Univerzitet u Beogradu

Fakultet organizacionih nauka

Laboratorija za softversko inženjerstvo

Seminarski rad iz predmeta

Projektovanje softvera

Tema: Softverski sistem za planiranje fitnes aktivnosti

|  |  |
| --- | --- |
| Profesor:  dr Siniša Vlajić | Student: |

Beograd, 2022.

Sadržaj

[1. Korisnički Zahtevi 4](#_Toc105701432)

[1.1 Verbalni Opis 4](#_Toc105701433)

[1.2 UML Slučajeva korišćenja 4](#_Toc105701434)

[1.3 Legenda akcija 5](#_Toc105701435)

[1.3.1 SK1: Unos novog korisnika 5](#_Toc105701436)

[1.3.2 SK2: Prijava korisnika 6](#_Toc105701437)

[1.3.3 SK3: Pretraživanje korisnika 7](#_Toc105701438)

[1.3.4 SK4: Brisanje korisnika 8](#_Toc105701439)

[1.3.5 SK5: Izmena korisnika 9](#_Toc105701440)

[1.3.6 SK6: Pretraživanje vežbi 10](#_Toc105701441)

[1.3.7 SK7: Unos nove vežbe 11](#_Toc105701442)

[1.3.8 SK8: Izmena vežbe 12](#_Toc105701443)

[1.3.9 SK9: Brisanje vežbe 13](#_Toc105701444)

[1.3.10 SK10: Unos novog plana 14](#_Toc105701445)

[1.3.11 SK11: Izmena plana 15](#_Toc105701446)

[1.3.12 SK12: Brisanje plana 16](#_Toc105701447)

[2. Analiza 17](#_Toc105701448)

[2.1 Ponašanje softverskog sistema – Sistemski dijagram sekvenci 17](#_Toc105701449)

[2.1.1 DS 1: Unos novog korisnika 17](#_Toc105701450)

[2.1.2 DS 2: Prijava korisnika 18](#_Toc105701451)

[2.1.3 DS 3: Pretraživanje korisnika 19](#_Toc105701452)

[2.1.4 DS 4: Brisanje korisnika 21](#_Toc105701453)

[2.1.5 DS 5: Izmena korisnika 23](#_Toc105701454)

[2.1.6 DS 6: Pretraživanje vežbi 25](#_Toc105701455)

[2.1.7 DS 7: Unos nove vežbe 27](#_Toc105701456)

[2.1.8 DS 8: Izmena vežbe 28](#_Toc105701457)

[2.1.9 DS 9: Brisanje vežbe 30](#_Toc105701458)

[2.1.10 DS 10: Unos novog plana 32](#_Toc105701459)

[2.1.11 DS 11: Izmena plana 33](#_Toc105701460)

[2.1.12 DS 12: Brisanje plana 35](#_Toc105701461)

[2.2. Ponašanje softverskog sistema – Definisanje ugovora o sistemskim operacijama 37](#_Toc105701462)

[2.3. Struktura softverskog sistema – Konceptualni (domenski) model 40](#_Toc105701463)

[2.4. Struktura softverskog sistema – Relacioni model 41](#_Toc105701464)

[2.5. Struktura softverskog sistema – Tabela Ograničenja 42](#_Toc105701465)

[3.Projektovanje 45](#_Toc105701466)

[3.1 Arhitektura softverskog sistema 45](#_Toc105701467)

[3.2 Projektovanje korisničkog interfejsa 46](#_Toc105701468)

[3.2.1 Projektovanje ekranskih formi 46](#_Toc105701469)

[3.2.1.1 SK1: Unos novog korisnika 48](#_Toc105701470)

[3.2.1.2 SK2: Prijava korisnika 50](#_Toc105701471)

[3.2.1.3 SK3: Pretraživanje korisnika 52](#_Toc105701472)

[3.2.1.4 SK4: Brisanje korisnika 55](#_Toc105701473)

[3.2.1.5 SK5: Izmena korisnika (složen slučaj korišćenja) 58](#_Toc105701474)

[3.2.1.6 SK6: Pretraživanje vežbi 62](#_Toc105701475)

[3.2.1.7 SK7: Unos nove vežbe 65](#_Toc105701476)

[3.2.1.8 SK8: Izmena vežbe 67](#_Toc105701477)

[3.2.1.9 SK9: Brisanje vežbe 71](#_Toc105701478)

[3.2.1.10 SK10: Unos novog plana 73](#_Toc105701479)

[3.2.1.11 SK11: Izmena plana (složen slučaj korišćenja) 75](#_Toc105701480)

[3.2.1.12 SK12: Brisanje plana 78](#_Toc105701481)

[3.3 Projektovanje aplikacione logike 80](#_Toc105701482)

[3.3.1 Komunikacija sa klijentima 80](#_Toc105701483)

[3.3.2 Kontroler Aplikacione Logike 82](#_Toc105701484)

[3.3.3 Projektovanje strukture sofverskog sistema 85](#_Toc105701485)

[3.3.4 Projektvovanje strukture sofverskog sistema 89](#_Toc105701486)

[3.3.5 Broker Baze Podataka 96](#_Toc105701487)

[3.4 Projektovanje skladišta podataka 97](#_Toc105701488)

[4. Implementacija 99](#_Toc105701489)

[5. Testiranje 100](#_Toc105701490)

[6. Literatura 100](#_Toc105701491)

# 1. Korisnički Zahtevi

## 1.1 Verbalni Opis

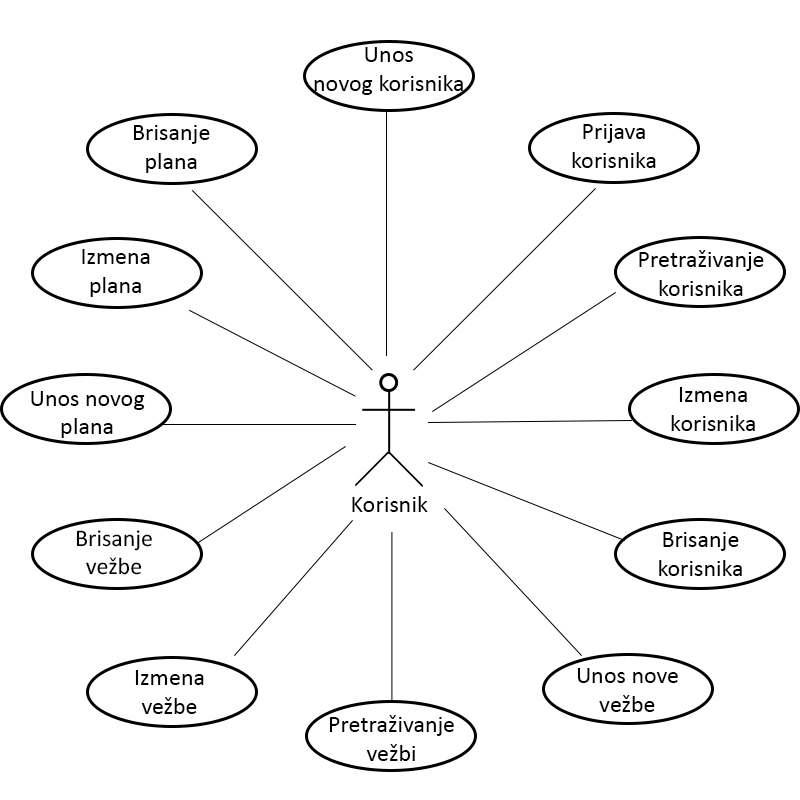
Potrebno je napraviti aplikaciju koja omogućava prijavljivanje korisnika, vođenje evidencije o njegovom planu vežbanja i promeni težine korisnika.

Aplikacija treba da vodi evidenciju o korisnicima, da omogući pretragu korisnika prema određenim vrednostima, kao i unos novih korisnika i izmenu podataka o postojećim i brisanje istih.

Takođe, potrebno je voditi evidenciju o vežbama koje postoje u bazi podataka i omogućiti pretragu prema zadatim vrednostima.

Za sve akcije koje menjaju bazu podataka potrebno je biti prijavljen na sistem.

## 1.2 UML Slučajeva korišćenja



Slika 1 - Dijagram slučajeva korišćenja

## 1.3 Legenda akcija

Aktor izvodi jednu od tri vrste akcija:

a) Aktor Priprema Ulaz (Ulazne Argumente) za Sistemsku Operaciju (**APUSO**).

b) Aktor Poziva sistem da izvrši Sistemsku Operaciju (**APSO**).

c) Aktor izvršava NeSistemsku Operaciju (**ANSO**).

Sistem izvodi dve akcije u kontinuitetu:

a) Sistem izvršava Sistemsku Operaciju(**SO**):

b) Rezultat sistemske operacije (Izlazni argumenti (**IA**)) se prosleđuje do aktora.

### 1.3.1 SK1: Unos novog korisnika

**Nаziv** **SK**

Unos korisnika

**Aktori SK**

*Korisnik*

**Učesnici SK**

*Korisnik i sistem (progrаm)*

**Preduslov**: Sistem je uključen i korisnikje ulogovan sa svojom šifrom. Sistem prikazuje formu za rad sa korisnicima.

**Osnovni scenаrio SK**

1. **Korisnik** unosi podаtke o novom *korisniku*. (APUSO)
2. **Korisnik** kontroliše dа li je korektno uneo podаtke o novom *korisniku*. (ANSO)
3. **Korisnik** pozivа *sistem* dа zаpаmti podаtke o *korisniku*. (APSO)
4. **Sistem** pаmti podаtke o *korisniku*. (SO)
5. **Sistem** prikаzuje *korisniku* zаpаmćenog *korisnika* i poruku: ʺUspesno sacuvan korisnik ʺ. (IA)

Alternativni scenariji

5.1 Ukoliko sistem ne može dа zаpаmti podаtke o novom *korisniku* on prikаzuje *korisniku* poruku “Greska pri cuvanju korisnika ”. (IA)

### 1.3.2 SK2: Prijava korisnika

**Nаziv** **SK**

Prijava korisnika

**Aktori SK**

*Korisnik*

**Učesnici SK**

*Korisnik* *i sistem (progrаm)*

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

**Osnovni scenаrio SK**

1. **Korisnik** unosi podаtke za autentifikaciju. (APUSO)
2. **Korisnik** pozivа *sistem* dа pronađe korisnika sa zadatim podacima. (APSO)
3. **Sistem** pretražuje korisnike. (SO)
4. **Sistem** prikаzuje *korisniku* poruku: ʺUspešno ste se prijavili ʺ. (IA)

Alternativni scenariji

4.1. Ukoliko sistem ne može da nađe korisnika prikazuje poruku “Greška prilikom prijavljivanja na sistem!”. (IA)

### 1.3.3 SK3: Pretraživanje korisnika

**Nаziv SK**

Pretrаživаnje korisnina

**Aktori SK**

*Korisnik*

**Učesnici SK**

*Korisnik i sistem (progrаm*)

**Preduslov**: Sistem je uključen. Sistem prikazuje formu za rad sa korisnicima.

**Osnovni scenаrio SK**

1. **Korisnik** unosi vrednost po kojoj pretrаžuje *korisnike*. (APUSO)
2. **Korisnik** pozivа *sistem* dа nаđe *korisnike* po zаdаtoj vrednosti. (APSO)
3. **Sistem** trаži *korisnike* po zаdаtoj vrednosti. (SO)
4. **Sistem** prikazuje *korisniku* nađene *korisnike*. (IA)
5. **Korisnik** bira*korisnika* čije podatke želi da vidi. (APUSO)
6. **Korisnik** poziva sistem da prikaže podatke o izabranom *korisniku*. (APSO)
7. **Sistem** traži podatke o izabranom *korisniku*. (SO)
8. **Sistem** prikаzuje *korisniku* podаtke o *korisniku*. (IA)

Alternativni scenariji

4.1 Ukoliko sistem ne može dа nаđe *korisnike* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe *korisnike* po zаdаtoj vrednosti”. Prekida se izvršenje scenaria. (IA)

8.1. Ukoliko sistem ne može da nađe izabranog *korisnika* on prikazuje korisniku poruku: “Sistem ne može da nađe podatke o izabranom korisniku ”. (IA)

### 1.3.4 SK4: Brisanje korisnika

**Nаziv** **SK**

Brisanje korisnika

**Aktori SK**

*Korisnik*

**Učesnici SK**

*Korisnik i sistem (progrаm*)

**Preduslov**: Sistem je uključen i korisnikje ulogovan sa svojom šifrom. Sistem prikazuje formu za rad sa korisnicima.

**Osnovni scenаrio SK**

1. **Korisnik** unosi vrednost po kojoj se pretrаžuje *korisnike*. (APUSO)
2. **Korisnik** pozivа *sistem* dа nаđe *korisnike* po zаdаtoj vrednosti. (APSO)
3. **Sistem** trаži *korisnike* po zаdаtoj vrednosti. (SO)
4. **Sistem** prikаzuje *korisniku* pronađene *korisnike*. (IA)
5. **Korisnik** bira *korisnika* kojeg želi da obriše. (APUSO)
6. **Korisnik** poziva *sistem* da obriše izabranog *korisnika*. (APSO)
7. **Sistem** briše izabranog *korisnika*. (SO)
8. **Sistem** prikazuje *korisniku* poruku: “Korisnik obrisan ”. (IA)

Alternativni scenariji

4.1 Ukoliko sistem ne može dа nаđe *korisnika* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe *korisnike* po zаdаtoj vrednosti”. Prekida se izvršenje scenaria. (IA)

8.1. Ukoliko sistem ne može da obriše *korisnika* on prikazuje korisniku poruku: “Sistem ne može da obriše korisnika ”. (IA)

### 1.3.5 SK5: Izmena korisnika

**Nаziv SK**

Izmenа korisnika

**Aktori SK**

*Korisnik*

**Učesnici SK**

*Korisnik i sistem (progrаm*)

**Preduslov**: Sistem je uključen i korisnikje ulogovan sa svojom šifrom. Sistem prikazuje formu za rad sa korisnicima.

**Osnovni scenаrio SK**

1. **Korisnik** unosi vrednost po kojoj se pretrаžuju *korisnici*. (APUSO)
2. **Korisnik** pozivа *sistem* dа nаđe *korisnike* po zаdаtoj vrednosti. (APSO)
3. **Sistem** trаži *korisnike* po zаdаtoj vrednosti. (SO)
4. **Sistem** prikаzuje *korisniku* pronađene *korisnike.* (IA)
5. **Korisnik** bira *korisnika* kojeg želi da izmeni. (APUSO)
6. **Korisnik** pozivа *sistem* da učita podatke o izabranom *korisniku.* (APSO)
7. **Sistem** trаži podatke o izabranom *korisniku.* (SO)
8. **Sistem** prikаzuje *korisniku* podatke o izabranom *korisniku.* (IA)
9. **Korisnik** unosi (menja) *korisnika*. (APUSO)
10. **Korisnik** kontroliše dа li je korektno uneo podаtke o *korisniku*. (ANSO)
11. **Korisnik** pozivа *sistem* dа zаpаmti podatke o *korisniku*.(APSO)
12. **Sistem** pаmti podatke o *korisniku*. (SO)
13. **Sistem** prikаzuje *korisniku* poruku: “Korisnik uspešno ažuriran“. (IA)

Alternativni scenariji

4.1 Ukoliko sistem ne može dа nаđe *korisnike* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe *korisnike* po zаdаtoj vrednosti”. Prekida se izvršenje scenaria. (IA)

8.1 Ukoliko sistem ne može dа nađe *korisnika* on prikаzuje korisniku poruku: “Sistem ne može da nađe podatke o izabranom korisniku”. Prekidа se izvršenje scenаriа. (IA)

13.1 Ukoliko sistem ne može dа zаpаmti izmenjene podаtke o *korisniku* on prikаzuje korisniku poruku “Sistem ne može dа zаpаmti korisnika”. (IA)

### 1.3.6 SK6: Pretraživanje vežbi

**Nаziv SK**

Pretrаživаnje vežbi

**Aktori SK**

*Korisnik*

**Učesnici SK**

*Korisnik i sistem (progrаm*)

**Preduslov**: Sistem je uključen. Sistem prikazuje formu za rad sa vežbama ili korisnički panel.

**Osnovni scenаrio SK**

1. **Korisnik** unosi vrednost po kojoj pretrаžuje *vežbe*. (APUSO)
2. **Korisnik** pozivа *sistem* dа nаđe *vežbe* po zаdаtoj vrednosti. (APSO)
3. **Sistem** trаži *vežbe* po zаdаtoj vrednosti. (SO)
4. **Sistem** prikazuje *korisniku* nađene *vežbe*. (IA)
5. **Korisnik** bira*vežbe* čije podatke želi da vidi. (APUSO)
6. **Korisnik** poziva sistem da prikaže podatke o izabranim *vežbama*. (APSO)
7. **Sistem** traži podatke o izabranoj *vežbi*. (SO)
8. **Sistem** prikаzuje *korisniku* podаtke o *vežbi*. (IA)

Alternativni scenariji

4.1 Ukoliko sistem ne može dа nаđe *vežbe* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe *vežbu* po zаdаtoj vrednosti”. Prekida se izvršenje scenaria. (IA)

8.1. Ukoliko sistem ne može da nađe izabranog *vežbe* on prikazuje korisniku poruku: “Sistem ne može da nađe podatke o izabranoj vežbi”. (IA)

### 1.3.7 SK7: Unos nove vežbe

**Nаziv** **SK**

Unos vežbe

**Aktori SK**

*Korisnik*

**Učesnici SK**

*Korisnik i sistem (progrаm)*

**Preduslov**: Sistem je uključen i korisnikje ulogovan sa svojom šifrom. Sistem prikazuje formu za rad sa vežbama.

**Osnovni scenаrio SK**

1. **Korisnik** unosi podаtke o novoj *vežbi*. (APUSO)
2. **Korisnik** kontroliše dа li je korektno uneo podаtke o novoj *vežbi*. (ANSO)
3. **Korisnik** pozivа *sistem* dа zаpаmti podаtke o *vežbi*. (APSO)
4. **Sistem** pаmti podаtke o *vežbi*. (SO)
5. **Sistem** prikаzuje *korisniku* zаpаmćenog *korisnika* i poruku: „Uspesno sacuvana vezbaʺ. (IA)

Alternativni scenariji

5.1 Ukoliko sistem ne može dа zаpаmti podаtke o novoj vežbi on prikаzuje *korisniku* poruku “Greska pri cuvanju vezbe”. (IA)

### 1.3.8 SK8: Izmena vežbe

**Nаziv SK**

Izmenа vežbe

**Aktori SK**

*Korisnik*

**Učesnici SK**

*Korisnik i sistem (progrаm*)

**Preduslov**: Sistem je uključen i korisnikje ulogovan sa svojom šifrom. Sistem prikazuje formu za rad sa vežbama.

**Osnovni scenаrio SK**

1. **Korisnik** unosi vrednost po kojoj se pretrаžuju *vežbe*. (APUSO)
2. **Korisnik** pozivа *sistem* dа nаđe *vežbe* po zаdаtoj vrednosti. (APSO)
3. **Sistem** trаži *vežbe* po zаdаtoj vrednosti. (SO)
4. **Sistem** prikаzuje *korisniku* pronađene *vežbe*. (IA)
5. **Korisnik** bira *vežbu* koju želi da izmeni. (APUSO)
6. **Korisnik** pozivа *sistem* da učita podatke o izabranoj *vežbi.* (APSO)
7. **Sistem** trаži podatke o izabranoj *vežbi.* (SO)
8. **Sistem** prikаzuje *korisniku* podatke o izabranoj *vežbi.* (IA)
9. **Korisnik** unosi (menja) *vežbu*. (APUSO)
10. **Korisnik** kontroliše dа li je korektno uneo podаtke o *vežbi*. (ANSO)
11. **Korisnik** pozivа *sistem* dа zаpаmti podatke o *vežbi*.(APSO)
12. **Sistem** pаmti podatke o *vežbi*. (SO)
13. **Sistem** prikаzuje *korisniku* poruku: „Uspesno sacuvana vezba“. (IA)

Alternativni scenariji

4.1 Ukoliko sistem ne može dа nаđe *vežbe* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe *vežbu* po zаdаtoj vrednosti”. Prekida se izvršenje scenaria. (IA)

8.1. Ukoliko sistem ne može da nađe izabranog *vežbe* on prikazuje korisniku poruku: “Sistem ne može da nađe podatke o izabranoj vežbi”. (IA)

13.1 Ukoliko sistem ne može dа zаpаmti izmenjene podаtke o *vežbi* on prikаzuje korisniku poruku “ Greska pri cuvanju vezbe”. (IA)

### 1.3.9 SK9: Brisanje vežbe

**Nаziv** **SK**

Brisanje vežbe

**Aktori SK**

*Korisnik*

**Učesnici SK**

*Korisnik i sistem (progrаm*)

**Preduslov**: Sistem je uključen i korisnikje ulogovan sa svojom šifrom. Sistem prikazuje formu za rad sa vežbama.

**Osnovni scenаrio SK**

1. **Korisnik** unosi vrednost po kojoj se pretrаžuju *vežbe*. (APUSO)
2. **Korisnik** pozivа *sistem* dа nаđe *vežbe* po zаdаtoj vrednosti. (APSO)
3. **Sistem** trаži *vežbe* po zаdаtoj vrednosti. (SO)
4. **Sistem** prikаzuje *korisniku* pronađene *vežbe*. (IA)
5. **Korisnik** bira *vežbu* koju želi da obriše. (APUSO)
6. **Korisnik** poziva *sistem* da obriše izabranu *vežbu*. (APSO)
7. **Sistem** briše izabranu *vežbu*. (SO)
8. **Sistem** prikazuje *korisniku* poruku: “Sistem je obrisao *vežbu* ”. (IA)

