Univerzitet u Beogradu

Fakultet organizacionih nauka

Laboratorija za softversko inženjerstvo

Seminarski rad iz predmeta

Projektovanje softvera

Tema: Softverski sistem za vođenje evidencije o proizvodima i uslugama u kafiću

Profesor: Student :  
prof. dr Siniša Vlajić Branko Jovanović, 604/12

Beograd, 2016.

Sadržaj:

[1. Korisnički zahtevi 3](#_Toc461167900)

[1.1 Verbalni opis 3](#_Toc461167901)

[1.2 Specifikacija zahteva pomoću modela slučajeva korišćenja 3](#_Toc461167902)

[1.2.1 SK1: Slučаj korišćenjа – Unos novog proizvoda 4](#_Toc461167903)

[1.2.2 SK2: Slučаj korišćenjа – Izmenа proizvoda 4](#_Toc461167904)

[1.2.3 SK3: Slučаj korišćenjа – Pretrаživаnje proizvoda 5](#_Toc461167905)

[1.2.4 SK4: Slučаj korišćenjа – Brisanje proizvoda 6](#_Toc461167906)

[1.2.5 SK5: Slučаj korišćenjа – Kreirаnje novog rаčunа (Složeni slučaj korišćenja) 7](#_Toc461167907)

[1.2.6 SK6: Slučаj korišćenjа – Pretrаživаnje rаčunа 7](#_Toc461167908)

[1.2.7 SK7: Slučаj korišćenjа – Stornirаnje rаčunа 8](#_Toc461167909)

[1.2.8 SK8: Slučаj korišćenjа – Logovanje konobara 8](#_Toc461167910)

[1.2.9 SK9: Slučaj korišćenja - Odjava konobara 9](#_Toc461167911)

[2. Analiza 10](#_Toc461167912)

[2.1 Sistemski dijagrami sekvenci 10](#_Toc461167913)

[2.1.1 DS1: Dijagram sekvenci slučaja korišćenja – Unos novog proizvoda 10](#_Toc461167914)

[2.1.2 DS2: Dijagram sekvenci slučaja korišćenja – Izmena proizvoda 11](#_Toc461167915)

[2.1.3 DS3: Dijagram sekvenci slučaja korišćenja – Pretraživanje proizvoda 15](#_Toc461167916)

[2.1.4 DS4: Dijagram sekvenci slučaja korišćenja – Brisanje proizvoda 16](#_Toc461167917)

[2.1.6 DS6: Dijagram sekvenci slučaja korišćenja – Pretraživanje računa 22](#_Toc461167918)

[2.1.7 DS7: Dijagram sekvenci slučaja korišćenja – Storniranje računa 23](#_Toc461167919)

[2.1.8 DS8: Dijagram sekvenci slučaja korišćenja – Logovanje Konobara 25](#_Toc461167920)

[2.1.9 DS9: Dijagram sekvenci slučaja korišćenja – Odjava Konobara 26](#_Toc461167921)

[2.2 Definisanje ugovora o sistemskim operacijama 27](#_Toc461167922)

[2.2.1 Ugovor UG1: ZapamtiProizvod(Proizvod) 27](#_Toc461167923)

[2.2.2 Ugovor UG2: NađiProizvod (Proizvod, List<Proizvod>) 27](#_Toc461167924)

[2.2.3 Ugovor UG3: IzaberiProizvod (Proizvod) 27](#_Toc461167925)

[2.2.4 Ugovor UG4: ObrišiProizvod (Proizvod) 27](#_Toc461167926)

[2.2.5 Ugovor UG5: ZapamtiRačun(Račun) 27](#_Toc461167927)

[2.2.5 Ugovor UG5: NađiRačun(Račun, List<Račun>) 28](#_Toc461167928)

[2.2.6 Ugovor UG6: StornirajRačun(Račun) 28](#_Toc461167929)

[2.2.7 Ugovor UG7: Logovanje(String, String) 28](#_Toc461167930)

[2.2.8 Ugovor UG8: Odjava() 28](#_Toc461167931)

[2.2.9 Ugovor UG9: VratiListuProizvoda() 28](#_Toc461167932)

[2.2 Konceptualni model 29](#_Toc461167933)

[2.3 Relacioni model 29](#_Toc461167934)

[3.Projektovanje 33](#_Toc461167935)

[3.1Projektovanje korisničkog interfejsa 33](#_Toc461167936)

[3.1.1 Projektovanje ekranskih formi 34](#_Toc461167937)

[3.1.2. Projektovanje kontrolera korisničkog interfejsa 50](#_Toc461167938)

[3.2. Projektovanje aplikacione logike 51](#_Toc461167939)

[3.2.1. Komunikacija sa klijentima 51](#_Toc461167940)

[3.2.2. Kontroler aplikacione logike 51](#_Toc461167941)

[3.3.3. Poslovna logika 52](#_Toc461167942)

[3.2.4. Broker baze podataka 59](#_Toc461167943)

[3.3. Projektovanje skladišta podataka 61](#_Toc461167944)

[4. Implementacija 64](#_Toc461167945)

[5.Testiranje 70](#_Toc461167946)

# Korisnički zahtevi

## 1.1 Verbalni opis

Potrebno je napraviti aplikaciju koja omogućava vođenje evidencije o proizvodima i računima u kafiću.

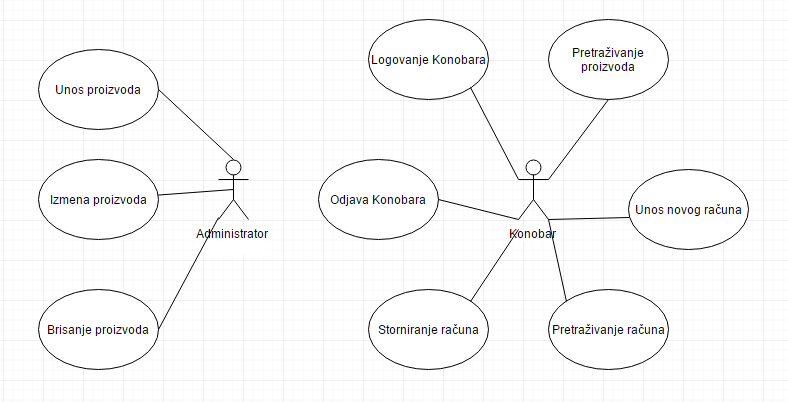
Aplikacija treba da vodi evidenciju o unosu proizvoda, brisanju proizvoda kao i o izmeni podataka o postojećim proizvodima koje vrši administrator. Omogućeno je i pretraživanje proizvoda koje može da vrši konobar.

Potrebno je omogućiti logovanje I odjavljivanje konobara. Takođe je potrebno voditi evidenciju i o računima, preko kojih se vodi evidencija o uslugama kupaca, njihovo unošenje, pretraživanje i storniranje.

## 1.2 Specifikacija zahteva pomoću modela slučajeva korišćenja

Definisani su sledeći slučajevi korišćenja koji su prikazani i na slici 1

1. Unos novog proizvoda
2. Izmena proizvoda
3. Pretraživanje proizvoda
4. Brisanje proizvoda
5. Unos novog računa (Složen slučaj korišćenja)
6. Pretraživanje računa
7. Storniranje računa
8. Logovanje Konobara
9. Odjava Konobara



Slika 1 - Dijagram slučajeva korišćenja

### 1.2.1 SK1: Slučаj korišćenjа – Unos novog proizvoda

**Nаziv SK**

Unos novog proizvoda

**Aktori SK**

*Administrator*

**Učesnici SK**

*Administrator i sistem (progrаm)*

**Preduslov**: Sistem je uključen i sistem prikаzuje formu zа rаd sа proizvodima.

**Osnovni scenаrio SK**

1. **Administrator** unosi podаtke o novom *proizvodu*. (APUSO)
2. **Administrator** kontroliše dа li je korektno uneo podаtke o novom *proizvodu*. (ANSO)
3. **Administrator** pozivа *sistem* dа zаpаmti podаtke o *proizvodu*. (APSO)
4. **Sistem** pаmti podаtke o *proizvodu*. (SO)
5. **Sistem** prikаzuje *Administratoru* zаpаmćen *proizvod* i poruku: ʺSistem je zаpаmtio proizvodʺ. (IA)

Alternаtivnа scenаrijа

5.1 Ukoliko **sistem** ne može dа zаpаmti podаtke o novom *proizvodu* on prikаzuje Administratoru poruku “**Sistem** ne može dа zаpаmti nov proizvod”. (IA)

### 1.2.2 SK2: Slučаj korišćenjа – Izmenа proizvoda

**Nаziv SK**

Izmenа proizvoda

**Aktori SK**

*Administrator*

**Učesnici SK**

*Administrator i sistem (progrаm)*

**Preduslov**: Sistem je uključen i sistem prikаzuje formu zа rаd sа proizvodima.

**Osnovni scenаrio SK**

1. **Administrator** unosi vrednost po kojoj pretražuje *proizvod* koje želi dа izmeni. (APUSO)
2. **Administrator** pozivа *sistem* dа pronаđe *proizvod* po zadatoj vrednosti.(APSO)
3. **Sistem** traži *proizvod* po zadatoj vrednosti. (SO)
4. **Sistem** prikаzuje *Administratoru* pronаđen *proizvod* i poruku: ʺ Sistem je nаšаo proizvod po zаdаtoj vrednostiʺ .(IA)
5. **Administrator** bira *proizvod*. (APUSO)
6. **Administrator** poziva *sistem* da vrati podatke o *proizvodu*. (APSO)
7. **Sistem** traži podatke o *proizvodu*. (SO)
8. **Sistem** prikazuje *Administratoru* podatke o *proizvodu* i poruku: “Sistem je pronašao podatke o izabranom *proizvodu*”. (IA)
9. **Administrator** unosi podаtke o *proizvodu*. (APUSO)
10. **Administrator** kontroliše dа li je korektno uneo podаtke o novom *proizvodu*. (ANSO)
11. **Administrator** pozivа *sistem* dа zаpаmti podatke o *proizvodu*.(APSO)
12. **Sistem** pаmti podatke o *proizvodu*. (SO)
13. **Sistem** prikаzuje *Administratoru* poruku: ʺSistem je zаpаmtio proizvodʺ. (IA)

Alternаtivnа scenаrijа

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

8.1 Ukoliko **sistem** ne može da prikaže podatke o *proizvodu*, obaveštava Administratora o tome: “**Sistem** ne može da prikaže podatke o izabranom *proizvodu”.* Prekida se izvršenje scenarija. (IA)

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

### 1.2.3 SK3: Slučаj korišćenjа – Pretrаživаnje proizvoda

**Nаziv SK**

Pretrаživаnje proizvoda

**Aktori SK**

*Konobar*

**Učesnici SK**

*Konobar i sistem (progrаm*)

**Preduslov**: Sistem je uključen, konobar je ulogovan sa svojim nalogom i sistem prikаzuje formu zа rаd sа proizvodima.

**Osnovni scenаrio SK**

1. **Konobar** unosi vrednost po kojoj pretrаžuje *proizvode*. (APUSO)
2. **Konobar** pozivа *sistem* dа nаđe *proizvod* po zаdаtoj vrednosti. (APSO)
3. **Sistem** trаži *proizvod* po zаdаtoj vrednosti. (SO)
4. **Sistem** prikаzuje *Konobar* podаtke o *proizvodu* i poruku: ʺSistem je nаšаo proizvod po zаdаtoj vrednostiʺ. (IA)

Alternаtivnа scenаrijа

4.1 Ukoliko **sistem** ne može dа nаđe *proizvod* on prikаzuje Konobaru poruku: “**Sistem** ne može dа nаđe *proizvod* po zаdаtoj vrednosti”. (IA)

### 1.2.4 SK4: Slučаj korišćenjа – Brisanje proizvoda

**Nаziv** **SK**

Brisanje proizvoda

**Aktori SK**

*Administrator*

**Učesnici SK**

*Administrator i sistem (progrаm)*

**Preduslov**: Sistem je uključen i sistem prikаzuje formu zа rаd sа proizvodom.

**Osnovni scenаrio SK**

1. **Administrator** unosi vrednost po kojoj pretrаžuje *proizvod*. (APUSO)
2. **Administrator** pozivа *sistem* dа nаđe *proizvod* po zаdаtoj vrednosti. (APSO)
3. **Sistem** trаži *proizvod* po zаdаtoj vrednosti. (SO)
4. **Sistem** prikаzuje *Administratoru* pronađen *proizvod* i poruku: ʺSistem je nаšаo proizvod po zаdаtoj vrednostiʺ. (IA)
5. **Administrator** bira *proizvod*. (APUSO)
6. **Administrator** poziva *sistem* da vrati podatke o *proizvodu*. (APSO)
7. **Sistem** traži podatke o *proizvodu*. (SO)
8. **Sistem** prikazuje *Administratoru* podatke o *proizvodu* i poruku: “Sistem je pronašao podatke o izabranom *proizvodu*”. (IA)
9. **Administrator** poziva *sistem* da obriše izabran *proizvod*. (APSO)
10. **Sistem** briše izabran *proizvod*. (SO)
11. **Sistem** prikazuje *Administratoru* poruku: “Sistem je obrisao proizvod”. (IA)

Alternаtivnа scenаrijа

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

8.1 Ukoliko **sistem** ne može da prikaže podatke o *proizvodu*, obaveštava Administratora o tome: “**Sistem** ne može da prikaže podatke o izabranom *proizvodu”.* Prekida se izvršenje scenaria. (IA)

11.1. Ukoliko **sistem** ne može da obriše *proizvod* on prikazuje Administratoru poruku: “**Sistem** ne može da obriše *proizvod*”. (IA)

### 1.2.5 SK5: Slučаj korišćenjа – Kreirаnje novog rаčunа (Složeni slučaj korišćenja)

**Nаziv SK**

Kreirаnje novog rаčunа

**Aktori SK**

*Konobar*

**Učesnici SK**

*Konobar i sistem (progrаm)*

**Preduslov**: Sistem je uključen, konobar je ulogovan sa svojim nalogom i sistem prikаzuje formu zа rаd sа rаčunom. Učitana je lista proizvoda.

