Univerzitet u Beogradu

Fakultet organizacionih nauka

Laboratorija za softversko inženjerstvo

Seminarski rad iz predmeta

Projektovanje softvera

Tema: Softverski sistem za kupovinu i prodaju muzičkih kompozicija

|  |  |
| --- | --- |
| Profesor:  dr Siniša Vlajić | Student:  Leon Vilović 252/10/I |

Beograd, 2020.

# Sadržaj

[Sadržaj 2](#_Toc34518477)

[1. Korisnički Zahtevi 4](#_Toc34518478)

[1.1 Verbalni Opis 4](#_Toc34518479)

[1.2 UML Slučajeva korišćenja 4](#_Toc34518480)

[1.3 Legenda akcija 5](#_Toc34518481)

[1.3.1 SK1: Unos nove Kompozicije (složen slucaj) 5](#_Toc34518482)

[1.3.2 SK2: Pretraživanje kompozicija 6](#_Toc34518483)

[1.3.3 SK3: Brisanje kompozicije 7](#_Toc34518484)

[1.3.4 SK4: Izmena kompozicije 8](#_Toc34518485)

[1.3.5 SK5: Preslušavanje kompozicije 9](#_Toc34518486)

[1.3.6 SK6: Pretraživanje korisnika 10](#_Toc34518487)

[1.3.7 SK7: Kupovina kompozicije (složen slucaj) 11](#_Toc34518488)

[1.3.8 SK8: Brisanje korisnika 12](#_Toc34518489)

[1.3.9 SK9: Prijava korisnika 13](#_Toc34518490)

[1.3.10 SK10: Unos korisnika 14](#_Toc34518491)

[2. Analiza 15](#_Toc34518492)

[2.1 Ponašanje softverskog sistema – Sistemski dijagram sekvenci 15](#_Toc34518493)

[2.1.1 DS 1: Unos nove Kompozicije 15](#_Toc34518494)

[2.1.2 DS 2: Pretraživanje kompozicija 16](#_Toc34518495)

[2.1.3 DS 3: Brisanje kompozicija 18](#_Toc34518496)

[2.1.4 DS 4: Izmena kompozicije 20](#_Toc34518497)

[2.1.5 DS 5: Preslušavanje kompozicije 22](#_Toc34518498)

[2.1.6 DS 6: Pretraživanje korisnika 24](#_Toc34518499)

[2.1.7 DS 7: Kupovina kompozicije 26](#_Toc34518500)

[2.1.8 DS 8: Brisanje korisnika 28](#_Toc34518501)

[2.1.9 DS 9: Prijava korisnika 30](#_Toc34518502)

[2.1.10 DS 10: Unos korisnika 31](#_Toc34518503)

[2.2. Ponašanje softverskog sistema – Definisanje ugovora o sistemskim operacijama 33](#_Toc34518504)

[2.3. Struktura softverskog sistema – Konceptualni (domenski) model 35](#_Toc34518505)

[2.4. Struktura softverskog sistema – Relacioni model 36](#_Toc34518506)

[2.5. Struktura softverskog sistema – Tabela Ograničenja 37](#_Toc34518507)

[3.Projektovanje 40](#_Toc34518508)

[3.1 Arhitektura softverskog sistema 40](#_Toc34518509)

[3.2 Projektovanje korisničkog interfejsa 41](#_Toc34518510)

[3.2.1 Projektovanje ekranskih formi 41](#_Toc34518511)

[3.2.1.1 SK1 - Unos nove Kompozicije 43](#_Toc34518512)

[3.2.1.2 SK2 - Pretraživanje kompozicija 45](#_Toc34518513)

[3.2.1.3 SK3: Brisanje kompozicije 48](#_Toc34518514)

[3.2.1.4 SK4: Izmena kompozicije 51](#_Toc34518515)

[3.2.1.5 SK5: Preslušavanje kompozicije 54](#_Toc34518516)

[3.2.1.6 SK6: Pretraživanje korisnika 57](#_Toc34518517)

[3.2.1.7 SK7: Kupovina kompozicije (složen slucaj) 60](#_Toc34518518)

[3.2.1.8 SK8: Brisanje korisnika 63](#_Toc34518519)

[3.2.1.9 SK9: Prijava korisnika 66](#_Toc34518520)

[3.2.1.10 SK10: Unos korisnika 68](#_Toc34518521)

[3.3 Projektovanje aplikacione logike 71](#_Toc34518522)

[3.3.1 Komunikacija sa klientima 71](#_Toc34518523)

[3.3.2 Kontroler Aplikacione Logike 73](#_Toc34518524)

[3.3.3 Projektvovanje strukture sofverskog sistema 76](#_Toc34518525)

# 1. Korisnički Zahtevi

## 1.1 Verbalni Opis

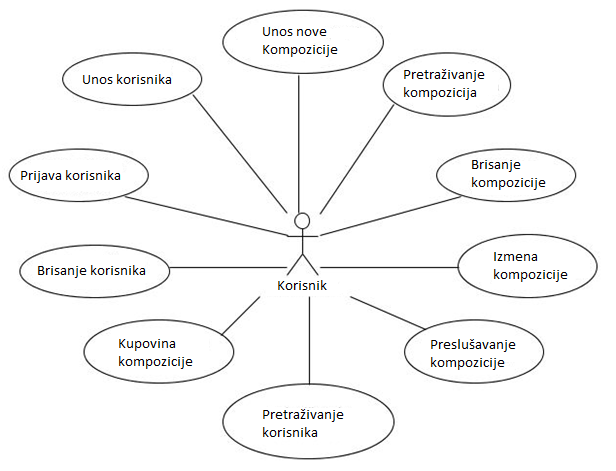
Potrebno je napraviti aplikaciju koja omogućava prijavljivanje korisnika, vođenje evidencije o kompozicijama i kupovini i prodaji kompozicija.

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 kompozicijama 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 - 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 nove Kompozicije (složen slucaj)

**Nаziv SK**

Unos nove kompozicije

**Aktori SK**

*Korisnik*

**Učesnici SK**

*Korisnik i sistem (progrаm*)

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

**Osnovni scenаrio SK**

1. **Korisnik** unosi podаtke o novoj *kompoziciji*. (APUSO)
2. **Korisnik** kontroliše dа li je korektno uneo podаtke i put ka audio fajlu *kompozicije*. (ANSO)
3. **Korisnik** pozivа *sistem* dа zаpаmti podаtke o *kompoziciji*. (APSO)
4. **Sistem** pаmti podаtke o *kompoziciji*. (SO)
5. **Sistem** prikаzuje *kompozitoru* zаpаmćenu *kompoziciju* i poruku: ʺSistem je zаpаmtio kompozicijuʺ. (IA)

Alternаtivnа scenаrijа

5.1 Ukoliko sistem ne može dа zаpаmti podаtke o novoj *kompoziciji* on prikаzuje korisniku poruku “Sistem ne može dа zаpаmti novu kompoziciju”. (IA)

### 1.3.2 SK2: Pretraživanje kompozicija

**Nаziv SK**

Pretrаživаnje kompozicija

**Aktori SK**

*Korisnik*

**Učesnici SK**

*Korisnik i sistem (progrаm*)

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

**Osnovni scenаrio SK**

1. **Korisnik** unosi vrednost po kojoj pretrаžuje *kompozicije*. (APUSO)
2. **Korisnik** pozivа *sistem* dа nаđe *kompozicije* po zаdаtoj vrednosti. (APSO)
3. **Sistem** trаži *kompozicije* po zаdаtoj vrednosti. (SO)
4. **Sistem** prikazuje *korisniku* nađene *kompozicije* i poruku ʺSistem je pronašao kompozicijeʺ. (IA)
5. **Korisnik** bira*kompoziciju* čije podatke želi da vidi. (APUSO)
6. **Korisnik** poziva sistem da prikaže podatke o izabranoj *kompoziciji*. (APSO)
7. **Sistem** traži podatke o izabranoj *kompoziciji*. (SO)
8. **Sistem** prikаzuje *korisniku* podаtke o *kompoziciji* i poruku: ʺOvo su podaci o vašoj kompozicijiʺ. (IA)

Alternаtivnа scenаrijа

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

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

### 1.3.3 SK3: Brisanje kompozicije

**Nаziv** **SK**

Brisanje kompozicija

**Aktori SK**

*Korisnik*

**Učesnici SK**

*Korisnik i sistem (progrаm)*

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

**Osnovni scenаrio SK**

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

Alternаtivnа scenаrijа

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

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

### 1.3.4 SK4: Izmena kompozicije

**Nаziv SK**

Izmenа kompozicije

**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 kompozicijama.

**Osnovni scenаrio SK**

1. **Korisnik** unosi vrednost po kojoj se pretrаžuju *kompozicije*. (APUSO)
2. **Korisnik** pozivа *sistem* dа nаđe *kompozicije* po zаdаtoj vrednosti. (APSO)
3. **Sistem** trаži *kompozicije* po zаdаtoj vrednosti. (SO)
4. **Sistem** prikаzuje *korisniku* pronađene *kompozicije* i poruku ʺSistem je pronašao kompozicijeʺ. (IA)
5. **Korisnik** bira *kompoziciju* koju želi da izmeni. (APUSO)
6. **Korisnik** pozivа *sistem* da učita podatke o izabranoj *kompoziciji.* (APSO)
7. **Sistem** trаži podatke o izabranoj *kompoziciji.* (SO)
8. **Sistem** prikаzuje *korisniku* podatke o izabranoj *kompoziciji.* (IA)
9. **Korisnik** unosi (menja) *kompoziciju*. (APUSO)
10. **Korisnik** kontroliše dа li je korektno uneo podаtke o *kompoziciji*. (ANSO)
11. **Korisnik** pozivа *sistem* dа zаpаmti podatke o *kompoziciji*.(APSO)
12. **Sistem** pаmti podatke o *kompoziciji*. (SO)
13. **Sistem** prikаzuje *korisniku* poruku: ʺSistem je zаpаmtio kompozicijuʺ. (IA)

Alternаtivnа scenаrijа

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

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

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

### 1.3.5 SK5: Preslušavanje kompozicije

**Nаziv SK**

Preslušavanje kompozicija

**Aktori SK**

*Korisnik*

**Učesnici SK**

*Korisnik i sistem (progrаm*)

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

**Osnovni scenаrio SK**

1. **Korisnik** unosi vrednost po kojoj pretrаžuje *kompozicije*. (APUSO)
2. **Korisnik** pozivа *sistem* dа nаđe *kompozicije* po zаdаtoj vrednosti. (APSO)
3. **Sistem** trаži *kompozicije* po zаdаtoj vrednosti. (SO)
4. **Sistem** prikazuje *korisniku* nađene *kompozicije* i poruku ʺSistem je pronašao kompozicijeʺ. (IA)
5. **Korisnik** bira*kompoziciju* koju želi da čuje. (APUSO)
6. **Korisnik** poziva sistem da prikaže podatke o izabranoj *kompoziciji*. (APSO)
7. **Sistem** traži podatke izabrane *kompozicije*. (SO)
8. **Sistem** prikаzuje *korisniku* podаtke o *kompoziciji,* poruku ʺOvo su podaci o vašoj kompozicijiʺ. (IA)
9. **Korisnik** bira*zvučni zapis* koju želi da čuje. (APUSO)
10. **Korisnik** poziva sistem da pusti audio fajl izabrane *kompozicijie*. (APSO)
11. **Sistem** prikаzuje *korisniku* podаtke o *audio fajlu* i poruku ʺAudio fajl je pustenʺ i *audio fajl* se pušta.(IA)

Alternаtivnа scenаrijа

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

8.1 Ukoliko sistem ne može dа nađe *kompoziciju* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe izabranu *kompoziciju*”. (IA)

11.1 Ukoliko sistem ne može dа nađe *zvučni zapis* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe izabranui *zvučni zapis*”. (IA)

### 1.3.6 SK6: Pretraživanje korisnika

**Nаziv SK**

Pretrаživаnje kompozicija

**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* i poruku ʺSistem je pronašao korisnikeʺ. (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* i poruku ʺOvo su podaci o vašem korisnikuʺ. (IA)

Alternаtivnа scenаrijа

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)

### 1.3.7 SK7: Kupovina kompozicije (složen slucaj)

**Nаziv SK**

Kupovina kompozicija

**Aktori SK**

*Korisnik*

**Učesnici SK**

*Korisnik i sistem (progrаm*)

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

**Osnovni scenаrio SK**

1. **Korisnik** unosi vrednost po kojoj pretrаžuje *kompozicije*. (APUSO)
2. **Korisnik** pozivа *sistem* dа nаđe *kompozicije* po zаdаtoj vrednosti. (APSO)
3. **Sistem** trаži *kompozicije* po zаdаtoj vrednosti. (SO)
4. **Sistem** prikazuje *korisniku* nađene *kompozicije* i poruku ʺSistem je pronašao kompozicijeʺ. (IA)
5. **Korisnik** bira*kompoziciju* koju želi da kupi. (APUSO)
6. **Korisnik** poziva sistem da pokrene transakciju. (APSO)
7. **Sistem** traži podatke izabrane *kompozicije*. (SO)
8. **Sistem** prikаzuje *korisniku* podаtke o *kompoziciji,* poruku ʺKupili ste kompozicijuʺ i prepusta mu audio fajl. (IA)

Alternаtivnа scenаrijа

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

8.1 Ukoliko sistem ne može dа pronađe *audio fajl* ponistava se transakcija i prikаzuje se korisniku poruka: “Sistem ne može dа pristupi audio fajlu”. (IA)

### 1.3.8 SK8: 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* i poruku ʺSistem je pronašao korisnikeʺ. (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: “Sistem je obrisao *korisnika* ”. (IA)

Alternаtivnа scenаrijа

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)

### 1.3.9 SK9: 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 rad sa korisnicima.

**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)

Alternаtivnа scenаrijа

4.1. Ukoliko sistem ne može da nađe korisnika prikazuje poruku “Sistem ne može da nađe korisnika na osnovu unetih vrednosti”. (IA)

### 1.3.10 SK10: Unos 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: ʺSistem je zаpаmtio korisnika ʺ. (IA)

Alternаtivnа scenаrijа

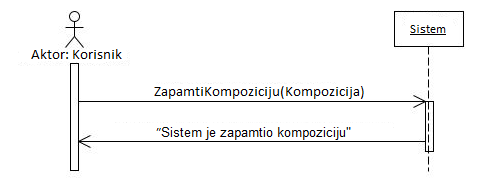
5.1 Ukoliko sistem ne može dа zаpаmti podаtke o novom *korisniku* on prikаzuje *korisniku* poruku “Sistem ne može dа zаpаmti novog korisnika”. (IA)

# 2. Analiza

## 2.1 Ponašanje softverskog sistema – Sistemski dijagram sekvenci

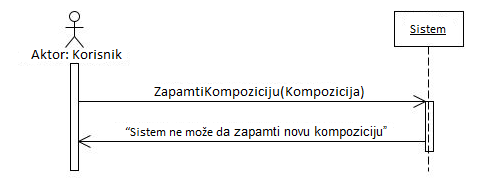
### 2.1.1 DS 1: Unos nove Kompozicije

1. **Korisnik** pozivа *sistem* dа zаpаmti podаtke o *kompoziciji*. (APSO)
2. **Sistem** prikаzuje *kompozitoru* zаpаmćenu *kompoziciju* i poruku: ʺSistem je zаpаmtio kompozicijuʺ. (IA)



Alternаtivnа scenаrijа

2.1 Ukoliko sistem ne može dа zаpаmti podаtke o novoj *kompoziciji* on prikаzuje kompozitoru poruku “Sistem ne može dа zаpаmti novu kompoziciju”. (IA)

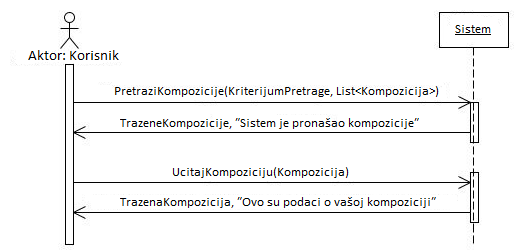


Sa navedenih sekvencnih dijagrama uočava jedna sistemska operacija:

1. signal **ZapamtiKompoziciju**(Kompozicija)

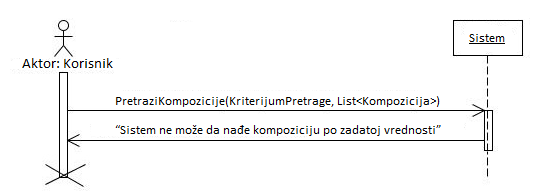
### 2.1.2 DS 2: Pretraživanje kompozicija

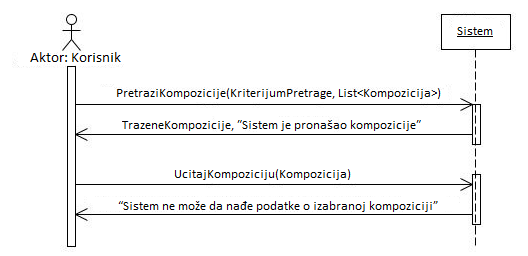
1. **Korisnik** pozivа *sistem* dа nаđe *kompozicije* po zаdаtoj vrednosti. (APSO)
2. **Sistem** prikazuje *korisniku* nađene *kompozicije* i poruku ʺSistem je pronašao kompozicijeʺ. (IA)
3. **Korisnik** poziva sistem da prikaže podatke o izabranoj *kompoziciji*. (APSO)
4. **Sistem** prikаzuje *korisniku* podаtke o *kompoziciji* i poruku: ʺOvo su podaci o vašoj kompozicijiʺ. (IA)



Alternаtivnа scenаrijа

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



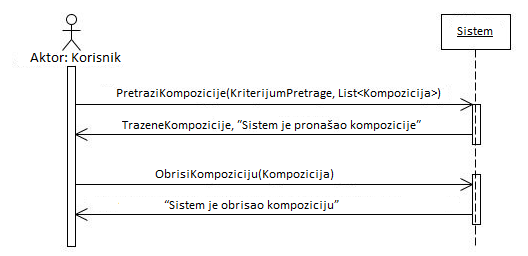
4.1. Ukoliko sistem ne može da nađe izabranu *kompoziciju* on prikazuje korisniku poruku: “Sistem ne može da nađe podatke o izabranoj kompoziciji”. (IA)

Sa navedenih sekvencnih dijagrama uočavaju se sledece sistemske operacije:

1. signal **PretraziKompozicije**(KriterijumPretrage, List<Kompozicija>)
2. signal **UcitajKompoziciju**(Kompozicija)

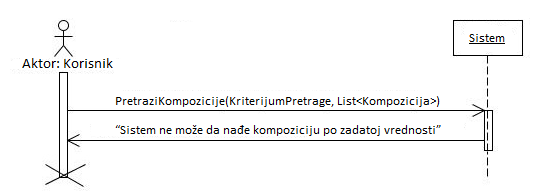
### 2.1.3 DS 3: Brisanje kompozicija

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

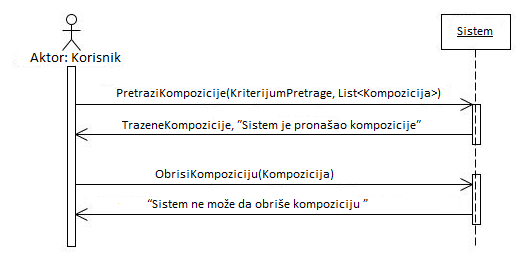


Alternаtivnа scenаrijа

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



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

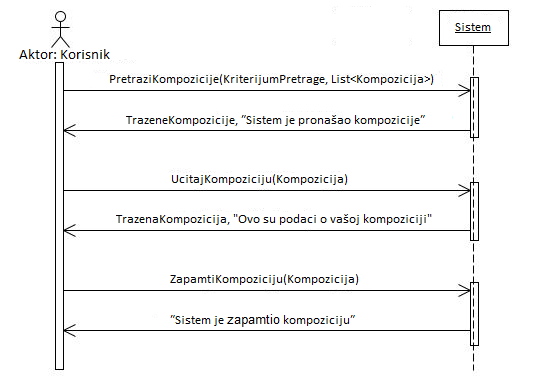


Sa navedenih sekvencnih dijagrama uočavaju se sledece sistemske operacije:

1. signal **PretraziKompozicije**(KriterijumPretrage, List<Kompozicija>)
2. signal **ObrisiKompoziciju**(Kompozicija)

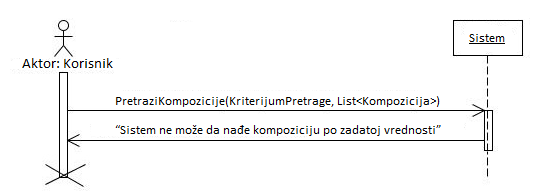
### 2.1.4 DS 4: Izmena kompozicije

1. **Korisnik** pozivа *sistem* dа nаđe *kompozicije* po zаdаtoj vrednosti. (APSO)
2. **Sistem** prikаzuje *korisniku* pronađene *kompozicije* i poruku ʺSistem je pronašao kompozicijeʺ. (IA)
3. **Korisnik** pozivа *sistem* da učita podatke o izabranoj *kompoziciji.* (APSO)
4. **Sistem** prikаzuje *korisniku* podatke o izabranoj *kompoziciji* i poruku: ʺOvo je vaša kompozicijaʺ*.* (IA)
5. **Korisnik** pozivа *sistem* dа zаpаmti podatke o *kompoziciji*.(APSO)
6. **Sistem** prikаzuje *korisniku* poruku: ʺSistem je zаpаmtio kompozicijuʺ. (IA)

