Univerzitet u Beogradu

Fakultet organizacionih nauka

Laboratorija za softversko inženjerstvo

Seminarski rad iz predmeta

Projektovanje softvera

Tema: Softverski sistem za praćenje rada bioskopa

|  |  |
| --- | --- |
| Profesor:  dr Siniša Vlajić | Student:  Nikolina Momčilović 130/12 |

Beograd, 2016.

Sadržaj

[1. Korisnički zahtevi 3](#_Toc447274827)

[1.1. Verbalni opis 3](#_Toc447274828)

[1.2. Slučajevi korišćenja 4](#_Toc447274829)

[SK 1: Slučaj korišćenja – Prijavljivanje radnika 5](#_Toc447274830)

[SK 2: Slučaj korišćenja – Postavljanje novog filma na repertoar 6](#_Toc447274831)

[SK 3: Slučaj korišćenja – Pretraživanje filmova 7](#_Toc447274832)

[SK 4: Slučaj korišćenja – Izmena podataka o filmu 8](#_Toc447274833)

[SK 5: Slučaj korišćenja – Brisanje filma sa repertoara 9](#_Toc447274834)

[SK 6: Slučaj korišćenja – Rezervacija karata 10](#_Toc447274835)

[SK 7: Slučaj korišćenja – Pretraživanje rezervacija 11](#_Toc447274836)

[SK 8: Slučaj korišćenja – Brisanje rezervacije 12](#_Toc447274837)

[SK 9: Slučaj korišćenja – Prodaja karata 13](#_Toc447274838)

[2. Analiza 14](#_Toc447274839)

[2.1. Ponašanje softverskog sistema – Sistemski dijagram sekvenci 14](#_Toc447274840)

[DS 1: Dijagram sekvenci slučaja korišćenja – Prijavljivanje radnika 14](#_Toc447274841)

[DS 2: Dijagram sekvenci slučaja korišćenja – Postavljanje novog filma na repertoar 15](#_Toc447274842)

[DS 3: Dijagram sekvenci slučaja korišćenja – Pretraživanje filmova 17](#_Toc447274843)

[DS 4: Dijagram sekvenci slučaja korišćenja – Izmena podataka o filmu 19](#_Toc447274844)

[DS 5: Dijagram sekvenci slučaja korišćenja – Brisanje filma sa repertoara 22](#_Toc447274845)

[DS 6: Dijagram sekvenci slučaja korišćenja – Rezervacija karata 24](#_Toc447274846)

[DS 7: Dijagram sekvenci slučaja korišćenja – Pretraživanje rezervacija 26](#_Toc447274847)

[DS 8: Dijagram sekvenci slučaja korišćenja – Brisanje rezervacije 28](#_Toc447274848)

[DS 9: Dijagram sekvenci slučaja korišćenja – Prodaja karata 30](#_Toc447274849)

[2.2. Ponašanje softverskog sistema – Definisanje ugovora o sistemskim operacijama 32](#_Toc447274850)

[2.3. Struktura softverskog sistema – Konceptualni (domenski) model 36](#_Toc447274851)

[2.4. Struktura softverskog sistema – Relacioni model 37](#_Toc447274852)

# Korisnički zahtevi

## Verbalni opis

Potrebno je napraviti aplikaciju koja omogućava prijavljivanje radnika, vođenje evidencije o filmovima koji su trenutno na repertoaru i rezervacijama za određene filmove, kao i mogućnost za kupovinu karata.

Vođenje evidencije o filmovima koji su trenutno na repertoaru podrazumeva dodavanje novih filmova na repertoar, uklanjanje filmova sa repertoara, izmenu podataka o već dodatim filmovima, kao i pretragu filmova koji postoje na repertoaru.

Prilikom dodavanja novih filmova na repertoar potrebno je voditi računa o tome u kojim salama će se film prikazivati, i da li su konkretne sale slobodne u terminu u kom želimo da ih zakupimo.

Takođe, u aplikaciji je moguće izvršiti i rezervaciju karata. Zatim aplikacija omogućava pregled i pretragu tih rezervacija, kao i brisanje određene rezervacije.

Pomoću aplikacije moguća je kupovina karata koje su vec prethodno rezervisane, ali i karata koje prethodno nisu bile rezervisane, ako postoji slobodnog mesta u konkretnom terminu.

Da bi radnik mogao da vrši izmene nad podacima u sistemu, potrebno je da bude prijavljen na sistem.

## Slučajevi korišćenja

U konkretnom slučaju identifikovani su sledeći slučajevi korišćenja koji su prikazani i na slici 1:

1. Prijavljivanje radnika
2. Postavljanje novog filma na repertoar
3. Pretraživanje filmova
4. Izmena podataka o filmu
5. Brisanje filma sa repertoara
6. Rezervacija karata
7. Pretraživanje rezervacija
8. Brisanje rezervacije
9. Prodaja karata



Slika 1. Dijagram slučajeva korišćenja

### SK 1: Slučaj korišćenja – Prijavljivanje radnika

**Naziv SK**

Prijavljivanje radnika

**Aktori SK**

**Radnik**

**Učesnici SK**

**Radnik** i **sistem** (program)

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

**Osnovni scenario SK**

1. **Radnik** unosi podatke za autentifikaciju radnika. (APUSO)
2. **Radnik** poziva **sistem** da pronađe radnika sa zadatim podacima. (APSO)
3. **Sistem** pretražuje radnike. (SO)
4. **Sistem** prikazuje poruku da je radnik uspešno prijavljen i početnu formu. (IA)

**Alternativna scenarija**

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

### SK 2: Slučaj korišćenja – Postavljanje novog filma na repertoar

**Naziv SK**

Postavljanje novog filma na repertoar

**Aktori SK**

**Radnik**

**Učesnici SK**

**Radnik** i **sistem** (program)

**Preduslov:** **Sistem** je uključen i **radnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje početnu formu i tab *repertoar*.

**Osnovni scenario SK**

1. **Radnik** poziva **sistem** da kreira *novi film*. (APSO)
2. **Sistem** kreira *novi film*. (SO)
3. **Sistem** prikazuje **radniku** *novi film*. (IA)
4. **Radnik** unosi podatke o *novom filmu*. (APUSO)
5. **Radnik** kontroliše da li je korektno uneo podatke o *novom filmu*. (ANSO)
6. **Radnik** poziva **sistem** da zapamti podatke o *novom filmu*. (APSO)
7. **Sistem** pamti podatke o novom filmu. (SO)
8. **Sistem** prikazuje **radniku** poruku: “**Sistem** je zapamtio *film*”. (IA)

**Alternativna scenarija**

3.1. Ukoliko **sistem** ne može da kreira *novi film* on prikazuje **radniku** poruku: “**Sistem** ne može da kreira *novi film*”. Prekida se izvršavanje scenarija. (IA)

8.1. Ukoliko **sistem** ne može da zapamti podatke o *novom* *filmu* on prikazuje **radniku** poruku “**Sistem** ne može da zapamti *novi film*”. (IA)

### SK 3: Slučaj korišćenja – Pretraživanje filmova

**Naziv SK**

Pretraživanje filmova

**Aktori SK**

**Radnik**

**Učesnici SK**

**Radnik** i **sistem** (program)

**Preduslov:** **Sistem** je uključen i **radnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje početnu formu i tab *repertoar*.

**Osnovni scenario SK**

1. **Radnik** unosi vrednosti po kojim pretražuje *filmove*. (APUSO)
2. **Radnik** poziva **sistem** da nađe *filmove* po zadatim vrednostima. (APSO)
3. **Sistem** traži *filmove* po zadatim vrednostima. (SO)
4. **Sistem** prikazuje **radniku** nađene *filmove*. (IA)
5. **Radnik** bira *film* čije podatke želi da vidi. (APUSO)
6. **Radnik** poziva **sistem** da prikaže podatke o izabranom *filmu*. (APSO)
7. **Sistem** pronalazi podatke o izabranom *filmu*. (SO)
8. **Sistem** prikazuje **radniku** podatke o izabranom *filmu*. (IA)

**Alternativna scenarija**

4.1. Ukoliko **sistem** ne može da nađe *filmove* on prikazuje **radniku** poruku: “**Sistem** ne može da nađe *filmove* po zadatim vrednostima”. Prekida se izvršenje scenarija. (IA)

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

### SK 4: Slučaj korišćenja – Izmena podataka o filmu

**Naziv SK**

Izmena podataka o filmu

**Aktori SK**

**Radnik**

**Učesnici SK**

**Radnik** i **sistem** (program)

**Preduslov:** **Sistem** je uključen i **radnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje početnu formu i tab *repertoar*.

**Osnovni scenario SK**

1. **Radnik** unosi vrednosti po kojima pretražuje *filmove*. (APUSO)
2. **Radnik** poziva **sistem** da nađe *filmove* po zadatim vrednostima. (APSO)
3. **Sistem** traži *filmove* po zadatim vrednostima. (SO)
4. **Sistem** prikazuje **radniku** nađene *filmove*. (IA)
5. **Radnik** bira *film* koji želi da izmeni. (APUSO)
6. **Radnik** poziva **sistem** da učita podatke o izabranom *filmu*. (APSO)
7. **Sistem** pronalazi podatke o izabranom *filmu*. (SO)
8. **Sistem** prikazuje **radniku** podatke o izabranom *filmu*. (IA)
9. **Radnik** unosi (menja) podatke o *filmu*. (APUSO)
10. **Radnik** kontroliše da li je korektno uneo podatke o *filmu*. (ANSO)
11. **Radnik** poziva **sistem** da zapamti podatke o *filmu*. (APSO)
12. **Sistem** pamti podatke o *filmu*. (SO)
13. **Sistem** prikazuje **radniku** poruku: “**Sistem** je zapamtio *film*.” (IA)

**Alternativna scenarija:**

4.1. Ukoliko **sistem** ne može da nađe *filmove* on prikazuje **radniku** poruku: “**Sistem** ne može da nađe *filmove* po zadatim vrednostima”. Prekida se izvršenje scenarija. (IA)

8.1. Ukoliko **sistem** ne može da nađe *film* on prikazuje **radniku** poruku: “**Sistem** ne može da nađe izabran *film*”. Prekida se izvršenje scenarija. (IA)

13.1. Ukoliko **sistem** ne može da zapamti podatke o *filmu* on prikazuje **radniku** poruku: “**Sistem** ne može da zapamti *film*”. (IA)

### SK 5: Slučaj korišćenja – Brisanje filma sa repertoara

**Naziv SK**

Brisanje filma sa repertoara

**Aktori SK**

**Radnik**

**Učesnici SK**

**Radnik** i **sistem** (program)

**Preduslov:** **Sistem** je uključen i **radnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje početnu formu i tab *repertoar*.

**Osnovni scenario SK**

1. **Radnik** unosi vrednosti po kojim pretražuje *filmove*. (APUSO)
2. **Radnik** poziva **sistem** da nađe *filmove* po zadatim vrednostima. (APSO)
3. **Sistem** traži *filmove* po zadatim vrednostima. (SO)
4. **Sistem** prikazuje **radniku** nađene *filmove*. (IA)
5. **Radnik** bira *film* kojeg želi da obriše. (APUSO)
6. **Radnik** poziva **sistem** da obriše *film*. (APSO)
7. **Sistem** briše *film*. (SO)
8. **Sistem** prikazuje **radniku** poruku: “**Sistem** je obrisao *film*.” (IA)

**Alternativna scenarija**

4.1. Ukoliko **sistem** ne može da nađe *filmove* on prikazuje **radniku** poruku: “**Sistem** ne može da nađe *filmove* po zadatim vrednostima”. Prekida se izvršenje scenarija. (IA)

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

### SK 6: Slučaj korišćenja – Rezervacija karata

**Naziv SK**

Rezervacija karata

**Aktori SK**

**Radnik**

**Učesnici SK**

**Radnik** i **sistem** (program)

**Preduslov:** **Sistem** je uključen i **radnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje početnu formu i tab *repertoar*.

**Osnovni scenario SK**

1. **Radnik** poziva **sistem** da kreira *novu rezervaciju*. (APSO)
2. **Sistem** kreira *novu rezervaciju*. (SO)
3. **Sistem** prikazuje **radniku** *novu* *rezervaciju*. (IA)
4. **Radnik** unosi podatke o *novoj rezervaciji*. (APUSO)
5. **Radnik** kontroliše da li je korektno uneo podatke o *novoj rezervaciji*. (ANSO)
6. **Radnik** poziva **sistem** da zapamti podatke o *novoj rezervaciji*. (APSO)
7. **Sistem** pamti podatke o *novoj rezervaciji*. (SO)
8. **Sistem** prikazuje **radniku** poruku:”**Sistem** je zapamtio *rezervaciju*”. (IA)

**Alternativna scenarija**

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

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

### SK 7: Slučaj korišćenja – Pretraživanje rezervacija

**Naziv SK**

Pretraživanje rezervacija

**Aktori SK**

**Radnik**

**Učesnici SK**

**Radnik** i **sistem** (program)

**Preduslov:** **Sistem** je uključen i **radnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje početnu formu i tab *rezervacije*.

**Osnovni scenario SK**

1. **Radnik** unosi vrednosti po kojima pretražuje *rezervacije*. (APUSO)
2. **Radnik** poziva **sistem** da nađe *rezervacije* po zadatim vrednostima. (APSO)
3. **Sistem** traži *rezervacije* po zadatim vrednostima. (SO)
4. **Sistem** prikazuje **radniku** podatke o *rezervacijama*. (IA)
5. **Radnik** bira *rezervaciju* za koju želi da vidi podatke. (APUSO)
6. **Radnik** poziva **sistem** da prikaže podatke o izabranoj *rezervaciji*. (APSO)
7. **Sistem** traži podatke o izabranoj *rezervaciji*. (SO)
8. **Sistem** prikazuje **radniku** podatke o izabranoj *rezervaciji*. (IA)

**Alternativna scenarija**

4.1. Ukoliko **sistem** ne može da nađe *rezervaciju* on prikazuje **radniku** poruku: “**Sistem** ne može da nađe *rezervaciju* po zadatim vrednostima”. Prekida se izvršenje scenarija. (IA)

8.1. Ukoliko **sistem** ne može da nađe izabranu *rezervaciju* on prikazuje **radniku** poruku: “**Sistem** ne može da nađe podatke o izabranoj *rezervaciji*”. (IA)

### SK 8: Slučaj korišćenja – Brisanje rezervacije

**Naziv SK**

Brisanje rezervacije

**Aktori SK**

**Radnik**

**Učesnici SK**

**Radnik** i **sistem** (program)

**Preduslov:** **Sistem** je uključen i **radnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje početnu formu i tab *rezervacije*.

**Osnovni scenario SK**

1. **Radnik** unosi vrednosti po kojim pretražuje *rezervacije*. (APUSO)
2. **Radnik** poziva **sistem** da nađe *rezervacije* po zadatim vrednostima. (APSO)
3. **Sistem** traži *rezervacije* po zadatim vrednostima. (SO)
4. **Sistem** prikazuje **radniku** nađene *rezervacije*. (IA)
5. **Radnik** bira *rezervaciju* koju želi da obriše. (APUSO)
6. **Radnik** poziva **sistem** da obriše *rezervaciju*. (APSO)
7. **Sistem** briše *rezervaciju*. (SO)
8. **Sistem** prikazuje **radniku** poruku: “**Sistem** je obrisao *rezervaciju*.” (IA)

**Alternativna scenarija**

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

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

### SK 9: Slučaj korišćenja – Prodaja karata

**Naziv SK**

Prodaja karata

**Aktori SK**

**Radnik**

**Učesnici SK**

**Radnik** i **sistem** (program)

**Preduslov:** **Sistem** je uključen i **radnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje početnu formu i tab *rezervacije* ili tab *repertoar*.