Alternativni scenariji

4.1 Ukoliko sistem ne može dа nаđe *korisnika* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe *vežbe*  po zаdаtoj vrednosti”. Prekida se izvršenje scenaria. (IA)

8.1. Ukoliko sistem ne može da obriše *korisnika* on prikazuje korisniku poruku: “Sistem ne može da obriše vežbu ”. (IA)

### 1.3.10 SK10: Unos novog plana

**Nаziv** **SK**

Unos korisnika

**Aktori SK**

*Korisnik*

**Učesnici SK**

*Korisnik i sistem (progrаm)*

**Preduslov**: Sistem je uključen i korisnikje ulogovan sa svojom šifrom. Sistem prikazuje korisnički panel.

**Osnovni scenаrio SK**

1. **Korisnik** unosi podаtke o novom *planu*. (APUSO)
2. **Korisnik** kontroliše dа li je korektno uneo podаtke o novom *planu*. (ANSO)
3. **Korisnik** pozivа *sistem* dа zаpаmti podаtke o *planu*. (APSO)
4. **Sistem** pаmti podаtke o *planu*. (SO)
5. **Sistem** prikаzuje *korisniku* zаpаmćeni *plan* i poruku: ʺPlan uspešno dodatʺ. (IA)

Alternativni scenariji

5.1 Ukoliko sistem ne može dа zаpаmti podаtke o novom *planu* on prikаzuje *korisniku* poruku “Greska pri cuvanju plana”. (IA)

### 1.3.11 SK11: Izmena plana

**Nаziv SK**

Izmenа plana

**Aktori SK**

*Korisnik*

**Učesnici SK**

*Korisnik i sistem (progrаm*)

**Preduslov**: Sistem je uključen i korisnikje ulogovan sa svojom šifrom. Sistem prikazuje korisnički panel.

**Osnovni scenаrio SK**

1. **Korisnik** unosi vrednost po kojoj se pretrаžuju *planovi*. (APUSO)
2. **Korisnik** pozivа *sistem* dа nаđe *planove* po zаdаtoj vrednosti. (APSO)
3. **Sistem** trаži *planove* po zаdаtoj vrednosti. (SO)
4. **Sistem** prikаzuje *korisniku* pronađene *planove*. (IA)
5. **Korisnik** bira *plan* koju želi da izmeni. (APUSO)
6. **Korisnik** pozivа *sistem* da učita podatke o izabranom *planu.* (APSO)
7. **Sistem** trаži podatke o izabranom *planu.* (SO)
8. **Sistem** prikаzuje *korisniku* podatke o izabranom *planu.* (IA)
9. **Korisnik** unosi (menja) *plan*. (APUSO)
10. **Korisnik** kontroliše dа li je korektno uneo podаtke o *planu*. (ANSO)
11. **Korisnik** pozivа *sistem* dа zаpаmti podatke o *planu*.(APSO)
12. **Sistem** pаmti podatke o *planu*. (SO)
13. **Sistem** prikаzuje *korisniku* poruku: „Plan uspešno sačuvan“. (IA)

Alternativni scenariji

4.1 Ukoliko sistem ne može dа nаđe *planove* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe izabran plan”. Prekida se izvršenje scenaria. (IA)

8.1 Ukoliko sistem ne može dа nađe *plan* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe izabran plan”. Prekidа se izvršenje scenаriа. (IA)

13.1 Ukoliko sistem ne može dа zаpаmti izmenjene podаtke o *planu* on prikаzuje korisniku poruku “Greska pri cuvanju plana”. (IA)

### 1.3.12 SK12: Brisanje plana

**Nаziv** **SK**

Brisanje korisnika

**Aktori SK**

*Korisnik*

**Učesnici SK**

*Korisnik i sistem (progrаm*)

**Preduslov**: Sistem je uključen i korisnikje ulogovan sa svojom šifrom. Sistem prikazuje korisnički panel.

**Osnovni scenаrio SK**

1. **Korisnik** unosi vrednost po kojoj se pretrаžuje *planove*. (APUSO)
2. **Korisnik** pozivа *sistem* dа nаđe *planove* po zаdаtoj vrednosti. (APSO)
3. **Sistem** trаži *planove* po zаdаtoj vrednosti. (SO)
4. **Sistem** prikаzuje *korisniku* pronađene *planove*. (IA)
5. **Korisnik** bira *plan* kojeg želi da obriše. (APUSO)
6. **Korisnik** poziva *sistem* da obriše izabrani *plan*. (APSO)
7. **Sistem** briše izabrani *plan*. (SO)
8. **Sistem** prikazuje *korisniku* poruku: “Plan uspesno obrisan”. (IA)

Alternativni scenariji

4.1 Ukoliko sistem ne može dа nаđe *planove* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe izabran *plan*”. Prekida se izvršenje scenaria. (IA)

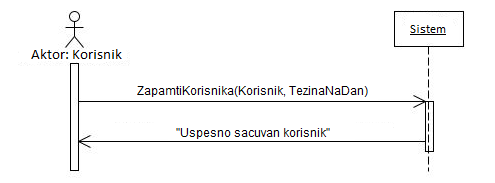
8.1. Ukoliko sistem ne može da obriše *plan* on prikazuje korisniku poruku: “Greska prilikom brisanja plana ”. (IA)

# 2. Analiza

## 2.1 Ponašanje softverskog sistema – Sistemski dijagram sekvenci

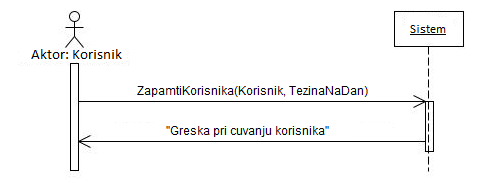
### 2.1.1 DS 1: Unos novog korisnika

1. **Korisnik** pozivа *sistem* dа zаpаmti podаtke o *korisniku*. (APSO)
2. **Sistem** prikаzuje *korisniku* zаpаmćenog *korisnika* i poruku: ʺUspesno sacuvan korisnik ʺ. (IA)



Alternativni scenariji

2.1 Ukoliko sistem ne može dа zаpаmti podаtke o novom *korisniku* on prikаzuje *korisniku* poruku “Greska pri cuvanju korisnika ”. (IA)

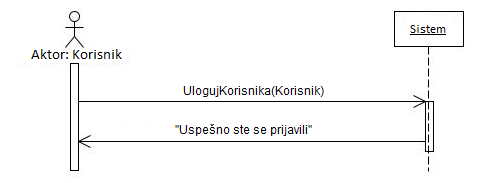


Sa navedenih sekvencnih dijagrama uočava jedna sistemska operacija:

1. signal **ZapamtiKorisnika**(Korisnik, TezinaNaDan)

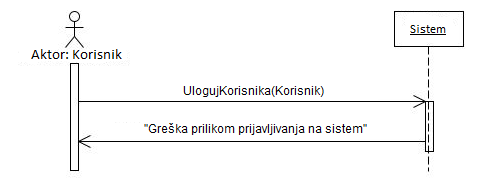
### 2.1.2 DS 2: Prijava korisnika

1. **Korisnik** pozivа *sistem* dа pronađe korisnika sa zadatim podacima. (APSO)
2. **Sistem** prikаzuje *korisniku* poruku: ʺUspešno ste se prijavili ʺ. (IA)



Alternativni scenariji

2.1. Ukoliko sistem ne može da nađe korisnika prikazuje poruku “Greška prilikom prijavljivanja na sistem!”. (IA)

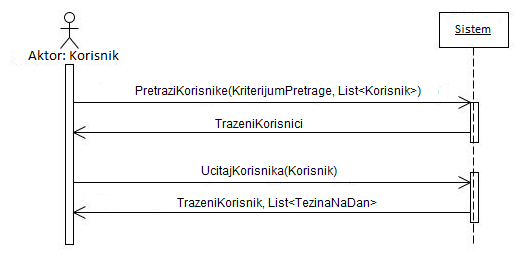


Sa navedenih sekvencnih dijagrama uočavaju se sledece sistemske operacije:

1. signal **UlogujKorisnika**(Korisnik)

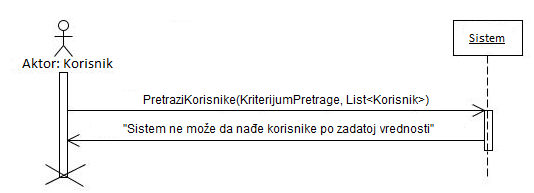
### 2.1.3 DS 3: Pretraživanje korisnika

1. **Korisnik** pozivа *sistem* dа nаđe *korisnike* po zаdаtoj vrednosti. (APSO)
2. **Sistem** prikazuje *korisniku* nađene *korisnike*. (IA)
3. **Korisnik** poziva sistem da prikaže podatke o izabranom *korisniku*. (APSO)
4. **Sistem** prikаzuje *korisniku* podаtke o *korisniku*. (IA)

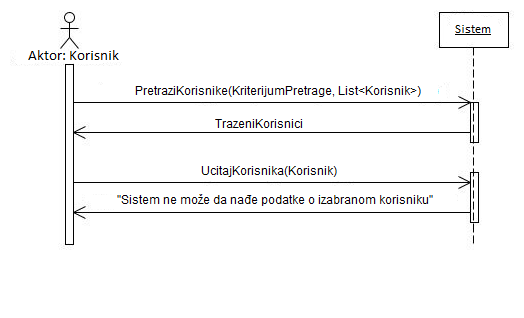


Alternativni scenariji

* 1. Ukoliko sistem ne može dа nаđe *korisnike* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe *korisnike* po zаdаtoj vrednosti”. Prekida se izvršenje scenaria. (IA)



4.1. Ukoliko sistem ne može da nađe izabranog *korisnika* on prikazuje korisniku poruku: “Sistem ne može da nađe podatke o izabranom korisniku ”. (IA)

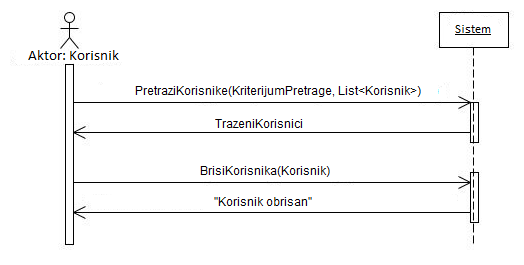


Sa navedenih sekvencnih dijagrama uočavaju se sledece sistemske operacije:

1. signal **PretraziKorisnike**(KriterijumPretrage, List<Korisnik>)
2. signal **UcitajKorisnika**(Korisnik)

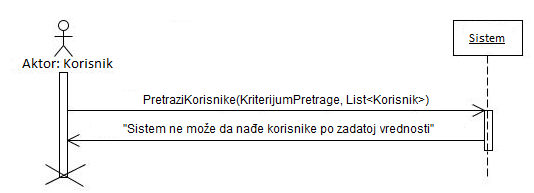
### 2.1.4 DS 4: Brisanje korisnika

1. **Korisnik** pozivа *sistem* dа nаđe *korisnike* po zаdаtoj vrednosti. (APSO)
2. **Sistem** prikаzuje *korisniku* pronađene *korisnike*. (IA)
3. **Korisnik** poziva *sistem* da obriše izabranog *korisnika*. (APSO)
4. **Sistem** prikazuje *korisniku* poruku: “Korisnik obrisan ”. (IA)

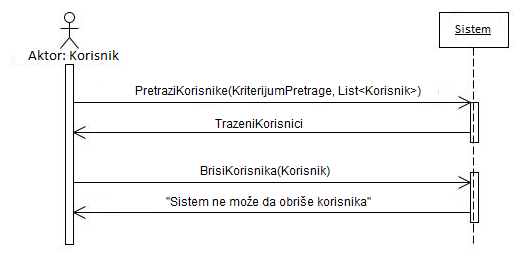


Alternativni scenariji

2.1 Ukoliko sistem ne može dа nаđe *korisnika* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe *korisnike* po zаdаtoj vrednosti”. Prekida se izvršenje scenaria. (IA)



4.1. Ukoliko sistem ne može da obriše *korisnika* on prikazuje korisniku poruku: “Sistem ne može da obriše korisnika ”. (IA)

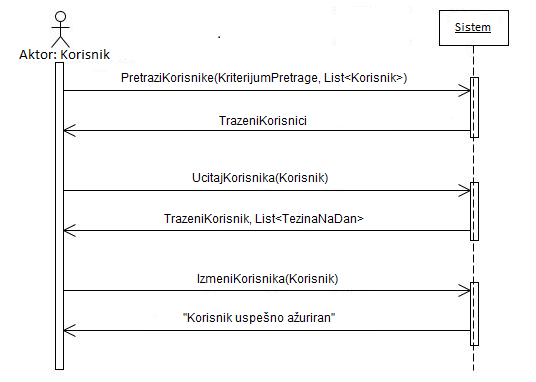


Sa navedenih sekvencnih dijagrama uočavaju se sledece sistemske operacije:

1. signal **PretraziKorisnike**(KriterijumPretrage, List<Korisnik>)
2. signal **BrisiKorisnika**(Korisnik)

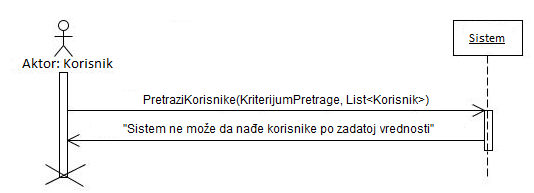
### 2.1.5 DS 5: Izmena korisnika

1. **Korisnik** pozivа *sistem* dа nаđe *korisnike* po zаdаtoj vrednosti. (APSO)
2. **Sistem** prikаzuje *korisniku* pronađene *korisnike.* (IA)
3. **Korisnik** pozivа *sistem* da učita podatke o izabranom *korisniku.* (APSO)
4. **Sistem** prikаzuje *korisniku* podatke o izabranom *korisniku.* (IA)
5. **Korisnik** pozivа *sistem* dа zаpаmti podatke o *korisniku*.(APSO)
6. **Sistem** prikаzuje *korisniku* poruku: “Korisnik uspešno ažuriran“. (IA)

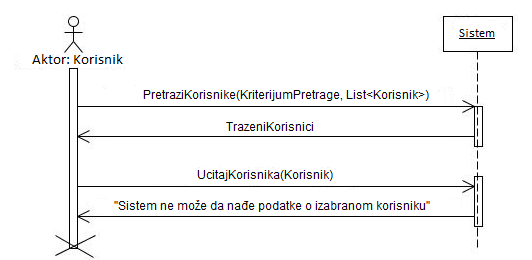


Alternativni scenariji

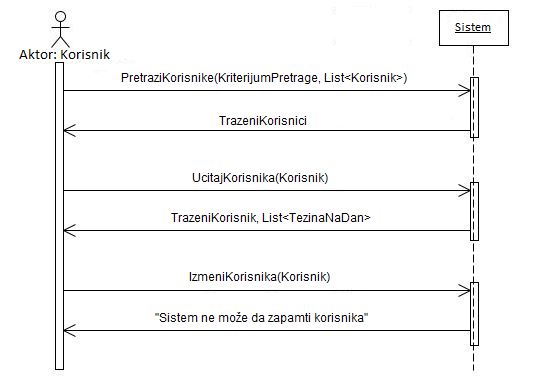
2.1 Ukoliko sistem ne može dа nаđe *korisnike* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe *korisnike* po zаdаtoj vrednosti”. Prekida se izvršenje scenaria. (IA)



4.1 Ukoliko sistem ne može dа nađe *korisnika* on prikаzuje korisniku poruku: “Sistem ne može da nađe podatke o izabranom korisniku”. Prekidа se izvršenje scenаriа. (IA)



6.1 Ukoliko sistem ne može dа zаpаmti izmenjene podаtke o *korisniku* on prikаzuje korisniku poruku “Sistem ne može dа zаpаmti korisnika”. (IA)

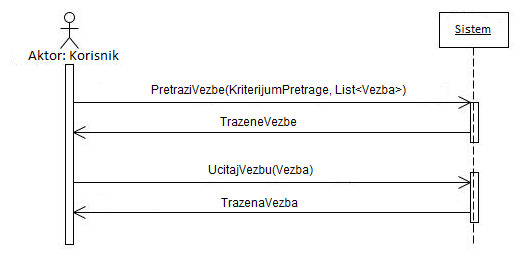


Sa navedenih sekvencnih dijagrama uočavaju se sledece sistemske operacije:

1. signal **PretraziKorisnike**(KriterijumPretrage, List<Korisnik>)
2. signal **UcitajKorisnika**(Korisnik)
3. signal **IzmeniKorisnika**(Korisnik)

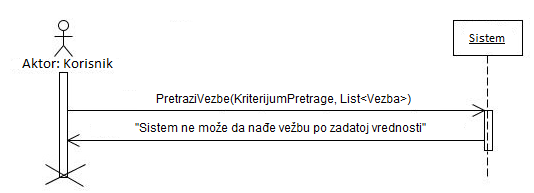
### 2.1.6 DS 6: Pretraživanje vežbi

1. **Korisnik** pozivа *sistem* dа nаđe *vežbe* po zаdаtoj vrednosti. (APSO)
2. **Sistem** prikazuje *korisniku* nađene *vežbe*. (IA)
3. **Korisnik** poziva sistem da prikaže podatke o izabranim *vežbama*. (APSO)
4. **Sistem** prikаzuje *korisniku* podаtke o *vežbi*. (IA)

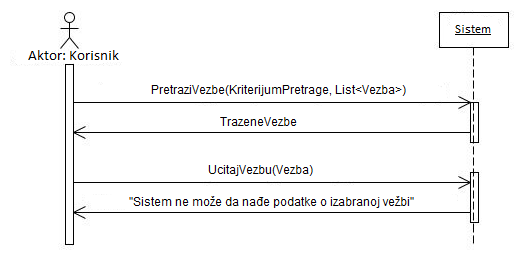


Alternativni scenariji

2.1 Ukoliko sistem ne može dа nаđe *vežbe* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe *vežbu* po zаdаtoj vrednosti”. Prekida se izvršenje scenaria. (IA)



4.1. Ukoliko sistem ne može da nađe izabranog *vežbe* on prikazuje korisniku poruku: “Sistem ne može da nađe podatke o izabranoj vežbi”. (IA)



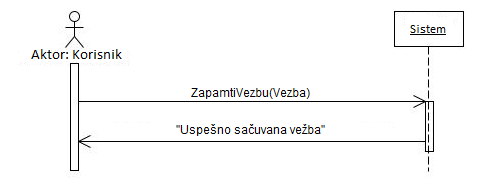
Sa navedenih sekvencnih dijagrama uočavaju se sledece sistemske operacije:

1. signal **PretraziVezbe**(KriterijumPretrage, List<Vezba>)
2. signal **UcitajVezbu**(Vezba)

### 2.1.7 DS 7: Unos nove vežbe

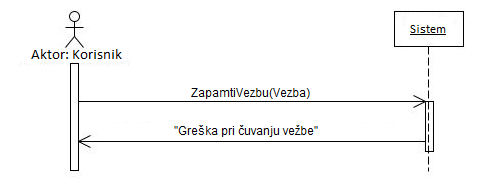
**Osnovni scenаrio SK**

1. **Korisnik** pozivа *sistem* dа zаpаmti podаtke o *vežbi*. (APSO)
2. **Sistem** prikаzuje *korisniku* zаpаmćenog *korisnika* i poruku: „Uspesno sacuvana vezbaʺ. (IA)



Alternativni scenariji

2.1 Ukoliko sistem ne može dа zаpаmti podаtke o novoj vežbi on prikаzuje *korisniku* poruku “Greska pri cuvanju vezbe”. (IA)

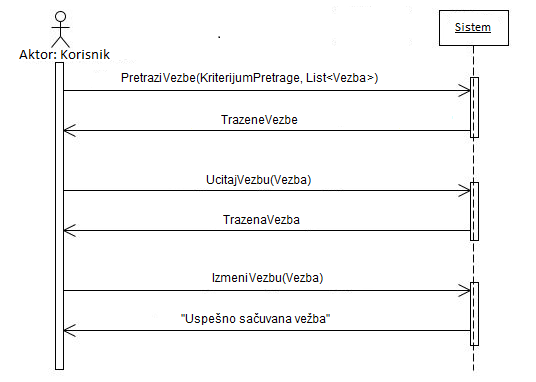


Sa navedenih sekvencnih dijagrama uočavaju se sledece sistemske operacije:

1. signal **ZapamtiVezbu**(Vezba)

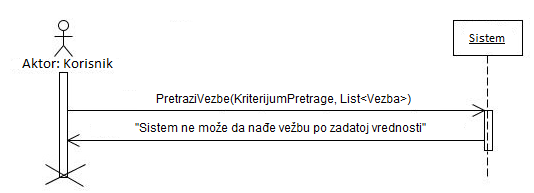
### 2.1.8 DS 8: Izmena vežbe

1. **Korisnik** pozivа *sistem* dа nаđe *vežbe* po zаdаtoj vrednosti. (APSO)
2. **Sistem** prikаzuje *korisniku* pronađene *vežbe*. (IA)
3. **Korisnik** pozivа *sistem* da učita podatke o izabranoj *vežbi.* (APSO)
4. **Sistem** prikаzuje *korisniku* podatke o izabranoj *vežbi.* (IA)
5. **Korisnik** pozivа *sistem* dа zаpаmti podatke o *vežbi*.(APSO)
6. **Sistem** prikаzuje *korisniku* poruku: „Uspesno sacuvana vezba“. (IA)

