Univerzitet u Beogradu

Fakultet organizacionih nauka

Laboratorija za softversko inženjerstvo

Seminarski rad

Predmet: Projektovanje softvera

Tema: Softverski sistem za evidenciju faktura o prodaji

|  |  |
| --- | --- |
| Profesor: | Student: |
| prof dr Siniša Vlajić | Jelena Đorđević 6/09/I |

Beograd, 2013.

**Sadržaj**

[Prikupljanje zahteva 4](#_Toc361038997)

[Verbalni opis 4](#_Toc361038998)

[Slučajevi korišćenja 4](#_Toc361038999)

[Slučaj korišćenja 1: Unos nove fakture 5](#_Toc361039000)

[Slučaj korišćenja 2: Izmena fakture 6](#_Toc361039001)

[Slučaj korišćenja 3: Storniranje fakture 7](#_Toc361039002)

[Slučaj korišćenja 4: Pretraga faktura 8](#_Toc361039003)

[Slučaj korišćenja 5: Unos novog proizvoda 9](#_Toc361039004)

[Slučaj korišćenja 6: Izmena proizvoda 10](#_Toc361039005)

[Slučaj korišćenja 7: Unos poslovnog partnera 11](#_Toc361039006)

[Slučaj korišćenja 8: Izmena poslovnog partnera 12](#_Toc361039007)

[Slučaj korišćenja 9: Prijavljivanje korisnika 13](#_Toc361039008)

[Analiza 14](#_Toc361039009)

[Dijagrami sekvenci 14](#_Toc361039010)

[DS1: Dijagram sekvenci slučaja korišćenja –Unos fakture 14](#_Toc361039011)

[DS2: Dijagram sekvenci slučaja korišćenja – Izmena fakture 16](#_Toc361039012)

[DS3: Dijagram sekvenci slučaja korišćenja – Storniranje fakture 18](#_Toc361039013)

[DS4: Dijagram sekvenci slučaja korišćenja – Pretraga faktura 20](#_Toc361039014)

[DS5: Dijagram sekvenci slučaja korišćenja – Unos novog proizvoda 21](#_Toc361039015)

[DS6: Dijagram sekvenci slučaja korišćenja – Izmena proizvoda 23](#_Toc361039016)

[DS7: Dijagram sekvenci slučaja korišćenja – Unos poslovnog partnera 25](#_Toc361039017)

[DS8: Dijagram sekvenci slučaja korišćenja – Izmena poslovnog partnera 26](#_Toc361039018)

[DS9: Dijagram sekvenci slučaja korišćenja – Prijavljivanje korisnika 28](#_Toc361039019)

[Ugovori o sistemskim operacijama 29](#_Toc361039020)

[Ugovor UG1: KreirajNovu 29](#_Toc361039021)

[Ugovor UG2: Zapamti 29](#_Toc361039022)

[Ugovor UG3: Obradi 29](#_Toc361039023)

[Ugovor UG4: PretraziFakture 29](#_Toc361039024)

[Ugovor UG5: NadjiFakturu 29](#_Toc361039025)

[Ugovor UG6: Storniraj 30](#_Toc361039026)

[Ugovor UG7: KreirajNoviProizvod 30](#_Toc361039027)

[Ugovor UG8: ZapamtiProizvod 30](#_Toc361039028)

[Ugovor UG9: PretraziProizvode 30](#_Toc361039029)

[Ugovor UG10: NadjiProizvod 30](#_Toc361039030)

[Ugovor UG11: KreirajNovogPoslovnogPartnera 30](#_Toc361039031)

[Ugovor UG12: ZapamtiPoslovnogPartnera 31](#_Toc361039032)

[Ugovor UG13: PretraziPoslovnePartnere 31](#_Toc361039033)

[Ugovor UG14: NadjiPoslovnogPartnera 31](#_Toc361039034)

[Ugovor UG15: NadjiKorisnika 31](#_Toc361039035)

[Konceptualni model 32](#_Toc361039036)

[Relacioni model 32](#_Toc361039037)

[Projektovanje 36](#_Toc361039038)

[Arhitektura softverskog sistema 36](#_Toc361039039)

[Projektovanje korisničkog interfejsa 37](#_Toc361039040)

[Projektovanje ekranskih formi 37](#_Toc361039041)

[Projektovanje kontrolera korisničkog interfejsa 90](#_Toc361039042)

[Projektovanje aplikacione logike 94](#_Toc361039043)

[Kontroler aplikacione logike 94](#_Toc361039044)

[Poslovna logika 96](#_Toc361039045)

[Broker baze podataka 102](#_Toc361039046)

[Projektovanje skladišta podataka 107](#_Toc361039047)

# Prikupljanje zahteva

## Verbalni opis

Potrebno je napraviti aplikaciju koja će omogućiti vođenje evidencije o izlaznim fakturama preduzeća, unos nove fakture, izmenu postojeće, storniranje fakture i pretragu unetih faktura. Za svaku fakturu treba uneti poslovnog partnera, datum prodaje i proizvode koji su predmet poslovanja. Takođe treba voditi evidenciju o vrednosti PDV – a za svaku fakturu.

U aplikaciji treba voditi evidenciju o proizvodima koje preduzeće proizvodi i prodaje, njihovo dodavanje i izmenu. Svaki proizvod ima svoj jedinstveni identifikator, naziv, cenu i jedinicu mere.

Zatim treba voditi evidenciju o poslovnim partnerima sa kojima preduzeće posluje, njihovo dodavanje i izmenu. O poslovnim partnerima treba čuvati podatke: naziv, PIB, matični broj preduzeća, adresu i kontakt.

Da bi korisnik koristio aplikaciju mora da bude prijavljen na sistem.

## Slučajevi korišćenja

Identifikovani su sledeći slučajevi korišćenja:

1. Unos nove fakture
2. Izmena fakture
3. Pretraga faktura
4. Storniranje fakura
5. Unos novog proizvoda
6. Izmena proizvoda
7. Unos poslovnog partnera
8. Izmena poslovnog partnera
9. Prijavljivanje korisnika



Slika 1: Dijagram slučajeva korišćenja

### Slučaj korišćenja 1: Unos nove fakture

##### **Naziv SK**

Unos nove fakture

##### **Aktori SK**

Korisnik

##### **Učesnici SK**

Korisnik i sistem (program)

##### **Preduslov**: Sistem je uključen i korisnik je prijavljen pod svojom šifrom. Sistem prikazuje početnu formu.

##### **Osnovni scenario SK**

1. Korisnik **poziva** sistem da kreira novu fakturu. (APSO)
2. Sistem **kreira** novu fakturu. (SO)
3. Sistem **prikazuje** korisniku novu fakturu i poruku: “Sistem je kreirao novu fakturu”. (IA)
4. Korisnik **unosi** podatke u novu fakturu. (APUSO)
5. Korisnik **kontroliše** da li je korektno uneo podatke u novu fakturu. (ANSO)
6. Korisnik **poziva** sistem da zapamti podatke o fakturi. (APSO)
7. Sistem **pamti** podatke o fakturi. (SO)
8. Sistem **prikazuje** korisniku zapamćenu fakturu i poruku: “Sistem je sačuvao fakturu”. (IA)
9. Korisnik **poziva** sistem da obradi fakturu. (APSO)
10. Sistem **obrađuje** fakturu. (SO)
11. Sistem **prikazuje** korisniku obrađenu fakturu i poruku: “Sistem je obradio fakturu”. (IA)

##### **Alternativna scenarija**

3.1 Ukoliko sistem ne može da kreira fakturu on prikazuje korisniku poruku: “Sistem ne može da kreira novu fakturu”. Prekida se izvršenje scenaria. (IA)

8.1 Ukoliko sistem ne može da zapamti podatke o fakturi on prikazuje korisniku poruku “Sistem ne može da zapamti fakturu”. Prekida se izvršenje scenaria. (IA)

11.1 Ukoliko sistem ne može da obradi fakturu on prikazuje korisniku poruku “Sistem ne može da obradi fakturu”. (IA)

### Slučaj korišćenja 2: Izmena fakture

##### **Naziv SK**

Izmena fakture

##### **Aktori SK**

Korisnik

##### **Učesnici SK**

Korisnik i sistem (program)

##### **Preduslov**: Sistem je uključen i korisnik je prijavljen pod svojom šifrom. Sistem prikazuje formu za pretragu faktura.

##### **Osnovni scenario SK**

1. Korisnik **unosi** vrednost po kojoj pretražuje fakture. (APUSO)
2. Korisnik **poziva** sistem da nađe fakture po zadatoj vrednosti. (APSO)
3. Sistem **traži** fakturu po zadatoj vrednosti. (SO)
4. Sistem **prikazuje** korisniku pronađene fakture. (IA)
5. Korisnik **bira** fakturu. (APUSO)
6. Korisnik **poziva** sistem da nađe sve podatke o izabranoj fakturi. (APSO)
7. Sistem **traži** podatke o izabranoj fakturi.(SO)
8. Sistem **prikazuje** korisniku sve podatke o fakturi. (IA)
9. Korisnik **unosi (menja)** podatke o fakturi. (APUSO)
10. Korisnik **kontroliše** da li je korektno uneo podatke o fakturi. (ANSO)
11. Korisnik **poziva** sistem da zapamti podatke o fakturi. (APSO)
12. Sistem **pamti** podatke o fakturi. (SO)
13. Sistem **prikazuje** korisniku zapamćenu fakturu i poruku: “Sistem je zapamtio fakturu.” (IA)
14. Korisnik **poziva** sistem da obradi fakturu. (APSO)
15. Sistem **obrađuje** fakturu. (SO)
16. Sistem **prikazuje** prodavcu obrađenu fakturu i poruku: “Sistem je obradio fakturu.” (IA)

##### **Alternativna scenarija**

4.1 Ukoliko sistem ne može da nađe nijednu fakturu on prikazuje korisniku poruku: “Sistem ne može da nađe fakturu po zadatoj vrednosti”. (IA)

8.1 Ukoliko sistem ne može da nađe podatke o fakturi on prikazuje korisniku poruku: “Sistem ne može da nađe fakturu po zadatoj vrednosti”. (IA)

13.1 Ukoliko sistem ne može da zapamti podatke o izmenjenoj fakturi on prikazuje korisniku poruku “Sistem ne može da zapamti novu fakturu”. Prekida se izvršenje scenaria. (IA)

16.1 Ukoliko sistem ne može da obradi fakturu on prikazuje korisniku poruku: “Sistem ne može da obradi fakturu.” (IA)

### Slučaj korišćenja 3: Storniranje fakture

##### **Naziv SK**

Storniranje fakture

##### **Aktori SK**

Korisnik

##### **Učesnici SK**

Korisnik i sistem (program)

##### **Preduslov**: Sistem je uključen i korisnik je prijavljen pod svojom šifrom. Sistem prikazuje formu za pretragu faktura.

