Univerzitet u Beogradu

Fakultet organizacionih nauka

Laboratorija za softversko inženjerstvo

Seminarski rad iz predmeta

Projektovanje softvera

Tema: Softverski sistem za rezervaciju avionskih karata u avio kompanijama u C# okruženju

Profesor: Student:  
 dr Siniša Vlajić Ivana Todorović 1072/17

Beograd, 2019.

Sadržaj:

[1. Korisnički zahtevi 1](#_Toc18306465)

[1.1. Verbalni opis 1](#_Toc18306466)

[1.2. Slučajevi korišćenja 2](#_Toc18306467)

[1.2.1. SK 1: Slučаj korišćenjа – Prijavljivanje zaposlenog 3](#_Toc18306468)

[1.2.2. SK 2: Slučаj korišćenjа – Unos novog putnika 4](#_Toc18306469)

[1.2.3. SK 3: Slučаj korišćenjа – Izmenа podataka o putniku 5](#_Toc18306470)

[1.2.4. SK 4: Slučаj korišćenjа – Pretraživanje putnika 6](#_Toc18306471)

[1.2.5. SK 5: Slučаj korišćenjа – Brisanje putnika 7](#_Toc18306472)

[1.2.6. SK 6: Slučаj korišćenjа – Kreiranje nove rezervacije (Složen slučaj korišćenja) 8](#_Toc18306473)

[1.2.7. SK 7: Slučаj korišćenjа – Izmena rezervacije (Složen slučaj korišćenja) 9](#_Toc18306474)

[1.2.8. SK 8: Slučaj korišćenja – Pretraživanje rezervacija 10](#_Toc18306475)

[1.2.9. SK 9: Slučаj korišćenjа – Brisanje rezervacije 11](#_Toc18306476)

[2. Analiza 12](#_Toc18306477)

[2.1. Ponašanje softverskog sistema – Sistemski dijagram sekvenci 12](#_Toc18306478)

[2.1.1. DS 1: Dijagram sekvenci slučaja korišćenja – Prijavljivanje zaposlenog 12](#_Toc18306479)

[2.1.2. DS 2: Dijagram sekvenci slučaja korišćenja – Unos novog putnika 13](#_Toc18306480)

[2.1.3. DS 3: Dijagram sekvenci slučaja korišćenja – Izmena podataka o putniku 14](#_Toc18306481)

[2.1.4. DS 4: Dijagram sekvenci slučaja korišćenja – Pretraživanje putnika 18](#_Toc18306482)

[2.1.5. DS 5: Dijagram sekvenci slučaja korišćenja – Brisanje putnika 20](#_Toc18306483)

[2.1.6. DS 6: Dijagram sekvenci slučaja korišćenja – Kreiranje rezervacije 23](#_Toc18306484)

[2.1.7. DS 7: Dijagram sekvenci slučaja korišćenja - Izmena rezervacije 26](#_Toc18306485)

[2.1.8. DS 8: Dijagram sekvenci slučaja korišćenja – Pretraživanje rezervacija 29](#_Toc18306486)

[2.1.9. DS 9: Dijagram sekvenci slučaja korišćenja – Brisanje rezervacije 31](#_Toc18306487)

[2.2. Ponašanje softverskog sistema – Definisanje ugovora o sistemskim operacijama 35](#_Toc18306488)

[2.2.1. Ugovor UG 1:PronađiZaposlenog 35](#_Toc18306489)

[2.2.2. Ugovor UG 2:KreirajPutnika 35](#_Toc18306490)

[2.2.3. Ugovor UG 2:ZapamtiPutnika 35](#_Toc18306491)

[2.2.4. Ugovor UG 3:PretraživanjePutnika 35](#_Toc18306492)

[2.2.5. Ugovor UG 4:UčitajPutnika 35](#_Toc18306493)

[2.2.6. Ugovor UG 5:ObrišiPutnika 35](#_Toc18306494)

[2.2.7. Ugovor UG 6:UčitajSvePutnike 36](#_Toc18306495)

[2.2.8. Ugovor UG 7:UčitajSveLetove 36](#_Toc18306496)

[2.2.9. Ugovor UG 8:KreirajRezervaciju 36](#_Toc18306497)

[2.2.10. Ugovor UG 9:ZapamtiRezervaciju 36](#_Toc18306498)

[2.2.11. Ugovor UG 10:PretraživanjeRezervacija 36](#_Toc18306499)

[2.2.12. Ugovor UG 11:UčitajRezervaciju 36](#_Toc18306500)

[2.2.13. Ugovor UG 12:ObrišiRezervaciju 36](#_Toc18306501)

[2.3. Struktura softverskog sistema – Konceptualni (domenski) model 37](#_Toc18306502)

[2.4. Struktura softverskog sistema – Relacioni model 38](#_Toc18306503)

[3. Projektovanje 44](#_Toc18306504)

[3.1. Arhitektura softverskog sistema 44](#_Toc18306505)

[3.2. Projektovanje korisničkog interfejsa 45](#_Toc18306506)

[3.2.1. Projektovanje ekranskih formi 46](#_Toc18306507)

[SK 1: Slučаj korišćenjа – Prijavljivanje zaposlenog 46](#_Toc18306508)

[SK 2: Slučаj korišćenjа – Unos novog putnika 48](#_Toc18306509)

[SK 3: Slučаj korišćenjа – Izmenа podataka o putniku 50](#_Toc18306510)

[SK 4: Slučаj korišćenjа – Pretraživanje putnika 55](#_Toc18306511)

[SK 5: Slučаj korišćenjа – Brisanje putnika 59](#_Toc18306512)

[SK 6: Slučаj korišćenjа – Kreiranje nove rezervacije (Složen slučaj korišćenja) 64](#_Toc18306513)

[SK 7: Slučаj korišćenjа – Izmena rezervacije (Složen slučaj korišćenja) 67](#_Toc18306514)

[SK 8: Slučaj korišćenja – Pretraživanje rezervacija 73](#_Toc18306515)

[SK 9: Slučаj korišćenjа – Brisanje rezervacije 76](#_Toc18306516)

[3.2.2. Projektovanje kontrolera korisničkog interfejsa 81](#_Toc18306517)

[3.3. Projektovanje aplikacione logike 83](#_Toc18306518)

[3.3.1. Kontroler aplikacione logike 83](#_Toc18306519)

[3.3.2. Sistemske operacije 86](#_Toc18306520)

[Ugovor UG 1:PronađiZaposlenog 87](#_Toc18306521)

[Ugovor UG 2:KreirajPutnika 88](#_Toc18306522)

[Ugovor UG 3:ZapamtiPutnika 88](#_Toc18306523)

[Ugovor UG 4:PretraživanjePutnika 88](#_Toc18306524)

[Ugovor UG 5:UčitajPutnika 89](#_Toc18306525)

[Ugovor UG 6:ObrišiPutnika 89](#_Toc18306526)

[Ugovor UG 7:UčitajSvePutnike 90](#_Toc18306527)

[Ugovor UG 8:UčitajSveLetove 90](#_Toc18306528)

[Ugovor UG 9:KreirajRezervaciju 90](#_Toc18306529)

[Ugovor UG 10:ZapamtiRezervaciju 91](#_Toc18306530)

[Ugovor UG 11:PretraživanjeRezervacija 91](#_Toc18306531)

[Ugovor UG 12:UčitajRezervaciju 92](#_Toc18306532)

[Ugovor UG 13:ObrišiRezervaciju 92](#_Toc18306533)

[3.3.3. Domenske klase 94](#_Toc18306534)

[3.3.4. Broker baze podataka 95](#_Toc18306535)

[3.3.5. Projektovanje skladišta podataka 97](#_Toc18306536)

[4. Implementacija 99](#_Toc18306537)

[5. Testiranje 100](#_Toc18306538)

[6. Literatura 101](#_Toc18306539)

# 

# 1. Korisnički zahtevi

## 1.1. Verbalni opis

Potrebno je napraviti softverski sistem za avio kompaniju koji će omogućiti vođenje evidencije o rezervacijama avionskih karata za letove širom sveta. Napredovanjem tehnologije ovaj softver bi danas mogao biti veoma unapređen i zamenio bi zastareli način evidentiranja klijenata i svih potrebnih podataka u enormno velike količine papirologije.

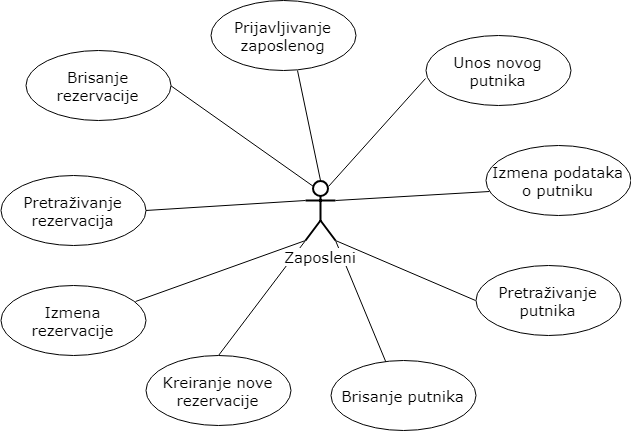
Kreiranje rezervacije avionske karte se vrši elektronskim putem. Korisnik ovog sistema je zaposleni u avio kompaniji koji mora biti prijavljen na sistem kako bi mogao da izvrši bilo koju operaciju. Zaposleni ima mogućnost da pored unosa nove rezervacije, izmeni podatke o postojećoj, pretraži rezervacije ili da ih obriše.

U sistemu se vodi evidencija o svakom klijentu to jest putniku koji želi da rezerviše svoju kartu za željeni let. Shodno tome, potrebno je obezbediti sledeće funkcionalnosti: unos novog putnika, izmena podataka o putniku, pretraživanje putnika i brisanje putnika.

## 1.2. Slučajevi korišćenja

Identifikovani su sledeći slučajevi korišćenja (prikazani takođe na slici 1) :

1. Prijavljivanje zaposlenog
2. Unos novog putnika
3. Izmena podataka o putniku
4. Pretraživanje putnika
5. Brisanje putnika
6. Kreiranje nove rezervacije (složen slučaj korišćenja)
7. Izmena rezervacije (složen slučaj korišćenja)
8. Pretraživanje rezervacija
9. Brisanje rezervacije



Slika 1 - Dijagram slučajeva korišćenja

### 1.2.1. SK 1: Slučаj korišćenjа – Prijavljivanje zaposlenog

**Nаziv SK**

Prijavljivanje *zaposlenog*

**Aktori SK**

**Zaposleni**

**Učesnici SK**

**Zaposleni**i**sistem** *(progrаm)*

**Preduslov**: **Sistem** je uključen i prikаzuje formu zа prijavljivanje *zaposlenog*.

**Osnovni scenаrio SK**

1. **Zaposleni** unosi identifikacione podatke tj. korisničko ime i šifru. (APUSO)
2. **Zaposleni** kontroliše dа li je korektno uneo identifikacione podаtke tj. korisničko ime i sifru. (ANSO)
3. **Zaposleni** pozivа **sistem** da pronađe *zaposlenog* sa zadatim podacima. (APSO)
4. **Sistem** pretražuje *zaposlenog*. (SO)
5. **Sistem** prikаzuje *zaposlenom* poruku: “Uspešno ste se prijavili na sistem”, a zatim mu omogućava pristup sistemu. (IA)

**Alternаtivnа scenаrijа**

5.1. Ukoliko **sistem** ne može da pronađe zadatog *zaposlenog*, on prikazuje **zaposlenom** poruku “**Sistem** ne može da pronađe *zaposlenog* sa unetim podacima”. (IA)

### 1.2.2. SK 2: Slučаj korišćenjа – Unos novog putnika

**Nаziv SK**

Unos *novog putnika*

**Aktori SK**

**Zaposleni**

**Učesnici SK**

**Zaposleni**i**sistem** *(progrаm)*

**Preduslov**: **Sistem** je uključen i **zaposleni** je ulogovan pod svojom šifrom. Sistem prikazuje formu za unos *novog putnika*.

**Osnovni scenаrio SK**

1. **Zaposleni** unosi podatke o *novom putniku*. (APUSO)

2. **Zaposleni** kontroliše da li je korektno uneo podatke o *novom putniku*. (ANSO)

3. **Zaposleni** poziva **sistem** da zapamti podatke o *novom putniku*. (APSO)

4. **Sistem** pamti podatke o *novom putniku*. (SO)

5. **Sistem** prikazuje **zaposlenom** zapamćenog putnika i poruku: “**Sistem** je zapamtio *novog putnika*“.(IA)

**Alternаtivnа scenаrijа**

5.1. Ukoliko **sistem** ne može da zapamti podatke o *novom putniku* on prikazuje **zaposlenom** poruku “**Sistem** ne može da zapamti *novog putnika*”. (IA)

### 1.2.3. SK 3: Slučаj korišćenjа – Izmenа podataka o putniku

**Nаziv SK**

Izmenа podataka o *putniku*

**Aktori SK**

**Zaposleni**

**Učesnici SK**

**Zaposleni**i**sistem** *(progrаm)*

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

**Osnovni scenаrio SK**

1. **Zaposleni** unosi vrednost po kojoj pretražuje *putnike*. (APUSO)

2. **Zaposleni** poziva **sistem** da nađe *putnike* po zadatoj vrednosti. (APSO)

3. **Sistem** traži *putnike* po zadatoj vrednosti. (SO)

4. **Sistem** prikazuje **zaposlenom** *putnike* i poruku: “**Sistem** je našao putnike po zadatoj vrednosti ”. (IA)

5. **Zaposleni** bira *putnika* čije podatke želi da izmeni. (APUSO)

6. **Zaposleni** poziva **sistem** da učita podatke o odabranom *putniku.* (APSO)

7. **Sistem** učitava podatke o odabranom *putniku*. (SO)

8. **Sistem** prikazuje **zaposlenom** podatke o *putniku* i poruku: “Odabrani *putnik* je prikazan.” (IA)

9. **Zaposleni** unosi (menja) podatke o *putniku*. (APUSO)

10. **Zaposleni** kontroliše da li je korektno uneo podatke o *putniku*. (ANSO)

11. **Zaposleni** poziva **sistem** da zapamti podatke o *putniku*. (APSO)

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

13. **Sistem** prikazuje **zaposlenom** zapamćenog *putnika* i poruku: “**Sistem** je zapamtio *putnika*.” (IA)

**Alternаtivnа scenаrijа**

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

8.1. Ukoliko **sistem** ne može da učita *putnika* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da prikaže odabranog *putnika*”. Prekida se izvršenje scenarija. (IA)

13.1. Ukoliko **sistem** ne može da zapamti podatke o *putniku* on prikazuje **zaposlenom** poruku “**Sistem** ne može da zapamti *putnika*”. (IA)

### 1.2.4. SK 4: Slučаj korišćenjа – Pretraživanje putnika

**Naziv SK**

Pretraživanje *putnika*

**Aktori SK**

**Zaposleni**

**Učesnici SK**

**Zaposleni** i **sistem** (program)

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

**Osnovni scenario SK**

1. **Zaposleni** unosi vrednost(tj. kriterijum) po kojoj pretražuje *putnike*. (APUSO)

2. **Zaposleni** poziva **sistem** da nađe *putnike* po zadatoj vrednosti. (APSO)

3. **Sistem** pretražuje *putnike* po zadatoj vrednosti. (SO)

4. **Sistem** prikazuje **zaposlenom** *putnike* i poruku: “**Sistem** je našao putnike po zadatoj vrednosti ”. (IA)

5. **Zaposleni** bira *putnika* čije podatke želi da vidi. (APUSO)

6. **Zaposleni** poziva **sistem** da učita podatke o odabranom *putniku*. (APSO)

7. **Sistem** učitava podatke o odabranom *putniku*. (SO)

8. **Sistem** prikazuje **zaposlenom** podatke o odabranom *putniku* i poruku: “Odabrani *putnik* je prikazan.” (IA)

**Alternativna scenarija**

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

8.1. Ukoliko **sistem** ne može da učita podatke o odabranom *putniku* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da prikaže podatke o odabranom *putniku*.”(IA)

### 1.2.5. SK 5: Slučаj korišćenjа – Brisanje putnika

**Naziv SK**

Brisanje *putnika*

**Aktori SK**

**Zaposleni**

**Učesnici SK**

**Zaposleni** i **sistem** (program)

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

**Osnovni scenario SK**

1. **Zaposleni** unosi vrednost po kojoj pretražuje *putnike*. (APUSO)

2. **Zaposleni** poziva **sistem** da nađe *putnike* po zadatoj vrednosti. (APSO)

3. **Sistem** traži *putnike* po zadatoj vrednosti. (SO)

4. **Sistem** prikazuje **zaposlenom** *putnike* i poruku: “**Sistem** je našao putnike po zadatoj vrednosti ”. (IA)

5. **Zaposleni** bira *putnika* kojeg želi da obriše. (APUSO)

6. **Zaposleni** poziva **sistem** da učita podatke o odabranom *putniku*. (APSO)

7. **Sistem** učitava podatke o odabranom *putniku*. (SO)

8. **Sistem** prikazuje **zaposlenom** podatke o *putniku* i poruku: “Odabrani *putnik* je prikazan.” (IA)

9. **Zaposleni** poziva **sistem** da obriše *putnika*. (APSO)

10. **Sistem** briše *putnika*. (SO)

11. **Sistem** prikazuje **zaposlenom** poruku: “**Sistem** je obrisao *putnika*.” (IA)

**Alternativna scenarija**

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

8.1. Ukoliko **sistem** ne može da učita *putnika* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da prikaže odabranog *putnika*“. Prekida se izvršenje scenarija. (IA)

11.1. Ukoliko **sistem** ne može da obriše *putnika* on prikazuje **zaposlenom** poruku “**Sistem** ne može da obriše *putnika*”. (IA)

### 1.2.6. SK 6: Slučаj korišćenjа – Kreiranje nove rezervacije (Složen slučaj korišćenja)

**Naziv SK**

Kreiranje *nove rezervacije*

**Aktori SK**

**Zaposleni**

**Učesnici SK**

**Zaposleni** i **sistem** (program)

**Preduslov**: **Sistem** je uključen i **zaposleni** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za kreiranje *nove* *rezervacije*. Učitana je *lista putnika* i *lista letova*.

**Osnovni scenario SK**

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

2. **Sistem** kreira *novu rezervaciju*. (SO)

3. **Sistem** prikazuje **zaposlenom** *novu rezervaciju* i poruku: “**Sistem** je kreirao *novu rezervaciju*“.(IA)

4. **Zaposleni** unosi podatke o *novoj rezervaciji*. (APUSO)

5. **Zaposleni** kontroliše da li je korektno uneo podatke o *novoj rezervaciji*. (ANSO)

6. **Zaposleni** poziva **sistem** da zapamti podatke o *rezervaciji*. (APSO)

7. **Sistem** pamti podatke o *rezervaciji*. (SO)

8. **Sistem** prikazuje **zaposlenom** zapamćenu *rezervaciju* i poruku: “**Sistem** je zapamtio *rezervaciju* ”. (IA)

**Alternativna scenarija**

3.1. Ukoliko **sistem** ne može da kreira *novu rezervaciju* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da kreira *novu rezervaciju* ”. Prekida se izvršenje scenarija. (IA)

8.1. Ukoliko **sistem** ne može da zapamti podatke o *rezervaciji* on prikazuje **zaposlenom** poruku “**Sistem** ne može da zapamti *rezervaciju.* (IA)

### 1.2.7. SK 7: Slučаj korišćenjа – Izmena rezervacije (Složen slučaj korišćenja)

**Nаziv SK**

Izmenа *rezervacije*

**Aktori SK**

**Zaposleni**

**Učesnici SK**

**Zaposleni** *i* **sistem** *(progrаm)*

**Preduslov**: **Sistem** je uključen i **zaposleni** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa *rezervacijama*. Učitana je *lista putnika* i *lista letova*.