**Osnovni scenario SK**

1. **Radnik** poziva **sistem** da kreira *nove karte*. (APSO)
2. **Sistem** kreira *nove karte*. (SO)
3. **Sistem** prikazuje **radniku** *nove karte*. (IA)
4. **Radnik** unosi podatke o *novim kartama*. (APUSO)
5. **Radnik** kontroliše da li je korektno uneo podatke o *novim kartama*. (ANSO)
6. **Radnik** poziva **sistem** da zapamti podatke o *novim kartama*. (APSO)
7. **Sistem** pamti podatke o *novim kartama*. (SO)
8. **Sistem** prikazuje **radniku** poruku: “**Sistem** je zapamtio *karte*”. (IA)

**Alternativna scenarija**

3.1. Ukoliko **sistem** ne može da kreira *nove karte* on prikazuje **radniku** poruku: “**Sistem** ne može da kreira *nove karte*”. Prekida se izvršavanje scenarija. (IA)

8.1. Ukoliko **sistem** ne može da zapamti podatke o *novim kartama* on prikazuje **radniku** poruku “**Sistem** ne može da zapamti *nove karte*”. (IA)

# Analiza

## Ponašanje softverskog sistema – Sistemski dijagram sekvenci

### DS 1: Dijagram sekvenci slučaja korišćenja – Prijavljivanje radnika

1. **Radnik** poziva **sistem** da pronađe radnika sa zadatim podacima. (APSO)
2. **Sistem** prikazuje poruku da je radnik uspešno prijavljen i početnu formu. (IA)



**Alternativna scenarija**

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



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

1. *signal* **NadjiRadnika**(*Radnik*)

### DS 2: Dijagram sekvenci slučaja korišćenja – Postavljanje novog filma na repertoar

1. **Forma** poziva **sistem** da prikaže podatke o *radnicima.*
2. **Sistem** prikazuje **formi** podatke o *radnicima.*
3. **Radnik** poziva **sistem** da kreira *novi film*. (APSO)
4. **Sistem** prikazuje **radniku** *novi film*. (IA)
5. **Radnik** poziva **sistem** da zapamti podatke o *novom filmu*. (APSO)
6. **Sistem** prikazuje **radniku** poruku: “**Sistem** je zapamtio *film*”. (IA)



**Alternativna scenarija**

2.1. Ukoliko **sistem** ne može da nađe listu *radnika* on prikazuje **formi** poruku: “**Sistem** ne može da nađe listu *radnika*”.



4.1. Ukoliko **sistem** ne može da kreira *novi film* on prikazuje **radniku** poruku: “**Sistem** ne može da kreira *novi film*”. Prekida se izvršavanje scenarija. (IA)



6.1. Ukoliko **sistem** ne može da zapamti podatke o *novom* *filmu* on prikazuje **radniku** poruku “**Sistem** ne može da zapamti *novi film*”. (IA)



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

1. *signal* **UcitajListuRadnika**(*List<Radnik>*)
2. *signal* **KreirajNoviFilm**(*Film*)
3. *signal* **ZapamtiFilm**(*Film*)

### DS 3: Dijagram sekvenci slučaja korišćenja – Pretraživanje filmova

1. **Radnik** poziva **sistem** da nađe *filmove* po zadatim vrednostima. (APSO)
2. **Sistem** prikazuje **radniku** nađene *filmove*. (IA)
3. **Radnik** poziva **sistem** da prikaže podatke o izabranom *filmu*. (APSO)
4. **Sistem** prikazuje **radniku** podatke o izabranom *filmu*. (IA)



**Alternativna scenarija**

2.1. Ukoliko **sistem** ne može da nađe *filmove* on prikazuje **radniku** poruku: “**Sistem** ne može da nađe *filmove* po zadatim vrednostima”. Prekida se izvršenje scenarija. (IA)



4.1. Ukoliko **sistem** ne može da nađe izabrani *film* on prikazuje **radniku** poruku: “**Sistem** ne može da nađe podatke o izabranom *filmu*”. (IA)



Sa navedenih sekvencnih dijagrama uočavaju se 2 sistemske operacije koje treba projektovati:

1. *signal* **PretraziFilmove**(*KriterijumPretrage, List<Film>*)
2. *signal* **UcitajFilm**(*Film*)

### DS 4: Dijagram sekvenci slučaja korišćenja – Izmena podataka o filmu

1. **Radnik** poziva **sistem** da nađe *filmove* po zadatim vrednostima. (APSO)
2. **Sistem** prikazuje **radniku** nađene *filmove*. (IA)
3. **Radnik** poziva **sistem** da učita podatke o izabranom *filmu*. (APSO)
4. **Sistem** prikazuje **radniku** podatke o izabranom *filmu*. (IA)
5. **Radnik** poziva **sistem** da zapamti podatke o *filmu*. (APSO)
6. **Sistem** prikazuje **radniku** poruku: “**Sistem** je zapamtio *film*.” (IA)



**Alternativna scenarija:**

2.1. Ukoliko **sistem** ne može da nađe *filmove* on prikazuje **radniku** poruku: “**Sistem** ne može da nađe *filmove* po zadatim vrednostima”. Prekida se izvršenje scenarija. (IA)



4.1. Ukoliko **sistem** ne može da nađe *film* on prikazuje **radniku** poruku: “**Sistem** ne može da nađe izabran *film*”. Prekida se izvršenje scenarija. (IA)



6.1. Ukoliko **sistem** ne može da zapamti podatke o *filmu* on prikazuje **radniku** poruku: “**Sistem** ne može da zapamti *film*”. (IA)



Sa navedenih sekvencnih dijagrama uočavaju se 3 sistemske operacije koje treba projektovati:

1. *signal* **PretraziFilmove**(*KriterijumPretrage, List<Film>*)
2. *signal* **UcitajFilmove**(*Film*)
3. *signal* **ZapamtiFilm**(*Film*)

### DS 5: Dijagram sekvenci slučaja korišćenja – Brisanje filma sa repertoara

1. **Radnik** poziva **sistem** da nađe *filmove* po zadatim vrednostima. (APSO)
2. **Sistem** prikazuje **radniku** nađene *filmove*. (IA)
3. **Radnik** poziva **sistem** da obriše *film*. (APSO)
4. **Sistem** prikazuje **radniku** poruku: “**Sistem** je obrisao *film*.” (IA)



**Alternativna scenarija**

2.1. Ukoliko **sistem** ne može da nađe *filmove* on prikazuje **radniku** poruku: “**Sistem** ne može da nađe *filmove* po zadatim vrednostima”. Prekida se izvršenje scenarija. (IA)



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



Sa navedenih sekvencnih dijagrama uočavaju se 2 sistemske operacije koje treba projektovati:

1. *signal* **PretraziFilmove**(*KriterijumPretrage, List<Film>*)
2. *signal* **ObrisiFilm**(*Film*)

### DS 6: Dijagram sekvenci slučaja korišćenja – Rezervacija karata

1. **Forma** poziva **sistem** da prikaže podatke o *radnicima.*
2. **Sistem** prikazuje **formi** podatke o *radnicima.*
3. **Forma** poziva **sistem** da prikaže podatke o *projekcijama.*
4. **Sistem** prikazuje **formi** podatke o *projekcijama.*
5. **Radnik** poziva **sistem** da kreira *novu rezervaciju*. (APSO)
6. **Sistem** prikazuje **radniku** *novu rezervaciju*. (IA)
7. **Radnik** poziva **sistem** da zapamti podatke o *novoj rezervaciji*. (APSO)
8. **Sistem** prikazuje **radniku** poruku:”**Sistem** je zapamtio *rezervaciju*”. (IA)



**Alternativna scenarija**

2.1. Ukoliko **sistem** ne može da nađe listu *radnika* on prikazuje **formi** poruku: “**Sistem** ne može da nađe listu *radnika*”.



4.1. Ukoliko **sistem** ne može da nađe listu *projekcija* on prikazuje **formi** poruku: “**Sistem** ne može da nađe listu *projekcija*”.



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



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



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

1. *signal* **UcitajListuRadnika**(*List<Radnik>*)
2. *signal* **UcitajListuProjekcija**(*List<Projekcija>*)
3. *signal* **KreirajNovuRezervaciju** (*Rezervacija*)
4. *signal* **ZapamtiRezervaciju**(*Rezervacija*)

### DS 7: Dijagram sekvenci slučaja korišćenja – Pretraživanje rezervacija

1. **Radnik** poziva **sistem** da nađe *rezervacije* po zadatim vrednostima. (APSO)
2. **Sistem** prikazuje **radniku** podatke o *rezervacijama*. (IA)
3. **Radnik** poziva **sistem** da prikaže podatke o izabranoj *rezervaciji*. (APSO)
4. **Sistem** prikazuje **radniku** podatke o izabranoj *rezervaciji*. (IA)



**Alternativna scenarija**

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



4.1. Ukoliko **sistem** ne može da nađe izabranu *rezervaciju* on prikazuje **radniku** poruku: “**Sistem** ne može da nađe podatke o izabranoj *rezervaciji*”. (IA)



Sa navedenih sekvencnih dijagrama uočavaju se 2 sistemske operacije koje treba projektovati:

1. *signal* **PretraziRezervacije**(*KriterijumPretrage, List<Rezervacija>*)
2. *signal* **UcitajRezervaciju**(*Rezervacija*)

### DS 8: Dijagram sekvenci slučaja korišćenja – Brisanje rezervacije

1. **Radnik** poziva **sistem** da nađe *rezervacije* po zadatim vrednostima. (APSO)
2. **Sistem** prikazuje **radniku** nađene *rezervacije*. (IA)
3. **Radnik** poziva **sistem** da obriše *rezervaciju*. (APSO)
4. **Sistem** prikazuje **radniku** poruku: “**Sistem** je obrisao *rezervaciju*.” (IA)



**Alternativna scenarija**

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



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



Sa navedenih sekvencnih dijagrama uočavaju se 2 sistemske operacije koje treba projektovati:

1. *signal* **PretraziRezervacije**(*KriterijumPretrage, List<Rezervacija>*)
2. *signal* **ObrisiRezervaciju**(*Rezervacija*)

### DS 9: Dijagram sekvenci slučaja korišćenja – Prodaja karata

1. **Forma** poziva **sistem** da prikaže podatke o *radnicima.*
2. **Sistem** prikazuje **formi** podatke o *radnicima.*
3. **Forma** poziva **sistem** da prikaže podatke o *projekcijama.*
4. **Sistem** prikazuje **formi** podatke o *projekcijama.*
5. **Forma** poziva **sistem** da prikaže podatke o *sedištima.*
6. **Sistem** prikazuje **formi** podatke o *sedištima.*
7. **Radnik** poziva **sistem** da kreira *nove karte*. (APSO)
8. **Sistem** prikazuje **radniku** *nove karte*. (IA)
9. **Radnik** poziva **sistem** da zapamti podatke o *novim kartama*. (APSO)
10. **Sistem** prikazuje **radniku** poruku: “**Sistem** je zapamtio *karte*”. (IA)



**Alternativna scenarija**

2.1. Ukoliko **sistem** ne može da nađe listu *radnika* on prikazuje **formi** poruku: “**Sistem** ne može da nađe listu *radnika*”.



4.1. Ukoliko **sistem** ne može da nađe listu *projekcija* on prikazuje **formi** poruku: “**Sistem** ne može da nađe listu *projekcija*”.



6.1. Ukoliko **sistem** ne može da nađe listu *sedišta* on prikazuje **formi** poruku: “**Sistem** ne može da nađe listu *sedišta*”.



8.1. Ukoliko **sistem** ne može da kreira *nove karte* on prikazuje **radniku** poruku: “**Sistem** ne može da kreira *nove karte*”. Prekida se izvršavanje scenarija. (IA)



10.1. Ukoliko **sistem** ne može da zapamti podatke o *novim kartama* on prikazuje **radniku** poruku “**Sistem** ne može da zapamti *nove karte*”. (IA)



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

1. *signal* **UcitajListuRadnika**(*List<Radnik>*)
2. *signal* **UcitajListuProjekcija**(*List<Projekcija>*)
3. *signal* **UcitajListuSedista**(*List<Sediste>*)
4. *signal* **KreirajNoveKarte** (List<*Karta>*)
5. *signal* **ZapamtiKarte**(*List<Karta>*)

Kao rezultat analize scenarija dobijeno je ukupno 16 sistemskih operacija koje treba projektovati:

1. *signal* **NadjiRadnika**(*Radnik*)
2. *signal* **UcitajListuRadnika**(*List<Radnik>*)
3. *signal* **KreirajNoviFilm**(*Film*)
4. *signal* **ZapamtiFilm**(*Film*)
5. *signal* **PretraziFilmove**(*KriterijumPretrage, List<Film>*)
6. *signal* **UcitajFilm**(*Film*)
7. *signal* **ObrisiFilm**(*Film*)
8. *signal* **UcitajListuProjekcija**(*List<Projekcija>*)
9. *signal* **KreirajNovuRezervaciju**(*Rezervacija*)
10. *signal* **ZapamtiRezervaciju**(*Rezervacija*)
11. *signal* **PretraziRezervacije**(*KriterijumPretrage, List<Rezervacija>*)
12. *signal* **UcitajRezervaciju**(*Rezervacija*)
13. *signal* **ObrisiRezervaciju**(*Rezervacija*)
14. *signal* **UcitajListuSedista**(*List<Sediste>*)
15. *signal* **KreirajNoveKarte**(*List<Karta>*)
16. *signal* **ZapamtiKarte**(*List<Karta>*)

## Ponašanje softverskog sistema – Definisanje ugovora o sistemskim operacijama

**Ugovor UG1: NadjiRadnika**

Operacija: NadjiRadnika(*Radnik*):signal;

Veza sa SK: SK1

Preduslovi: -

Postuslovi: *-*

**Ugovor UG2: UcitajListuRadnika**

Operacija: UcitajListuRadnika(*List<Radnik>*):signal;

Veza sa SK: SK3, SK6, SK9

Preduslovi: -

Postuslovi: -

**Ugovor UG3: KreirajNoviFilm**

Operacija: KreirajNoviFilm(*Film*):signal;

Veza sa SK: SK2

Preduslovi: *Prosto vrednosno ograničenje nad objektom Film mora biti zadovoljeno.*

*Strukturno ograničenje nad objektom Film mora biti zadovoljeno.*

Postuslovi: *Kreiran je novi film.*

**Ugovor UG4: ZapamtiFilm**

Operacija: ZapamtiFilm(*Film*):signal;

Veza sa SK: SK2, SK4

Preduslovi: *Prosto vrednosno ograničenje nad objektom Film mora biti zadovoljeno.*

*Strukturno ograničenje nad objektom Film mora biti zadovoljeno.*

Postuslovi: *Podaci o filmu su zapamćeni.*

**Ugovor UG5: PretraziFilmove**

Operacija: PretraziFilmove(*KriterijumPretrage,List<Film>*):signal;

Veza sa SK: SK3, SK4, SK5

Preduslovi: -

Postuslovi: -

**Ugovor UG6: UcitajFilm**

Operacija: UcitajFilm(*Film*):signal;

Veza sa SK: SK3, SK4

Preduslovi: -

Postuslovi: -

**Ugovor UG7:** **ObrisiFilm**

Operacija: ObrisiFilm(*Film*):signal;

Veza sa SK: SK5

Preduslovi: -

Postuslovi: *Film je obrisan. Strukturno ograničenje nad objektom Film mora biti zadovoljeno.*

**Ugovor UG8: UcitajListuProjekcija**

Operacija: UcitajListuProjekcija(*List<Projekcija>*):signal;

Veza sa SK: SK6, SK9

Preduslovi: -

Postuslovi: -

**Ugovor UG9: KreirajNovuRezervaciju**

Operacija: KreirajNovuRezervaciju(*Rezervacija*):signal;

Veza sa SK: SK6

Preduslovi: *Ukoliko je popunjen kapacitet sale za izabrani termin ne može se izvršiti sistemska operacija.*

*Prosto vrednosno ograničenje nad objektom Rezervacija mora biti zadovoljeno.*

*Strukturno ograničenje nad objektom Rezervacija mora biti zadovoljeno.*

Postuslovi: *Kreirana je nova rezervacija.*

**Ugovor UG10: ZapamtiRezervaciju**

Operacija: ZapamtiRezervaciju(*Rezervacija*):signal;