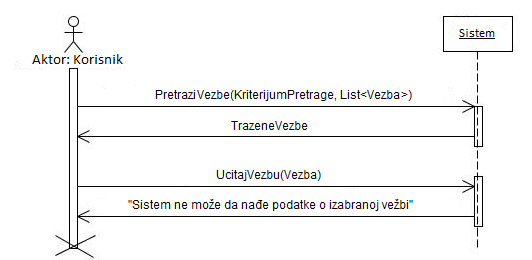


Alternativni scenariji

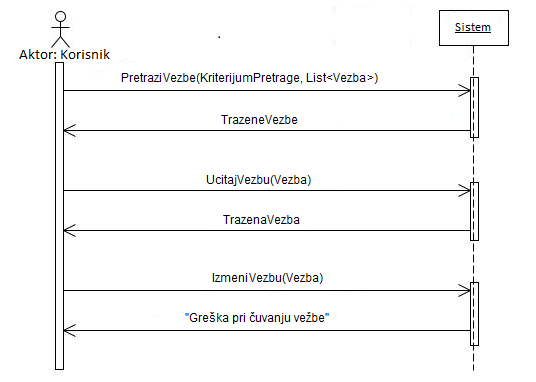
* 1. Ukoliko sistem ne može dа nаđe *vežbe* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe *vežbu* po zаdаtoj vrednosti”. Prekida se izvršenje scenaria. (IA)



4.1. Ukoliko sistem ne može da nađe izabranog *vežbe* on prikazuje korisniku poruku: “Sistem ne može da nađe podatke o izabranoj vežbi”. (IA)



6.1 Ukoliko sistem ne može dа zаpаmti izmenjene podаtke o *vežbi* on prikаzuje korisniku poruku “ Greska pri cuvanju vezbe”. (IA)

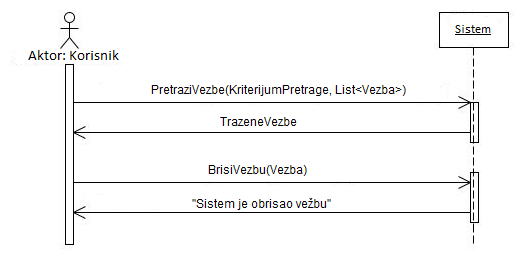


Sa navedenih sekvencnih dijagrama uočavaju se sledece sistemske operacije:

1. signal **PretraziVezbe**(KriterijumPretrage, List<Vezba>)
2. signal **UcitajVezbu**(Vezba)
3. signal **IzmeniVezbu**(Vezba)

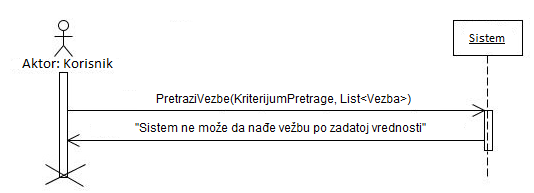
### 2.1.9 DS 9: Brisanje vežbe

1. **Korisnik** pozivа *sistem* dа nаđe *vežbe* po zаdаtoj vrednosti. (APSO)
2. **Sistem** prikаzuje *korisniku* pronađene *vežbe*. (IA)
3. **Korisnik** poziva *sistem* da obriše izabranu *vežbu*. (APSO)
4. **Sistem** prikazuje *korisniku* poruku: “Sistem je obrisao *vežbu*”. (IA)

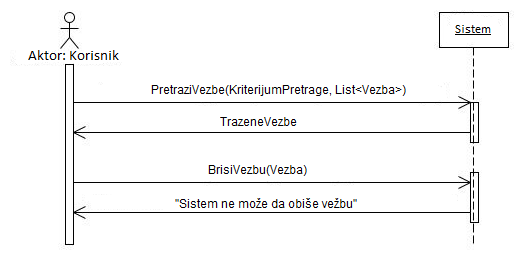


Alternativni scenariji

* 1. Ukoliko sistem ne može dа nаđe *korisnika* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe *vežbe* po zаdаtoj vrednosti”. Prekida se izvršenje scenaria. (IA)



4.1. Ukoliko sistem ne može da obriše *korisnika* on prikazuje korisniku poruku: “Sistem ne može da obriše *vežbu*”. (IA)

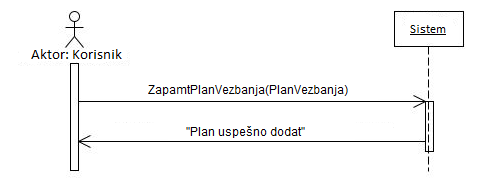


Sa navedenih sekvencnih dijagrama uočavaju se sledece sistemske operacije:

1. signal **PretraziVezbe**(KriterijumPretrage, List<Vezba>)
2. signal **BrisiVezbu**(Vezba)

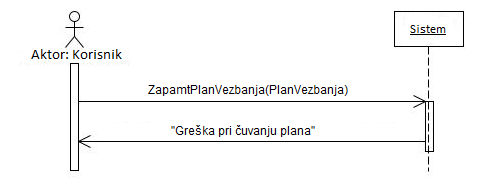
### 2.1.10 DS 10: Unos novog plana

1. **Korisnik** pozivа *sistem* dа zаpаmti podаtke o *planu*. (APSO)
2. **Sistem** prikаzuje *korisniku* zаpаmćeni *plan* i poruku: ʺPlan uspešno dodatʺ. (IA)



Alternativni scenariji

2.1 Ukoliko sistem ne može dа zаpаmti podаtke o novom *planu* on prikаzuje *korisniku* poruku “Greska pri cuvanju plana”. (IA)

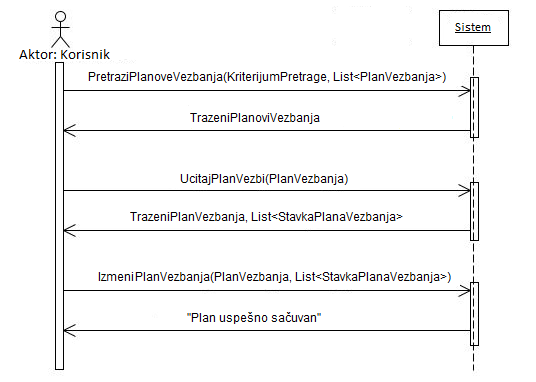


Sa navedenih sekvencnih dijagrama uočavaju se sledece sistemske operacije:

1. signal **ZapamtiPlanVezbanja**(PlanVezbanja)

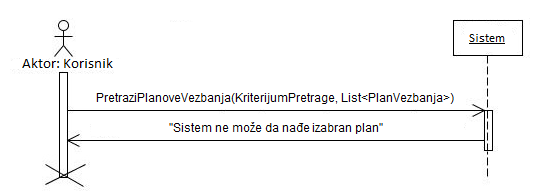
### 2.1.11 DS 11: Izmena plana

1. **Korisnik** pozivа *sistem* dа nаđe *planove* po zаdаtoj vrednosti. (APSO)
2. **Sistem** prikаzuje *korisniku* pronađene *planove*. (IA)
3. **Korisnik** pozivа *sistem* da učita podatke o izabranom *planu.* (APSO)
4. **Sistem** prikаzuje *korisniku* podatke o izabranom *planu.* (IA)
5. **Korisnik** pozivа *sistem* dа zаpаmti podatke o *planu*.(APSO)
6. **Sistem** prikаzuje *korisniku* poruku: „Plan uspešno sačuvan“. (IA)

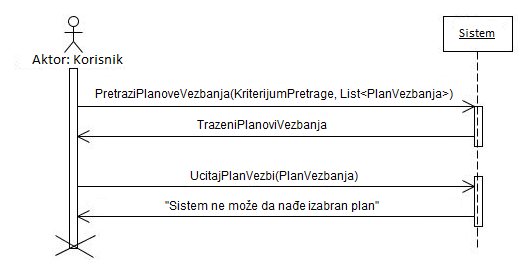


Alternativni scenariji

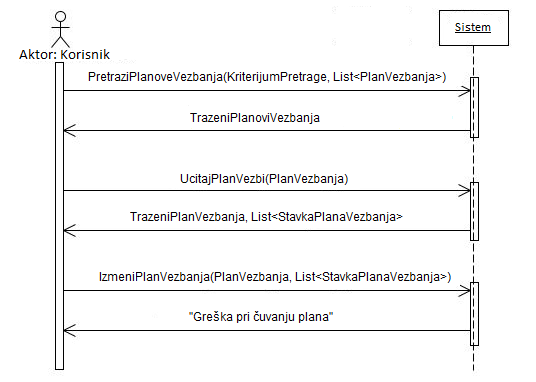
2.1 Ukoliko sistem ne može dа nаđe *planove* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe izabran plan”. Prekida se izvršenje scenaria. (IA)



4.1 Ukoliko sistem ne može dа nađe *plan* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe izabran plan”. Prekidа se izvršenje scenаriа. (IA)



6.1 Ukoliko sistem ne može dа zаpаmti izmenjene podаtke o *planu* on prikаzuje korisniku poruku “Greska pri cuvanju plana”. (IA)

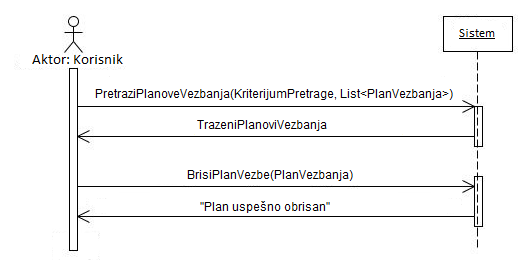


Sa navedenih sekvencnih dijagrama uočavaju se sledece sistemske operacije:

1. signal **PretraziPlanoveVezbanja**(KriterijumPretrage, List<PlanVezbanja>)
2. signal **UcitajPlanVezbi**(PlanVezbanja)
3. signal **IzmeniPlanVezbanja**(PlanVezbanja)

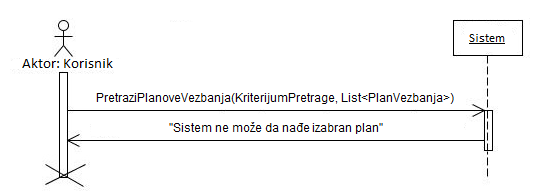
### 2.1.12 DS 12: Brisanje plana

1. **Korisnik** pozivа *sistem* dа nаđe *planove* po zаdаtoj vrednosti. (APSO)
2. **Sistem** prikаzuje *korisniku* pronađene *planove*. (IA)
3. **Korisnik** poziva *sistem* da obriše izabrani *plan*. (APSO)
4. **Sistem** prikazuje *korisniku* poruku: “Plan uspesno obrisan”. (IA)

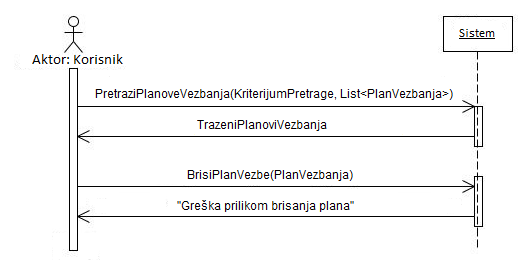


Alternativni scenariji

2.1 Ukoliko sistem ne može dа nаđe *planove* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe izabran *plan*”. Prekida se izvršenje scenaria. (IA)



4.1. Ukoliko sistem ne može da obriše *plan* on prikazuje korisniku poruku: “Greska prilikom brisanja plana ”. (IA)



Sa navedenih sekvencnih dijagrama uočavaju se sledece sistemske operacije:

1. signal **PretraziPlanoveVezbanja**(KriterijumPretrage, List<PlanVezbanja>)
2. signal **BrisiPlanVezbe**(PlanVezbanja)

Kao rezultat analize scenarija dobijeno je ukupno 10 sistemskih operacija koje treba projektovati:

1. signal **ZapamtiKorisnika**(Korisnik, TezinaNaDan)
2. signal **UlogujKorisnika**(Korisnik)
3. signal **PretraziKorisnike**(KriterijumPretrage, List<Korisnik>)
4. signal **UcitajKorisnika**(Korisnik)
5. signal **BrisiKorisnika**(Korisnik)
6. signal **IzmeniKorisnika**(Korisnik)
7. signal **PretraziVezbe**(KriterijumPretrage, List<Vezba>)
8. signal **UcitajVezbu**(Vezba)
9. signal **ZapamtiVezbu**(Vezba)
10. signal **IzmeniVezbu**(Vezba)
11. signal **BrisiVezbu**(Vezba)
12. signal **ZapamtiPlanVezbanja**(PlanVezbanja)
13. signal **PretraziPlanoveVezbanja**(KriterijumPretrage, List<PlanVezbanja>)
14. signal **UcitajPlanVezbi**(PlanVezbanja)
15. signal **IzmeniPlanVezbanja**(PlanVezbanja)
16. signal **BrisiPlanVezbe**(PlanVezbanja)

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

**Ugovor UG1: ZapamtiKorisnika**

Operacija: ZapamtiKorisnika(*Korisnik, TezinaNaDan*):signal;

Veza sa SK: SK1

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

Postuslovi: *Navodi se rezultat operacije.*

**Ugovor UG2: UlogujKorisnika**

Operacija: UlogujKorisnika(Korisnik):signal;

Veza sa SK: SK2

Preduslovi: -

Postuslovi: *Navodi se rezultat operacije.*

**Ugovor UG3: PretraziKorisnike**

Operacija: PretraziKorisnike(KriterijumPretrage, List<Korisnik>):signal;

Veza sa SK: SK3, SK4, SK5

Preduslovi: -

Postuslovi: *Navodi se rezultat operacije.*

**Ugovor UG4: UcitajKorisnika**

Operacija: UcitajKorisnika(Korisnik):signal;

Veza sa SK: SK3, SK5

Preduslovi: -

Postuslovi: *Navodi se rezultat operacije.*

**Ugovor UG5: BrisiKorisnika**

Operacija: BrisiKorisnika(Korisnik):signal;

Veza sa SK: SK4

Preduslovi: S*trukturna ograničenja moraju biti zadovoljena nad objektom Korisnik.*

Postuslovi: *Navodi se rezultat operacije.*

**Ugovor UG6: IzmeniKorisnika**

Operacija: IzmeniKorisnika(Korisnik):signal;

Veza sa SK: SK5

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

Postuslovi: *Navodi se rezultat operacije.*

**Ugovor UG7: PretraziVezbe**

Operacija: PretraziVezbe(KriterijumPretrage, List<Vezba>):signal;

Veza sa SK: SK6, SK8, SK9

Preduslovi: -

Postuslovi: *Navodi se rezultat operacije.*

**Ugovor UG8: UcitajVezbu**

Operacija: UcitajVezbu(Vezba):signal;

Veza sa SK: SK6, SK8

Preduslovi: -

Postuslovi: *Navodi se rezultat operacije.*

**Ugovor UG9: ZapamtiVezbu**

Operacija: ZapamtiVezbu(Vezba):signal;

Veza sa SK: SK7

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

Postuslovi: *Navodi se rezultat operacije.*

**Ugovor UG10: IzmeniVezbu**

Operacija: IzmeniVezbu(Vezba):signal;

Veza sa SK: SK8

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

Postuslovi: *Navodi se rezultat operacije.*

**Ugovor UG11: BrisiVezbu**

Operacija: BrisiVezbu(Vezba):signal;

Veza sa SK: SK9

Preduslovi: S*trukturna ograničenja moraju biti zadovoljena nad objektom Vezba.*

Postuslovi: *Navodi se rezultat operacije.*

**Ugovor UG12: ZapamtiPlanVezbanja**

Operacija: ZapamtiPlanVezbanja(PlanVezbanja):signal;

Veza sa SK: SK10

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

Postuslovi: *Navodi se rezultat operacije.*

**Ugovor UG13: PretraziPlanoveVezbanja**

Operacija: PretraziPlanoveVezbanja(KriterijumPretrage, List<PlanVezbanja>);

Veza sa SK: SK11, SK12

Preduslovi: -

Postuslovi: *Navodi se rezultat operacije.*

**Ugovor UG14: UcitajPlanVezbi**

Operacija: UcitajPlanVezbi(PlanVezbanja):signal;

Veza sa SK: SK11

Preduslovi: -

Postuslovi: *Navodi se rezultat operacije.*

**Ugovor UG15: IzmeniPlanVezbanja**

Operacija: IzmeniPlanVezbanja(PlanVezbanja):signal;

Veza sa SK: SK11

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

Postuslovi: *Navodi se rezultat operacije.*

**Ugovor UG16: BrisiPlanVezbe**

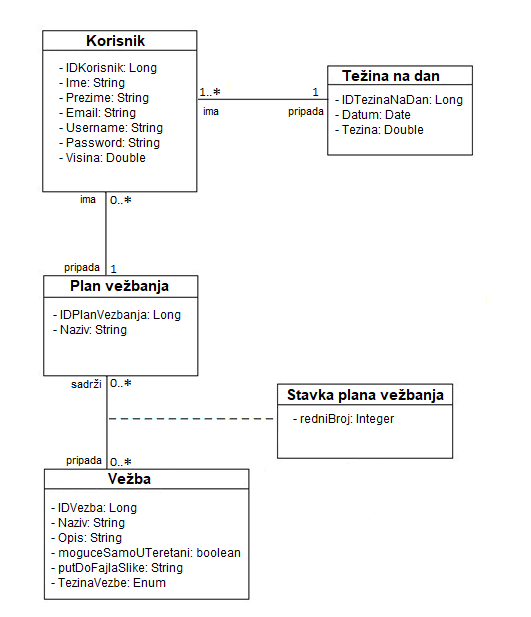
Operacija: BrisiPlanVezbe(PlanVezbanja):signal;

Veza sa SK: SK12

Preduslovi: S*trukturna ograničenja moraju biti zadovoljena nad objektom PlanVezbanja.*

Postuslovi: *Navodi se rezultat operacije.*

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



## 2.4. Struktura softverskog sistema – Relacioni model

Korisnik(IDKorisnik, Ime, Prezime, Email, Username, Password, Visina)

PlanVežbanja(IDPlanVezbanja, Naziv, *IDKorisnik*)

Vežba(IDVezba, Naziv, Opis, MoguceSamoUTeretani, PutDoFajlaSlike, TezinaVezbe)

TežinaNaDan(IDTezinaNaDan, Datum, Tezina, *IDKorisnik*)

StavkaPlanaVežbanja(IDPlanVezbanja, IDVezba, RedniBroj)

## 2.5. Struktura softverskog sistema – Tabela Ograničenja

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Korisnik** | | **Prosto vrednosno ograničenje** | | **Složeno vrednosno ograničenje** | | **Strukturno ograničenje** |
| **Atributi** | **Ime** | **Tip atributa** | **Vrednost atributa** | **Međuzav. atributa jedne tabele** | **Međuzav. atributa više tabela** | INSERT /  UPDATE CASCADES  TezinaNaDan, PlanVezbanja  DELETE CASCADES  TezinaNaDan, PlanVezbanja |
| IDKorisnik | Integer | not null and >0 |  |  |
| Ime | String | not null and length<45 |  |  |
| Prezime | String | not null and length<45 |  |  |
| Email | String | not null and length<45 |  |  |
| Username | String | not null and length<45 |  |  |
| Password | String | not null and length>5 and length<45 |  |  |
| Visina | Double | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Tezina Na Dan** | | **Prosto vrednosno ograničenje** | | **Složeno vrednosno ograničenje** | | **Strukturno ograničenje** |
| **Atributi** | **Ime** | **Tip atributa** | **Vrednost atributa** | **Međuzav. atributa jedne tabele** | **Međuzav. atributa više tabela** | INSERT /  UPDATE /  DELETE RESTRICTED Korisnik |
| IDTezinaNaDan | Integer | not null and >0 |  |  |
| Datum | DateTime | not null |  |  |
| Tezina | Double | not null |  |  |
| IDKorisnik | Integer | not null and >0 |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Plan Vezbanja** | | **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 StavkaPlana  DELETE CASCADES StavkaPlana |
| IDPlanVezbanja | Integer | not null and >0 |  |  |
| Naziv | String | not null |  |  |
| IDKorisnik | Integer | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Stavka Plana** | | **Prosto vrednosno ograničenje** | | **Složeno vrednosno ograničenje** | | **Strukturno ograničenje** |
| **Atributi** | **Ime** | **Tip atributa** | **Vrednost atributa** | **Međuzav. atributa jedne tabele** | **Međuzav. atributa više tabela** | INSERT /  UPDATE /  DELETE / |
| RedniBroj | Integer | not null and >0 |  |  |
| IDKorisnik | Integer | not null and >0 |  |  |
| IDPlanVezbanja | Integer | not null and >0 |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Vezba** | | **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 StavkaPlana  DELETE CASCADES StavkaPlana |
| IDVezba | Integer | not null and >0 |  |  |
| Naziv | String | not null and length<45 |  |  |
| Opis | String | not null and length<255 |  |  |
| moguceSamoUTeretani | Boolean | (default:0) |  |  |
| PutDoFajla | String | not null and length<45 |  |  |
| TezinaVezbe | String | in(“Laka“,“Srednja“ ,“Teska“) (default:“Laka“) |  |  |

# 3.Projektovanje

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

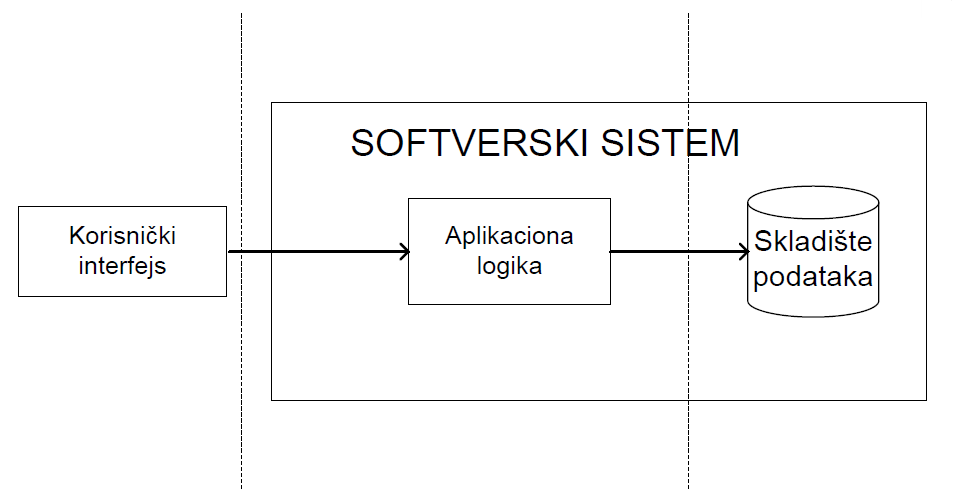
## 3.1 Arhitektura softverskog sistema

Arrhitektura sistema je tronivojska i sastoji se od sledećih nivoa:

• korisnički interfejs

• aplikaciona logika

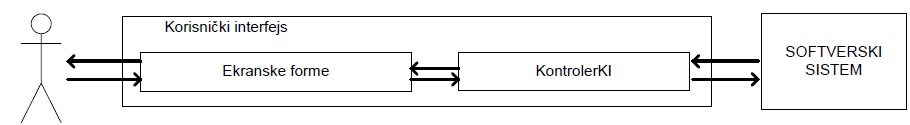
• skladište podataka

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

Slika 1. Tronivojska Arhitektura

## 3.2 Projektovanje korisničkog interfejsa

Projektovanje korisničkog interfejsa predstavlja realizaciju ulaza i/ili izlaza softverskog sistema.