**Osnovni scenаrio SK**

1. **Zaposleni** unosi vrednost po kojoj pretražuje *rezervacije*. (APUSO)

2. **Zaposleni** poziva **sistem** da nađe *rezervacije* po zadatoj vrednosti. (APSO)

3. **Sistem** pretražuje *rezervacije* po zadatoj vrednosti. (SO)

4. **Sistem** prikazuje **zaposlenom** *rezervacije* i poruku: “**Sistem** je našao rezervacije po zadatoj vrednosti ”. (IA)

5. **Zaposleni** bira *rezervaciju* koju želi da izmeni. (APUSO)

6. **Zaposleni** poziva **sistem** da učita odabranu *rezervaciju*. (APSO)

7. **Sistem** učitava podatke o odabranoj *rezervaciji*. (SO)

8. **Sistem** prikazuje **zaposlenom** podatke o *rezervaciji* i poruku: “Odabrana *rezervacija* je prikazana”. (IA)

9. **Zaposleni** unosi (menja) podatke o *rezervaciji*. (APUSO)

10. **Zaposleni** kontroliše da li je korektno uneo podatke o *rezervaciji*. (ANSO)

11. **Zaposleni** poziva **sistem** da zapamti podatke o *rezervaciji*. (APSO)

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

13. **Sistem** prikazuje **zaposlenom** zapamćenu *rezervaciju* i poruku: “**Sistem** je zapamtio *rezervaciju*.” (IA)

**Alternаtivnа scenаrijа**

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

8.1. Ukoliko **sistem** ne može da učita *rezervaciju* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da prikaže odabranu *rezervaciju*”. Prekida se izvršenje scenarija. (IA)

13.1. Ukoliko **sistem** ne može da zapamti podatke o *rezervaciji* on prikazuje **zaposlenom** poruku “**Sistem** ne može da zapamti *rezervaciju*”. (IA)

### 1.2.8. SK 8: Slučaj korišćenja – Pretraživanje rezervacija

**Naziv SK**

Pretraživanje *rezervacija*

**Aktori SK**

**Zaposleni**

**Učesnici SK**

**Zaposleni** i **sistem** (program)

**Preduslov**: **Sistem** je uključen i **zaposleni** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa *rezervacijama*. Učitana je *lista putnika*.

**Osnovni scenario SK**

1. **Zaposleni** unosi vrednost(tj. kriterijum) po kojoj pretražuje *rezervacije*. (APUSO)

2. **Zaposleni** poziva **sistem** da nađe *rezervacije* po zadatoj vrednosti. (APSO)

3. **Sistem** pretražuje *rezervacije* po zadatoj vrednosti. (SO)

4. **Sistem** prikazuje **zaposlenom** *rezervacije* i poruku: “**Sistem** je našao rezervacije po zadatoj vrednosti ”. (IA)

5. **Zaposleni** bira *rezervaciju* čije podatke želi da vidi. (APUSO)

6. **Zaposleni** poziva **sistem** da učita podatke o odabranoj *rezervaciji*. (APSO)

7. **Sistem** učitava podatke o odabranoj *rezervaciji*. (SO)

8. **Sistem** prikazuje **zaposlenom** podatke o odabranoj *rezervaciji* i poruku: “Odabrana *rezervacija* je prikazana.” (IA)

**Alternativna scenarija**

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

8.1. Ukoliko **sistem** ne može da učita podatke o odabranoj *rezervaciji* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da prikaže podatke o odabranoj *rezervaciji*.”(IA)

### 1.2.9. SK 9: Slučаj korišćenjа – Brisanje rezervacije

**Nаziv** **SK**

Brisanje rezervacije

**Aktori SK**

**Zaposleni**

**Učesnici SK**

**Zaposleni** i **sistem** (program)

**Preduslov**: **Sistem** je uključen i **zaposleni** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa *rezervacijom*. Učitana je *lista putnika* i *lista letova*.

**Osnovni scenario SK**

1. **Zaposleni** unosi vrednost po kojoj pretražuje *rezervacije*. (APUSO)

2. **Zaposleni** poziva **sistem** da nađe *rezervacije* po zadatoj vrednosti. (APSO)

3. **Sistem** pretražuje *rezervacije* po zadatoj vrednosti. (SO)

4. **Sistem** prikazuje **zaposlenom** *rezervacije* i poruku: “**Sistem** je našao rezervacije po zadatoj vrednosti ”. (IA)

5. **Zaposleni** bira *rezervaciju* koju želi da obriše. (APUSO)

6. **Zaposleni** poziva **sistem** da učita podatke o odabranoj *rezervaciji*. (APSO)

7. **Sistem** učitava podatke o odabranoj *rezervaciji*. (SO)

8. **Sistem** prikazuje **zaposlenom** podatke o *rezervaciji* i poruku: “Odabrana *rezervacija* je prikazana.” (IA)

9. **Zaposleni** poziva **sistem** da obriše *rezervaciju*. (APSO)

10. **Sistem** briše *rezervaciju*. (SO)

11. **Sistem** prikazuje **zaposlenom** poruku: “**Sistem** je obrisao *rezervaciju*.” (IA)

**Alternativna scenarija**

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

8.1. Ukoliko **sistem** ne može da učita odabranu *rezervaciju* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da prikaže odabranu *rezervaciju*“. Prekida se izvršenje scenarija. (IA)

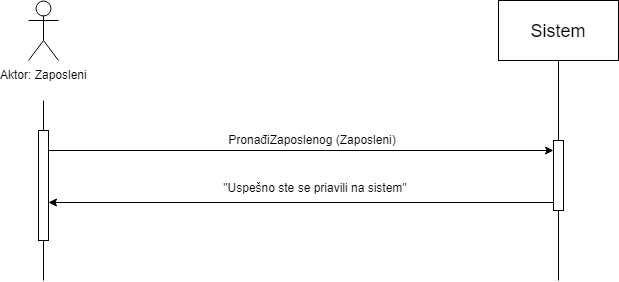
11.1. Ukoliko **sistem** ne može da obriše *rezervaciju* on prikazuje **zaposlenom** poruku “**Sistem** ne može da obriše *rezervaciju*”. (IA)

# 2. Analiza

## 2.1. Ponašanje softverskog sistema – Sistemski dijagram sekvenci

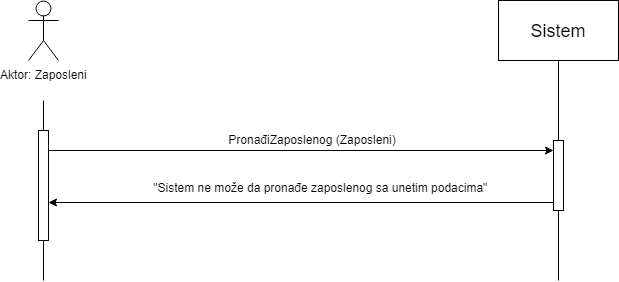
### 2.1.1. DS 1: Dijagram sekvenci slučaja korišćenja – Prijavljivanje zaposlenog

1. **Zaposleni** pozivа **sistem** da pronađe *zaposlenog* sa zadatim podacima. (APSO)
2. **Sistem** prikаzuje *zaposlenom* poruku: “Uspešno ste se prijavili na sistem”, a zatim mu omogućava pristup sistemu. (IA)



**Alternativna scenarija**

* 1. Ukoliko **sistem** ne može da pronađe zadatog *zaposlenog*, on prikazuje **zaposlenom** poruku “**Sistem** ne može da pronađe *zaposlenog* sa unetim podacima”. (IA)



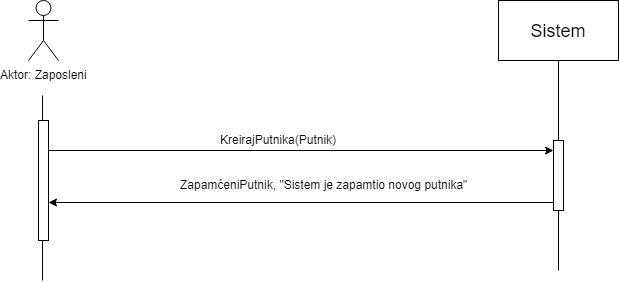
Sa navedenih sekvencnih dijagrama uočava se jedna sistemska operacija koju treba projektovati:

1. *signal* **PronadjiZaposlenog**(*Zaposleni*);

### 2.1.2. DS 2: Dijagram sekvenci slučaja korišćenja – Unos novog putnika

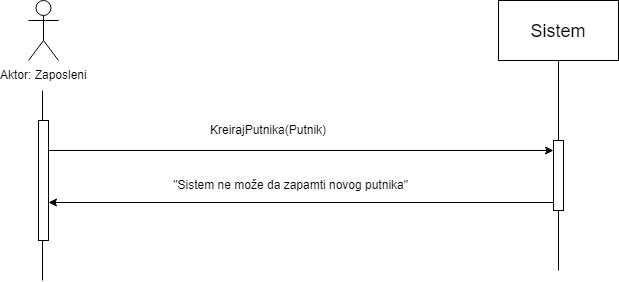
1. **Zaposleni** poziva **sistem** da zapamti podatke o *novom putniku*. (APSO)

2. **Sistem** prikazuje **zaposlenom** zapamćenog putnika i poruku: “**Sistem** je zapamtio *novog putnika*“.(IA)



**Alternativna scenarija**

2.1 Ukoliko **sistem** ne može da zapamti podatke o *novom putniku* on prikazuje **zaposlenom** poruku “**Sistem** ne može da zapamti *novog putnika*”. (IA)



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

1. *signal* **KreirajPutnika**(*Putnik*);

### 2.1.3. DS 3: Dijagram sekvenci slučaja korišćenja – Izmena podataka o putniku

1. **Zaposleni** poziva **sistem** da nađe *putnike* po zadatoj vrednosti. (APSO)

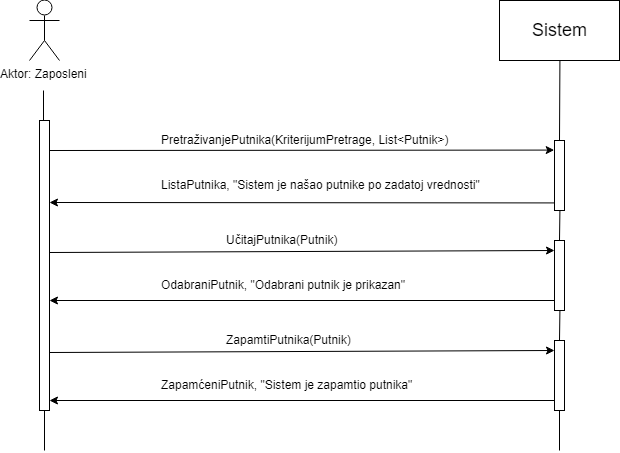
2. **Sistem** prikazuje **zaposlenom** *putnike* i poruku: “**Sistem** je našao putnike po zadatoj vrednosti ”. (IA)

3. **Zaposleni** poziva **sistem** da učita podatke o odabranom *putniku.* (APSO)

4. **Sistem** prikazuje **zaposlenom** podatke o *putniku* i poruku: “Odabrani *putnik* je prikazan.” (IA)

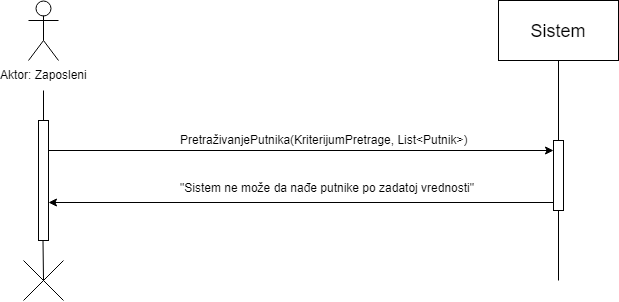
5. **Zaposleni** poziva **sistem** da zapamti podatke o *putniku*. (APSO)

6. **Sistem** prikazuje **zaposlenom** zapamćenog *putnika* i poruku: “**Sistem** je zapamtio *putnika*.” (IA)

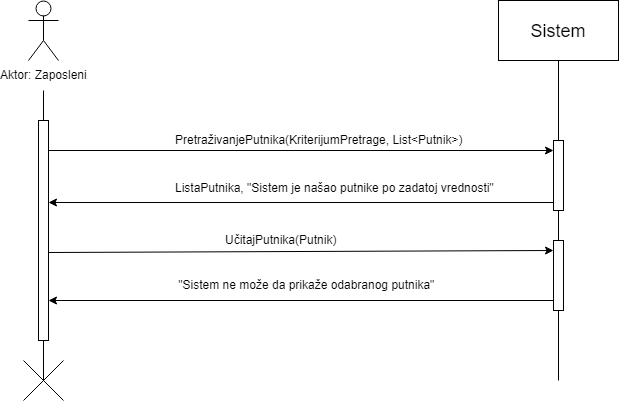


**Alternativna scenarija**

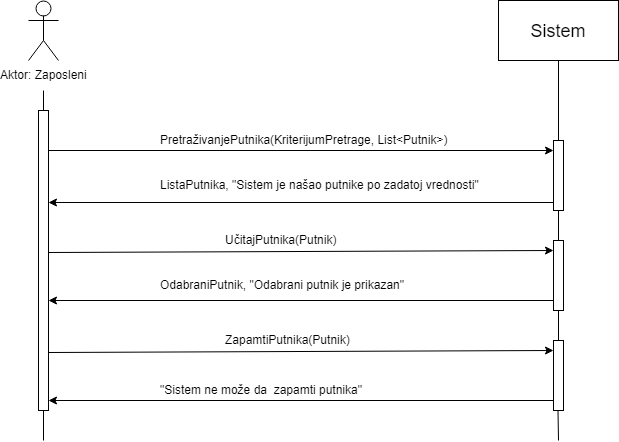
2.1 Ukoliko **sistem** ne može da nađe *putnike* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da nađe *putnike* po zadatoj vrednosti”. Prekida se izvršenje scenarija. (IA)



4.1 Ukoliko **sistem** ne može da učita *putnika* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da prikaže odabranog *putnika*”. Prekida se izvršenje scenarija. (IA)



6.1 Ukoliko **sistem** ne može da zapamti podatke o *putniku* on prikazuje **zaposlenom** poruku “**Sistem** ne može da zapamti *putnika*”. (IA)



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

1. *signal* **PretraživanjePutnika**(*KriterijumPretrage, List<Putnik>*);
2. *signal* **UčitajPutnika**(*Putnik*);
3. *signal* **ZapamtiPutnika**(*Putnik*);

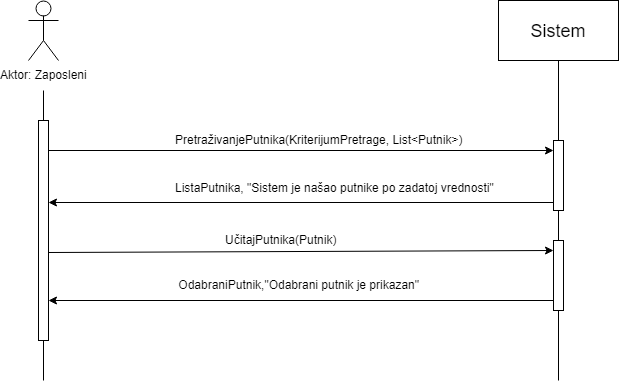
### 2.1.4. DS 4: Dijagram sekvenci slučaja korišćenja – Pretraživanje putnika

1. **Zaposleni** poziva **sistem** da nađe *putnike* po zadatoj vrednosti. (APSO)

2. **Sistem** prikazuje **zaposlenom** *putnike* i poruku: “**Sistem** je našao putnike po zadatoj vrednosti ”. (IA)

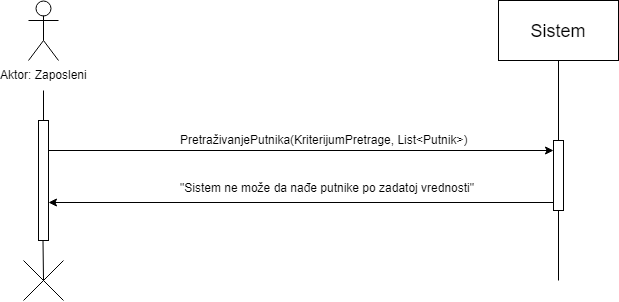
3. **Zaposleni** poziva **sistem** da učita podatke o odabranom *putniku*. (APSO)

4. **Sistem** prikazuje **zaposlenom** podatke o odabranom *putniku* i poruku: “Odabrani *putnik* je prikazan.” (IA)

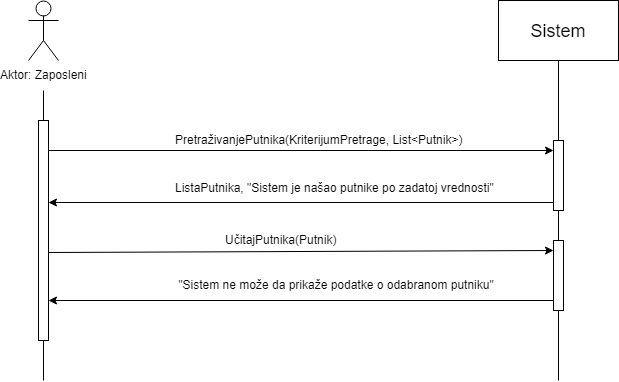
****

**Alternativna scenarija**

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



4.1. Ukoliko **sistem** ne može da učita podatke o odabranom *putniku* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da prikaže podatke o odabranom *putniku*.”(IA)



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

1. *signal* **PretraživanjePutnika**(*KriterijumPretrage, List<Putnik>*);
2. *signal* **UčitajPutnika**(*Putnik*);

### 2.1.5. DS 5: Dijagram sekvenci slučaja korišćenja – Brisanje putnika

1. **Zaposleni** poziva **sistem** da nađe *putnike* po zadatoj vrednosti. (APSO)

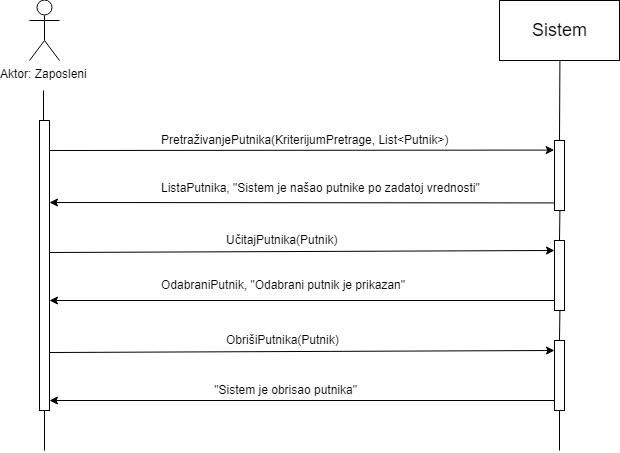
2. **Sistem** prikazuje **zaposlenom** *putnike* i poruku: “**Sistem** je našao putnike po zadatoj vrednosti ”. (IA)

3. **Zaposleni** poziva **sistem** da učita podatke o odabranom *putniku*. (APSO)

4. **Sistem** prikazuje **zaposlenom** podatke o *putniku* i poruku: “Odabrani *putnik* je prikazan.” (IA)

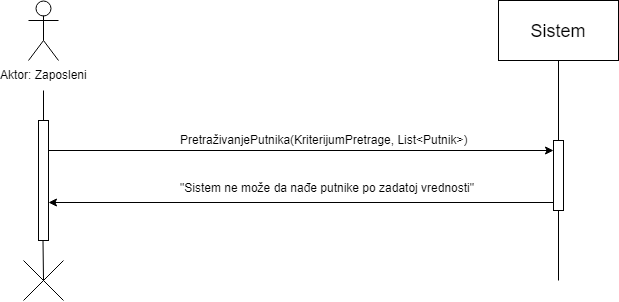
5. **Zaposleni** poziva **sistem** da obriše *putnika*. (APSO)

6. **Sistem** prikazuje **zaposlenom** poruku: “**Sistem** je obrisao *putnika*.” (IA)