Veza sa SK: SK6

Preduslovi: *Ukoliko je popunjen kapacitet sale za izabrani termin ne može se izvršiti sistemska operacija.*

*Prosto vrednosno ograničenje nad objektom Rezervacija mora biti zadovoljeno.*

*Strukturno ograničenje nad objektom Rezervacija mora biti zadovoljeno.*

Postuslovi: *Podaci o rezervaciji su zapamćeni.*

**Ugovor UG11: PretraziRezervacije**

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

Veza sa SK: SK7, SK8

Preduslovi: -

Postuslovi: -

**Ugovor UG12: UcitajRezervaciju**

Operacija: UcitajRezervaciju(*Rezervacija*):signal;

Veza sa SK: SK7

Preduslovi: -

Postuslovi: -

**Ugovor UG13: ObrisiRezervaciju**

Operacija: ObrisiRezervaciju(*Rezervacija*):signal;

Veza sa SK: SK8

Preduslovi:

Postuslovi: *Rezervacija je obrisana. Strukturno ograničenje nad objektom Rezervacija mora biti zadovoljeno.*

**Ugovor UG14: UcitajListuSedista**

Operacija: UcitajListuSedista(*List<Sediste>*):signal;

Veza sa SK: SK9

Preduslovi: -

Postuslovi: -

**Ugovor UG15: KreirajNoveKarte**

Operacija: KreirajNoveKarte(List<*Karta>*):signal;

Veza sa SK: SK9

Preduslovi: *Ukoliko sedišta nisu rezervisana i popunjen je kapacitet sale za izabrani termin ne može se izvršiti sistemska operacija.*

*Prosto vrednosno ograničenje nad objektom Karta mora biti zadovoljeno.*

*Strukturno ograničenje nad objektom Karta mora biti zadovoljeno*

Postuslovi: *Kreirane su nove karte*

**Ugovor UG16: ZapamtiKarte**

Operacija: ZapamtiKarte(List<*Karta>*):signal;

Veza sa SK: SK9

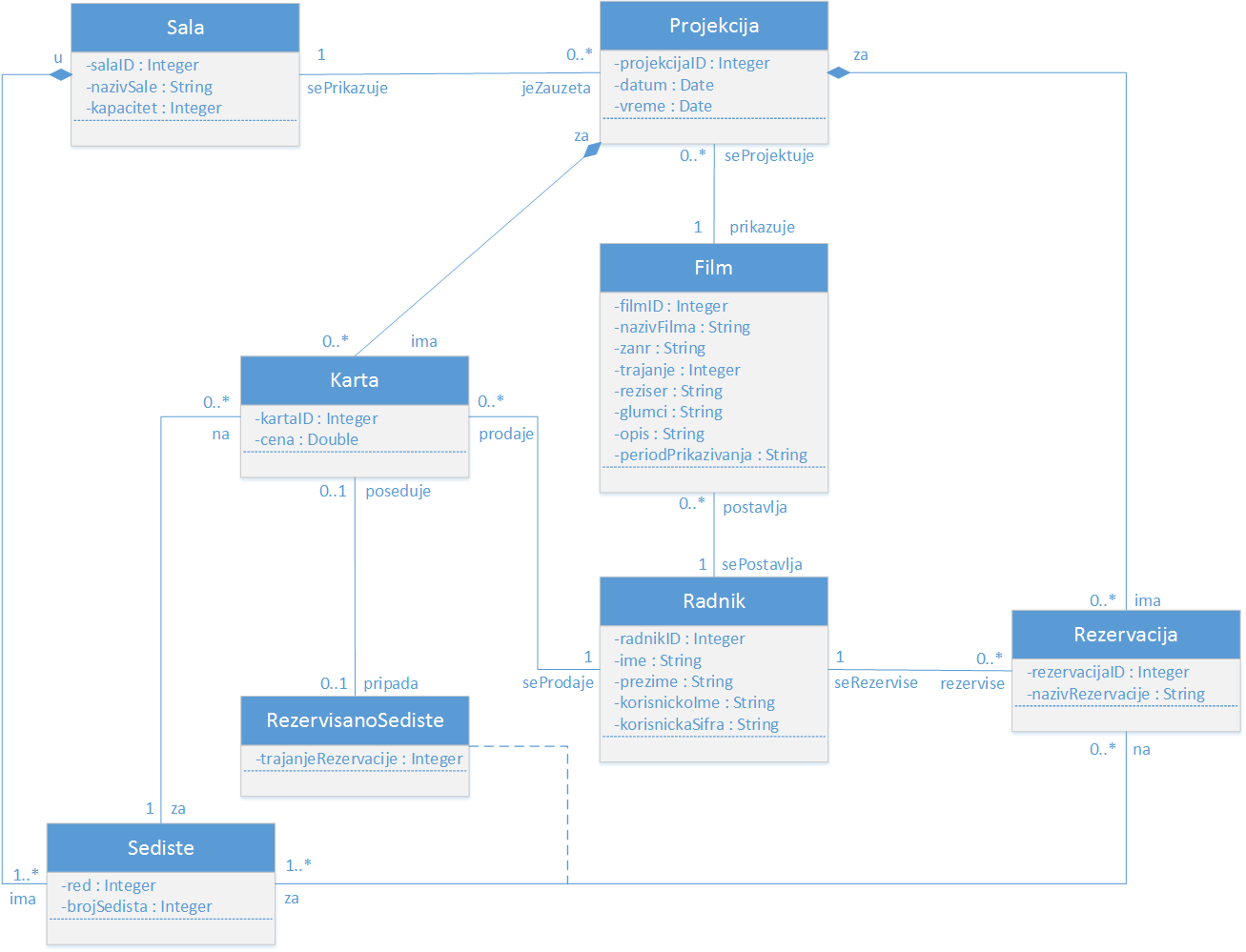
Preduslovi: *Ukoliko sedišta nisu rezervisana i popunjen je kapacitet sale za izabrani termin ne može se izvršiti sistemska operacija.*

*Prosto vrednosno ograničenje nad objektom Karta mora biti zadovoljeno.*

*Strukturno ograničenje nad objektom Karta mora biti zadovoljeno*

Postuslovi: *Podaci o kartama su zapamćeni.*

## Struktura softverskog sistema – Konceptualni (domenski) model



## Struktura softverskog sistema – Relacioni model

Radnik(radnikID, ime, prezime, korisnickoIme, korisnickaSifra)

Film(filmID, nazivFilma, zanr, trajanje, reziser, glumci, opis, periodPrikazivanja, *radnikID*)

Sala(salaID, nazivSale, kapacitet)

Projekcija(projekcijaID, datum, vreme*, filmID, salaID*)

Sediste(salaID, red, brojSedista)

Karta(projekcijaID, kartaID, cena*, radnikID, salaID, red, brojSedista*)

Rezervacija(projekcijaID, rezervacijaID, nazivRezervacije, *radnikID*)

RezervisanoSediste(projekcijaID, rezervacijaID, salaID, red, brojSedista,trajanjeRezervacije)

KartaZaRezervisanoSediste(projekcijaID, rezervacijaID, salaID, red, brojSedista, KartaID)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Radnik** | | **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 Film, Karta, Rezervacija  DELETE RESTRICTED Film, Karta,  Rezervacija |
| radnikID | String | not null |  |  |
| ime | String | not null |  |  |
| prezime | String | not null |  |  |
| korisnickoIme | String | not null |  |  |
| korisnickaSifra | String | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Film** | | **Prosto vrednosno ograničenje** | | **Složeno vrednosno ograničenje** | | **Strukturno ograničenje** |
| **Atributi** | **Ime** | **Tip atributa** | **Vrednost atributa** | **Međuzav. atributa jedne tabele** | **Međuzav. atributa više tabela** | INSERT RESTRICTED Radnik  UPDATE CASCADES Projekcija, RESTRICTED Radnik  DELETE RESTRICTED Projekcija |
| filmID | String | not null |  |  |
| nazivFilma | String | not null |  |  |
| zanr | String | not null |  |  |
| trajanje | Integer | not null |  |  |
| reziser | String | not null |  |  |
| glumci | String | not null |  |  |
| opis | String | not null |  |  |
| periodPrikazivanja | String | not null |  |  |
| radnikID | String | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Sala** | | **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 Sediste, Projekcija  DELETE CASCADES Sediste RESTRICTED Projekcija |
| salaID | String | not null |  |  |
| nazivSale | String | not null |  |  |
| kapacitet | Integer | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Projekcija** | | **Prosto vrednosno ograničenje** | | **Složeno vrednosno ograničenje** | | **Strukturno ograničenje** |
| **Atributi** | **Ime** | **Tip atributa** | **Vrednost atributa** | **Međuzav. atributa jedne tabele** | **Međuzav. atributa više tabela** | INSERT RESTRICTED Film, Sala  UPDATE CASCADES Karta, Rezervacija, RESTRICTED Film, Sala  DELETE CASCADES Karta, Rezervacija |
| projekcijaID | String | not null |  |  |
| datum | Date | not null |  |  |
| vreme | Date | not null |  |  |
| FilmID | String | not null |  |  |
| SalaID | String | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Sediste** | | **Prosto vrednosno ograničenje** | | **Složeno vrednosno ograničenje** | | **Strukturno ograničenje** |
| **Atributi** | **Ime** | **Tip atributa** | **Vrednost atributa** | **Međuzav. atributa jedne tabele** | **Međuzav. atributa više tabela** | INSERT RESTRICTED Sala  UPDATE CASCADES RezervisanoSediste, Karta, RESTRICTED Sala  DELETE RESTRICTED RezervisanoSediste, Karta |
| salaID | String | not null |  |  |
| red | Integer | not null and >0 |  |  |
| brojSedista | Integer | not null and >0 |  |  |

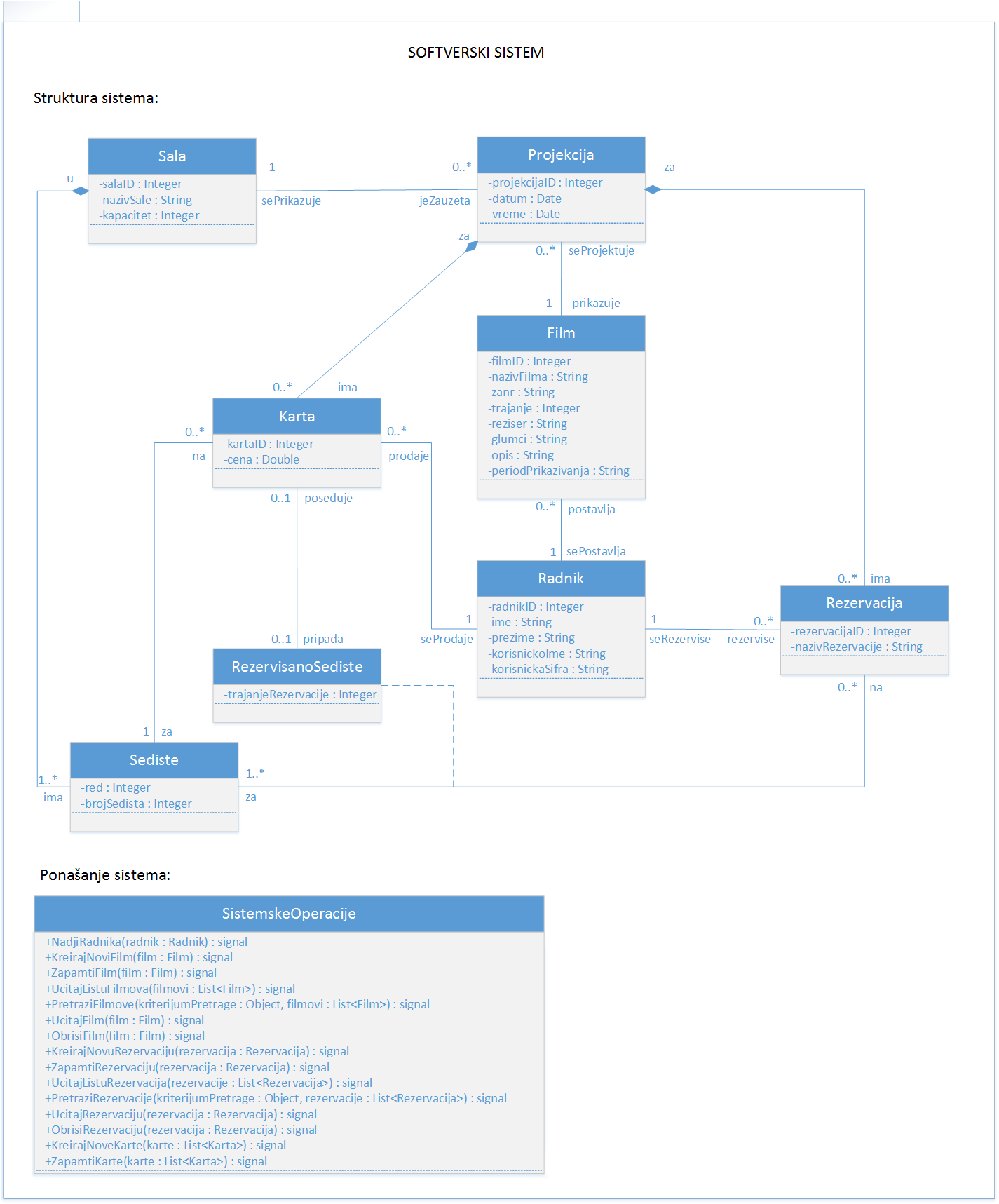
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Karta** | | **Prosto vrednosno ograničenje** | | **Složeno vrednosno ograničenje** | | **Strukturno ograničenje** |
| **Atributi** | **Ime** | **Tip atributa** | **Vrednost atributa** | **Međuzav. atributa jedne tabele** | **Međuzav. atributa više tabela** | INSERT RESTRICTED Radnik, Projekcija, Sediste  UPDATE CASCADES KartaZaRezervisanoSediste, RESTRICTED Radnik, Projekcija, Sediste  DELETE RESTRICTED KartaZaRezervisanoSediste |
| projekcijaID | Integer | not null |  |  |
| kartaID | Integer | not null |  |  |
| cena | Double | not null |  |  |
| radnikID | Integer | not null |  |  |
| salaID | Integer | not null |  |  |
| red | Integer | not null and >0 |  |  |
| brojSedista | Integer | not null and >0 |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Rezervacija** | | **Prosto vrednosno ograničenje** | | **Složeno vrednosno ograničenje** | | **Strukturno ograničenje** |
| **Atributi** | **Ime** | **Tip atributa** | **Vrednost atributa** | **Međuzav. atributa jedne tabele** | **Međuzav. atributa više tabela** | INSERT RESTRICTED Radnik, Projekcija  UPDATE CASCADES RezervisanoSediste RESTRICTED Radnik, Projekcija  DELETE RESTRICTED RezervisanoSediste |
| projekcijaID | Integer | not null |  |  |
| rezervacijaID | Integer | not null |  |  |
| nazivRezervacije | String | not null |  |  |
| radnikID | Integer | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela RezervisanoSediste** | | **Prosto vrednosno ograničenje** | | **Složeno vrednosno ograničenje** | | **Strukturno ograničenje** |
| **Atributi** | **Ime** | **Tip atributa** | **Vrednost atributa** | **Međuzav. atributa jedne tabele** | **Međuzav. atributa više tabela** | INSERT RESTRICTED Rezervacija, Sediste  UPDATE CASCADES KartaZaRezervisanoSediste, RESTRICTED Rezervacija, Sediste  DELETE RESTRICTED KartaZaRezervisanoSediste |
| projekcijaID | Integer | not null |  |  |
| rezervacijaID | Integer | not null |  |  |
| salaID | String | not null |  |  |
| red | Integer | not null and >0 |  |  |
| brojSedista | Integer | not null and >0 |  |  |
| trajanjeRezervacije | String | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela KartaZaRezervisanoSediste** | | **Prosto vrednosno ograničenje** | | **Složeno vrednosno ograničenje** | | **Strukturno ograničenje** |
| **Atributi** | **Ime** | **Tip atributa** | **Vrednost atributa** | **Međuzav. atributa jedne tabele** | **Međuzav. atributa više tabela** | INSERT RESTRICTED RezervisanoSediste, Karta  UPDATE RESTRICTED RezervisanoSediste, Karta  DELETE / |
| projekcijaID | Integer | not null |  |  |
| rezervacijaID | Integer | not null |  |  |
| salaID | Integer | not null |  |  |
| red | Integer | not null and >0 |  |  |
| brojSedista | Integer | not null and >0 |  |  |
| KartaID | Integer | not null |  |  |

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



# Projektovanje

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

## Arhitektura softverskog sistema

Arhitektura softverskog sistem je tronivojska i sastoji se od sledećih nivoa:

* Korisnički interfejs
* Aplikaciona logika
* Skladište podataka

Nivo korisničkog interfejsa je na strani klijenta, a aplikacaiona logika i skladište podataka na strani servera.