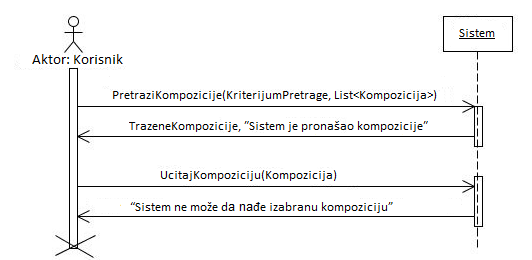


Alternаtivnа scenаrijа

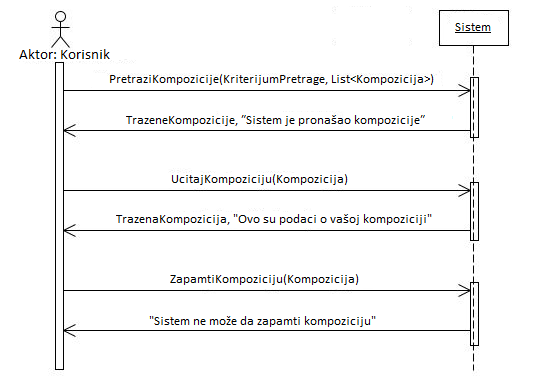
2.1 Ukoliko sistem ne može dа nađe *kompoziciju* on prikаzuje kompozitoru poruku: “Sistem ne može dа nаđe *kompoziciju* po zadatoj vrednosti”. Prekidа se izvršenje scenаriа. (IA)



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



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

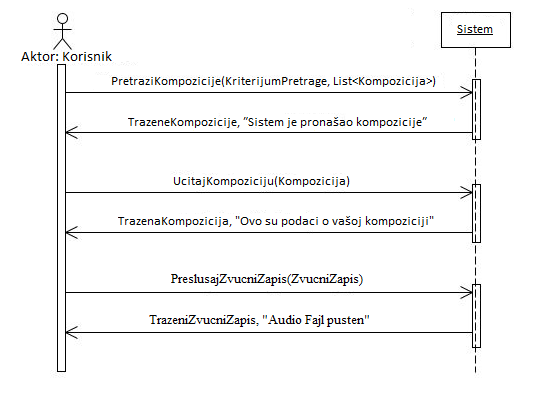


Sa navedenih sekvencnih dijagrama uočavaju se sledece sistemske operacije:

1. signal **PretraziKompozicije**(KriterijumPretrage, List<Kompozicija>)
2. signal **UcitajKompoziciju**(Kompozicija)
3. signal **ZapamtiKompoziciju**(Kompozicija)

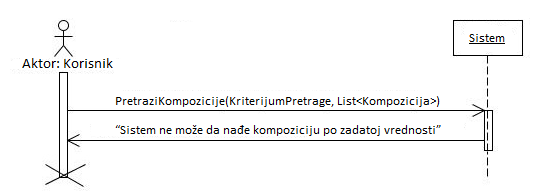
### 2.1.5 DS 5: Preslušavanje kompozicije

1. **Korisnik** pozivа *sistem* dа nаđe *kompozicije* po zаdаtoj vrednosti. (APSO)
2. **Sistem** prikazuje *korisniku* nađene *kompozicije* i poruku ʺSistem je pronašao kompozicijeʺ. (IA)
3. **Korisnik** poziva sistem da prikaže podatke o izabranoj *kompoziciji*. (APSO)
4. **Sistem** prikаzuje *korisniku* podаtke o *kompoziciji,* poruku ʺOvo su podaci o vašoj kompozicijiʺ.(IA)
5. **Korisnik** poziva sistem da pusti audio fajl izabrane *kompozicijie*. (APSO)
6. **Sistem** prikаzuje *korisniku* podаtke o *audio fajlu* i poruku ʺAudio fajl je pustenʺ i *audio fajl* se pušta.(IA)

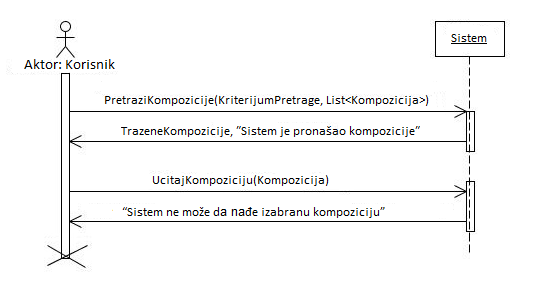


Alternаtivnа scenаrijа

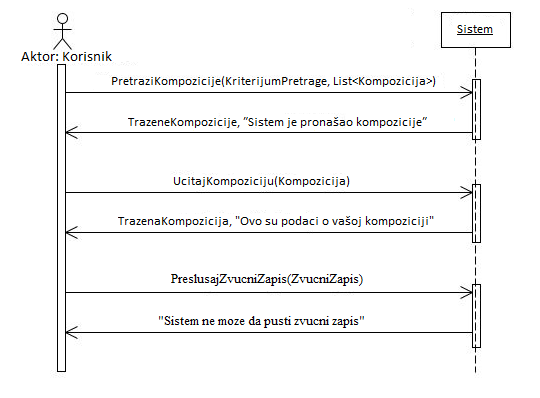
2.1 Ukoliko sistem ne može dа nаđe *kompozicije* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe *kompozicije* po zаdаtoj vrednosti”. Prekida se izvršenje scenaria. (IA)



4.1 Ukoliko sistem ne može dа nađe *kompoziciju* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe izabranu *kompoziciju*”. (IA)



6.1 Ukoliko sistem ne može dа nađe *zvučni zapis* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe izabranui *zvučni zapis*”. (IA)

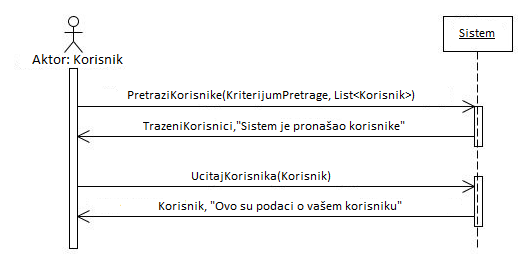


Sa navedenih sekvencnih dijagrama uočavaju se sledece sistemske operacije:

1. signal **PretraziKompozicije**(KriterijumPretrage, List<Kompozicija>)
2. signal **UcitajKompoziciju**(Kompozicija)
3. signal **PreslusajZvucniZapis**(ZvucniZapis)

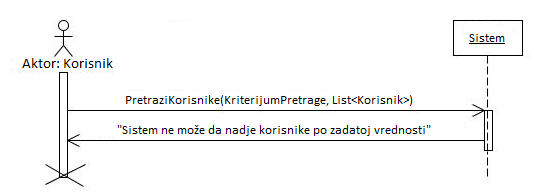
### 2.1.6 DS 6: Pretraživanje korisnika

1. **Korisnik** pozivа *sistem* dа nаđe *korisnike* po zаdаtoj vrednosti. (APSO)
2. **Sistem** prikazuje *korisniku* nađene *korisnike* i poruku ʺSistem je pronašao korisnikeʺ. (IA)
3. **Korisnik** poziva sistem da prikaže podatke o izabranom *korisniku*. (APSO)
4. **Sistem** prikаzuje *korisniku* podаtke o *korisniku* i poruku ʺOvo su podaci o vašem korisnikuʺ. (IA)

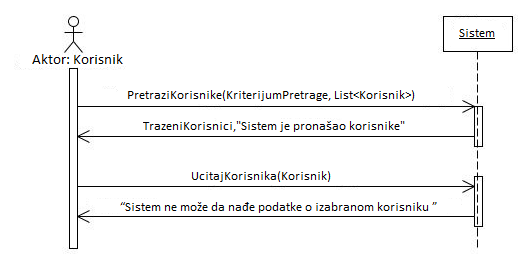


Alternаtivnа scenаrijа

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

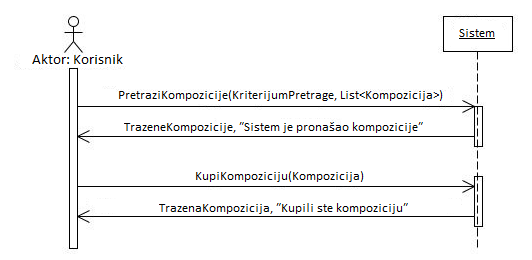


Sa navedenih sekvencnih dijagrama uočavaju se sledece sistemske operacije:

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

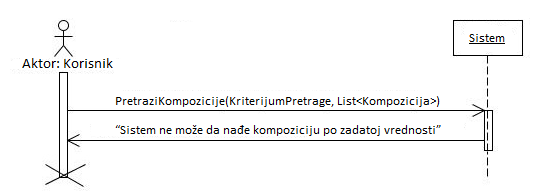
### 2.1.7 DS 7: Kupovina kompozicije

1. **Korisnik** pozivа *sistem* dа nаđe *kompozicije* po zаdаtoj vrednosti. (APSO)
2. **Sistem** prikazuje *korisniku* nađene *kompozicije* i poruku ʺSistem je pronašao kompozicijeʺ. (IA)
3. **Korisnik** poziva sistem da pokrene transakciju. (APSO)
4. **Sistem** prikаzuje *korisniku* podаtke o *kompoziciji,* poruku ʺKupili ste kompozicijuʺ i prepusta mu audio fajl. (IA)

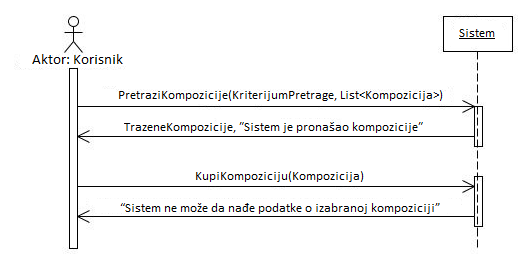


Alternаtivnа scenаrijа

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



4.1 Ukoliko sistem ne može dа pronađe *audio fajl* ponistava se transakcija i prikаzuje se korisniku poruka: “Sistem ne može dа pristupi audio fajlu”. (IA)

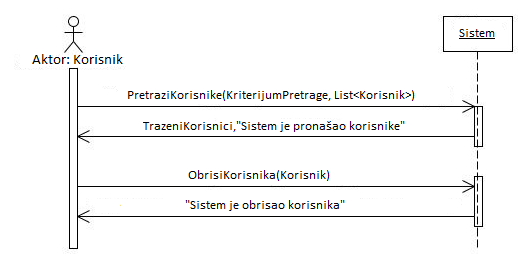


Sa navedenih sekvencnih dijagrama uočavaju se sledece sistemske operacije:

1. signal **PretraziKompozicije**(KriterijumPretrage, List<Kompozicija>)
2. signal **KupiKompoziciju**(Kompozicija)

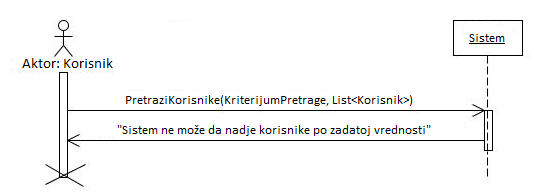
### 2.1.8 DS 8: Brisanje korisnika

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

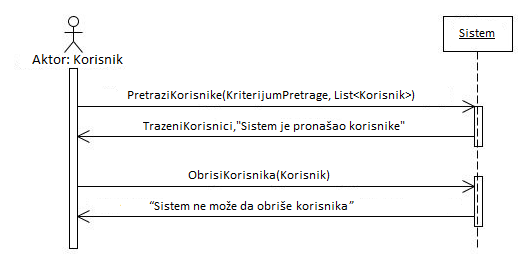


Alternаtivnа scenаrijа

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)

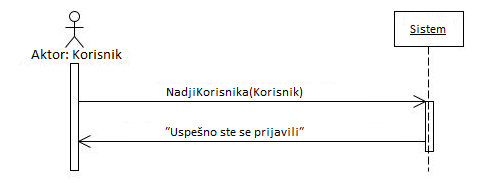


Sa navedenih sekvencnih dijagrama uočavaju se sledece sistemske operacije:

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

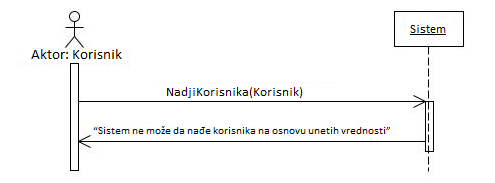
### 2.1.9 DS 9: Prijava korisnika

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



Alternаtivnа scenаrijа

2.1. Ukoliko sistem ne može da nađe korisnika prikazuje poruku “Sistem ne može da nađe korisnika na osnovu unetih vrednosti”. (IA)

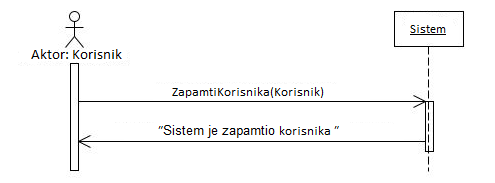


Sa navedenih sekvencnih dijagrama uočava jedna sistemska operacija:

1. signal **NadjiKorisnika**(Korisnik)

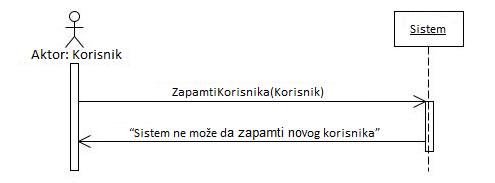
### 2.1.10 DS 10: Unos korisnika

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



Alternаtivnа scenаrijа

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



Sa navedenih sekvencnih dijagrama uočava jedna sistemska operacija:

1. signal **ZapamtiKorisnika**(Korisnik)

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

1. signal **ZapamtiKompoziciju**(Kompozicija)
2. signal **PretraziKompozicije**(KriterijumPretrage, List<Kompozicija>)
3. signal **UcitajKompoziciju**(Kompozicija)
4. signal **ObrisiKompoziciju**(Kompozicija)
5. signal **KupiKompoziciju**(Kompozicija)
6. signal **ZapamtiKorisnika**(Korisnik)
7. signal **NadjiKorisnika**(Korisnik)
8. signal **PretraziKorisnike**(KriterijumPretrage, List<Korisnik>)
9. signal **ObrisiKorisnika**(Korisnik)
10. signal **UcitajKorisnika**(Korisnik)
11. signal **PreslusajZvucniZapis**(ZvucniZapis)

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

**Ugovor UG1: ZapamtiKompoziciju**

Operacija: ZapamtiKompoziciju(*Kompozicija*):signal;

Veza sa SK: SK1, SK4

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

Postuslovi: *Navodi se rezultat operacije .*

**Ugovor UG2: PretraziKompozicije**

Operacija: PretraziKompozicije(KriterijumPretrage, List<Kompozicija):signal;

Veza sa SK: SK2, SK3, SK4, SK5, SK7

Preduslovi: -

Postuslovi: *Navodi se rezultat operacije .*

**Ugovor UG3: UcitajKompoziciju**

Operacija: UcitajKompoziciju(Kompozicija):signal;

Veza sa SK: SK2, SK4, SK5

Preduslovi: -

Postuslovi: *Navodi se rezultat operacije .*

**Ugovor UG4: ObrisiKompoziciju**

Operacija: ObrisiKompoziciju(Kompozicija):signal;

Veza sa SK: SK3

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

Postuslovi: *Navodi se rezultat operacije .*

**Ugovor UG5: KupiKompoziciju**

Operacija: KupiKompoziciju(Kompozicija):signal;

Veza sa SK: SK7

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

Postuslovi: *Navodi se rezultat operacije .*

**Ugovor UG6: ZapamtiKorisnika**

Operacija: ZapamtiKorisnika(Korisnik):signal;

Veza sa SK: SK10

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

Postuslovi: *Navodi se rezultat operacije .*

**Ugovor UG7: NadjiKorisnika**

Operacija: NadjiKorisnika(Korisnik):signal;

Veza sa SK: SK9

Preduslovi: -

Postuslovi: *Navodi se rezultat operacije .*

**Ugovor UG8: PretraziKorisnike**

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

Veza sa SK: SK6, SK8

Preduslovi: -

Postuslovi: *Navodi se rezultat operacije .*

**Ugovor UG9: ObrisiKorisnika**

Operacija: ObrisiKorisnika(Korisnik):signal;

Veza sa SK: SK8

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

Postuslovi: *Navodi se rezultat operacije .*

**Ugovor UG10: UcitajKorisnika**

Operacija: UcitajKorisnika(Korisnik):signal;

Veza sa SK: SK6

Preduslovi: -

Postuslovi: *Navodi se rezultat operacije .*

**Ugovor UG11: PreslusajZvucniZapis**

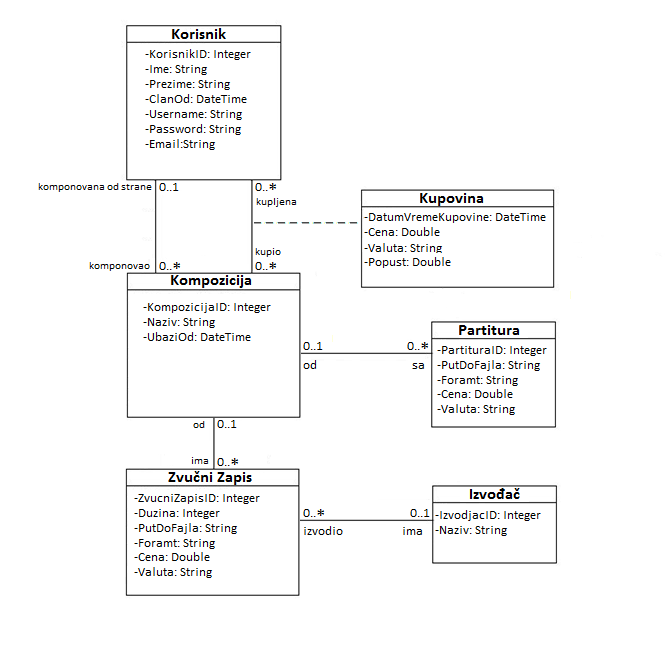
Operacija: UcitajKorisnika(Korisnik):signal;

Veza sa SK: SK5

Preduslovi: -

Postuslovi: *Navodi se rezultat operacije .*

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



## 2.4. Struktura softverskog sistema – Relacioni model

Korisnik(KorisnikID, Ime, Prezime, ClanOd, Username, Password, Email)

Kompozicija(KompozicijaID, Naziv, UbaziOd, *KorisnikID*)

Kupovina(KorisnikID, KompozicijaID, DatumVremeKupovine, Cena, Valuta, Popust)

ZvučniZapis(ZvucniZapisID, Duzina, PutDoFajla, Format, Cena, Valuta, *KompozicijaID*, *IzvodjacID*)

Izvođač(IzvodjacID, Naziv)

Partitura(PartituraID, PutDoFajla, Format, Cena, Valuta, *KompozicijaID*)

## 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  Kompozicija  DELETE RESTRICTED  Kompozicija |
| KorisnikID | Integer | not null and >0 |  |  |
| Ime | String | not null |  |  |
| Prezime | String | not null |  |  |
| ClanOd | DateTime | - |  |  |
| Username | String | not null |  |  |
| Password | String | not null and length>5 |  |  |
| Email | String | - |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Kompozicija** | | **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 Zvučni Zapis, Partitura  DELETE RESTRICTED Zvučni Zapis, Partitura |
| KompozicijaID | Integer | not null and >0 |  |  |
| Naziv | String | not null |  |  |
| UbaziOd | DateTime | not null |  |  |
| KorisnikID | Integer | - |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Kupovina** | | **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 / |
| KorisnikID | Integer | not null and >0 |  |  |
| KompozicijaID | Integer | not null |  |  |
| DatumVreme  Kupovine | DateTime | not null |  |  |
| Cena | Double | (default:0) |  |  |
| Valuta | String | (default:“EUR“) |  |  |
| Popust | Double | >=0, <=1, (default:0) |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Zvučni Zapis** | | **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 / |
| ZvucniZapisID | Integer | not null and >0 |  |  |
| Duzina | Integer | not null |  |  |
| PutDoFajla | String | not null |  |  |
| Format | String | in(“mp3“,“waw“ ,“flac“) (default:“mp3“) |  |  |
| Cena | Double | (default:0) |  |  |
| Valuta | String | (default:“EUR“) |  |  |
| KompozicijaID | Integer | not null |  |  |
| IzvodjacID | Integer | - |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Izvođač** | | **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 Zvučni Zapis  DELETE RESTRICTED  Zvučni Zapis |
| IzvodjacID | Integer | not null and >0 |  |  |
| Naziv | String | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Partitura** | | **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 / |
| PartituraID | Integer | not null and >0 |  |  |
| PutDoFajla | String | not null |  |  |
| Format | String | (default:“png“) |  |  |
| Cena | Double | (default:0) |  |  |
| Valuta | String | (default:“EUR“) |  |  |
| KompozicijaID | Integer | not null |  |  |

# 3.Projektovanje

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

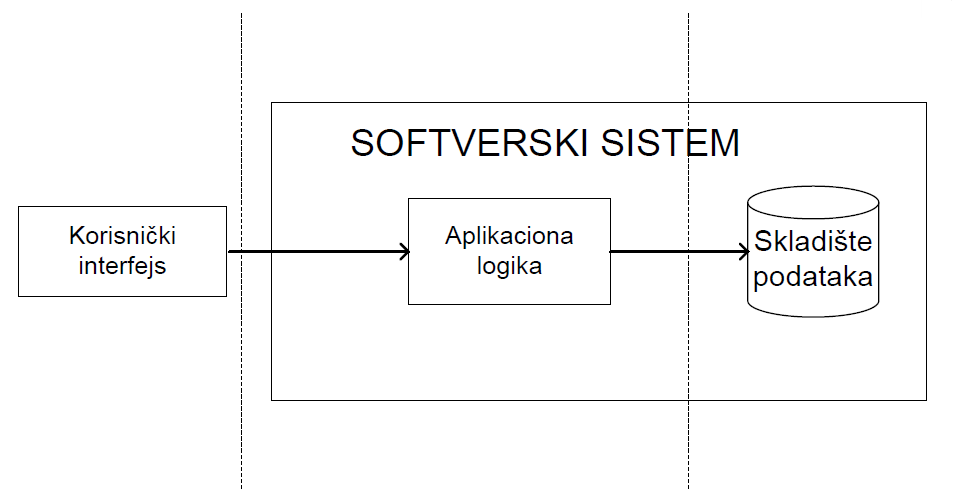
## 3.1 Arhitektura softverskog sistema

Arrhitektura sistema je tronivojska i sastoji se od sledećih nivoa:

• korisnički interfejs

• aplikaciona logika

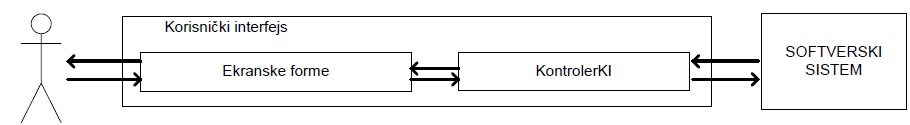
• skladište podataka

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.