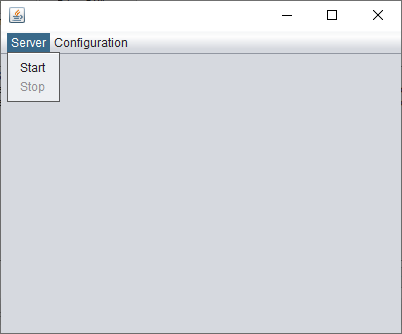
Slika 2. Struktura korisnickog interfejsa

Ekranska forma ima ulogu da prihvati podatke koje korisnik unosi, prihvata događaje koje pravi korisnik, poziva kontrolera korisničkog interfejsa kako bi mu prosledio podatke i prikazuje podatke dobijene od kontrolera korisničkog interfejsa.

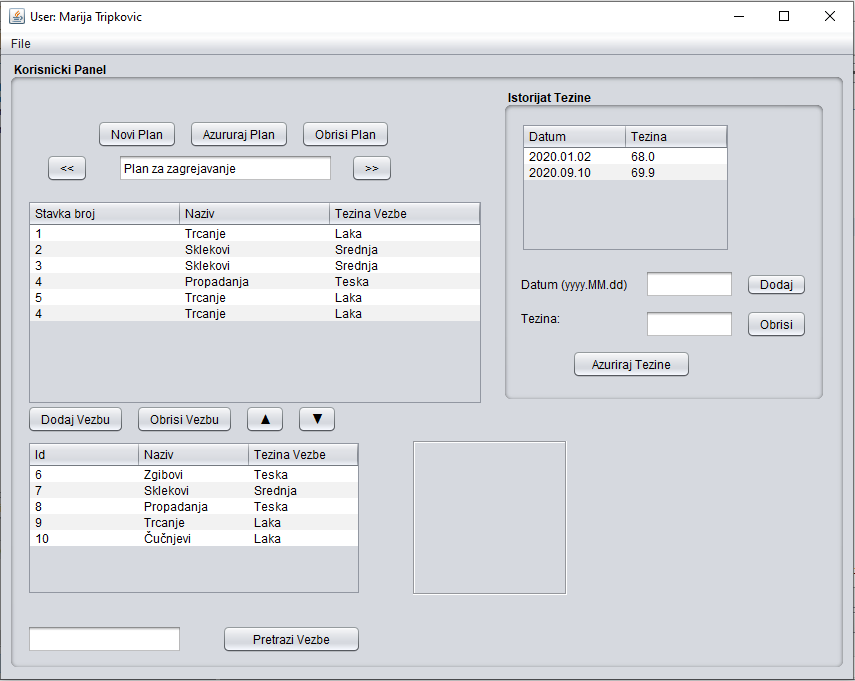
## 3.2.1 Projektovanje ekranskih formi

Korisnički interfejs defenisan 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:



Glavna ekranska forma klijentskog dela aplikacije izgledala bi ovako:



Kroz slučajeve korišćenja projektovane su i ostale ekranske forme koje će aplikacija posedovati, a koje se pozivaju iz glavne ekranske forme klijentskog dela aplikacije.

### 3.2.1.1 SK1: Unos novog korisnika

**Nаziv** **SK**

Unos korisnika

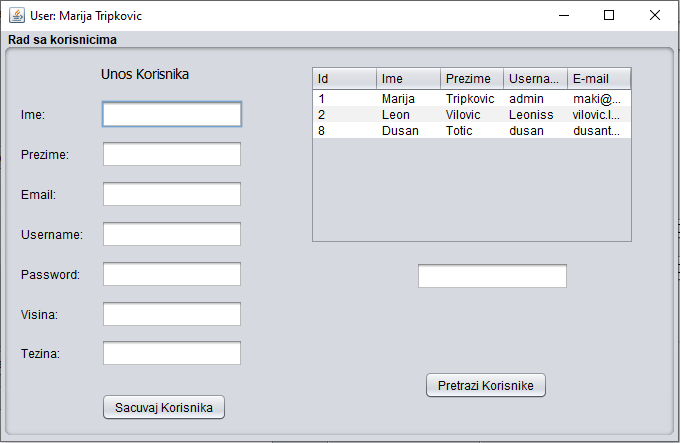
**Aktori SK**

*Korisnik*

**Učesnici SK**

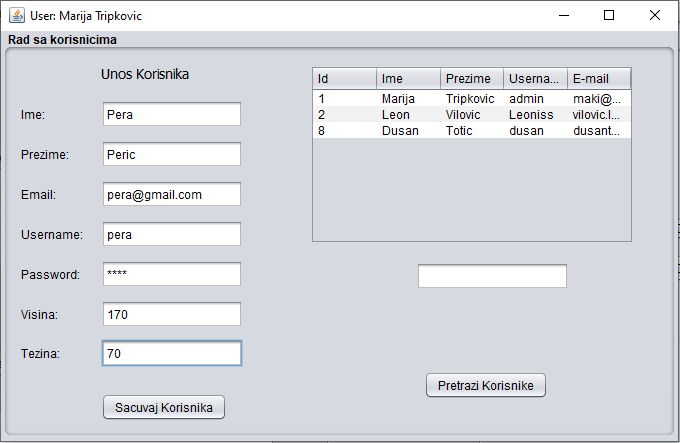
*Korisnik i sistem (progrаm)*

**Preduslov**: Sistem je uključen i korisnikje ulogovan sa svojom šifrom. Sistem prikazuje formu za rad sa korisnicima.

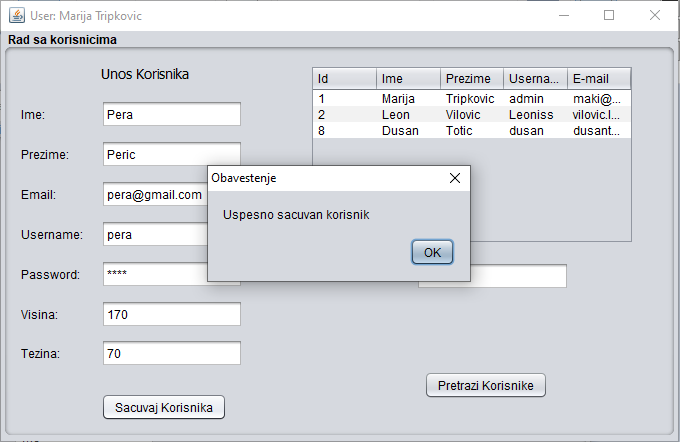


**Osnovni scenаrio SK**

1. **Korisnik** unosi podаtke o novom *korisniku*. (APUSO)*Opis akcije*: Sva polja su obavezna i nad svim poljima se vrši validacija

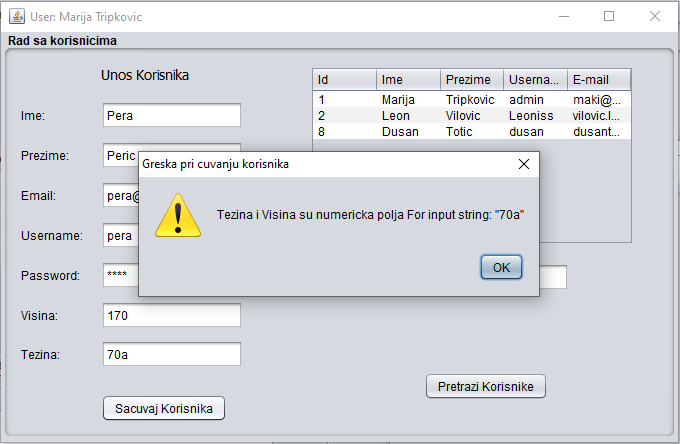


1. **Korisnik** kontroliše dа li je korektno uneo podаtke o novom *korisniku*. (ANSO)
2. **Korisnik** pozivа *sistem* dа zаpаmti podаtke o *korisniku*. (APSO)
3. **Sistem** pаmti podаtke o *korisniku*. (SO)
4. **Sistem** prikаzuje *korisniku* zаpаmćenog *korisnika* i poruku: ʺUspesno sacuvan korisnik ʺ. (IA)



Alternativni scenariji

5.1 Ukoliko sistem ne može dа zаpаmti podаtke o novom *korisniku* on prikаzuje *korisniku* poruku “Greska pri cuvanju korisnika ”. (IA)



### 3.2.1.2 SK2: Prijava korisnika

**Nаziv** **SK**

Prijava korisnika

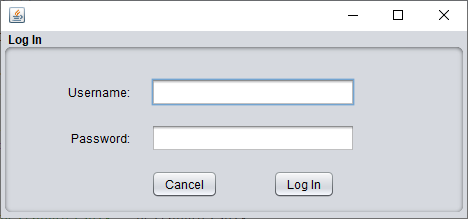
**Aktori SK**

*Korisnik*

**Učesnici SK**

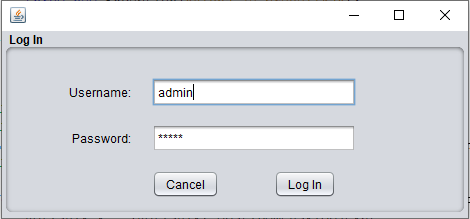
*Korisnik* *i sistem (progrаm)*

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

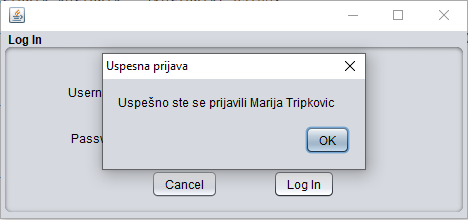


**Osnovni scenаrio SK**

1. **Korisnik** unosi podаtke za autentifikaciju. (APUSO)

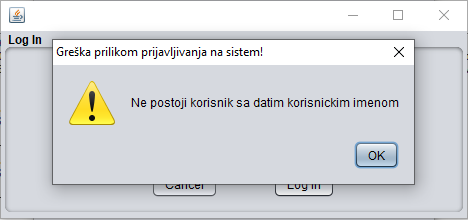
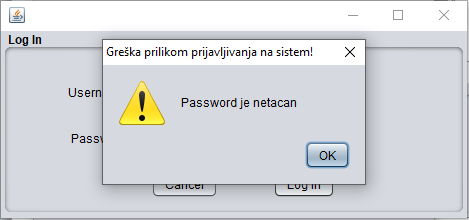


1. **Korisnik** pozivа *sistem* dа pronađe korisnika sa zadatim podacima. (APSO)
2. **Sistem** pretražuje korisnike. (SO)
3. **Sistem** prikаzuje *korisniku* poruku: ʺUspešno ste se prijavili ʺ. (IA)



Alternativni scenariji

* 1. Ukoliko sistem ne može da nađe korisnika prikazuje poruku “Greška prilikom prijavljivanja na sistem!”. (IA)



### 3.2.1.3 SK3: Pretraživanje korisnika

**Nаziv SK**

Pretrаživаnje korisnina

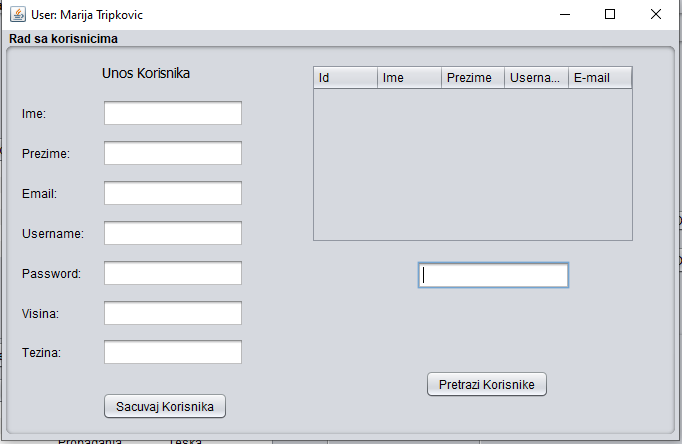
**Aktori SK**

*Korisnik*

**Učesnici SK**

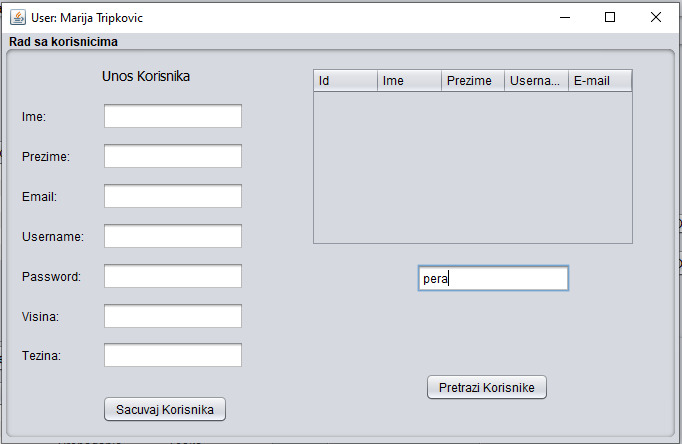
*Korisnik i sistem (progrаm*)

**Preduslov**: Sistem je uključen. Sistem prikazuje formu za rad sa korisnicima.

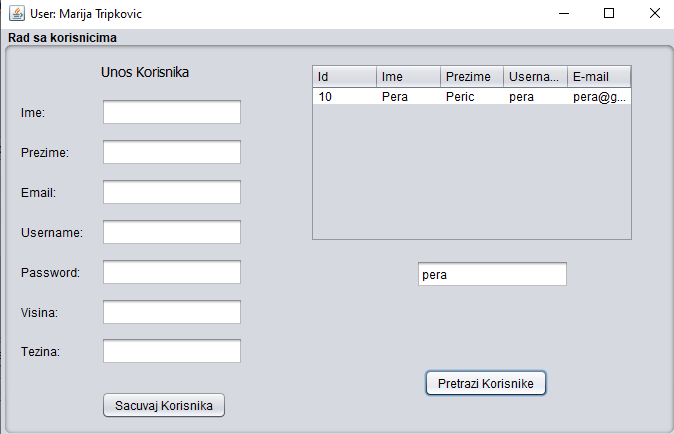


**Osnovni scenаrio SK**

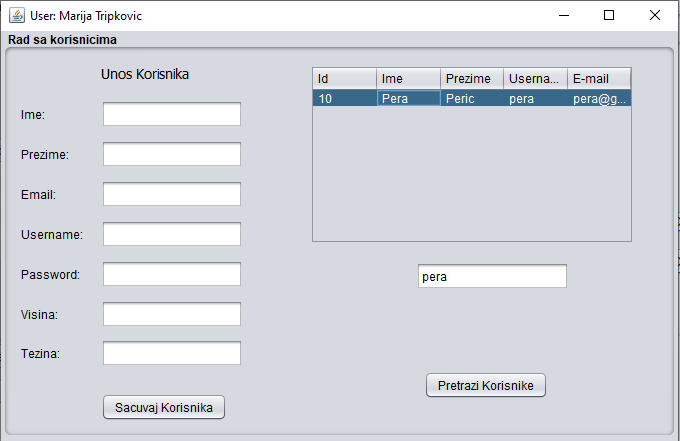
1. **Korisnik** unosi vrednost po kojoj pretrаžuje *korisnike*. (APUSO)



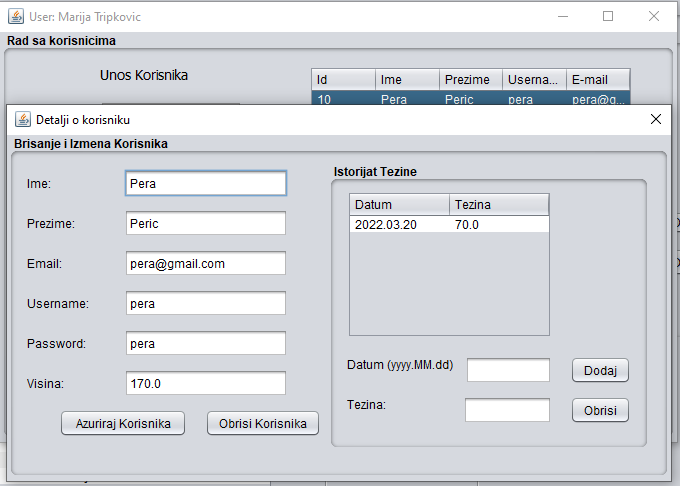
1. **Korisnik** pozivа *sistem* dа nаđe *korisnike* po zаdаtoj vrednosti. (APSO)
2. **Sistem** trаži *korisnike* po zаdаtoj vrednosti. (SO)
3. **Sistem** prikazuje *korisniku* nađene *korisnike*. (IA)



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

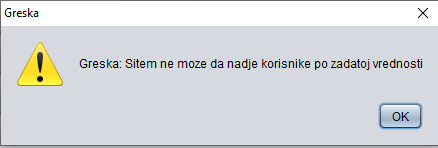


1. **Korisnik** poziva sistem da prikaže podatke o izabranom *korisniku*. (APSO)
2. **Sistem** traži podatke o izabranom *korisniku*. (SO)
3. **Sistem** prikаzuje *korisniku* podаtke o *korisniku*. (IA)

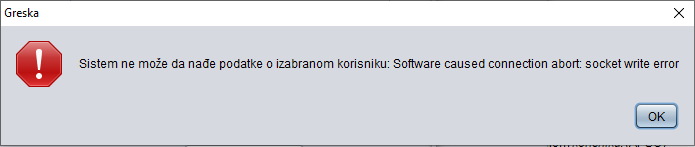


Alternativni scenariji

4.1 Ukoliko sistem ne može dа nаđe *korisnike* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe *korisnike* po zаdаtoj vrednosti”. Prekida se izvršenje scenaria. (IA)



8.1. Ukoliko sistem ne može da nađe izabranog *korisnika* on prikazuje korisniku poruku: “Sistem ne može da nađe podatke o izabranom korisniku ”. (IA)



### 3.2.1.4 SK4: Brisanje korisnika

**Nаziv** **SK**

Brisanje korisnika

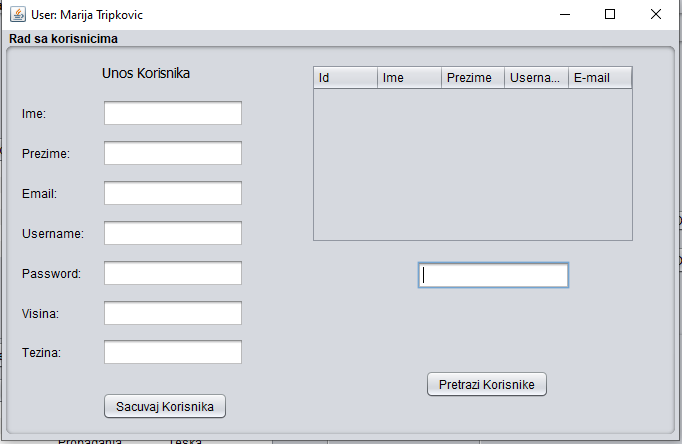
**Aktori SK**

*Korisnik*

**Učesnici SK**

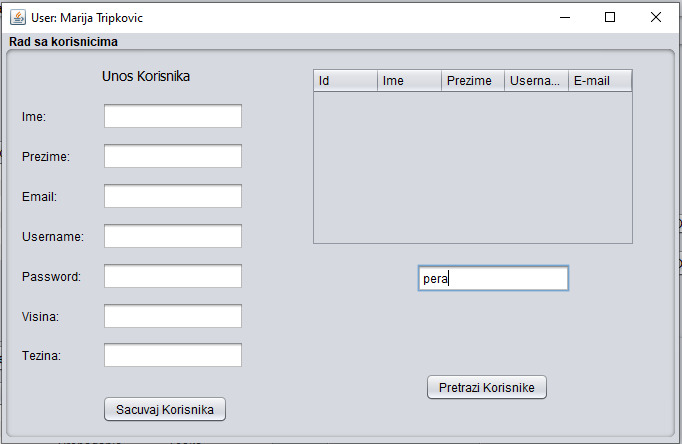
*Korisnik i sistem (progrаm*)