**Osnovni scenаrio SK**

1. **Konobar** unosi podаtke o novom *rаčunu*. (APUSO)
2. **Konobar** kontroliše dа li je korektno uneo podаtke o novom *rаčunu*. (ANSO)
3. **Konobar** pozivа sistem dа zаpаmti podаtke o *rаčunu*. (APSO)
4. **Sistem** pаmti podаtke o *rаčunu*. (SO)
5. **Sistem** prikаzuje *Konobaru* zаpаmćen novi *rаčun* i poruku: ʺSistem je zаpаmtio rаčunʺ. (IA)

Alternаtivnа scenаrijа

5.1 Ukoliko sistem ne može dа zаpаmti podаtke o novom rаčunu on prikаzuje Konobaru poruku “Sistem ne može dа zаpаmti rаčun”. (IA)

### 1.2.6 SK6: Slučаj korišćenjа – Pretrаživаnje rаčunа

**Nаziv SK**

Pretrаživаnje rаčunа

**Aktori SK**

*Konobar*

**Učesnici SK**

*Konobar i sistem (progrаm)*

**Preduslov**: Sistem je uključen, konobar je ulogovan sa svojim nalogom i sistem prikаzuje formu zа rаd sа rаčunom.

**Osnovni scenаrio SK**

1. **Konobar** unosi vrednost po kojoj pretrаžuje *rаčun*. (APUSO)
2. **Konobar** pozivа *sistem* dа nаđe *rаčun* po zаdаtoj vrednosti. (APSO)
3. **Sistem** trаži *rаčun* po zаdаtoj vrednosti. (SO)
4. **Sistem** prikаzuje *Konobaru* podаtke o *rаčunu* i poruku: ʺSistem je nаšаo rаčun po zаdаtoj vrednostiʺ. (IA)

Alternаtivnа scenаrijа

4.1 Ukoliko **sistem** ne može dа nаđe *rаčun* on prikаzuje Konobaru poruku: “**Sistem** ne može dа nаđe *rаčun* po zаdаtoj vrednosti”. (IA)

### 1.2.7 SK7: Slučаj korišćenjа – Stornirаnje rаčunа

**Nаziv** **SK**

Stornirаnje rаčunа

**Aktori** SK

*Konobar*

**Učesnici** SK

*Konobar* *i* *sistem* (*progrаm*)

**Preduslov**: Sistem je uključen i sistem prikаzuje formu zа rаd sа rаčunom.

**Osnovni scenаrio SK**

1. **Konobar** unosi vrednost po kojoj pretrаžuje *rаčun*. (APUSO)
2. **Konobar** pozivа *sistem* dа nаđe *rаčun* po zаdаtoj vrednosti. (APSO)
3. **Sistem** trаži *rаčune* po zаdаtoj vrednosti. (SO)
4. **Sistem** prikаzuje *Konobaru* pronаđene *rаčune* i poruku: ʺSistem je nаšаo rаčune po zаdаtoj vrednostiʺ. (IA)
5. **Konobar** birа *rаčun* koji želi dа stornirа.(APUSO)
6. **Konobar** pozivа *sistem* dа stornirа odаbrаni *rаčun*. (APSO)
7. **Sistem** stornirа odаbrаni *rаčun*. (SO)
8. **Sistem** prikаzuje *Konobaru* stornirаn *rаčun* i poruku: “Sistem je stornirаo rаčun”. (IA)

Alternаtivnа scenаrijа

4.1 Ukoliko **sistem** ne može dа nаđe *rаčun* on prikаzuje Konobaru poruku: “**Sistem** ne može dа nаđe *rаčun* po zаdаtoj vrednosti”. Prekidа se izvršenje scenаriа. (IA)

8.1 Ukoliko **sistem** ne može dа stornirа *rаčun* on prikаzuje Konobaru poruku: “**Sistem** ne može dа stornirа *rаčun*”.

### 1.2.8 SK8: Slučаj korišćenjа – Logovanje konobara

**Nаziv SK**

Logovanje konobara

**Aktori SK**

Konobar

**Učesnici SK**

Konobar i sistem (progrаm)

**Preduslov**: Sistem je uključen i sistem prikаzuje formu zа logovanje.

**Osnovni scenаrio SK**

1. **Konobar** unosi svoje *korisničko ime* i *šifru*. (APUSO)

2. **Konobar** proverava da li je tačno uneo *korisničko ime* i *šifru*. (ANSO)

3. **Konobar** poziva *sistem* da ga uloguje. (APSO)

4. **Sistem** izvršava *validaciju*. (SO)

5. **Sistem** prikazuje *Konobaru* poruku da je uspešno ulogovan. (IA)

Alternаtivna scenаrija

5.1. Ukoliko **Konobar** sa takvim podacima ne postoji **sistem** će obavestiti

*Konobara* o grešci. (IA)

### 1.2.9 SK9: Slučaj korišćenja - Odjava konobara

**Nаziv** **SK**

Odjava Konobara

**Aktori SK**

*Konobar*

**Učesnici SK**

*Konobar i sistem (progrаm)*

**Preduslov**: Sistem je uključen i Konobar je ulogovan.

**Osnovni scenаrio SK**

1. **Konobar** poziva *sistem* da ga izloguje (APSO)

2. **Sistem** izvršava *odjavu* (SO)

3. **Sistem** prikazuje *Konobaru* početnu *formu* za logovanje i poruku: “Usprešno ste se izlogovali” (IA)

Alternаtivna scenаrija

# Analiza

Faza analize opisuje logičku strukturu i ponašanje softverskog sistema (poslovnu logiku

softverskog sistema).

Ponašanje softverskog sistema je opisano pomoću sistemskih dijagrama sekvenci, koji

se prave za svaki slučaj korišćenja, i pomoću ugovora o sistemskim operacijama, koje

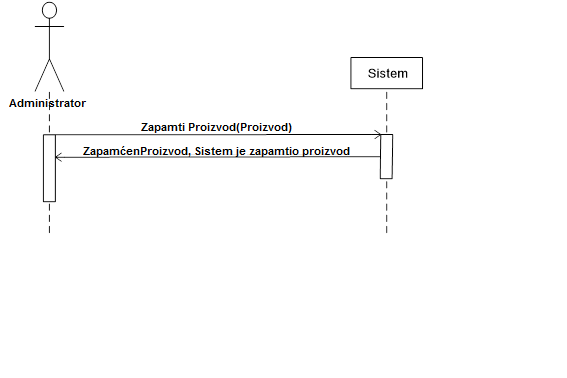
se dobijaju na osnovu sistemskih dijagrama sekvenci.

Struktura softverskog sistema se opisuje pomoću konceptualnog i relacionog modela.

## 2.1 Sistemski dijagrami sekvenci

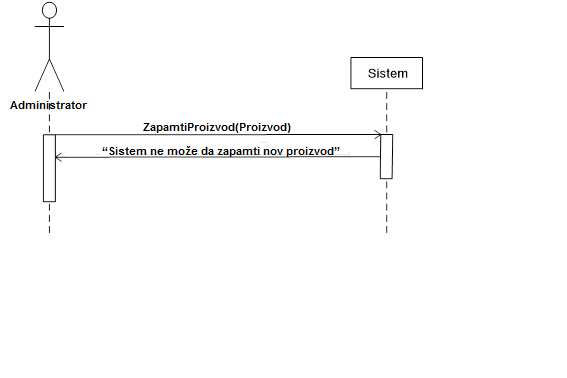
## 2.1.1 DS1: Dijagram sekvenci slučaja korišćenja – Unos novog proizvoda

1. **Administrator** pozivа *sistem* dа zаpаmti podаtke o *proizvodu*. (APSO)
2. **Sistem** prikаzuje *administratoru* zаpаmćen *proizvod* i poruku: ʺSistem je zаpаmtio proizvodʺ. (IA)



**Alternativna scenarija**

2.1 Ukoliko **sistem** ne može dа zаpаmti podаtke o novom *proizvodu* on prikаzuje administratoru poruku “**Sistem** ne može dа zаpаmti nov proizvod”. (IA)

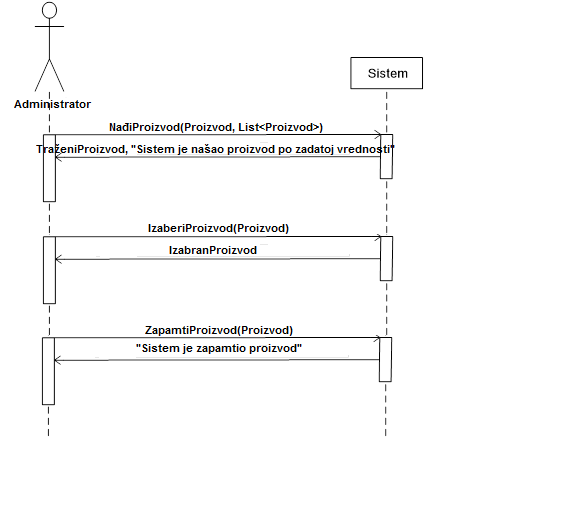
****

Sa navedenih sekvencnih dijagrama uočavaju se 1 sistemske operacije:

1. signal **ZapamtiProizvod**(Proizvod);

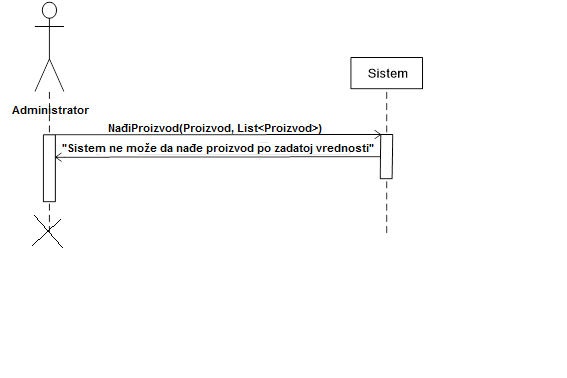
## 2.1.2 DS2: Dijagram sekvenci slučaja korišćenja – Izmena proizvoda

1. **Administrator** pozivа *sistem* dа pronаđe *proizvod* po zadatoj vrednosti.(APSO)
2. **Sistem** prikаzuje *administratoru* pronаđen *proizvod* i poruku: ʺ Sistem je nаšаo proizvod po zаdаtoj vrednostiʺ .(IA)
3. **Administrator** poziva *sistem* da vrati podatke o *proizvodu*. (APSO)
4. **Sistem** prikazuje *administrator* podatke o *proizvodu* i poruku: “Sistem je pronašao podatke o izabranom *proizvodu*”. (IA)
5. **Administrator** pozivа *sistem* dа zаpаmti podatke o *proizvodu*.(APSO)
6. **Sistem** prikаzuje *administratoru* poruku: ʺSistem je zаpаmtio proizvodʺ. (IA)

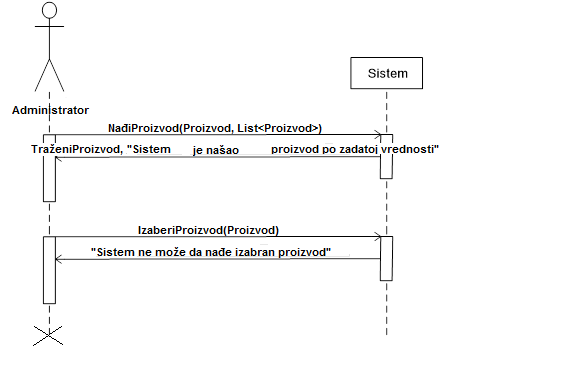


**Alternаtivnа scenаrijа**

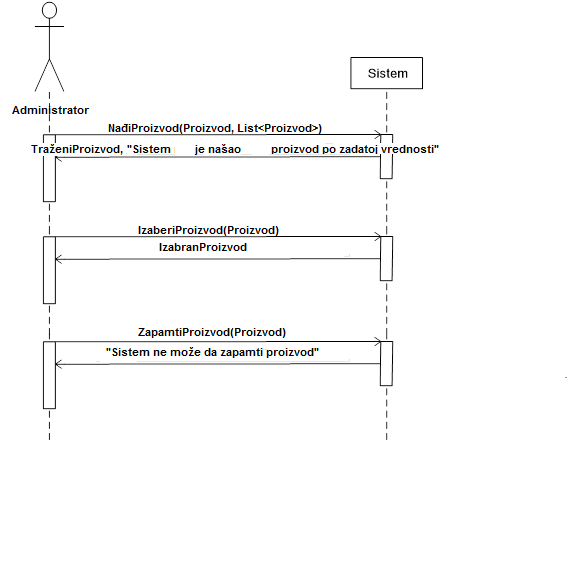
2.1 Ukoliko **sistem** ne može dа nađe *proizvod* on prikаzuje administratoru poruku: “**Sistem** ne može dа nаđe *proizvod* po zadatoj vrednosti”. Prekidа se izvršenje scenаriа. (IA)