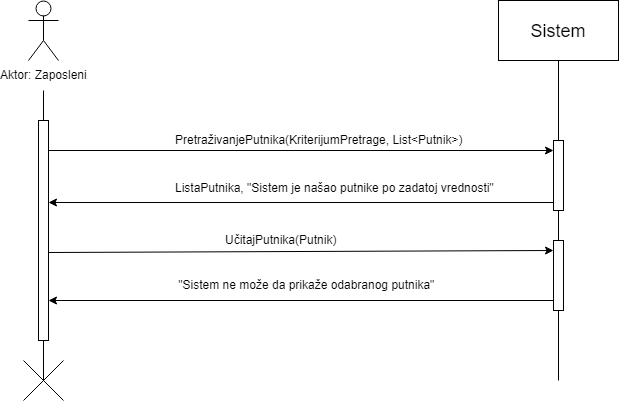


**Alternativna scenarija**

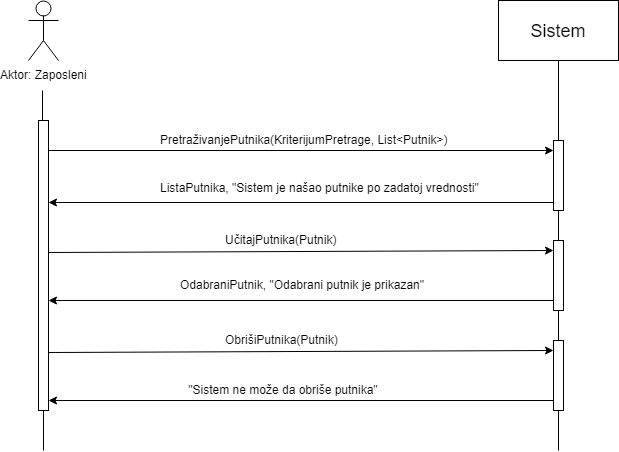
2.1 Ukoliko **sistem** ne može da nađe *putnike* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da nađe *putnike* po zadatoj vrednosti”. Prekida se izvršenje scenarija. (IA)



4.1 Ukoliko **sistem** ne može da učita *putnika* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da prikaže odabranog *putnika*“. Prekida se izvršenje scenarija. (IA)



6.1 Ukoliko **sistem** ne može da obriše *putnika* on prikazuje **zaposlenom** poruku “**Sistem** ne može da obriše *putnika*”. (IA)

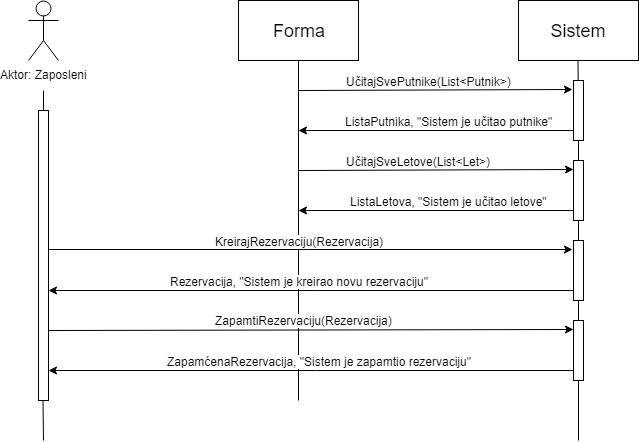


Sa navedenih sekvencnih dijagrama uočavaju se tri sistemske operacije koje treba projektovati:

1. *signal* **PretraživanjePutnika**(*KriterijumPretrage, List<Putnik>*);
2. *signal* **UčitajPutnika**(*Putnik*);
3. *signal* **ObrišiPutnika**(*Putnik*);

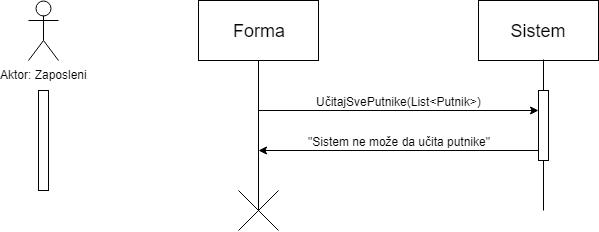
### 2.1.6. DS 6: Dijagram sekvenci slučaja korišćenja – Kreiranje rezervacije

1. **Forma** poziva **sistem** da učita *listu putnika*. (APSO)
2. **Sistem** vraća **formi** *listu putnika* i poruku: “**Sistem** je učitao putnike“. (IA)
3. **Forma** poziva **sistem** da učita *listu letova*. (APSO)
4. **Sistem** vraća **formi** *listu letova* i poruku: “**Sistem** je učitao letove“. (IA)
5. **Zaposleni** poziva **sistem** da kreira *novu rezervaciju*. (APSO)
6. **Sistem** prikazuje **zaposlenom** *novu rezervaciju* i poruku: “**Sistem** je kreirao *novu rezervaciju*“.(IA)
7. **Zaposleni** poziva **sistem** da zapamti podatke o *rezervaciji*. (APSO)
8. **Sistem** prikazuje **zaposlenom** zapamćenu *rezervaciju* i poruku: “**Sistem** je zapamtio *rezervaciju* ”. (IA)

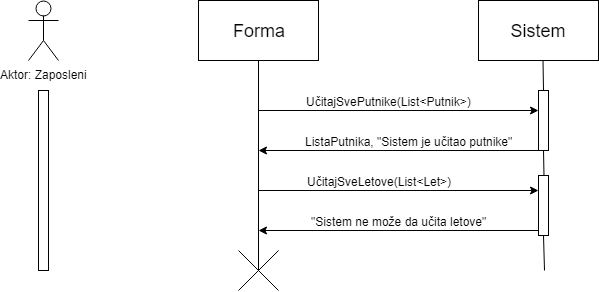
****

**Alternativna scenarija**

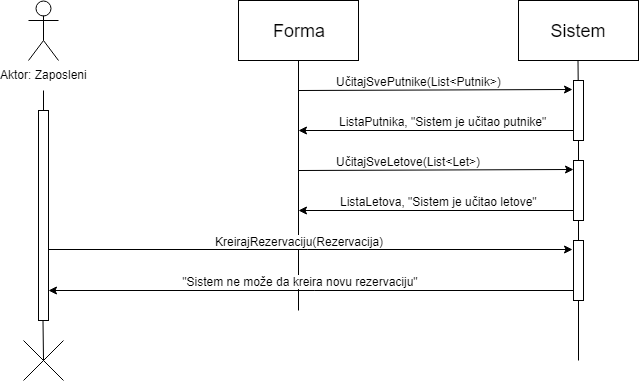
2.1 Ukoliko **sistem** ne može da učita *listu putnika* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da učita putnike”. Prekida se izvršenje scenarija. (IA)



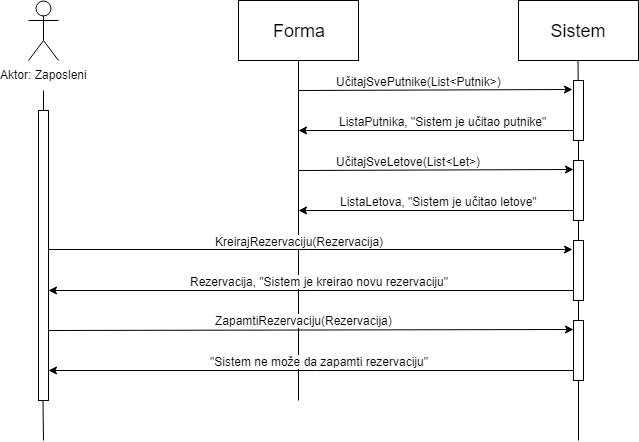
4.1. Ukoliko **sistem** ne može da učita *listu letova* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da učita letove”. Prekida se izvršenje scenarija. (IA)



6.1 Ukoliko **sistem** ne može da kreira *novu rezervaciju* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da kreira *novu rezervaciju* ”. Prekida se izvršenje scenarija. (IA)



* 1. Ukoliko **sistem** ne može da zapamti podatke o *rezervaciji* on prikazuje **zaposlenom** poruku “**Sistem** ne može da zapamti *rezervaciju.* (IA)

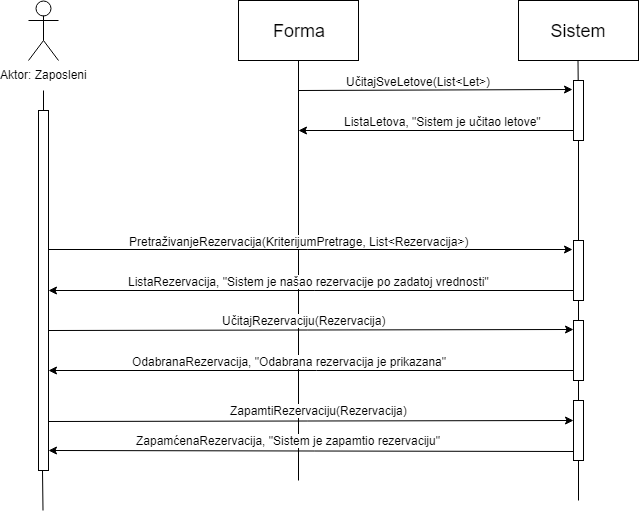


Sa navedenih sekvencnih dijagrama uočavaju se četiri sistemske operacije koje treba projektovati:

1. *signal* **UčitajSvePutnike**(*List<Putnik>*);
2. *signal* **UčitajSveLetove**(*List<Let>*);
3. *signal* **KreirajRezervaciju**(*Rezervacija*);
4. *signal* **ZapamtiRezervaciju**(*Rezervacija*);

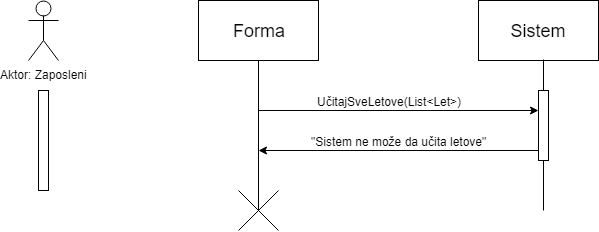
### 2.1.7. DS 7: Dijagram sekvenci slučaja korišćenja - Izmena rezervacije

1. **Forma** poziva **sistem** da učita *listu letova*. (APSO)
2. **Sistem** vraća **formi** *listu letova* i poruku: “**Sistem** je učitao letove“. (IA)
3. **Zaposleni** poziva **sistem** da nađe *rezervacije* po zadatoj vrednosti. (APSO)
4. **Sistem** prikazuje **zaposlenom** *rezervacije* i poruku: “**Sistem** je našao rezervacije po zadatoj vrednosti ”. (IA)
5. **Zaposleni** poziva **sistem** da učita odabranu *rezervaciju*. (APSO)
6. **Sistem** prikazuje **zaposlenom** podatke o *rezervaciji* i poruku: “Odabrana *rezervacija* je prikazana”. (IA)
7. **Zaposleni** poziva **sistem** da zapamti podatke o *rezervaciji*. (APSO)
8. **Sistem** prikazuje **zaposlenom** zapamćenu *rezervaciju* i poruku: “**Sistem** je zapamtio *rezervaciju*.” (IA)

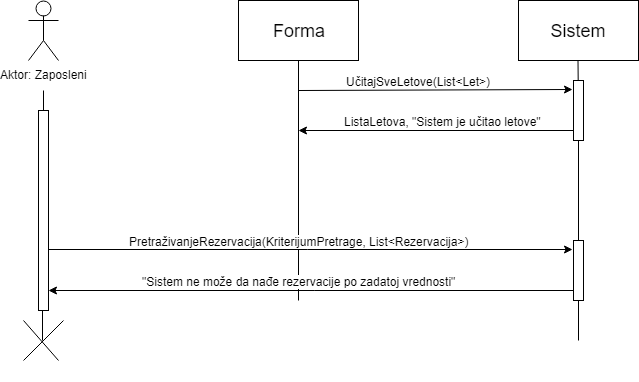


**Alternativna scenarija**

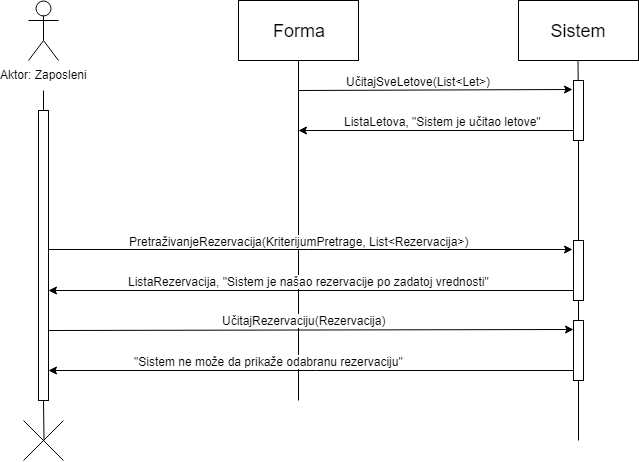
2.1. Ukoliko **sistem** ne može da učita *listu letova* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da učita letove”. Prekida se izvršenje scenaria. (IA)



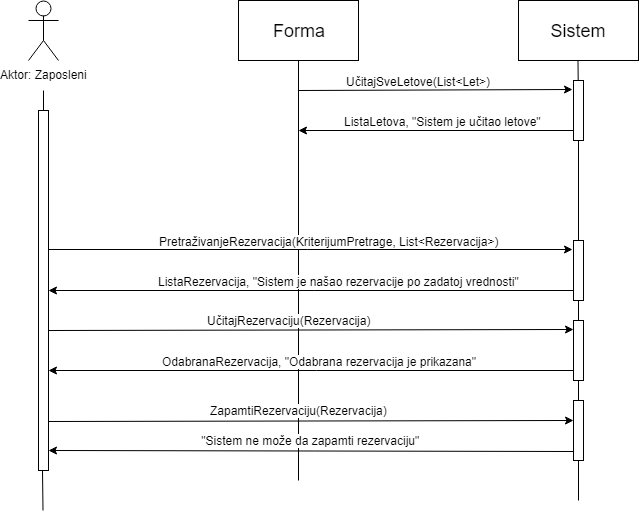
4.1 Ukoliko **sistem** ne može da nađe *rezervacije* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da nađe *rezervacije* po zadatoj vrednosti”. Prekida se izvršenje scenarija. (IA)



6.1 Ukoliko **sistem** ne može da učita *rezervaciju* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da prikaže odabranu *rezervaciju*”. Prekida se izvršenje scenarija. (IA)



8.1 Ukoliko **sistem** ne može da zapamti podatke o *rezervaciji* on prikazuje **zaposlenom** poruku “**Sistem** ne može da zapamti *rezervaciju*”. (IA)



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

1. *signal* **UčitajSveLetove**(*List<Let>*);
2. *signal* **PretraživanjeRezervacija**(*KriterijumPretrage,List<Rezervacija>*);
3. *signal* **UčitajRezervaciju**(*Rezervacija*);
4. *signal* **ZapamtiRezervaciju**(*Rezervacija*);

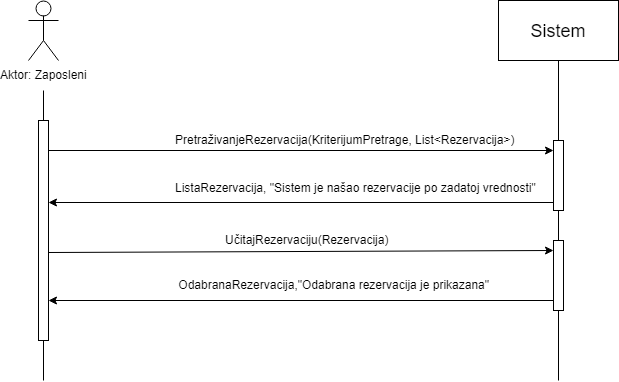
### 2.1.8. DS 8: Dijagram sekvenci slučaja korišćenja – Pretraživanje rezervacija

1. **Zaposleni** poziva **sistem** da nađe *rezervacije* po zadatoj vrednosti. (APSO)

2. **Sistem** prikazuje **zaposlenom** *rezervacije* i poruku: “**Sistem** je našao rezervacije po zadatoj vrednosti ”. (IA)

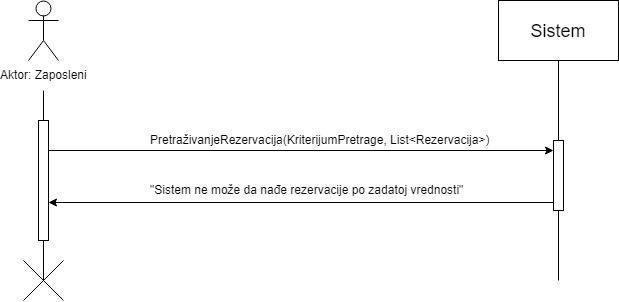
3. **Zaposleni** poziva **sistem** da učita podatke o odabranoj *rezervaciji*. (APSO)

4. **Sistem** prikazuje **zaposlenom** podatke o odabranoj *rezervaciji* i poruku: “Odabrana *rezervacija* je prikazana.” (IA)

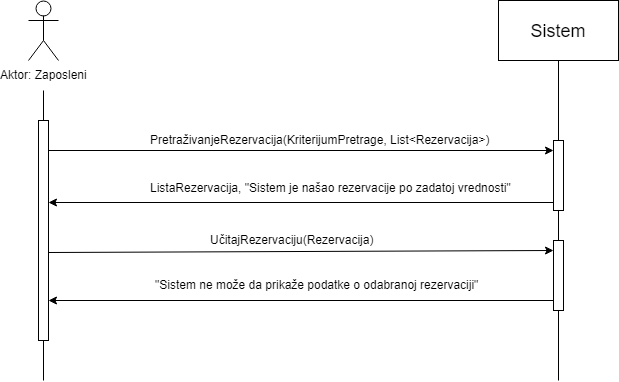


**Alternativna scenarija**

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



4.1 Ukoliko **sistem** ne može da učita podatke o odabranoj *rezervaciji* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da prikaže podatke o odabranoj *rezervaciji*.”(IA)



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

1. *signal* **PretraživanjeRezervacija**(*KriterijumPretrage,List<Rezervacija>*);
2. *signal* **UčitajRezervaciju**(*Rezervacija*);

### 2.1.9. DS 9: Dijagram sekvenci slučaja korišćenja – Brisanje rezervacije

1. **Zaposleni** poziva **sistem** da nađe *rezervacije* po zadatoj vrednosti. (APSO)

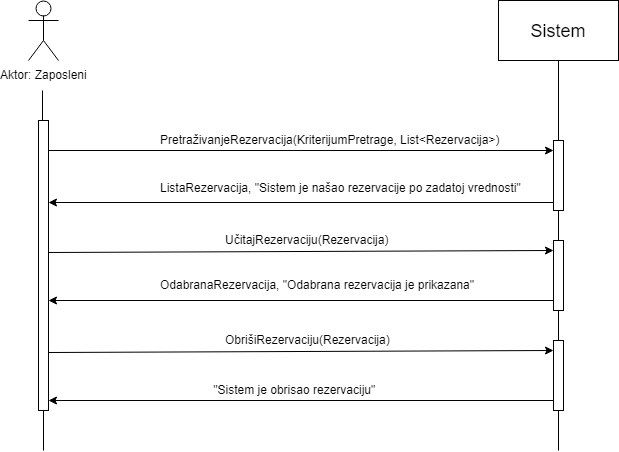
2. **Sistem** prikazuje **zaposlenom** *rezervacije* i poruku: “**Sistem** je našao rezervacije po zadatoj vrednosti ”. (IA)

3. **Zaposleni** poziva **sistem** da učita podatke o odabranoj *rezervaciji*. (APSO)

4. **Sistem** prikazuje **zaposlenom** podatke o *rezervaciji* i poruku: “Odabrana *rezervacija* je prikazana.” (IA)

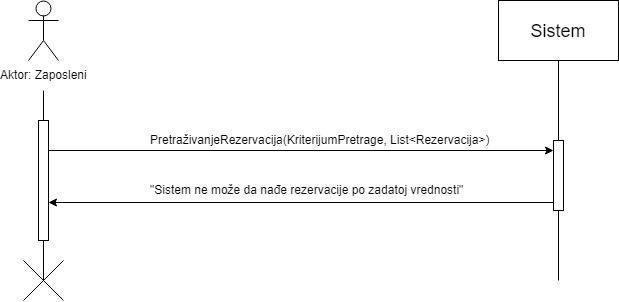
5. **Zaposleni** poziva **sistem** da obriše *rezervaciju*. (APSO)