**Preduslov**: Sistem je uključen i korisnikje ulogovan sa svojom šifrom. Sistem prikazuje formu za rad sa korisnicima.

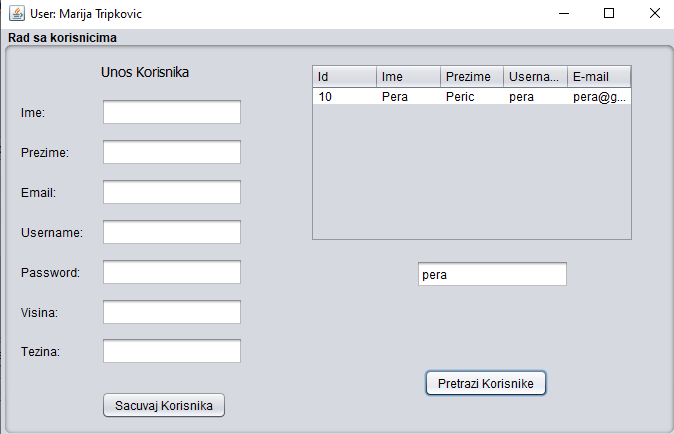


**Osnovni scenаrio SK**

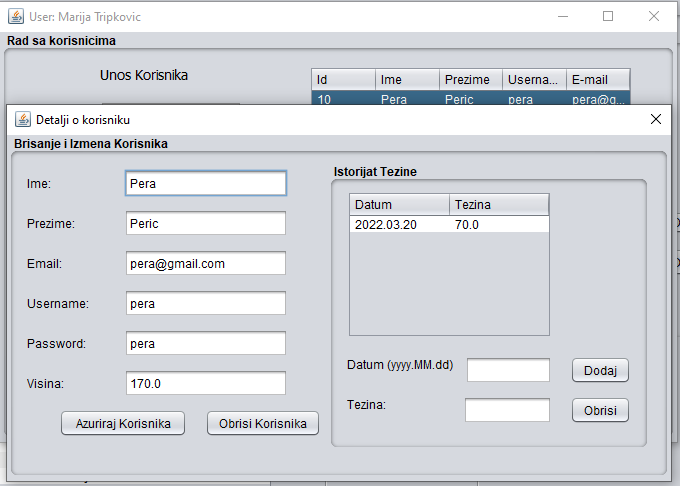
1. **Korisnik** unosi vrednost po kojoj se pretrаžuje *korisnike*. (APUSO)



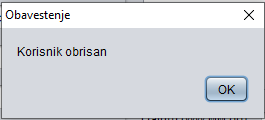
1. **Korisnik** pozivа *sistem* dа nаđe *korisnike* po zаdаtoj vrednosti. (APSO)
2. **Sistem** trаži *korisnike* po zаdаtoj vrednosti. (SO)
3. **Sistem** prikаzuje *korisniku* pronađene *korisnike*. (IA)



1. **Korisnik** bira *korisnika* kojeg želi da obriše. (APUSO)
2. **Korisnik** poziva *sistem* da obriše izabranog *korisnika*. (APSO)

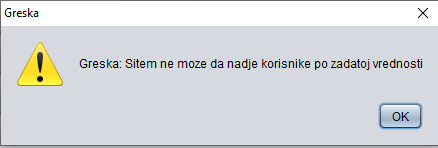


1. **Sistem** briše izabranog *korisnika*. (SO)
2. **Sistem** prikazuje *korisniku* poruku: “Korisnik obrisan ”. (IA)

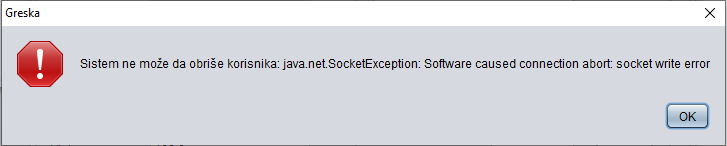


Alternativni scenariji

4.1 Ukoliko sistem ne može dа nаđe *korisnika* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe *korisnike* po zаdаtoj vrednosti”. Prekida se izvršenje scenaria. (IA)



8.1. Ukoliko sistem ne može da obriše *korisnika* on prikazuje korisniku poruku: “Sistem ne može da obriše korisnika ”. (IA)



### 3.2.1.5 SK5: Izmena korisnika (složen slučaj korišćenja)

**Nаziv SK**

Izmenа korisnika

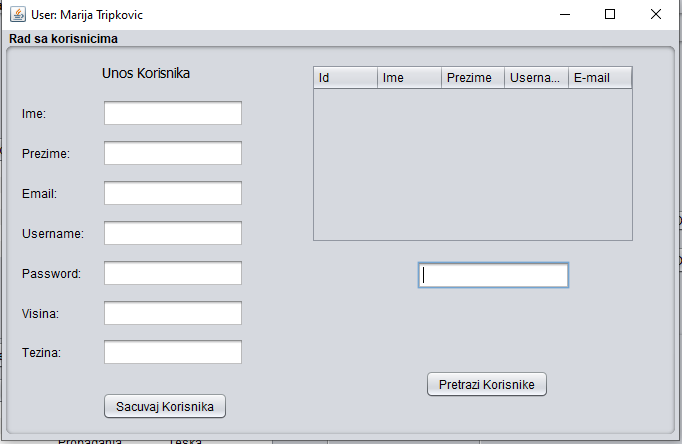
**Aktori SK**

*Korisnik*

**Učesnici SK**

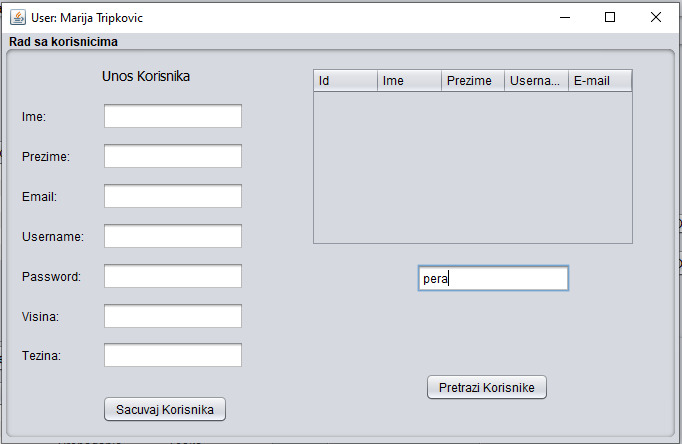
*Korisnik i sistem (progrаm*)

**Preduslov**: Sistem je uključen i korisnikje ulogovan sa svojom šifrom. Sistem prikazuje formu za rad sa korisnicima.

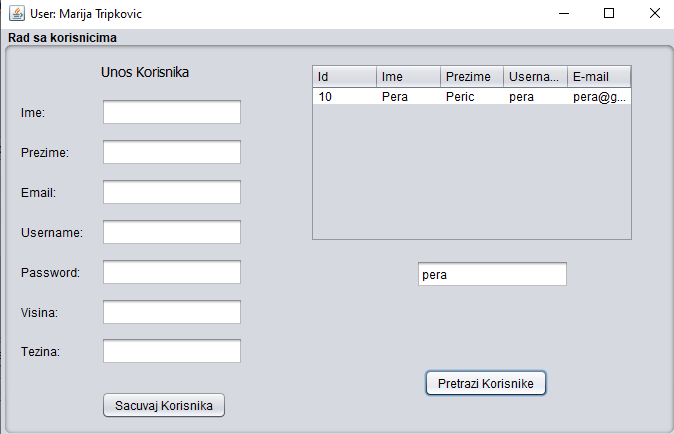


**Osnovni scenаrio SK**

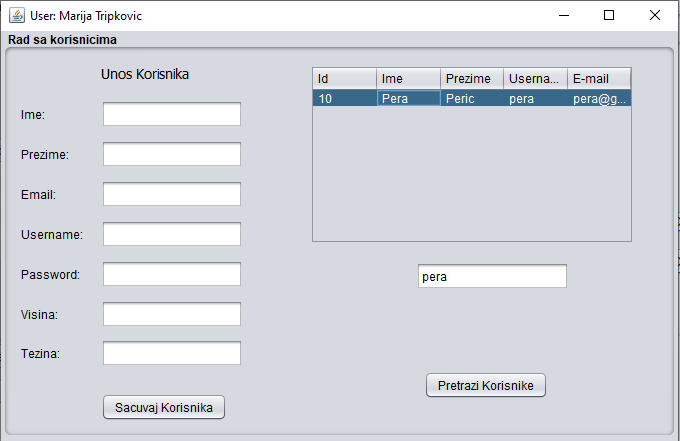
1. **Korisnik** unosi vrednost po kojoj se pretrаžuju *korisnici*. (APUSO)



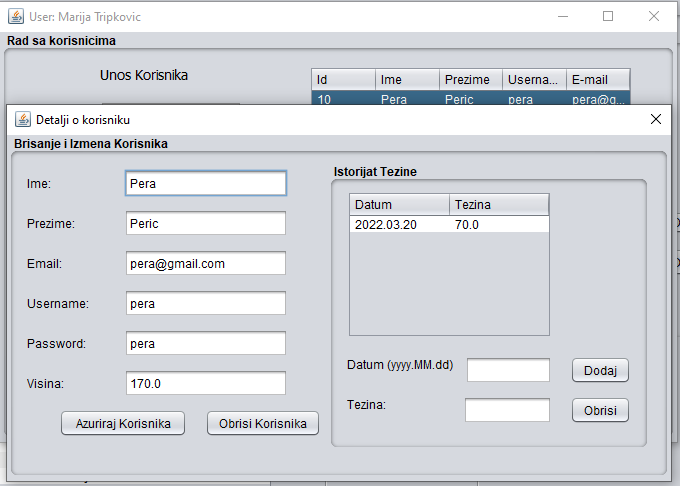
1. **Korisnik** pozivа *sistem* dа nаđe *korisnike* po zаdаtoj vrednosti. (APSO)
2. **Sistem** trаži *korisnike* po zаdаtoj vrednosti. (SO)
3. **Sistem** prikаzuje *korisniku* pronađene *korisnike.* (IA)



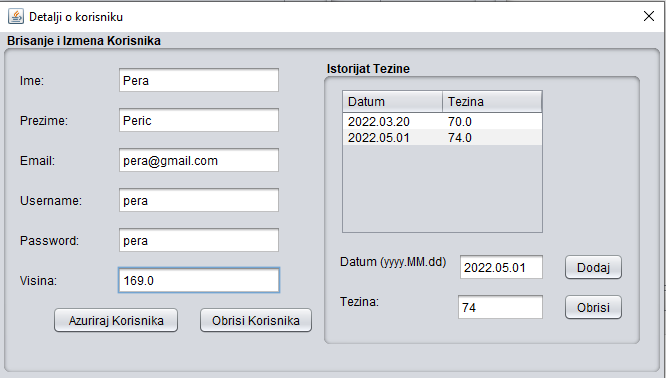
1. **Korisnik** bira *korisnika* kojeg želi da izmeni. (APUSO)



1. **Korisnik** pozivа *sistem* da učita podatke o izabranom *korisniku.* (APSO)
2. **Sistem** trаži podatke o izabranom *korisniku.* (SO)
3. **Sistem** prikаzuje *korisniku* podatke o izabranom *korisniku.* (IA)

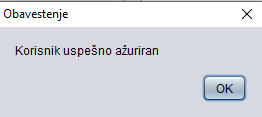


1. **Korisnik** unosi (menja) *korisnika*. (APUSO)



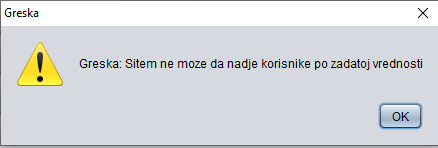
*Opis akcije*: Pored uobičajenih izmena, moguće je menjati i stavke istorijata težine

1. **Korisnik** kontroliše dа li je korektno uneo podаtke o *korisniku*. (ANSO)
2. **Korisnik** pozivа *sistem* dа zаpаmti podatke o *korisniku*.(APSO)
3. **Sistem** pаmti podatke o *korisniku*. (SO)
4. **Sistem** prikаzuje *korisniku* poruku: “Korisnik uspešno ažuriran“. (IA)

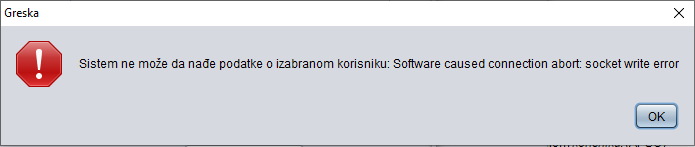


Alternativni scenariji

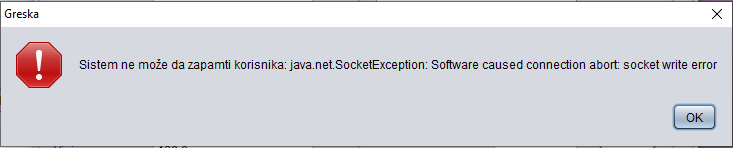
4.1 Ukoliko sistem ne može dа nаđe *korisnike* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe *korisnike* po zаdаtoj vrednosti”. Prekida se izvršenje scenaria. (IA)



8.1 Ukoliko sistem ne može dа nađe *korisnika* on prikаzuje korisniku poruku: “Sistem ne može da nađe podatke o izabranom korisniku”. Prekidа se izvršenje scenаriа. (IA)



13.1 Ukoliko sistem ne može dа zаpаmti izmenjene podаtke o *korisniku* on prikаzuje korisniku poruku “Sistem ne može dа zаpаmti korisnika”. (IA)



### 3.2.1.6 SK6: Pretraživanje vežbi

**Nаziv SK**

Pretrаživаnje vežbi

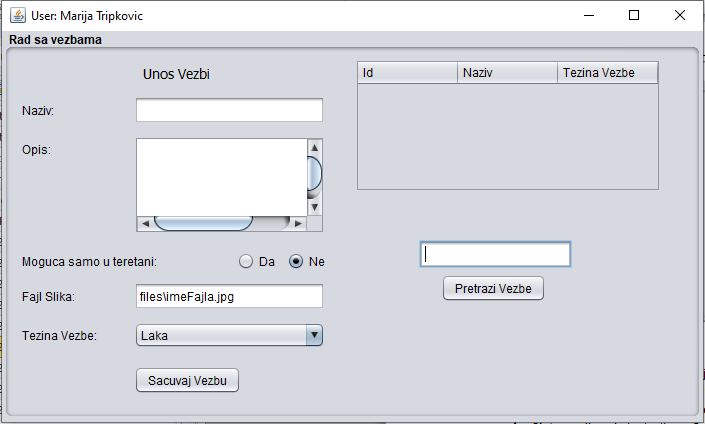
**Aktori SK**

*Korisnik*

**Učesnici SK**

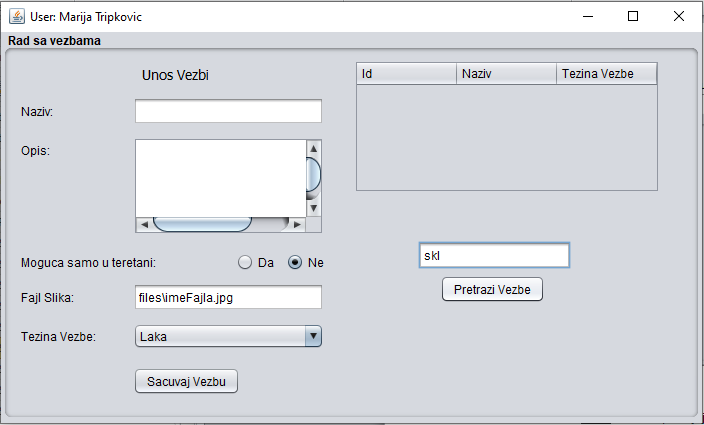
*Korisnik i sistem (progrаm*)

**Preduslov**: Sistem je uključen. Sistem prikazuje formu za rad sa vežbama ili korisnički panel.



**Osnovni scenаrio SK**

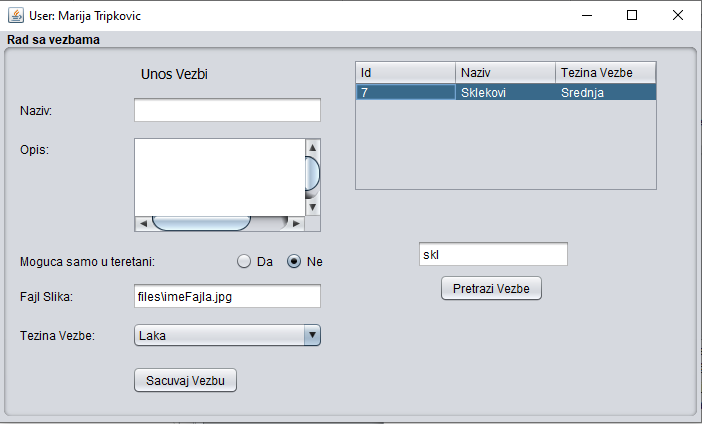
1. **Korisnik** unosi vrednost po kojoj pretrаžuje *vežbe*. (APUSO)



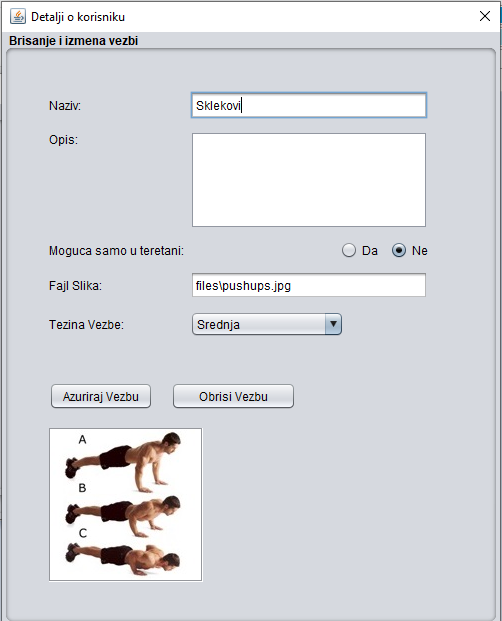
1. **Korisnik** pozivа *sistem* dа nаđe *vežbe* po zаdаtoj vrednosti. (APSO)
2. **Sistem** trаži *vežbe* po zаdаtoj vrednosti. (SO)
3. **Sistem** prikazuje *korisniku* nađene *vežbe*. (IA)



1. **Korisnik** bira*vežbe* čije podatke želi da vidi. (APUSO)
2. **Korisnik** poziva sistem da prikaže podatke o izabranim *vežbama*. (APSO)

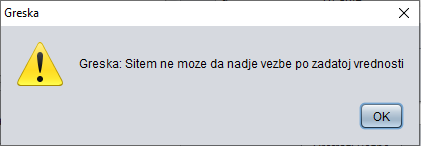


1. **Sistem** traži podatke o izabranoj *vežbi*. (SO)
2. **Sistem** prikаzuje *korisniku* podаtke o *vežbi*. (IA)

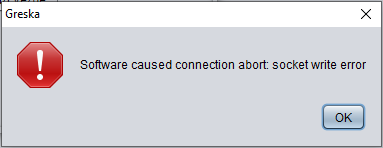


Alternativni scenariji

4.1 Ukoliko sistem ne može dа nаđe *vežbe* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe *vežbu* po zаdаtoj vrednosti”. Prekida se izvršenje scenaria. (IA)



8.1. Ukoliko sistem ne može da nađe izabrane *vežbe* on prikazuje korisniku poruku: “Sistem ne može da nađe podatke o izabranoj vežbi”. (IA)



### 3.2.1.7 SK7: Unos nove vežbe

**Nаziv** **SK**

Unos vežbe

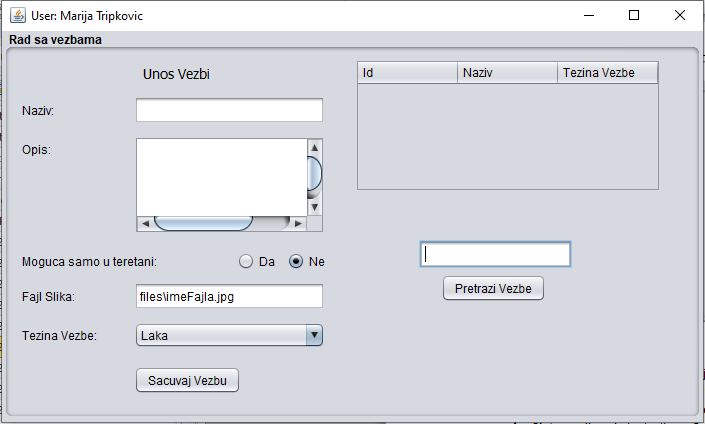
**Aktori SK**

*Korisnik*

**Učesnici SK**

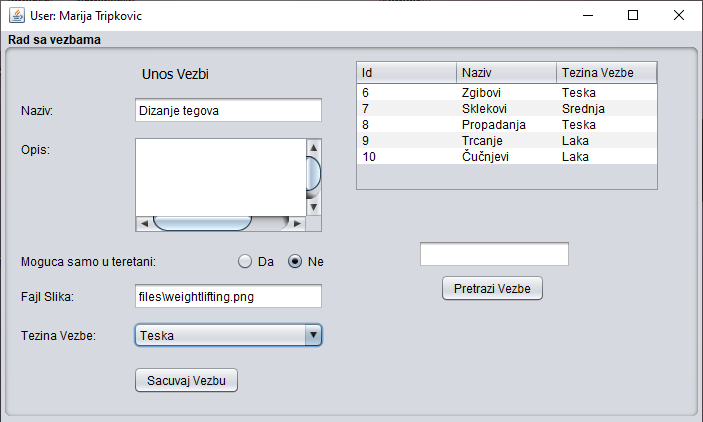
*Korisnik i sistem (progrаm)*

**Preduslov**: Sistem je uključen i korisnikje ulogovan sa svojom šifrom. Sistem prikazuje formu za rad sa vežbama.



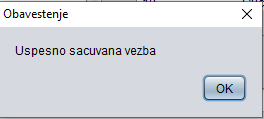
**Osnovni scenаrio SK**

1. **Korisnik** unosi podаtke o novoj *vežbi*. (APUSO)



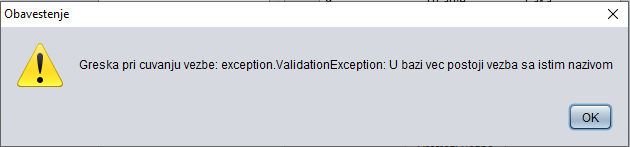
*Opis akcije*: Polje naziv je obavezno i mora biti jedinstveno. Opis može da sadrži najviše 255 karaktera.