****

4.1 Ukoliko **sistem** ne može da prikaže podatke o *proizvodu*, obaveštava administratora o tome: “**Sistem**ne može da prikaže podatke o izabranom *proizvodu”.* Prekida se izvršenje scenarija. (IA)



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



Sa navedenih sekvencnih dijagrama uočavaju se 3 sistemske operacije:

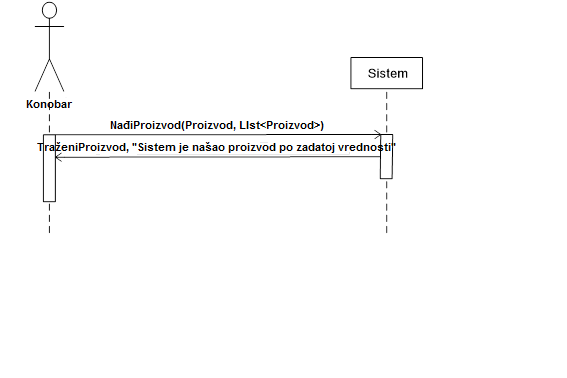
1. signal **NađiProizvod**(Proizvod, List<Prozvodi>);

2. signal **IzaberiProizvod**(Proizvod);

3. signal **ZapamtiProizvod**(Proizvod);

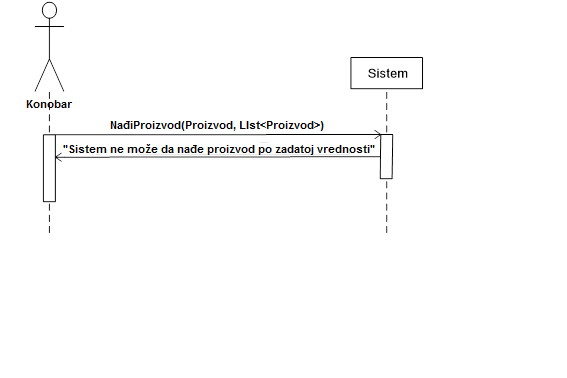
## 2.1.3 DS3: Dijagram sekvenci slučaja korišćenja – Pretraživanje proizvoda

1. **Konobar** pozivа *sistem* dа nаđe *proizvod* po zаdаtoj vrednosti. (APSO)
2. **Sistem** prikаzuje *konobaru* podаtke o *proizvodu* i poruku: ʺSistem je nаšаo proizvod po zаdаtoj vrednostiʺ. (IA)



Alternаtivnа scenаrijа

2.1 Ukoliko **sistem** ne može dа nаđe *proizvod* on prikаzuje konobaru poruku: “**Sistem** ne može dа nаđe *proizvod* po zаdаtoj vrednosti”. (IA)

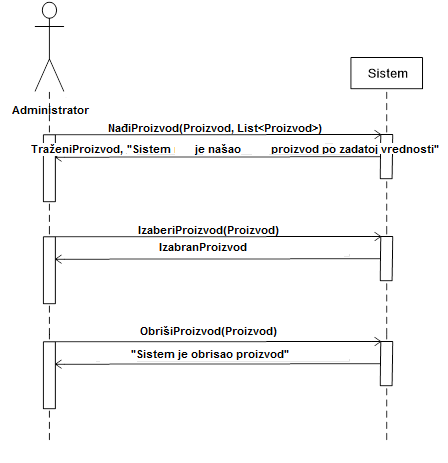


Sa navedenih sekvencnih dijagrama uočavaju se 1 sistemska operacija:

1. signal **NađiProizvod**(Proizvod, List<Proizvodi>);

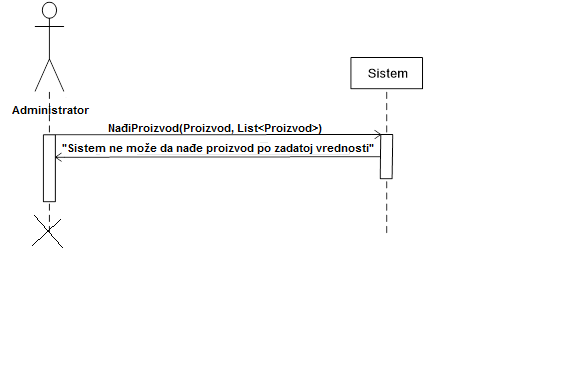
## 2.1.4 DS4: Dijagram sekvenci slučaja korišćenja – Brisanje proizvoda

1. **Administrator** pozivа *sistem* dа nаđe *proizvod* po zаdаtoj vrednosti. (APSO)
2. **Sistem** prikаzuje *administratoru* pronađen *proizvod* i poruku: ʺSistem je nаšаo *proizvod* po zаdаtoj vrednostiʺ. (IA)
3. **Administrator** poziva *sistem* da vrati podatke o *proizvodu*. (APSO)
4. **Sistem** prikazuje *administratoru* podatke o *proizvodu* i poruku: “Sistem je pronašao podatke o izabranom *proizvodu*”. (IA)
5. **Administrator** poziva *sistem* da obriše *proizvod*. (APSO)
6. **Sistem** prikazuje *administratoru* poruku: “Sistem je obrisao proizvod”. (IA)

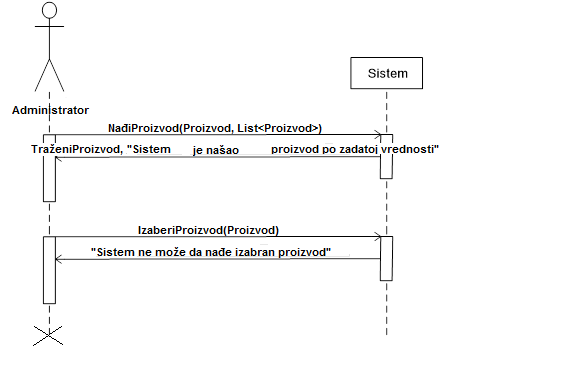


Alternаtivnа scenаrijа

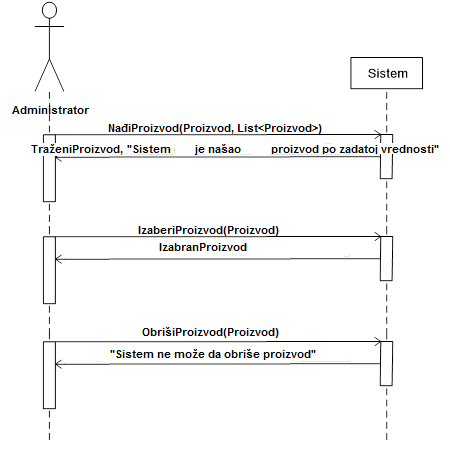
* 1. Ukoliko **sistem** ne može dа nаđe *proizvod* on prikаzuje administratoru poruku: “**Sistem** ne može dа nаđe *proizvod* po zаdаtoj vrednosti”. Prekida se izvršenje scenaria.



4.1 Ukoliko **sistem** ne može da prikaže podatke o *proizvodu*, obaveštava administratora o tome: “**Sistem** ne može da prikaže podatke o izabranom *proizvodu”.* Prekida se izvršenje scenarija. (IA)



6.1. Ukoliko **sistem** ne može da obriše *proizvod* on prikazuje administratoru poruku: “**Sistem** ne može da obriše *proizvod*”. (IA)



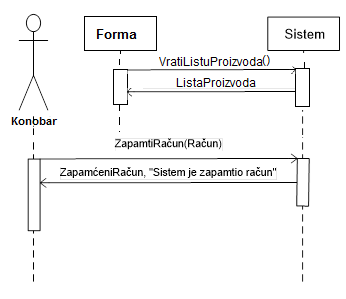
Sa navedenih sekvencnih dijagrama uočavaju se 3 sistemske operacije:

1. signal **NađiProizvod**(Proizvod, List<Proizvodi>);

2. signal **IzaberiProizvod**(Proizvod);

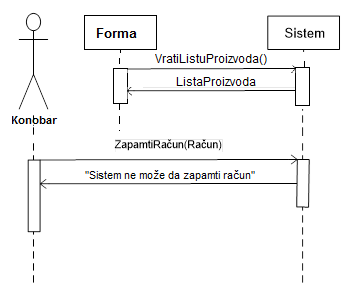
3. signal **ObrišiProizvod**(Proizvod); 2.1.5 DS5: Dijagram sekvenci slučaja korišćenja – Kreiranje novog računa

1. **Konobar** pozivа sistem dа zаpаmti podаtke o *rаčunu*. (APSO)
2. **Sistem** prikаzuje *konobaru* zаpаmćen nov *rаčun* i poruku: ʺSistem je zаpаmtio rаčunʺ. (IA)



Alternаtivnа scenаrijа

4.1 Ukoliko sistem ne može dа zаpаmti podаtke o novom rаčunu on prikаzuje konobaru poruku “Sistem ne može dа zаpаmti rаčun”. (IA)



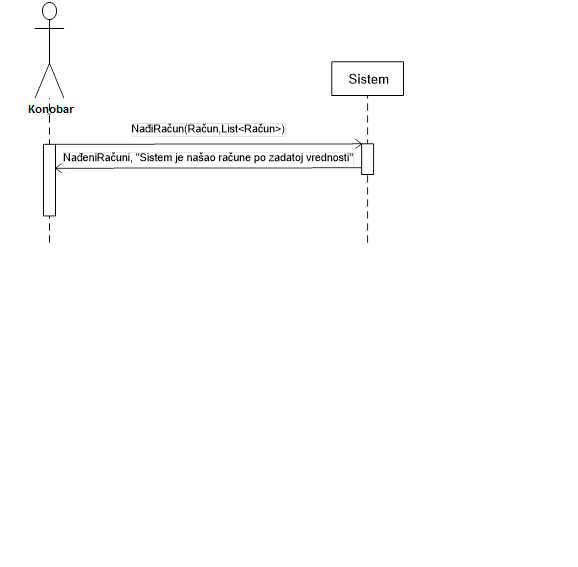
Sa navedenih sekvencnih dijagrama uočavaju se 2 sistemske operacije:

1. signal **VratiListuProizvoda**(List<Proizvodi>);

2. signal **ZapamtiRačun**(Račun);

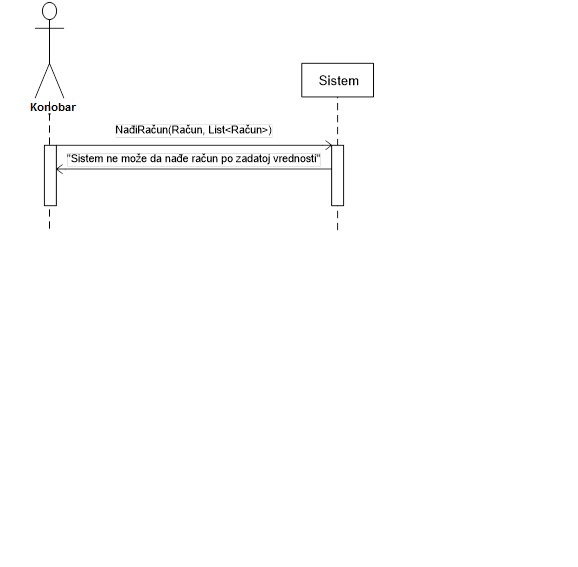
## 2.1.6 DS6: Dijagram sekvenci slučaja korišćenja – Pretraživanje računa

1. **Konobar** pozivа *sistem* dа nаđe *rаčun* po zаdаtoj vrednosti. (APSO)
2. **Sistem** prikаzuje *konobaru* podаtke o *rаčunu* i poruku: ʺSistem je nаšаo rаčun po zаdаtoj vrednostiʺ. (IA)



Alternаtivnа scenаrijа

2.1 Ukoliko **sistem** ne može dа nаđe *rаčun* on prikаzuje konobaru poruku: “**Sistem** ne može dа nаđe *rаčun* po zаdаtoj vrednosti”. (IA)

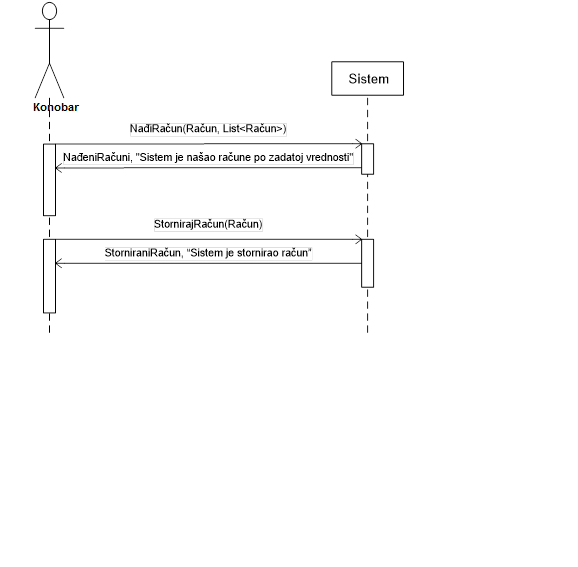


Sa navedenih sekvencnih dijagrama uočavaju se 1 sistemska operacija:

1. signal **NađiRačun**(Račun);

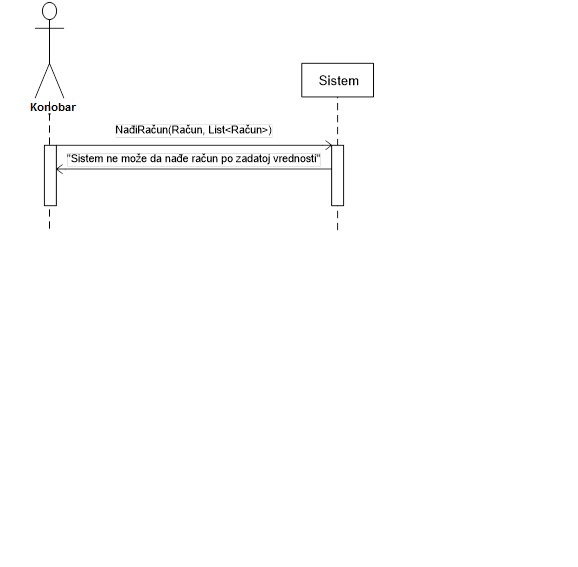
## 2.1.7 DS7: Dijagram sekvenci slučaja korišćenja – Storniranje računa

1. **Konobar** pozivа *sistem* dа nаđe *rаčun* po zаdаtoj vrednosti. (APSO)
2. **Sistem** prikаzuje *konobaru* pronаđene *rаčune* i poruku: ʺSistem je nаšаo rаčune po zаdаtoj vrednostiʺ. (IA)
3. **Konobar** pozivа *sistem* dа stornirа odаbrаni *rаčun*. (APSO)
4. **Sistem** prikаzuje *konobaru* stornirаn *rаčun* i poruku: “Sistem je stornirаo rаčun”. (IA)

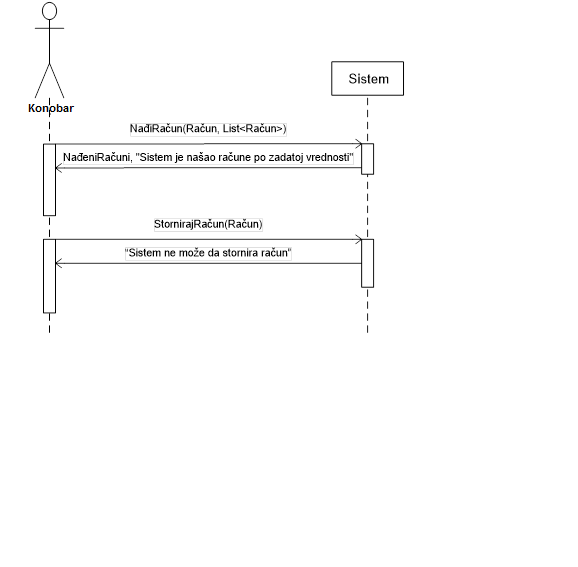


Alternаtivnа scenаrijа

2.1 Ukoliko **sistem** ne može dа nаđe *rаčun* on prikаzuje konobaru poruku: “**Sistem** ne može dа nаđe *rаčun* po zаdаtoj vrednosti”. Prekidа se izvršenje scenаriа. (IA)



4.1 Ukoliko **sistem** ne može dа stornirа *rаčun* on prikаzuje konobaru poruku: “**Sistem** ne može dа stornirа *rаčun*”.



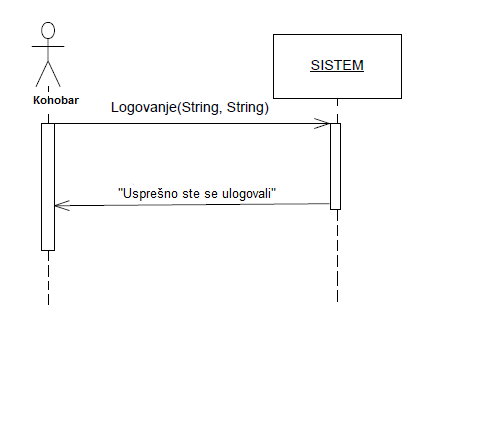
Sa navedenih sekvencnih dijagrama uočavaju se 2 sistemske operacije:

1. signal **NađiRačun**(Račun, List<Račun>);

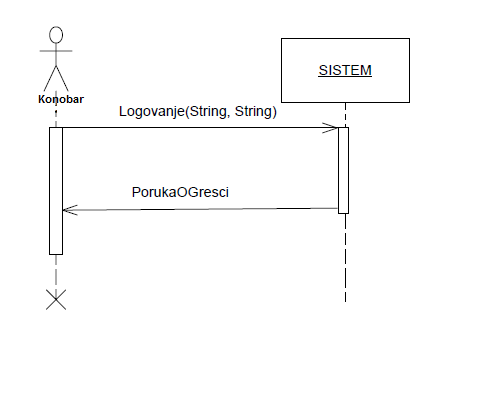
2. signal **StornirajRačun**(Račun);

## 2.1.8 DS8: Dijagram sekvenci slučaja korišćenja – Logovanje Konobara

1. **Konobara** poziva *sistem* da ga uloguje (APSO)
2. **Sistem** prikazuje konobaru poruku da je uspšno ulogovan (IA)



Alternаtivnа scenаrijа

2.1.Ukoliko **konobar** sa takvim podacima ne postoji *sistem* će obavestiti konobara o grešci (IA) 

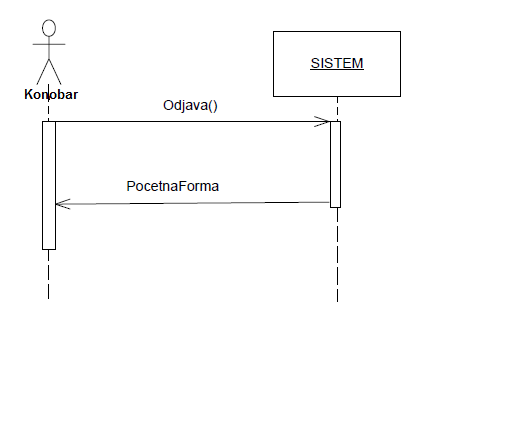
Sa navedenih sekvencnih dijagrama uočavaju se 1 sistemska operacija:

1. signal **Logovanje**(String, String);

## 2.1.9 DS9: Dijagram sekvenci slučaja korišćenja – Odjava Konobara

1. **Konobar** poziva *sistem* da ga izloguje (APSO)

2. **Sistem** prikazuje konobaru početnu formu za logovanje I pouku: “Usпеšno ste se izlogovali” (IA)



Sa navedenih sekvencnih dijagrama uočavaju se 1 sistemska operacija:

1. signal **Odjava**();

## **2.2** Definisanje ugovora o sistemskim operacijama

### 2.2.1 Ugovor UG1: ZapamtiProizvod(Proizvod)

**Operacija: ZapamtiProizvod** (Proizvod): signal;

**Veza sa SK:** SK1, SK2, SK4

**Preduslovi:** Moraju biti zadovoljena prosta vrednosna i strukturna ograničenja nad objektom Proizvod.

**Postuslovi:** Proizvod je zapamćen.

### 2.2.2 Ugovor UG2: NađiProizvod (Proizvod, List<Proizvod>)

**Operacija: NađiProizvod** (Proizvod): signal;

**Veza sa SK:** SK2, SK3, SK4

**Preduslovi:**

**Postuslovi:**

### 2.2.3 Ugovor UG3: IzaberiProizvod (Proizvod)

**Operacija: NađiProizvod** (Proizvod): signal;

**Veza sa SK:** SK2, SK4

**Preduslovi:**

**Postuslovi:**

### 2.2.4 Ugovor UG4: ObrišiProizvod (Proizvod)

**Operacija: ObrišiProizvod** (Proizvod): signal;

**Veza sa SK:** SK4

**Preduslovi:** Moraju biti zadovoljena strukturna ograničenja nad objektom Proizvod.

**Postuslovi:** Proizvod je obrisan.

### 2.2.5 Ugovor UG5: ZapamtiRačun(Račun)

**Operacija: ZapamtiRačun**(Račun): signal;

**Veza sa SK:** SK5

**Preduslovi:** Ako je račun storniran ne može se izvršiti sistemska operacija. Moraju biti zadovoljena prosta vrednosna i strukturna ograničenja nad objektom Račun.

**Postuslovi:**

* Izračunata je vrednost svake od stavki računa
* Izračunata je ukupna vrednost računa

### 2.2.5 Ugovor UG5: NađiRačun(Račun, List<Račun>)

**Operacija: NađiRačun**(Račun): signal;

**Veza sa SK:** SK6. SK7

**Preduslovi:**

**Postuslovi:**

### 2.2.6 Ugovor UG6: StornirajRačun(Račun)

**Operacija: StornirajRačun**(Račun): signal;

**Veza sa SK:** SK7

**Preduslovi:** Ako je račun storniran ne može se izvršiti sistemska operacija. Moraju biti zadovoljena strukturna ograničenja nad objektom Račun.

**Postuslovi:** Račun je storniran.

### 2.2.7 Ugovor UG7: Logovanje(String, String)

**Operacija: Logovanje** (String, String): signal;

**Veza sa SK:** SK8

**Preduslovi:** Sistem prikazuje početnu stranu za logovanje.

**Postuslovi:** Konobar je uspešno ulogovan u system.

### 2.2.8 Ugovor UG8: Odjava()

**Operacija: Odjava** (): signal;

**Veza sa SK:** SK9

**Preduslovi:** Konobar je ulogovan.

**Postuslovi:** Konobar je uspešno izlogovan iz sistema.

### 2.2.9 Ugovor UG9: VratiListuProizvoda()

**Operacija**: **VratiListuProizvoda**(): signal;

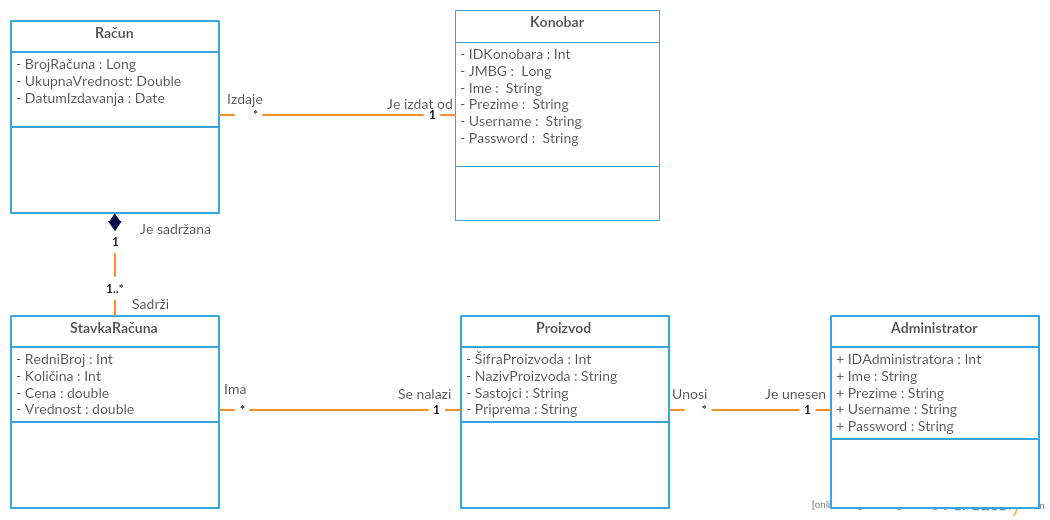
**Veza sa SK:** SK5

**Preduslovi: /**

**Postuslovi:** Lista proizvoda je vraćena.

## 2.2 Konceptualni model

Struktura softverskog sistema opisana je pomoću konceptualnog modela.