##### **Osnovni scenario SK**

1. Korisnik **unosi** vrednost po kojoj pretražuje fakture. (APUSO)
2. Korisnik **poziva** sistem da nađe fakture po zadatoj vrednosti. (APSO)
3. Sistem **traži** fakturu po zadatoj vrednosti. (SO)
4. Sistem **prikazuje** korisniku pronađene fakture. (IA)
5. Korisnik **bira** fakturu. (APUSO)
6. Korisnik **poziva** sistem da nađe sve podatke o izabranoj fakturi. (APSO)
7. Sistem **traži** podatke o izabranoj fakturi.(SO)
8. Sistem **prikazuje** korisniku sve podatke o fakturi. (IA)
9. Korisnik **poziva** sistem da stornira zadatu fakturu. (APSO)
10. Sistem **stornira** fakturu. (SO)
11. Sistem **prikazuje** korisniku storniranu fakturu i poruku: “Sistem je stornirao fakturu.” (IA)

##### **Alternativna scenarija**

4.1 Ukoliko sistem ne može da nađe nijednu fakturu on prikazuje korisniku poruku: “Sistem ne može da nađe fakturu po zadatoj vrednosti”. (IA)

8.1 Ukoliko sistem ne može da nađe podatke o fakturi on prikazuje korisniku poruku: “Sistem ne može da nađe fakturu po zadatoj vrednosti”. (IA)

11.1 Ukoliko sistem ne može da stornira fakturu on prikazuje korisniku poruku: “Sistem ne može da stornira fakturu”. (IA)

### Slučaj korišćenja 4: Pretraga faktura

##### **Naziv SK**

Pretraga faktura

##### **Aktori SK**

Korisnik

##### **Učesnici SK**

Korisnik i sistem (program)

##### **Preduslov**: Sistem je uključen i korisnik je prijavljen pod svojom šifrom. Sistem prikazuje formu za pretragu faktura.

##### **Osnovni scenario SK**

1. Korisnik **unosi** vrednost po kojoj pretražuje fakture. (APUSO)
2. Korisnik **poziva** sistem da nađe fakturu po zadatoj vrednosti. (APSO)
3. Sistem **traži** fakturu po zadatoj vrednosti. (SO)
4. Sistem **prikazuje** korisniku pronađene fakture. (IA)
5. Korisnik **bira** fakturu. (APUSO)
6. Korisnik **poziva** sistem da nađe sve podatke o izabranoj fakturi. (APSO)
7. Sistem **traži** podatke o izabranoj fakturi. (SO)
8. Sistem **prikazuje** korisniku sve podatke o fakturi. (IA)

##### **Alternativna scenarija**

4.1 Ukoliko sistem ne može da nađe nijednu fakturu on prikazuje korisniku poruku: “Sistem ne može da nađe fakturu po zadatoj vrednosti”. (IA)

8.1 Ukoliko sistem ne može da nađe podatke o fakturi on prikazuje korisniku poruku: “Sistem ne može da nađe fakturu po zadatoj vrednosti”. (IA)

### Slučaj korišćenja 5: Unos novog proizvoda

##### **Naziv SK**

Unos novog proizvoda

##### **Aktori SK**

Korisnik

##### **Učesnici SK**

Korisnik i sistem (program)

##### **Preduslov**: Sistem je uključen i korisnik je prijavljen pod svojom šifrom. Sistem prikazuje početnu formu.

##### **Osnovni scenario SK**

1. Korisnik **poziva** sistem da kreira novu proizvod. (APSO)
2. Sistem **kreira** novi proizvod. (SO)
3. Sistem **prikazuje** korisniku novi proizvod i poruku: “Sistem je kreirao novi proizvod”. (IA)
4. Korisnik **unosi** podatke o novom proizvodu. (APUSO)
5. Korisnik **kontroliše** da li je korektno uneo podatke o proizvodu. (ANSO)
6. Korisnik **poziva** sistem da zapamti podatke o proizvodu. (APSO)
7. Sistem **pamti** podatke o proizvodu. (SO)
8. Sistem **prikazuje** korisniku zapamćeni proizvod i poruku: “Sistem je sačuvao proizvod”. (IA)

##### **Alternativna scenarija**

3.1 Ukoliko sistem ne može da kreira proizvod on prikazuje korisniku poruku: “Sistem ne može da kreira novi proizvod”. Prekida se izvršenje scenaria. (IA)

8.1 Ukoliko sistem ne može da zapamti podatke o proizvodu on prikazuje korisniku poruku “Sistem ne može da zapamti proizvod”. Prekida se izvršenje scenaria. (IA)

### Slučaj korišćenja 6: Izmena proizvoda

##### **Naziv SK**

Izmena proizvoda

##### **Aktori SK**

Korisnik

##### **Učesnici SK**

Korisnik i sistem (program)

##### **Preduslov**: Sistem je uključen i korisnik je prijavljen pod svojom šifrom. Sistem prikazuje formu za pretragu proizvoda.

##### **Osnovni scenario SK**

1. Korisnik **unosi** vrednost po kojoj pretražuje proizvode. (APUSO)
2. Korisnik **poziva** sistem da nađe proizvode po zadatoj vrednosti. (APSO)
3. Sistem **traži** proizvod po zadatoj vrednosti. (SO)
4. Sistem **prikazuje** korisniku pronađene proizvode. (IA)
5. Korisnik **bira** proizvod. (APUSO)
6. Korisnik **poziva** sistem da nađe sve podatke o izabranom proizvodu. (APSO)
7. Sistem **traži** podatke o izabranom proizvodu. (SO)
8. Sistem **prikazuje** korisniku sve podatke o proizvodu. (IA)
9. Korisnik **unosi (menja)** podatke o proizvodu. (APUSO)
10. Korisnik **kontroliše** da li je korektno uneo podatke o proizvodu. (ANSO)
11. Korisnik **poziva** sistem da zapamti podatke o proizvodu. (APSO)
12. Sistem **pamti** podatke o proizvodu. (SO)
13. Sistem **prikazuje** korisniku zapamćen proizvod i poruku: “Sistem je zapamtio proizvod.” (IA)

##### **Alternativna scenarija**

4.1 Ukoliko sistem ne može da nađe nijedan proizvod on prikazuje korisniku poruku: “Sistem ne može da nađe proizvod po zadatoj vrednosti”. (IA)

8.1 Ukoliko sistem ne može da nađe podatke o proizvodu on prikazuje korisniku poruku: “Sistem ne može da nađe proizvod po zadatoj vrednosti”. (IA)

13.1 Ukoliko sistem ne može da zapamti podatke o izmenjenom proizvodu on prikazuje korisniku poruku “Sistem ne može da zapamti izmenjeni proizvod”. Prekida se izvršenje scenaria. (IA)

### Slučaj korišćenja 7: Unos poslovnog partnera

##### **Naziv SK**

Unos novog poslovnog partnera

##### **Aktori SK**

Korisnik

##### **Učesnici SK**

Korisnik i sistem (program)

##### **Preduslov**: Sistem je uključen i korisnik je prijavljen pod svojom šifrom. Sistem prikazuje početnu formu.

##### **Osnovni scenario SK**

1. Korisnik **poziva** sistem da kreira novog poslovnog partnera. (APSO)
2. Sistem **kreira** novog poslovnog partnera. (SO)
3. Sistem **prikazuje** korisniku novog poslovnog partnera i poruku: “Sistem je kreirao novog poslovnog partnera”. (IA)
4. Korisnik **unosi** podatke o novom poslovnom partneru. (APUSO)
5. Korisnik **kontroliše** da li je korektno uneo podatke o poslovnom partneru. (ANSO)
6. Korisnik **poziva** sistem da zapamti podatke o poslovnom partneru. (APSO)
7. Sistem **pamti** podatke o poslovnom partneru. (SO)
8. Sistem **prikazuje** korisniku zapamćenog poslovnog partnera i poruku: “Sistem je sačuvao poslovnog partnera”. (IA)

##### **Alternativna scenarija**

3.1 Ukoliko sistem ne može da kreira poslovnog partnera on prikazuje korisniku poruku: “Sistem ne može da kreira novog poslovnog partnera”. Prekida se izvršenje scenaria. (IA)

8.1 Ukoliko sistem ne može da zapamti podatke o poslovnom partneru on prikazuje korisniku poruku “Sistem ne može da zapamti poslovnog partnera”. Prekida se izvršenje scenaria. (IA)

### Slučaj korišćenja 8: Izmena poslovnog partnera

##### **Naziv SK**

Izmena poslovnog partnera

##### **Aktori SK**

Korisnik

##### **Učesnici SK**

Korisnik i sistem (program)

##### **Preduslov**: Sistem je uključen i korisnik je prijavljen pod svojom šifrom. Sistem prikazuje formu za pretragu poslovnih partnera.

##### **Osnovni scenario SK**

1. Korisnik **unosi** vrednost po kojoj pretražuje poslovne partnere. (APUSO)
2. Korisnik **poziva** sistem da nađe poslovne partnere po zadatoj vrednosti. (APSO)
3. Sistem **traži** poslovne partnere po zadatoj vrednosti. (SO)
4. Sistem **prikazuje** korisniku pronađene poslovne partnere. (IA)
5. Korisnik **bira** poslovnog partnera. (APUSO)
6. Korisnik **poziva** sistem da nađe sve podatke o poslovnom partneru. (APSO)
7. Sistem **traži** podatke o izabranom poslovnom partneru. (SO)
8. Sistem **prikazuje** korisniku sve podatke o poslovnom partneru. (IA)
9. Korisnik **unosi (menja)** podatke o poslovnom partneru. (APUSO)
10. Korisnik **kontroliše** da li je korektno uneo podatke o poslovnom partneru. (ANSO)
11. Korisnik **poziva** sistem da zapamti podatke o poslovnom partneru. (APSO)
12. Sistem **pamti** podatke o poslovnom partneru. (SO)
13. Sistem **prikazuje** korisniku zapamćenog poslovnog partnera i poruku: “Sistem je zapamtio poslovnog partnera.” (IA)

##### **Alternativna scenarija**

4.1 Ukoliko sistem ne može da nađe nijedanog poslovnog partnera on prikazuje korisniku poruku: “Sistem ne može da nađe poslovnog partnera po zadatoj vrednosti”. (IA)

8.1 Ukoliko sistem ne može da nađe podatke o poslovnom partneru on prikazuje korisniku poruku: “Sistem ne može da nađe poslovnog partnera po zadatoj vrednosti”. (IA)

13.1 Ukoliko sistem ne može da zapamti podatke o izmenjenom poslovnom partneru on prikazuje korisniku poruku “Sistem ne može da zapamti izmenjenog poslovnog partnera ”. Prekida se izvršenje scenaria. (IA)

### Slučaj korišćenja 9: Prijavljivanje korisnika

##### **Naziv SK:**

Prijavljivanje korisnika

##### **Aktori SK:**

Korisnik

##### **Učesnici SK:**

Korisnik i sistem (program)

##### **Preduslov:** Sistem je uključen i prikazuje formu za prijavu.

##### **Osnovni scenario SK:**

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

##### **Alternativna scenarija:**

4.1 Ukoliko sistem ne može da nađe korisnika prikazuje poruku da korisnik nije uneo tačne podatke. (IA)

# Analiza

## Dijagrami sekvenci

### DS1: Dijagram sekvenci slučaja korišćenja –Unos fakture

1. Korisnik **poziva** sistem da kreira novu fakturu. (APSO)
2. Sistem **prikazuje** korisniku novu fakturu i poruku: “Sistem je kreirao novu fakturu”. (IA)
3. Korisnik **poziva** sistem da zapamti podatke o fakturi. (APSO)
4. Sistem **prikazuje** korisniku zapamćenu fakturu i poruku: “Sistem je sačuvao fakturu”. (IA)
5. Korisnik **poziva** sistem da obradi fakturu. (APSO)
6. Sistem **prikazuje** korisniku obrađenu fakturu i poruku: “Sistem je obradio fakturu”. (IA)

##### 

Slika 2 - Dijagram sekvenci SK Unos fakture, osnovni scenario

##### **Alternativna scenarija**

2.1 Ukoliko sistem ne može da kreira fakturu on prikazuje korisniku poruku: “Sistem ne može da kreira novu fakturu”. Prekida se izvršenje scenarija. (IA)



Slika 3 - - Dijagram sekvenci SK Unos fakture, alternativni scenario

4.1 Ukoliko sistem ne može da zapamti podatke o fakturi on prikazuje korisniku poruku “Sistem ne može da zapamti fakturu”. Prekida se izvršenje scenarija. (IA)