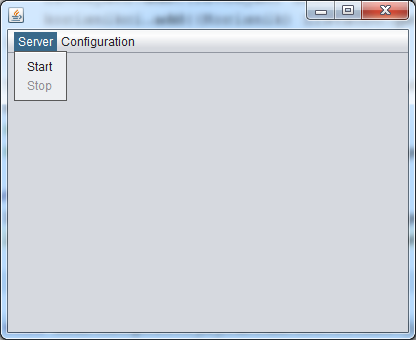
Slika 2. Struktura korisnickog interfejsa

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.

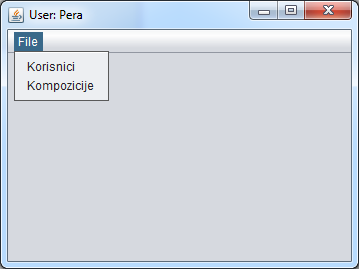
## 3.2.1 Projektovanje ekranskih formi

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 menija glavne ekranske forme klijentskog dela aplikacije.

### 3.2.1.1 SK1 - Unos nove Kompozicije

**Nаziv SK**

Unos nove kompozicije

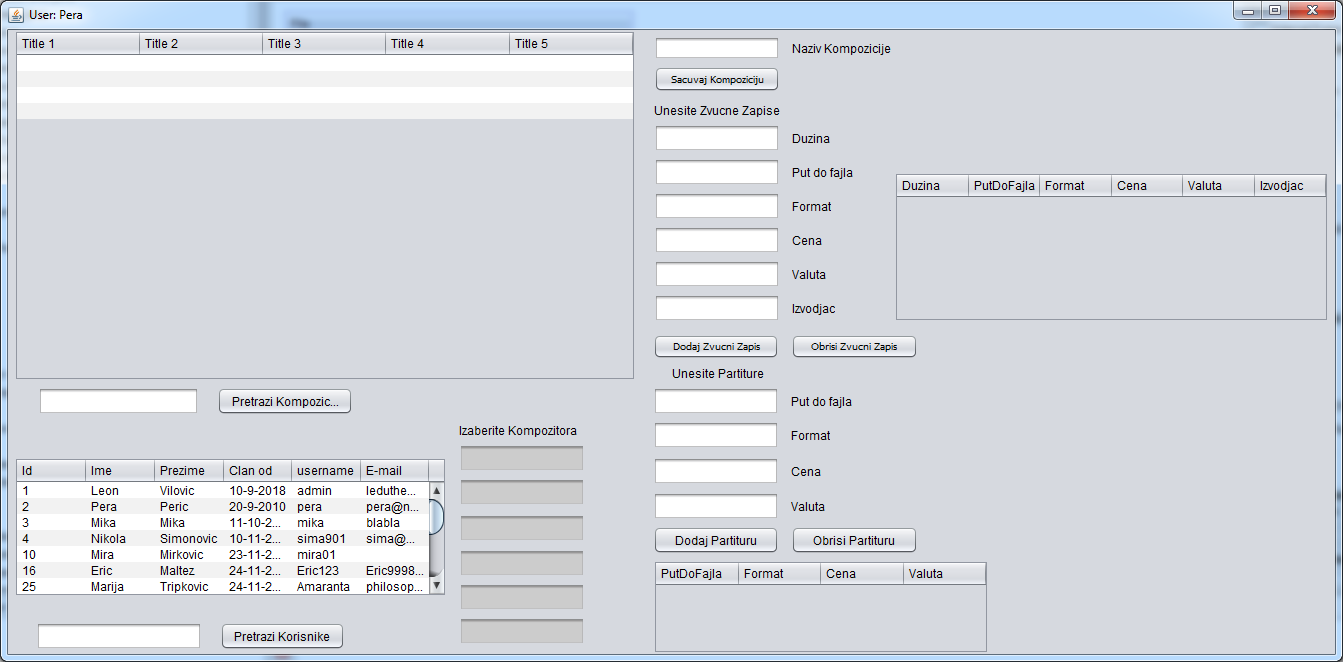
**Aktori SK**

*Korisnik*

**Učesnici SK**

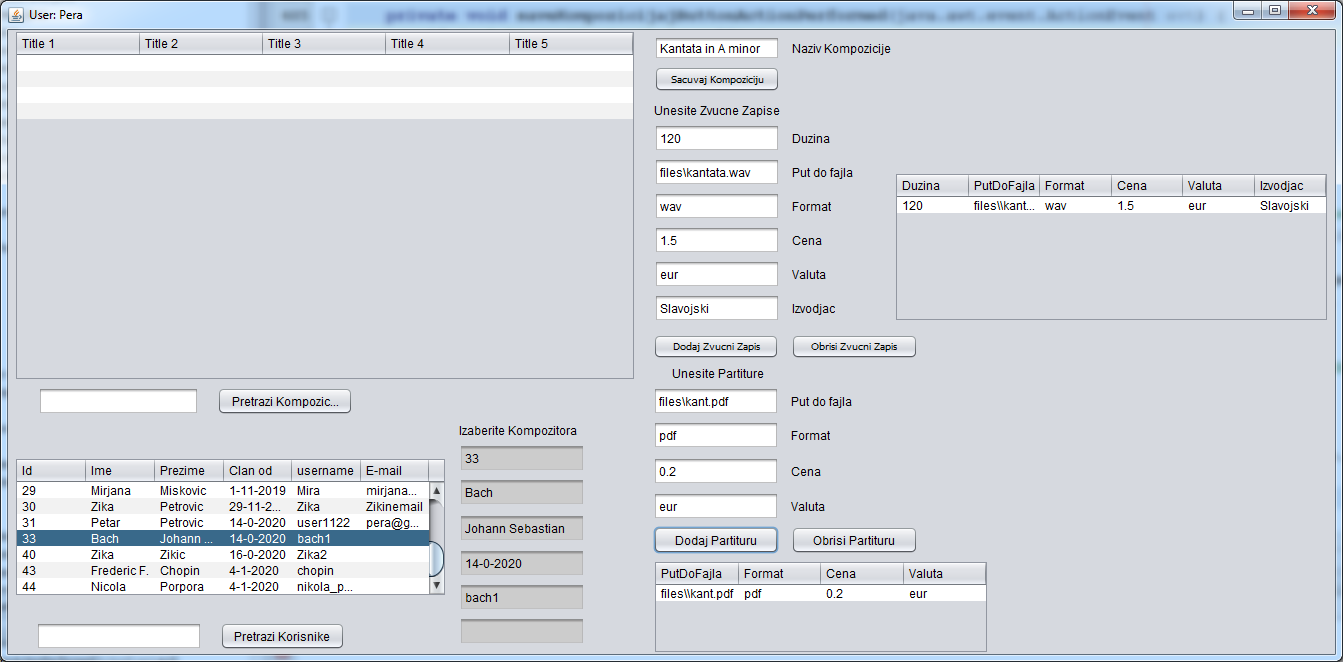
*Korisnik i sistem (progrаm*)

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



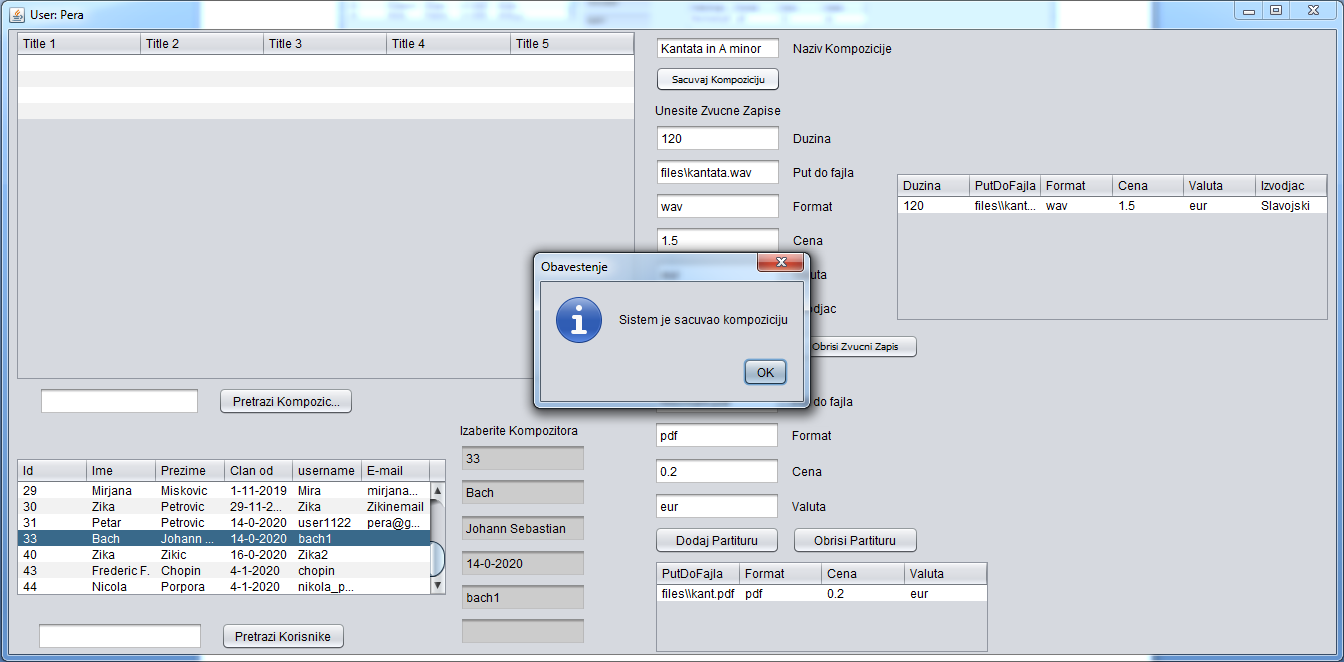
**Osnovni scenаrio SK**

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



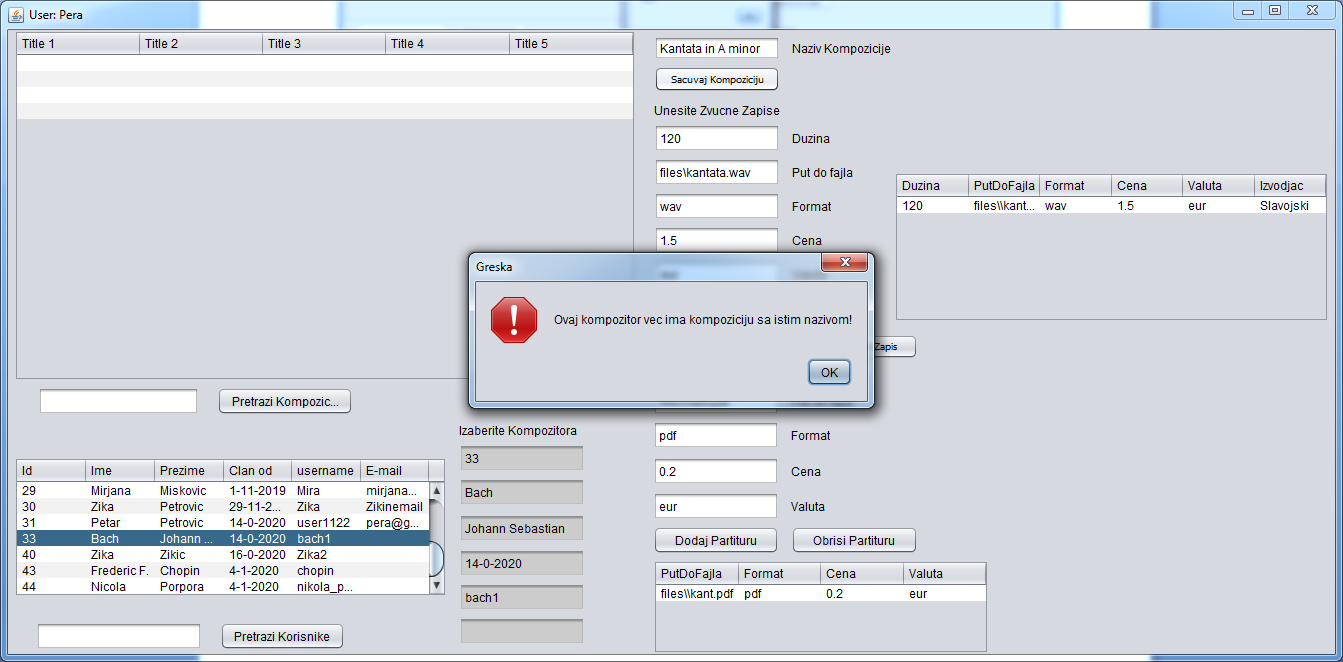
*Opis akcije*: Korisnik pritiskom na kolonu u tabeli sa korinicima bira kompozitora kompozicije. Korisnik moze ako hoce da pripremi partiture ili zvucne zapise kompozicije pritiskom na dugme dodaj partituru ili dodaj zvucni zapis. Nije obavezno dodati partiture i zvučne zapise. Moguće je dodati više zvučnih zapisa ili više partitura.

1. **Korisnik** kontroliše dа li je korektno uneo podаtke i put ka audio fajlu *kompozicije*. (ANSO)
2. **Korisnik** pozivа *sistem* dа zаpаmti podаtke o *kompoziciji*. (APSO)
3. **Sistem** pаmti podаtke o *kompoziciji*. (SO)
4. **Sistem** prikаzuje *kompozitoru* zаpаmćenu *kompoziciju* i poruku: ʺSistem je sacuvao kompozicijuʺ. (IA)



Alternаtivnа scenаrijа

5.1 Ukoliko sistem ne može dа zаpаmti podаtke o novoj *kompoziciji* on prikаzuje korisniku poruku sa objašnjenem zasto kompozicija nije zapamćena. (IA)



### 3.2.1.2 SK2 - Pretraživanje kompozicija

**Nаziv SK**

Pretrаživаnje kompozicija

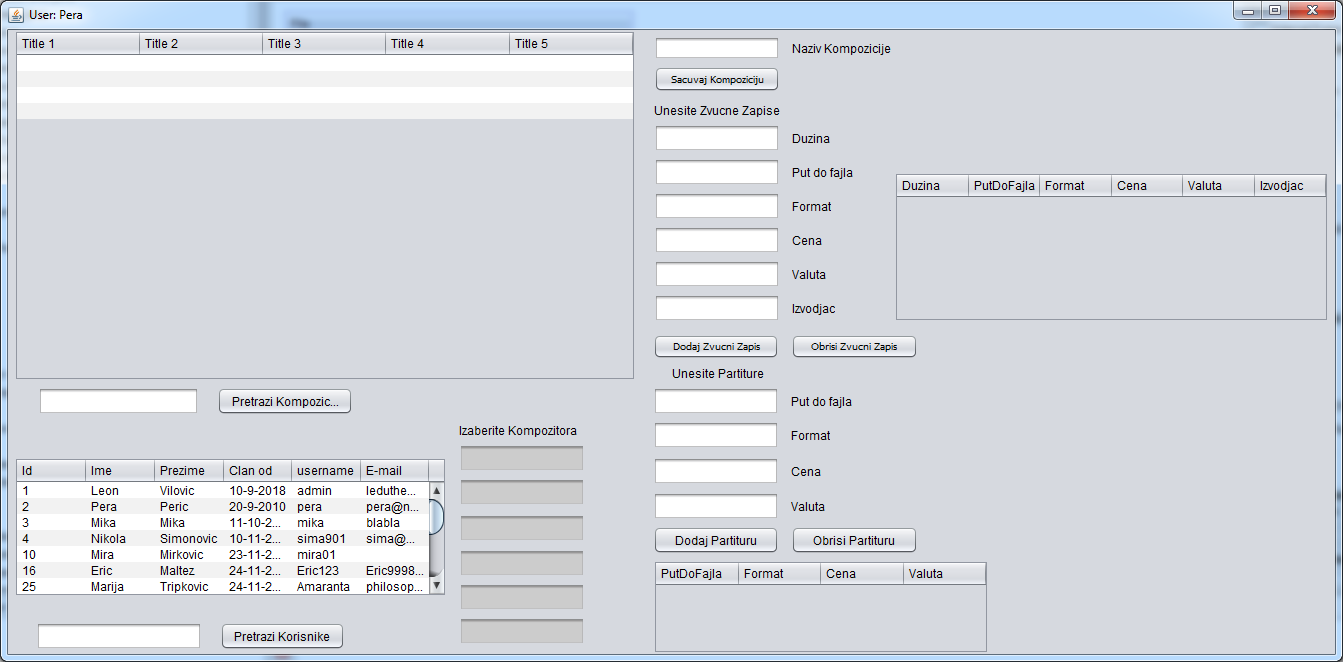
**Aktori SK**

*Korisnik*

**Učesnici SK**

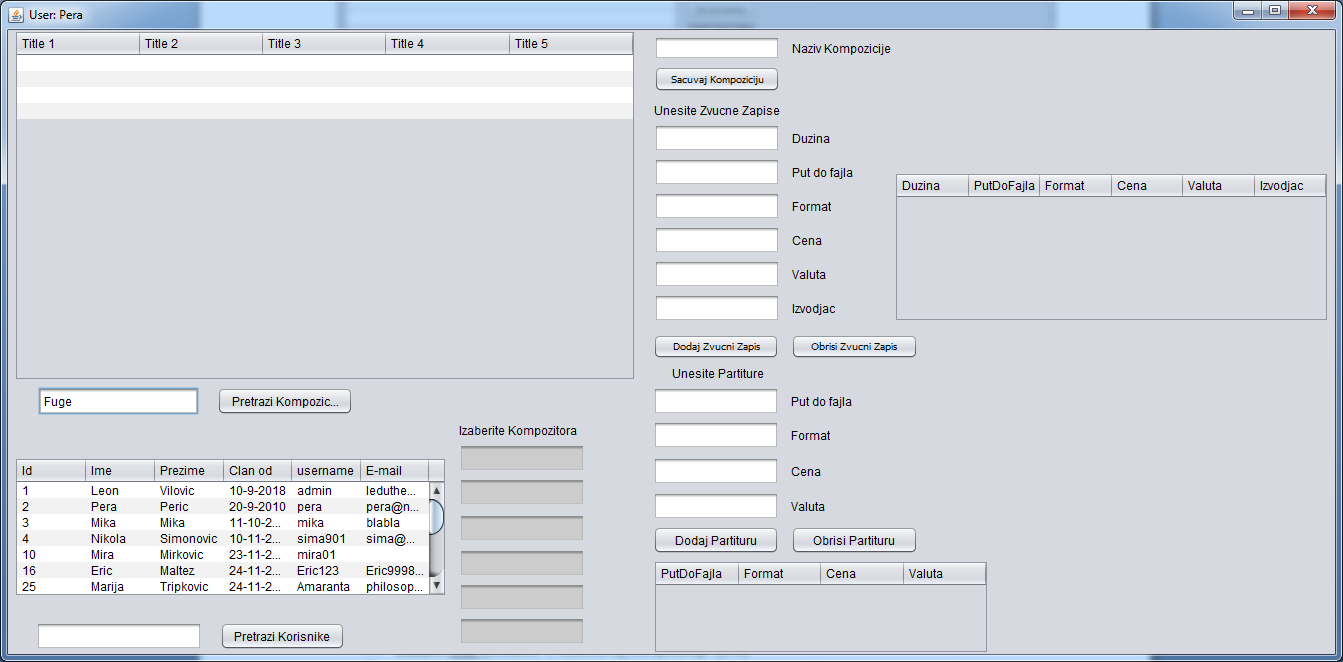
*Korisnik i sistem (progrаm*)