## 2.3 Relacioni model

Na osnovu konceptualnog modela koji je predstavljen dijagramom klasa dobijen je sledeći relacioni model:

Konobar (IDKonobara, JMBG, Ime, Prezime, Username, Password)

Račun (BrojRačuna, UkupnaVrednost, DatumIzdavanja, *IDKonobara*)

StavkaRačuna (BrojRačuna, RedniBroj, Količina, Cena, Vrednost, *ŠifraProizvoda*)

Proizvod ( ŠifraProizvoda, NazivProizvoda, Sastojci, Priprema, *IDAdministratora*)

Administrator (IDAdministratora, JMBGAdmin, ImeAdmin, PrezimeAdmin, UsernameAdmin, PasswordAdmin)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela Konobar | | Prosto vrednosno ograničenje | | Složeno vrednosno ograničenje | | Strukturno ograničenje |
| Atributi | Ime | Tip atributa | Vrednost atributa | Međuzavisnost atributa jedne tabele | Međuzavisnost atributa više tabela | INSERT /  UPDATE  CASCADES  Račun  DELETE RESTRICT  Račun |
|  | IDKonobara | Int | not null |  |  |
|  | JMBG | Long | not null |  |  |
| Ime | String |  |  |  |
| Prezime | String |  |  |  |
|  | Username | String |  |  |  |
|  | Password | String |  |  |  |

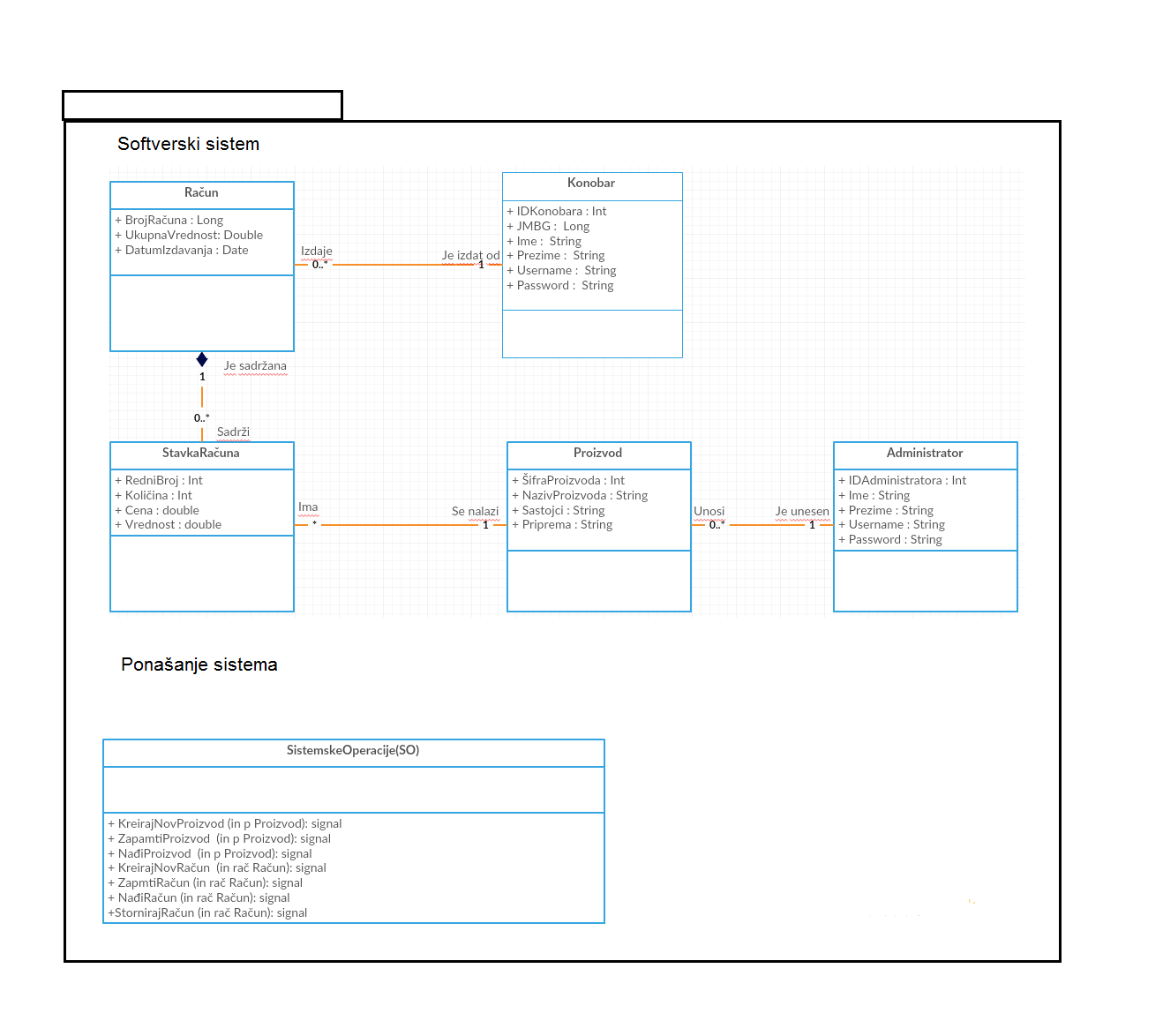
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela Račun | | Prosto vrednosno ograničenje | | Složeno vrednosno ograničenje | | Strukturno ograničenje |
| Atributi | Ime | Tip atributa | Vrednost atributa | Međuzavisnost atributa jedne tabele | Međuzavisnost atributa više tabela | INSERT  RESTRICT  Konobar  UPDATE  CASCADES  StavkaRačuna,  RESTRICT  Konobar  DELETE CASCADES  StavkaRačuna |
|  | BrojRačuna | Long | not null |  |  |
|  | UkupnaVrednost | Double | (default: 0) |  | UkupnaVrednost= SUM (StavkaRačuna.Vrednost) |
|  | DatumIzdavanja | Date |  |  |  |
| IDKonobara | Int | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela StavkaRačuna | | Prosto vrednosno ograničenje | | Složeno vrednosno ograničenje | | Strukturno ograničenje |
| Atributi | Ime | Tip atributa | Vrednost atributa | Međuzavisnost atributa jedne tabele | Međuzavisnost atributa više tabela | INSERT  RESTRICT  Račun,  Proizvod  UPDATE  RESTRICT  Račun,  Proizvod  DELETE / |
|  | BrojRačuna | Long | not null |  |  |
| RedniBroj | Int | >0 |  |  |
| Količina | Int | >0 (default:0) |  |  |
|  | Cena | Double | >0 (default:0) |  |  |
|  | Vrednost | Double | (default:0) |  |  |
|  | ŠifraProizvoda | Int | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela Proizvod | | Prosto vrednosno ograničenje | | Složeno vrednosno ograničenje | | Strukturno ograničenje |
| Atributi | Ime | Tip atributa | Vrednost atributa | Međuzavisnost atributa jedne tabele | Međuzavisnost atributa više tabela | INSERT  RESTRICT  Administrator  UPDATE  RESTRICT  Administrator,  CASCADES  StavkaRačuna  DELETE  RESTRICT |
|  | ŠifraProizvoda | Long | not null |  |  |
| NazivProizvoda | String |  |  |  |
| Sastojci | String |  |  |  |
|  | Priprema | String | not null |  |  |
|  | IDAdministratora | Int | not null |  |  | StavkaRačuna |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela Administrator | | Prosto vrednosno ograničenje | | Složeno vrednosno ograničenje | | Strukturno ograničenje |
| Atributi | Ime | Tip atributa | Vrednost atributa | Međuzavisnost atributa jedne tabele | Međuzavisnost atributa više tabela | INSERT /  UPDATE CASCADES  Proizvod  DELETE RESTRICT  Proizvod |
|  | IDAdministratora | Int | not null |  |  |
|  | JMBGAdmin | Long | not null |  |  |
| ImeAdmin | String |  |  |  |
| PrezimeAdmin | String |  |  |  |
|  | UsernameAdmin | String |  |  |  |
|  | PasswordAdmin | String |  |  |  |

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



# 3.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

Tronivojska arhitektura

## 3.1Projektovanje 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

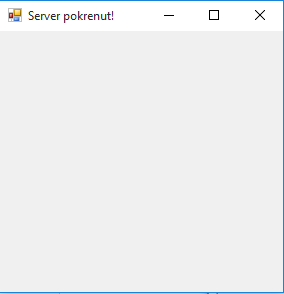
Kontroler KI

Struktura korisničkog interfejsa

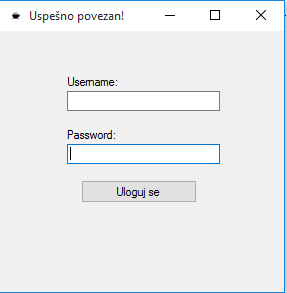
### 3.1.1 Projektovanje ekranskih formi

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

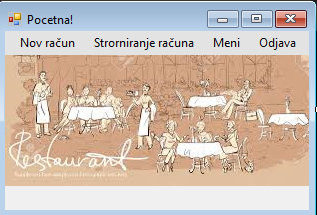
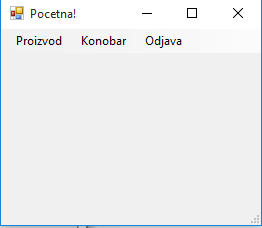
Glavna ekranska forma serverskog dela aplikacije treba da izgleda ovako:



Glavna ekranska forma klijentskog dela aplikacije treba da izgleda ovako:



Gde se u zavisnosti od toga da li smo se ulogovali kao administrator ili konobar imamo dve različite glavne forme.



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

#### 3.1.1.1 SK1: Slučаj korišćenjа – Unos novog proizvoda

**Nаziv SK**

Unos novog proizvoda

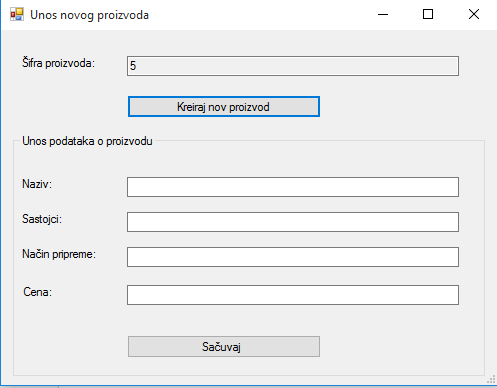
**Aktori SK**

*Administrator*

**Učesnici SK**

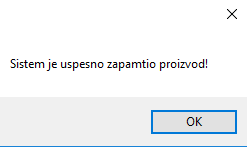
*Administrator i sistem (progrаm)*

**Preduslov**: Sistem je uključen i sistem prikаzuje formu zа rаd sа proizvodima.



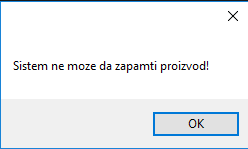
**Osnovni scenаrio SK**

1. **Administrator** unosi podаtke o novom *proizvodu*. (APUSO)
2. **Administrator** kontroliše dа li je korektno uneo podаtke o novom *proizvodu*. (ANSO)
3. **Administrator** pozivа *sistem* dа zаpаmti podаtke o *proizvodu*. (APSO)
4. **Sistem** pаmti podаtke o *proizvodu*. (SO)
5. **Sistem** prikаzuje *Administratoru* zаpаmćen *proizvod* i poruku: ʺSistem je zаpаmtio proizvodʺ. (IA)



Alternаtivnа scenаrijа

5.1 Ukoliko **sistem** ne može dа zаpаmti podаtke o novom *proizvodu* on prikаzuje Administratoru poruku “**Sistem** ne može dа zаpаmti nov proizvod”. (IA)



#### 3.1.1.2 SK2: Slučаj korišćenjа – Izmenа proizvoda

**Nаziv SK**

Izmenа proizvoda

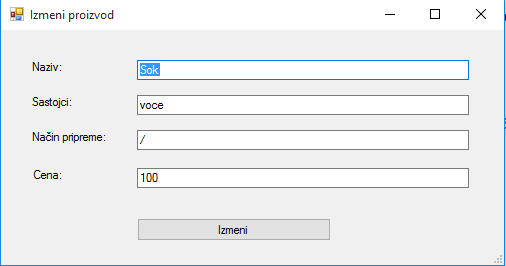
**Aktori SK**

*Administrator*

**Učesnici SK**

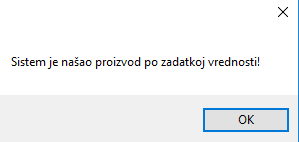
*Administrator i sistem (progrаm)*

**Preduslov**: Sistem je uključen i sistem prikаzuje formu zа rаd sа proizvodima.

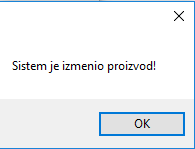


**Osnovni scenаrio SK**

1. **Administrator** unosi vrednost po kojoj pretražuje *proizvod* koje želi dа izmeni. (APUSO)
2. **Administrator** pozivа *sistem* dа pronаđe *proizvod* po zadatoj vrednosti.(APSO)
3. **Sistem** traži *proizvod* po zadatoj vrednosti. (SO)
4. **Sistem** prikаzuje *Administratoru* pronаđen *proizvod* i poruku: ʺ Sistem je nаšаo proizvod po zаdаtoj vrednostiʺ .(IA)
5. **Administrator** bira *proizvod*. (APUSO)
6. **Administrator** poziva *sistem* da vrati podatke o *proizvodu*. (APSO)
7. **Sistem** traži podatke o *proizvodu*. (SO)
8. **Sistem** prikazuje *Administratoru* podatke o *proizvodu* i poruku: “Sistem je pronašao podatke o izabranom *proizvodu*”. (IA)

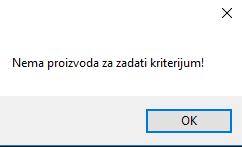


1. **Administrator** unosi podаtke o *proizvodu*. (APUSO)
2. **Administrator** kontroliše dа li je korektno uneo podаtke o novom *proizvodu*. (ANSO)
3. **Administrator** pozivа *sistem* dа zаpаmti podatke o *proizvodu*.(APSO)
4. **Sistem** pаmti podatke o *proizvodu*. (SO)
5. **Sistem** prikаzuje *Administratoru* poruku: ʺSistem je izmenio proizvodʺ. (IA)