Slika 4 - Dijagram sekvenci SK Unos fakture, alternativni scenario

6.1 Ukoliko sistem ne može da obradi fakturu on prikazuje korisniku poruku “Sistem ne može da obradi fakturu”. Prekida se izvršenje scenarija. (IA)



Slika 5 - Dijagram sekvenci SK Unos fakture, alternativni scenario

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

1. signal **KreirajNovu**(Faktura);
2. signal **Zapamti**(Faktura);
3. signal **Obradi**(Faktura);

### DS2: Dijagram sekvenci slučaja korišćenja – Izmena fakture

1. Korisnik **poziva** sistem da nađe fakture po zadatoj vrednosti. (APSO)
2. Sistem **prikazuje** korisniku pronađene fakture. (IA)
3. Korisnik **poziva** sistem da nađe sve podatke o izabranoj fakturi. (APSO)
4. Sistem **prikazuje** korisniku sve podatke o fakturi. (IA)
5. Korisnik **poziva** sistem da zapamti podatke o fakturi. (APSO)
6. Sistem **prikazuje** korisniku zapamćenu fakturu i poruku: “Sistem je zapamtio fakturu.” (IA)
7. Korisnik **poziva** sistem da obradi fakturu. (APSO)
8. Sistem **prikazuje** prodavcu obrađenu fakturu i poruku: “Sistem je obradio fakturu.” (IA)

##### 

Slika 6 - Dijagram sekvenci SK Izmena fakture, osnovni scenario

##### **Alternativna scenarija**

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



Slika 7 - Dijagram sekvenci SK Izmena fakture, alternativni scenario

4.1 Ukoliko sistem ne može da nađe podatke o fakturi on prikazuje korisniku poruku: “Sistem ne može da nađe fakturu po zadatoj vrednosti”. Prekida se izvršenje scenarija. (IA)



Slika 8 - Dijagram sekvenci SK Izmena fakture, alternativni scenario

6.1 Ukoliko sistem ne može da zapamti podatke o izmenjenoj fakturi on prikazuje korisniku poruku “Sistem ne može da zapamti novu fakturu”. Prekida se izvršenje scenarija. (IA)



Slika 9 - Dijagram sekvenci SK Izmena fakture, alternativni scenario

8.1 Ukoliko sistem ne može da obradi fakturu on prikazuje korisniku poruku: “Sistem ne može da obradi fakturu”. Prekida se Izvršenje scenarija. (IA)



Slika 10 - Dijagram sekvenci SK Izmena fakture, alternativni scenario

Sa navedenih sekvencnih dijagrama uočavaju se 4 sistemske operacije:

1. signal PretraziFakture(List<Faktura>, KriterijumPretrage);
2. signal NadjiFakturu(Faktura, KriterijumPretrage);
3. signal Zapamti(Faktura);
4. signal Obradi(Faktura);

### DS3: Dijagram sekvenci slučaja korišćenja – Storniranje fakture

1. Korisnik **poziva** sistem da nađe fakture po zadatoj vrednosti. (APSO)
2. Sistem **prikazuje** korisniku pronađene fakture. (IA)
3. Korisnik **poziva** sistem da nađe sve podatke o izabranoj fakturi. (APSO)
4. Sistem **prikazuje** korisniku sve podatke o fakturi. (IA)
5. Korisnik **poziva** sistem da stornira zadatu fakturu. (APSO)
6. Sistem **prikazuje** korisniku storniranu fakturu i poruku: “Sistem je stornirao fakturu.” (IA)

##### 

Slika 11 - Dijagram sekvenci SK Storniranje fakture, osnovni scenario

##### **Alternativna scenarija**

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



Slika 12 - Dijagram sekvenci SK Storniranje fakture, alternativni scenario

4.1 Ukoliko sistem ne može da nađe podatke o fakturi on prikazuje korisniku poruku: “Sistem ne može da nađe fakturu po zadatoj vrednosti”. Prekida se izvršenje scenarija. (IA)



Slika 13 - Dijagram sekvenci SK Storniranje fakture, alternativni scenario

6.1 Ukoliko sistem ne može da stornira fakturu on prikazuje korisniku poruku: “Sistem ne može da stornira fakturu”. Prekida se izvršenje scenarija. (IA)



Slika 14 - Dijagram sekvenci SK Storniranje fakture, alternativni scenario

Sa navedenih sekvencnih dijagrama uočavaju se 3 sistemske operacije:

1. signal PretraziFakture(List<Fatura>, KriterijumPretrage);
2. signal NadjiFakturu(Faktura, KriterijumPretrage);
3. signal Storniraj(Faktura);

### DS4: Dijagram sekvenci slučaja korišćenja – Pretraga faktura

1. Korisnik **poziva** sistem da nađe fakturu po zadatoj vrednosti. (APSO)
2. Sistem **prikazuje** korisniku pronađene fakture. (IA)
3. Korisnik **poziva** sistem da nađe sve podatke o izabranoj fakturi. (APSO)
4. Sistem **prikazuje** korisniku sve podatke o fakturi. (IA)

##### 

Slika 15 - Dijagram sekvenci SK Pretraga faktura, osnovni scenario

##### **Alternativna scenarija**

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



Slika 16 - Dijagram sekvenci SK Pretraga faktura, alternativni scenario

4.1 Ukoliko sistem ne može da nađe podatke o fakturi on prikazuje korisniku poruku: “Sistem ne može da nađe fakturu po zadatoj vrednosti”. Prekida se izvršenje scenarija. (IA)



Slika 17 - Dijagram sekvenci SK Pretraga faktura, alternativni scenario

Sa navedenih sekvencnih dijagrama uočavaju se 2 sistemske operacije:

1. signal PretraziFakture(List<Faktura>, KriterijumPretrage);
2. signal NadjiFakturu(Faktura, KriterijumPretrage);

### DS5: Dijagram sekvenci slučaja korišćenja – Unos novog proizvoda

1. Korisnik **poziva** sistem da kreira novu proizvod. (APSO)
2. Sistem **prikazuje** korisniku novi proizvod i poruku: “Sistem je kreirao novi proizvod”. (IA)
3. Korisnik **poziva** sistem da zapamti podatke o proizvodu. (APSO)
4. Sistem **prikazuje** korisniku zapamćeni proizvod i poruku: “Sistem je sačuvao proizvod”. (IA)

##### 

Slika 18 - Dijagram sekvenci SK Unos novog proizvoda, osnovni scenario

##### **Alternativna scenarija**

2.1 Ukoliko sistem ne može da kreira proizvod on prikazuje korisniku poruku: “Sistem ne može da kreira novi proizvod”. Prekida se izvršenje scenarija. (IA)



Slika 19 - Dijagram sekvenci SK Unos novog proizvoda, alternativni scenario

4.1 Ukoliko sistem ne može da zapamti podatke o proizvodu on prikazuje korisniku poruku “Sistem ne može da zapamti proizvod”. Prekida se izvršenje scenarija. (IA)



Slika 20 - Dijagram sekvenci SK Unos novog proizvoda, alternativni scenario

Sa navedenih sekvencnih dijagrama uočavaju se 2 sistemske operacije:

1. signal KreirajNoviProizvod(Proizvod);
2. signal ZapamtiProizvod(Proizvod);

### DS6: Dijagram sekvenci slučaja korišćenja – Izmena proizvoda

1. Korisnik **poziva** sistem da nađe proizvode po zadatoj vrednosti. (APSO)
2. Sistem **prikazuje** korisniku pronađene proizvode. (IA)
3. Korisnik **poziva** sistem da nađe sve podatke o izabranom proizvodu. (APSO)
4. Sistem **prikazuje** korisniku sve podatke o proizvodu. (IA)
5. Korisnik **poziva** sistem da zapamti podatke o proizvodu. (APSO)
6. Sistem **prikazuje** korisniku zapamćen proizvod i poruku: “Sistem je zapamtio proizvod.” (IA)

##### 

Slika 21 - Dijagram sekvenci SK Izmena proizvoda, osnovni scenario

##### **Alternativna scenarija**

2.1 Ukoliko sistem ne može da nađe nijedan proizvod on prikazuje korisniku poruku: “Sistem ne može da nađe proizvod po zadatoj vrednosti”. (IA)



Slika 22 - Dijagram sekvenci SK Izmena proizvoda, alternativni scenario

4.1 Ukoliko sistem ne može da nađe podatke o proizvodu on prikazuje korisniku poruku: “Sistem ne može da nađe proizvod po zadatoj vrednosti”. Prekida se izvršenje scenarija. (IA)



Slika 23 - Dijagram sekvenci SK Izmena proizvoda, alternativni scenario

6.1 Ukoliko sistem ne može da zapamti podatke o izmenjenom proizvodu on prikazuje korisniku poruku “Sistem ne može da zapamti izmenjeni proizvod”. Prekida se izvršenje scenarija. (IA)



Slika 24 - Dijagram sekvenci SK Izmena proizvoda, alternativni scenario

Sa navedenih sekvencnih dijagrama uočavaju se 3 sistemske operacije:

1. signal PretraziProizvode(List<Proizvod>, KriterijumPretrage);
2. signal NadjiProizvod(Proizvod, KriterijumPretrage);
3. signal ZapamtiProizvod(Proizvod);

### DS7: Dijagram sekvenci slučaja korišćenja – Unos poslovnog partnera

1. Korisnik **poziva** sistem da kreira novog poslovnog partnera. (APSO)
2. Sistem **prikazuje** korisniku novog poslovnog partnera i poruku: “Sistem je kreirao novog poslovnog partnera”. (IA)
3. Korisnik **poziva** sistem da zapamti podatke o poslovnom partneru. (APSO)
4. Sistem **prikazuje** korisniku zapamćenog poslovnog partnera i poruku: “Sistem je sačuvao poslovnog partnera”. (IA)

##### 

Slika 25 - Dijagram sekvenci SK Unos poslovnog partnera, osnovni scenario

##### **Alternativna scenarija**

2.1 Ukoliko sistem ne može da kreira poslovnog partnera on prikazuje korisniku poruku: “Sistem ne može da kreira novog poslovnog partnera”. Prekida se izvršenje scenarija. (IA)



Slika 26 - Dijagram sekvenci SK Unos poslovnog partnera, alternativni scenario

4.1 Ukoliko sistem ne može da zapamti podatke o poslovnom partneru on prikazuje korisniku poruku “Sistem ne može da zapamti poslovnog partnera”. Prekida se izvršenje scenarija. (IA)



Slika 27 - Dijagram sekvenci SK Unos poslovnog partnera, alternativni scenario

Sa navedenih sekvencnih dijagrama uočavaju se 2 sistemske operacije:

1. signal KreirajNovogPoslovnogPartnera(PoslovniPartner);
2. signal ZapamtiPoslovnogPartnera(PoslovniPartner);

### DS8: Dijagram sekvenci slučaja korišćenja – Izmena poslovnog partnera

1. Korisnik **poziva** sistem da nađe poslovne partnere po zadatoj vrednosti. (APSO)
2. Sistem **prikazuje** korisniku pronađene poslovne partnere. (IA)
3. Korisnik **poziva** sistem da nađe sve podatke o poslovnom partneru. (APSO)
4. Sistem **prikazuje** korisniku sve podatke o poslovnom partneru. (IA)
5. Korisnik **poziva** sistem da zapamti podatke o poslovnom partneru. (APSO)
6. Sistem **prikazuje** korisniku zapamćenog poslovnog partnera i poruku: “Sistem je zapamtio poslovnog partnera.” (IA)

##### 

Slika 28 - Dijagram sekvenci SK Izmena poslovnog partnera, osnovni scenario

##### **Alternativna scenarija**

2.1 Ukoliko sistem ne može da nađe nijedanog poslovnog partnera on prikazuje korisniku poruku: “Sistem ne može da nađe poslovnog partnera po zadatoj vrednosti”. Prekida se izvršenje scenarija. (IA)