1. **Korisnik** kontroliše dа li je korektno uneo podаtke o novoj *vežbi*. (ANSO)
2. **Korisnik** pozivа *sistem* dа zаpаmti podаtke o *vežbi*. (APSO)
3. **Sistem** pаmti podаtke o *vežbi*. (SO)
4. **Sistem** prikаzuje *korisniku* zаpаmćenog *korisnika* i poruku: „Uspesno sacuvana vezbaʺ. (IA)



Alternativni scenariji

* 1. Ukoliko sistem ne može dа zаpаmti podаtke o novoj vežbi on prikаzuje *korisniku* poruku “Greska pri cuvanju vezbe”. (IA)



### 3.2.1.8 SK8: Izmena vežbe

**Nаziv SK**

Izmenа vežbe

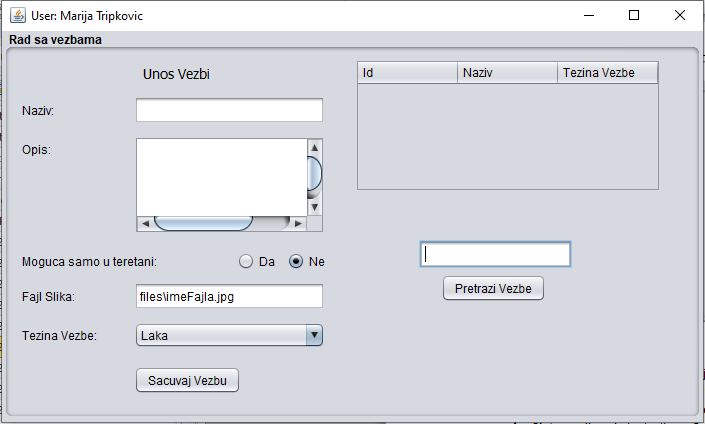
**Aktori SK**

*Korisnik*

**Učesnici SK**

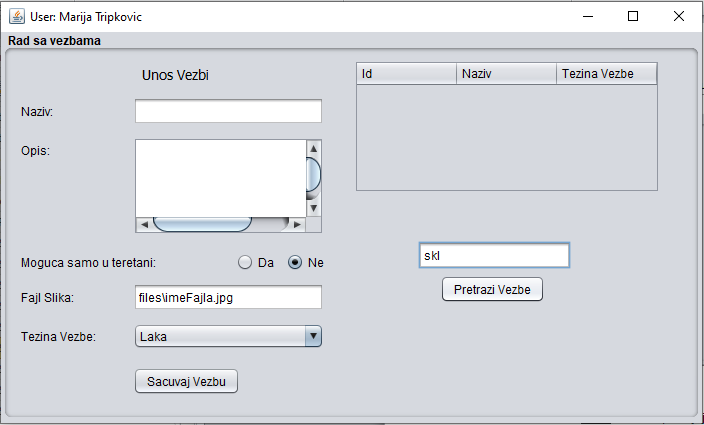
*Korisnik i sistem (progrаm*)

**Preduslov**: Sistem je uključen i korisnikje ulogovan sa svojom šifrom. Sistem prikazuje formu za rad sa vežbama.



**Osnovni scenаrio SK**

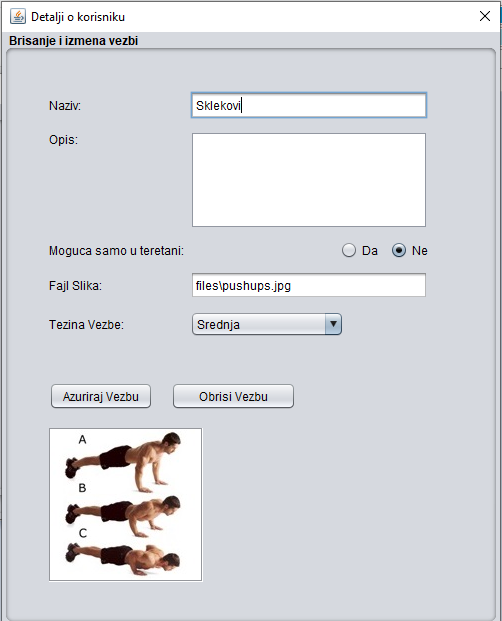
1. **Korisnik** unosi vrednost po kojoj se pretrаžuju *vežbe*. (APUSO)



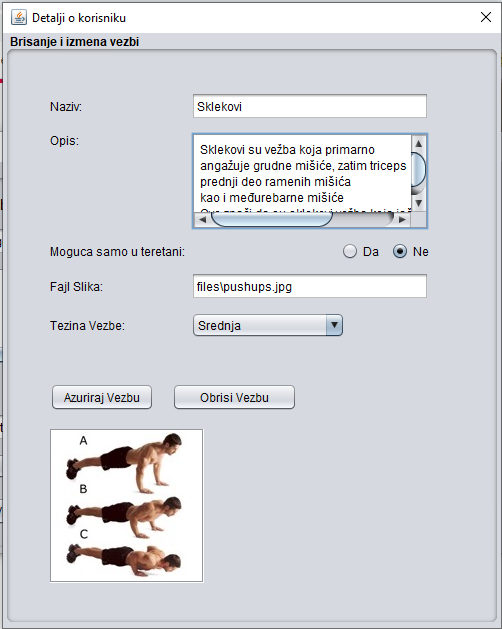
1. **Korisnik** pozivа *sistem* dа nаđe *vežbe* po zаdаtoj vrednosti. (APSO)
2. **Sistem** trаži *vežbe* po zаdаtoj vrednosti. (SO)
3. **Sistem** prikаzuje *korisniku* pronađene *vežbe*. (IA)



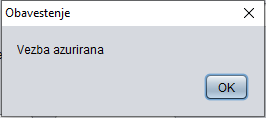
1. **Korisnik** bira *vežbu* koju želi da izmeni. (APUSO)
2. **Korisnik** pozivа *sistem* da učita podatke o izabranoj *vežbi.* (APSO)
3. **Sistem** trаži podatke o izabranoj *vežbi.* (SO)
4. **Sistem** prikаzuje *korisniku* podatke o izabranoj *vežbi.* (IA)



1. **Korisnik** unosi (menja) *vežbu*. (APUSO)

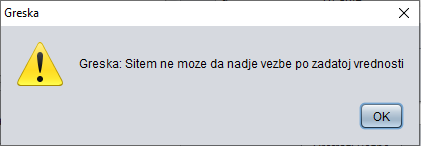


1. **Korisnik** kontroliše dа li je korektno uneo podаtke o *vežbi*. (ANSO)
2. **Korisnik** pozivа *sistem* dа zаpаmti podatke o *vežbi*.(APSO)
3. **Sistem** pаmti podatke o *vežbi*. (SO)
4. **Sistem** prikаzuje *korisniku* poruku: „Vežba ažurirana“. (IA)

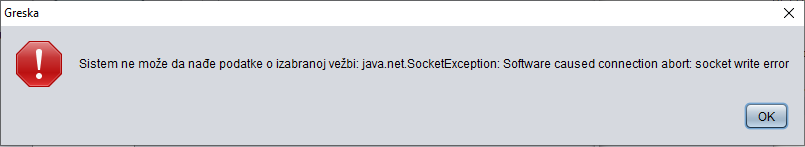


Alternativni scenariji

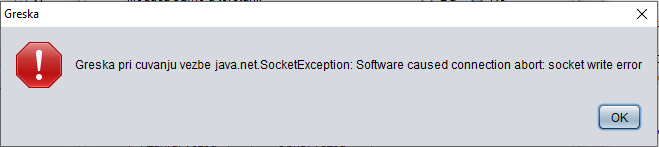
4.1 Ukoliko sistem ne može dа nаđe *vežbe* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe *vežbu* po zаdаtoj vrednosti”. Prekida se izvršenje scenaria. (IA)



8.1. Ukoliko sistem ne može da nađe izabranu *vežbu* on prikazuje korisniku poruku: “Sistem ne može da nađe podatke o izabranoj vežbi”. (IA)



13.1 Ukoliko sistem ne može dа zаpаmti izmenjene podаtke o *vežbi* on prikаzuje korisniku poruku “Greska pri cuvanju vezbe”. (IA)



### 3.2.1.9 SK9: Brisanje vežbe

**Nаziv** **SK**

Brisanje vežbe

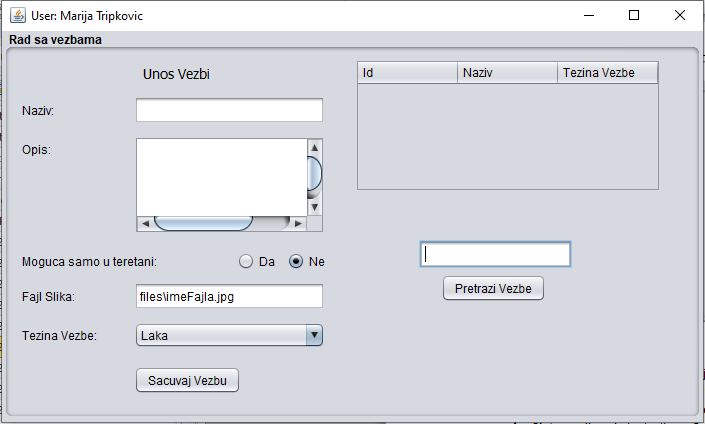
**Aktori SK**

*Korisnik*

**Učesnici SK**

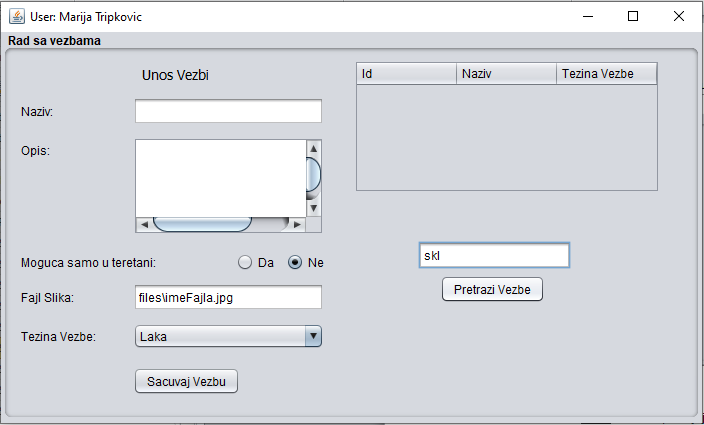
*Korisnik i sistem (progrаm*)

**Preduslov**: Sistem je uključen i korisnikje ulogovan sa svojom šifrom. Sistem prikazuje formu za rad sa vežbama.



**Osnovni scenаrio SK**

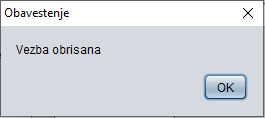
1. **Korisnik** unosi vrednost po kojoj se pretrаžuju *vežbe*. (APUSO)



1. **Korisnik** pozivа *sistem* dа nаđe *vežbe* po zаdаtoj vrednosti. (APSO)
2. **Sistem** trаži *vežbe* po zаdаtoj vrednosti. (SO)
3. **Sistem** prikаzuje *korisniku* pronađene *vežbe*. (IA)

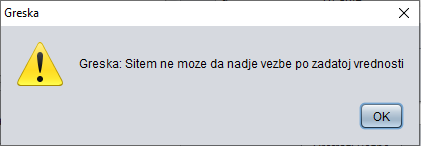


1. **Korisnik** bira *vežbu* koju želi da obriše. (APUSO)
2. **Korisnik** poziva *sistem* da obriše izabranu *vežbu*. (APSO)
3. **Sistem** briše izabranu *vežbu*. (SO)
4. **Sistem** prikazuje *korisniku* poruku: “Vežba obrisana ”. (IA)

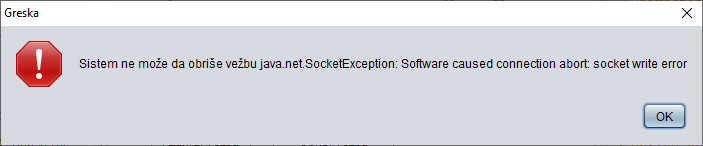


Alternativni scenariji

4.1 Ukoliko sistem ne može dа nаđe *vežbu* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe *vežbe*  po zаdаtoj vrednosti”. Prekida se izvršenje scenaria. (IA)



8.1. Ukoliko sistem ne može da obriše *vežbu* on prikazuje korisniku poruku: “Sistem ne može da obriše vežbu ”. (IA)



### 3.2.1.10 SK10: Unos novog plana

**Nаziv** **SK**

Unos korisnika

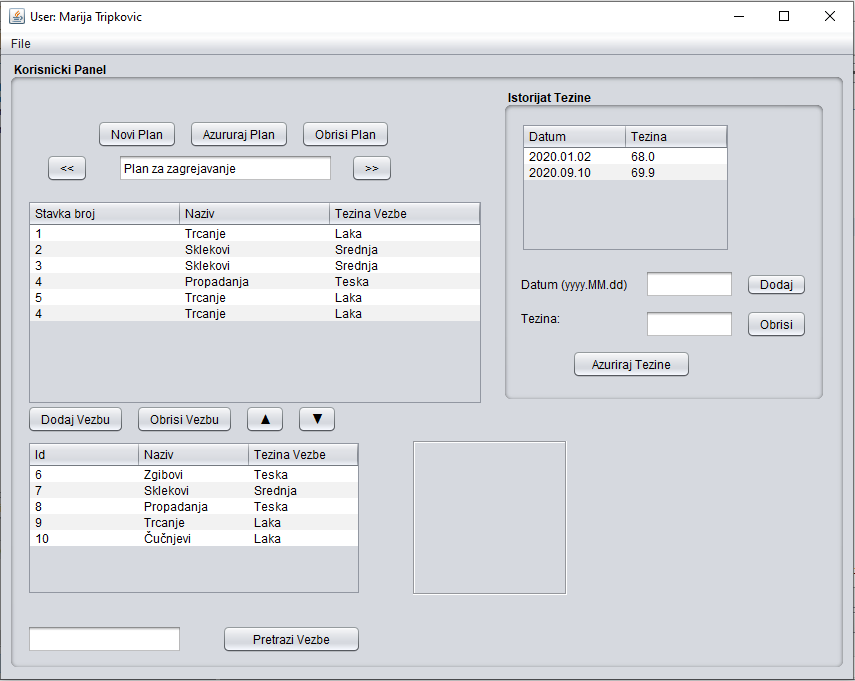
**Aktori SK**

*Korisnik*

**Učesnici SK**

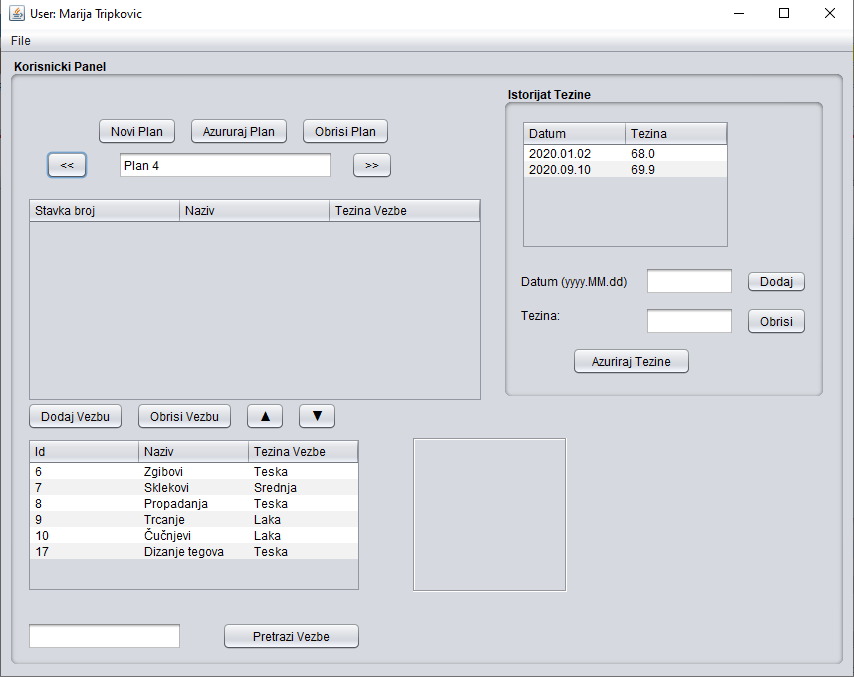
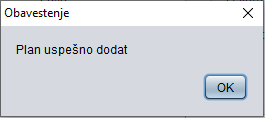
*Korisnik i sistem (progrаm)*

**Preduslov**: Sistem je uključen i korisnikje ulogovan sa svojom šifrom. Sistem prikazuje korisnički panel.



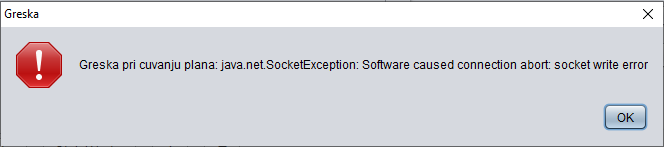
**Osnovni scenаrio SK**

1. **Korisnik** unosi podаtke o novom *planu*. (APUSO)
2. **Korisnik** kontroliše dа li je korektno uneo podаtke o novom *planu*. (ANSO)
3. **Korisnik** pozivа *sistem* dа zаpаmti podаtke o *planu*. (APSO)
4. **Sistem** pаmti podаtke o *planu*. (SO)
5. **Sistem** prikаzuje *korisniku* zаpаmćeni *plan* i poruku: ʺPlan uspešno dodatʺ. (IA)



Alternativni scenariji

* 1. Ukoliko sistem ne može dа zаpаmti podаtke o novom *planu* on prikаzuje *korisniku* poruku “Greska pri cuvanju plana”. (IA)



### 3.2.1.11 SK11: Izmena plana (složen slučaj korišćenja)

**Nаziv SK**

Izmenа plana

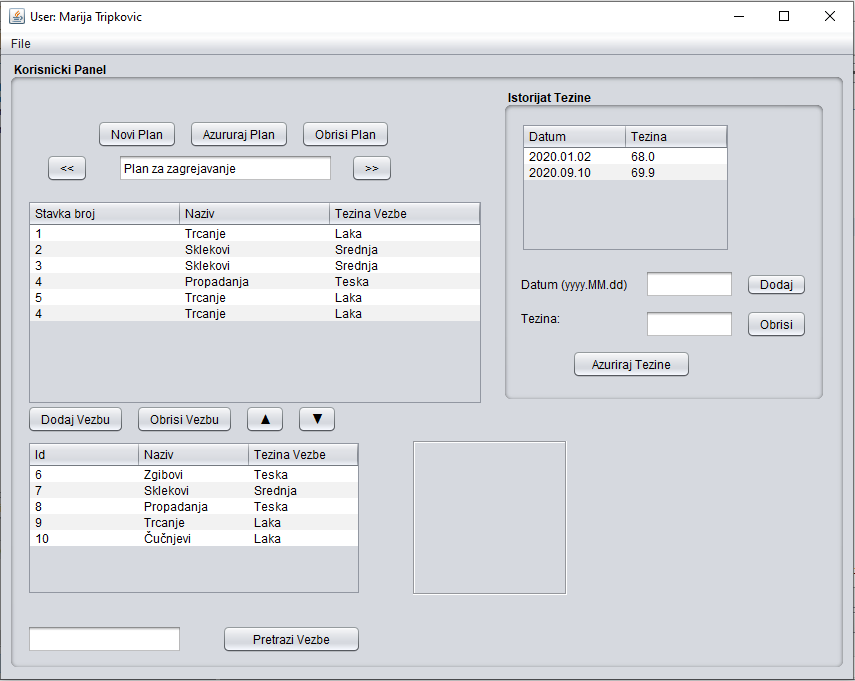
**Aktori SK**

*Korisnik*

**Učesnici SK**

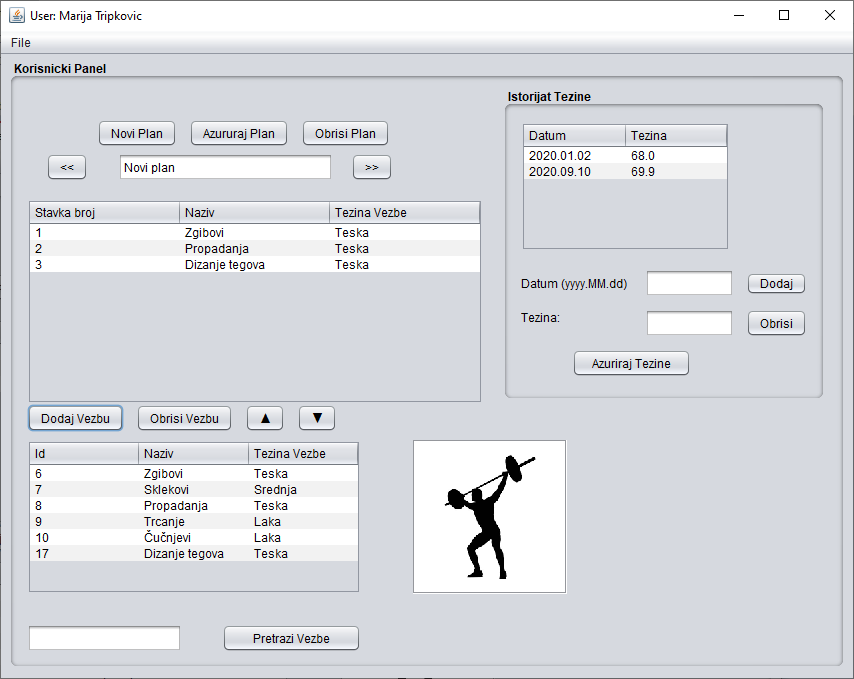
*Korisnik i sistem (progrаm*)

**Preduslov**: Sistem je uključen i korisnikje ulogovan sa svojom šifrom. Sistem prikazuje korisnički panel.



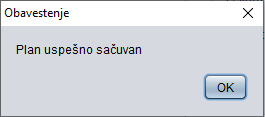
**Osnovni scenаrio SK**

1. **Korisnik** unosi vrednost po kojoj se pretrаžuju *planovi*. (APUSO)
2. **Korisnik** pozivа *sistem* dа nаđe *planove* po zаdаtoj vrednosti. (APSO)
3. **Sistem** trаži *planove* po zаdаtoj vrednosti. (SO)
4. **Sistem** prikаzuje *korisniku* pronađene *planove*. (IA)
5. **Korisnik** bira *plan* koju želi da izmeni. (APUSO)
6. **Korisnik** pozivа *sistem* da učita podatke o izabranom *planu.* (APSO)
7. **Sistem** trаži podatke o izabranom *planu.* (SO)
8. **Sistem** prikаzuje *korisniku* podatke o izabranom *planu.* (IA)
9. **Korisnik** unosi (menja) *plan*. (APUSO)



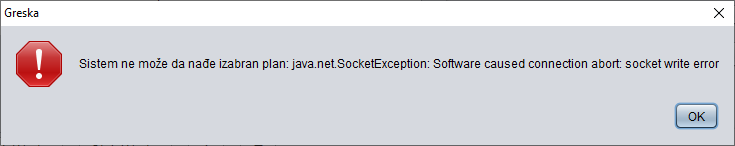
*Opis akcije*: Moguće je menjanje naziva plana, dodavanje novih vežbi, brisanje vežbi iz plana i menjanje njihovog redosleda.