6. **Sistem** prikazuje **zaposlenom** poruku: “**Sistem** je obrisao *rezervaciju*.” (IA)

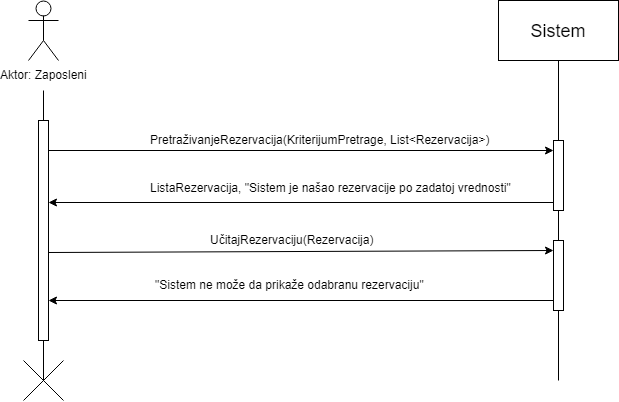


**Alternativna scenarija**

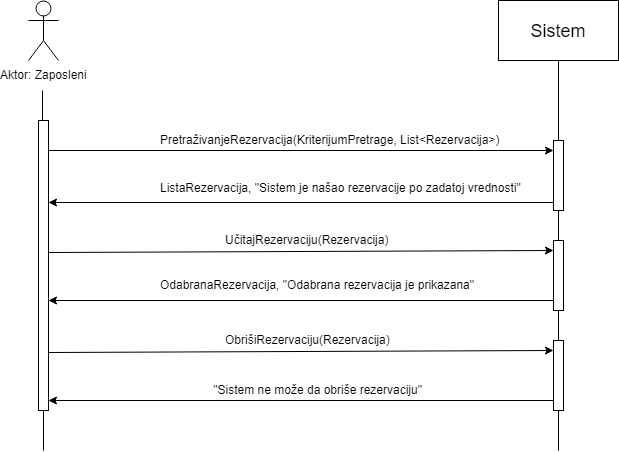
2.1 Ukoliko **sistem** ne može da nađe *rezervacije* on prikazuje zaposlenom poruku: “**Sistem** ne može da nađe *rezervacije* po zadatoj vrednosti”. Prekida se izvršenje scenarija (IA)



4.1 Ukoliko **sistem** ne može da učita odabranu *rezervaciju* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da prikaže odabranu *rezervaciju*“. Prekida se izvršenje scenarija. (IA)



6.1 Ukoliko **sistem** ne može da obriše *rezervaciju* on prikazuje **zaposlenom** poruku “**Sistem** ne može da obriše *rezervaciju*”. (IA)



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

1. *signal* **PretraživanjeRezervacija**(*KriterijumPretrage,List<Rezervacija>*);
2. *signal* **UčitajRezervaciju**(*Rezervacija*);
3. *signal* **ObrišiRezervaciju**(*Rezervacija*);

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

*signal* **PronađiZaposlenog**(*Zaposleni*);

*signal* **KreirajPutnika**(*Putnik*);

*signal* **ZapamtiPutnika**(*Putnik*);

*signal* **PretraživanjePutnika**(*KriterijumPretrage, List<Putnik>*);

*signal* **UčitajPutnika**(*Putnik*);

*signal* **ObrišiPutnika**(*Putnik*);

*signal* **UčitajSvePutnike**(*List<Putnik>*);

*signal* **UčitajSveLetove**(*List<Let>*);

*signal* **KreirajRezervaciju**(*Rezervacija*);

*signal* **ZapamtiRezervaciju**(*Rezervacija*);

*signal* **PretraživanjeRezervacija**(*KriterijumPretrage, List<Rezervacija>*);

*signal* **UčitajRezervaciju**(*Rezervacija*);

*signal* **ObrišiRezervaciju**(*Rezervacija*);

## 2.2. Ponašanje softverskog sistema – Definisanje ugovora o sistemskim operacijama

### 2.2.1. Ugovor UG 1:PronađiZaposlenog

Operacija: **PronađiZaposlenog** (*Zaposleni*):signal;

Veza sa SK: SK1

Preduslovi: -

Postuslovi: -

### 2.2.2. Ugovor UG 2:KreirajPutnika

Operacija: **KreirajPutnika** (Putnik):signal;

Veza sa SK: SK2

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

Postuslovi: *Kreiran je novi putnik.*

### 2.2.3. Ugovor UG 2:ZapamtiPutnika

Operacija: **ZapamtiPutnika** (Putnik):signal;

Veza sa SK: SK3

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

Postuslovi: *Podaci o putniku su zapamćeni.*

### 2.2.4. Ugovor UG 3:PretraživanjePutnika

Operacija: **PretraživanjePutnika** (KriterijumPretrage, List<Putnik>):signal;

Veza sa SK: SK3, SK4, SK5

Preduslovi: *-*

Postuslovi: *-*

### 2.2.5. Ugovor UG 4:UčitajPutnika

Operacija: **UčitajPutnika** (Putnik):signal;

Veza sa SK: SK3, SK4, SK5

Preduslovi: *-*

Postuslovi: *-*

### 2.2.6. Ugovor UG 5:ObrišiPutnika

Operacija: **ObrišiPutnika** (Putnik):signal;

Veza sa SK: SK5

Preduslovi: *Strukturna ograničenja nad objektom Putnik moraju biti zadovoljena.*

Postuslovi: *Putnik je obrisan.*

### 2.2.7. Ugovor UG 6:UčitajSvePutnike

Operacija: **PretražiSvePutnike** (*List<Putnik>*)

Veza sa SK: SK6

Preduslovi: -

Postuslovi: -

### 2.2.8. Ugovor UG 7:UčitajSveLetove

Operacija: **PretražiSveLetove** (List*<Let>*):signal;

Veza sa SK: SK6,SK7

Preduslovi: *-*

Postuslovi: *-*

### 2.2.9. Ugovor UG 8:KreirajRezervaciju

Operacija: **KreirajRezervaciju** (*Rezervacija*)

Veza sa SK: SK6

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

Postuslovi: Kreirana je nova rezervacija.

### 2.2.10. Ugovor UG 9:ZapamtiRezervaciju

Operacija: **ZapamtiRezervaciju** (Rezervacija):signal;

Veza sa SK: SK6,SK7

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

Postuslovi: *Podaci o rezervaciji su zapamćeni.*

### 2.2.11. Ugovor UG 10:PretraživanjeRezervacija

Operacija: **PretraživanjeRezervacija** (KriterijumPretrage, List<Rezervacija>)::signal;

Veza sa SK: SK7,SK8,SK9

Preduslovi: *-*

Postuslovi: *-*

### 2.2.12. Ugovor UG 11:UčitajRezervaciju

Operacija: **UčitajRezervaciju** (Rezervacija):signal;

Veza sa SK: SK7,SK8,SK9

Preduslovi: *-*

Postuslovi: *-*

### 2.2.13. Ugovor UG 12:ObrišiRezervaciju

Operacija: **ObrišiRezervaciju** (Rezervacija):signal;

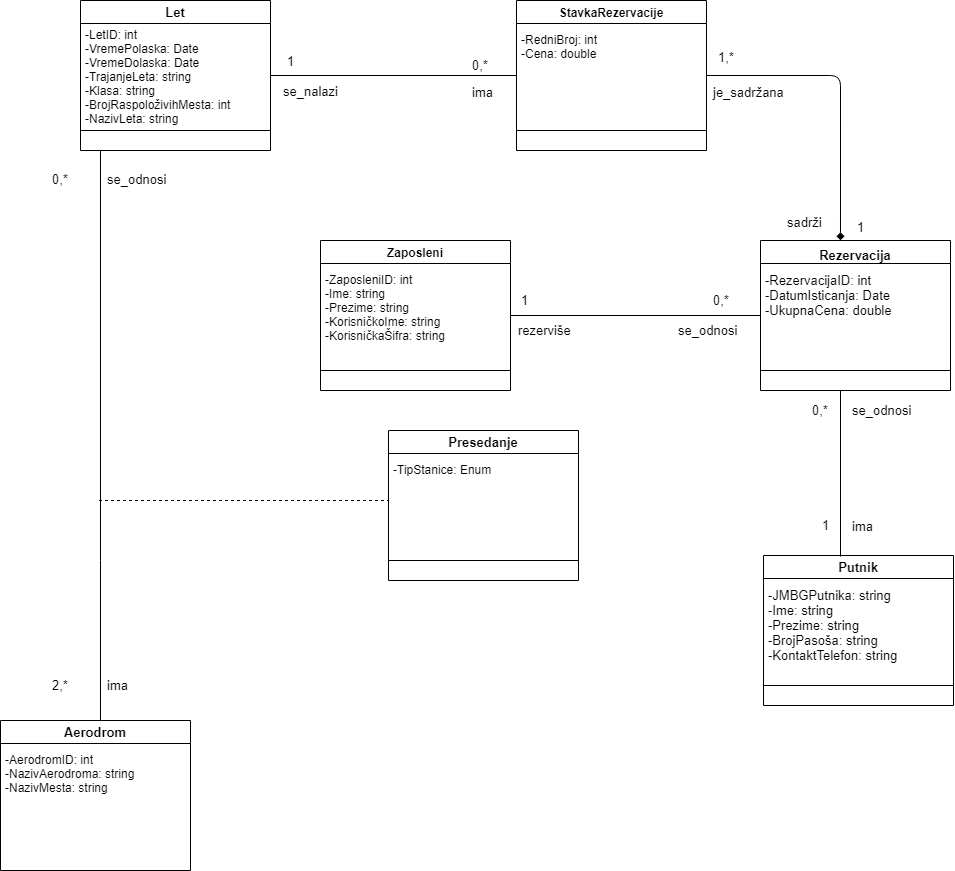
Veza sa SK: SK9

Preduslovi: *Strukturna ograničenja nad objektom Rezervacija moraju biti zadovoljena.*

Postuslovi: *Rezervacija je obrisana.*

## 2.3. Struktura softverskog sistema – Konceptualni (domenski) model

Struktura softverskog sistema opisana je pomoću konceptualnog modela.



Slika 2 - Konceptualni model softverskog sistema

## 2.4. Struktura softverskog sistema – Relacioni model

Let(LetID, VremePolaska, VremeDolaska, TrajanjeLeta, Klasa, BrojRaspoloživihMesta, NazivLeta)

Rezervacija(RezervacijaID, DatumIsticanja, UkupnaCena, *ZaposleniID*, *PutnikID*)

StavkaRezervacije(*RezervacijaID*, RedniBroj, Cena, *LetID*)

Zaposleni(ZaposleniID, Ime, Prezime, KorisničkoIme, KorisničkaŠifra)

Putnik(JMBGPutnika, Ime, Prezime, BrojPasoša, KontaktTelefon)

Aerodrom(AerodromID, NazivAerodroma, NazivMesta)

Presedanje(*AerodromID, LetID*, TipStanice)

Podvučena polja predstavljaju primarne ključeve odgovarajućih relacija, dok ”*italic*” polja predstavljaju spoljne ključeve na druge relacije.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Let** | | **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  StavkaRezervacije,Presedanje  DELETE  RESTRICTED  StavkaRezervacije, Presedanje |
| LetID | Integer | not null and > 0 |  |  |
| VremePolaska | Date | not null |  |  |
| VremeDolaska | Date | not null |  |  |
| TrajanjeLeta | String | not null |  |  |
| Klasa | String | not null |  |  |
| BrojRaspoloživihMesta | Integer | not null and >=0 |  |  |
| NazivLeta | String | not null |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Rezervacija** | | **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, Putnik  UPDATE  CASCADES  StavkaRezervacije,  RESTRICTED  Zaposleni,Putnik  DELETE  CASCADES  StavkaRezervacije |
| RezervacijaID | Integer | not null and > 0 |  |  |
| DatumIsticanja | Date | not null |  |  |
| UkupnaCena | Double | (default:0) |  | UkupnaCena=SUM(StavkaRezervacije.Cena) |
| ZaposleniID | Integer | not null and >0 |  |  |
| PutnikID | Integer | not null and > 0 |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela:StavkaRezervacije** | | **Prosto vrednosno ograničenje** | | **Složeno vrednosno ograničenje** | | **Strukturno ograničenje** |
| **Atributi** | **Ime** | **Tip atributa** | **Vrednost atributa** | **Međuzav. atributa jedne tabele** | **Međuzav.**  **atributa više tabela** | INSERT  RESTRICTED  Rezervacija,  Let  UPDATE  RESTRICTED  Rezervacija,  Let  DELETE / |
| RezervacijaID | Integer | not null and > 0 |  |  |
| RedniBroj | Integer | not null and > 0 |  |  |
| Cena | Double | >0(default:0) |  |  |
| LetID | Integer | not null and > 0 |  |  |

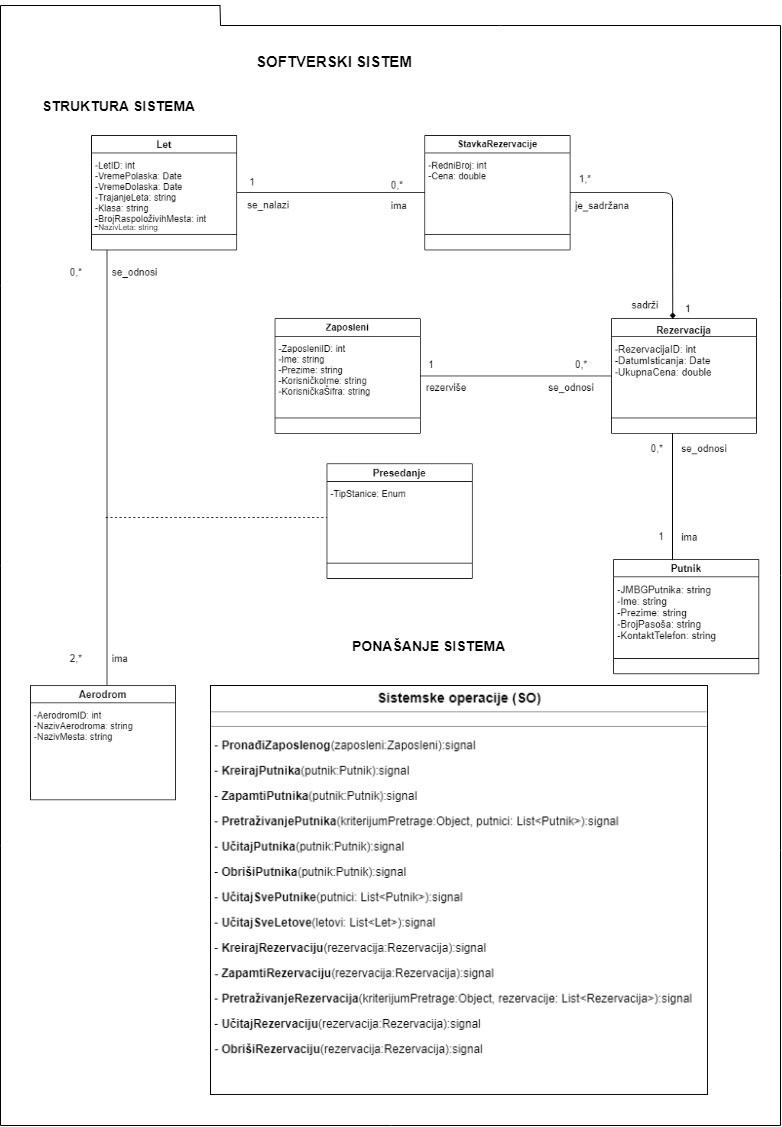
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Zaposleni** | | **Prosto vrednosno ograničenje** | | **Složeno vrednosno ograničenje** | | **Strukturno ograničenje** |
| **Atributi** | **Ime** | **Tip atributa** | **Vrednost atributa** | **Međuzav. atributa jedne tabele**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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Putnik** | | **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  Rezervacija  DELETE  RESTRICTED  Rezervacija |
| JMBGPutnika | String | not null |  |  |
| Ime | String | not null |  |  |
| Prezime | String | not null |  |  |
|  | BrojPasoša | String | not null |  |  |  |
|  | KontaktTelefon | String | not null |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Aerodrom** | | **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  Presedanje  DELETE  RESTRICTED  Presedanje |
| AerodromID | Integer | not null and > 0 |  |  |
| NazivAerodroma | String | not null |  |  |
|  | NazivMesta | String | not null |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Presedanje** | | **Prosto vrednosno ograničenje** | | **Složeno vrednosno ograničenje** | | **Strukturno ograničenje** |
| **Atributi** | **Ime** | **Tip atributa** | **Vrednost atributa** | **Međuzav. atributa jedne tabele** | **Međuzav.**  **atributa više tabela** | INSERT  RESTRICTED  Aerodrom, Let  UPDATE  RESTRICTED  Aerodrom, Let  DELETE / |
| TipStanice | Enum | not null |  |  |

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

**

Slika 3 - Struktura i ponašanje softverskog sistema

# 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 4 - Troslojna arhitektura

Skladište podataka

**Korisnički interfejs**

Ekranska forma

KontrolerKI

**Aplikaciona logika**

KontrolerAL

Poslovna logika

Database broker

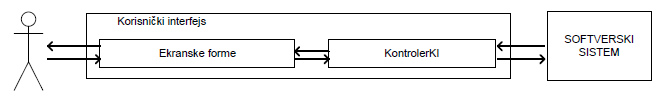
Slika 5 - *Troslojna arhitektura – detaljniji prikaz*

## 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.

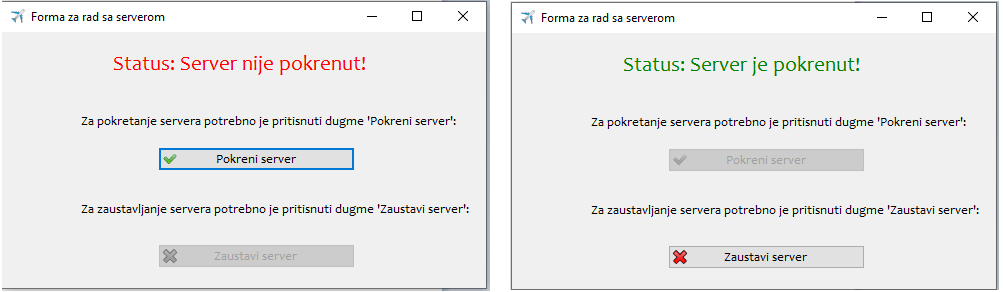


Slika 6 - Struktura korisnickog interfejsa

### 3.2.1. Projektovanje ekranskih formi

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

Glavna ekranska forma serverskog dela aplikacije treba da izgleda ovako:



Preko slučajeva korišćenja projektovane su i sve ostale ekranske forme koje će aplikacija posedovati, a koje se pozivaju iz menija glavne ekranske forme klijentskog dela aplikacije.