Slika 29 - Dijagram sekvenci SK Izmena poslovnog partnera, alternativni scenario

4.1 Ukoliko sistem ne može da nađe podatke o poslovnom partneru on prikazuje korisniku poruku: “Sistem ne može da nađe poslovnog partnera po zadatoj vrednosti”. Prekida se izvršenje scenarija. (IA)



Slika 30 - Dijagram sekvenci SK Izmena poslovnog partnera, alternativni scenario

6.1 Ukoliko sistem ne može da zapamti podatke o izmenjenom poslovnom partneru on prikazuje korisniku poruku “Sistem ne može da zapamti izmenjenog poslovnog partnera ”. Prekida se izvršenje scenarija. (IA)



Slika 31 - Dijagram sekvenci SK Izmena poslovnog partnera, alternativni scenario

Sa navedenih sekvencnih dijagrama uočavaju se 3 sistemske operacije:

1. signal PretraziPoslovnePartnere(List<PoslovniPartner>, KriterijumPretrage);
2. signal NadjiPoslovnogPartnera(PoslovniPartner, KriterijumPretrage);
3. signal ZapamtiPoslovnogPartnera(PoslovniPartner);

### DS9: Dijagram sekvenci slučaja korišćenja – Prijavljivanje korisnika

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

##### 

Slika 32 - Dijagram sekvenci SK Prijavljivanje korisnika, osnovni scenario

##### **Alternativna scenarija**

2.1 Ukoliko sistem ne može da nađe korisnika prikazuje poruku da korisnik nije uneo tačne podatke. (IA)



Slika 33 - Dijagram sekvenci SK Prijavljivanje korisnika, alternativni scenario

Sa navedenih sekvencnih dijagrama uočava se 1 sistemska operacija:

1. signal NadjiKorisnika(Korisnik, KriterijumPretrage);

## Ugovori o sistemskim operacijama

### Ugovor UG1: KreirajNovu

**Operacija: KreirajNovu**(Faktura):signal;

**Veza sa SK**: SK1

**Preduslovi**:

**Postuslovi**: Napravljena je nova faktura.   
 Zadovoljena strukturna ograničenja nad tabelom Faktura.

### Ugovor UG2: Zapamti

**Operacija: Zapamti**(Faktura):signal;

**Veza sa SK**: SK1, SK2

**Preduslovi**: Ako je faktura obrađena ili stornirana ne može se izvršiti sistemska operacija.  
 Prosto vrednosno ograničenje nad *Faktura* objektom mora biti zadovoljeno.

**Postuslovi**: Izračunata je vrednost svake od stavki fakture.  
 Izračunata je ukupna vrednost fakture.   
 Zadovoljena strukturna ograničenja nad tabelom Faktura.

### Ugovor UG3: Obradi

**Operacija: Obradi**(Faktura):signal;

**Veza sa SK**: SK1, SK2

**Preduslovi**: Ako je faktura obrađena ili stornirana ne može se izvršiti sistemska operacija.  
 Prosto vrednosno ograničenja nad Faktura objektom mora biti zadovoljeno.

**Postuslovi**: Izračunata je vrednost svake od stavki fakture.  
 Izračunata je ukupna vrednost fakture.  
 Faktura je obrađena.  
 Zadovoljena strukturna ograničenj nad tabelom Faktura.

### Ugovor UG4: PretraziFakture

**Operacija: PretraziFakture**(List<Faktura>, kriterijumPretrage):signal;

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

**Preduslovi**:

**Postuslovi**:

### Ugovor UG5: NadjiFakturu

**Operacija: NadjiFakturu**(Faktura, kriterijumPretrage):signal;

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

**Preduslovi**:

**Postuslovi**:

### Ugovor UG6: Storniraj

**Operacija: Storniraj**(Faktura):signal;

**Veza sa SK**: SK3

**Preduslovi**: Ako je faktura stornirana ne može se izvršiti sistemska operacija.

**Postuslovi**: Faktura je stornirana.   
 Zadovoljena strukturna ograničenja nad tabelom Faktura.

### Ugovor UG7: KreirajNoviProizvod

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

**Veza sa SK**: SK5

**Preduslovi**:

**Postuslovi**: Kreiran je novi proizvod.   
 Zadovoljena strukturna ograničenja nad tabelom Proizvod.

### Ugovor UG8: ZapamtiProizvod

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

**Veza sa SK**: SK5, SK6

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

**Postuslovi**: Zapamćen je novi proizvod.   
 Zadovoljena strukturna ograničenja nad tabelom Proizvod.

### Ugovor UG9: PretraziProizvode

**Operacija: PretraziProizvode**(List<Proizvod>, kriterijumPretrage):signal;

**Veza sa SK**: SK6

**Preduslovi**:

**Postuslovi**:

### Ugovor UG10: NadjiProizvod

**Operacija: NadjiProizvod**(Proizvod, kriterijumPretrage):signal;

**Veza sa SK**: SK6

**Preduslovi**:

**Postuslovi**:

### Ugovor UG11: KreirajNovogPoslovnogPartnera

**Operacija: KreirajNovogPoslovnogPartnera**(PoslovniPartner):signal;

**Veza sa SK**: SK7

**Preduslovi**:

**Postuslovi**: Kreiran je novi poslovni partner.   
 Zadovoljena strukturna ograničenja nad tabelom PoslovniPartner.

### Ugovor UG12: ZapamtiPoslovnogPartnera

**Operacija: ZapamtiPoslovnogPartnera**(PoslovniPartner):signal;

**Veza sa SK**: SK7, SK8

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

**Postuslovi**: Zapamćen je novi poslovni partner.   
 Zadovoljena strukturna ograničenja nad tabelom PoslovniPartner.

### Ugovor UG13: PretraziPoslovnePartnere

**Operacija: PretraziPoslovnePartnere**(List<PoslovniPartner>, kriterijumPretrage):signal;

**Veza sa SK**: SK8

**Preduslovi**:

**Postuslovi**:

### Ugovor UG14: NadjiPoslovnogPartnera

**Operacija: NadjiPoslovnogPartnera**(PoslovniPartner, kriterijumPretrage):signal;

**Veza sa SK**: SK8

**Preduslovi**:

**Postuslovi**:

### Ugovor UG15: NadjiKorisnika

**Operacija: NadjiKorisnika**(Korisnik, kriterijumPretrage):signal;

**Veza sa SK**: SK9

**Preduslovi**:

**Postuslovi**:

## Konceptualni model

Struktura softverskog sistema prikazana je konceptualnim modelom na sledećoj slici.



Slika 34 - Konceptualni model

## Relacioni model

Na osnovu konceptualnog modela dobijamo relacioni model:

**Korisnik**(korisnikID, ime, prezime, email, korisnickoIme, korisnickaSifra)

**Mesto**(ptt, nazivMesta)

**PoslovniPartner**(poslovniPartnerID, pib, naziv, ulica, broj, kontakt, *ptt*)

**Proizvod**(proizvodID, naziv, cena, jedMere)

**Faktura**(idFakture, brojFakture, brojOtpremnice, datum,osnovica, pdv, ukupno, obradjena, stornirana, *poslovniPartnerID*)

**StavkaFakture**(brojFakture, rb, kolicina, rabat, iznosRabata, poreskaOsnovica, pdv, iznosPDVa, vrednostSaPDVom, *proizvodID*)

Tabela 1 - Ograničenja nad tabelom Korisnik

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela Korisnik | | Prosto vrednosno ograničenje | | Složeno vrednosno ograničenje | | Strukturno ograničenje |
| Atributi | Ime | Tip atributa | Vrednost atributa | Međuzavisnost atributa jedne tabele | Međuzavisnost atributa više tabela | INSERT /  UPDATE/  DELETE/ |
| korisnikID | Int | not null |  |  |
| Ime | String | not null |  |  |
| Prezime | String | not null |  |  |
| Email | String | not null |  |  |
| korisnickoIme | String | not null |  |  |
| korisnickaSifra | String | not null |  |  |

Tabela 2 - Ograničenja nad tabelom Mesto

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela Mesto | | 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 PoslovniPartner  DELETE RESTRICT PoslovniPartner |
| Ptt | Int | Not null |  |  |
| naziv | String | Not null |  |  |

Tabela 3 - Ograničenja nad tabelom PoslovniPartner

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela PoslovniPartner | | Prosto vrednosno ograničenje | | Složeno vrednosno ograničenje | | Strukturno ograničenje |
| Atributi | Ime | Tip atributa | Vrednost atributa | Međuzavisnost atributa jedne tabele | Međuzavisnost atributa više tabela | INSERT RESTRICTED Mesto  UPDATE CASCADES Faktura  RESTRICTED Mesto  DELETE RESTRICTED Faktura  RESTRICTED Mesto |
| poslovniPartnerID | Int | Not null |  |  |
| pib | String | Not null |  |  |
| naziv | String | Not null |  |  |
| ulica | String | Not null |  |  |
| broj | String | Not null |  |  |
| kontakt | String | Not null |  |  |
| ptt | int | Not null |  |  |

Tabela 4 - Ograničenja nad tabelom Proizvod

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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 /  UPDATE CASCADES StavkaFakture  DELETE  RESTRICTED StavkaFakture |
| proizvodID | Int | Not null |  |  |
| naziv | String | Not null |  |  |
| cena | Double | Not null and >0 |  |  |
| jedMere | String | Not null |  |  |

Tabela 5 - Ograničenja nad tabelom StavkaFakture

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela StavkaFakture | | Prosto vrednosno ograničenje | | Složeno vrednosno ograničenje | | Strukturno ograničenje |
| Atributi | Ime | Tip atributa | Vrednost atributa | Međuzavisnost atributa jedne tabele | Međuzavisnost atributa više tabela | INSERT RESTRECTED Faktura RESTRICTED Proizvod  UPDATE RESTRECTED Faktura  DELETE  RESTRICTED Proizvod |
| brojFakture | String | Not null |  |  |
| rb | Int | Not null and >0 |  |  |
| kolicina | Double | >0 (default: 0) |  |  |
| rabat | Double | >0 (default: 7) |  |  |
| iznosRabata | Double | >0 (default: 0) |  | iznosRabata = kolicina \* proizvod.cena \*rabat |
| poreskaOsnovica | Double | >0 (default: 0) |  | poreskaOsnovica = kolicina \* proizvod.cena - iznosRabata |
| pdv | Double | >0, <1 (default: 0.2) |  |  |
| iznosPDVa |  |  | iznosPDVa = poreskaOsnovica \* pdv |  |
| vrednostSaPDVom | Double | >0 (default: 0) | vrednostSaPDVom = poreskaOsnovica + iznosPDVa |  |
| proizvodID | Int | Not null |  |  |

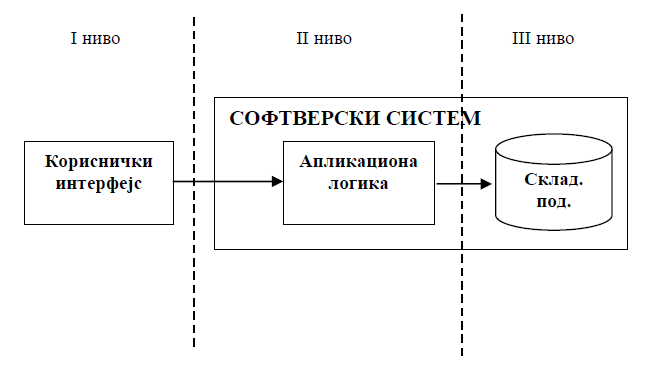
Tabela 6 - Ograničenja nad tabelom Faktura

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela Faktura | | Prosto vrednosno ograničenje | | Složeno vrednosno ograničenje | | Strukturno ograničenje |
| Atributi | Ime | Tip atributa | Vrednost atributa | Međuzavisnost atributa jedne tabele | Međuzavisnost atributa više tabela | INSERT RESTRECTED PoslovniPartner  UPDATE CASCADES StavkaFakture  RESTRICTED PoslovniPartner  DELETE CASCADES StavkaFakture  RESTRICTED  PoslovniPartner |
| idFakture | Int | Not null |  |  |
| brojFakture | String | Not null |  |  |
| datum | Date | Not null |  |  |
| osnovica | Double | (default:0) |  | osnovica = SUM( stavkaFakture.  poreskaOsnovica) |
| pdv | Double | (default:0) |  | pdv = SUM( stavkaFakture.  iznosPDVa) |
| ukupno | Double | (default:0) | Ukupno = osnovica + pdv |  |
| obradjena | Boolean | (default:false) |  |  |
| stornirana | Boolean | (default:false) |  |  |
| poslovniPartnerID | int | Not null |  |  |

# Projektovanje

### Arhitektura softverskog sistema

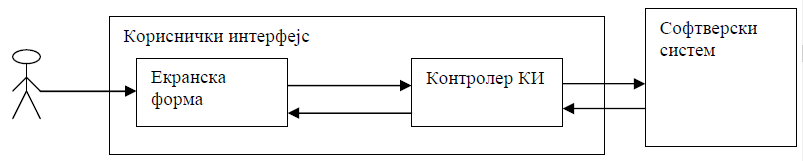
U ovom seminarskom radu korišćena je klasična tronivojska arhitektura, detaljnije prikazana na narednoj slici.