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

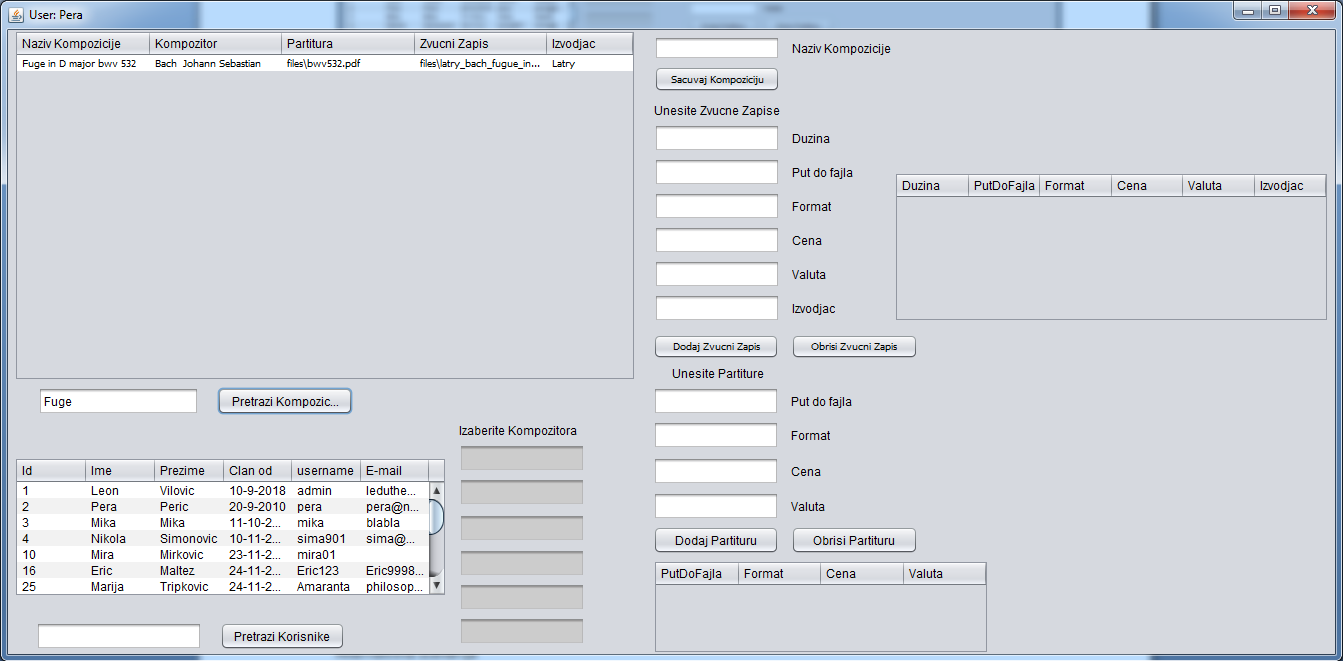


**Osnovni scenаrio SK**

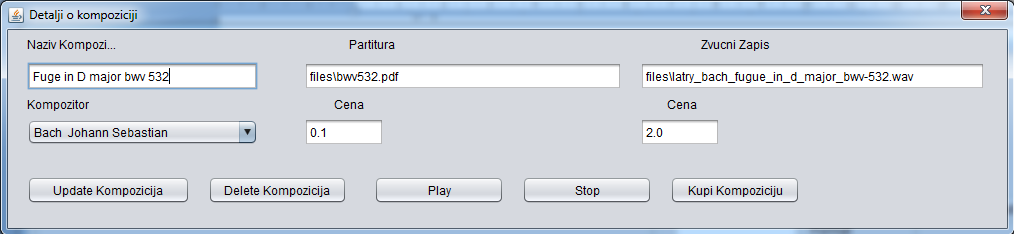
1. **Korisnik** unosi vrednost po kojoj pretrаžuje *kompozicije*. (APUSO)



1. **Korisnik** pozivа *sistem* dа nаđe *kompozicije* po zаdаtoj vrednosti. (APSO)
2. **Sistem** trаži *kompozicije* po zаdаtoj vrednosti. (SO)
3. **Sistem** prikazuje *korisniku* nađene *kompozicije* i poruku ʺSistem je pronašao kompozicijeʺ. (IA)

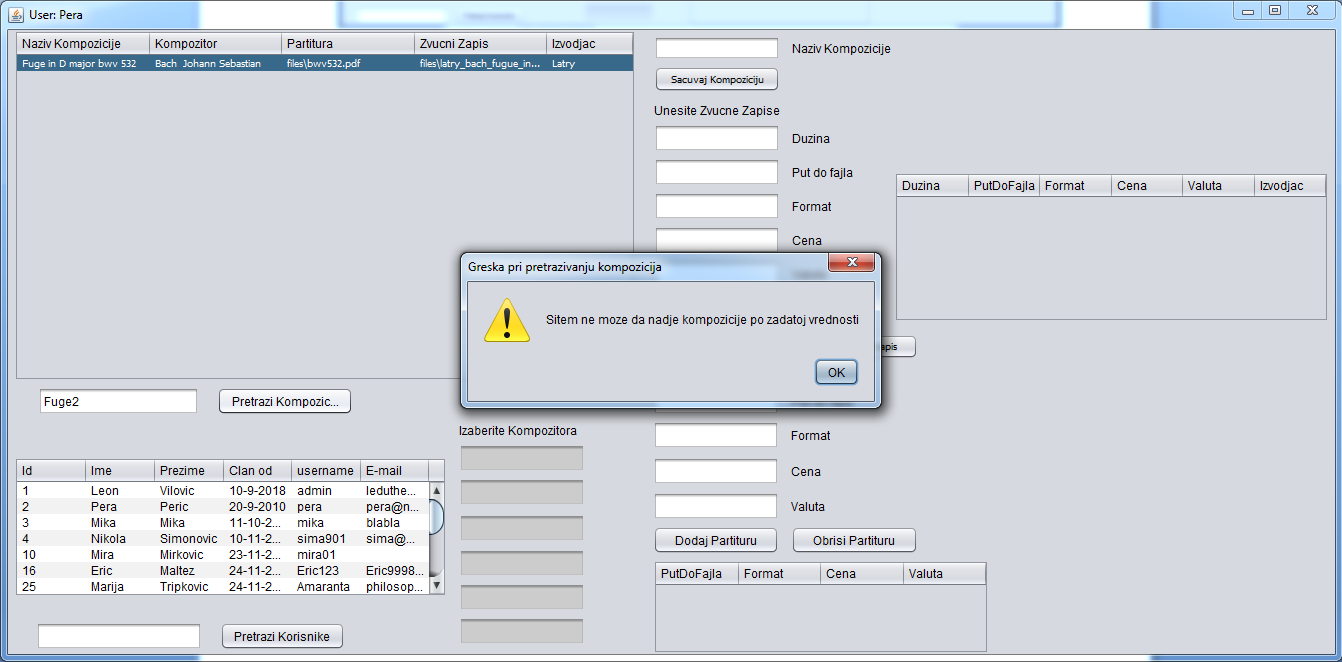


1. **Korisnik** bira*kompoziciju* čije podatke želi da vidi. (APUSO)
2. **Korisnik** poziva sistem da prikaže podatke o izabranoj *kompoziciji*. (APSO)
3. **Sistem** traži podatke o izabranoj *kompoziciji*. (SO)
4. **Sistem** prikаzuje *korisniku* podаtke o *kompoziciji* i poruku: ʺOvo su podaci o vašoj kompozicijiʺ. (IA)



Alternаtivnа scenаrijа

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



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

### 3.2.1.3 SK3: Brisanje kompozicije

**Nаziv** **SK**

Brisanje kompozicija

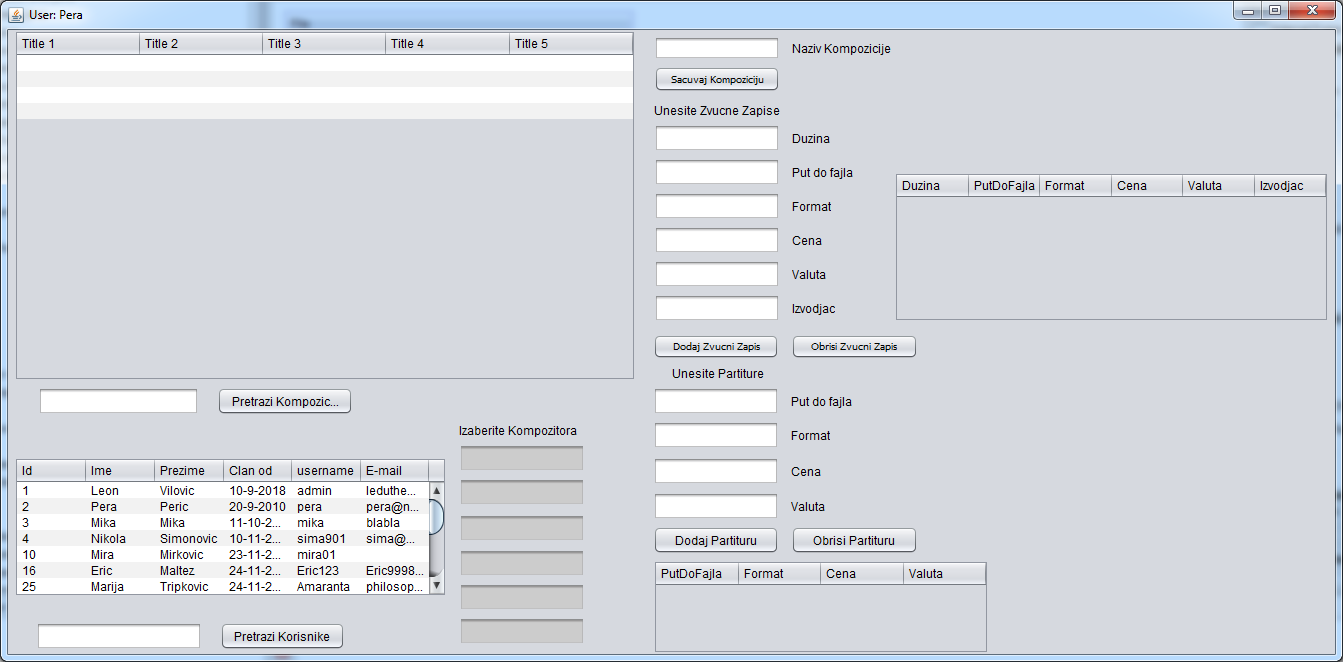
**Aktori SK**

*Korisnik*

**Učesnici SK**

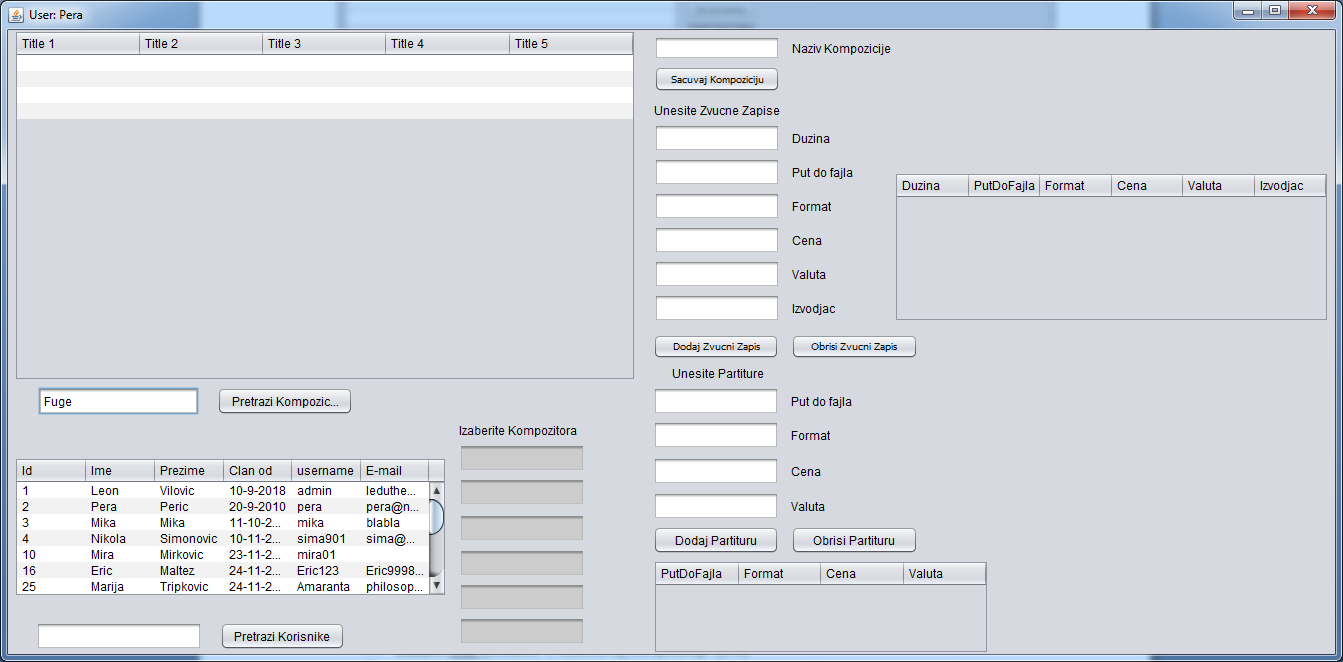
*Korisnik i sistem (progrаm)*

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

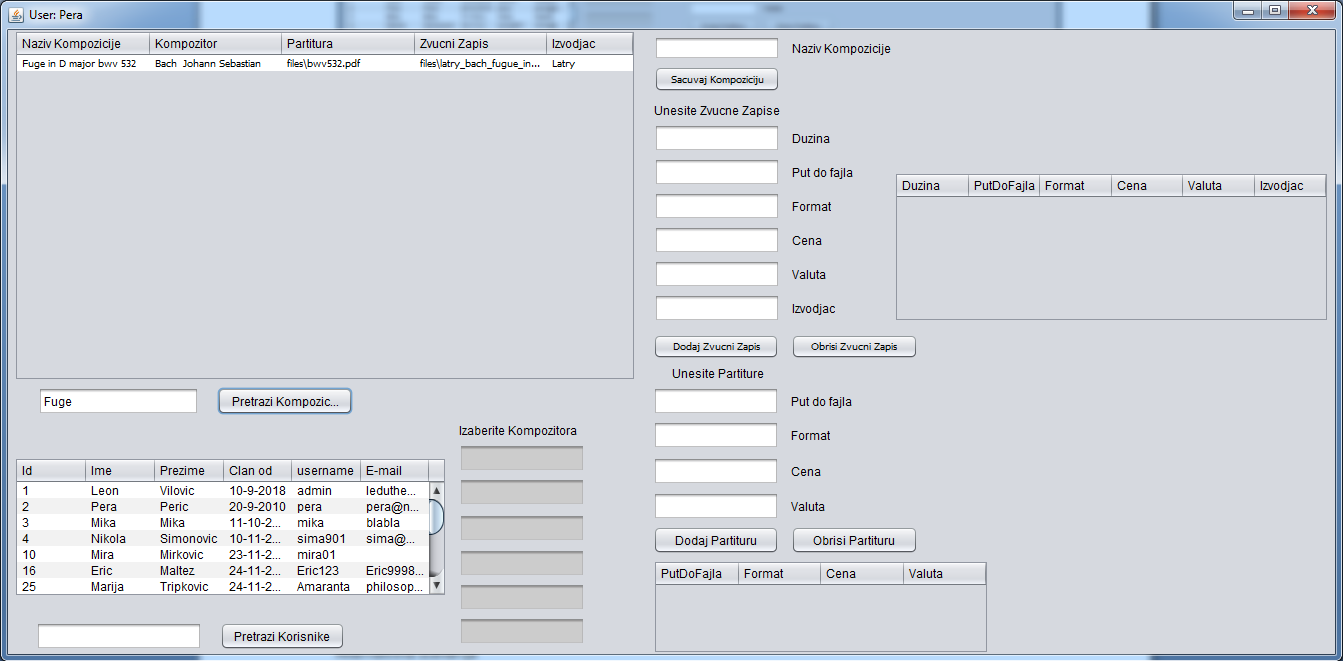


**Osnovni scenаrio SK**

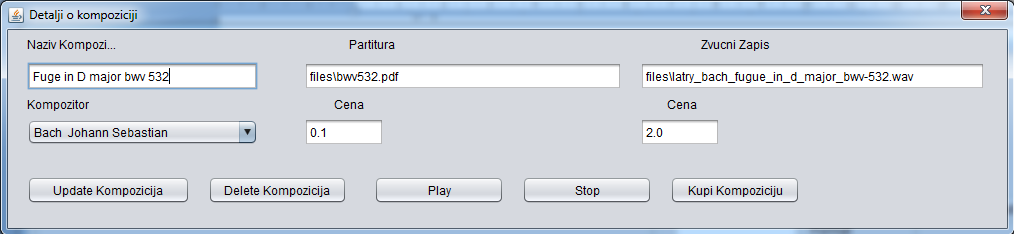
1. **Korisnik** unosi vrednost po kojoj se pretrаžuju *kompozicije*. (APUSO)



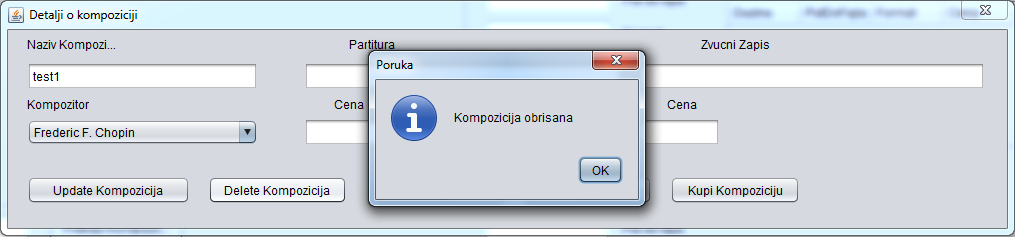
1. **Korisnik** pozivа *sistem* dа nаđe *kompozicije* po zаdаtoj vrednosti. (APSO)
2. **Sistem** trаži *kompozicije* po zаdаtoj vrednosti. (SO)
3. **Sistem** prikаzuje *korisniku* pronađene *kompozicije* i poruku ʺSistem je pronašao kompozicijeʺ. (IA)



1. **Korisnik** bira *kompoziciju* koju želi da obriše. (APUSO)

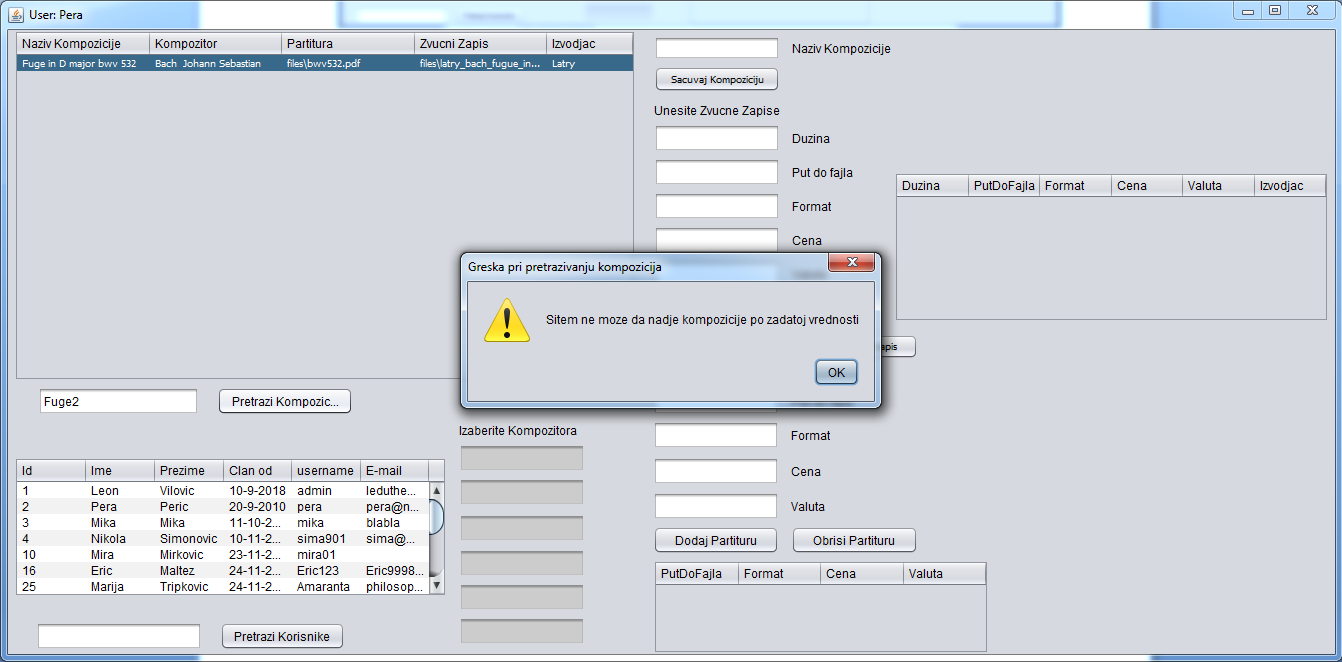


1. **Korisnik** poziva *sistem* da obriše izabranu *kompoziciju*. (APSO)
2. **Sistem** briše izabranu *kompoziciju*. (SO)
3. **Sistem** prikazuje *korisniku* poruku: “Sistem je obrisao kompoziciju”. (IA)