### SK 1: Slučаj korišćenjа – Prijavljivanje zaposlenog

**Nаziv SK**

Prijavljivanje *zaposlenog*

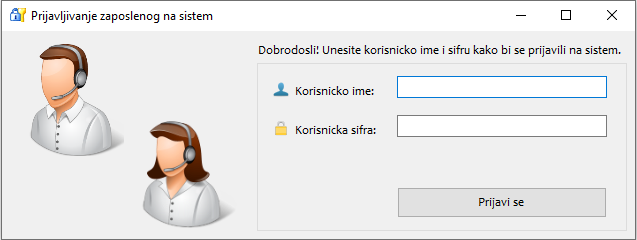
**Aktori SK**

**Zaposleni**

**Učesnici SK**

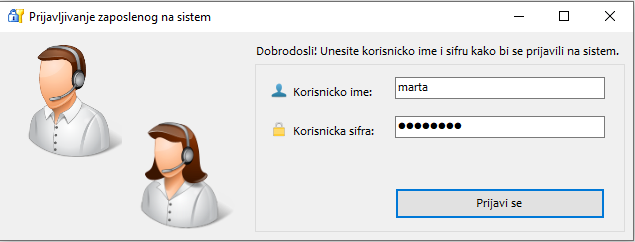
**Zaposleni**i**sistem** *(progrаm)*

**Preduslov**: **Sistem** je uključen i prikаzuje formu zа prijavljivanje *zaposlenog*.

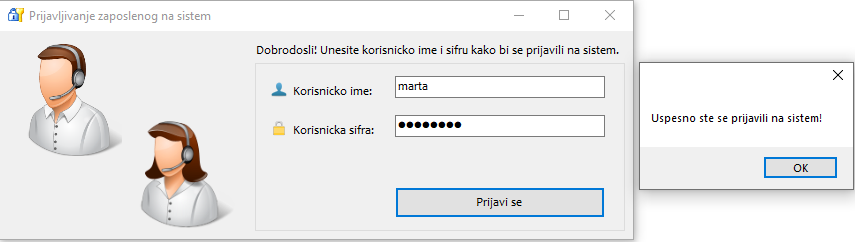


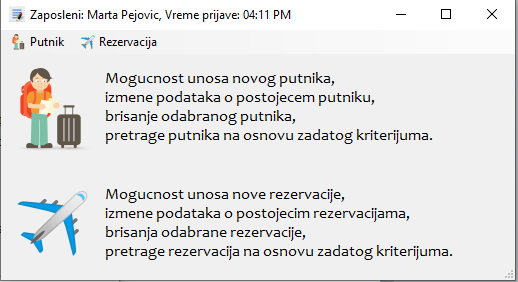
**Osnovni scenаrio SK**

1. **Zaposleni** unosi identifikacione podatke tj. korisničko ime i šifru. (APUSO)

****

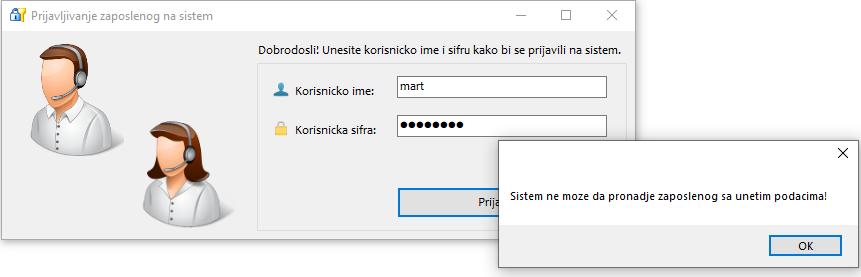
1. **Zaposleni** kontroliše dа li je korektno uneo identifikacione podаtke tj. korisničko ime i sifru. (ANSO)
2. **Zaposleni** pozivа **sistem** da pronađe *zaposlenog* sa zadatim podacima. (APSO)
3. **Sistem** pretražuje *zaposlenog*. (SO)
4. **Sistem** prikаzuje *zaposlenom* poruku: “Uspešno ste se prijavili na sistem”, a zatim mu omogućava pristup sistemu. (IA)





**Alternаtivnа scenаrijа**

5.1. Ukoliko **sistem** ne može da pronađe zadatog *zaposlenog*, on prikazuje **zaposlenom** poruku “**Sistem** ne može da pronađe *zaposlenog* sa unetim podacima”. (IA)



### SK 2: Slučаj korišćenjа – Unos novog putnika

**Nаziv SK**

Unos *novog putnika*

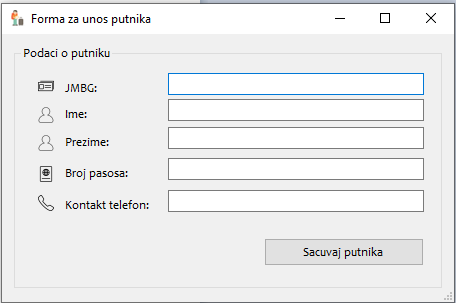
**Aktori SK**

**Zaposleni**

**Učesnici SK**

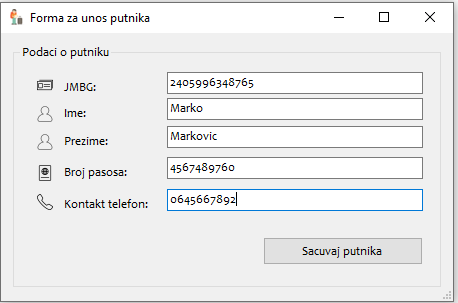
**Zaposleni**i**sistem** *(progrаm)*

**Preduslov**: **Sistem** je uključen i **zaposleni** je ulogovan pod svojom šifrom. Sistem prikazuje formu za unos *novog putnika*.



**Osnovni scenаrio SK**

1. **Zaposleni** unosi podatke o *novom putniku*. (APUSO)

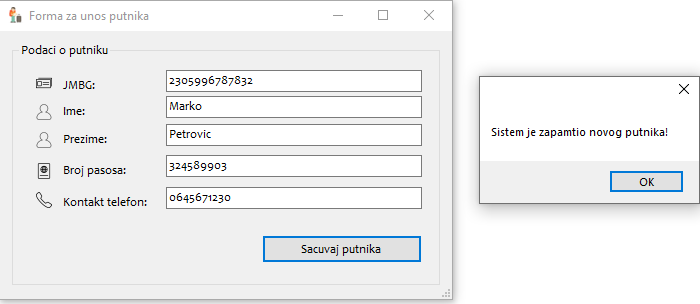


2. **Zaposleni** kontroliše da li je korektno uneo podatke o *novom putniku*. (ANSO)

3. **Zaposleni** poziva **sistem** da zapamti podatke o *novom putniku*. (APSO)

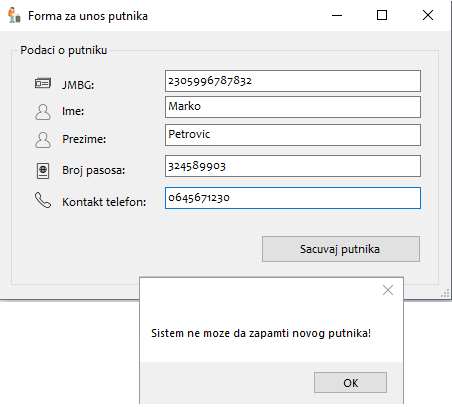
4. **Sistem** pamti podatke o *novom putniku*. (SO)

5. **Sistem** prikazuje **zaposlenom** zapamćenog putnika i poruku: “**Sistem** je zapamtio *novog putnika*“.(IA)



**Alternаtivnа scenаrijа**

5.1. Ukoliko **sistem** ne može da zapamti podatke o *novom putniku* on prikazuje **zaposlenom** poruku “**Sistem** ne može da zapamti *novog putnika*”. (IA)



### SK 3: Slučаj korišćenjа – Izmenа podataka o putniku

**Nаziv SK**

Izmenа podataka o *putniku*

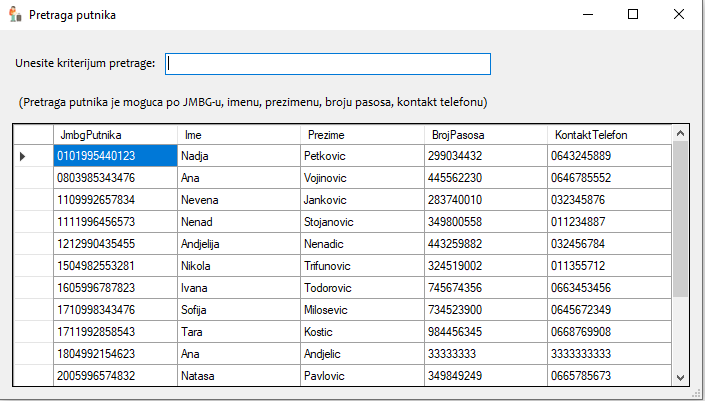
**Aktori SK**

**Zaposleni**

**Učesnici SK**

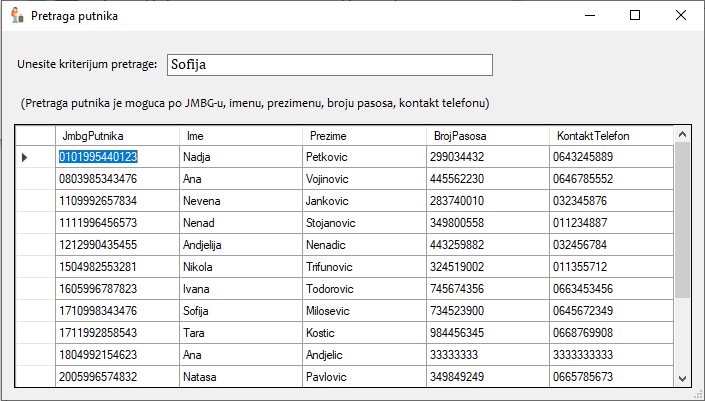
**Zaposleni**i**sistem** *(progrаm)*

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



**Osnovni scenаrio SK**

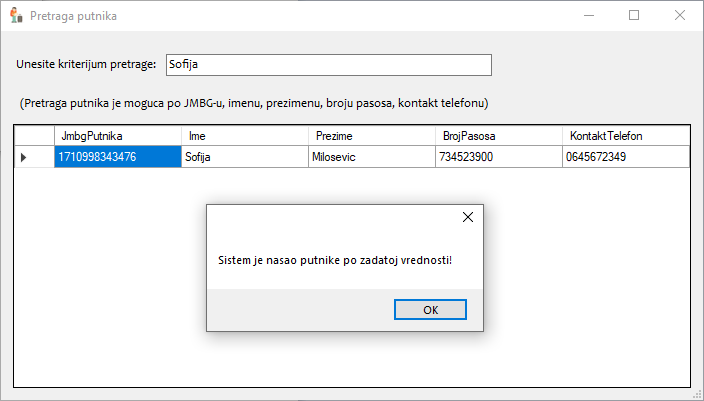
1. **Zaposleni** unosi vrednost po kojoj pretražuje *putnike*. (APUSO)



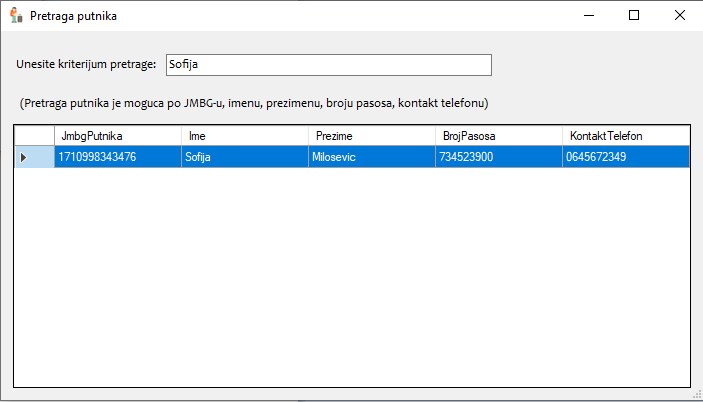
2. **Zaposleni** poziva **sistem** da nađe *putnike* po zadatoj vrednosti. (APSO)

3. **Sistem** traži *putnike* po zadatoj vrednosti. (SO)

4. **Sistem** prikazuje **zaposlenom** *putnike* i poruku: “**Sistem** je našao putnike po zadatoj vrednosti ”. (IA)



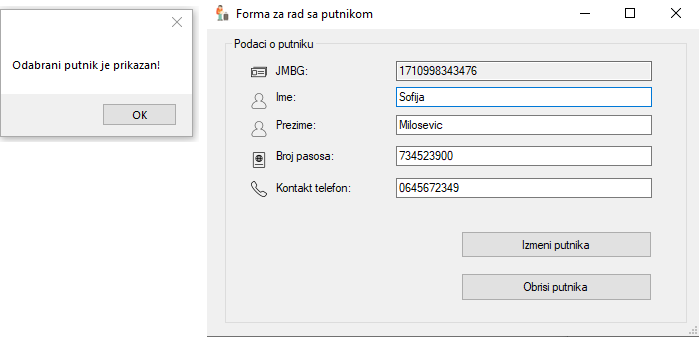
5. **Zaposleni** bira *putnika* čije podatke želi da izmeni. (APUSO)



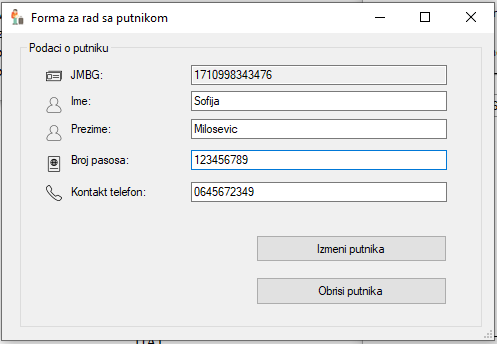
6. **Zaposleni** poziva **sistem** da učita podatke o odabranom *putniku.* (APSO)

7. **Sistem** učitava podatke o odabranom *putniku*. (SO)

8. **Sistem** prikazuje **zaposlenom** podatke o *putniku* i poruku: “Odabrani *putnik* je prikazan.” (IA)



9. **Zaposleni** unosi (menja) podatke o *putniku*. (APUSO)

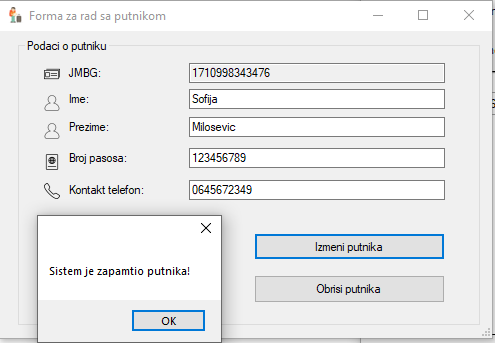


10. **Zaposleni** kontroliše da li je korektno uneo podatke o *putniku*. (ANSO)

11. **Zaposleni** poziva **sistem** da zapamti podatke o *putniku*. (APSO)

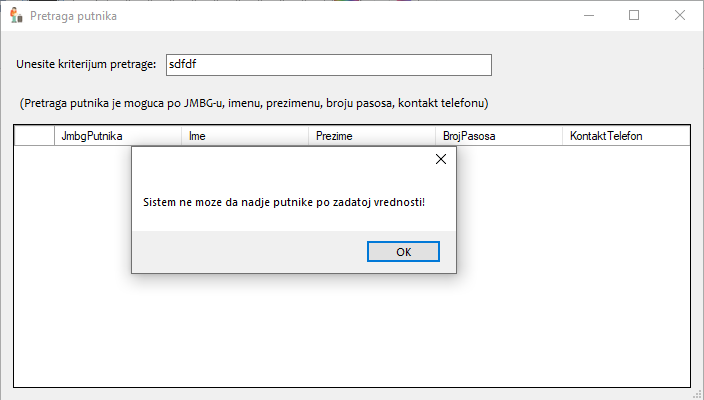
12. **Sistem** pamti podatke o *putniku*. (SO)

13. **Sistem** prikazuje **zaposlenom** zapamćenog *putnika* i poruku: “**Sistem** je zapamtio *putnika*.” (IA)

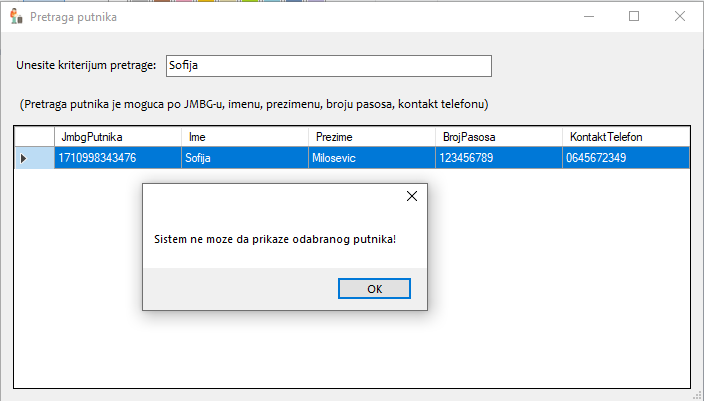


**Alternаtivnа scenаrijа**

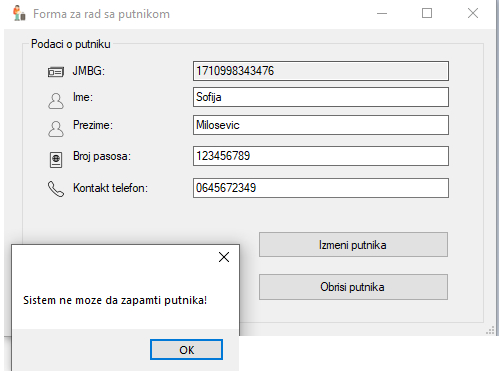
4.1. Ukoliko **sistem** ne može da nađe *putnike* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da nađe *putnike* po zadatoj vrednosti”. Prekida se izvršenje scenarija. (IA)



8.1. Ukoliko **sistem** ne može da učita *putnika* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da prikaže odabranog *putnika*”. Prekida se izvršenje scenarija. (IA)



13.1. Ukoliko **sistem** ne može da zapamti podatke o *putniku* on prikazuje **zaposlenom** poruku “**Sistem** ne može da zapamti *putnika*”. (IA)



### SK 4: Slučаj korišćenjа – Pretraživanje putnika

**Naziv SK**

Pretraživanje *putnika*

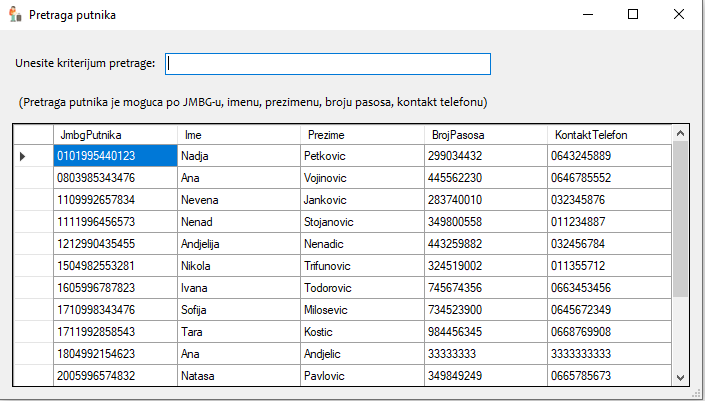
**Aktori SK**

**Zaposleni**

**Učesnici SK**

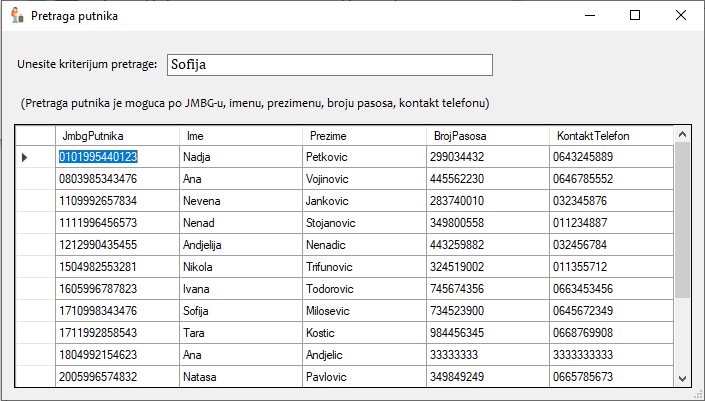
**Zaposleni** i **sistem** (program)

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



**Osnovni scenario SK**

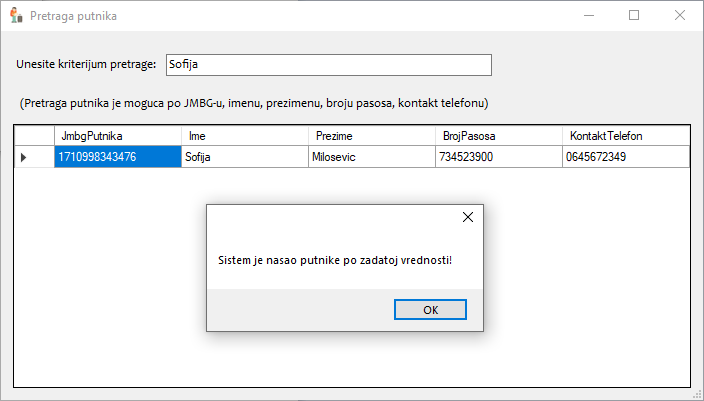
1. **Zaposleni** unosi vrednost(tj. kriterijum) po kojoj pretražuje *putnike*. (APUSO)