I nivo

II nivo

III nivo

SOFTVERSKI SISTEM

Skladište podataka

Aplikaciona logika

Korisnički interfejs

Slika 2. Tronivojska arhitektura

## Projektovanje korisničkog interfejsa

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

Softverski sistem

Korisnički interfejs

Ekranska forma

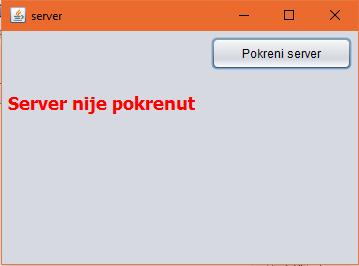
Kontroler KI

Slika 3. Struktura korisničkog interfejsa

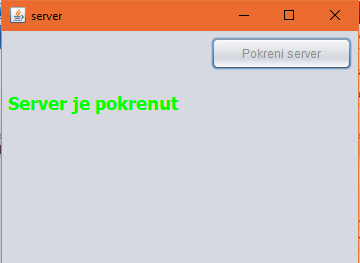
### Projektovanje ekranskih formi

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

Izgled ekrena na serverskoj strani prilikom samog startovanja istog:



Prva slika pokazuje da se server nije startovao normalno, dok se na slici ispod vidi odziv sistema prilikom uspešnog pokretanja istog. Nakon pokretanja servera prelazi se na klijentski deo aplikacije.



Pokretanjem klijenta dolazi se do ulazne login forme koja sadrži validacije podataka korisnickoIme i korisnickaSifra.

#### *SK 1: Slučaj korišćenja – Prijavljivanje radnika*

**Naziv SK**

Prijavljivanje radnika

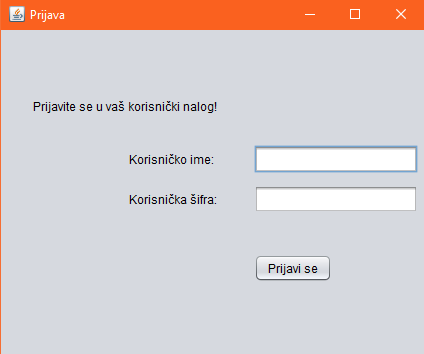
**Aktori SK**

**Radnik**

**Učesnici SK**

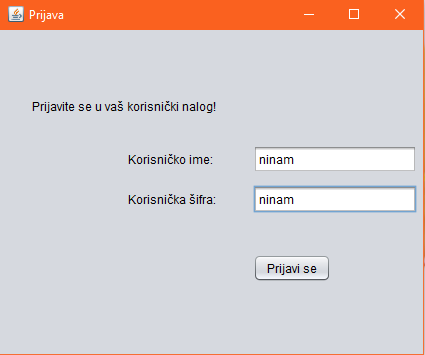
**Radnik** i **sistem** (program)

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



**Osnovni scenario SK**

1. **Radnik** unosi podatke za autentifikaciju radnika. (APUSO)



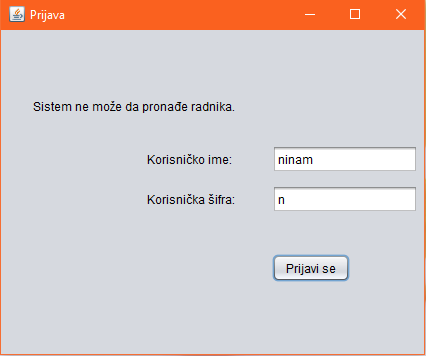
1. **Radnik** poziva **sistem** da pronađe radnika sa zadatim podacima. (APSO)

*Opis akcije: Radnik klikom na dugme “Prijavi se” poziva sistemsku operaciju nadjiRadnika(Radnik) koja traži radnika.*

1. **Sistem** pretražuje bibliotekare. (SO)
2. **Sistem** prikazuje poruku da je biliotekar uspešno prijavljen i početnu formu. (IA)

**Alternativna scenarija**

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



#### SK 2: Slučaj korišćenja – Postavljanje novog filma na repertoar

**Naziv SK**

Postavljanje novog filma na repertoar

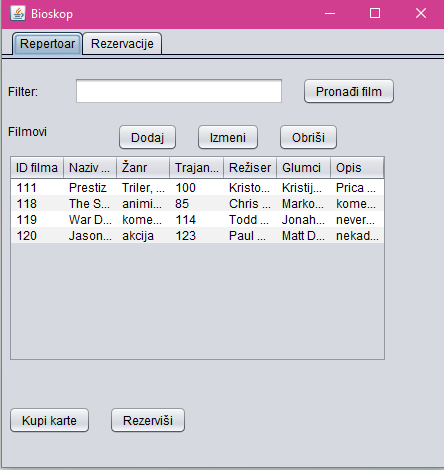
**Aktori SK**

**Radnik**

**Učesnici SK**

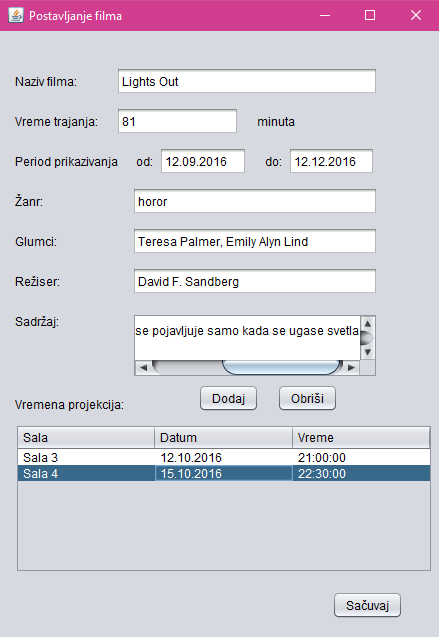
**Radnik** i **sistem** (program)

**Preduslov:** **Sistem** je uključen i **radnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje početnu formu i tab *repertoar*.

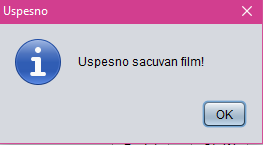


**Osnovni scenario SK**

1. **Radnik** poziva **sistem** da kreira *novi film*. (APSO)
2. **Sistem** kreira *novi film*. (SO)
3. **Sistem** prikazuje **radniku** *novi film*. (IA)
4. **Radnik** unosi podatke o *novom filmu*. (APUSO)



1. **Radnik** kontroliše da li je korektno uneo podatke o *novom filmu*. (ANSO)
2. **Radnik** poziva **sistem** da zapamti podatke o *novom filmu*. (APSO)
3. **Sistem** pamti podatke o novom filmu. (SO)
4. **Sistem** prikazuje **radniku** poruku: “**Sistem** je zapamtio *film*”. (IA)



**Alternativna scenarija**

3.1. Ukoliko **sistem** ne može da kreira *novi film* on prikazuje **radniku** poruku: “**Sistem** ne može da kreira *novi film*”. Prekida se izvršavanje scenarija. (IA)

8.1. Ukoliko **sistem** ne može da zapamti podatke o *novom* *filmu* on prikazuje **radniku** poruku “**Sistem** ne može da zapamti *novi film*”. (IA)

#### SK 3: Slučaj korišćenja – Pretraživanje filmova

**Naziv SK**

Pretraživanje filmova

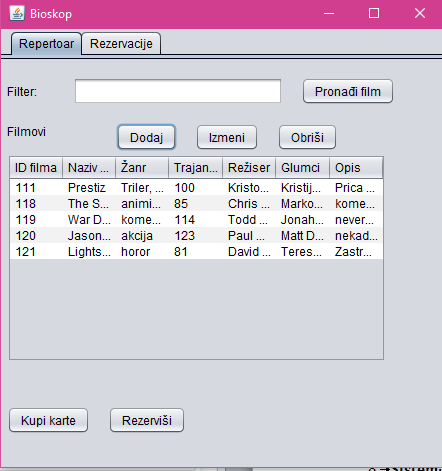
**Aktori SK**

**Radnik**

**Učesnici SK**

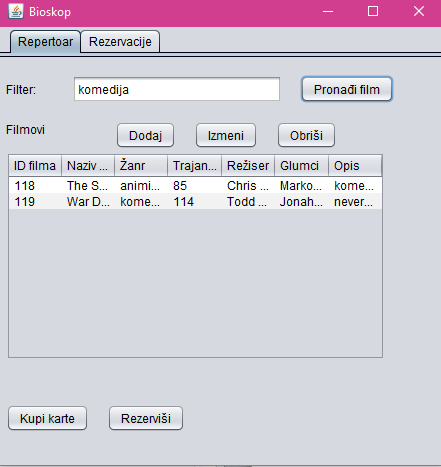
**Radnik** i **sistem** (program)

**Preduslov:** **Sistem** je uključen i **radnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje početnu formu i tab *repertoar*.

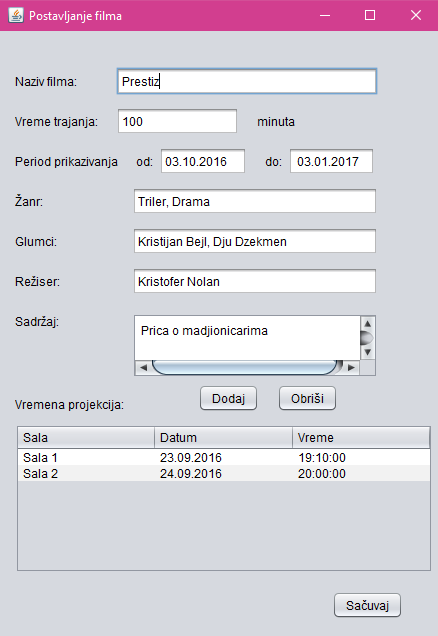


**Osnovni scenario SK**

1. **Radnik** unosi vrednosti po kojim pretražuje *filmove*. (APUSO)
2. **Radnik** poziva **sistem** da nađe *filmove* po zadatim vrednostima. (APSO)
3. **Sistem** traži *filmove* po zadatim vrednostima. (SO)
4. **Sistem** prikazuje **radniku** nađene *filmove*. (IA)



1. **Radnik** bira *film* čije podatke želi da vidi. (APUSO)
2. **Radnik** poziva **sistem** da prikaže podatke o izabranom *filmu*. (APSO)
3. **Sistem** pronalazi podatke o izabranom *filmu*. (SO)
4. **Sistem** prikazuje **radniku** podatke o izabranom *filmu*. (IA)



**Alternativna scenarija**

4.1. Ukoliko **sistem** ne može da nađe *filmove* on prikazuje **radniku** poruku: “**Sistem** ne može da nađe *filmove* po zadatim vrednostima”. Prekida se izvršenje scenarija. (IA)

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

#### SK 4: Slučaj korišćenja – Izmena podataka o filmu

**Naziv SK**

Izmena podataka o filmu

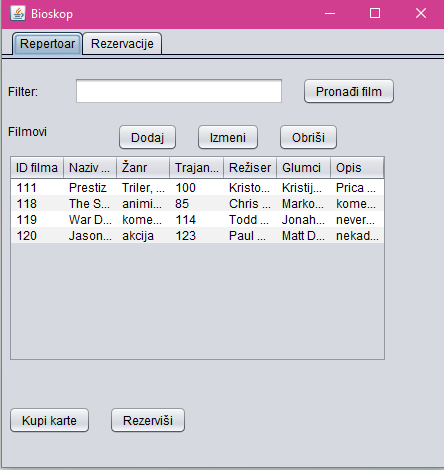
**Aktori SK**

**Radnik**

**Učesnici SK**

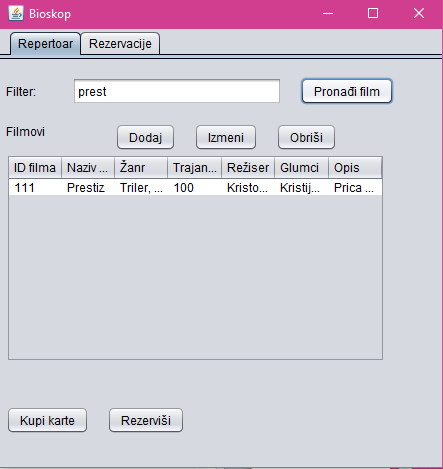
**Radnik** i **sistem** (program)

**Preduslov:** **Sistem** je uključen i **radnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje početnu formu i tab *repertoar*.

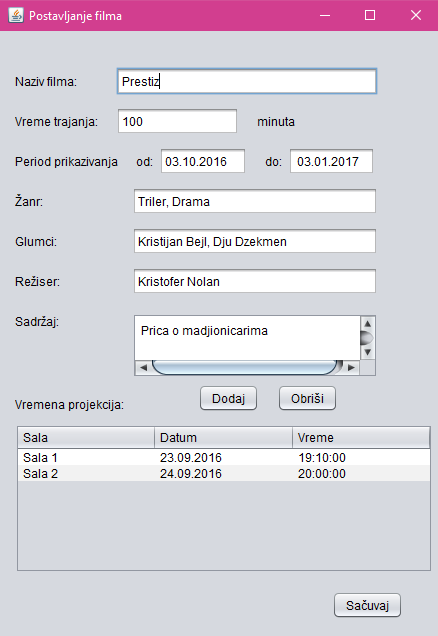


**Osnovni scenario SK**

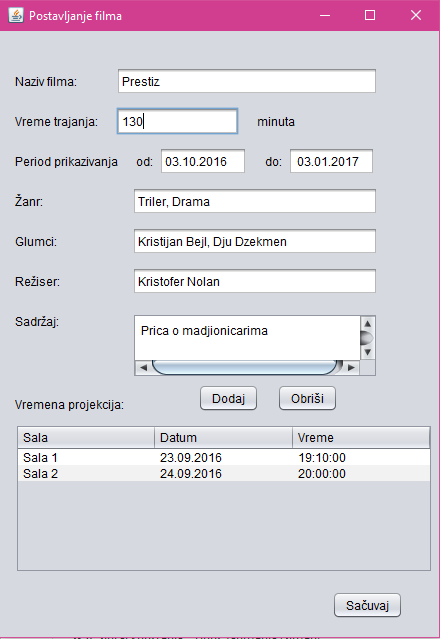
1. **Radnik** unosi vrednosti po kojima pretražuje *filmove*. (APUSO)
2. **Radnik** poziva **sistem** da nađe *filmove* po zadatim vrednostima. (APSO)
3. **Sistem** traži *filmove* po zadatim vrednostima. (SO)
4. **Sistem** prikazuje **radniku** nađene *filmove*. (IA)



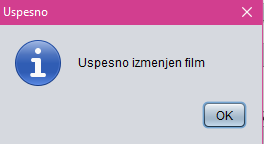
1. **Radnik** bira *film* koji želi da izmeni. (APUSO)
2. **Radnik** poziva **sistem** da učita podatke o izabranom *filmu*. (APSO)
3. **Sistem** pronalazi podatke o izabranom *filmu*. (SO)
4. **Sistem** prikazuje **radniku** podatke o izabranom *filmu*. (IA)



1. **Radnik** unosi (menja) podatke o *filmu*. (APUSO)
2. **Radnik** kontroliše da li je korektno uneo podatke o *filmu*. (ANSO)