Alternаtivnа scenаrijа

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



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

### 3.2.1.4 SK4: Izmena kompozicije

**Nаziv SK**

Izmenа kompozicije

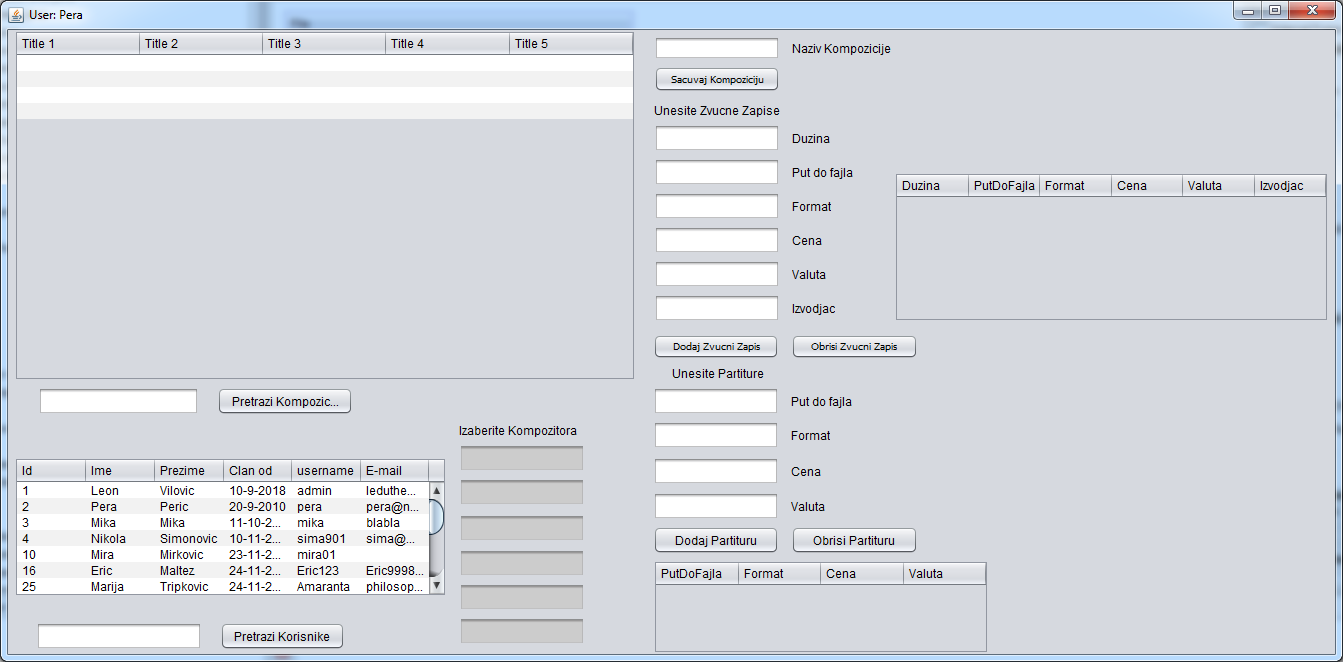
**Aktori SK**

*Korisnik*

**Učesnici SK**

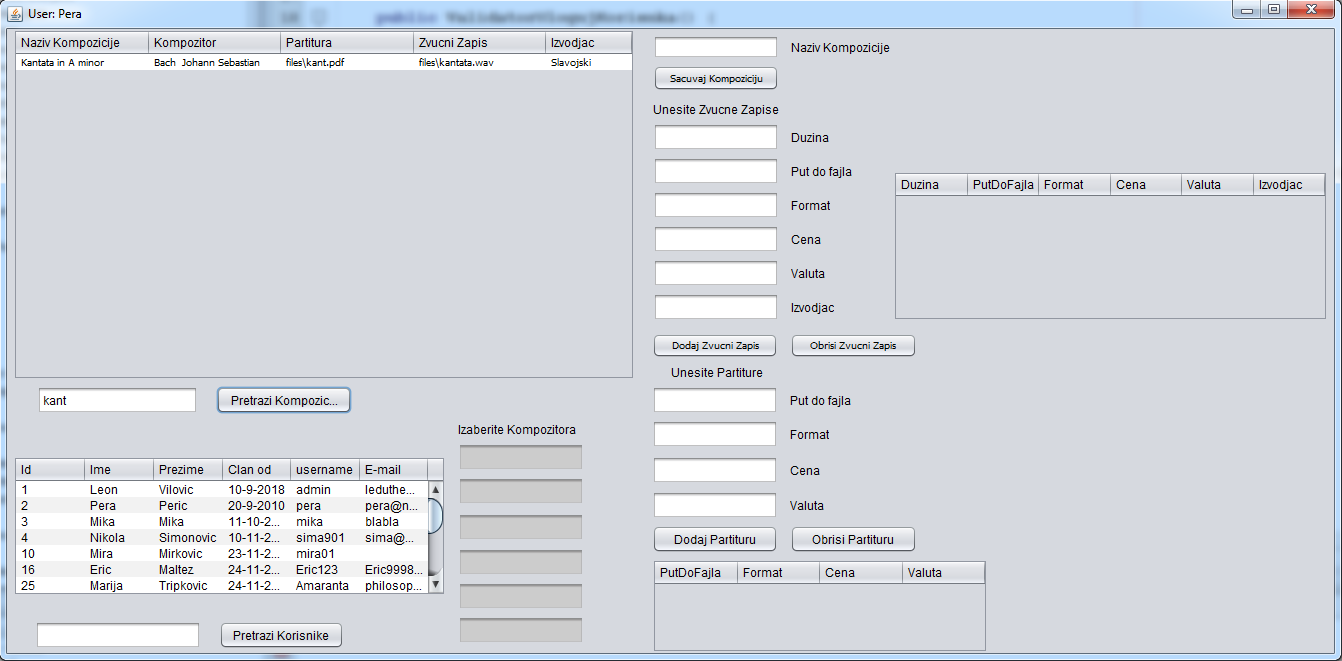
*Korisnik i sistem (progrаm*)

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

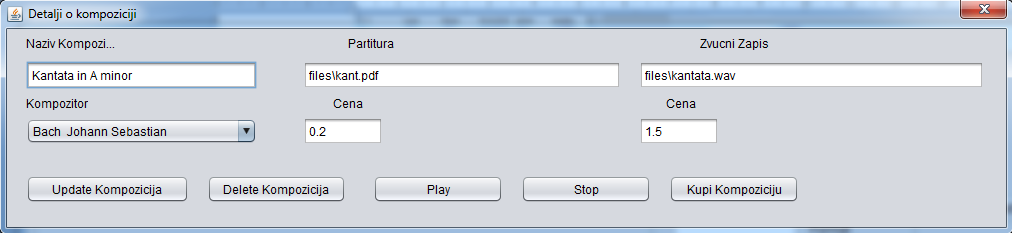


**Osnovni scenаrio SK**

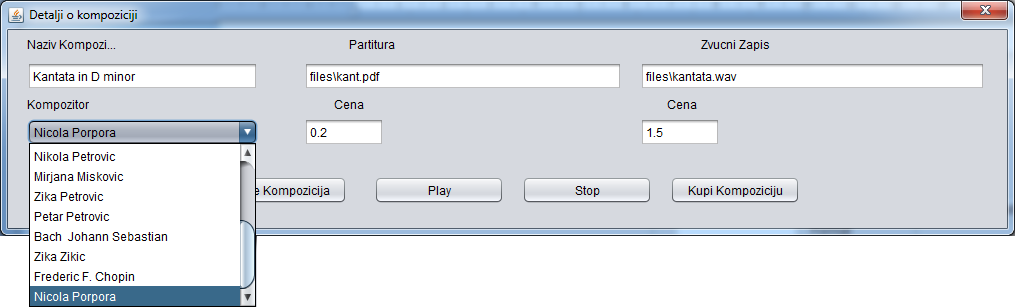
1. **Korisnik** unosi vrednost po kojoj se pretrаžuju *kompozicije*. (APUSO)



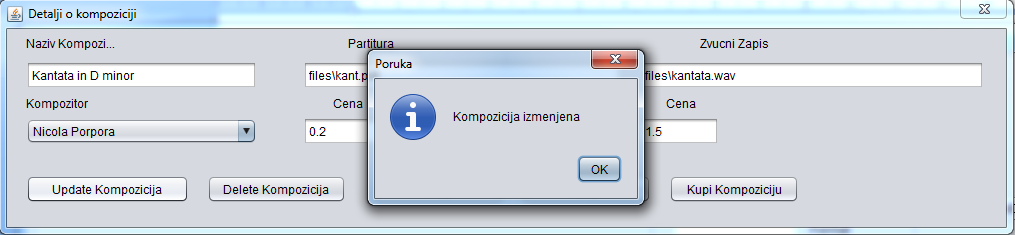
1. **Korisnik** pozivа *sistem* dа nаđe *kompozicije* po zаdаtoj vrednosti. (APSO)
2. **Sistem** trаži *kompozicije* po zаdаtoj vrednosti. (SO)
3. **Sistem** prikаzuje *korisniku* pronađene *kompozicije* i poruku ʺSistem je pronašao kompozicijeʺ. (IA)
4. **Korisnik** bira *kompoziciju* koju želi da izmeni. (APUSO)
5. **Korisnik** pozivа *sistem* da učita podatke o izabranoj *kompoziciji.* (APSO)
6. **Sistem** trаži podatke o izabranoj *kompoziciji.* (SO)



1. **Sistem** prikаzuje *korisniku* podatke o izabranoj *kompoziciji.* (IA)
2. **Korisnik** unosi (menja) *kompoziciju*. (APUSO)

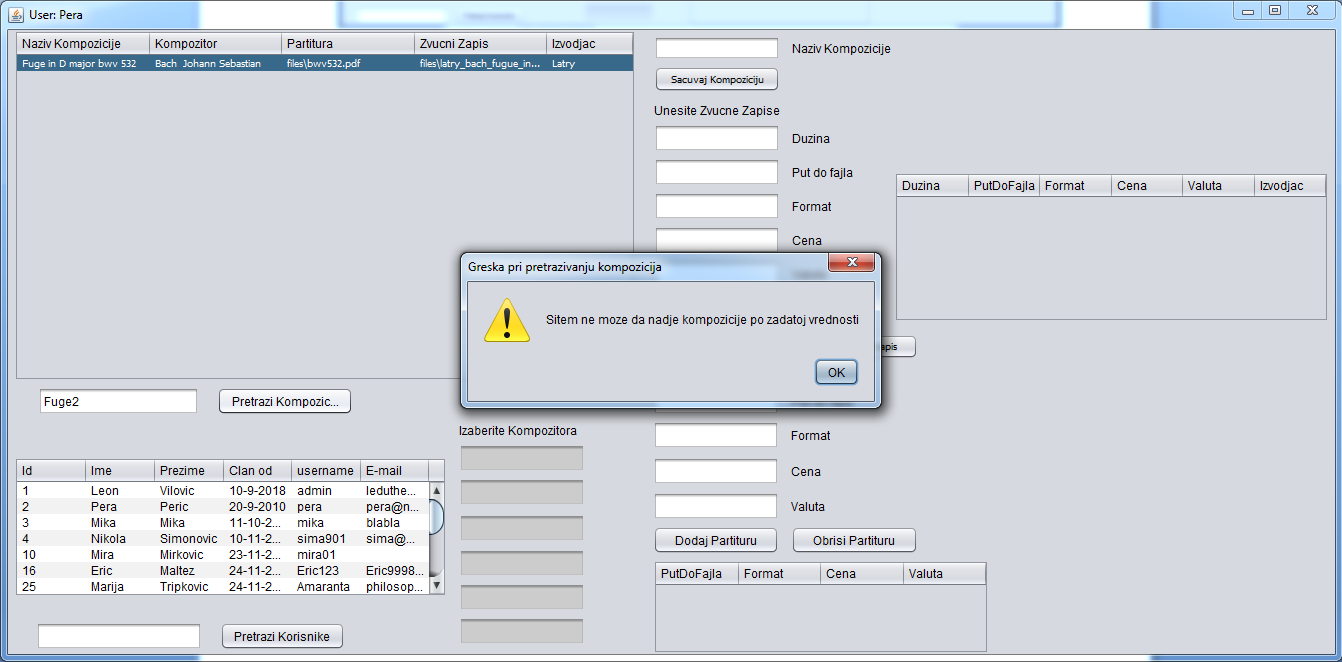


1. **Korisnik** kontroliše dа li je korektno uneo podаtke o *kompoziciji*. (ANSO)
2. **Korisnik** pozivа *sistem* dа zаpаmti podatke o *kompoziciji*.(APSO)
3. **Sistem** pаmti podatke o *kompoziciji*. (SO)
4. **Sistem** prikаzuje *korisniku* poruku: ʺSistem je zаpаmtio kompozicijuʺ. (IA)



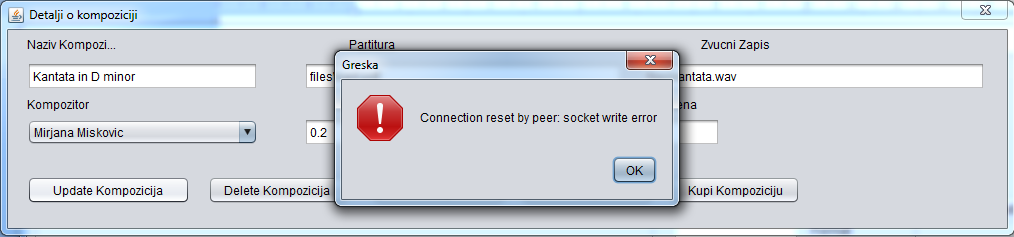
Alternаtivnа scenаrijа

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



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

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



### 3.2.1.5 SK5: Preslušavanje kompozicije

**Nаziv SK**

Preslušavanje kompozicija

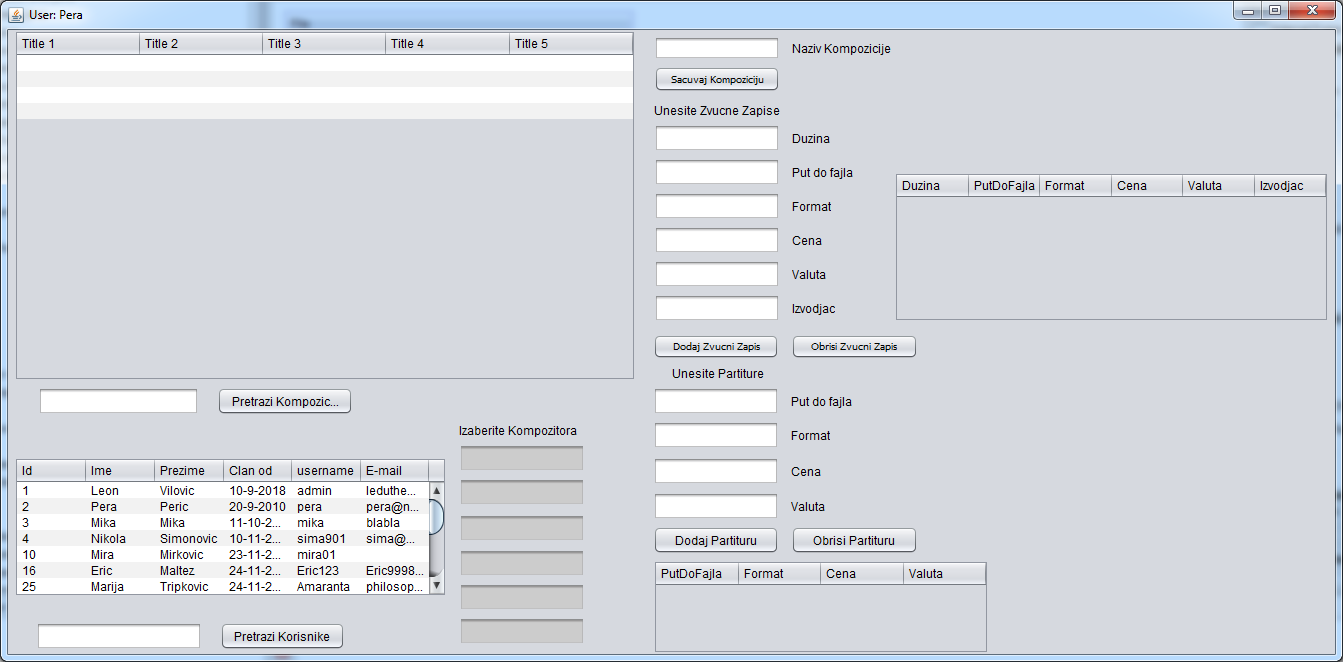
**Aktori SK**

*Korisnik*

**Učesnici SK**

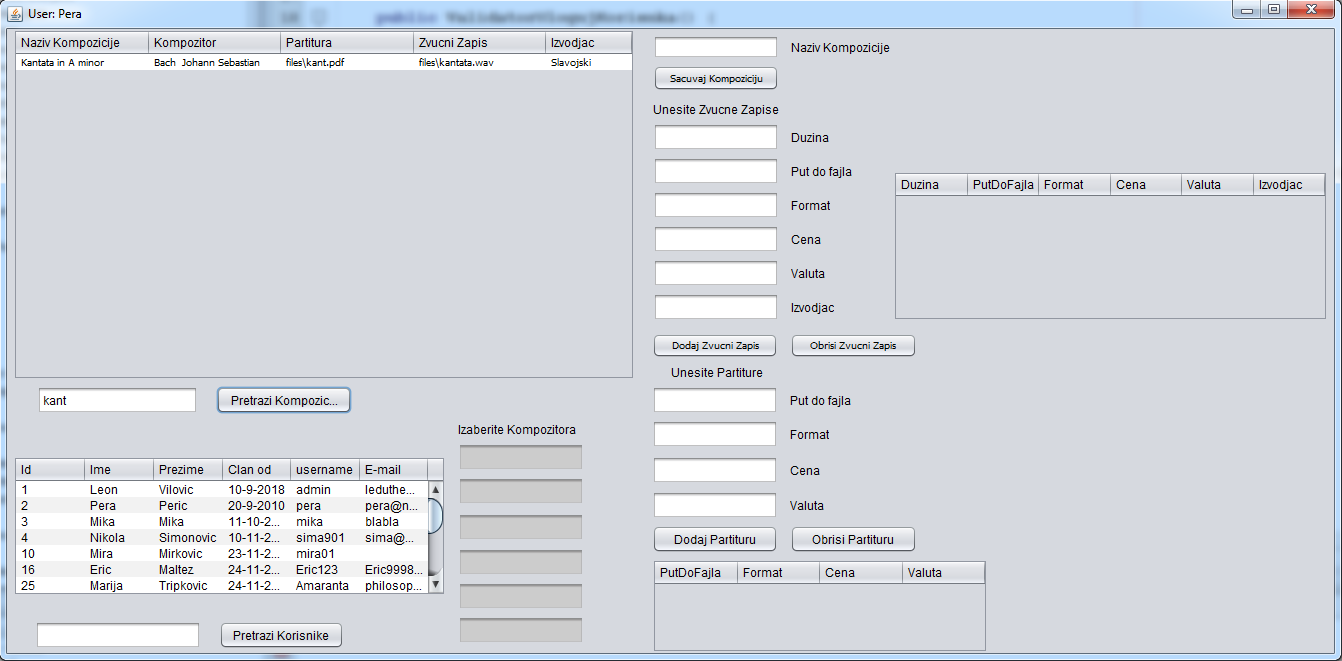
*Korisnik i sistem (progrаm*)