Slika 35 - Tronivojska arhitektura

## Projektovanje korisničkog interfejsa

Na narednoj slici je prikazana struktura korisničkog interfejsa.



Slika 36 - struktura korisničkog interfejsa

### Projektovanje ekranskih formi

#### Slučaj korišćenja 1: Unos nove fakture

##### **Naziv SK**

Unos nove fakture

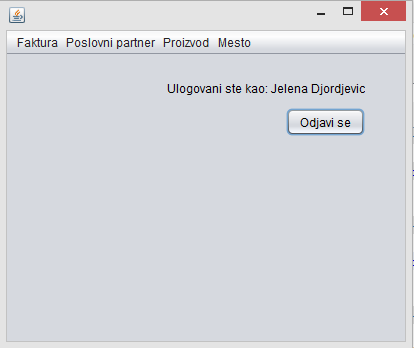
##### **Aktori SK**

Korisnik

##### **Učesnici SK**

Korisnik i sistem (program)

##### **Preduslov**: Sistem je uključen i korisnik je prijavljen pod svojom šifrom. Sistem prikazuje početnu formu.

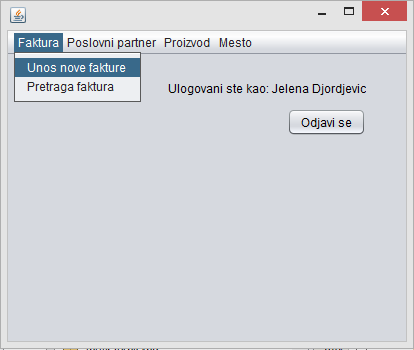


Slika 37 - Početna forma

##### **Osnovni scenario SK**

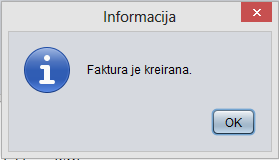
1. Korisnik **poziva** sistem da kreira novu fakturu. (APSO)

*Opis akcije*: Klikom na stavku menija Faktura, *Unos nove fakture,* korisnik poziva sistemsku operaciju ***kreiraj(Faktura)*** koja pravi novu fakturu.

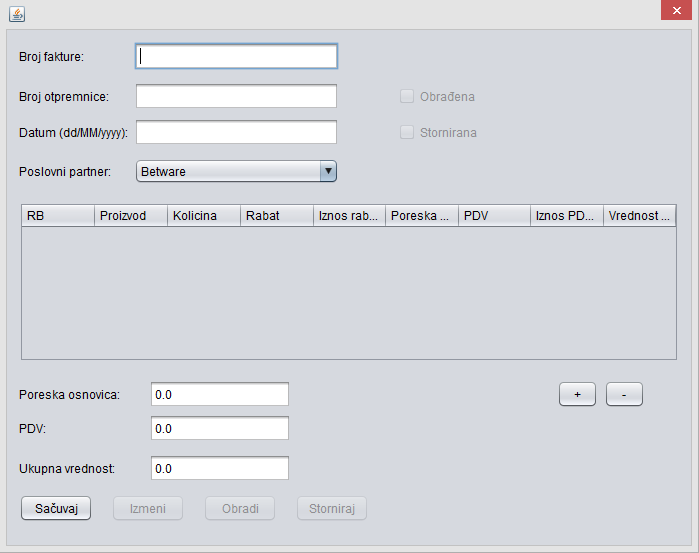


Slika 38 - Kreiranje nove fakture

1. Sistem **kreira** novu fakturu. (SO)
2. Sistem **prikazuje** korisniku novu fakturu i poruku: “Sistem je kreirao novu fakturu”. (IA)

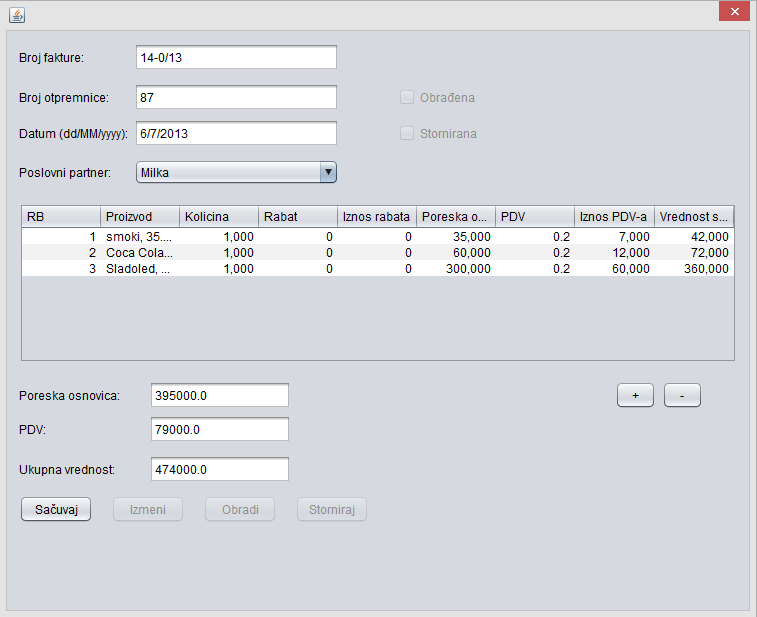


Slika 39 - Poruka



Slika 40 - Nova faktura

1. Korisnik **unosi** podatke u novu fakturu. (APUSO)

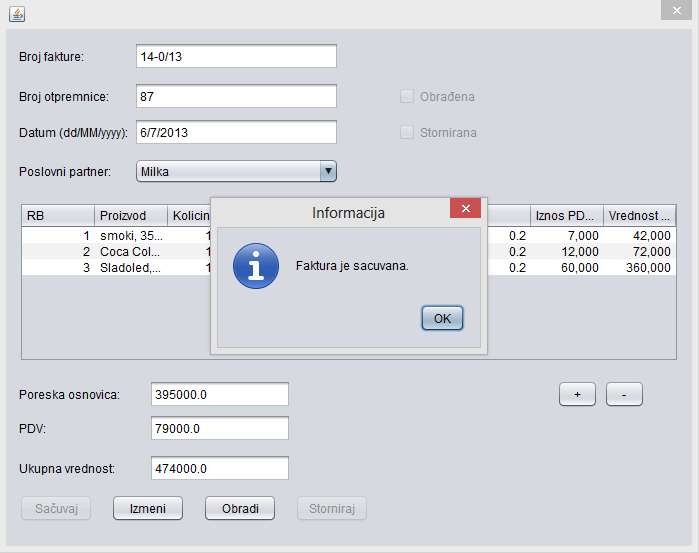


Slika 41 - Unos podataka u novu fakturu

1. Korisnik **kontroliše** da li je korektno uneo podatke u novu fakturu. (ANSO)
2. Korisnik **poziva** sistem da zapamti podatke o fakturi. (APSO)

*Opis akcije*: Korisnik klikom na dugme „Sačuvaj“ poziva sistemsku operaciju ***Zapamti(Faktura)*** koja čuva novu fakturu.

1. Sistem **pamti** podatke o fakturi. (SO)
2. Sistem **prikazuje** korisniku zapamćenu fakturu i poruku: “Sistem je sačuvao fakturu”. (IA)

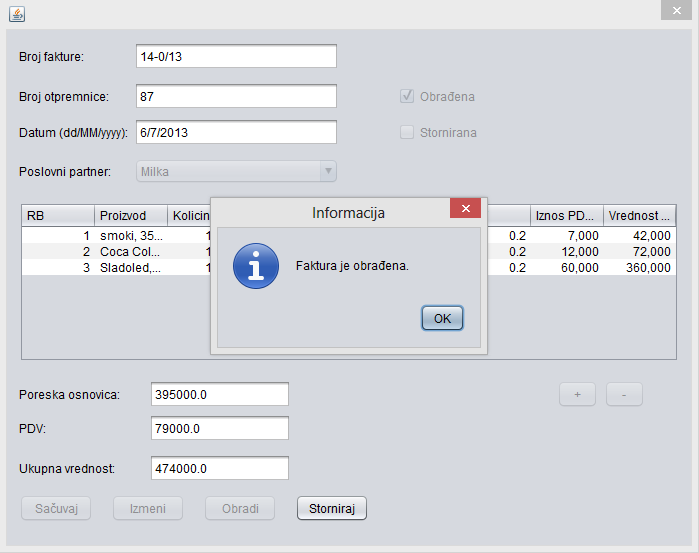


Slika 42 - Sačuvana faktura

1. Korisnik **poziva** sistem da obradi fakturu. (APSO)

*Opis akcije*: Korisnik klikom na dugme „Obradi“ poziva sistemsku operaciju ***Obradi(Faktura)*** koja obrađuje novu fakturu.

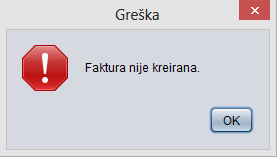
1. Sistem **obrađuje** fakturu. (SO)
2. Sistem **prikazuje** korisniku obrađenu fakturu i poruku: “Sistem je obradio fakturu”. (IA)



Slika 43 - Obrađena faktura

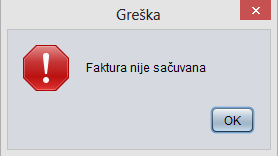
##### **Alternativna scenarija**

3.1 Ukoliko sistem ne može da kreira fakturu on prikazuje korisniku poruku: “Sistem ne može da kreira novu fakturu”. Prekida se izvršenje scenaria. (IA)



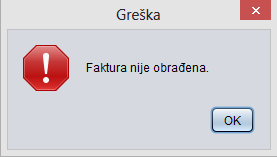
Slika 44 - Greška

8.1 Ukoliko sistem ne može da zapamti podatke o fakturi on prikazuje korisniku poruku “Sistem ne može da zapamti fakturu”. Prekida se izvršenje scenaria. (IA)



Slika 45 - Greška

11.1 Ukoliko sistem ne može da obradi fakturu on prikazuje korisniku poruku “Sistem ne može da obradi fakturu”. (IA)



Slika 46 - Greška

#### Slučaj korišćenja 2: Izmena fakture

##### **Naziv SK**

Izmena fakture

##### **Aktori SK**

Korisnik

##### **Učesnici SK**

Korisnik i sistem (program)

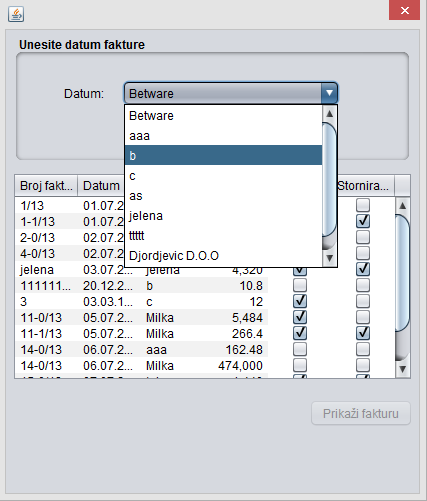
##### **Preduslov**: Sistem je uključen i korisnik je prijavljen pod svojom šifrom. Sistem prikazuje formu za pretragu faktura.