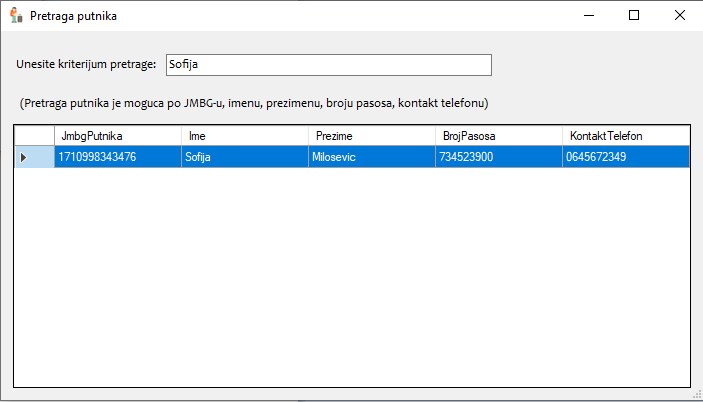
2. **Zaposleni** poziva **sistem** da nađe *putnike* po zadatoj vrednosti. (APSO)

3. **Sistem** pretražuje *putnike* po zadatoj vrednosti. (SO)

4. **Sistem** prikazuje **zaposlenom** *putnike* i poruku: “**Sistem** je našao putnike po zadatoj vrednosti ”. (IA)



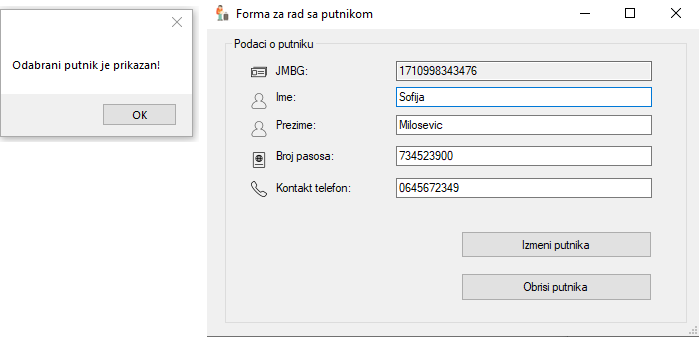
5. **Zaposleni** bira *putnika* čije podatke želi da vidi. (APUSO)



6. **Zaposleni** poziva **sistem** da učita podatke o odabranom *putniku*. (APSO)

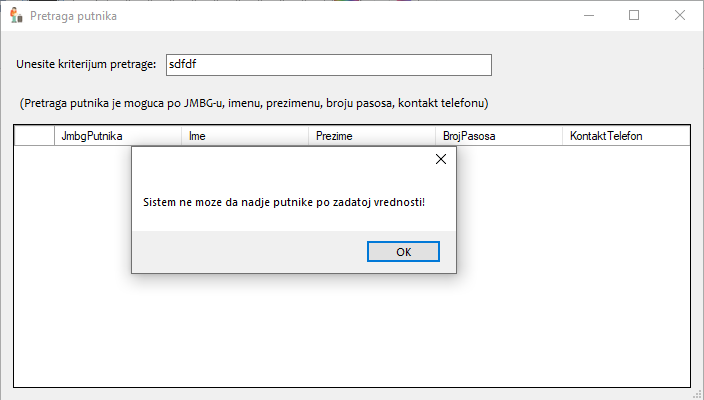
7. **Sistem** učitava podatke o odabranom *putniku*. (SO)

8. **Sistem** prikazuje **zaposlenom** podatke o odabranom *putniku* i poruku: “Odabrani *putnik* je prikazan.” (IA)

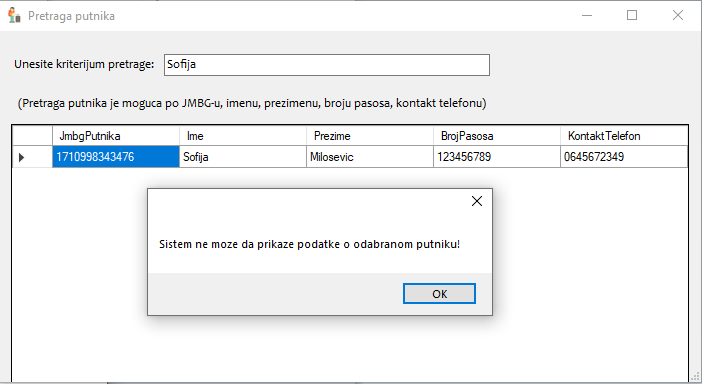


**Alternativna scenarija**

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



8.1. Ukoliko **sistem** ne može da učita podatke o odabranom *putniku* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da prikaže podatke o odabranom *putniku*.”(IA)



### SK 5: Slučаj korišćenjа – Brisanje putnika

**Naziv SK**

Brisanje *putnika*

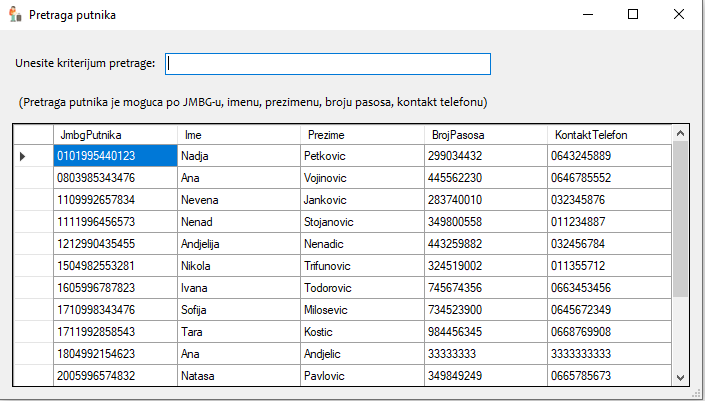
**Aktori SK**

**Zaposleni**

**Učesnici SK**

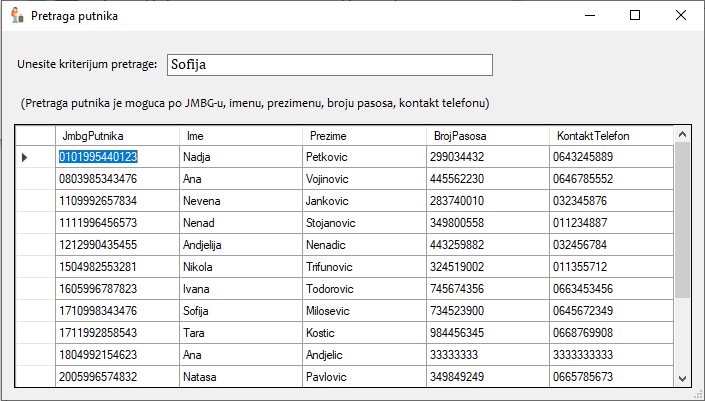
**Zaposleni** i **sistem** (program)

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



**Osnovni scenario SK**

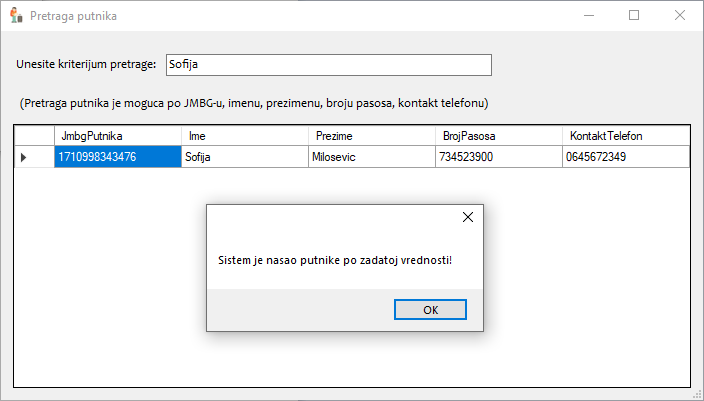
1. **Zaposleni** unosi vrednost po kojoj pretražuje *putnike*. (APUSO)



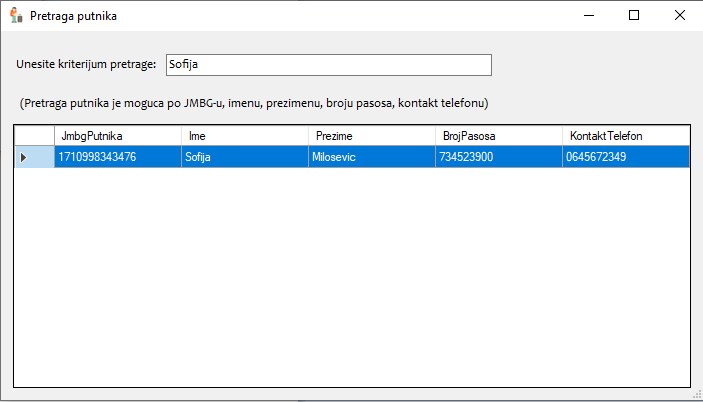
2. **Zaposleni** poziva **sistem** da nađe *putnike* po zadatoj vrednosti. (APSO)

3. **Sistem** traži *putnike* po zadatoj vrednosti. (SO)

4. **Sistem** prikazuje **zaposlenom** *putnike* i poruku: “**Sistem** je našao putnike po zadatoj vrednosti ”. (IA)



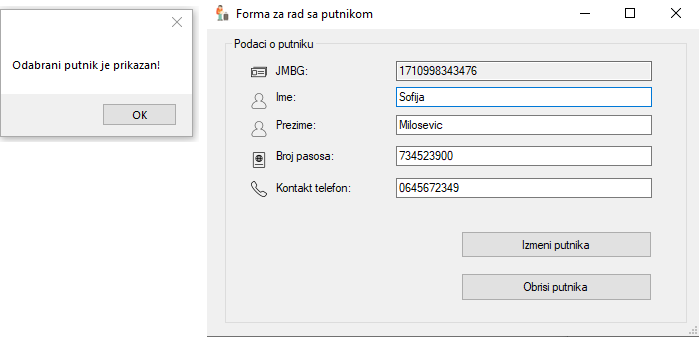
5. **Zaposleni** bira *putnika* kojeg želi da obriše. (APUSO)



6. **Zaposleni** poziva **sistem** da učita podatke o odabranom *putniku*. (APSO)

7. **Sistem** učitava podatke o odabranom *putniku*. (SO)

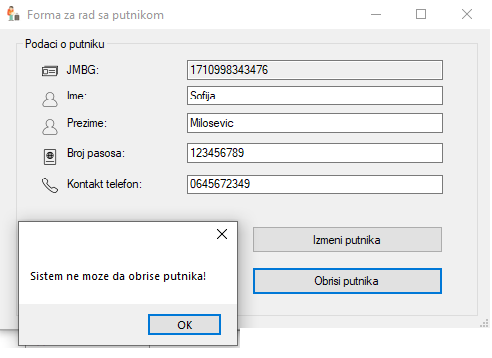
8. **Sistem** prikazuje **zaposlenom** podatke o *putniku* i poruku: “Odabrani *putnik* je prikazan.” (IA)



9. **Zaposleni** poziva **sistem** da obriše *putnika*. (APSO)

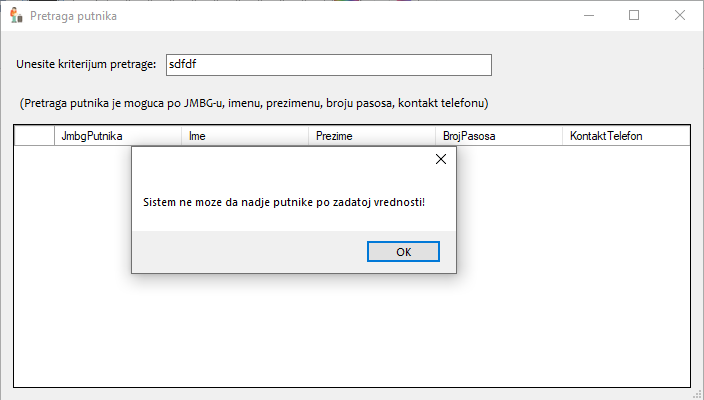
10. **Sistem** briše *putnika*. (SO)

11. **Sistem** prikazuje **zaposlenom** poruku: “**Sistem** je obrisao *putnika*.” (IA)

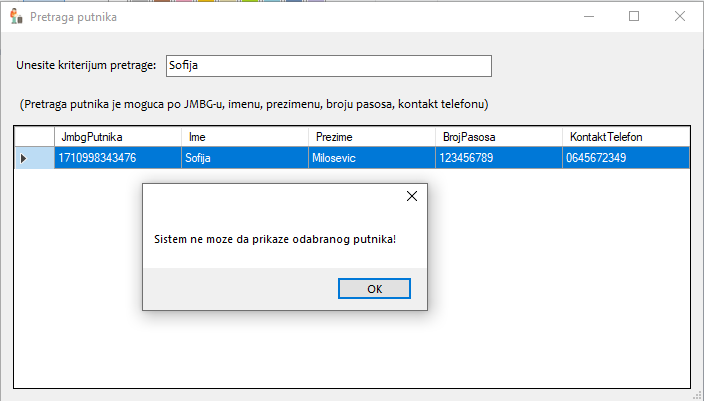


**Alternativna scenarija**

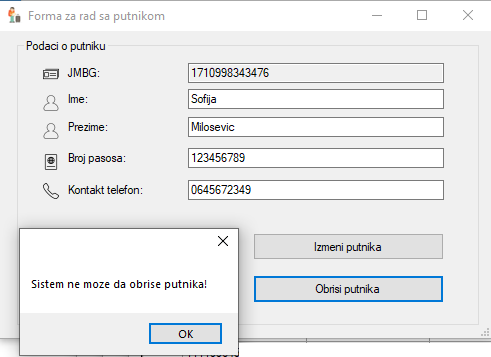
4.1. Ukoliko **sistem** ne može da nađe *putnike* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da nađe *putnike* po zadatoj vrednosti”. Prekida se izvršenje scenarija. (IA)



8.1. Ukoliko **sistem** ne može da učita *putnika* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da prikaže odabranog *putnika*“. Prekida se izvršenje scenarija. (IA)



11.1. Ukoliko **sistem** ne može da obriše *putnika* on prikazuje **zaposlenom** poruku “**Sistem** ne može da obriše *putnika*”. (IA)



### SK 6: Slučаj korišćenjа – Kreiranje nove rezervacije (Složen slučaj korišćenja)

**Naziv SK**

Kreiranje *nove rezervacije*

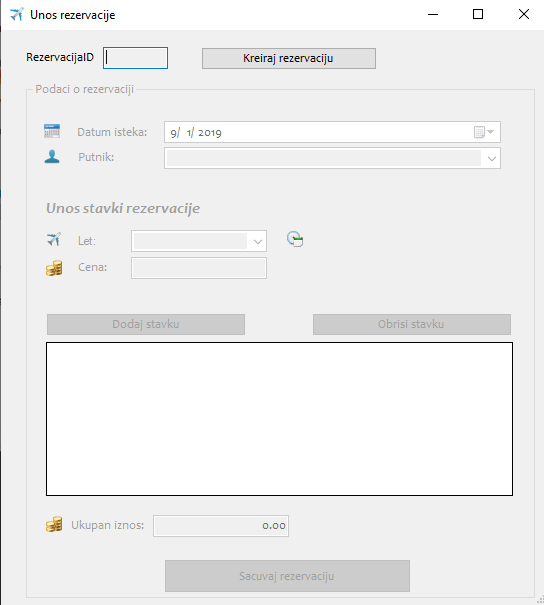
**Aktori SK**

**Zaposleni**

**Učesnici SK**

**Zaposleni** i **sistem** (program)

**Preduslov**: **Sistem** je uključen i **zaposleni** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za kreiranje *nove* *rezervacije*. Učitana je *lista putnika* i *lista letova*.

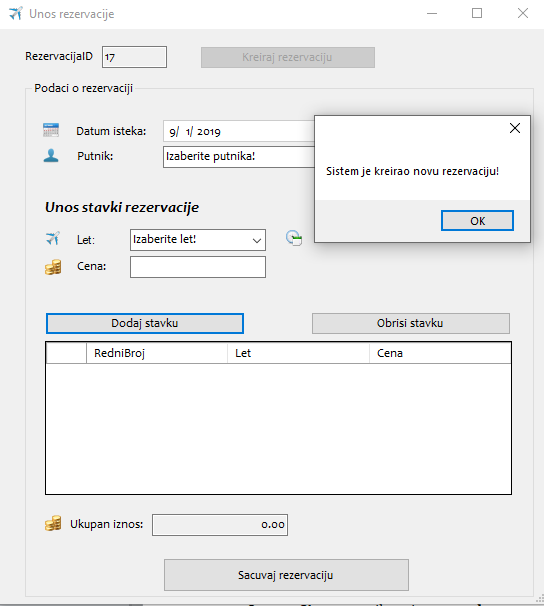


**Osnovni scenario SK**

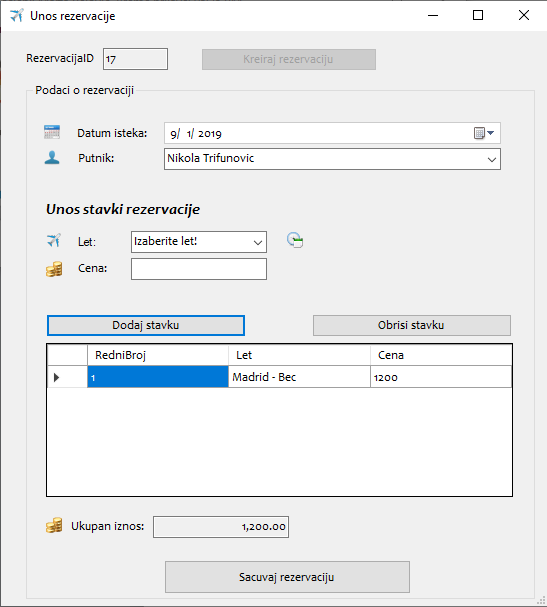
1. **Zaposleni** poziva **sistem** da kreira *novu rezervaciju*. (APSO)

2. **Sistem** kreira *novu rezervaciju*. (SO)

3. **Sistem** prikazuje **zaposlenom** *novu rezervaciju* i poruku: “**Sistem** je kreirao *novu rezervaciju*“.(IA)



4. **Zaposleni** unosi podatke o *novoj rezervaciji*. (APUSO)

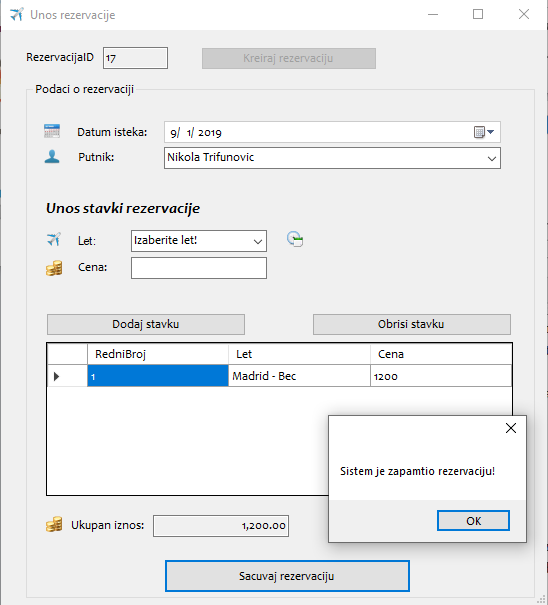


5. **Zaposleni** kontroliše da li je korektno uneo podatke o *novoj rezervaciji*. (ANSO)

6. **Zaposleni** poziva **sistem** da zapamti podatke o *rezervaciji*. (APSO)