1. **Radnik** poziva **sistem** da zapamti podatke o *filmu*. (APSO)
2. **Sistem** pamti podatke o *filmu*. (SO)
3. **Sistem** prikazuje **radniku** poruku: “**Sistem** je zapamtio *film*.” (IA)



**Alternativna scenarija:**

4.1. Ukoliko **sistem** ne može da nađe *filmove* on prikazuje **radniku** poruku: “**Sistem** ne može da nađe *filmove* po zadatim vrednostima”. Prekida se izvršenje scenarija. (IA)

8.1. Ukoliko **sistem** ne može da nađe *film* on prikazuje **radniku** poruku: “**Sistem** ne može da nađe izabran *film*”. Prekida se izvršenje scenarija. (IA)

13.1. Ukoliko **sistem** ne može da zapamti podatke o *filmu* on prikazuje **radniku** poruku: “**Sistem** ne može da zapamti *film*”. (IA)

#### SK 5: Slučaj korišćenja – Brisanje filma sa repertoara

**Naziv SK**

Brisanje filma sa repertoara

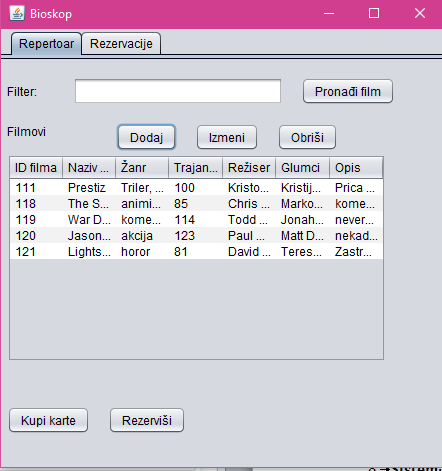
**Aktori SK**

**Radnik**

**Učesnici SK**

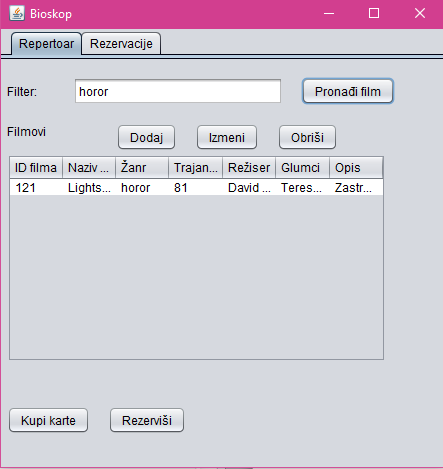
**Radnik** i **sistem** (program)

**Preduslov:** **Sistem** je uključen i **radnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje početnu formu i tab *repertoar*.

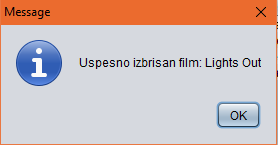


**Osnovni scenario SK**

1. **Radnik** unosi vrednosti po kojim pretražuje *filmove*. (APUSO)
2. **Radnik** poziva **sistem** da nađe *filmove* po zadatim vrednostima. (APSO)
3. **Sistem** traži *filmove* po zadatim vrednostima. (SO)
4. **Sistem** prikazuje **radniku** nađene *filmove*. (IA)



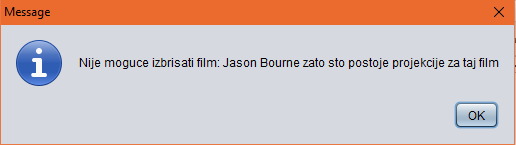
1. **Radnik** bira *film* kojeg želi da obriše. (APUSO)
2. **Radnik** poziva **sistem** da obriše *film*. (APSO)
3. **Sistem** briše *film*. (SO)
4. **Sistem** prikazuje **radniku** poruku: “**Sistem** je obrisao *film*.” (IA)



**Alternativna scenarija**

4.1. Ukoliko **sistem** ne može da nađe *filmove* on prikazuje **radniku** poruku: “**Sistem** ne može da nađe *filmove* po zadatim vrednostima”. Prekida se izvršenje scenarija. (IA)

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



#### SK 6: Slučaj korišćenja – Rezervacija karata

**Naziv SK**

Rezervacija karata

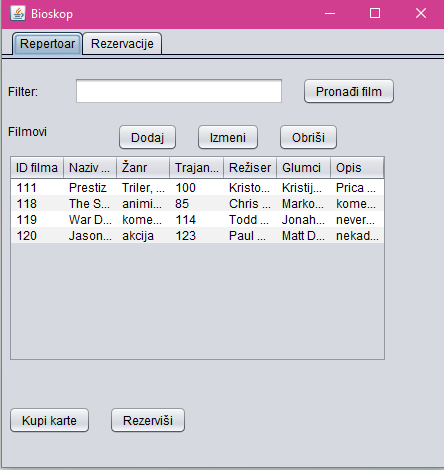
**Aktori SK**

**Radnik**

**Učesnici SK**

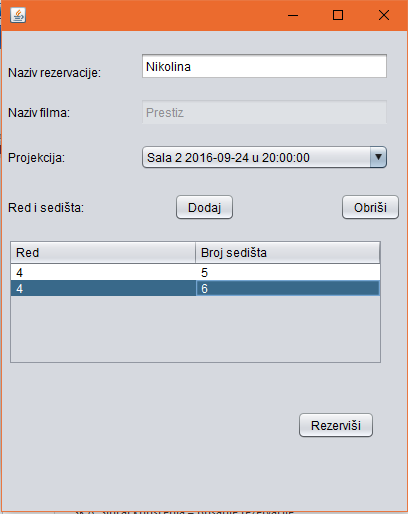
**Radnik** i **sistem** (program)

**Preduslov:** **Sistem** je uključen i **radnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje početnu formu i tab *repertoar*.

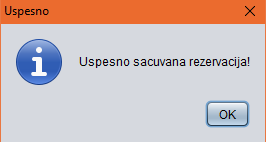


**Osnovni scenario SK**

1. **Radnik** poziva **sistem** da kreira *novu rezervaciju*. (APSO)
2. **Sistem** kreira *novu rezervaciju*. (SO)
3. **Sistem** prikazuje **radniku** *novu* *rezervaciju*. (IA)
4. **Radnik** unosi podatke o *novoj rezervaciji*. (APUSO)



1. **Radnik** kontroliše da li je korektno uneo podatke o *novoj rezervaciji*. (ANSO)
2. **Radnik** poziva **sistem** da zapamti podatke o *novoj rezervaciji*. (APSO)
3. **Sistem** pamti podatke o *novoj rezervaciji*. (SO)
4. **Sistem** prikazuje **radniku** poruku:”**Sistem** je zapamtio *rezervaciju*”. (IA)



**Alternativna scenarija**

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

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

#### SK 7: Slučaj korišćenja – Pretraživanje rezervacija

**Naziv SK**

Pretraživanje rezervacija

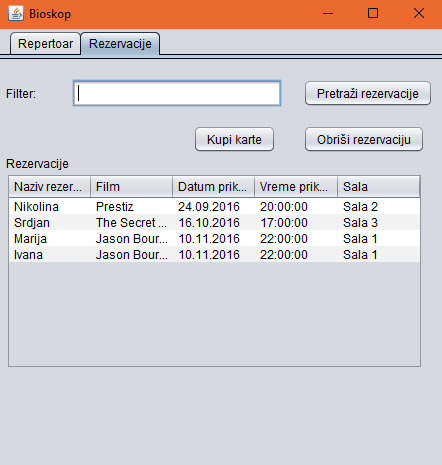
**Aktori SK**

**Radnik**

**Učesnici SK**

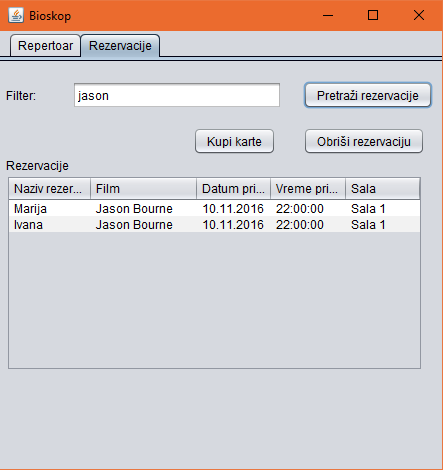
**Radnik** i **sistem** (program)

**Preduslov:** **Sistem** je uključen i **radnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje početnu formu i tab *rezervacije*.



**Osnovni scenario SK**

1. **Radnik** unosi vrednosti po kojima pretražuje *rezervacije*. (APUSO)
2. **Radnik** poziva **sistem** da nađe *rezervacije* po zadatim vrednostima. (APSO)
3. **Sistem** traži *rezervacije* po zadatim vrednostima. (SO)
4. **Sistem** prikazuje **radniku** podatke o *rezervacijama*. (IA)



1. **Radnik** bira *rezervaciju* za koju želi da vidi podatke. (APUSO)
2. **Radnik** poziva **sistem** da prikaže podatke o izabranoj *rezervaciji*. (APSO)
3. **Sistem** traži podatke o izabranoj *rezervaciji*. (SO)
4. **Sistem** prikazuje **radniku** podatke o izabranoj *rezervaciji*. (IA)

**Alternativna scenarija**

4.1. Ukoliko **sistem** ne može da nađe *rezervaciju* on prikazuje **radniku** poruku: “**Sistem** ne može da nađe *rezervaciju* po zadatim vrednostima”. Prekida se izvršenje scenarija. (IA)

8.1. Ukoliko **sistem** ne može da nađe izabranu *rezervaciju* on prikazuje **radniku** poruku: “**Sistem** ne može da nađe podatke o izabranoj *rezervaciji*”. (IA)

#### SK 8: Slučaj korišćenja – Brisanje rezervacije

**Naziv SK**

Brisanje rezervacije

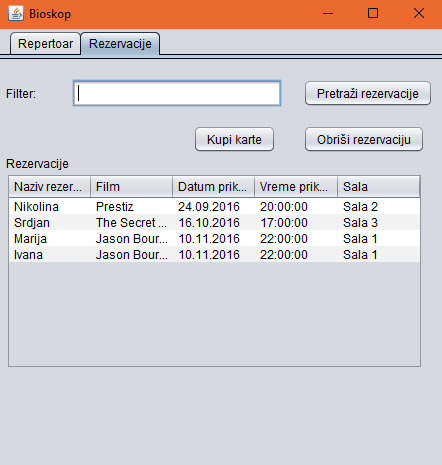
**Aktori SK**

**Radnik**

**Učesnici SK**

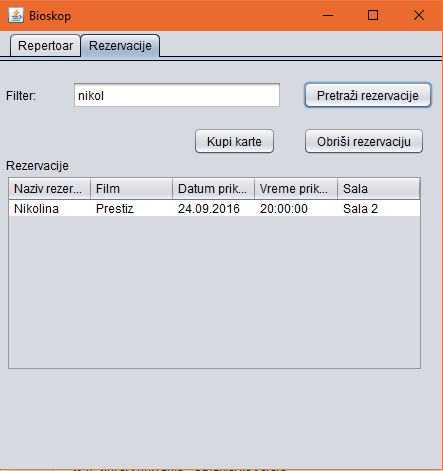
**Radnik** i **sistem** (program)

**Preduslov:** **Sistem** je uključen i **radnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje početnu formu i tab *rezervacije*.

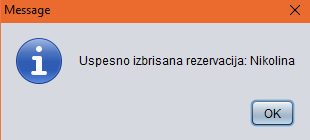


**Osnovni scenario SK**

1. **Radnik** unosi vrednosti po kojim pretražuje *rezervacije*. (APUSO)
2. **Radnik** poziva **sistem** da nađe *rezervacije* po zadatim vrednostima. (APSO)
3. **Sistem** traži *rezervacije* po zadatim vrednostima. (SO)
4. **Sistem** prikazuje **radniku** nađene *rezervacije*. (IA)



1. **Radnik** bira *rezervaciju* koju želi da obriše. (APUSO)
2. **Radnik** poziva **sistem** da obriše *rezervaciju*. (APSO)
3. **Sistem** briše *rezervaciju*. (SO)
4. **Sistem** prikazuje **radniku** poruku: “**Sistem** je obrisao *rezervaciju*.” (IA)



**Alternativna scenarija**

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

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

#### SK 9: Slučaj korišćenja – Prodaja karata

**Naziv SK**

Prodaja karata

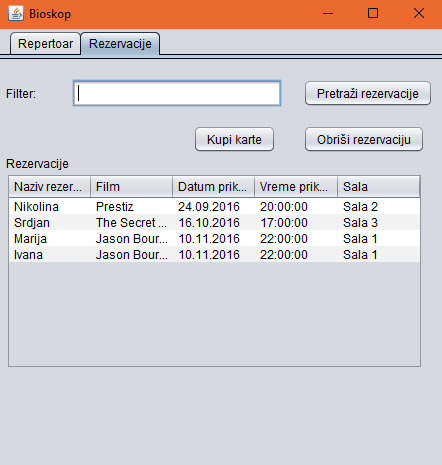
**Aktori SK**

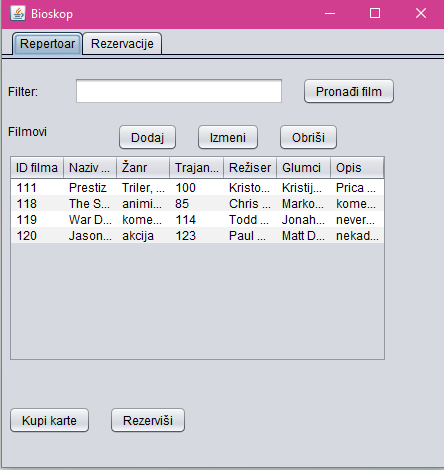
**Radnik**

**Učesnici SK**

**Radnik** i **sistem** (program)

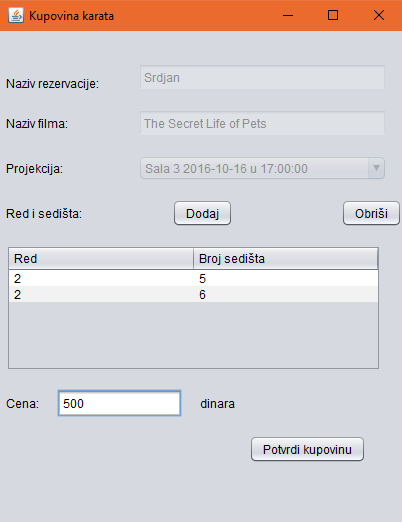
**Preduslov:** **Sistem** je uključen i **radnik** je ulogovan pod svojom šifrom. **Sistem** prikazuje početnu formu i tab *rezervacije* ili tab *repertoar*.