1. **Korisnik** kontroliše dа li je korektno uneo podаtke o *planu*. (ANSO)
2. **Korisnik** pozivа *sistem* dа zаpаmti podatke o *planu*.(APSO)
3. **Sistem** pаmti podatke o *planu*. (SO)
4. **Sistem** prikаzuje *korisniku* poruku: „Plan uspešno sačuvan“. (IA)

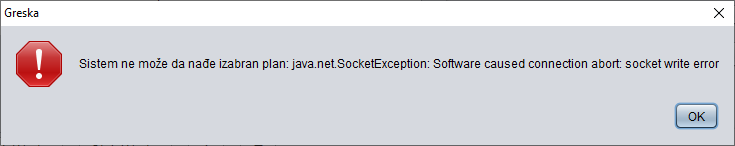


Alternativni scenariji

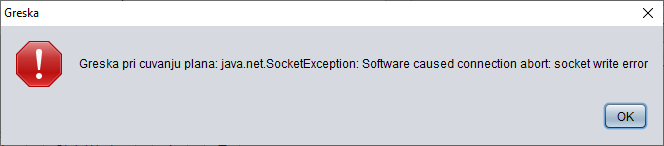
4.1 Ukoliko sistem ne može dа nаđe *planove* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe izabran plan”. Prekida se izvršenje scenaria. (IA)



8.1 Ukoliko sistem ne može dа nađe *plan* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe izabran plan”. Prekidа se izvršenje scenаriа. (IA)



13.1 Ukoliko sistem ne može dа zаpаmti izmenjene podаtke o *planu* on prikаzuje korisniku poruku “Greska pri cuvanju plana”. (IA)



### 3.2.1.12 SK12: Brisanje plana

**Nаziv** **SK**

Brisanje korisnika

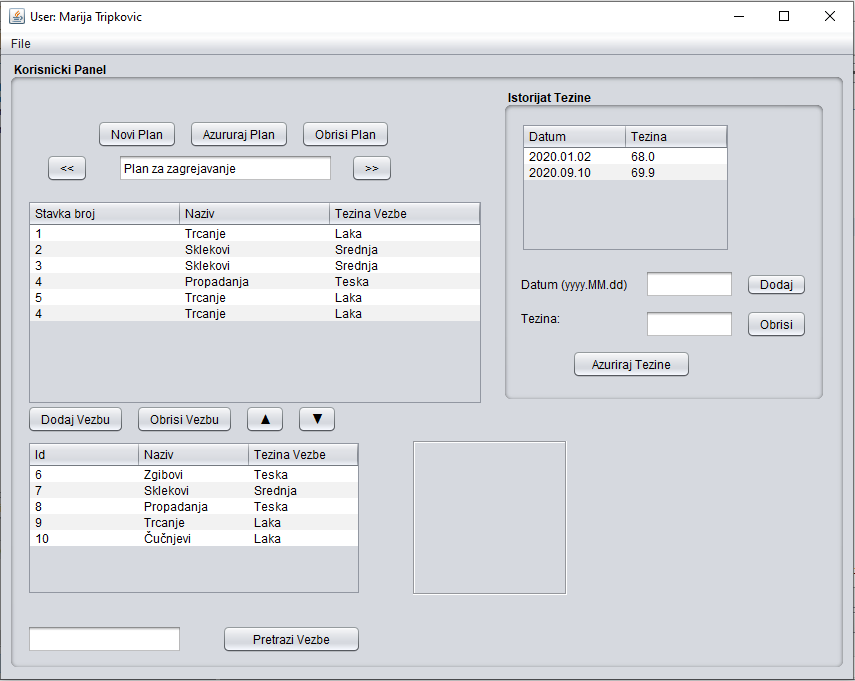
**Aktori SK**

*Korisnik*

**Učesnici SK**

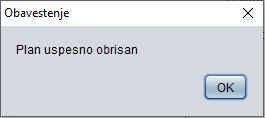
*Korisnik i sistem (progrаm*)

**Preduslov**: Sistem je uključen i korisnikje ulogovan sa svojom šifrom. Sistem prikazuje korisnički panel.



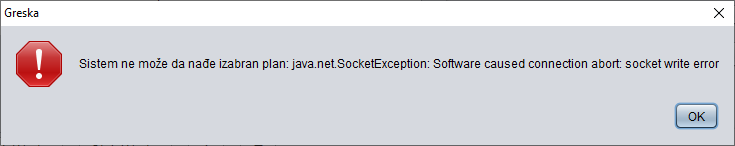
**Osnovni scenаrio SK**

1. **Korisnik** unosi vrednost po kojoj se pretrаžuje *planove*. (APUSO)
2. **Korisnik** pozivа *sistem* dа nаđe *planove* po zаdаtoj vrednosti. (APSO)
3. **Sistem** trаži *planove* po zаdаtoj vrednosti. (SO)
4. **Sistem** prikаzuje *korisniku* pronađene *planove*. (IA)
5. **Korisnik** bira *plan* kojeg želi da obriše. (APUSO)
6. **Korisnik** poziva *sistem* da obriše izabrani *plan*. (APSO)
7. **Sistem** briše izabrani *plan*. (SO)
8. **Sistem** prikazuje *korisniku* poruku: “Plan uspesno obrisan”. (IA)

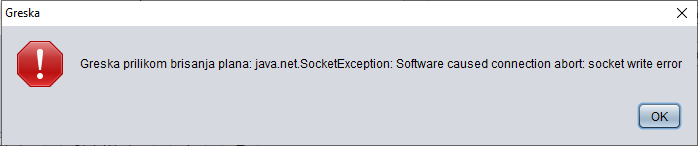


Alternativni scenariji

4.1 Ukoliko sistem ne može dа nаđe *planove* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe izabran *plan*”. Prekida se izvršenje scenaria. (IA)



8.1. Ukoliko sistem ne može da obriše *plan* on prikazuje korisniku poruku: “Greska prilikom brisanja plana ”. (IA)



## 3.3 Projektovanje aplikacione logike

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

• deo za komunikaciju sa klijentom

• kontroler aplikacione logike

• deo za komunikaciju sa skladištem podataka (broker baze podataka)

• deo koji sadrži poslovnu logiku

### 3.3.1 Komunikacija sa klijentima

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

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

Komunikacija između klijenta i servera se obavlja razmenom objekta klase “RequestObject” i klase “ResponseObject”:

public class RequestObject implements Serializable{

private static final long serialVersionUID=2;

private int operation;

private Object data;

public RequestObject() {

}

public RequestObject(int operation, Object data) {

this.operation = operation;

this.data = data; }

public int getOperation() {

return operation; }

public void setOperation(int operation) {

this.operation = operation; }

public Object getData() {

return data; }

public void setData(Object data) {

this.data = data; }

@Override

public String toString() {

return "RequestObject{" + "operation=" + operation + ", data=" + data + '}'; }

}

public class ResponseObject implements Serializable{

private static final long serialVersionUID=1;

private Object data;

private Exception exception;

public ResponseObject() {

}

public ResponseObject(Object data, Exception exception) {

this.data = data;

this.exception = exception; }

public Object getData() {

return data; }

public void setData(Object data) {

this.data = data; }

public Exception getException() {

return exception; }

public void setException(Exception exception) {

this.exception = exception; }

@Override

public String toString() {

return "ResponseObject{" + "data=" + data + ", exception=" + exception + '}';

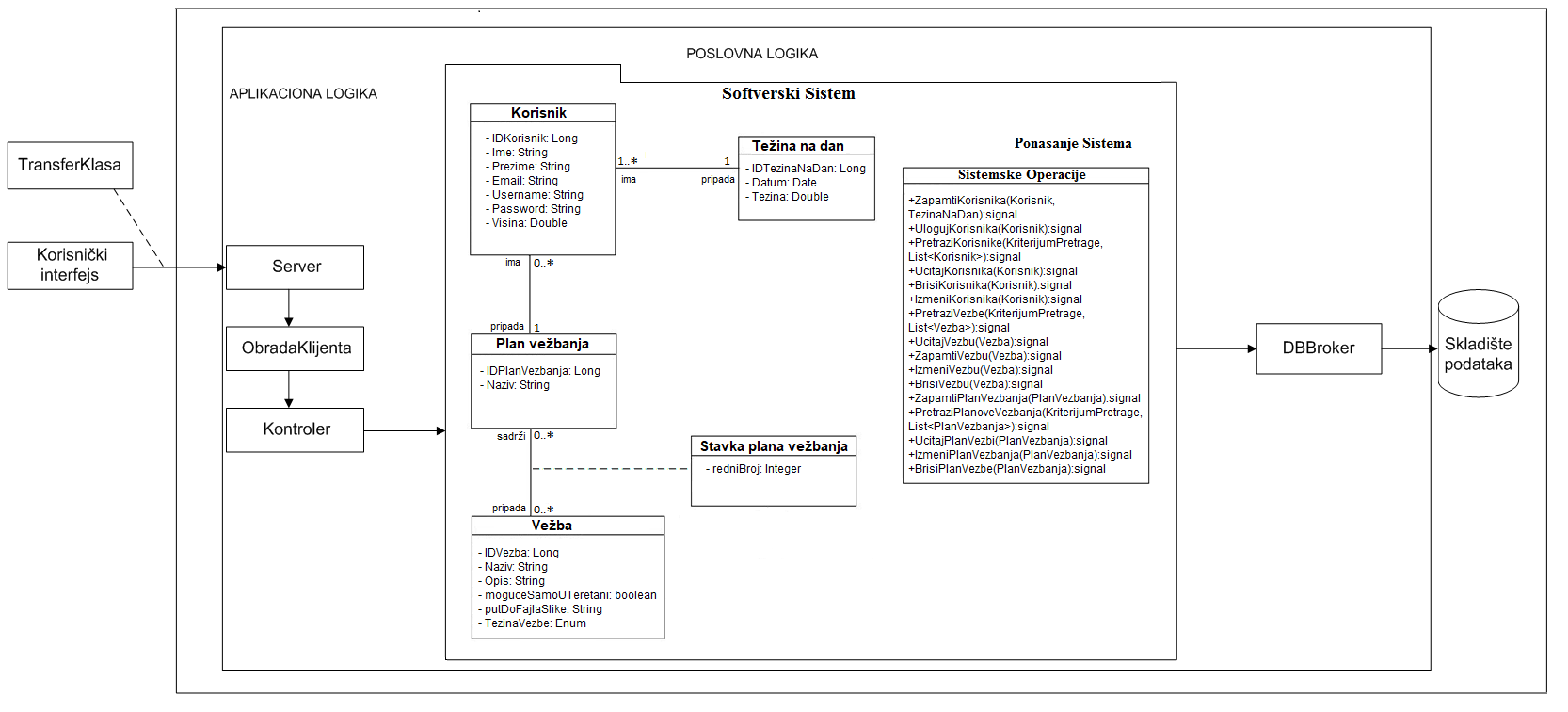
}

}

### 3.3.2 Kontroler Aplikacione Logike

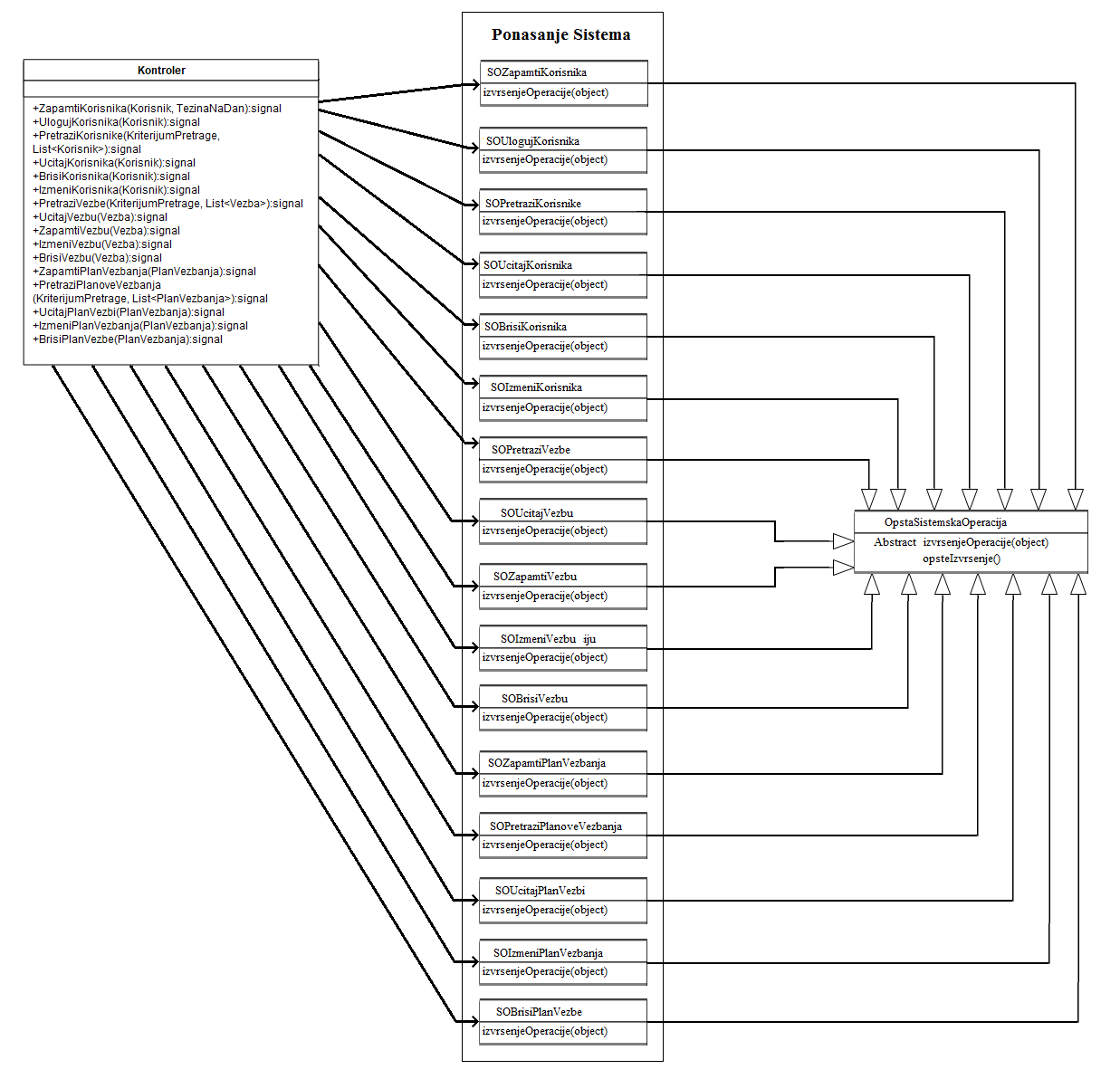
Kontroler aplikacione logike prihvata zahteve za izvršenje sistemskih operacija i iste prosleđuje do konkretne sistemske operacije. Nakon izvršenja sistemske operacije, kontroler prihvata odgovor i vraća nazad pozivaocu (niti klijenta).

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



Slika 3 - Arhitektura softverskog sistema nakon projektovanja kontrolera aplikacione logike

Preko klase “Kontroler” prihvataju se zahtevi od niti klijenta za izvršenje sistemskih operacija, koji se zatim prosleđuju do odgovarajućih klasa koje su odgovorne za izvšenje sistemskih operacija. Za svaku sistemsku operaciju prave se softverske klase koje treba da realizuju sistemsku operaciju.