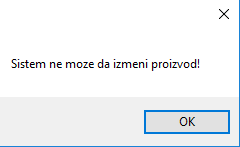
Alternаtivnа scenаrijа

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



8.1 Ukoliko **sistem** ne može da prikaže podatke o *proizvodu*, obaveštava Administratora o tome: “**Sistem** ne može da prikaže podatke o izabranom *proizvodu”.* Prekida se izvršenje scenarija. (IA)

13.1 Ukoliko **sistem** ne može dа zаpаmti izmenjene podаtke o *proizvodu* on prikаzuje Administratoru poruku “**Sistem** ne može dа izmeni *proizvod*”. (IA)



#### 3.1.1.3 SK3: Slučаj korišćenjа – Pretrаživаnje proizvoda

**Nаziv SK**

Pretrаživаnje proizvoda

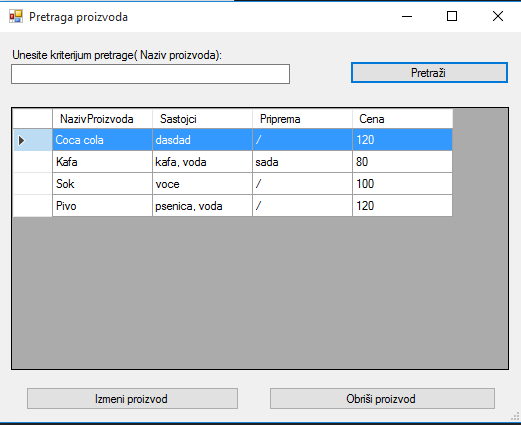
**Aktori SK**

*Konobar*

**Učesnici SK**

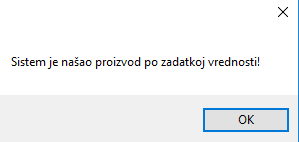
*Konobar i sistem (progrаm*)

**Preduslov**: Sistem je uključen, konobar je ulogovan sa svojim nalogom i sistem prikаzuje formu zа rаd sа proizvodima.



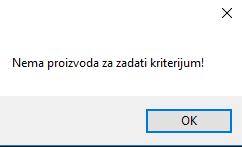
**Osnovni scenаrio SK**

1. **Konobar** unosi vrednost po kojoj pretrаžuje *proizvode*. (APUSO)
2. **Konobar** pozivа *sistem* dа nаđe *proizvod* po zаdаtoj vrednosti. (APSO)
3. **Sistem** trаži *proizvod* po zаdаtoj vrednosti. (SO)
4. **Sistem** prikаzuje *Konobar* podаtke o *proizvodu* i poruku: ʺSistem je nаšаo proizvod po zаdаtoj vrednostiʺ. (IA)



Alternаtivnа scenаrijа

* 1. Ukoliko **sistem** ne može dа nаđe *proizvod* on prikаzuje Konobaru poruku: “**Sistem** ne može dа nаđe *proizvod* po zаdаtoj vrednosti”. (IA)



#### 3.1.1.4 SK4: Slučаj korišćenjа – Brisanje proizvoda

**Nаziv** **SK**

Brisanje proizvoda

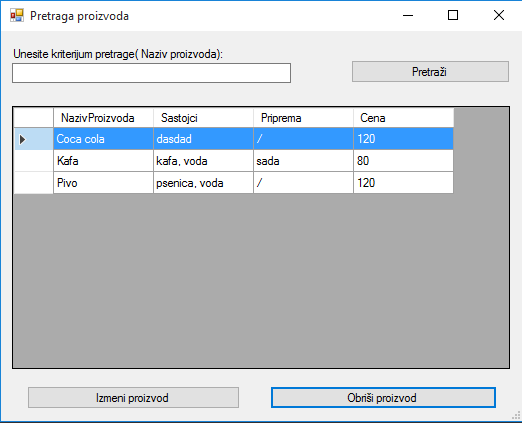
**Aktori SK**

*Administrator*

**Učesnici SK**

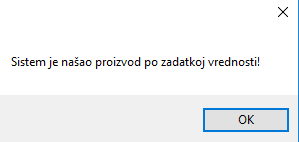
*Administrator i sistem (progrаm)*

**Preduslov**: Sistem je uključen i sistem prikаzuje formu zа rаd sа proizvodom.

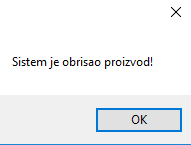


**Osnovni scenаrio SK**

1. **Administrator** unosi vrednost po kojoj pretrаžuje *proizvod*. (APUSO)
2. **Administrator** pozivа *sistem* dа nаđe *proizvod* po zаdаtoj vrednosti. (APSO)
3. **Sistem** trаži *proizvod* po zаdаtoj vrednosti. (SO)
4. **Sistem** prikаzuje *Administratoru* pronađen *proizvod* i poruku: ʺSistem je nаšаo proizvod po zаdаtoj vrednostiʺ. (IA)

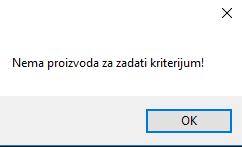


1. **Administrator** bira *proizvod*. (APUSO)
2. **Administrator** poziva *sistem* da vrati podatke o *proizvodu*. (APSO)
3. **Sistem** traži podatke o *proizvodu*. (SO)
4. **Sistem** prikazuje *Administratoru* podatke o *proizvodu* i poruku: “Sistem je pronašao podatke o izabranom *proizvodu*”. (IA)
5. **Administrator** poziva *sistem* da obriše izabran *proizvod*. (APSO)
6. **Sistem** briše izabran *proizvod*. (SO)
7. **Sistem** prikazuje *Administratoru* poruku: “Sistem je obrisao proizvod”. (IA)



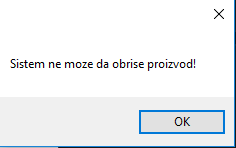
Alternаtivnа scenаrijа

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



8.1 Ukoliko **sistem** ne može da prikaže podatke o *proizvodu*, obaveštava Administratora o tome: “**Sistem** ne može da prikaže podatke o izabranom *proizvodu”.* Prekida se izvršenje scenaria. (IA)

11.1. Ukoliko **sistem** ne može da obriše *proizvod* on prikazuje Administratoru poruku: “**Sistem** ne može da obriše *proizvod*”. (IA)



#### 3.1.1.5 SK5: Slučаj korišćenjа – Kreirаnje novog rаčunа (Složeni slučaj korišćenja)

**Nаziv SK**

Kreirаnje novog rаčunа

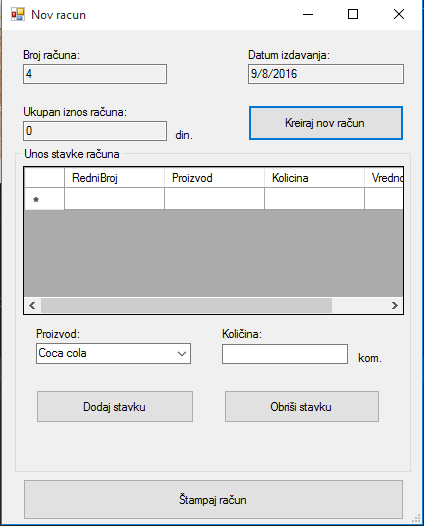
**Aktori SK**

*Konobar*

**Učesnici SK**

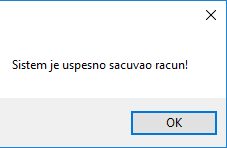
*Konobar i sistem (progrаm)*

**Preduslov**: Sistem je uključen, konobar je ulogovan sa svojim nalogom i sistem prikаzuje formu zа rаd sа rаčunom. Učitana je lista proizvoda.



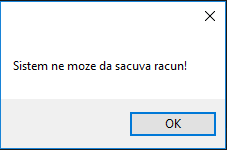
**Osnovni scenаrio SK**

1. **Konobar** unosi podаtke o novom *rаčunu*. (APUSO)
2. **Konobar** kontroliše dа li je korektno uneo podаtke o novom *rаčunu*. (ANSO)
3. **Konobar** pozivа sistem dа zаpаmti podаtke o *rаčunu*. (APSO)
4. **Sistem** pаmti podаtke o *rаčunu*. (SO)
5. **Sistem** prikаzuje *Konobaru* zаpаmćen novi *rаčun* i poruku: ʺSistem je zаpаmtio rаčunʺ. (IA)



Alternаtivnа scenаrijа

5.1 Ukoliko sistem ne može dа zаpаmti podаtke o novom rаčunu on prikаzuje Konobaru poruku “Sistem ne može dа zаpаmti rаčun”. (IA)



#### 3.1.1.6 SK6: Slučаj korišćenjа – Pretrаživаnje rаčunа

**Nаziv SK**

Pretrаživаnje rаčunа

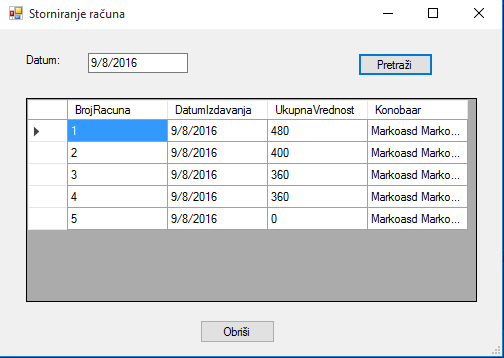
**Aktori SK**

*Konobar*

**Učesnici SK**

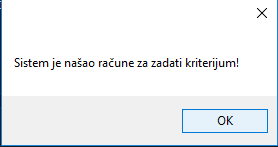
*Konobar i sistem (progrаm)*

**Preduslov**: Sistem je uključen, konobar je ulogovan sa svojim nalogom i sistem prikаzuje formu zа rаd sа rаčunom.



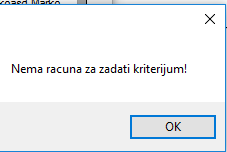
**Osnovni scenаrio SK**

1. **Konobar** unosi vrednost po kojoj pretrаžuje *rаčun*. (APUSO)
2. **Konobar** pozivа *sistem* dа nаđe *rаčun* po zаdаtoj vrednosti. (APSO)
3. **Sistem** trаži *rаčun* po zаdаtoj vrednosti. (SO)
4. **Sistem** prikаzuje *Konobaru* podаtke o *rаčunu* i poruku: ʺSistem je nаšаo rаčun po zаdаtoj vrednostiʺ. (IA)



Alternаtivnа scenаrijа

4.1 Ukoliko **sistem** ne može dа nаđe *rаčun* on prikаzuje Konobaru poruku: “**Sistem** ne može dа nаđe *rаčun* po zаdаtoj vrednosti”. (IA)



#### 3.1.1.7 SK7: Slučаj korišćenjа – Stornirаnje rаčunа

**Nаziv** **SK**

Stornirаnje rаčunа

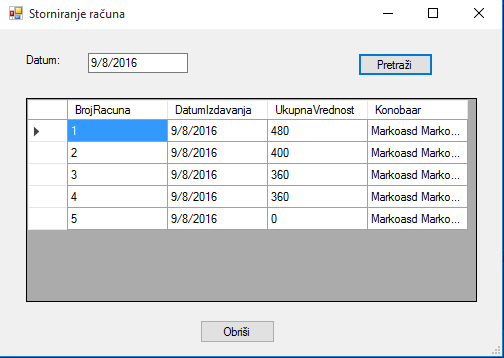
**Aktori** SK

*Konobar*

**Učesnici** SK

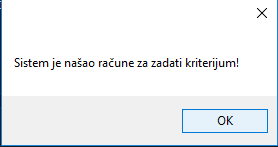
*Konobar* *i* *sistem* (*progrаm*)

**Preduslov**: Sistem je uključen i sistem prikаzuje formu zа rаd sа rаčunom.

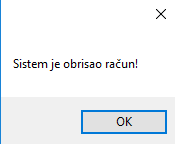


**Osnovni scenаrio SK**

1. **Konobar** unosi vrednost po kojoj pretrаžuje *rаčun*. (APUSO)
2. **Konobar** pozivа *sistem* dа nаđe *rаčun* po zаdаtoj vrednosti. (APSO)
3. **Sistem** trаži *rаčune* po zаdаtoj vrednosti. (SO)
4. **Sistem** prikаzuje *Konobaru* pronаđene *rаčune* i poruku: ʺSistem je nаšаo rаčune po zаdаtoj vrednostiʺ. (IA)

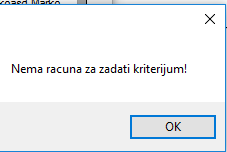


1. **Konobar** birа *rаčun* koji želi dа stornirа.(APUSO)
2. **Konobar** pozivа *sistem* dа stornirа odаbrаni *rаčun*. (APSO)
3. **Sistem** stornirа odаbrаni *rаčun*. (SO)
4. **Sistem** prikаzuje *Konobaru* stornirаn *rаčun* i poruku: “Sistem je stornirаo rаčun”. (IA)

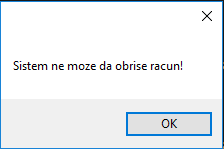


Alternаtivnа scenаrijа

4.1 Ukoliko **sistem** ne može dа nаđe *rаčun* on prikаzuje Konobaru poruku: “**Sistem** ne može dа nаđe *rаčun* po zаdаtoj vrednosti”. Prekidа se izvršenje scenаriа. (IA)



8.1 Ukoliko **sistem** ne može dа stornirа *rаčun* on prikаzuje Konobaru poruku: “**Sistem** ne može dа stornirа *rаčun*”.