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

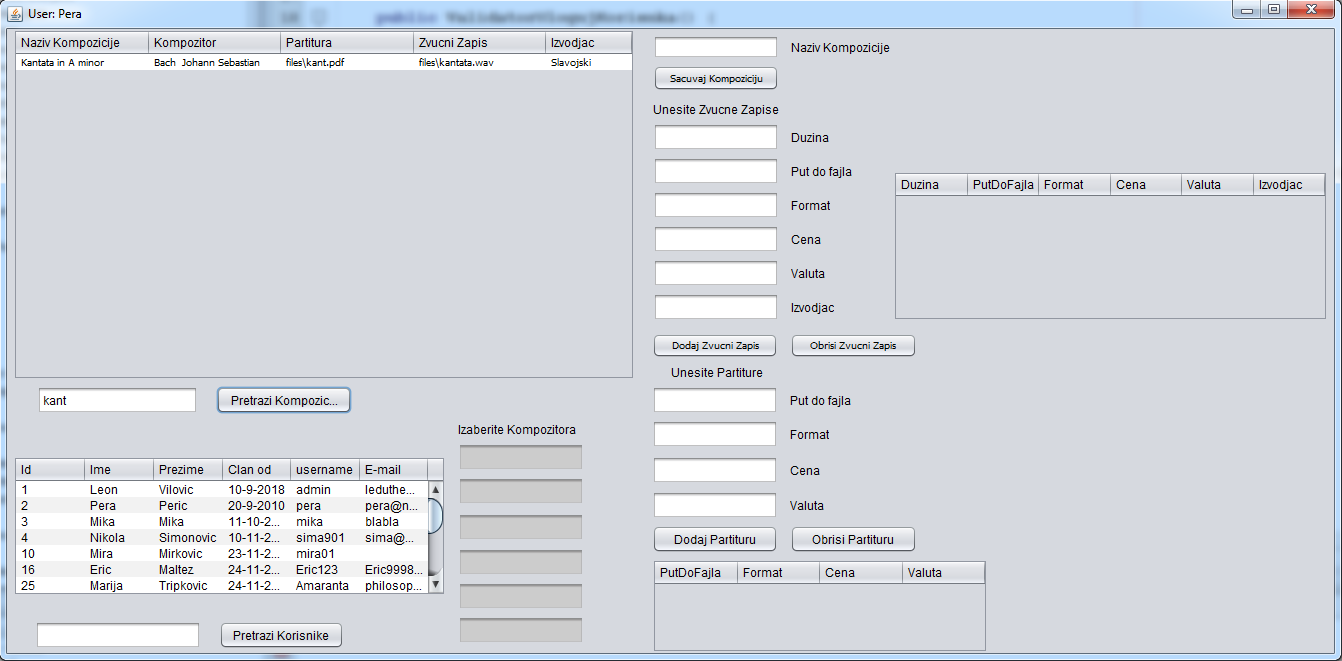


**Osnovni scenаrio SK**

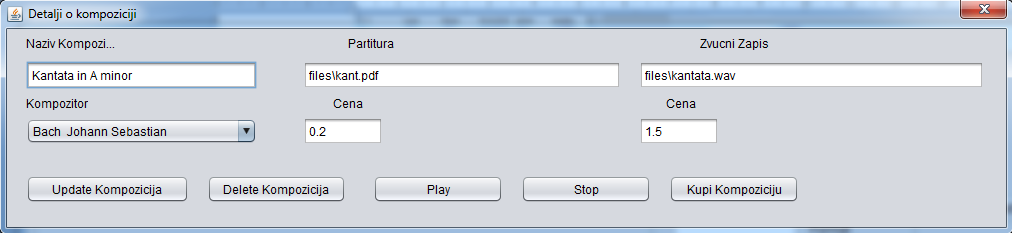
1. **Korisnik** unosi vrednost po kojoj pretrаžuje *kompozicije*. (APUSO)



1. **Korisnik** pozivа *sistem* dа nаđe *kompozicije* po zаdаtoj vrednosti. (APSO)
2. **Sistem** trаži *kompozicije* po zаdаtoj vrednosti. (SO)
3. **Sistem** prikazuje *korisniku* nađene *kompozicije* i poruku ʺSistem je pronašao kompozicijeʺ.(IA)

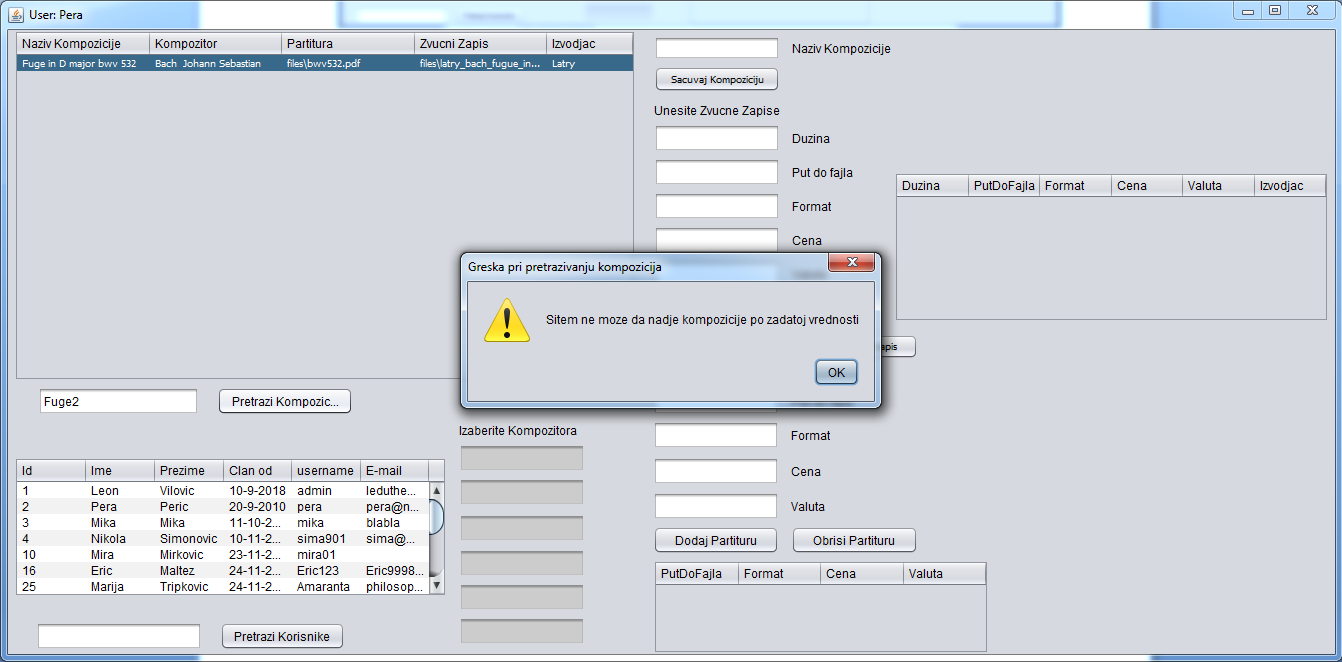


1. **Korisnik** bira*kompoziciju* koju želi da čuje. (APUSO)
2. **Korisnik** poziva sistem da prikaže podatke o izabranoj *kompoziciji*. (APSO)
3. **Sistem** traži podatke izabrane *kompozicije*. (SO)
4. **Sistem** prikаzuje *korisniku* podаtke o *kompoziciji,* poruku ʺOvo su podaci o vašoj kompozicijiʺ. (IA)



Alternаtivnа scenаrijа

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



8.1 Ukoliko sistem ne može dа nađe *kompoziciju* on prikаzuje korisniku poruku: “Sistem ne može dа nаđe izabranu *kompoziciju*”. (IA)

### 3.2.1.6 SK6: Pretraživanje korisnika

**Nаziv SK**

Pretrаživаnje kompozicija

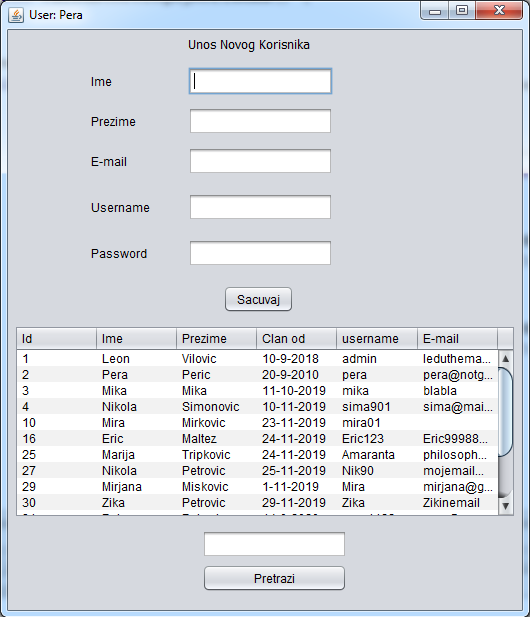
**Aktori SK**

*Korisnik*

**Učesnici SK**

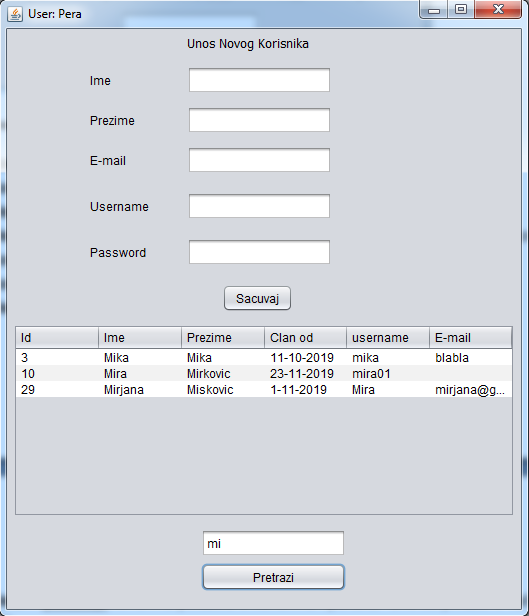
*Korisnik i sistem (progrаm*)

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

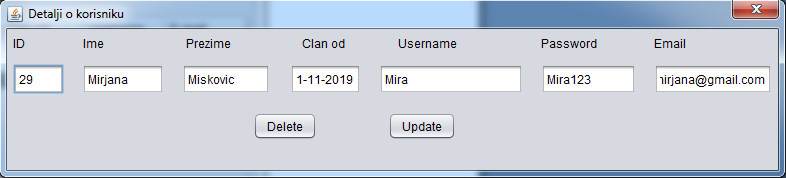


**Osnovni scenаrio SK**

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* i poruku ʺSistem je pronašao korisnikeʺ. (IA)

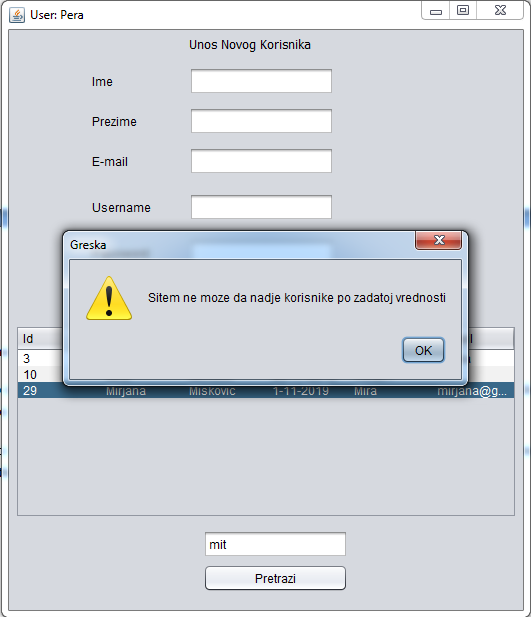


1. **Korisnik** bira*korisnika* čije podatke želi da vidi. (APUSO)
2. **Korisnik** poziva sistem da prikaže podatke o izabranom *korisniku*. (APSO)
3. **Sistem** traži podatke o izabranom *korisniku*. (SO)
4. **Sistem** prikаzuje *korisniku* podаtke o *korisniku* i poruku ʺOvo su podaci o vašem korisnikuʺ. (IA)

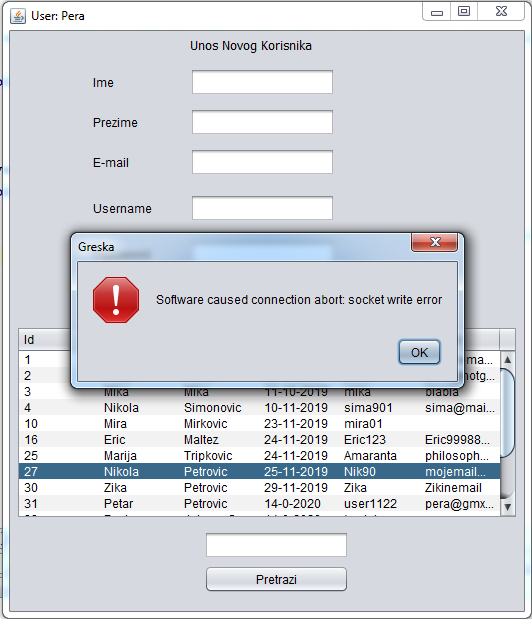


Alternаtivnа scenаrijа

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)



### 3.2.1.7 SK7: Kupovina kompozicije (složen slucaj)

**Nаziv SK**

Kupovina kompozicija

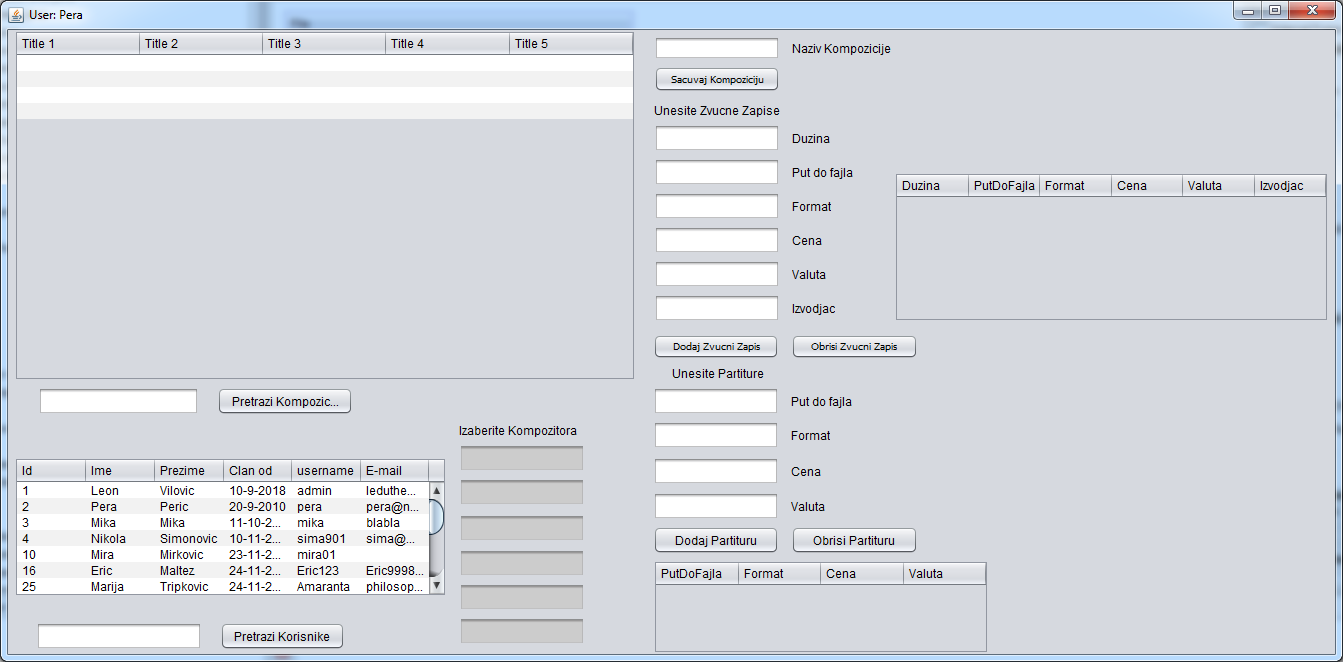
**Aktori SK**

*Korisnik*

**Učesnici SK**

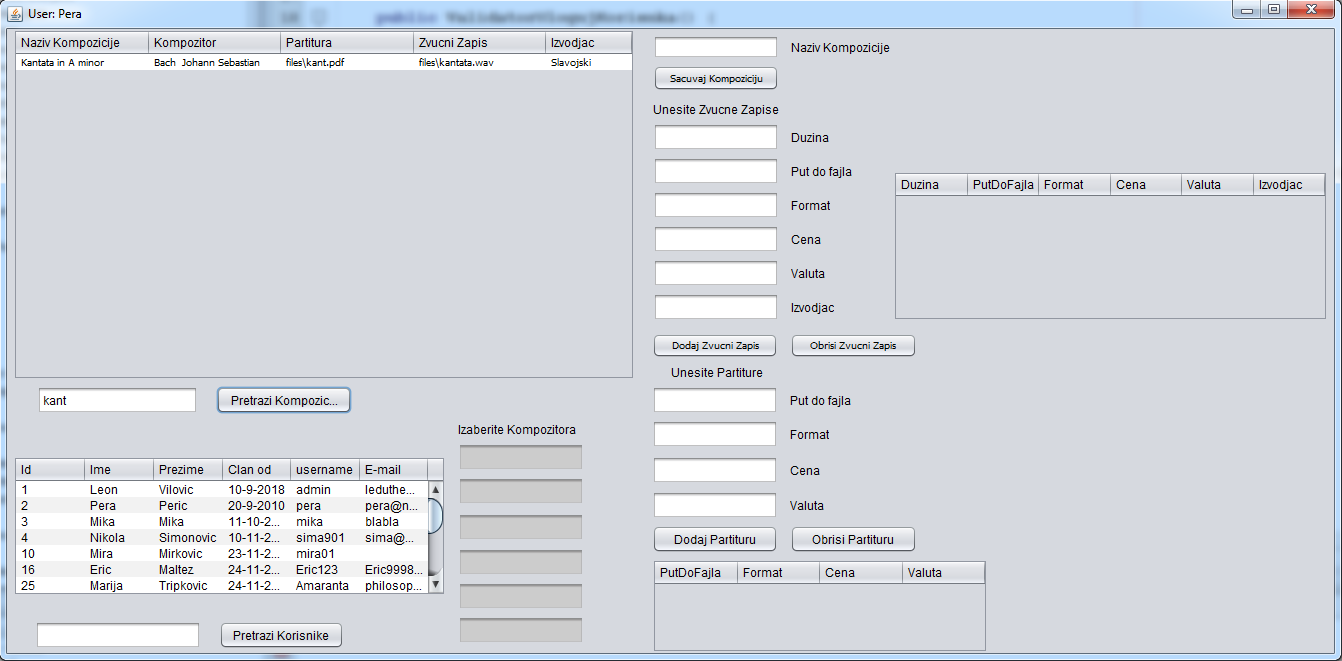
*Korisnik i sistem (progrаm*)

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

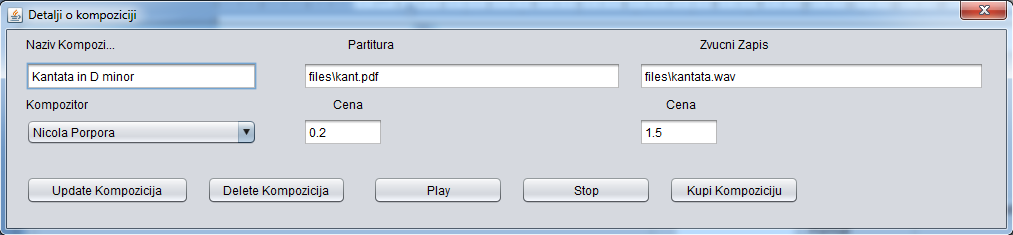


**Osnovni scenаrio SK**

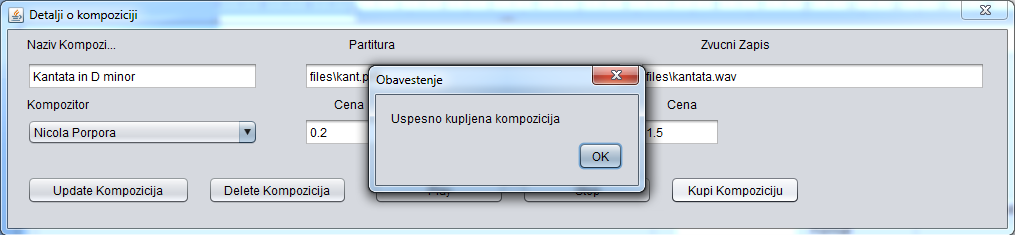
1. **Korisnik** unosi vrednost po kojoj pretrаžuje *kompozicije*. (APUSO)



1. **Korisnik** pozivа *sistem* dа nаđe *kompozicije* po zаdаtoj vrednosti. (APSO)
2. **Sistem** trаži *kompozicije* po zаdаtoj vrednosti. (SO)
3. **Sistem** prikazuje *korisniku* nađene *kompozicije* i poruku ʺSistem je pronašao kompozicijeʺ. (IA)
4. **Korisnik** bira*kompoziciju* koju želi da kupi. (APUSO)