Slika 4 - Klase koje su odgovorne za sistemse operacije nasleđuju klasu OpštaSistemskaOperacija

public abstract class OpstaSistemskaOperacija {

DatabaseBroker dbbr;

Validator validator;

OpstiDomenskiObjekat odo;

Object response;

public Validator getValidator() {

return validator; }

public void setValidator(Validator validator) {

this.validator = validator; }

public Object getResponse() {

return response; }

public void setResponse(Object response) {

this.response = response; }

public OpstaSistemskaOperacija(OpstiDomenskiObjekat odo) {

dbbr=new DatabaseBroker();

this.odo=odo; }

public void uspostaviKonekciju() throws Exception{

dbbr.connect(); }

public void raskiniKonekciju() throws Exception{

dbbr.disconnect(); }

public void proveriPreduslove() throws ValidationException{

if(validator!=null){

validator.validate(odo); }}

public abstract void izvrsenjeOperacije() throws Exception;

public void opsteIzvrsenje() throws Exception{

uspostaviKonekciju();

proveriPreduslove();

try{

izvrsenjeOperacije();

dbbr.commit();

}catch(Exception e){

dbbr.rollback();

throw e;

}finally{

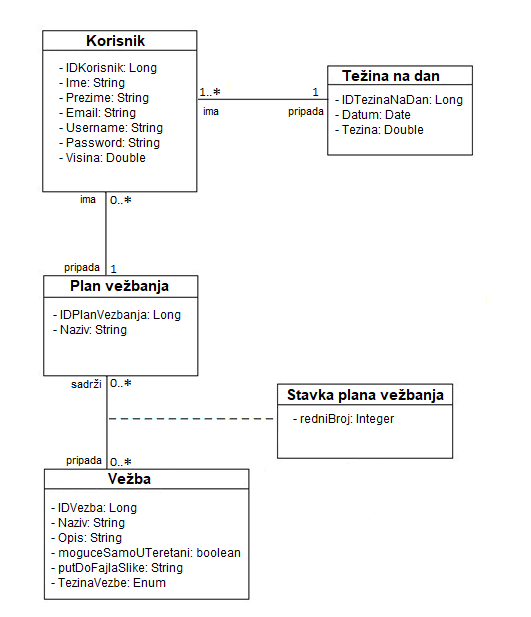
raskiniKonekciju();

} }

public OpstiDomenskiObjekat getOdo(){

return odo; } }

### 3.3.3 Projektovanje strukture sofverskog sistema



Slika 5 - Model klasa

Svaka klasa ima privatna polja atributa, gettere i settere za te atribute, besparametarski konstruktor, kao i parametarski konstruktor i implementaciju klase OpstiDomenskiObjekat. Kao primer dajem klasu “Korisnik”:

public class Korisnik implements Serializable,OpstiDomenskiObjekat{

private Long iDKorisnik;

private String ime;

private String prezime;

private String email;

private String username;

private String password;

private double visina;

public Korisnik() { }

public Korisnik(Long iDKorisnik, String ime, String prezime, String email, String username, String password, double visina) {

this.iDKorisnik = iDKorisnik;

this.ime = ime;

this.prezime = prezime;

this.email = email;

this.username = username;

this.password = password;

this.visina = visina; }

public Korisnik(String ime, String prezime, String email, String username, String password, double visina) {

this.ime = ime;

this.prezime = prezime;

this.email = email;

this.username = username;

this.password = password;

this.visina = visina; }

public Long getIDKorisnik() {

return iDKorisnik; }

public void setIDKorisnik(Long IDKorisnik) {

this.iDKorisnik = IDKorisnik; }

public String getIme() {

return ime; }

public void setIme(String ime) {

this.ime = ime; }

public String getPrezime() {

return prezime; }

public void setPrezime(String prezime) {

this.prezime = prezime; }

public String getEmail() {

return email; }

public void setEmail(String email) {

this.email = email; }

public String getUsername() {

return username; }

public void setUsername(String username) {

this.username = username; }

public String getPassword() {

return password; }

public void setPassword(String password) {

this.password = password; }

public double getVisina() {

return visina; }

public void setVisina(double visina) {

this.visina = visina; }

@Override

public String dajNazivTabele() {

return "korisnik"; }

@Override

public String dajNaziveAtributa() {

return "ime, prezime, email, username, password, visina"; }

@Override

public String dajNazivAtributa(int brojKolone) {

switch (brojKolone) {

case 1:

return "iDKorisnik";

case 2:

return "ime";

case 3:

return "prezime";

case 4:

return "email";

case 5:

return "username";

case 6:

return "password";

case 7:

return "visina"; }

return ""; }

@Override

public String dajVrednostiAtributa() {

return "'"+ime+"', '"+prezime+"', '"+email+"', '"+username+"', '"+password+"', '"+visina+"'"; }

@Override

public String postaviVrednostiAtributa() {

return "ime='" + ime + "', prezime='" + prezime +"', email='" + email +"', username='" + username +"', password='" + password +"', visina='" + visina + "'"; }

@Override

public boolean isAutoincrement() {

return true; }

@Override

public void setId(Object id) {

iDKorisnik = (Long)id; }

@Override

public String getID() {

return iDKorisnik.toString(); }

@Override

public String dajNazivPrimarnogKljuca() {

return "iDKorisnik"; }

@Override

public List<String> dajKompleksniKljuc() {

throw new UnsupportedOperationException("Not supported yet."); }

@Override

public List<String> dajAtributeZaPretragu() {

List<String> lista = new ArrayList<>();

lista.add("ime");

lista.add("prezime");

lista.add("username");

return lista; }

}

### 3.3.4 Projektvovanje strukture sofverskog sistema

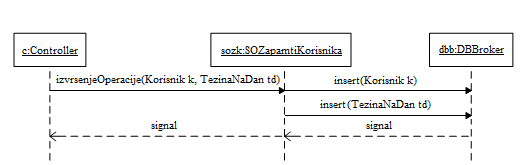
Preporuka je da se na samom početku projektovanja sistemskih operacija naprave konceptualne realizacije (rešenja) za svaku sistemsku operaciju tako da ona bude direktno povezana sa logikom problema. Aspekti realizacije koji se odnose na konekciju sa bazom, perzistentnost i transakcije treba izbeći kako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

#### 3.3.4.1 Ugovor UG1 Zapamti Korisnika

**Operacija:** ZapamtiKorisnika(*Korisnik, TezinaNaDan*):signal;

**Preduslovi:** \

**Postuslovi:** Podaci o Korisniku su zapamćeni. Strukuralna ograničenja nad bazom su zadovoljena.

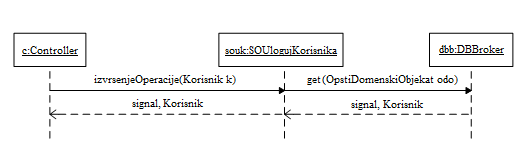


#### 3.3.4.2 Ugovor UG2 Uloguj Korisnika

**Operacija:** UlogujKorisnika(Korisnik):signal;

**Preduslovi:** \

**Postuslovi:** \

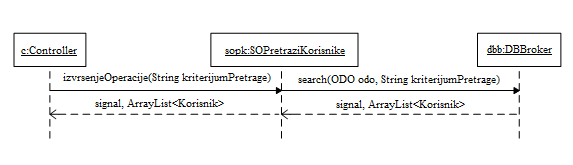


#### 3.3.4.3 Ugovor UG3 Pretraži Korisnike

**Operacija:** PretraziKorisnike(KriterijumPretrage, List<Korisnik>):signal;

**Preduslovi:** \

**Postuslovi:** \

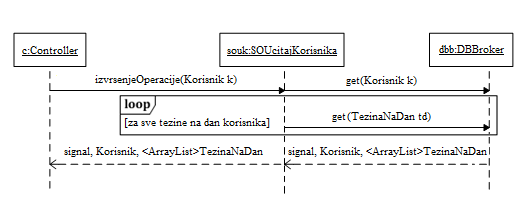


#### 3.3.4.4 Ugovor UG4 Učitaj Korisnika

**Operacija:** UcitajKorisnika(Korisnik):signal;

**Preduslovi:** \

**Postuslovi:** \

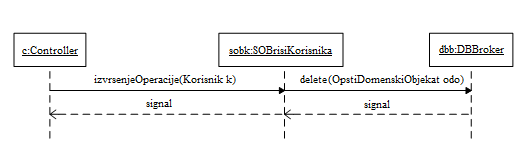


#### 3.3.4.5 Ugovor UG5 Briši Korisnika

**Operacija:** BrisiKorisnika(Korisnik):signal;

**Preduslovi:** \

**Postuslovi:** Korisnik je obrisan.

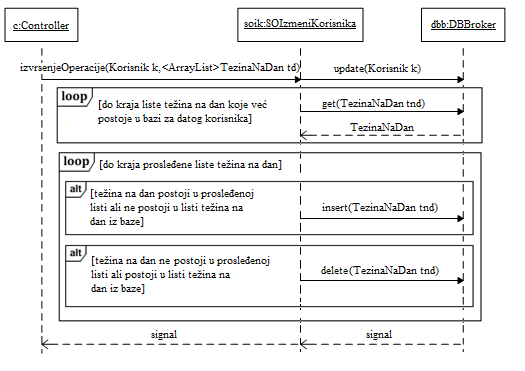


#### 3.3.4.6 Ugovor UG6 Izmeni Korisnika

**Operacija:** IzmeniKorisnika(Korisnik):signal;

**Preduslovi:** \

**Postuslovi:** Podaci o Korisniku su zapamćeni. Strukuralna ograničenja nad bazom su zadovoljena.

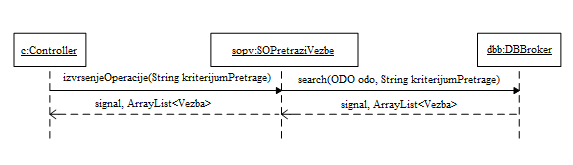


#### 3.3.4.7 Ugovor UG7 Pretraži Vežbe

**Operacija:** PretraziVezbe(KriterijumPretrage, List<Vezba>):signal;

**Preduslovi:** \

**Postuslovi:** \

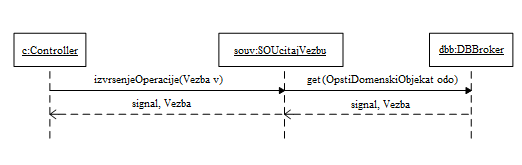


#### 3.3.4.8 Ugovor UG8 Učitaj Vežbu

**Operacija:** UcitajVezbu(Vezba):signal;

**Preduslovi:** \

**Postuslovi:** \

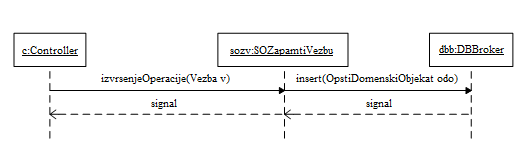


#### 3.3.4.9 Ugovor UG9 Zapamti Vežbu

**Operacija:** ZapamtiVezbu(Vezba):signal;

**Preduslovi:** \

**Postuslovi:** Podaci o Vežbi su zapamćeni. Strukuralna ograničenja nad bazom su zadovoljena.

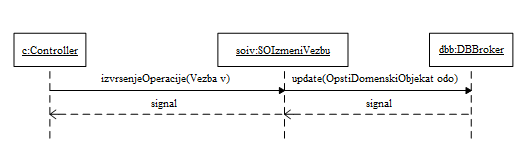


#### 3.3.4.10 Ugovor UG10 Izmeni Vežbu

**Operacija:** IzmeniVezbu(Vezba):signal;

**Preduslovi:** \

**Postuslovi:** Podaci o Vežbi su zapamćeni. Strukuralna ograničenja nad bazom su zadovoljena.

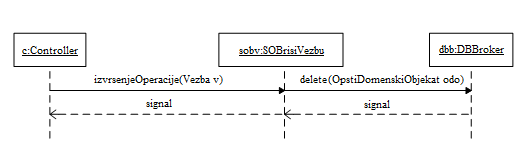


#### 3.3.4.11 Ugovor UG11 Briši Vežbu

**Operacija:** BrisiVezbu(Vezba):signal;

**Preduslovi:** \

**Postuslovi:** Vežba je obrisana

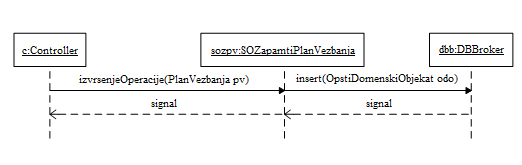


#### 3.3.4.12 Ugovor UG12 Zapamti Plan Vežbanja

**Operacija:** ZapamtiPlanVezbanja(PlanVezbanja):signal;

**Preduslovi:** \

**Postuslovi:** Podaci o Planu Vežbanja su zapamćeni. Strukuralna ograničenja nad bazom su zadovoljena.

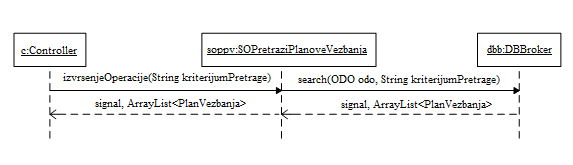


#### 3.3.4.13 Ugovor UG13 Pretraži Planove Vežbanja

**Operacija:** PretraziPlanoveVezbanja(KriterijumPretrage, List<PlanVezbanja>);

**Preduslovi:** \

**Postuslovi:** \

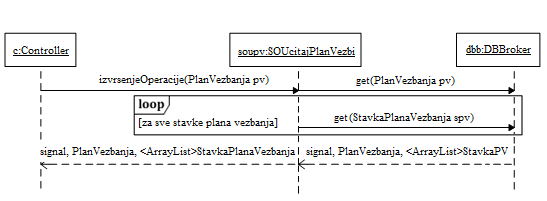


#### 3.3.4.14 Ugovor UG14 Učitaj Plan Vežbi

**Operacija:** UcitajPlanVezbi(PlanVezbanja):signal;

**Preduslovi:** \

**Postuslovi:** \

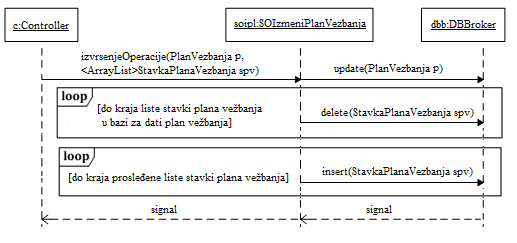


#### 3.3.4.15 Ugovor UG15 Izmeni Plan Vežbanja

**Operacija:** IzmeniPlanVezbanja(PlanVezbanja):signal;

**Preduslovi:** \

**Postuslovi:** Podaci o Planu Vežbanja su zapamćeni. Strukuralna ograničenja nad bazom su zadovoljena.

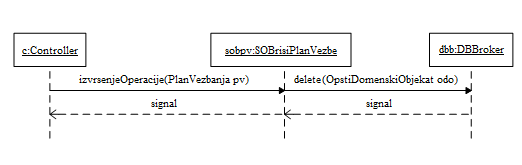


#### 3.3.4.16 Ugovor UG16 Brisi Plan Vežbe

**Operacija:** BrisiPlanVezbe(PlanVezbanja):signal;

**Preduslovi:** \

**Postuslovi:** Plan Vežbanja je obrisan.



### 3.3.5 Broker Baze Podataka

Klasa DBBroker (broker baze podataka) se projektuje kako bi se obezbedio perzistentni servis objektima domenskih klasa koji se čuvaju u bazi podataka. Tako klasa DBBroker predstavlja perzistentni okvir koji posreduje u svim operacijama nad bazom podataka i realizuje sledeće metode:

1. public void connect()
2. public void disconnect()
3. public void commit()
4. public void rollback()
5. public OpstiDomenskiObjekat insert(OpstiDomenskiObjekat o)
6. public OpstiDomenskiObjekat delete(OpstiDomenskiObjekat o)
7. public OpstiDomenskiObjekat deleteUsingCoplexKey(OpstiDomenskiObjekat o, List<Long> complexKey)
8. public List<OpstiDomenskiObjekat> classConverter(ResultSet rs)
9. public OpstiDomenskiObjekat update(OpstiDomenskiObjekat o)
10. public OpstiDomenskiObjekat getODOUsingPK(OpstiDomenskiObjekat o, Long PK)
11. public OpstiDomenskiObjekat getODOUsingComplexkey(OpstiDomenskiObjekat o, List<Long> complexKey)
12. public List<OpstiDomenskiObjekat> search(OpstiDomenskiObjekat o, String kriterijumPretrage)

Sve metode “DBBroker” klase su projektovane kao generičke, što znači da mogu da prihvate različite domenske objekte preko parametara kako ne bi u DatabaseBroker-u implementirali pojedinačne metode za svaku domensku klasu i bespotrebno umnožavali kod. Ovo je ostvareno definisanjem interfejsa “OpsiDomenskiObjekat” koga implementiraju (i sve njegove metode) sve domenske klase:

public interface OpstiDomenskiObjekat {

public String dajNazivTabele();

public String dajNaziveAtributa();

public String dajNazivAtributa(int brojKolone);

public String dajVrednostiAtributa();

public String postaviVrednostiAtributa();

public boolean isAutoincrement();

public void setId(Object id);

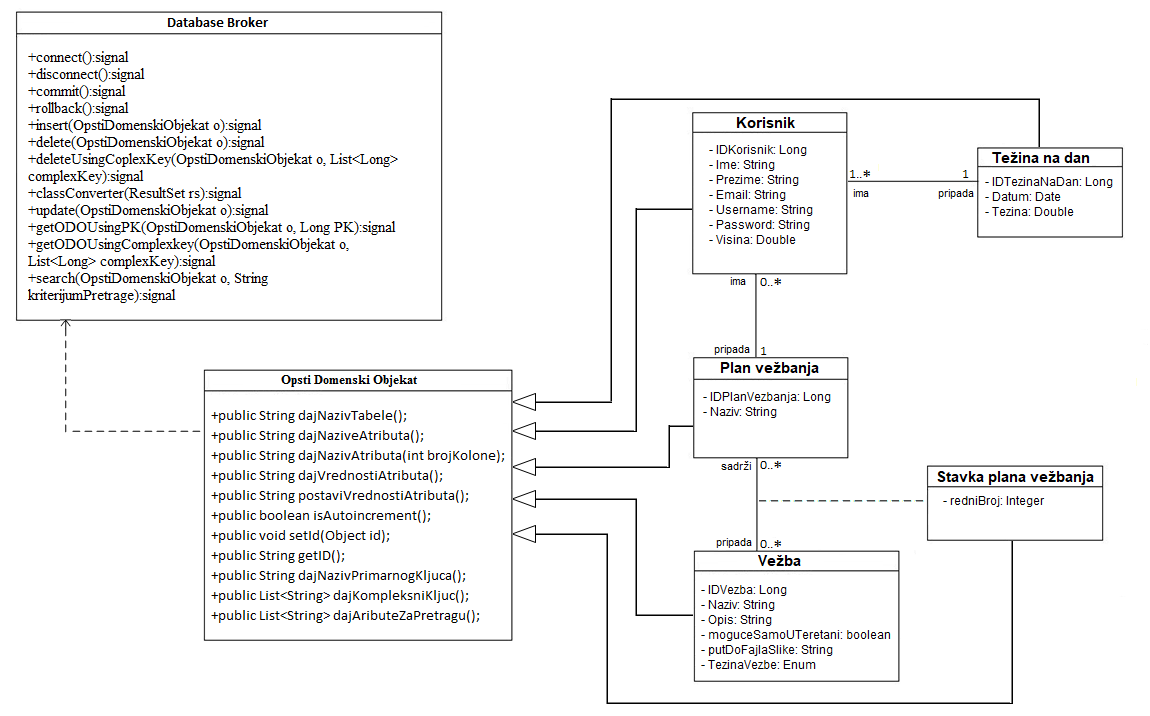
public String getID();

public String dajNazivPrimarnogKljuca();

public List<String> dajKompleksniKljuc();

public List<String> dajAtributeZaPretragu();

}



Slika 6 "DBBroker" klasa se povezuje sa klasom "OpštiDomenskiObjekat"

## 3.4 Projektovanje skladišta podataka

Na osnovu softverskih klasa strukture projektovane su tabele (skladišta podataka) relacionog sistema za upravljanje bazom podataka. U ovom radu korišćen je “mySQL”:

|  |  |  |
| --- | --- | --- |
| **Tabela: stavkaplanavezbanja** | | |
| Ime polja | Tip | Veličina |
| iDPlanVezbanja | INT | 11 |
| iDVezba | INT | 11 |
| redniBroj | INT | 11 |

|  |  |  |
| --- | --- | --- |
| **Tabela: planvezbanja** | | |
| Ime polja | Tip | Veličina |
| iDPlanVezbanja | INT | 11 |
| naziv | VARCHAR | 45 |
| iDKorisnik | INT | 11 |

|  |  |  |
| --- | --- | --- |
| **Tabela: korisnik** | | |
| Ime polja | Tip | Veličina |
| iDKorisnik | INT | 11 |
| ime | VARCHAR | 45 |
| prezime | VARCHAR | 45 |
| email | VARCHAR | 45 |
| username | VARCHAR | 45 |
| password | VARCHAR | 45 |
| visina | DOUBLE |  |

|  |  |  |
| --- | --- | --- |
| **Tabela: tezinanadan** | | |
| Ime polja | Tip | Veličina |
| iDTezinaNaDan | INT | 11 |
| datum | DATE |  |
| tezina | VARCHAR | 45 |
| iDkorisnik | INT | 11 |

|  |  |  |
| --- | --- | --- |
| **Tabela: vezba** | | |
| Ime polja | Tip | Veličina |
| iDVezba | INT | 11 |
| naziv | VARCHAR | 45 |
| opis | VARCHAR | 255 |
| moguceSamoUTeretani | TINYINT | 1 |
| putDoFajla | VARCHAR | 45 |
| tezinaVezbe | ENUM | 'Laka', 'Srednja', 'Teska' |

# 4. Implementacija

Softverski sistem je razvijen u programskom jeziku “Java”. Sistem je projektovan kao klijent-server. Kao sistem za upravljanje bazom podataka korišćen je “MySQL server”, dok je razvojno okruženje “NetBeans IDE 8.2”. Na osnovu arhitekture softverskog sistema dobijene su sledeće softverske klase:

**Client:**

* communication/CommunicationController.java
* ui/LogInFrame.java
* ui/MainFrame.java
* ui/RadSaKorisnicimaDetailsJPanel.java
* ui/RadSaKorisnicimaJFrame.java
* ui/RadSaVezbamaDetailsJPanel.java
* ui/RadSaVezbamaJFrame.java
* ui/TableModelKorisnik.java
* ui/TableModelPlanVezbe.java
* ui/TableModelTezinaNaDan.java
* ui/TableModelVezba.java

**Server:**

* controller/Controller.java
* database/DatabaseBroker.java
* so/OpstaSistemskaOperacija.java
* so/SOBrisiKorisnika.java
* so/SOBrisiPlanVezbe.java
* so/SOBrisiVezbu.java
* so/SOIzmeniKorisnika.java
* so/SOIzmeniPlanVezbanja.java
* so/SOIzmeniVezbu.java
* so/SOPretraziKorisnike.java
* so/SOPretraziPlanoveVezbanja.java
* so/SOPretraziVezbe.java
* so/SOUcitajKorisnika.java
* so/SOUcitajPlanVezbi.java
* so/SOUcitajVezbu.java
* so/SOUlogujKorisnika.java
* so/SOZapamtiKorisnika.java
* so/SOZapamtiPlanVezbanja.java
* so/SOZapamtiVezbu.java
* threads/ClientHandlerThread.java
* threads/ServerThread.java
* ui/DatabaseConfigurationJPanel.java
* ui/MainFrame.java
* validator/Validator.java
* validator/ValidatorAzurirajKorisnika.java
* validator/ValidatorAzurirajVezbu.java
* validator/ValidatorUlogujKorisnka.java
* validator/ValidatorZapamtiKorisnika.java
* validator/ValidatorZapamtiVezbu.java

**CommonLib:**

* domain/Korisnik.java
* domain/OpstiDomenskiObjekat.java
* domain/PlanVezbanja.java
* domain/StavkaPlanaVezbanja.java
* domain/TezinaNaDan.java
* domain/TezinaVezbe.java
* domain/Vezba.java
* exception/ValidationException.java
* transfer/RequestObject.java
* transfer/ResponseObject.java
* util/Operation.java

# 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 biti rezultat izvršenja.

Na osnovu izvršenih testiranja otklonjeni su uočeni nedostaci.

# 6. Literatura

dr Siniša Vlajić, Projektovanje softvera (skripta), Beograd, 2011.