#### 3.1.1.8 SK8: Slučаj korišćenjа – Logovanje konobara

**Nаziv SK**

Logovanje konobara

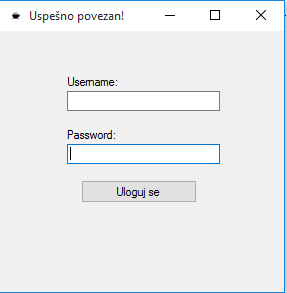
**Aktori SK**

Konobar

**Učesnici SK**

Konobar i sistem (progrаm)

**Preduslov**: Sistem je uključen i sistem prikаzuje formu zа logovanje.



**Osnovni scenаrio SK**

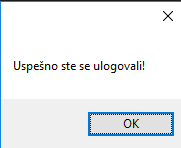
1. **Konobar** unosi svoje *korisničko ime* i *šifru*. (APUSO)

2. **Konobar** proverava da li je tačno uneo *korisničko ime* i *šifru*. (ANSO)

3. **Konobar** poziva *sistem* da ga uloguje. (APSO)

4. **Sistem** izvršava *validaciju*. (SO)

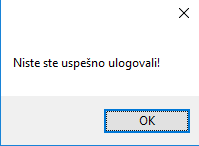
5. **Sistem** prikazuje *Konobaru* poruku da je uspešno ulogovan. (IA)



Alternаtivna scenаrija

5.1. Ukoliko **Konobar** sa takvim podacima ne postoji **sistem** će obavestiti

*Konobara* o grešci. (IA)



#### 3.1.1.9 SK9: Slučaj korišćenja - Odjava konobara

**Nаziv** **SK**

Odjava Konobara

**Aktori SK**

*Konobar*

**Učesnici SK**

*Konobar i sistem (progrаm)*

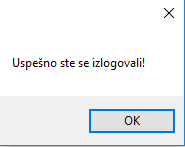
**Preduslov**: Sistem je uključen i Konobar je ulogovan.

**Osnovni scenаrio SK**

1. **Konobar** poziva *sistem* da ga izloguje (APSO)

2. **Sistem** izvršava *odjavu* (SO)

3. **Sistem** prikazuje *Konobaru* početnu *formu* za logovanje i poruku: “Usprešno ste se izlogovali” (IA)



Alternаtivna scenаrija

### 3.1.2. 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.

## 

## 3.2. 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.

### 3.2.1. Komunikacija sa klijentima

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

Slanje i primanje podataka od klijenta se obavlja razmenom objekta klase TransferObjekat i ostvaruje se preko soketa.

Klijent šalje zahtev za izvršenje neke od sistemskih operacija do odgovarajuće niti koja je povezana sa tim klijentom. Ta nit prihvata zahtev 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.

### 

### 3.2.2. 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).

### 

### 3.3.3. 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.

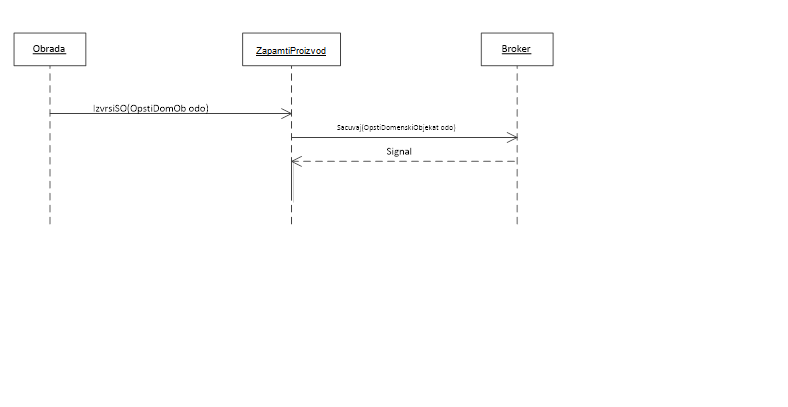
**Ugovor UG1: ZapamtiProizvod(Proizvod)**

**Operacija: ZapamtiProizvod** (Proizvod): signal;

**Veza sa SK:** SK1, SK2, SK4

**Preduslovi:** Moraju biti zadovoljena prosta vrednosna i strukturna ograničenja nad objektom Proizvod.

**Postuslovi:** Proizvod je zapamćen.



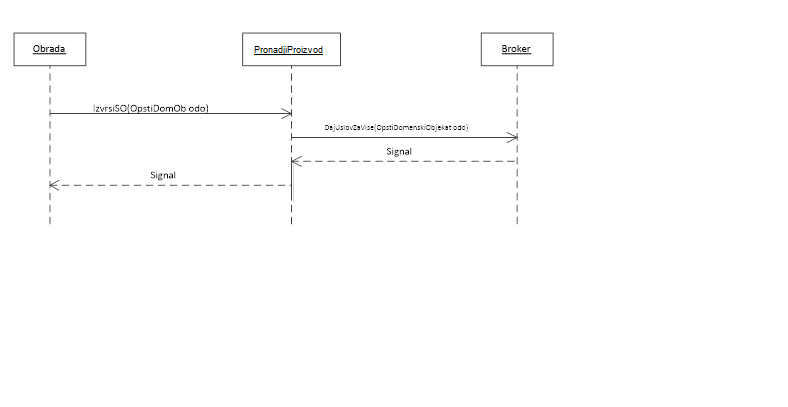
**Ugovor UG2: NađiProizvod (Proizvod, List<Proizvod>)**

**Operacija: NađiProizvod** (Proizvod): signal;

**Veza sa SK:** SK2, SK3, SK4

**Preduslovi:**

**Postuslovi:**



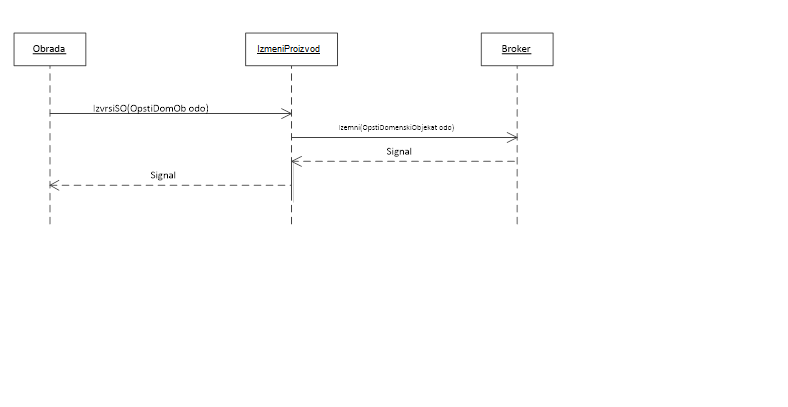
**Ugovor UG3: IzmeniProizvod (Proizvod)**

**Operacija: NađiProizvod** (Proizvod): signal;

**Veza sa SK:** SK2, SK4

**Preduslovi:**

**Postuslovi:**



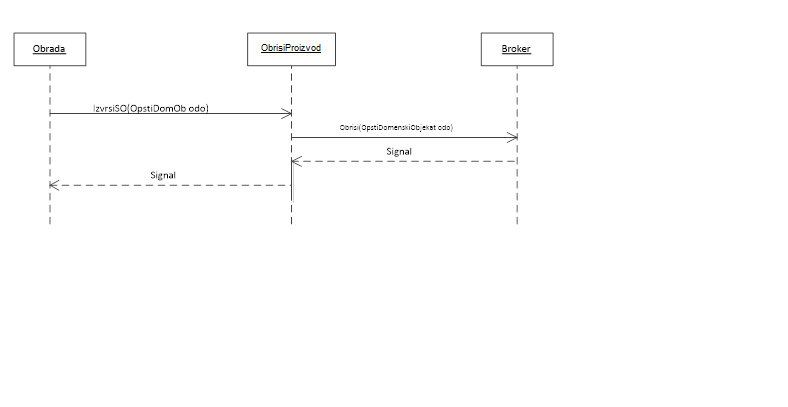
**Ugovor UG4: ObrišiProizvod (Proizvod)**

**Operacija: ObrišiProizvod** (Proizvod): signal;

**Veza sa SK:** SK4

**Preduslovi:** Moraju biti zadovoljena strukturna ograničenja nad objektom Proizvod.

**Postuslovi:** Proizvod je obrisan.



**Ugovor UG5: ZapamtiRačun(Račun)**

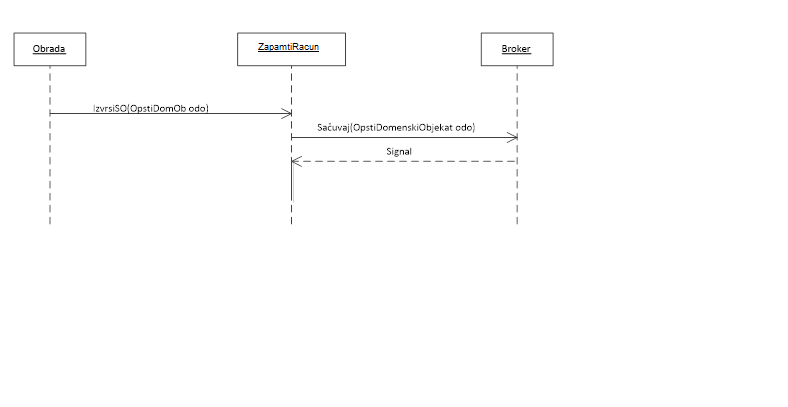
**Operacija: ZapamtiRačun**(Račun): signal;

**Veza sa SK:** SK5

**Preduslovi:** Ako je račun storniran ne može se izvršiti sistemska operacija. Moraju biti zadovoljena prosta vrednosna i strukturna ograničenja nad objektom Račun.

**Postuslovi:**

* Izračunata je vrednost svake od stavki računa
* Izračunata je ukupna vrednost računa



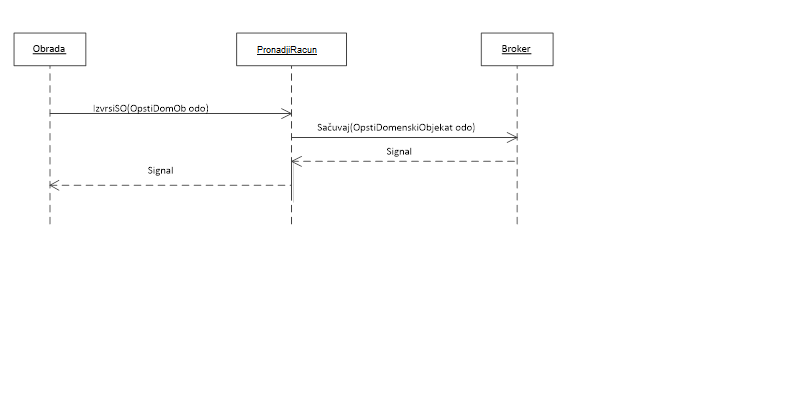
**Ugovor UG5: NađiRačun(Račun, List<Račun>)**

**Operacija: NađiRačun**(Račun): signal;

**Veza sa SK:** SK6. SK7

**Preduslovi:**

**Postuslovi:**



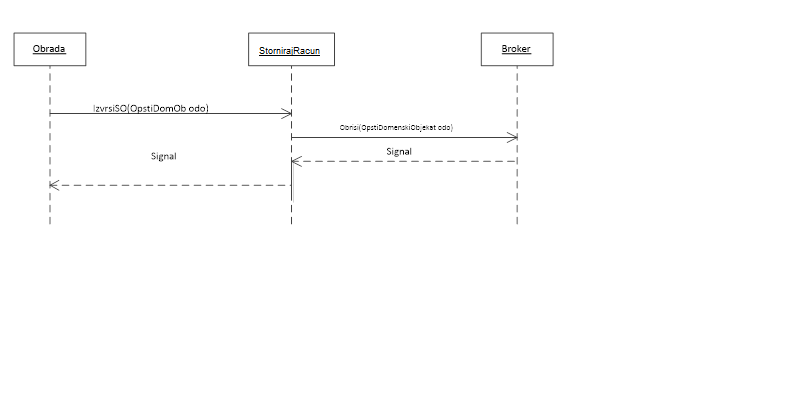
**Ugovor UG6: StornirajRačun(Račun)**

**Operacija: StornirajRačun**(Račun): signal;

**Veza sa SK:** SK7

**Preduslovi:** Ako je račun storniran ne može se izvršiti sistemska operacija. Moraju biti zadovoljena strukturna ograničenja nad objektom Račun.