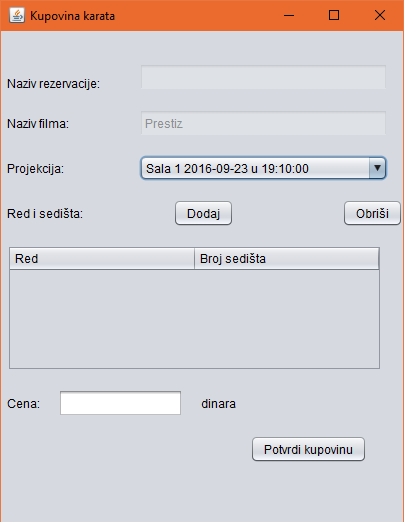
**Osnovni scenario SK**

1. **Radnik** poziva **sistem** da kreira *nove karte*. (APSO)
2. **Sistem** kreira *nove karte*. (SO)
3. **Sistem** prikazuje **radniku** *nove karte*. (IA)



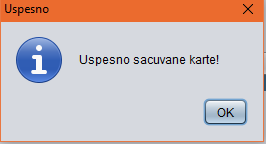
*Ukoliko je radnik u formu za kupovinu karata došao preko taba rezervacija, prikazuje mu se forma kupovina karata na kojoj popunjava samo cenu, ostali podaci se automatski popunjavaju na osnovu rezervacije.*

*Ukoliko je radnik u formu za kupovinu karata došao preko taba repertoar, prikazuje mu se sledeća forma.*



*Radnik zatim bira za koju projekciju se kupuju karte, unosi sedišta za koja se kupuju karte, kao i cenu karata.*

1. **Radnik** unosi podatke o *novim kartama*. (APUSO)
2. **Radnik** kontroliše da li je korektno uneo podatke o *novim kartama*. (ANSO)
3. **Radnik** poziva **sistem** da zapamti podatke o *novim kartama*. (APSO)
4. **Sistem** pamti podatke o *novim kartama*. (SO)
5. **Sistem** prikazuje **radniku** poruku: “**Sistem** je zapamtio *karte*”. (IA)



**Alternativna scenarija**

3.1. Ukoliko **sistem** ne može da kreira *nove karte* on prikazuje **radniku** poruku: “**Sistem** ne može da kreira *nove karte*”. Prekida se izvršavanje scenarija. (IA)

8.1. Ukoliko **sistem** ne može da zapamti podatke o *novim kartama* on prikazuje **radniku** poruku “**Sistem** ne može da zapamti *nove karte*”. (IA)

### Projektovanje kontrolera korisničkog interfejsa

Kontroler korisničkog interfejsa je odgovoran za:

* prihvatanje grafičkih objekata od ekranske forme,
* konvertovanje podataka koji se nalaze u grafičkim objektima u domenske objekte koji će biti prosleđeni preko mreže do aplikacionog servera
* konvertovanje domenskih objekata u grafičke objekte i prosleđuje ih do ekranske forme.

## Projektovanje aplikacione logike

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

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

### Komunikacija sa klijentima

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

Klijent šalje zahtev za izvršenje neke od sistemskih operacija do odgovarajuće niti koja je povezana sa tim klijentom. Ta nit prihvata zahtev i prosleđuje ga do kontrolera aplikacione logike. Nakon izvršenja sistemske operacije rezultat se, preko kontrolera aplikacione logike, vraća do niti klijenta koja taj rezultat šalje nazad do klijenta.

Komunikacija između klijenta i servera se obavlja razmenom objekta klase “KlijentskiTransferObjekat” i klasa “ServerObjectTransfer”:

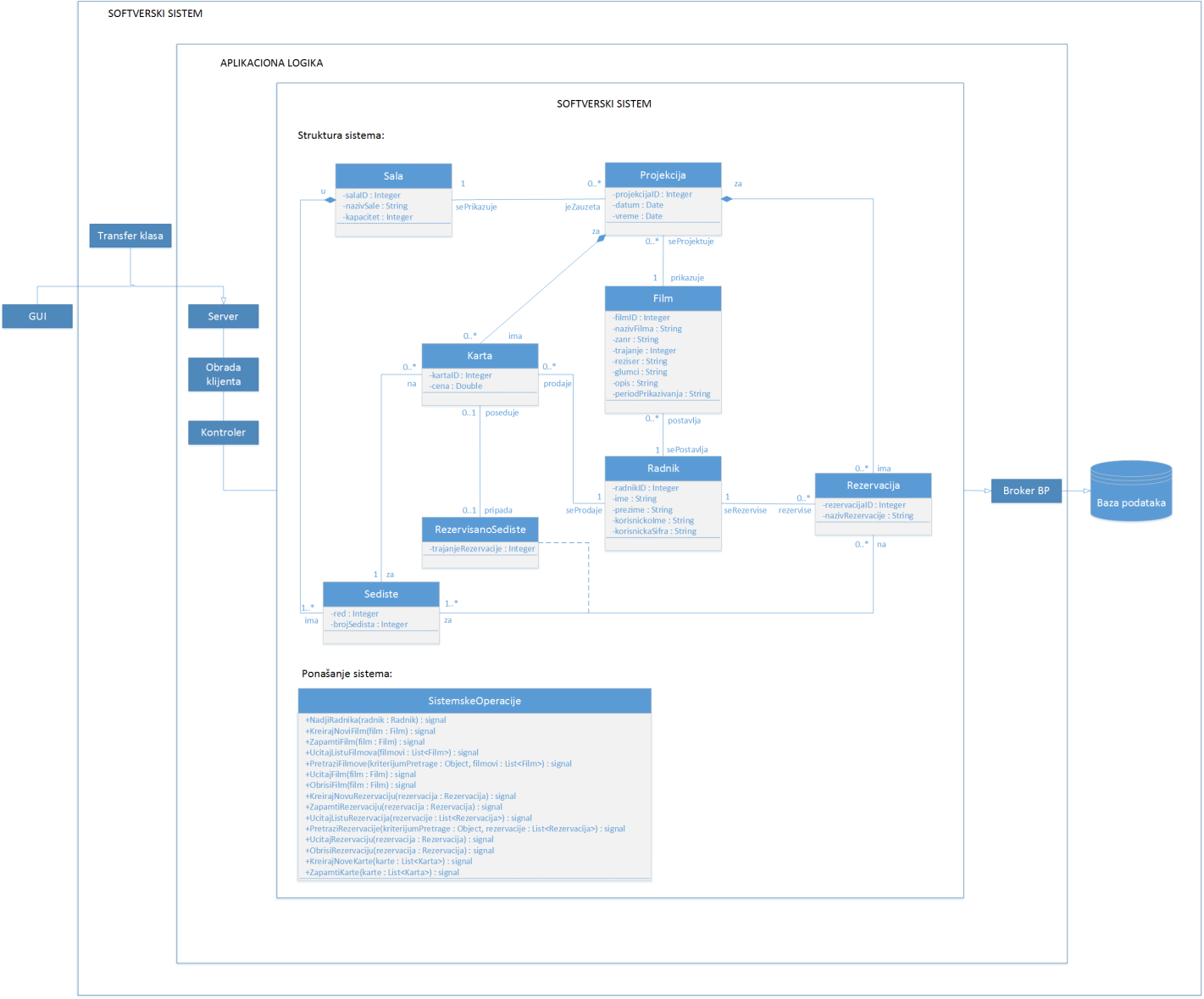
|  |
| --- |
| public class KlijentskiTransferObjekat implements Serializable{  private int operacija;  private Object parametar;  public KlijentskiTransferObjekat() {  }  public KlijentskiTransferObjekat(int operacija, Object parametar) {  this.operacija = operacija;  this.parametar = parametar;  }  public Object getParametar() {  return parametar;  }  public void setParametar(Object parametar) {  this.parametar = parametar;  }  public int getOperacija() {  return operacija;  }  public void setOperacija(int operacija) {  this.operacija = operacija;  }  } |

|  |
| --- |
| public class ServerskiTransferObjekat implements Serializable{  private Object podaci;  private int uspesnostIzvrsenjaOperacije;  private Exception exception;  public Object getPodaci() {  return podaci;  }  public void setPodaci(Object podaci) {  this.podaci = podaci;  }  public int getUspesnostIzvrsenjaOperacije() {  return uspesnostIzvrsenjaOperacije;  }  public void setUspesnostIzvrsenjaOperacije(int uspesnostIzvrsenjaOperacije) {  this.uspesnostIzvrsenjaOperacije = uspesnostIzvrsenjaOperacije;  }  public Exception getException() {  return exception;  }  public void setException(Exception exception) {  this.exception = exception;  }  } |

### Kontroler aplikacione logike

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

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.



### Poslovna logika

#### *Projektovanje ponašanja softverskog sistema – sistemske operacije*

Za svaku sistemsku operaciju treba napraviti konceptualna rešenja koja su direktno povezana sa logikom problema.

Za svaki od ugovora projektuje se konceptualno rešenje.

**Ugovor UG1: NadjiRadnika**

Operacija: NadjiRadnika(*Radnik*):signal;

Veza sa SK: SK1

Preduslovi: -

Postuslovi: *-*



**Ugovor UG2: UcitajListuRadnika**

Operacija: UcitajListuRadnika(*List<Radnik>*):signal;

Veza sa SK: SK3, SK6, SK9

Preduslovi: -

Postuslovi: -



**Ugovor UG3: KreirajNoviFilm**

Operacija: KreirajNoviFilm(*Film*):signal;

Veza sa SK: SK2

Preduslovi: *Prosto vrednosno ograničenje nad objektom Film mora biti zadovoljeno.*

*Strukturno ograničenje nad objektom Film mora biti zadovoljeno.*

Postuslovi: *Kreiran je novi film.*



**Ugovor UG4: ZapamtiFilm**

Operacija: ZapamtiFilm(*Film*):signal;

Veza sa SK: SK2, SK4

Preduslovi: *Prosto vrednosno ograničenje nad objektom Film mora biti zadovoljeno.*

*Strukturno ograničenje nad objektom Film mora biti zadovoljeno.*

Postuslovi: *Podaci o filmu su zapamćeni.*



**Ugovor UG5: PretraziFilmove**

Operacija: PretraziFilmove(*KriterijumPretrage,List<Film>*):signal;

Veza sa SK: SK3, SK4, SK5

Preduslovi: -

Postuslovi: -



**Ugovor UG6: UcitajFilm**

Operacija: UcitajFilm(*Film*):signal;

Veza sa SK: SK3, SK4

Preduslovi: -

Postuslovi: -



**Ugovor UG7:** **ObrisiFilm**

Operacija: ObrisiFilm(*Film*):signal;

Veza sa SK: SK5

Preduslovi: -

Postuslovi: *Film je obrisan. Strukturno ograničenje nad objektom Film mora biti zadovoljeno.*



**Ugovor UG8: UcitajListuProjekcija**

Operacija: UcitajListuProjekcija(*List<Projekcija>*):signal;

Veza sa SK: SK6, SK9

Preduslovi: -

Postuslovi: -



**Ugovor UG9: KreirajNovuRezervaciju**

Operacija: KreirajNovuRezervaciju(*Rezervacija*):signal;

Veza sa SK: SK6

Preduslovi: *Ukoliko je popunjen kapacitet sale za izabrani termin ne može se izvršiti sistemska operacija.*

*Prosto vrednosno ograničenje nad objektom Rezervacija mora biti zadovoljeno.*

*Strukturno ograničenje nad objektom Rezervacija mora biti zadovoljeno.*

Postuslovi: *Kreirana je nova rezervacija.*



**Ugovor UG10: ZapamtiRezervaciju**

Operacija: ZapamtiRezervaciju(*Rezervacija*):signal;

Veza sa SK: SK6

Preduslovi: *Ukoliko je popunjen kapacitet sale za izabrani termin ne može se izvršiti sistemska operacija.*

*Prosto vrednosno ograničenje nad objektom Rezervacija mora biti zadovoljeno.*

*Strukturno ograničenje nad objektom Rezervacija mora biti zadovoljeno.*

Postuslovi: *Podaci o rezervaciji su zapamćeni.*



**Ugovor UG11: PretraziRezervacije**

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

Veza sa SK: SK7, SK8

Preduslovi: -

Postuslovi: -



**Ugovor UG12: UcitajRezervaciju**

Operacija: UcitajRezervaciju(*Rezervacija*):signal;

Veza sa SK: SK7

Preduslovi: -

Postuslovi: -



**Ugovor UG13: ObrisiRezervaciju**

Operacija: ObrisiRezervaciju(*Rezervacija*):signal;

Veza sa SK: SK8

Preduslovi:

Postuslovi: *Rezervacija je obrisana. Strukturno ograničenje nad objektom Rezervacija mora biti zadovoljeno.*



**Ugovor UG14: UcitajListuSedista**

Operacija: UcitajListuSedista(*List<Sediste>*):signal;

Veza sa SK: SK9

Preduslovi: -

Postuslovi: -



**Ugovor UG15: KreirajNoveKarte**

Operacija: KreirajNoveKarte(List<*Karta>*):signal;

Veza sa SK: SK9

Preduslovi: *Ukoliko sedišta nisu rezervisana i popunjen je kapacitet sale za izabrani termin ne može se izvršiti sistemska operacija.*

*Prosto vrednosno ograničenje nad objektom Karta mora biti zadovoljeno.*

*Strukturno ograničenje nad objektom Karta mora biti zadovoljeno*

Postuslovi: *Kreirane su nove karte*



**Ugovor UG16: ZapamtiKarte**

Operacija: ZapamtiKarte(List<*Karta>*):signal;

Veza sa SK: SK9

Preduslovi: *Ukoliko sedišta nisu rezervisana i popunjen je kapacitet sale za izabrani termin ne može se izvršiti sistemska operacija.*

*Prosto vrednosno ograničenje nad objektom Karta mora biti zadovoljeno.*

*Strukturno ograničenje nad objektom Karta mora biti zadovoljeno*

Postuslovi: *Podaci o kartama su zapamćeni.*



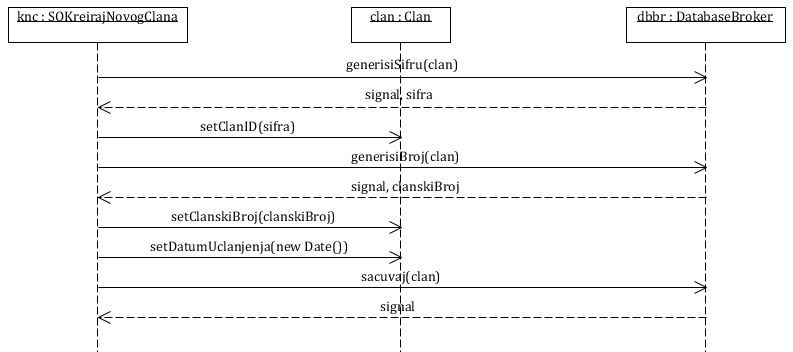
**UGOVOR UG1: KreirajNovogClana**

Operacija: KreirajNovogClana(*Clan*):signal;

Veza sa SK: SK1

Preduslovi: -

Postuslovi: *Kreiran je novi član.*



**UGOVOR UG2: ZapamtiClana**

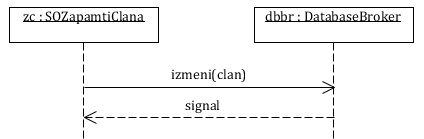
Operacija: ZapamtiClana(*Clan*):signal;

Veza sa SK: SK1, SK2

Preduslovi: *Prosto vrednosno ograničenje nad objektom Clan mora biti zadovoljeno.*

*Strukturno ograničenje nad objektom Clan mora biti zadovoljeno.*

Postuslovi: *Podaci o članu su zapamćeni.*



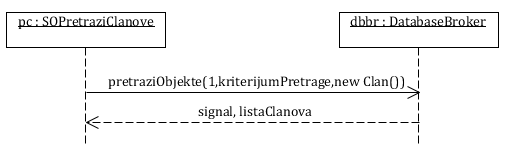
**UGOVOR UG3: PretraziClanove**

Operacija: PretraziClanove(*KriterijumPretrage,List<Clan>*):signal;

Veza sa SK: SK2, SK3, SK4

Preduslovi: -

Postuslovi: -



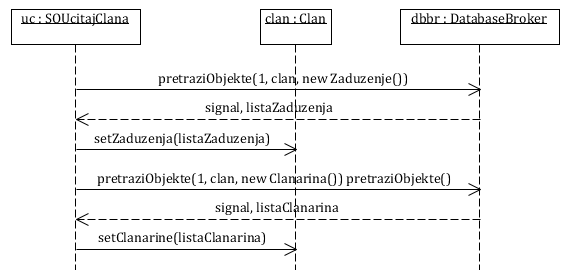
**UGOVOR UG4: UcitajClana**

Operacija: UcitajClana(*Clan*):signal;

Veza sa SK: SK2, SK4

Preduslovi: -

Postuslovi: -



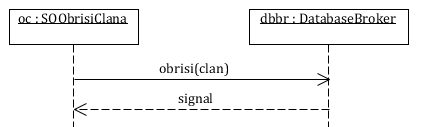
**UGOVOR UG5:** **ObrisiClana**

Operacija: ObrisiClana(*Clan*):signal;

Veza sa SK: SK3

Preduslovi: -

Postuslovi: *Član je obrisan. Strukturno ograničenje nad objektom Clan mora biti zadovoljeno.*