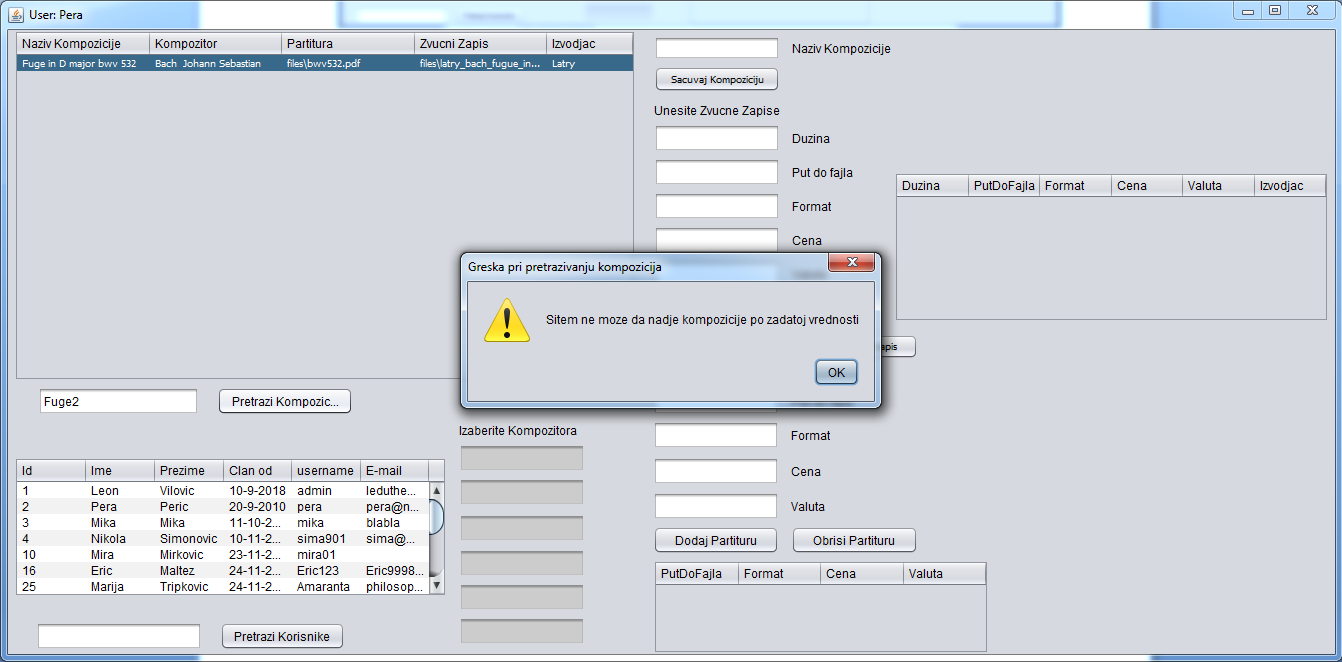


1. **Korisnik** poziva sistem da pokrene transakciju. (APSO)
2. **Sistem** traži podatke izabrane *kompozicije*. (SO)
3. **Sistem** prikаzuje *korisniku* podаtke o *kompoziciji,* poruku ʺKupili ste kompozicijuʺ. (IA)

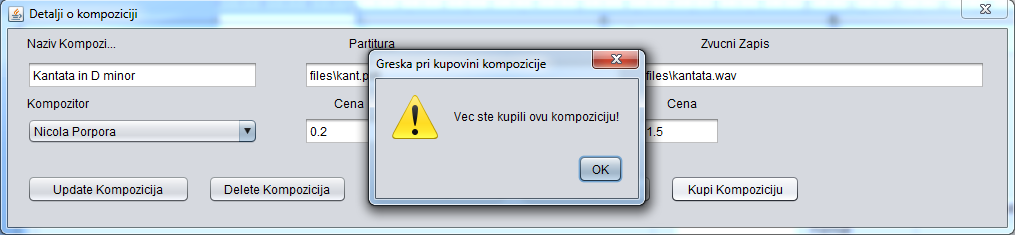


Alternаtivnа scenаrijа

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



8.1 Ukoliko sistem ne može dа pronađe *audio fajl* ili je korisnik već kupio kompoziciju, ponistava se transakcija i prikаzuje se poruka. (IA)



### 3.2.1.8 SK8: Brisanje korisnika

**Nаziv** **SK**

Brisanje korisnika

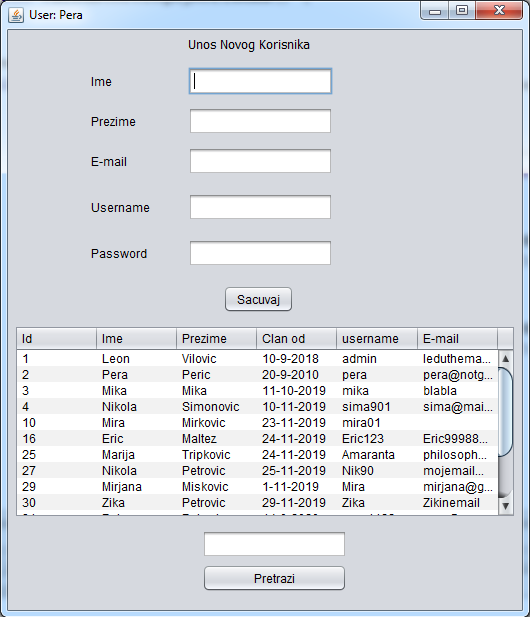
**Aktori SK**

*Korisnik*

**Učesnici SK**

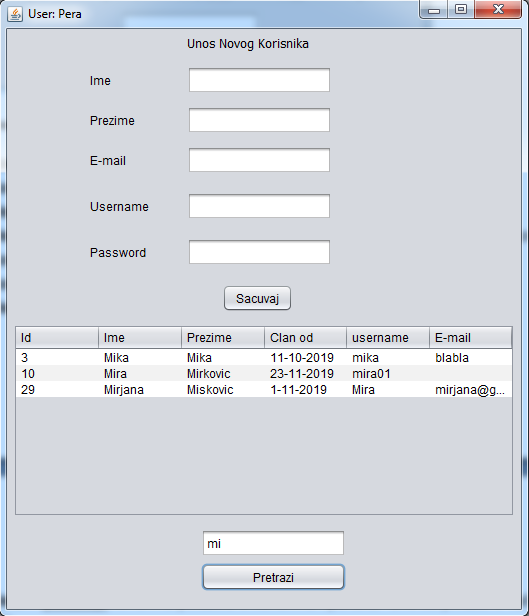
*Korisnik i sistem (progrаm*)

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

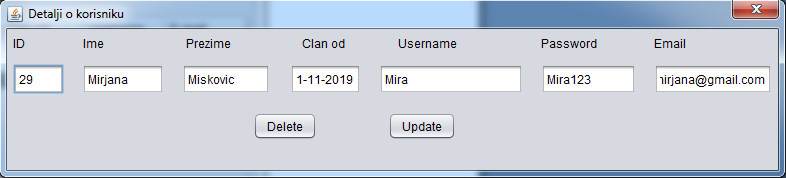


**Osnovni scenаrio SK**

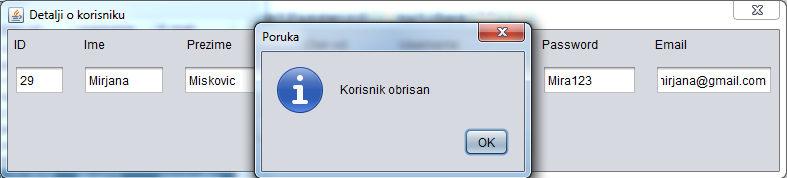
1. **Korisnik** unosi vrednost po kojoj se pretrаžuje *korisnike*. (APUSO)
2. **Korisnik** pozivа *sistem* dа nаđe *korisnike* po zаdаtoj vrednosti. (APSO)
3. **Sistem** trаži *korisnike* po zаdаtoj vrednosti. (SO)
4. **Sistem** prikаzuje *korisniku* pronađene *korisnike* i poruku ʺSistem je pronašao korisnikeʺ. (IA)



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

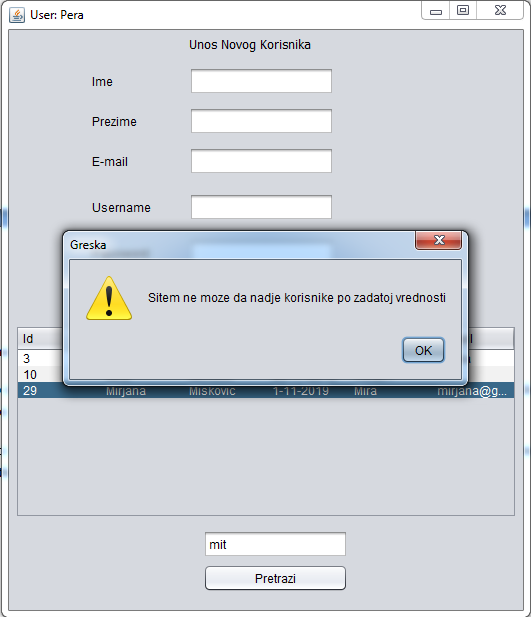


1. **Korisnik** poziva *sistem* da obriše izabranog *korisnika*. (APSO)
2. **Sistem** briše izabranog *korisnika*. (SO)
3. **Sistem** prikazuje *korisniku* poruku: “Sistem je obrisao *korisnika* ”. (IA)

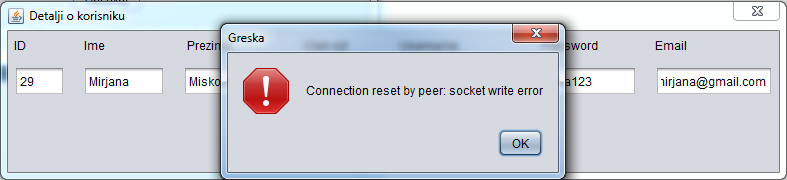


Alternаtivnа scenаrijа

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 sa objašnjenem zašto korisnik nije obrisan. (IA)



### 3.2.1.9 SK9: Prijava korisnika

**Nаziv** **SK**

Prijava korisnika

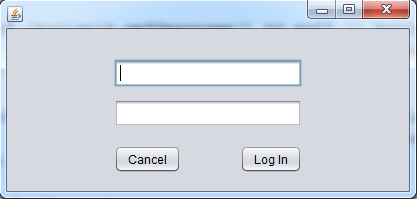
**Aktori SK**

*Korisnik*

**Učesnici SK**

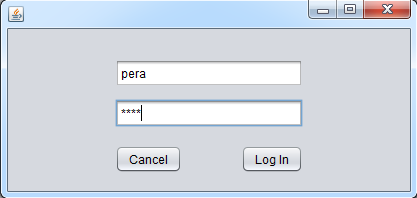
*Korisnik* *i sistem (progrаm)*

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

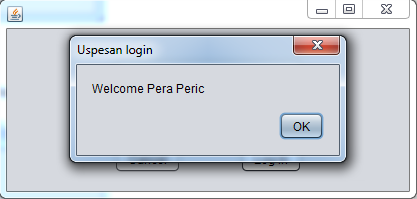


**Osnovni scenаrio SK**

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

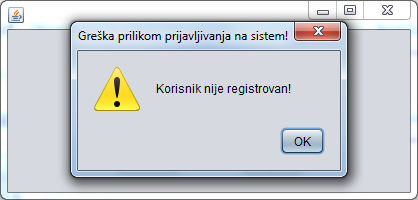


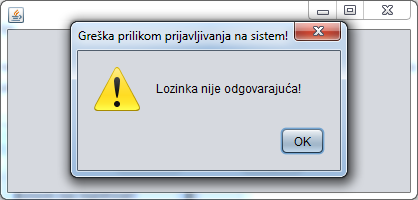
1. **Korisnik** pozivа *sistem* dа pronađe korisnika sa zadatim podacima. (APSO)
2. **Sistem** pretražuje korisnike. (SO)
3. **Sistem** prikаzuje *korisniku* poruku: ʺUspešno ste se prijavili ʺ. (IA)

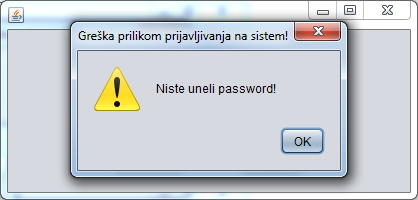


Alternаtivnа scenаrijа

4.1. Ukoliko sistem ne može da nađe korisnika prikazuje poruku “Sistem ne može da nađe korisnika na osnovu unetih vrednosti”. (IA)







### 3.2.1.10 SK10: Unos korisnika

**Nаziv** **SK**

Unos korisnika

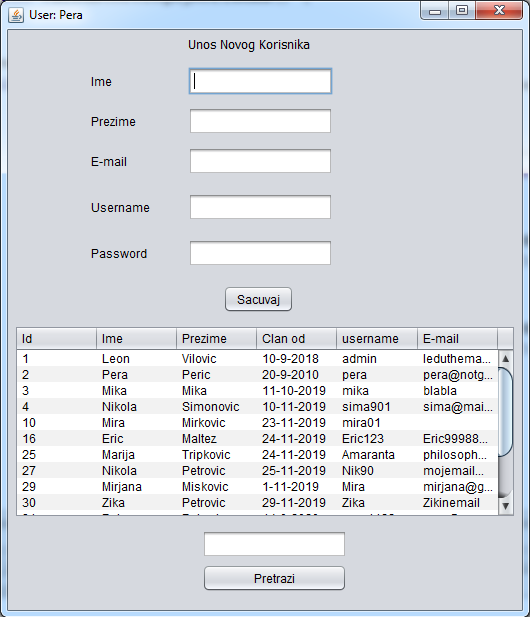
**Aktori SK**

*Korisnik*

**Učesnici SK**

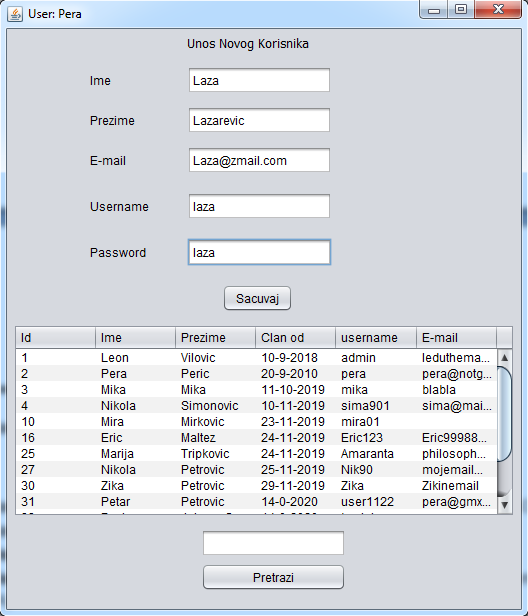
*Korisnik i sistem (progrаm)*

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

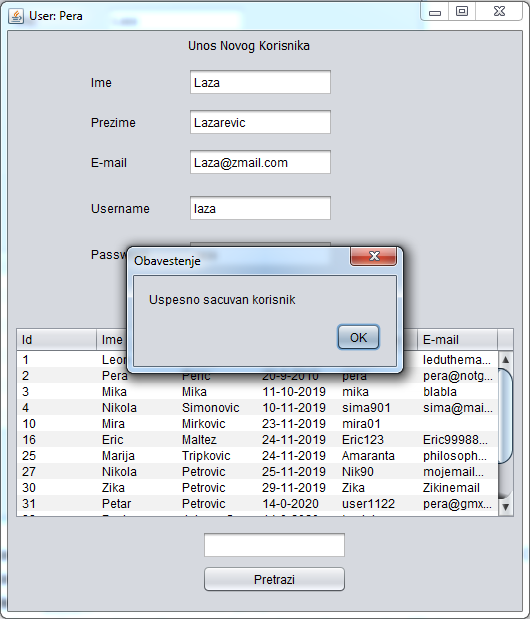
**

**Osnovni scenаrio SK**

1. **Korisnik** unosi podаtke o novom *korisniku*. (APUSO)

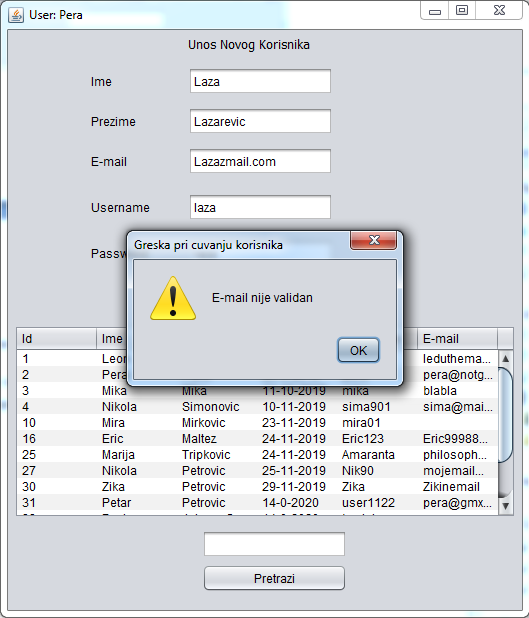


1. **Korisnik** kontroliše dа li je korektno uneo podаtke o novom *korisniku*. (ANSO)
2. **Korisnik** pozivа *sistem* dа zаpаmti podаtke o *korisniku*. (APSO)
3. **Sistem** pаmti podаtke o *korisniku*. (SO)
4. **Sistem** prikаzuje *korisniku* zаpаmćenog *korisnika* i poruku: ʺSistem je zаpаmtio korisnika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



Alternаtivnа scenаrijа

5.1 Ukoliko sistem ne može dа zаpаmti podаtke o novom *korisniku* on prikаzuje *korisniku* poruku “Sistem ne može dа zаpаmti novog korisnika”. (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 klientima

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 klasa “ 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() {

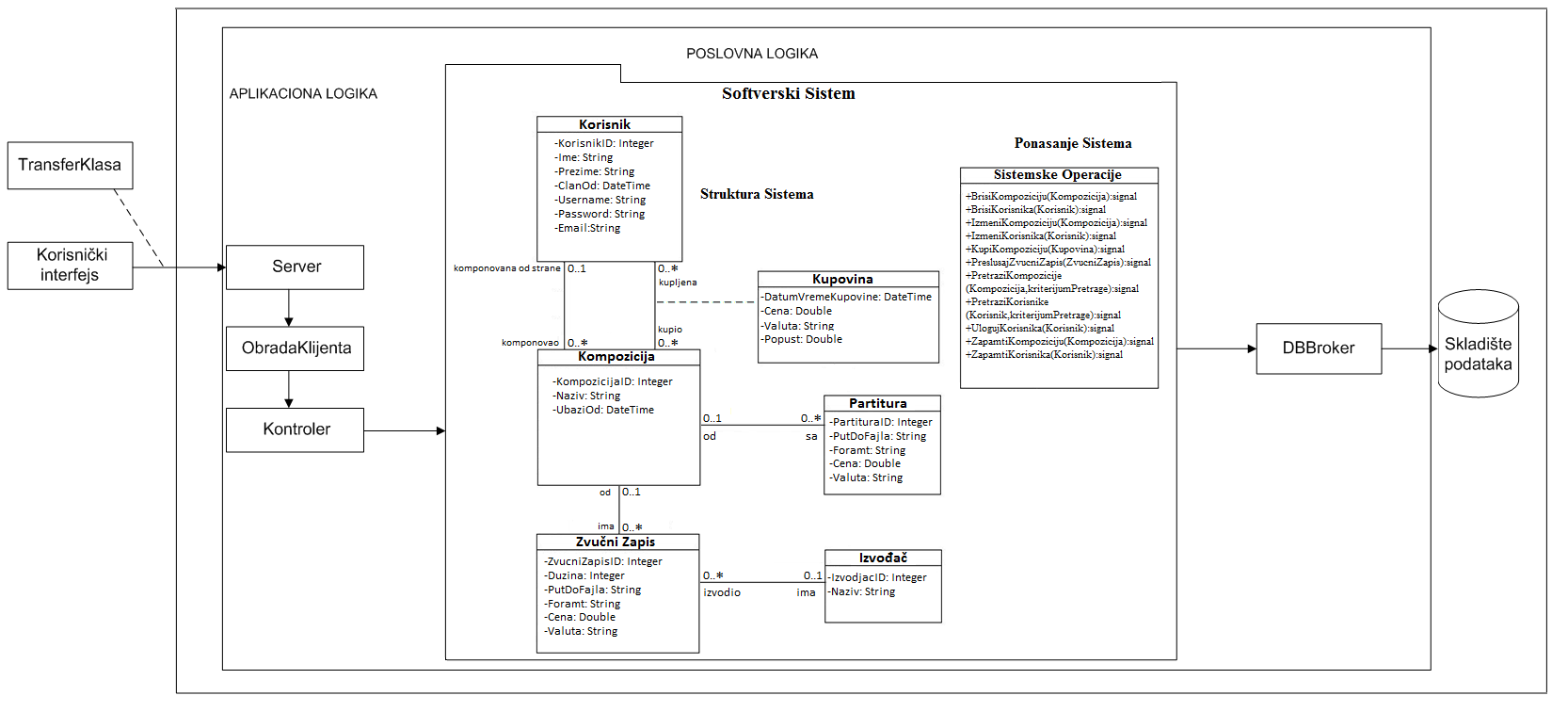
return "ResponseObject{" + "data=" + data + ", exception=" + exception + '}';}