Slika 47 - Forma za pretragu faktura

##### **Osnovni scenario SK**

1. Korisnik **unosi** vrednost po kojoj pretražuje fakture. (APUSO)

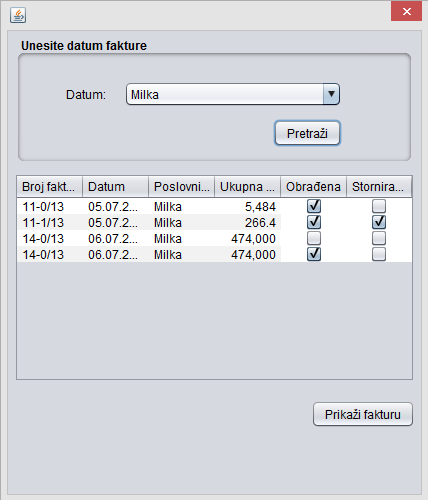


Slika 48 - Korisnik bira vrednost za pretragu

1. Korisnik **poziva** sistem da nađe fakture po zadatoj vrednosti. (APSO)

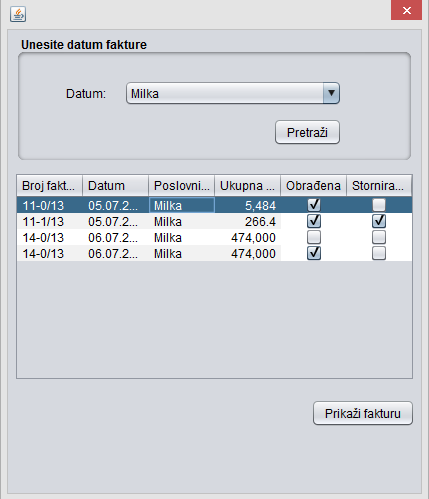
*Opis akcije:* Klikom na dugme “Pretraži” korisnik poziva sistemsku operaciju PretraziFakture(List<Faktura>, KriterijumPretrage) koja nalazi fakture po zadatoj vrednosti.

1. Sistem **traži** fakturu po zadatoj vrednosti. (SO)
2. Sistem **prikazuje** korisniku pronađene fakture. (IA)



Slika 49 - Sistem prikazuje fakture

1. Korisnik **bira** fakturu. (APUSO)

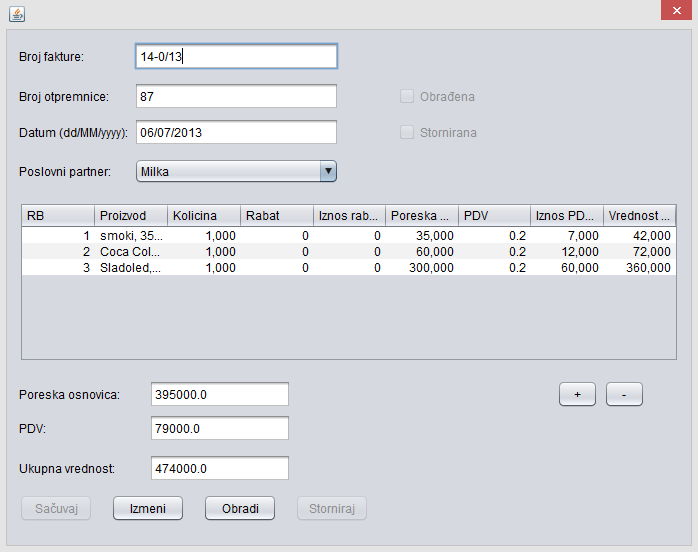


Slika 50 - Korisnik je izabrao fakturu

1. Korisnik **poziva** sistem da nađe sve podatke o izabranoj fakturi. (APSO)

*Opis akcije*: Klikom na dugme “Prikaži fakturu” korisnik poziva sistemsku operaciju NadjiFakturu(Faktura, KriterijumPretrage) koja nalazi podatke za izabranu fakturu.

1. Sistem **traži** podatke o izabranoj fakturi.(SO)
2. Sistem **prikazuje** korisniku sve podatke o fakturi. (IA)

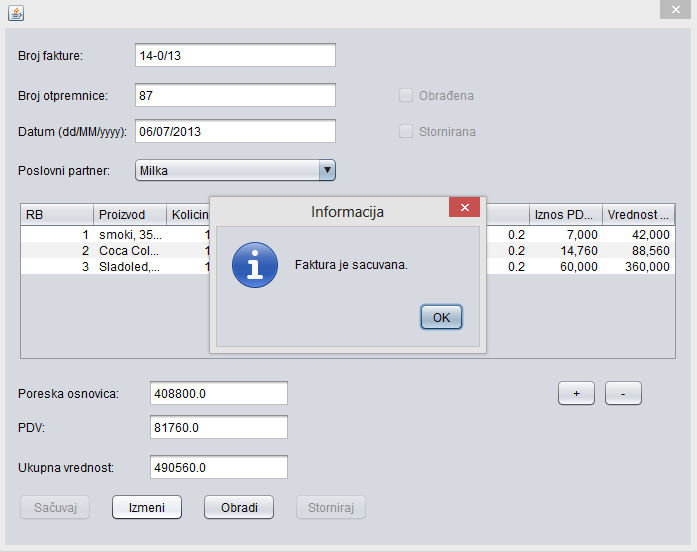


Slika 51 - Prikaz fakture

1. Korisnik **unosi (menja)** podatke o fakturi. (APUSO)
2. Korisnik **kontroliše** da li je korektno uneo podatke o fakturi. (ANSO)
3. Korisnik **poziva** sistem da zapamti podatke o fakturi. (APSO)

*Opis akcije*: Korisnik klikom na dugme „Izmeni“ poziva sistemsku operaciju ***Zapamti(Faktura)*** koja čuva izmenjenu fakturu.

1. Sistem **pamti** podatke o fakturi. (SO)
2. Sistem **prikazuje** korisniku zapamćenu fakturu i poruku: “Sistem je zapamtio fakturu.” (IA)

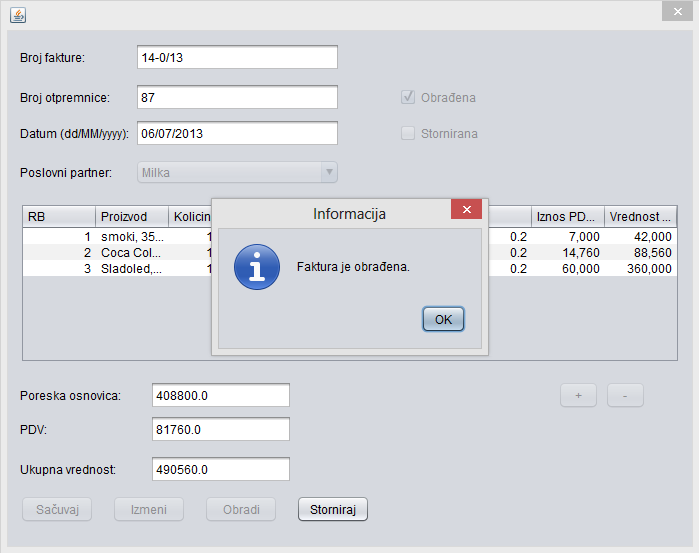


Slika 52 - Faktura je sačuvana

1. Korisnik **poziva** sistem da obradi fakturu. (APSO)

*Opis akcije*: Korisnik klikom na dugme „Obradi“ poziva sistemsku operaciju ***Obradi(Faktura)*** koja obrađujefakturu.

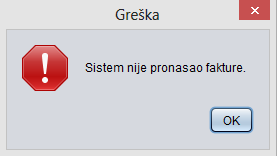
1. Sistem **obrađuje** fakturu. (SO)
2. Sistem **prikazuje** prodavcu obrađenu fakturu i poruku: “Sistem je obradio fakturu.” (IA)



Slika 53 – Faktura je obrađena

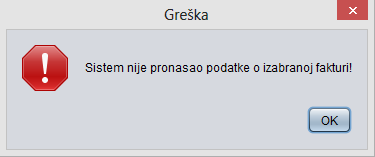
##### **Alternativna scenarija**

4.1 Ukoliko sistem ne može da nađe nijednu fakturu on prikazuje korisniku poruku: “Sistem ne može da nađe fakturu po zadatoj vrednosti”. (IA)



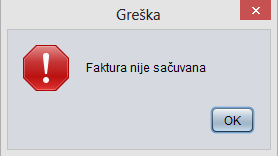
Slika 54 - Greška

8.1 Ukoliko sistem ne može da nađe podatke o fakturi on prikazuje korisniku poruku: “Sistem ne može da nađe fakturu po zadatoj vrednosti”. (IA)



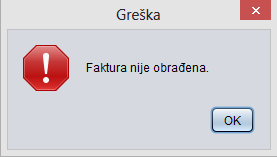
Slika 55 - Greška

13.1 Ukoliko sistem ne može da zapamti podatke o izmenjenoj fakturi on prikazuje korisniku poruku “Sistem ne može da zapamti novu fakturu”. Prekida se izvršenje scenaria. (IA)



Slika 56 - Greška

16.1 Ukoliko sistem ne može da obradi fakturu on prikazuje korisniku poruku: “Sistem ne može da obradi fakturu.” (IA)



Slika 57 - Greška

#### Slučaj korišćenja 3: Storniranje fakture

##### **Naziv SK**

Storniranje fakture

##### **Aktori SK**

Korisnik

##### **Učesnici SK**

Korisnik i sistem (program)

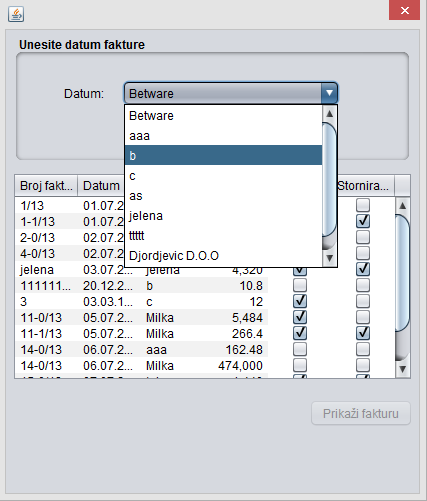
##### **Preduslov**: Sistem je uključen i korisnik je prijavljen pod svojom šifrom. Sistem prikazuje formu za pretragu faktura.