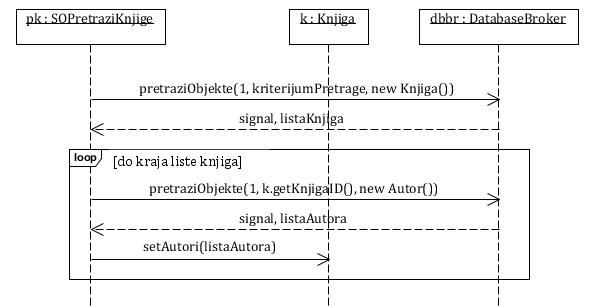
**UGOVOR UG6: PretraziKnjige**

Operacija: PretraziKnjige(*KriterijumPretrage, List<Knjiga>*):signal;

Veza sa SK: SK5

Preduslovi: -

Postuslovi: -



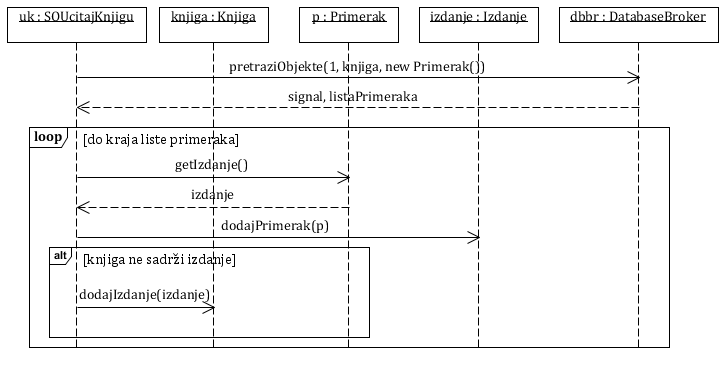
**UGOVOR UG7: UcitajKnjigu**

Operacija: UcitajKnjigu(*Knjiga*):signal;

Veza sa SK: SK5

Preduslovi: -

Postuslovi: -



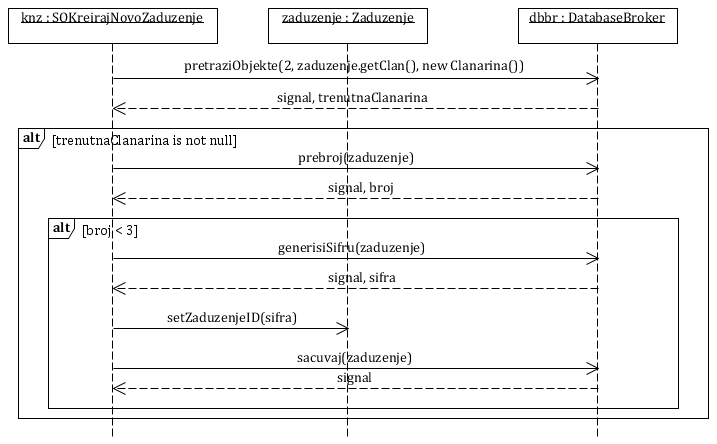
**UGOVOR UG8: KreirajNovoZaduzenje**

Operacija: KreirajNovoZaduzenje(*Zaduzenje*):signal;

Veza sa SK: SK6

Preduslovi: *Ako za datog člana postoje već tri zaduženja koja nisu razdužena ili ukoliko nije uplaćena članarina za dati period ne može se izvršiti sistemska operacija.*

Postuslovi: *Kreirano je novo zaduženje.*



**UGOVOR UG9: ZapamtiZaduzenje**

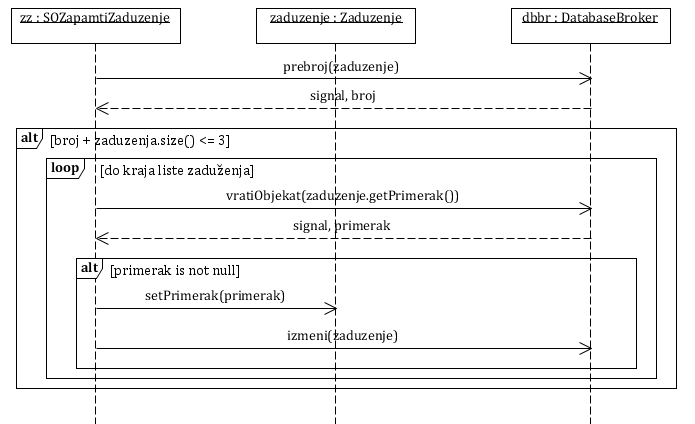
Operacija: ZapamtiZaduzenje(List<*Zaduzenje>*):signal;

Veza sa SK: SK6, SK7

Preduslovi: *Ako za datog člana postoje već tri zaduženja koja nisu razdužena ne može se izvršiti sistemska operacija. Prosto vrednosno ograničenje nad objektom Zaduzenje mora biti zadovoljeno.*

*Strukturno ograničenje nad objektom Zaduzenje mora biti zadovoljeno.*

Postuslovi: *Podaci o zaduženju su zapamćeni.*



**UGOVOR UG10: PretraziZaduzenja**

Operacija: PretraziZaduzenja(*KriterijumPretrage*, *List<Zaduzenje>*):signal;

Veza sa SK: SK7, SK8

Preduslovi: -

Postuslovi: -

****

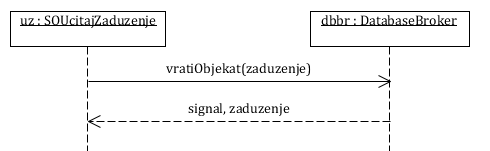
**UGOVOR UG11: UcitajZaduzenje**

Operacija: UcitajZaduzenje(*Zaduzenje*):signal;

Veza sa SK: SK7, SK8

Preduslovi: -

Postuslovi: -



**UGOVOR UG12: ZapamtiClanarinu**

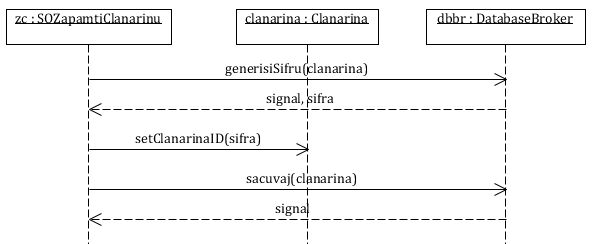
Operacija: ZapamtiClanarinu(*Clanarina*):signal;

Veza sa SK: SK9

Preduslovi: *Prosto vrednosno ograničenje nad objektom Clanarina mora biti zadovoljeno.*

*Strukturno ograničenje nad objektom Clanarina mora biti zadovoljeno.*

Postuslovi: *Podaci o članarini su zapamćeni.*



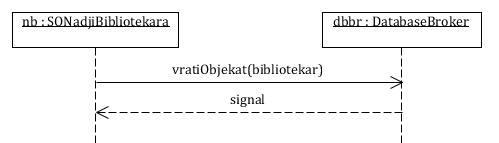
**UGOVOR UG13: NadjiBibliotekara**

Operacija: NadjiBibliotekara(*Bibliotekar*):signal;

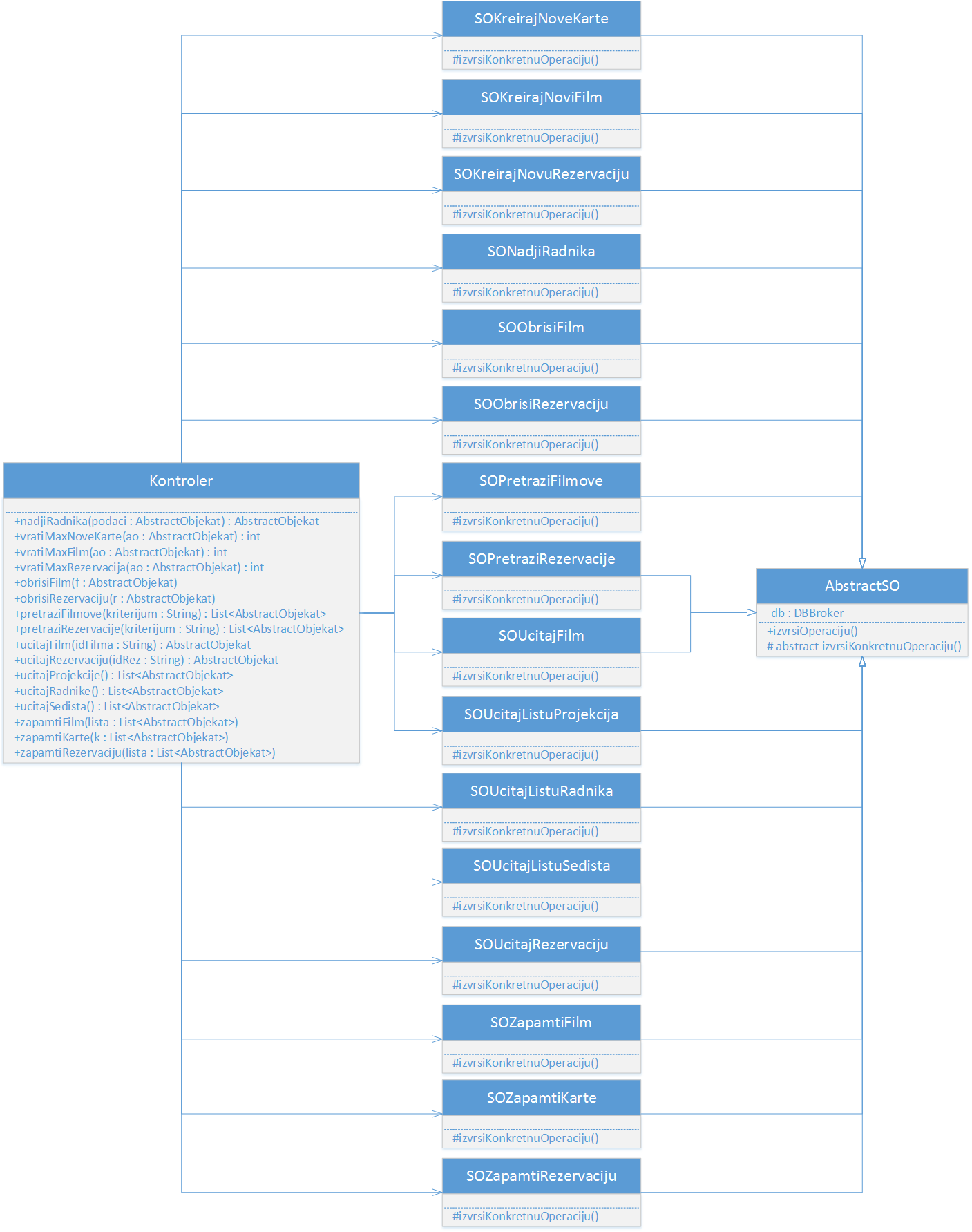
Veza sa SK: SK10

Preduslovi: -

Postuslovi: -



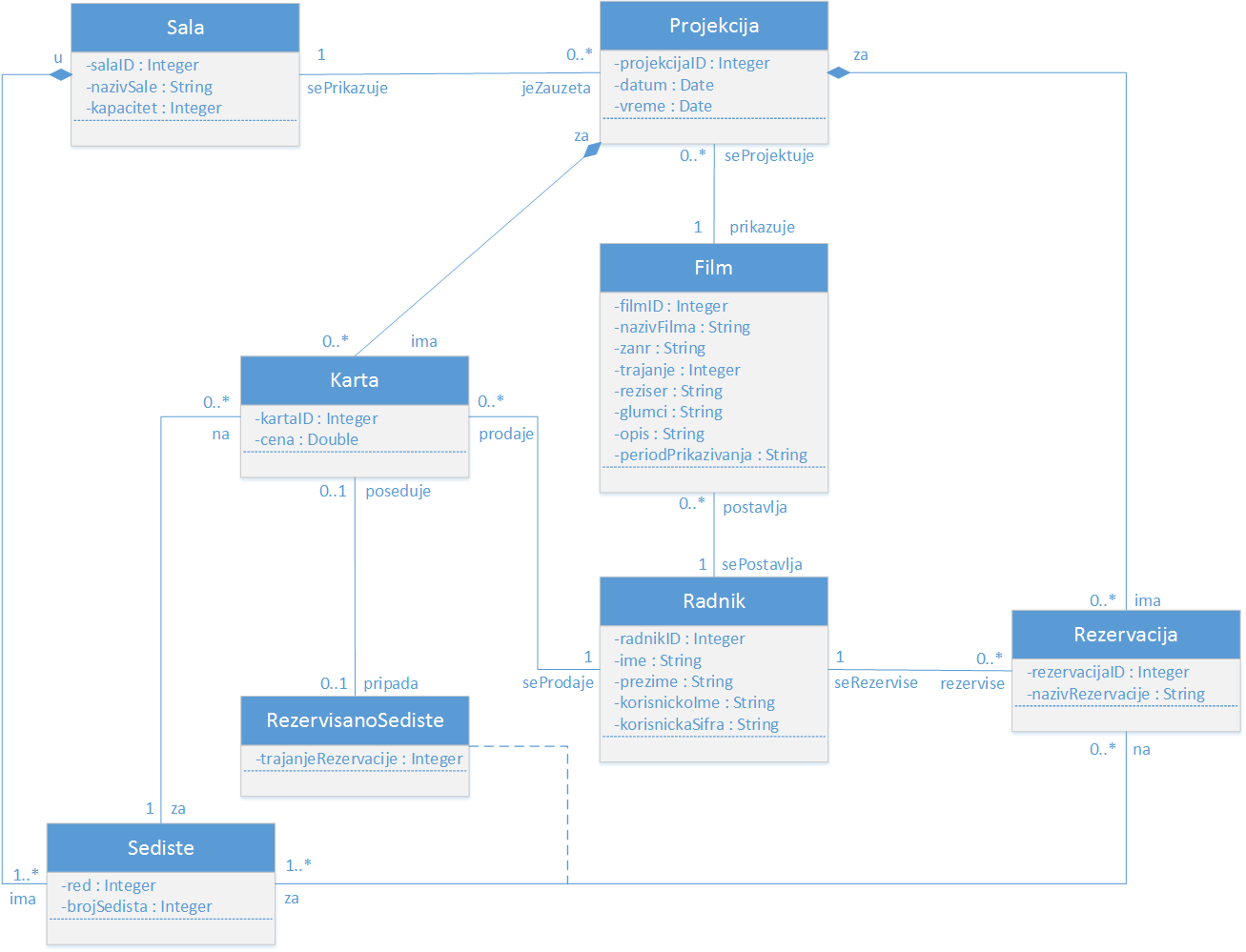
Klase koje su odgovorne za izvršenje sistemskih operacija nasleđuju klasu AbstractSO kako bi mogle da se povežu sa bazom i kako bi se njihovo izvršenje pratilo kao transakcija. AbstractSO predstavlja apstraktnu klasu čija glavna metoda (izvrsiOperaciju()) u sebi sadrži otvaranje konekcije sa bazom, proveru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



Slika 4. Dijagram klasa koji pokazuje vezu između kontrolera aplikacione logike i klasa odgovornih za izvršenje sistemskih operacija koje nasleđuju klasu AbstractSO

#### *Projektovanje strukture softverskog sistema*

Na osnovu konceptualnih klasa prave se softverske klase strukture. Svaka klasa ima privatna polja atributra, gettere i settere za te atribute, besparametaski konstruktor kao i parametarski konstruktor.