**Postuslovi:** Račun je storniran.



**Ugovor UG7: Logovanje(String, String)**

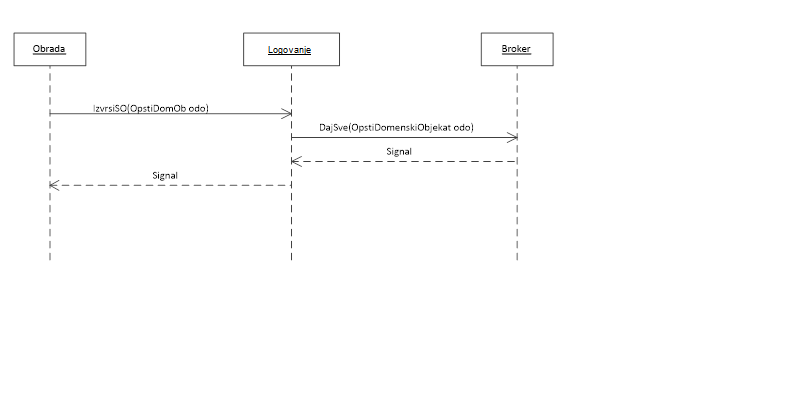
**Operacija: Logovanje** (String, String): signal;

**Veza sa SK:** SK8

**Preduslovi:** Sistem prikazuje početnu stranu za logovanje.

**Postuslovi:** Konobar je uspešno ulogovan u system.

**Ugovor UG9: VratiListuProizvoda()**



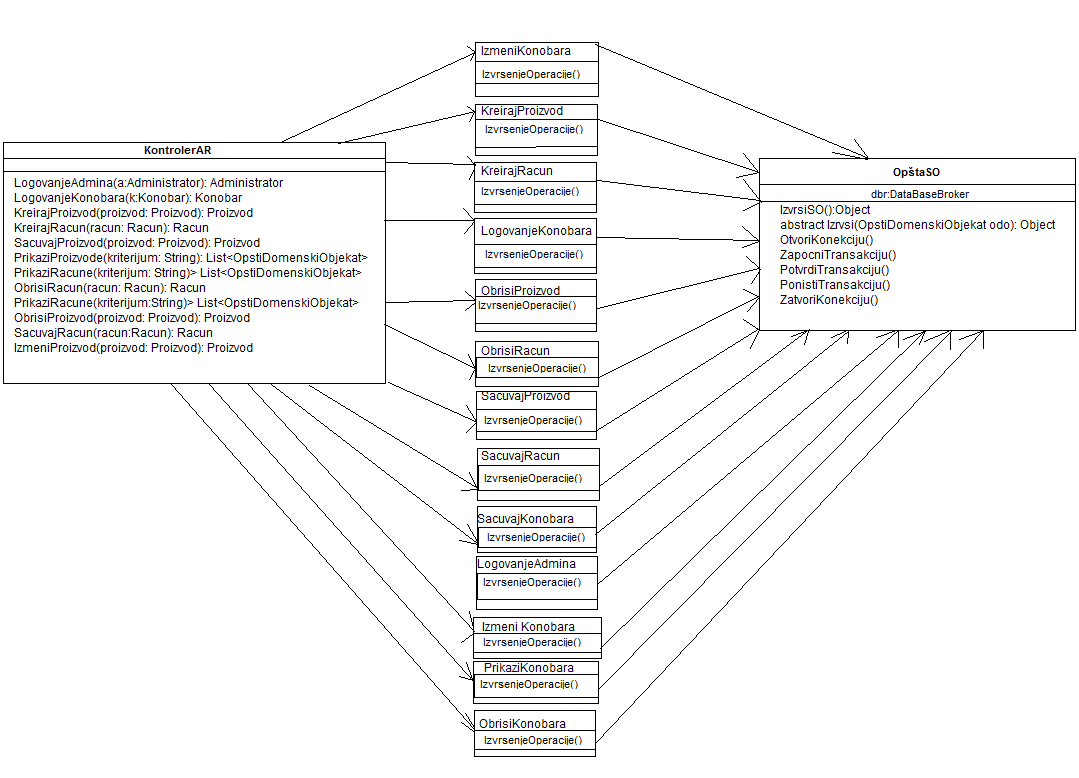
**Operacija**: **VratiListuProizvoda**(): signal;

**Veza sa SK:** SK5

**Preduslovi: /**

**Postuslovi:** Lista proizvoda je vraćena.



Klase koje su odgovorne za izvršenje sistemskih operacija nasleđuju klasu OpstaSistemskaOperacija kako bi mogle da se povežu sa bazom i kako bi se njihovo izvršenje pratilo kao transakcija. OpstaSistemskaOperacija predstavlja apstraktnu klasu čija glavna metoda (opsteIzvrsenjeSO()) 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 Dijagram klasa koji pokazuje vezu između kontrolera aplikacione logike i klasa odgovornih za izvršenje sistemskih operacija koje nasleđuju klasu OpstaSistemskaOperacija

**Projektovanje strukture softverskog sistema**

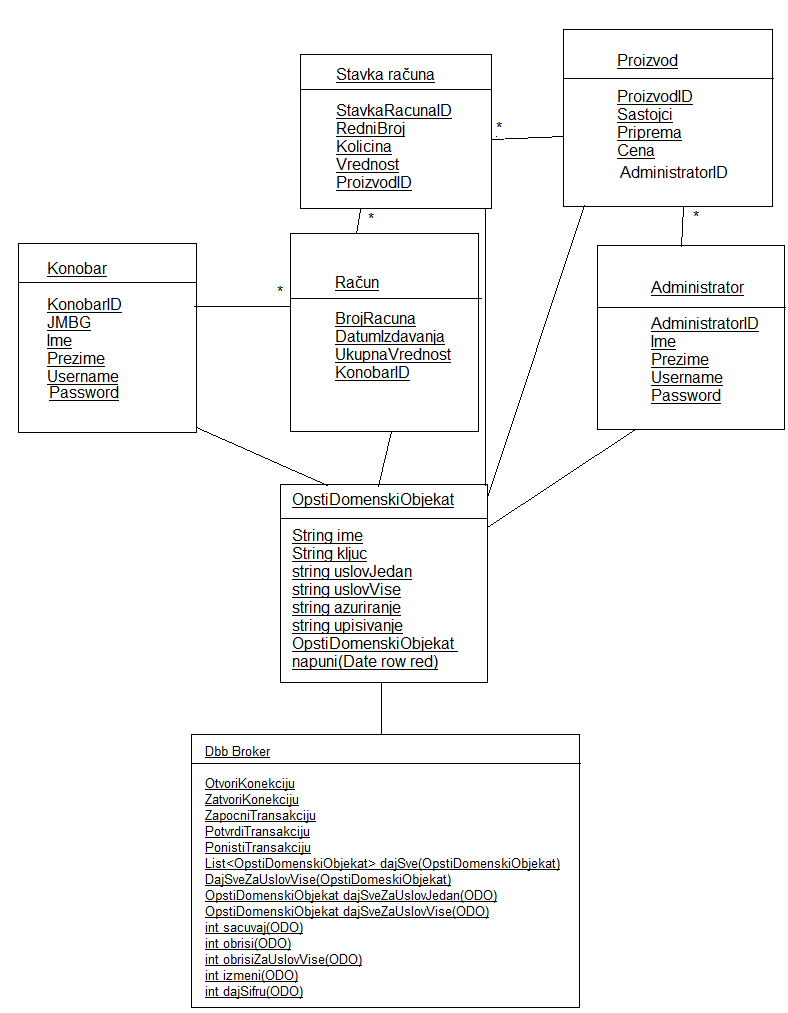
Na osnovu konceptualnih klasa prave se softverske klase strukture. Svaka klasa ima privatna polja atributra, gettere i settere za te atribute, besparametaski konstruktor kao i parametarski konstruktor.

### 3.2.4. Broker baze podataka

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

1. public void otvoriKonekciju()
2. public void zatvoriKonekciju()
3. public void zapocniTransakciju()
4. public void potvrdiTransakciju()
5. public void ponistiTransakciju()
6. public List<OpstiDomenskiObjekat>dajSve(OpstiDomenskiObjekat odo)
7. public List<OpstiDomenskiObjekat>dajSveZaUslov(OpstiDomenskiObjekat odo)
8. public List<OpstiDomenskiObjekat>dajSveZaUslovVise(OpstiDomenskiObjekat odo)
9. public OpstiDomenskiObjekat dajZaUslov(OpstiDomenskiObjekat odo)
10. public OpstiDomenskiObjekat dajZaUslovVise(OpstiDomenskiObjekat odo)
11. int ubaci (OpstiDomenskiObjekat odo)
12. int obrisi (OpstiDomenskiObjekat odo)
13. int izmeni (OpstiDomenskiObjekat odo)
14. int dajSifru (OpstiDomenskiObjekat odo)

Metode DatabaseBrokera 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. Takođe postoje i pojedinačne metode koje nisu mogle biti projektovane kao generičke.



Slika 3 DatabaseBroker se povezuje sa interfejsom OpstiDomenskiObjekat

## 3.3. Projektovanje skladišta podataka

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

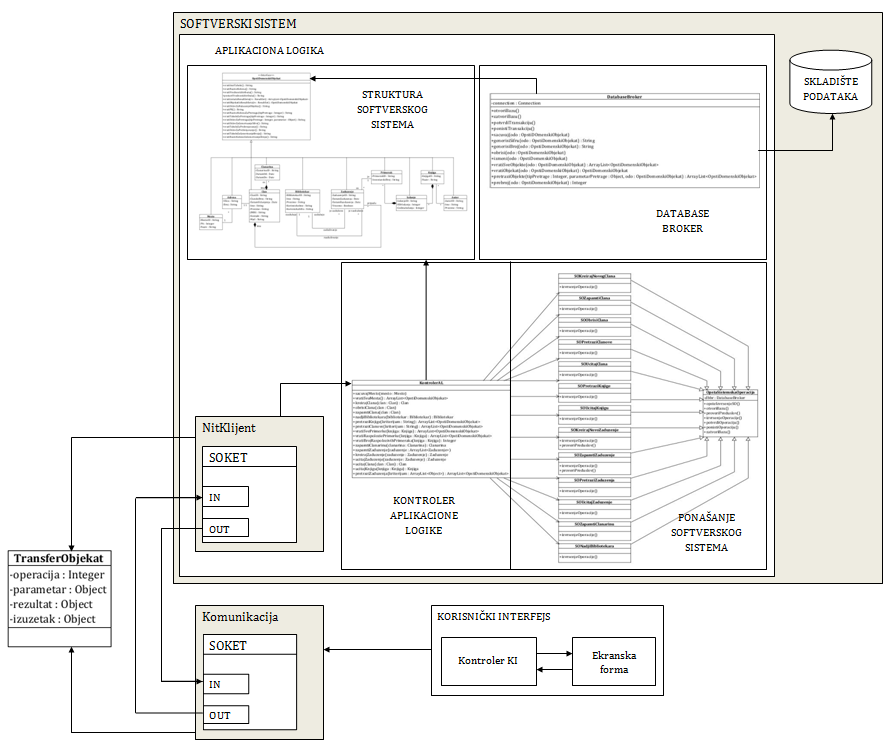
|  |  |
| --- | --- |
| **Table: Administrator** | |
| AdministratorID | Number |
| Ime | Text |
| Prezime | Text |
| Username | Text |
| Password | Text |

|  |  |
| --- | --- |
| **Table: Konobar** | |
| KonobarID | Number |
| JMBG | Number |
| Ime | Text |
| Prezime | Text |
| Username | Text |
| Password | Text |

|  |  |
| --- | --- |
| **Table: Proizvod** | |
| ŠifraProizvoda | Number |
| NazivProizvoda | Text |
| Sastojci | Text |
| Priprema | Text |
| Cena | Double |

|  |  |
| --- | --- |
| **Table: Racun** | |
| BrojRacuna | Number |
| DatumIzdavanja | Date/Time |
| UkupnaVrednost | Double |
| KonobarID | Number |

|  |  |
| --- | --- |
| **Table: StavkaRacuna** | |
| StavkaRacunaID | Number |
| RedniBroj | Number |
| Kolicina | Number |
| Vrednost | Double |
| SifraProizvoda | Number |



Slika 4 Arhitektura softverskog sistema

# 4. Implementacija

Softverski sistem, rezultat ovog rada, razvijen je u programskom jeziku C# i projektovan je kao klijent-server. Kao razvojno okruženje koriščen je VisualStudio 2012, a kao sistem za upravljanje bazom podataka korišćen je Microsoft Office Access.

Čitav sistem je realizovan u sedam projekta: Domen, KorisnickiInterfejs, KontorolerKI, Komunikacija, Server, SistemskeOperacije i Sesija. Na osnovu arhitekture softverskog sistema dobijene su sledeće softverske klase:

- **Domen**

* 
* 
* 
* 
* 
* 
* 

- **KorinickiInterfejs**

* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 

- **KontrolerKI**

* 
* 

-Komunikacija

* 
* 

-**Server**

* 
* 
* 
* 
* 
* 
* 

**-SistemskeOperacije**

* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 
* 

**-Sesija**

* 
* 

# 5.Testiranje

Svaki od implementiranih slučajeva korišćenja je testiran. Prilikom testiranja svakog

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

se utvrdilo kakav će rezultat izvršenja biti.

Na osnovu izvršenih testiranja otklonjeni su uočeni nedostaci.