Slika 58 - Forma za pretragu faktura

##### **Osnovni scenario SK**

1. Korisnik **unosi** vrednost po kojoj pretražuje fakture. (APUSO)

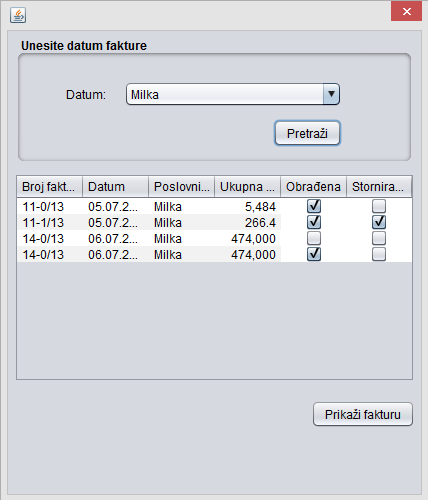


Slika 59 - Korisnik bira vrednost za pretragu

1. Korisnik **poziva** sistem da nađe fakture po zadatoj vrednosti. (APSO)

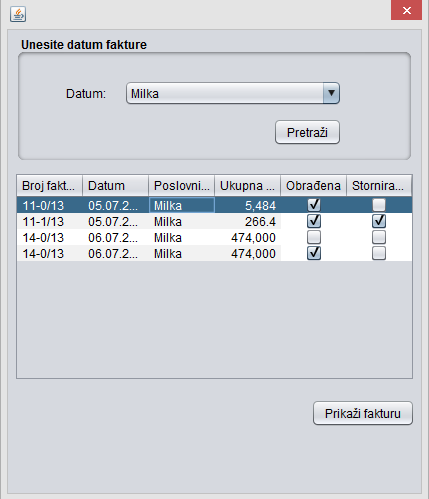
*Opis akcije:* Klikom na dugme “Pretraži” korisnik poziva sistemsku operaciju PretraziFakture(List<Faktura>, KriterijumPretrage) koja nalazi fakture po zadatoj vrednosti.

1. Sistem **traži** fakturu po zadatoj vrednosti. (SO)
2. Sistem **prikazuje** korisniku pronađene fakture. (IA)



Slika 60 - Sistem prikazuje fakture

1. Korisnik **bira** fakturu. (APUSO)

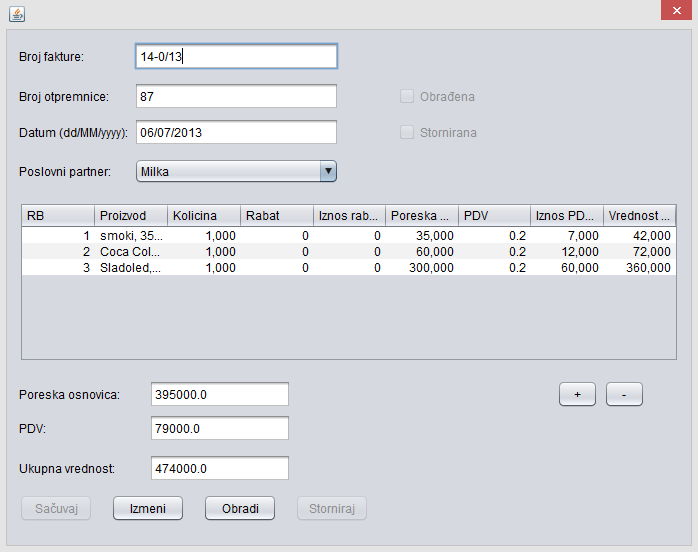


Slika 61 - Korisnik je izabrao fakturu

1. Korisnik **poziva** sistem da nađe sve podatke o izabranoj fakturi. (APSO)

*Opis akcije*: Klikom na dugme “Prikaži fakturu” korisnik poziva sistemsku operaciju NadjiFakturu(Faktura, KriterijumPretrage) koja nalazi podatke za izabranu fakturu.

1. Sistem **traži** podatke o izabranoj fakturi.(SO)
2. Sistem **prikazuje** korisniku sve podatke o fakturi. (IA)

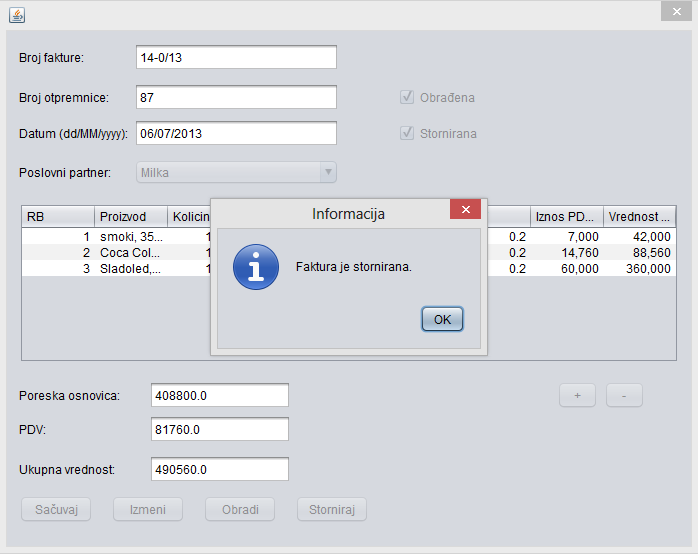


Slika 62 - Prikaz fakture

1. Korisnik **poziva** sistem da stornira zadatu fakturu. (APSO)
2. Sistem **stornira** fakturu. (SO)

*Opis akcije*: Klikom na dugme “Storniraj” korisnik poziva sistemsku operaciju Storniraj(Faktura) koja stornira fakturu.

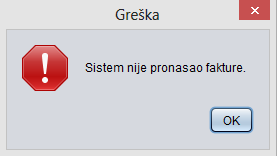
1. Sistem **prikazuje** korisniku storniranu fakturu i poruku: “Sistem je stornirao fakturu.” (IA)



Slika 63 - Faktura je stornirana

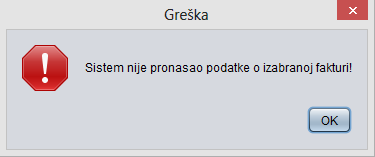
##### **Alternativna scenarija**

4.1 Ukoliko sistem ne može da nađe nijednu fakturu on prikazuje korisniku poruku: “Sistem ne može da nađe fakturu po zadatoj vrednosti”. (IA)



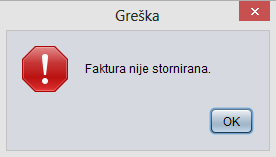
Slika 64 - Greška

8.1 Ukoliko sistem ne može da nađe podatke o fakturi on prikazuje korisniku poruku: “Sistem ne može da nađe fakturu po zadatoj vrednosti”. (IA)



Slika 65 - Greška

11.1 Ukoliko sistem ne može da stornira fakturu on prikazuje korisniku poruku: “Sistem ne može da stornira fakturu”. (IA)



Slika 66 - Greška

#### Slučaj korišćenja 4: Pretraga faktura

##### **Naziv SK**

Pretraga faktura

##### **Aktori SK**

Korisnik

##### **Učesnici SK**

Korisnik i sistem (program)

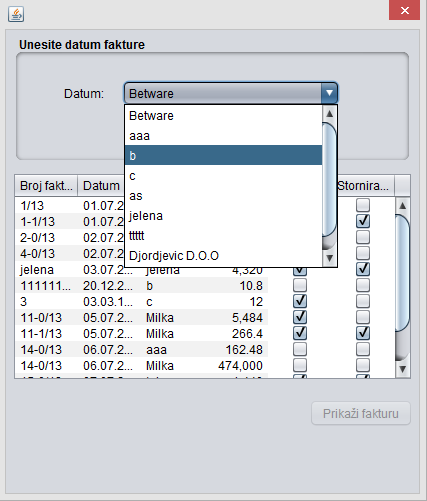
##### **Preduslov**: Sistem je uključen i korisnik je prijavljen pod svojom šifrom. Sistem prikazuje formu za pretragu faktura.



Slika 67 - Forma za pretragu faktura

##### **Osnovni scenario SK**

1. Korisnik **unosi** vrednost po kojoj pretražuje fakture. (APUSO)

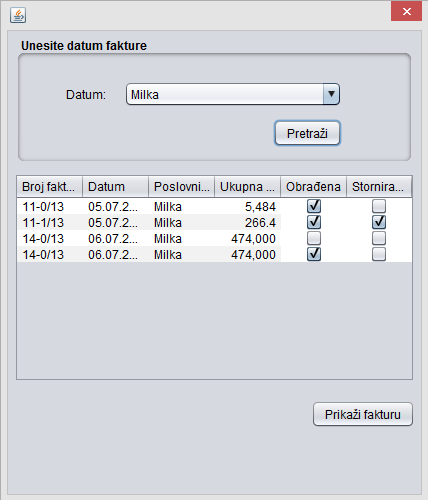


Slika 68 - Korisnik bira vrednost za pretragu

1. Korisnik **poziva** sistem da nađe fakture po zadatoj vrednosti. (APSO)

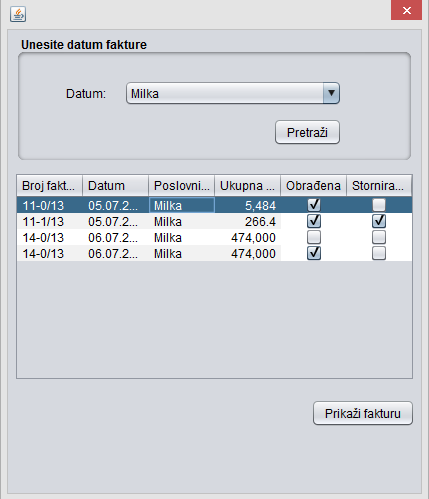
*Opis akcije:* Klikom na dugme “Pretraži” korisnik poziva sistemsku operaciju PretraziFakture(List<Faktura>, KriterijumPretrage) koja nalazi fakture po zadatoj vrednosti.

1. Sistem **traži** fakturu po zadatoj vrednosti. (SO)
2. Sistem **prikazuje** korisniku pronađene fakture. (IA)



Slika 69 - Sistem prikazuje fakture

1. Korisnik **bira** fakturu. (APUSO)

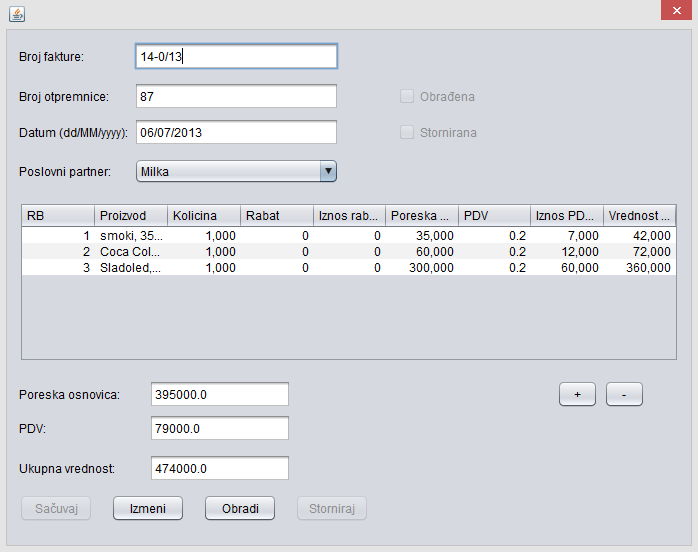


Slika 70 - Korisnik je izabrao fakturu

1. Korisnik **poziva** sistem da nađe sve podatke o izabranoj fakturi. (APSO)

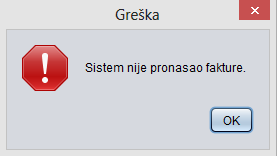
*Opis akcije*: Klikom na dugme “Prikaži fakturu” korisnik poziva sistemsku operaciju NadjiFakturu(Faktura, KriterijumPretrage) koja nalazi podatke za izabranu fakturu.