}

### 3.3.2 Kontroler Aplikacione Logike

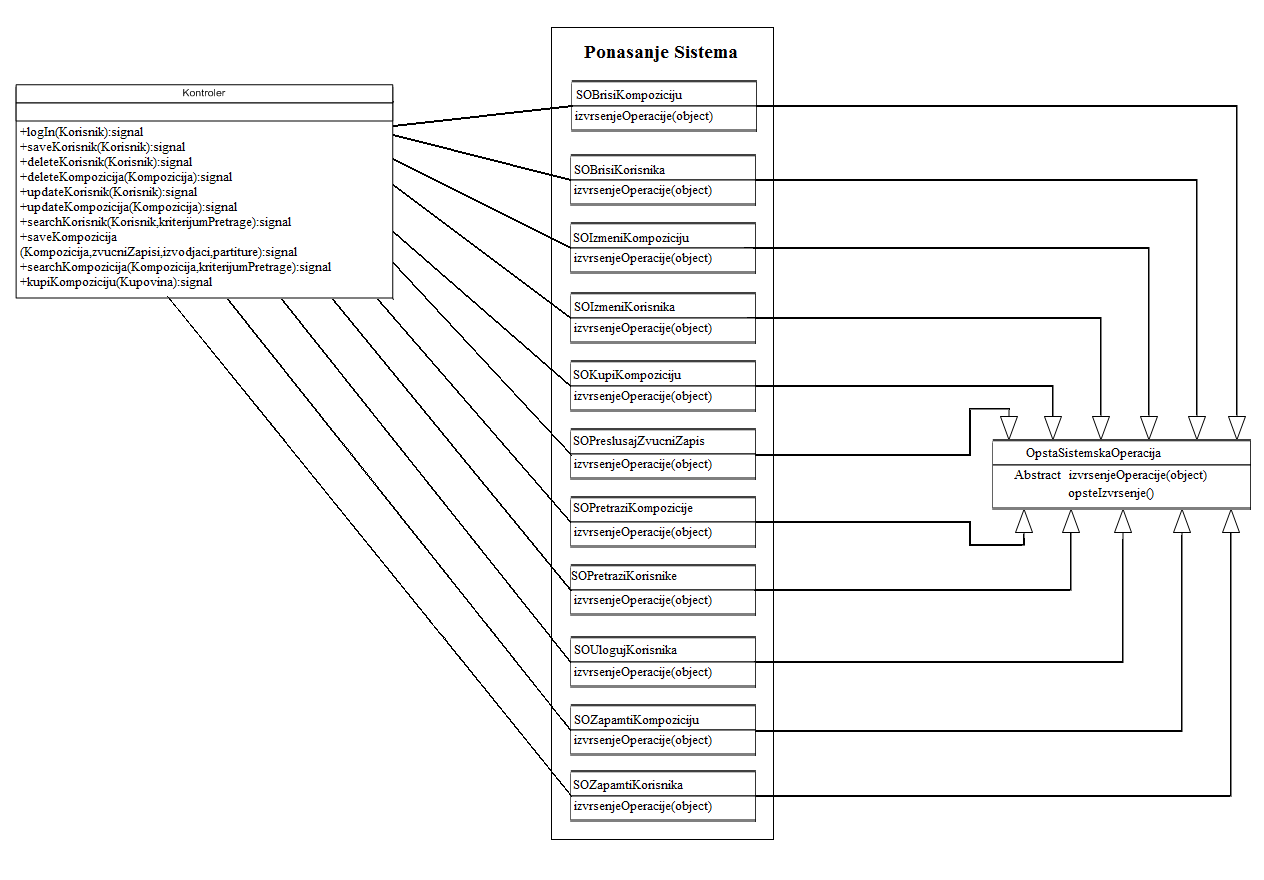
Kontroler aplikacione logike prihvata zahteve za izvršenje sistemskih operacija i iste prosleđuje do konkretne sistemske operacije. Nakon izvršenja sistemske operacije, kontroler prihvata odgovor i vraća nazad pozivaocu (niti klijenta).

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;

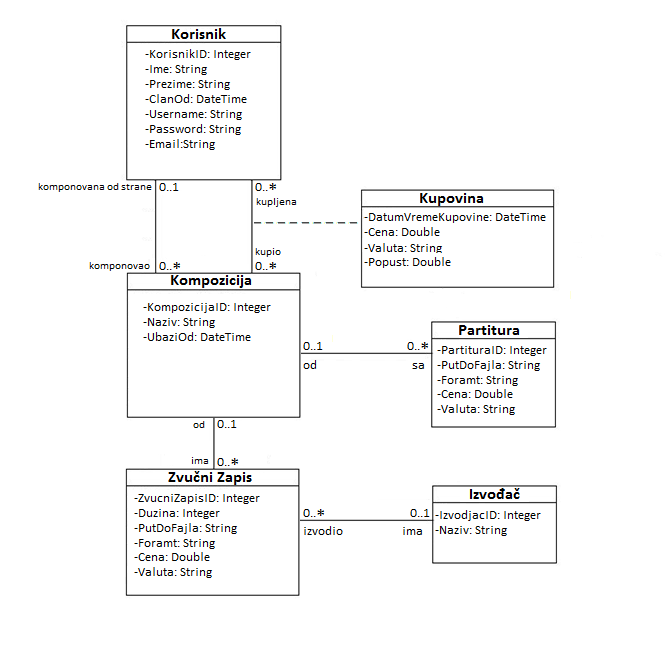
finally{

raskiniKonekciju();}}

public OpstiDomenskiObjekat getOdo(){

return odo;}}

### 3.3.3 Projektvovanje 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. Kao primer dajem klasu “ Korisnik ”:

public class Korisnik implements Serializable, domain.OpstiDomenskiObjekat {

private Long KorisnikID;

private String Ime;

private String Prezime;

private GregorianCalendar ClanOd;

private String Username;

private String Password;

private String Email;

public Long getKorisnikID() {

return KorisnikID;}

public void setKorisnikID(Long KorisnikID) {

this.KorisnikID = KorisnikID;}

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 GregorianCalendar getClanOd() {

return ClanOd;}

public void setClanOd(GregorianCalendar ClanOd) {

this.ClanOd = ClanOd;}

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 String getEmail() {

return Email;}

public void setEmail(String Email) {

this.Email = Email;}

public Korisnik(Long KorisnikID, String Ime, String Prezime, GregorianCalendar ClanOd, String Username, String Password, String Email) {

this.KorisnikID = KorisnikID;

this.Ime = Ime;

this.Prezime = Prezime;

this.ClanOd = ClanOd;

this.Username = Username;

this.Password = Password;

this.Email = Email;}

public Korisnik() {}

@Override

public String dajNazivTabele() {

return "korisnik";}

@Override

public String dajNaziveAtributa() {

return "Ime, Prezime, ClanOd, Username, Password, Email";}

@Override

public String dajNazivAtributa(int brojKolone) {

switch (brojKolone) {

case 1: return "KorisnikID";

case 2: return "Ime";

case 3: return "Prezime";

case 4: return "ClanOd";

case 5: return "Username";

case 6: return "Password";

case 7: return "Email";}

return "";}

@Override

public String dajVrednostiAtributa() {

return "'" + Ime + "', '" + Prezime + "', '" + ClanOd.get(Calendar.YEAR) + "-" + ClanOd.get(Calendar.MONTH) + "-" + ClanOd.get(Calendar.DAY\_OF\_MONTH) + "', '" + Username + "', '" + Password + "', '" + Email + "'";}

@Override

public boolean isAutoincrement() {

return true;}

@Override

public void setId(Object id) {

KorisnikID = (Long) id;}

@Override

public String dajNazivPrimarnogKljuca() {

return "KorisnikID"; }

@Override

public String getID() {

return KorisnikID.toString();}

@Override

public List<String> dajAributeZaPretragu() {

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

lista.add("Ime");

lista.add("Prezime");

lista.add("Username");

return lista; }

@Override

public String postaviVrednostiAtributa() {

return "Ime='" + Ime + "', Prezime='" + Prezime + "', Username='" + Username + "', Password='" + Password + "', Email='" + Email + "'";}

@Override

public List<String> dajKompleksniKljuc() {

throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.}

@Override

public String toString(){

return Ime+" "+Prezime;}

}

### 3.3.4 Projektvovanje strukture sofverskog sistema

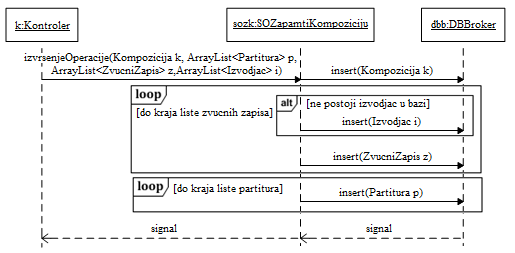
Preporuka je da se na samom početku projektovanja sistemskih operacija naprave konceptualne realizacije (rešenja) za svaku sistemsku operaciju tako da ona bude direktno povezana sa logikom problema. Aspekti realizacije koji se odnose na konekciju sa bazom, perzistentnost i transakcije treba izbeći kako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

#### 3.3.4.1 Ugovor UG1 Zapamti Kompoziciju

**Operacija:** ZapamtiKompoziciju(*Kompozicija*):signal;

**Preduslovi:** \

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

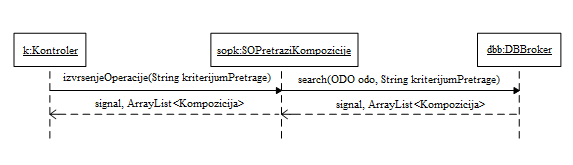


#### 3.3.4.2 Ugovor UG2 Pretrazi Kompozicije

**Operacija:** PretraziKompozicije(KriterijumPretrage,*List<Kompozicija>*):signal;

**Preduslovi:** \

**Postuslovi:** \

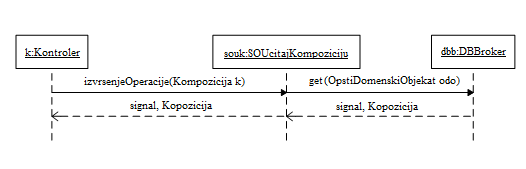


#### 3.3.4.3 Ugovor UG3 Ucitaj Kompoziciju

**Operacija:** UcitajKompoziciju(*Kompozicija*):signal;

**Preduslovi:** \

**Postuslovi:** \

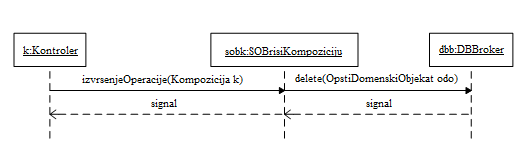


#### 3.3.4.4 Ugovor UG4 Obrisi Kompoziciju

**Operacija:** ObrisiKompoziciju (*Kompozicija*):signal;

**Preduslovi:** \

**Postuslovi:** Kompozicija je obrisana

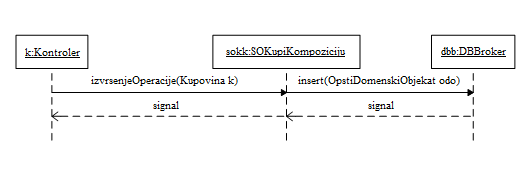


#### 3.3.4.5 Ugovor UG5 Kupi Kompoziciju

**Operacija:** KupiKompoziciju(*Kompozicija*):signal;

**Preduslovi:** \

**Postuslovi:** Kupovina je zapamćena

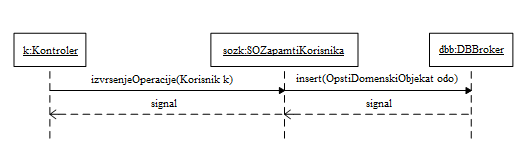


#### 3.3.4.6 Ugovor UG6 Zapamti Korisnika

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

**Preduslovi:** \

**Postuslovi:** Korisnik je zapamćen

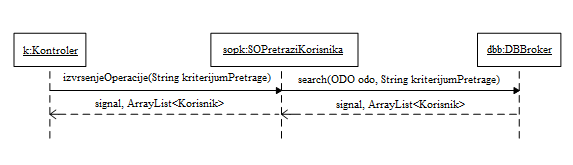


#### 3.3.4.7 Ugovor UG7 Pretrazi Korisnike

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

**Preduslovi:** \

**Postuslovi:** \

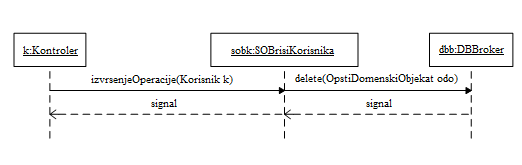


#### 3.3.4.8 Ugovor UG8 Obrisi Korisnika

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

**Preduslovi:** Odabrani korisnik postoji u bazi podataka

**Postuslovi:** Korisnik je obrisan

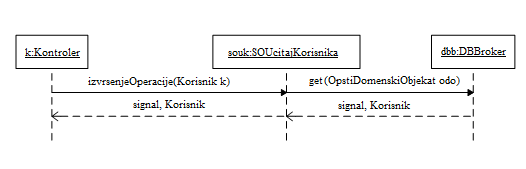


#### 3.3.4.9 Ugovor UG9Ucitaj Korisnika

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

**Preduslovi:** \

**Postuslovi:** \

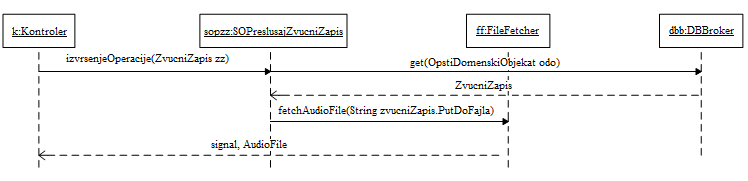


#### 3.3.4.10 Ugovor UG10 Preslusaj Zvucni Zapis

**Operacija:** PreslusajZvucniZapis(*ZvucniZapis*):signal;

**Preduslovi:** Audio fajl postoji na označenom direktorijumu

**Postuslovi:** Audio fajl je pusten



### 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 update(OpstiDomenskiObjekat o)
8. public List<OpstiDomenskiObjekat> classConverter(ResultSet rs)
9. public OpstiDomenskiObjekat getODOUsingPK(OpstiDomenskiObjekat o, Long PK)
10. public OpstiDomenskiObjekat getODOUsingComplexkey(OpstiDomenskiObjekat o, List<Long> complexKey)
11. public List<OpstiDomenskiObjekat> search(OpstiDomenskiObjekat o, String kriterijumPretrage)
12. public List<OpstiDomenskiObjekat> searchKompozicijaComplex(OpstiDomenskiObjekat o, String kriterijumPretrage)
13. public OpstiDomenskiObjekat logIn(OpstiDomenskiObjekat o)

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);

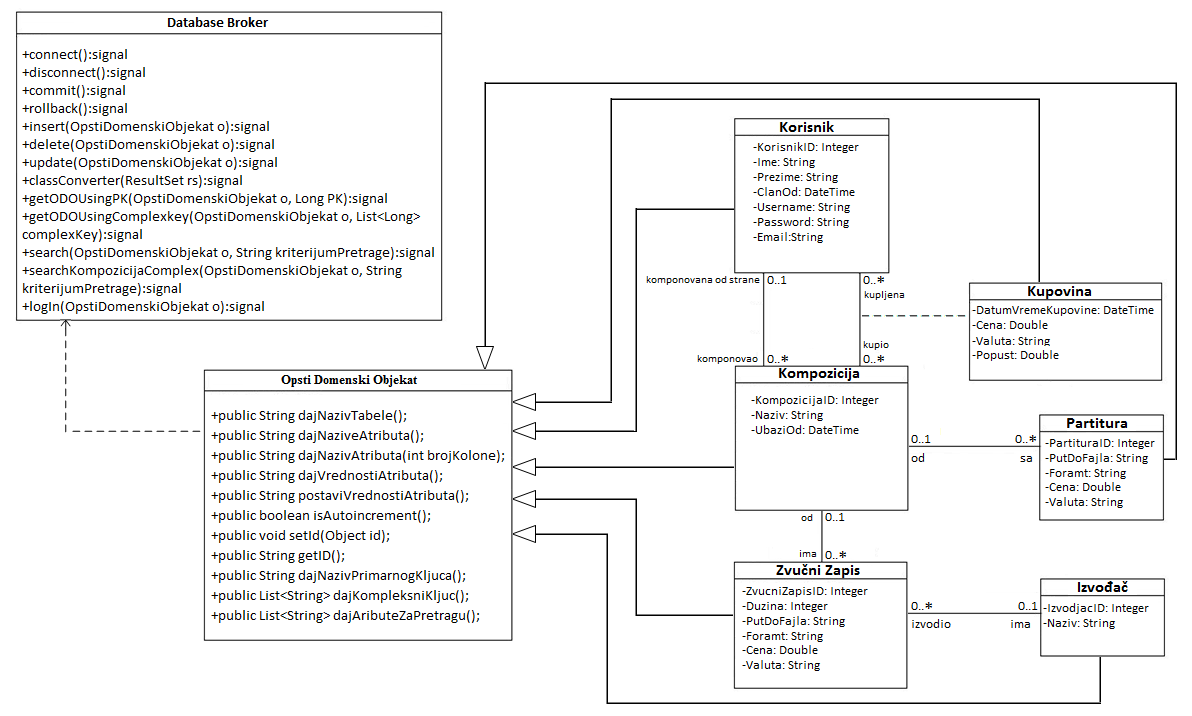
public String getID();

public String dajNazivPrimarnogKljuca();

public List<String> dajKompleksniKljuc();

public List<String> dajAributeZaPretragu();

}



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: Izvodjac** | | |
| Ime polja | Tip | Veličina |
| IzvodjacID | INT | 11 |
| Naziv | VARCHAR | 45 |

|  |  |  |
| --- | --- | --- |
| **Tabela: Kompozicija** | | |
| Ime polja | Tip | Veličina |
| KompozicijaID | INT | 11 |
| Naziv | VARCHAR | 45 |
| UBaziOd | DATETIME |  |
| KorisnikID | INT | 11 |

|  |  |  |
| --- | --- | --- |
| **Tabela: Korisnik** | | |
| Ime polja | Tip | Veličina |
| KorisnikID | INT | 11 |
| Ime | VARCHAR | 45 |
| Prezime | VARCHAR | 45 |
| ClanOd | DATETIME |  |
| Username | VARCHAR | 45 |
| Password | VARCHAR | 45 |
| Email | VARCHAR | 45 |

|  |  |  |
| --- | --- | --- |
| **Tabela: Kupovina** | | |
| Ime polja | Tip | Veličina |
| KorisnikID | INT | 11 |
| KompozicijaID | INT | 11 |
| DatumVremeKupovine | DATETIME |  |
| Cena | DOUBLE |  |
| Valuta | VARCHAR | 45 |
| Popust | DOUBLE |  |

|  |  |  |
| --- | --- | --- |
| **Tabela: Partitura** | | |
| Ime polja | Tip | Veličina |
| PartituraID | INT | 11 |
| PutDoFajla | VARCHAR | 45 |
| Format | VARCHAR | 10 |
| Cena | DOUBLE |  |
| Valuta | VARCHAR | 45 |
| KompozicijaID | INT | 11 |

|  |  |  |
| --- | --- | --- |
| **Tabela: Zvucni zapis** | | |
| Ime polja | Tip | Veličina |
| ZvucniZapisID | INT | 11 |
| Duzina | INT | 11 |
| PutDoFajla | VARCHAR | 200 |
| Format | VARCHAR | 45 |
| Cena | DOUBLE |  |
| Valuta | VARCHAR | 45 |
| KompozicijaID | INT | 11 |
| IzvodjacID | INT | 11 |

# 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 ravojno okruženje “NetBeans IDE 8.2”. Na osnovu arhitekture softverskog sistema dobijene su sledeće softverske klase:

**Client:**

* ui/view/KompozicijaDetailsPanel.java
* ui/view/KompozicijaFrame.java
* ui/view/LogInFrame.java
* ui/view/MainFrame.java
* ui/view/UserDetailsPanel.java
* ui/view/UserFrame.java
* ui/view/components/TableModelKorisnik.java
* ui/view/components/TableModelPartitura.java
* ui/view/components/TableModelSearchKompozicja.java
* ui/view/components/TableModelZvucniZapis.java
* communication/CommunicationController.java
* threads/AudioPlayerThread.java

**Server:**

* controller/Controller.java
* database/DatabaseBroker.java
* database/connection/ConnectionFactory.java
* files/FileFetcher.java
* so/OpstaSistemskaOperacija.java
* so/SOBrisiKompoziciju.java
* so/SOBrisiKorisnika.java
* so/SOIzmeniKompoziciju.java
* so/SOIzmeniKorisnika.java
* so/SOKupiKompoziciju.java
* so/SOPreslusajZvucniZapis.java
* so/SOPretraziKompozicije.java
* so/SOPretraziKorisnike.java
* so/SOUcitajKorisnika.java
* so/SOUlogujKorisnika.java
* so/SOZapamtiKompoziciju.java
* so/SOZapamtiKorisnika.java
* threads/ClientHandlerThread.java
* threads/ServerThread.java
* ui/DatabaseConfigurationJPanel.java
* ui/MainFrame.java
* validator/Validator.java
* validator/impl/ValidatorKupovinaKompozicije.java
* validator/impl/ValidatorUlogujKorisnka.java
* validator/impl/ValidatorZapamtiKompoziciju.java
* validator/impl/ValidatorZapamtiKorisnika.java

**CommonLib:**

* domain/Izvodjac.java
* domain/Kompozicija.java
* domain/Korisnik.java
* domain/Kupovina.java
* domain/OpstiDomenskiObjekat.java
* domain/Partitura.java
* domain/ZvucniZapis.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.