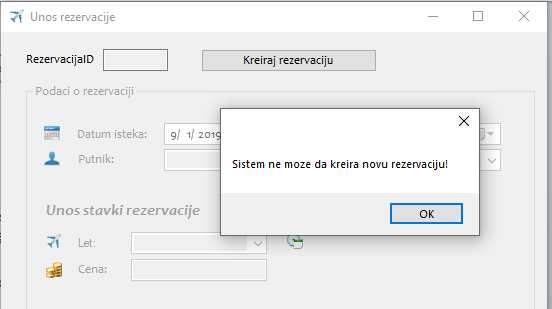
7. **Sistem** pamti podatke o *rezervaciji*. (SO)

8. **Sistem** prikazuje **zaposlenom** zapamćenu *rezervaciju* i poruku: “**Sistem** je zapamtio *rezervaciju* ”. (IA)

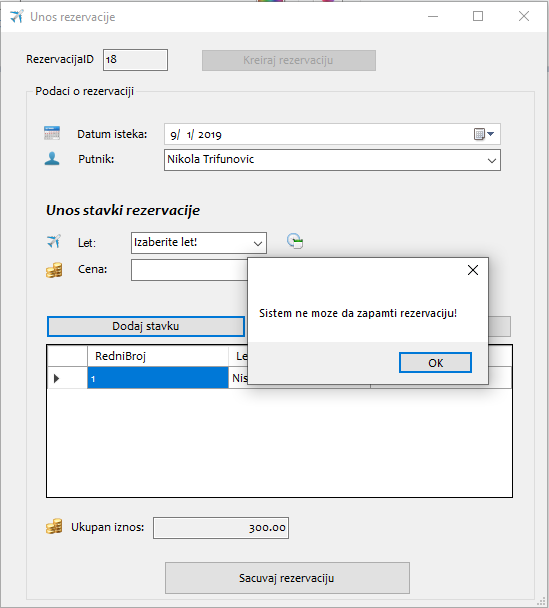


**Alternativna scenarija**

3.1. Ukoliko **sistem** ne može da kreira *novu rezervaciju* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da kreira *novu rezervaciju* ”. Prekida se izvršenje scenarija. (IA)



8.1. Ukoliko **sistem** ne može da zapamti podatke o *rezervaciji* on prikazuje **zaposlenom** poruku “**Sistem** ne može da zapamti *rezervaciju.* (IA)



### SK 7: Slučаj korišćenjа – Izmena rezervacije (Složen slučaj korišćenja)

**Nаziv SK**

Izmenа *rezervacije*

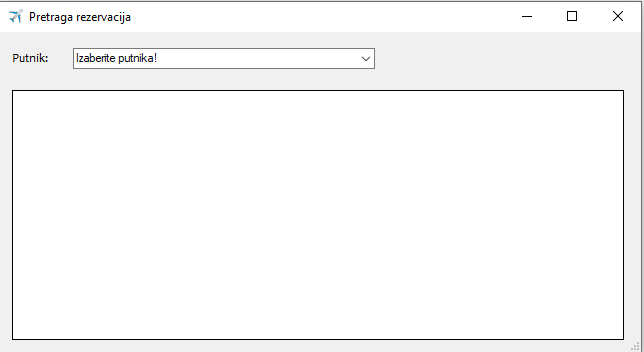
**Aktori SK**

**Zaposleni**

**Učesnici SK**

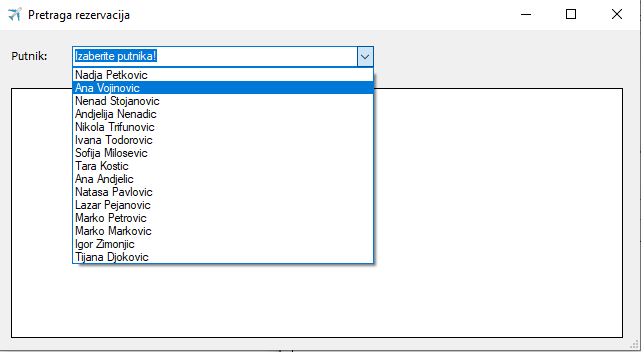
**Zaposleni** *i* **sistem** *(progrаm)*

**Preduslov**: **Sistem** je uključen i **zaposleni** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa *rezervacijama*. Učitana je *lista putnika* i *lista letova*.



**Osnovni scenаrio SK**

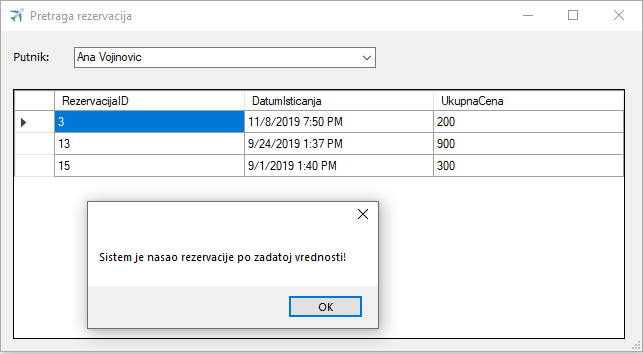
1. **Zaposleni** unosi vrednost po kojoj pretražuje *rezervacije*. (APUSO)



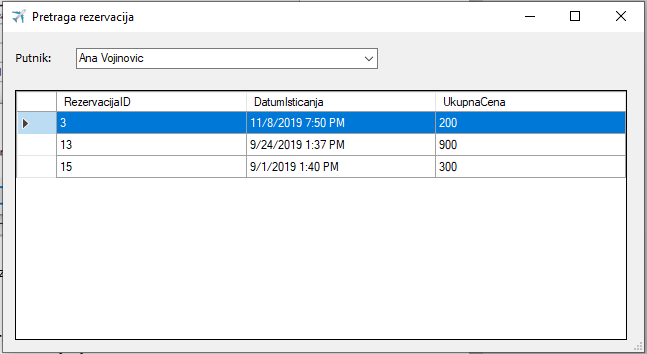
2. **Zaposleni** poziva **sistem** da nađe *rezervacije* po zadatoj vrednosti. (APSO)

3. **Sistem** pretražuje *rezervacije* po zadatoj vrednosti. (SO)

4. **Sistem** prikazuje **zaposlenom** *rezervacije* i poruku: “Sistem je našao rezervacije po zadatoj vrednosti ”. (IA)



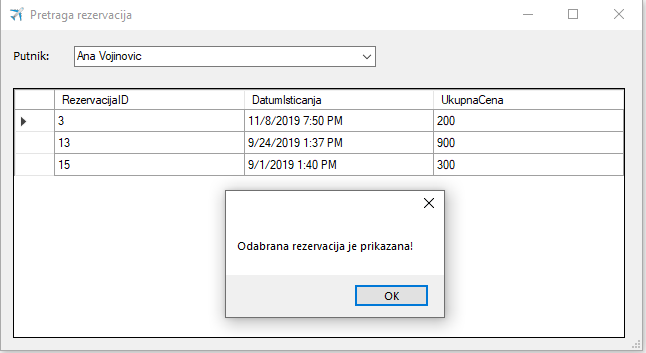
5. **Zaposleni** bira *rezervaciju* koju želi da izmeni. (APUSO)

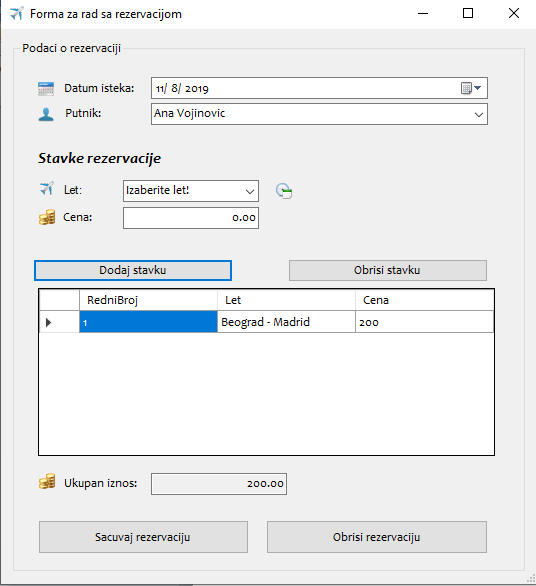


6. **Zaposleni** poziva **sistem** da učita odabranu *rezervaciju*. (APSO)

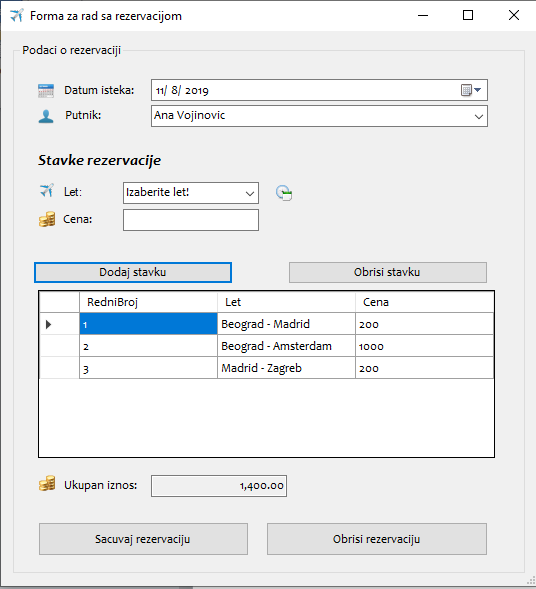
7. **Sistem** učitava podatke o odabranoj *rezervaciji*. (SO)

8. **Sistem** prikazuje **zaposlenom** podatke o *rezervaciji* i poruku: “Odabrana *rezervacija* je prikazana”. (IA)





9. **Zaposleni** unosi (menja) podatke o *rezervaciji*. (APUSO)

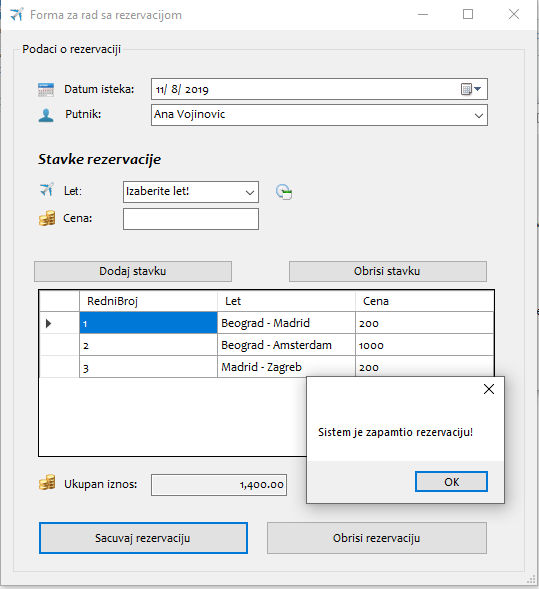


10. **Zaposleni** kontroliše da li je korektno uneo podatke o *rezervaciji*. (ANSO)

11. **Zaposleni** poziva **sistem** da zapamti podatke o *rezervaciji*. (APSO)

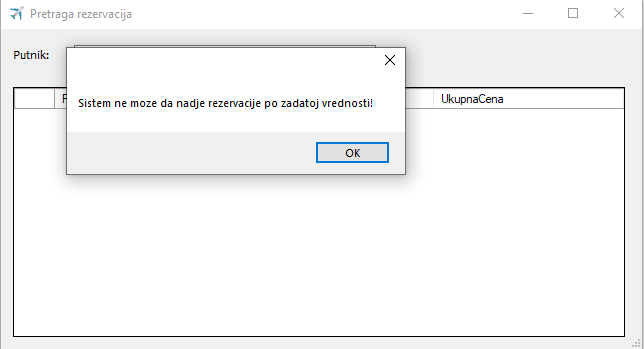
12. **Sistem** pamti podatke o *rezervaciji*. (SO)

13. **Sistem** prikazuje **zaposlenom** zapamćenu *rezervaciju* i poruku: “**Sistem** je zapamtio *rezervaciju*.” (IA)

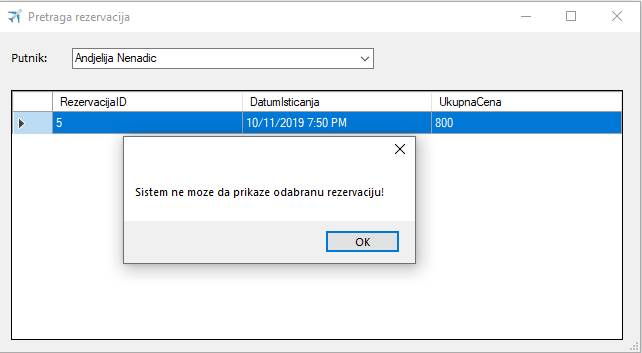


**Alternаtivnа scenаrijа**

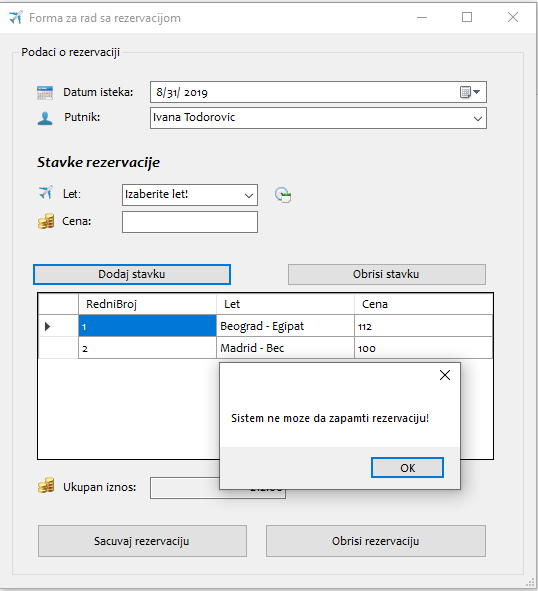
4.1. Ukoliko **sistem** ne može da nađe *rezervacije* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da nađe *rezervacije* po zadatoj vrednosti”. Prekida se izvršenje scenarija. (IA)



8.1. Ukoliko **sistem** ne može da učita *rezervaciju* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da prikaže odabranu *rezervaciju*”. Prekida se izvršenje scenarija. (IA)



13.1. Ukoliko **sistem** ne može da zapamti podatke o *rezervaciji* on prikazuje **zaposlenom** poruku “**Sistem** ne može da zapamti *rezervaciju*”. (IA)



### SK 8: Slučaj korišćenja – Pretraživanje rezervacija

**Naziv SK**

Pretraživanje *rezervacija*

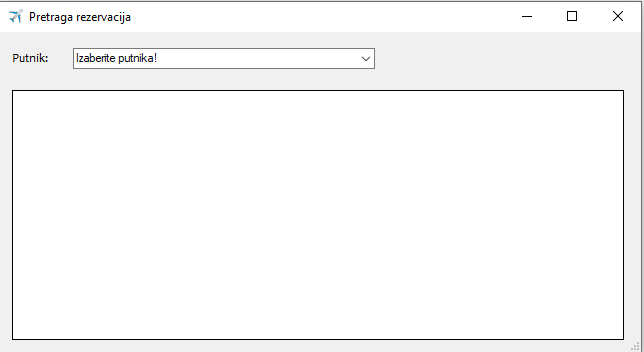
**Aktori SK**

**Zaposleni**

**Učesnici SK**

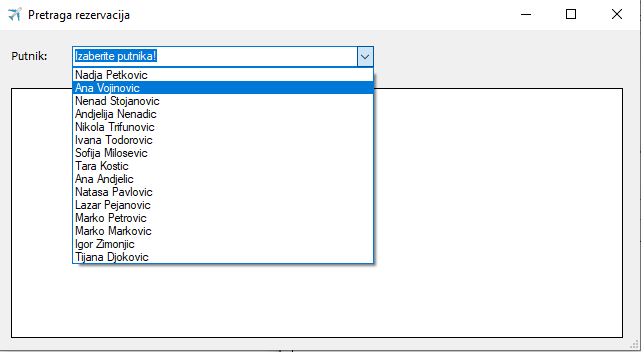
**Zaposleni** i **sistem** (program)

**Preduslov**: **Sistem** je uključen i **zaposleni** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa *rezervacijama*. Učitana je *lista putnika.*



**Osnovni scenario SK**

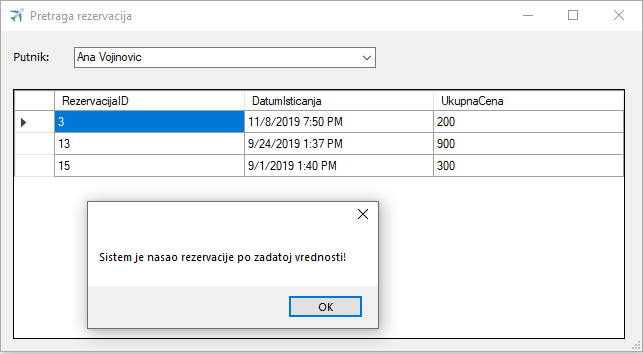
1. **Zaposleni** unosi vrednost(tj. kriterijum) po kojoj pretražuje *rezervacije*. (APUSO)



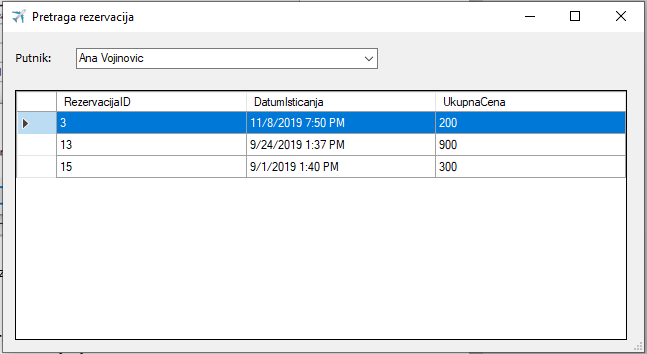
2. **Zaposleni** poziva **sistem** da nađe *rezervacije* po zadatoj vrednosti. (APSO)

3. **Sistem** pretražuje *rezervacije* po zadatoj vrednosti. (SO)

4. **Sistem** prikazuje **zaposlenom** *rezervacije* i poruku: “Sistem je našao rezervacije po zadatoj vrednosti ”. (IA)



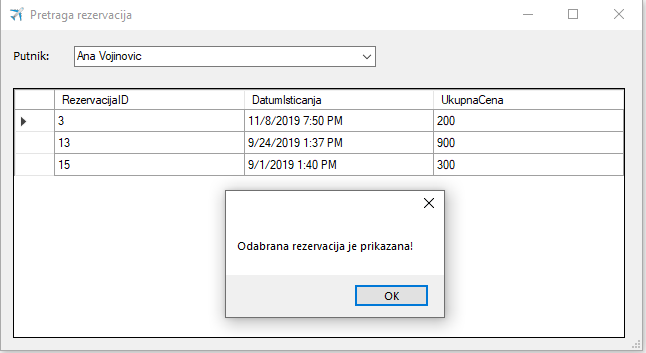
5. **Zaposleni** bira *rezervaciju* čije podatke želi da vidi. (APUSO)



6. **Zaposleni** poziva **sistem** da učita podatke o odabranoj *rezervaciji*. (APSO)

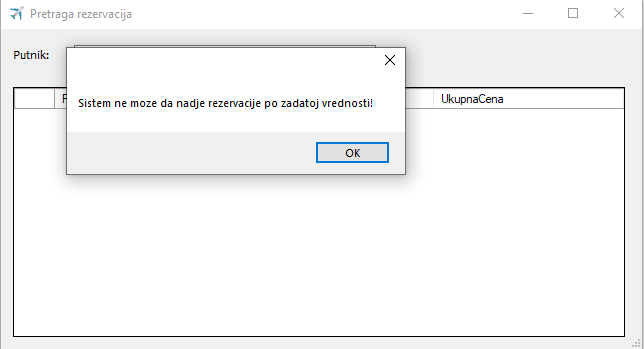
7. **Sistem** učitava podatke o odabranoj *rezervaciji*. (SO)

8. **Sistem** prikazuje **zaposlenom** podatke o odabranoj *rezervaciji* i poruku: “Odabrana *rezervacija* je prikazana.” (IA)

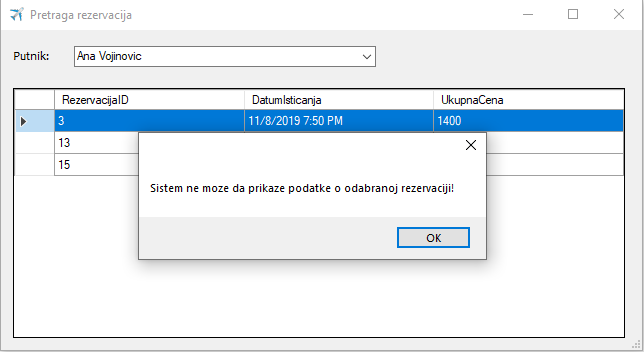


**Alternativna scenarija**

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



8.1. Ukoliko **sistem** ne može da učita podatke o odabranoj *rezervaciji* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da prikaže podatke o odabranoj *rezervaciji*.”(IA)



### SK 9: Slučаj korišćenjа – Brisanje rezervacije

**Nаziv** **SK**

Brisanje rezervacije

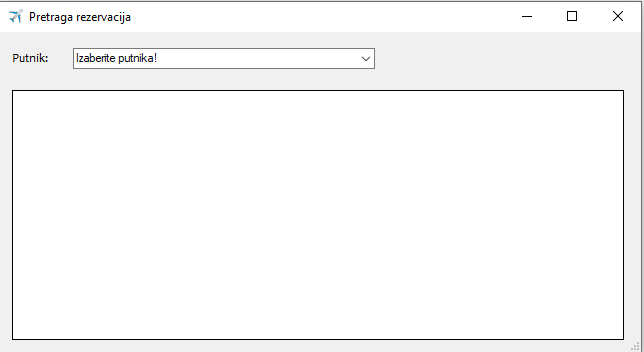
**Aktori SK**

**Zaposleni**

**Učesnici SK**

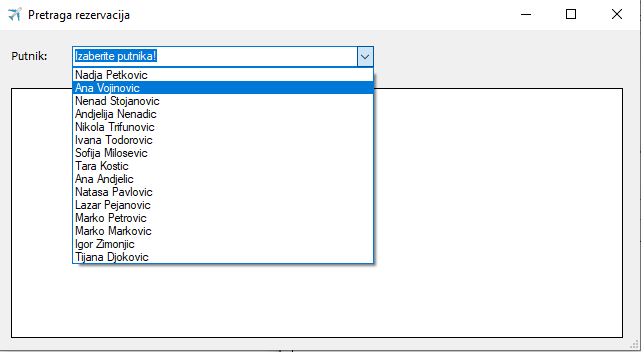
**Zaposleni** i **sistem** (program)

**Preduslov**: **Sistem** je uključen i **zaposleni** je ulogovan pod svojom šifrom. **Sistem** prikazuje formu za rad sa *rezervacijom*. Učitana je *lista putnika* i *lista letova*.



