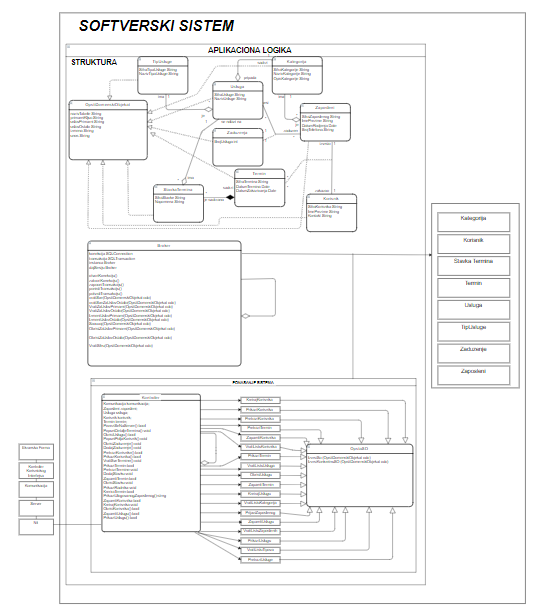






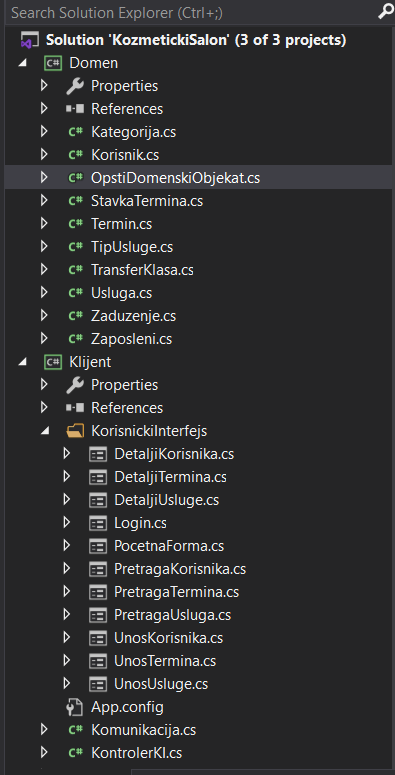
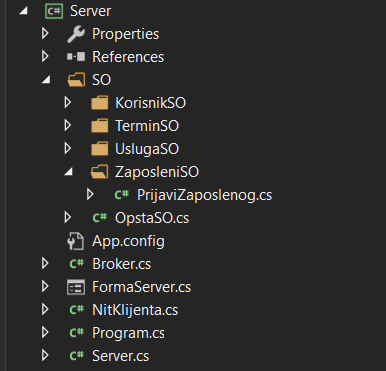
Na osnovu prethodnih celina, može se sastaviti cela arhitektura softverskog sistema za

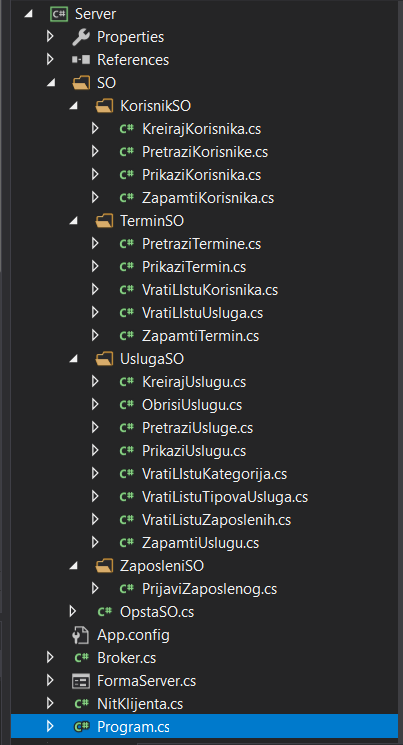
praćenje rada kozmetickog salona.



# **4.Implementacija**

Softverski sistem je razvijan u programskom jeziku “C#”. Sistem je projektovan kao klijent-server aplikacija. Kao sistem za upravljanje bazom podataka korišćen je Microsoft SQL Server Management Studio, dok je razvojno okruženje “Visual Studio”. 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**

Пројектовање софтвера – Скрипта, др Синиша Влајић, Београд – 2015.