Svaka klasa ima privatna polja atributa, gettere i settere za te atribute, besparametarski konstruktor, kao i parametarski konstruktor. Kao primer:

|  |
| --- |
| public class Film extends AbstractObjekat{  private int filmID;  private String nazivFilma;  private String zanr;  private int trajanje;  private String reziser;  private String glumci;  private String opis;  private String periodPrikazivanja;  private Radnik radnik;  public Film() {  }  public Film(int filmID, String nazivFilma, String zanr, int trajanje, String reziser, String glumci, String opis, String periodPrikazivanja, Radnik radnik) {  this.filmID = filmID;  this.nazivFilma = nazivFilma;  this.zanr = zanr;  this.trajanje = trajanje;  this.reziser = reziser;  this.glumci = glumci;  this.opis = opis;  this.periodPrikazivanja = periodPrikazivanja;  this.radnik = radnik;  }  public int getFilmID() {  return filmID;  }  public void setFilmID(int filmID) {  this.filmID = filmID;  }  public String getNazivFilma() {  return nazivFilma;  }  public void setNazivFilma(String nazivFilma) {  this.nazivFilma = nazivFilma;  }  public String getZanr() {  return zanr;  }  public void setZanr(String zanr) {  this.zanr = zanr;  }  public int getTrajanje() {  return trajanje;  }  public void setTrajanje(int trajanje) {  this.trajanje = trajanje;  }  public String getReziser() {  return reziser;  }  public void setReziser(String reziser) {  this.reziser = reziser;  }  public String getGlumci() {  return glumci;  }  public void setGlumci(String glumci) {  this.glumci = glumci;  }  public String getOpis() {  return opis;  }  public void setOpis(String opis) {  this.opis = opis;  }  public String getPeriodPrikazivanja() {  return periodPrikazivanja;  }  public void setPeriodPrikazivanja(String periodPrikazivanja) {  this.periodPrikazivanja = periodPrikazivanja;  }  public Radnik getRadnik() {  return radnik;  }  public void setRadnik(Radnik radnik) {  this.radnik = radnik;  }  @Override  public String toString() {  return "Film{" + "filmID=" + filmID + ", nazivFilma=" + nazivFilma + ", zanr=" + zanr + ", trajanje=" + trajanje + ", reziser=" + reziser + ", glumci=" + glumci + ", opis=" + opis + ", periodPrikazivanja=" + periodPrikazivanja + ", radnik=" + radnik + '}';  }    @Override  public String vratiImeTabele() {  return "film";  }  @Override  public String vratiParametre() {  return filmID + ", '" + nazivFilma + "', '" + zanr + "', " + trajanje + ", '" + reziser+ "', '" + glumci + "', '" + opis + "', '" + periodPrikazivanja + "', " + getRadnik().getRadnikID() + "" ;  }  @Override  public String vratiPK() {  return "filmID";  }  @Override  public int vratiVrednostPK() {  return filmID;  }  @Override  public List<AbstractObjekat> izRSuTabelu(ResultSet rs) {  List<AbstractObjekat> filmovi = new ArrayList<>();  try {  while (rs.next()) {  int filmID = rs.getInt("filmID");  String nazivFilma = rs.getString("nazivFilma");  String zanr = rs.getString("zanr");  int trajanje = rs.getInt("trajanje");  String reziser = rs.getString("reziser");  String glumci = rs.getString("glumci");  String opis = rs.getString("opis");  String periodPrikazivanja = rs.getString("periodPrikazivanja");  int radnikID = rs.getInt("radnikID");  Film f = new Film(filmID, nazivFilma, zanr, trajanje, reziser, glumci, opis, periodPrikazivanja, new Radnik(radnikID, null, null, null, null));  filmovi.add(f);  }  } catch (Exception e) {  System.out.println("Greska kod klijenata.");  }  return filmovi;  }  @Override  public String vratiUpdateUpit() {  return "nazivFilma='" + nazivFilma  + "',zanr='" + zanr  + "',trajanje=" + trajanje  + ",reziser='" + reziser  + "',glumci='" + glumci  + "',opis='" + opis  + "',periodPrikazivanja='" + periodPrikazivanja  + "',radnikID=" + getRadnik().getRadnikID() + "";  }  @Override  public String vratiSlozenPK() {  return "Nema";  }  } |

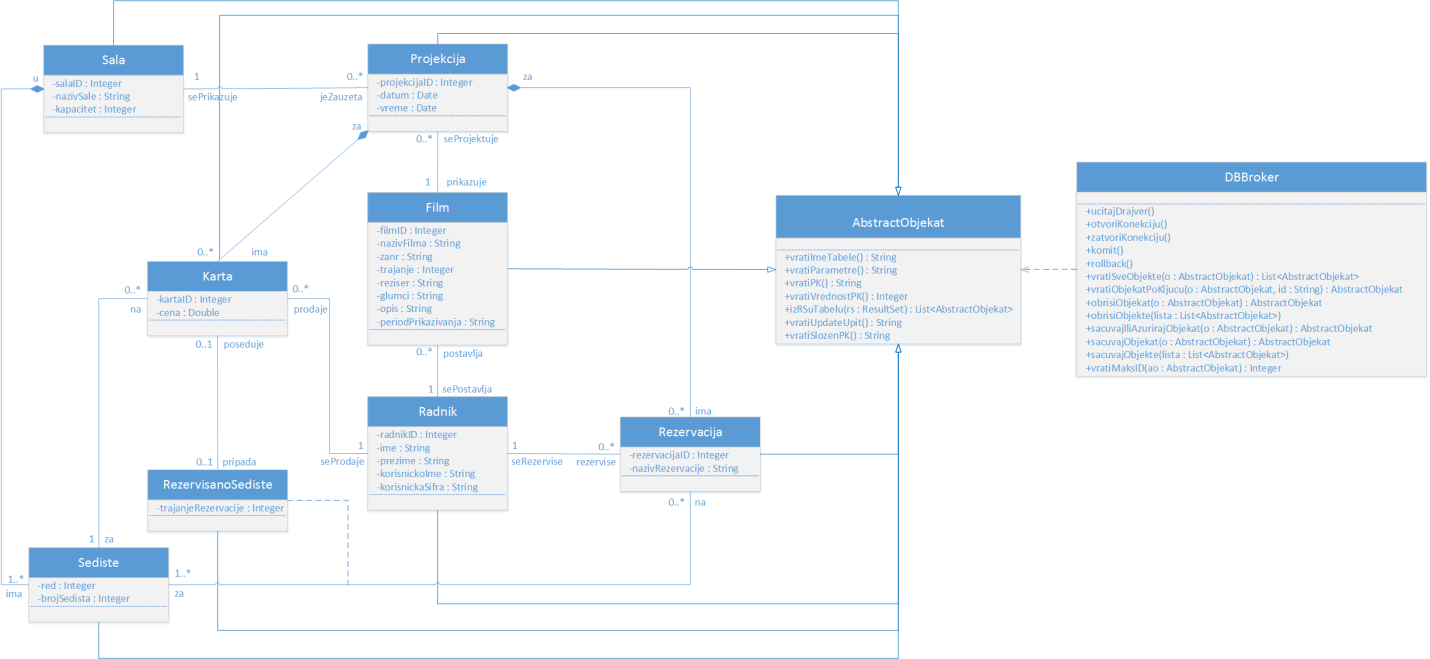
### Broker baze podataka

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

* public void ucitajDrajver ()
* public void otvoriKonekciju()
* public void zatvoriKonekciju()
* public void komit()
* public void rollback()
* public List<AbstractObjekat> vratiSveObjekte(AbstractObjekat o)
* public AbstractObjekat vratiObjekatPoKljucu(AbstractObjekat o, String id)
* public AbstractObjekat obrisiObjekat(AbstractObjekat o)
* public void obrisiObjekte(List<AbstractObjekat> lista)
* public AbstractObjekat sacuvajIliAzurirajObjekat(AbstractObjekat o)
* public AbstractObjekat sacuvajObjekat(AbstractObjekat o)
* public void sacuvajObjekte(List<AbstractObjekat> lista)
* public int vratiMaksID(AbstractObjekat ao)

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 klase “ AbstractObjekat ” koga nasleđuju (kao i sve njegove metode) sve domenske klase:

|  |
| --- |
| public abstract class AbstractObjekat implements Serializable{  public abstract String vratiImeTabele();  public abstract String vratiParametre();  public abstract String vratiPK();  public abstract int vratiVrednostPK();  public abstract List<AbstractObjekat> izRSuTabelu(ResultSet rs);  public abstract String vratiUpdateUpit();  public abstract String vratiSlozenPK();  } |



Slika 5. DBBroker se povezuje sa klasom AbstractObjekat

## Projektovanje skladišta podataka

Na osnovu softverskih klasa strukture projektovane su tabele (skladišta podataka) relacionog sistema za upravljanje bazom podataka (MySQL):

|  |  |  |
| --- | --- | --- |
| **Tabela Radnik** | | |
| **Ime** | **Tip atributa** | **Veličina** |
| radnikID | Int | 7 |
| ime | Text | 20 |
| prezime | Text | 20 |
| korisnickoIme | Text | 20 |
| korisnickaSifra | Text | 20 |

|  |  |  |
| --- | --- | --- |
| **Tabela Film** | | |
| **Ime** | **Tip atributa** | **Veličina** |
| filmID | Int | 7 |
| nazivFilma | Text | 50 |
| zanr | Text | 50 |
| trajanje | Int | 7 |
| reziser | Text | 50 |
| glumci | Text | 50 |
| opis | Text | 240 |
| periodPrikazivanja | Text | 30 |
| radnikID | Int | 7 |

|  |  |  |
| --- | --- | --- |
| **Tabela Sala** | | |
| **Ime** | **Tip atributa** | **Veličina** |
| salaID | Int | 7 |
| nazivSale | Text | 20 |
| kapacitet | Int | 7 |

|  |  |  |
| --- | --- | --- |
| **Tabela Projekcija** | | |
| **Ime** | **Tip atributa** | **Veličina** |
| projekcijaID | Int | 7 |
| datum | Date | / |
| vreme | Date | / |
| FilmID | Int | 7 |
| SalaID | Int | 7 |

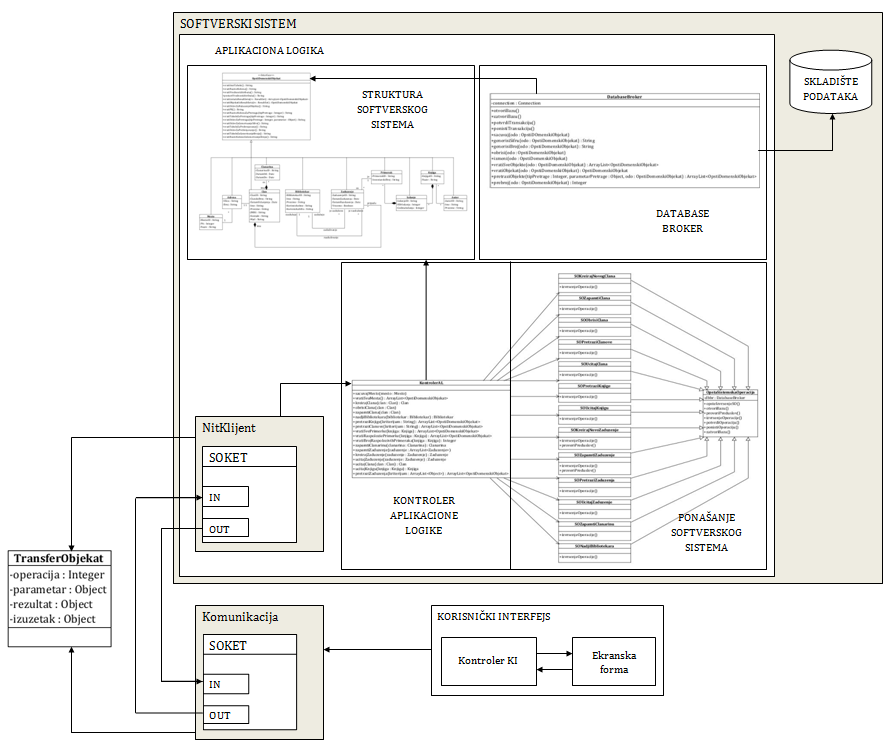
|  |  |  |
| --- | --- | --- |
| **Tabela Sediste** | | |
| **Ime** | **Tip atributa** | **Veličina** |
| salaID | Int | 7 |
| red | Int | 7 |
| brojSedista | Int | 7 |

|  |  |  |
| --- | --- | --- |
| **Tabela Karta** | | |
| **Ime** | **Tip atributa** | **Veličina** |
| projekcijaID | Int | 7 |
| kartaID | Int | 7 |
| cena | Double | / |
| radnikID | Int | 7 |
| salaID | Int | 7 |
| red | Int | 7 |
| brojSedista | Int | 7 |

|  |  |  |
| --- | --- | --- |
| **Tabela Rezervacija** | | |
| **Ime** | **Tip atributa** | **Veličina** |
| projekcijaID | Int | 7 |
| rezervacijaID | Int | 7 |
| nazivRezervacije | Text | 20 |
| radnikID | Int | 7 |

|  |  |  |
| --- | --- | --- |
| **Tabela RezervisanoSediste** | | |
| **Ime** | **Tip atributa** | **Veličina** |
| projekcijaID | Int | 7 |
| rezervacijaID | Int | 7 |
| salaID | Int | 7 |
| red | Int | 7 |
| brojSedista | Int | 7 |
| trajanjeRezervacije | Text | 7 |

|  |  |  |
| --- | --- | --- |
| **Tabela KartaZaRezervisanoSediste** | | |
| **Ime** | **Tip atributa** | **Veličina** |
| projekcijaID | Int | 7 |
| rezervacijaID | Int | 7 |
| salaID | Int | 7 |
| red | Int | 7 |
| brojSedista | Int | 7 |
| KartaID | Int | 7 |



Slika 6. Arhitektura softverskog sistema

# Implementacija

Softverski sistem, rezultat ovog rada, razvijen je u programskom jeziku Java i projektovan je kao klijent-server. Kao razvojno okruženje koriščen je NetBeans IDE 8.0.2, a kao sistem za upravljanje bazom podataka korišćen je MySQL.

Čitav sistem je realizovan u tri projekta: BioskopKlijent, BioskopServer i BioskopZajednicki. Na osnovu arhitekture softverskog sistema dobijene su sledeće softverske klase:

- BioskopKlijent

* forme/FormaKupovinaKarata.java
* forme/FormaPostavljanjeFilma.java
* forme/FormaRepertoar.java
* forme/FormaRezervacijaKarata.java
* forme/Glavna.java
* komunikacija/Komunikacija.java
* kontroler/Kontroler.java
* model/ModelTabeleFilm.java
* model/ModelTabeleRediSedista.java
* model/ModelTabeleRezervacija.java
* model/ModelTabeleVremenaProjekcija.java
* sesija/Sesija.java
* start/KlijentStart.java

- BioskopServer

* db/DBBroker.java
* forma/FormaStart.java
* kontroler/Kontroler.java
* so/AbstractSO.java
* so/SOKreirajNoveKarte.java
* so/SOKreirajNoviFilm.java
* so/SOKreirajNovuRezervaciju.java
* so/SONadjiRadnika.java
* so/SOObrisiFilm.java
* so/SOObrisiRezervaciju.java
* so/SOPretraziFilmove.java
* so/SOPretraziRezervacije.java
* so/SOUcitajFilm.java
* so/SOUcitajListuProjekcija.java
* so/SOUcitajListuRadnika.java
* so/SOUcitajListuSedista.java
* so/SOUcitajRezervaciju.java
* so/SOZapamtiFilm.java
* so/SOZapamtiKarte.java
* so/SOZapamtiRezervaciju.java
* start/SocketServer.java
* thread/ClientThread.java

- BioskopZajednicki

* domen/AbstractObjekat.java
* domen/Film.java
* domen/Karta.java
* domen/KartaZaRezervisanoSediste.java
* domen/Projekcija.java
* domen/Radnik.java
* domen/Rezervacija.java
* domen/RezervisanoSediste.java
* domen/Sala.java
* domen/Sediste.java
* konstante/Konstante.java
* transfer/KlijentskiTransferObjekat.java
* transfer/ServerskiTransferObjekat.java

# Testiranje

Svaki od implementiranih slučajeva korišćenja je testiran. Prilikom testiranja svakog

slučaja korišćenja, pored unetih pravilnih podataka, unošeni su i nepravilni podaci da bi

se utvrdilo kakav će rezultat izvršenja biti.

Na osnovu izvršenih testiranja otklonjeni su uočeni nedostaci.

# Literatura

Dr Siniša Vlajić, *Projektovanje softvera (skripta)*, Beograd, 2011.