**Osnovni scenario SK**

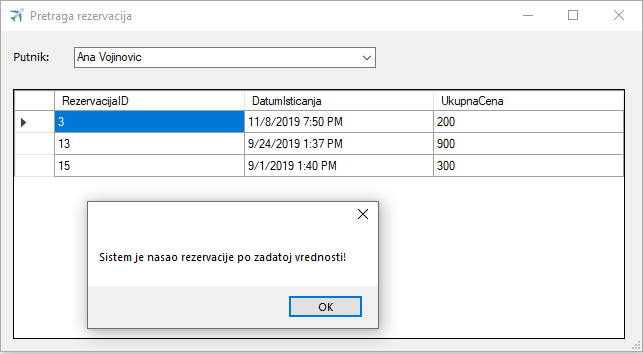
1. **Zaposleni** unosi vrednost po kojoj pretražuje *rezervacije*. (APUSO)



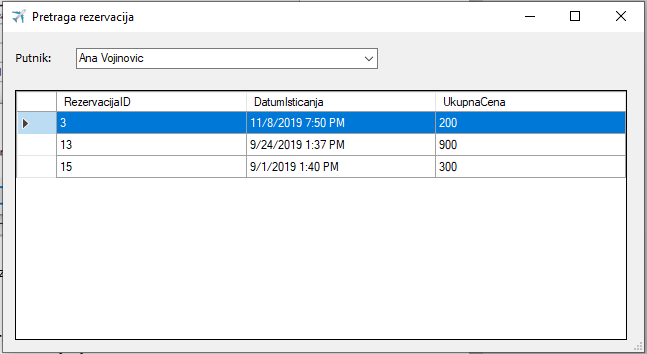
2. **Zaposleni** poziva **sistem** da nađe *rezervacije* po zadatoj vrednosti. (APSO)

3. **Sistem** pretražuje *rezervacije* po zadatoj vrednosti. (SO)

4. **Sistem** prikazuje **zaposlenom** *rezervacije* i poruku: “ Sistem je našao rezervacije po zadatoj vrednosti ”. (IA)



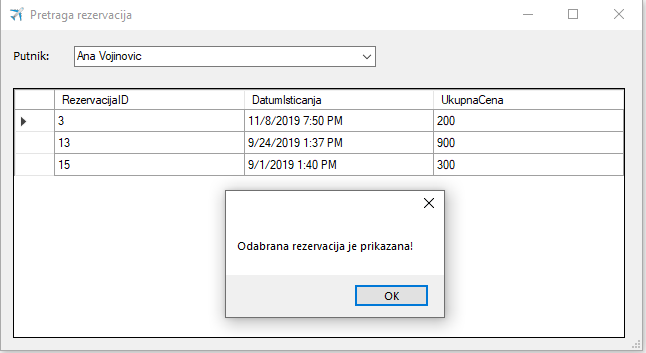
5. **Zaposleni** bira *rezervaciju* koju želi da obriše. (APUSO)



6. **Zaposleni** poziva **sistem** da učita podatke o odabranoj *rezervaciji*. (APSO)

7. **Sistem** učitava podatke o odabranoj *rezervaciji*. (SO)

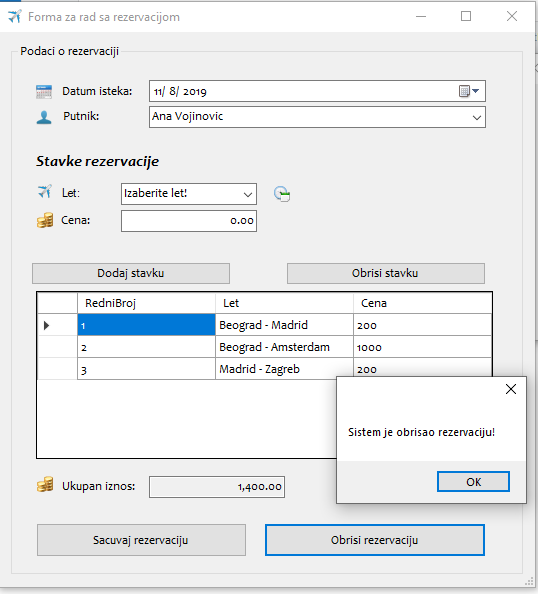
8. **Sistem** prikazuje **zaposlenom** podatke o *rezervaciji* i poruku: “Odabrana *rezervacija* je prikazana.” (IA)



9. **Zaposleni** poziva **sistem** da obriše *rezervaciju*. (APSO)

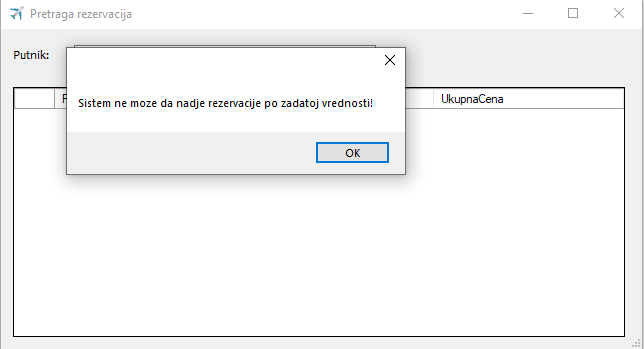
10. **Sistem** briše *rezervaciju*. (SO)

11. **Sistem** prikazuje **zaposlenom** poruku: “**Sistem** je obrisao *rezervaciju*.” (IA)

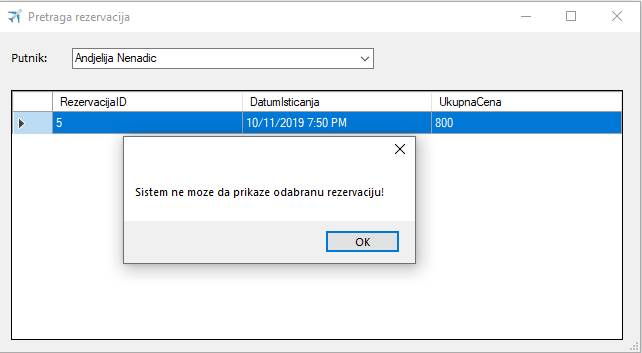


**Alternativna scenarija**

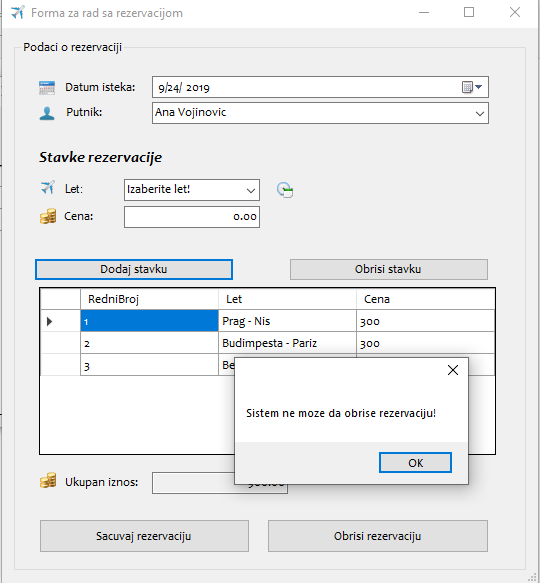
4.1. Ukoliko **sistem** ne može da nađe *rezervacije* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da nađe *rezervacije* po zadatoj vrednosti”. Prekida se izvršenje scenarija. (IA)



8.1. Ukoliko **sistem** ne može da učita odabranu *rezervaciju* on prikazuje **zaposlenom** poruku: “**Sistem** ne može da prikaže odabranu *rezervaciju*“. Prekida se izvršenje scenarija. (IA)



11.1. Ukoliko **sistem** ne može da obriše *rezervaciju* on prikazuje **zaposlenom** poruku “**Sistem** ne može da obriše *rezervaciju*”. (IA)



### 3.2.2. 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 7 - 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 System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Net;

using System.Runtime.Serialization.Formatters.Binary;

using System.Net.Sockets;

using Biblioteka;

namespace Komunikacija

{

public class Komunikacija

{

TcpClient klijent;

BinaryFormatter formater;

NetworkStream tok;

public bool poveziSeNaServer()

{

try

{

klijent = new TcpClient("127.0.0.1", 20000);

tok = klijent.GetStream();

formater = new BinaryFormatter();

return true;

}

catch (Exception)

{

return false;

}

}

public void kraj()

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.Kraj;

formater.Serialize(tok, transfer);

}

public Object NadjiZaposlenog(Zaposleni z)

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.NadjiZaposlenog;

transfer.TransferObjekat = z;

formater.Serialize(tok, transfer);

return (formater.Deserialize(tok) as TransferKlasa).Rezultat;

}

public Object ZapamtiPutnika(Putnik p)

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.ZapamtiPutnika;

transfer.TransferObjekat = p;

formater.Serialize(tok, transfer);

return (formater.Deserialize(tok) as TransferKlasa).Rezultat;

}

public Object KreirajPutnika(Putnik p)

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.KreirajPutnika;

transfer.TransferObjekat = p;

formater.Serialize(tok, transfer);

return (formater.Deserialize(tok) as TransferKlasa).Rezultat;

}

public Object PretraziPutnike(Putnik p)

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.PretraziPutnike;

transfer.TransferObjekat = p;

formater.Serialize(tok, transfer);

return (formater.Deserialize(tok) as TransferKlasa).Rezultat;

}

public Object UcitajPutnika(Putnik p)

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.UcitajPutnika;

transfer.TransferObjekat = p;

formater.Serialize(tok, transfer);

return (formater.Deserialize(tok) as TransferKlasa).Rezultat;

}

public Object ObrisiPutnika(Putnik p)

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.ObrisiPutnika;

transfer.TransferObjekat = p;

formater.Serialize(tok, transfer);

return (formater.Deserialize(tok) as TransferKlasa).Rezultat;

}

public Object UcitajListuPutnika()

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.UcitajListuPutnika;

transfer.TransferObjekat = new Putnik();

formater.Serialize(tok, transfer);

return (formater.Deserialize(tok) as TransferKlasa).Rezultat;

}

public Object UcitajListuLetova()

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.UcitajListuLetova;

transfer.TransferObjekat = new Let();

formater.Serialize(tok, transfer);

return (formater.Deserialize(tok) as TransferKlasa).Rezultat;

}

public Object KreirajRezervaciju()

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.KreirajRezervaciju;

transfer.TransferObjekat = new Rezervacija();

formater.Serialize(tok, transfer);

return (formater.Deserialize(tok) as TransferKlasa).Rezultat;

}

public Object ZapamtiRezervaciju(Rezervacija p)

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.ZapamtiRezervaciju;

transfer.TransferObjekat = p;

formater.Serialize(tok, transfer);

return (formater.Deserialize(tok) as TransferKlasa).Rezultat;

}

public Object PretraziRezervacije(Rezervacija p)

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.PretraziRezervacije;

transfer.TransferObjekat = p;

formater.Serialize(tok, transfer);

return (formater.Deserialize(tok) as TransferKlasa).Rezultat;

}

public Object UcitajRezervaciju(Rezervacija p)

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.UcitajRezervaciju;

transfer.TransferObjekat = p;

formater.Serialize(tok, transfer);

return (formater.Deserialize(tok) as TransferKlasa).Rezultat;

}

public Object ObrisiRezervaciju(Rezervacija p)

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.ObrisiRezervaciju;

transfer.TransferObjekat = p;

formater.Serialize(tok, transfer);

return (formater.Deserialize(tok) as TransferKlasa).Rezultat;

}

}

}

### 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);

}

}

### Ugovor UG 1:PronađiZaposlenog

Operacija: **PronađiZaposlenog** (*Zaposleni*):signal;

Veza sa SK: SK1

Preduslovi: -

Postuslovi: -



### Ugovor UG 2:KreirajPutnika

Operacija: **KreirajPutnika** (Putnik):signal;

Veza sa SK: SK2

Preduslovi: *Strukturna ograničenja nad objektom Putnik moraju biti zadovoljena.*

Postuslovi: *Putnik je sacuvan.*



### Ugovor UG 3:ZapamtiPutnika

Operacija: **ZapamtiPutnika** (Putnik):signal;

Veza sa SK: SK2,SK3

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

Postuslovi: *Podaci o putniku su zapamćeni.*



### Ugovor UG 4:PretraživanjePutnika

Operacija: **PretraživanjePutnika** (KriterijumPretrage, List<Putnik>):signal;

Veza sa SK: SK3, SK4, SK5

Preduslovi: *-*

Postuslovi: *-*



### Ugovor UG 5:UčitajPutnika

Operacija: **UčitajPutnika** (Putnik):signal;

Veza sa SK: SK3, SK4, SK5

Preduslovi: *-*

Postuslovi: *-*



### Ugovor UG 6:ObrišiPutnika

Operacija: **ObrišiPutnika** (Putnik):signal;

Veza sa SK: SK5

Preduslovi: *Strukturna ograničenja nad objektom Putnik moraju biti zadovoljena.*

Postuslovi: *Putnik je obrisan.*



### Ugovor UG 7:UčitajSvePutnike

Operacija: **PretražiSvePutnike** (*List<Putnik>*)

Veza sa SK: SK6

Preduslovi: -

Postuslovi: -



### Ugovor UG 8:UčitajSveLetove

Operacija: **PretražiSveLetove** (List*<Let>*):signal;

Veza sa SK: SK6,SK7

Preduslovi: *-*

Postuslovi: *-*



### Ugovor UG 9:KreirajRezervaciju

Operacija: **KreirajRezervaciju** (*Rezervacija*)

Veza sa SK: SK6

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

Postuslovi: Kreirana je nova rezervacija.



### Ugovor UG 10:ZapamtiRezervaciju

Operacija: **ZapamtiRezervaciju** (Rezervacija):signal;

Veza sa SK: SK6,SK7

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

Postuslovi: *Podaci o rezervaciji su zapamćeni.*



### Ugovor UG 11:PretraživanjeRezervacija

Operacija: **PretraživanjeRezervacija** (KriterijumPretrage, List<Rezervacija>)::signal;

Veza sa SK: SK7,SK8,SK9

Preduslovi: *-*

Postuslovi: *-*



### Ugovor UG 12:UčitajRezervaciju

Operacija: **UčitajRezervaciju** (Rezervacija):signal;

Veza sa SK: SK7,SK8,SK9

Preduslovi: *-*

Postuslovi: *-*



### Ugovor UG 13:ObrišiRezervaciju

Operacija: **ObrišiRezervaciju** (Rezervacija):signal;

Veza sa SK: SK9

Preduslovi: *Strukturna ograničenja nad objektom Rezervacija moraju biti zadovoljena.*

Postuslovi: *Rezervacija je obrisana.*





Slika 8 - Klase koje su odgovorne za SO nasleđuju klasu OpstaSO

### 3.3.3. Domenske klase

Na osnovi 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 *OpstiDomenskiObjekat* 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.Linq;

using System.Text;

using System.Data;

namespace Biblioteka

{

public interface OpstiDomenskiObjekat

{

string tabela

{

get;

}

string kljuc

{

get;

}

string uslovJedan

{

get;

}

string uslovVise

{

get;

}

string azuriranje

{

get;

}

string upisivanje

{

get;

}

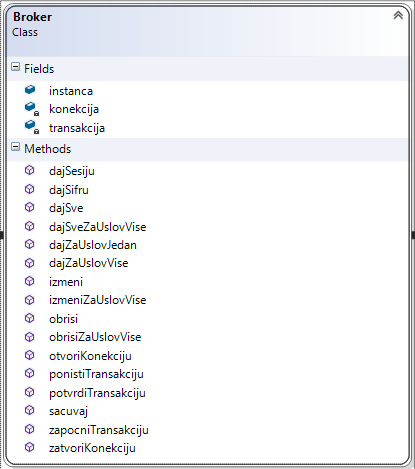
OpstiDomenskiObjekat napuni(DataRow red);

}

}

### 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:





Slika 9 - Database broker klasa se povezuje sa klasom OpstiDomenskiObjekat

### 3.3.5. Projektovanje skladišta podataka

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



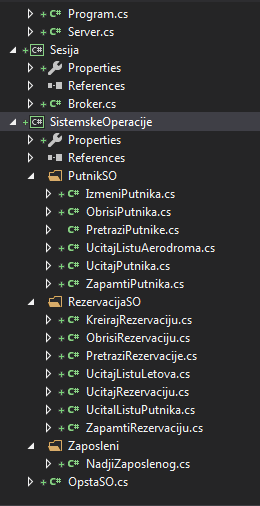
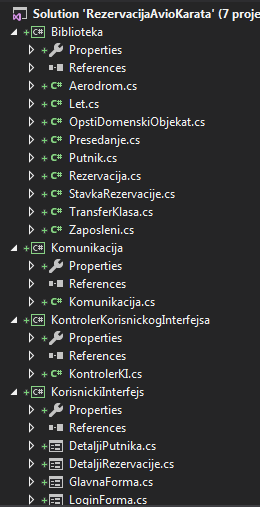
Slika 10 - ASSTBP Arhitektura soft. Sistema nakon projektovanja tabela baze podataka

# 4. Implementacija

Softverski sistem, rezultat ovog rada, razvijen je u programskom jeziku C#. Sistem je projektovan kao klijent-server. Kao razvojno okruženje korišćen je Visual Studio 2017. Kao sistem za upravljanjem bazom podataka korišćen je MSSQL.

Na osnovi 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 osnovi izvršenih testiranja otklonjeni su uočeni nedostaci.

# 6. Literatura

Projektovanje softvera – Skripta, dr Siniša Vlajić, Beograd - 2018.