1. Sistem **traži** podatke o izabranoj fakturi.(SO)
2. Sistem **prikazuje** korisniku sve podatke o fakturi. (IA)



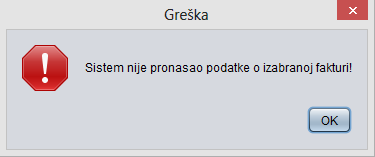
##### **Alternativna scenarija**

4.1 Ukoliko sistem ne može da nađe nijednu fakturu on prikazuje korisniku poruku: “Sistem ne može da nađe fakturu po zadatoj vrednosti”. (IA)



Slika 71 - Greška

8.1 Ukoliko sistem ne može da nađe podatke o fakturi on prikazuje korisniku poruku: “Sistem ne može da nađe fakturu po zadatoj vrednosti”. (IA)



Slika 72 - Greška

#### Slučaj korišćenja 5: Unos novog proizvoda

##### **Naziv SK**

Unos novog proizvoda

##### **Aktori SK**

Korisnik

##### **Učesnici SK**

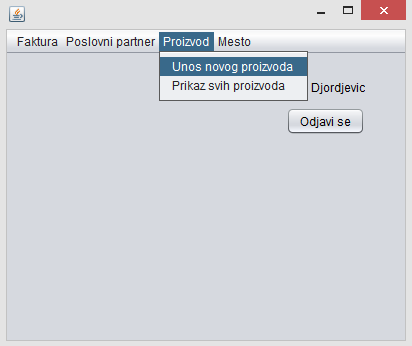
Korisnik i sistem (program)

##### **Preduslov**: Sistem je uključen i korisnik je prijavljen pod svojom šifrom. Sistem prikazuje početnu formu.

##### **Osnovni scenario SK**

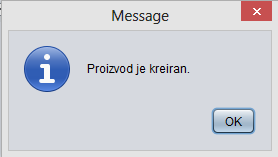
1. Korisnik **poziva** sistem da kreira novu proizvod. (APSO)

*Opis akcije*: Klikom na stavku menija Proizvod, *Unos novog proizvoda,* korisnik poziva sistemsku operaciju ***kreiraj(Proizvod)*** koja pravi novi proizvod.

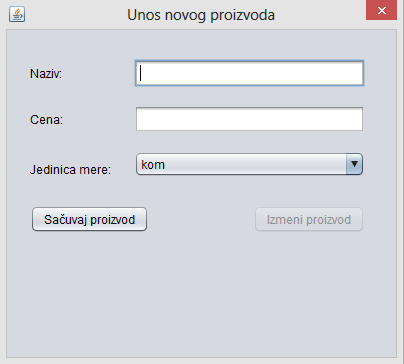


Slika 73 - Kreiranje novog proizvoda

1. Sistem **kreira** novi proizvod. (SO)
2. Sistem **prikazuje** korisniku novi proizvod i poruku: “Sistem je kreirao novi proizvod”. (IA)



Slika 74 – Poruka



Slika 75 - Kreiran novi proizvod

1. Korisnik **unosi** podatke o novom proizvodu. (APUSO)

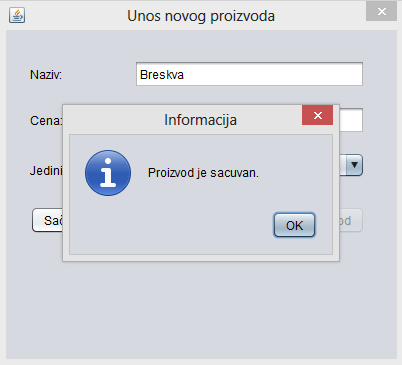


Slika 76 - Unos podataka

1. Korisnik **kontroliše** da li je korektno uneo podatke o proizvodu. (ANSO)
2. Korisnik **poziva** sistem da zapamti podatke o proizvodu. (APSO)

*Opis akcije*: Korisnik klikom na dugme „Sačuvaj“ poziva sistemsku operaciju ***Zapamti(Proizvod)*** koja čuva novi proizvod.

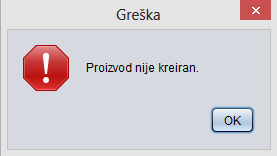
1. Sistem **pamti** podatke o proizvodu. (SO)
2. Sistem **prikazuje** korisniku zapamćeni proizvod i poruku: “Sistem je sačuvao proizvod”. (IA)



Slika 77 - Proizvod je sačuvan

##### **Alternativna scenarija**

3.1 Ukoliko sistem ne može da kreira proizvod on prikazuje korisniku poruku: “Sistem ne može da kreira novi proizvod”. Prekida se izvršenje scenaria. (IA)



Slika 78 - Greška

8.1 Ukoliko sistem ne može da zapamti podatke o proizvodu on prikazuje korisniku poruku “Sistem ne može da zapamti proizvod”. Prekida se izvršenje scenaria. (IA)



Slika 79 - Greška

#### Slučaj korišćenja 6: Izmena proizvoda

##### **Naziv SK**

Izmena proizvoda

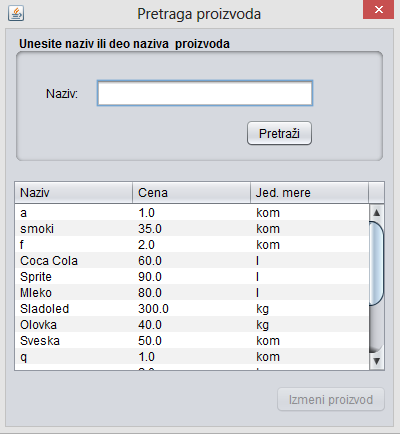
##### **Aktori SK**

Korisnik

##### **Učesnici SK**

Korisnik i sistem (program)

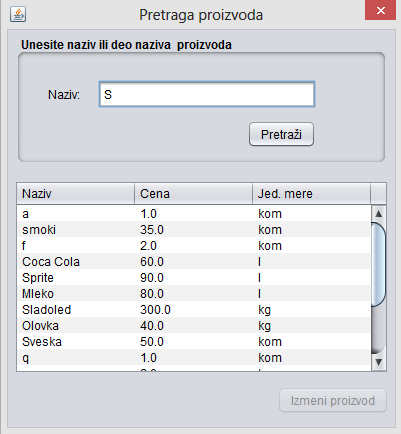
##### **Preduslov**: Sistem je uključen i korisnik je prijavljen pod svojom šifrom. Sistem prikazuje formu za pretragu proizvoda.



Slika 80 - Forma za pretragu proizvoda

##### **Osnovni scenario SK**

1. Korisnik **unosi** vrednost po kojoj pretražuje proizvode. (APUSO)

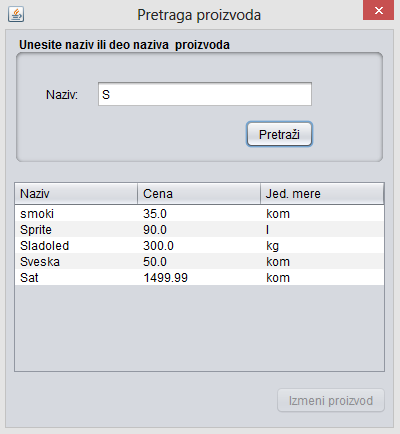


Slika 81 - Unos vrednosti za pretragu

1. Korisnik **poziva** sistem da nađe proizvode po zadatoj vrednosti. (APSO)

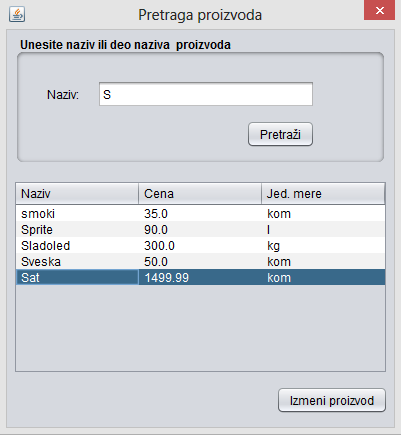
Opis akcije: Korisnik klikom na dugme “Pretraži” poziva sistemsku operaciju PretraziProizvode(List<Proizvod>, KriterijumPretrage) koja vraća listu proizvoda.

1. Sistem **traži** proizvod po zadatoj vrednosti. (SO)
2. Sistem **prikazuje** korisniku pronađene proizvode. (IA)



Slika 82 - Pronađeni proizvodi

1. Korisnik **bira** proizvod. (APUSO)

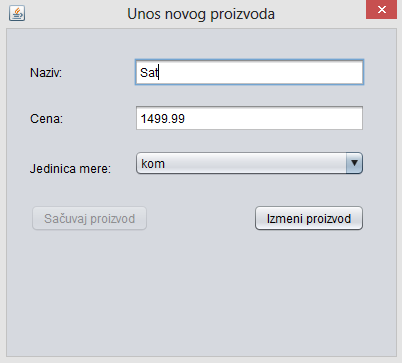


Slika 83 - Korisnik bira proizvod

1. Korisnik **poziva** sistem da nađe sve podatke o izabranom proizvodu. (APSO)

Opis akcije: Korisnik klikom na dugme “Izmeni proizvod” poziva sistemsku operaciju NadjiProizvod(Proizvod, KriterijumPretrage) koja vraća podatke o izabranom proizvodu.

1. Sistem **traži** podatke o izabranom proizvodu. (SO)
2. Sistem **prikazuje** korisniku sve podatke o proizvodu. (IA)



Slika 84 - Podaci o izabranom proizvodu

1. Korisnik **unosi (menja)** podatke o proizvodu. (APUSO)
2. Korisnik **kontroliše** da li je korektno uneo podatke o proizvodu. (ANSO)
3. Korisnik **poziva** sistem da zapamti podatke o proizvodu. (APSO)

*Opis akcije*: Korisnik klikom na dugme „Izmeni proizvod“ poziva sistemsku operaciju ***Zapamti(Proizvod)*** koja čuva proizvod.

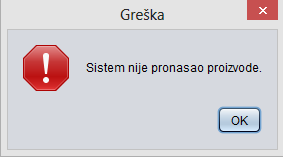
1. Sistem **pamti** podatke o proizvodu. (SO)
2. Sistem **prikazuje** korisniku zapamćen proizvod i poruku: “Sistem je zapamtio proizvod.” (IA)



Slika 85 - Proizvod je sačuvan

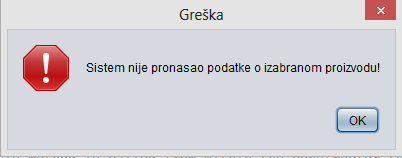
##### **Alternativna scenarija**

4.1 Ukoliko sistem ne može da nađe nijedan proizvod on prikazuje korisniku poruku: “Sistem ne može da nađe proizvod po zadatoj vrednosti”. (IA)



Slika 86 - Greška

8.1 Ukoliko sistem ne može da nađe podatke o proizvodu on prikazuje korisniku poruku: “Sistem ne može da nađe proizvod po zadatoj vrednosti”. (IA)



Slika 87 - Greška

13.1 Ukoliko sistem ne može da zapamti podatke o izmenjenom proizvodu on prikazuje korisniku poruku “Sistem ne može da zapamti izmenjeni proizvod”. Prekida se izvršenje scenaria. (IA)



Slika 88 - Greška

#### Slučaj korišćenja 7: Unos poslovnog partnera

##### **Naziv SK**

Unos novog poslovnog partnera

##### **Aktori SK**

Korisnik

##### **Učesnici SK**

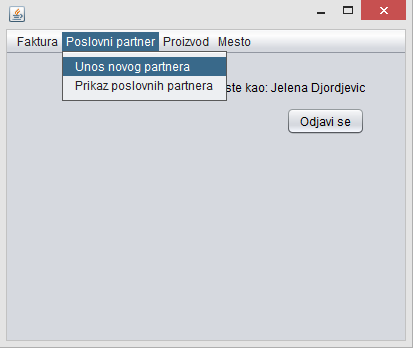
Korisnik i sistem (program)

##### **Preduslov**: Sistem je uključen i korisnik je prijavljen pod svojom šifrom. Sistem prikazuje početnu formu.

##### **Osnovni scenario SK**

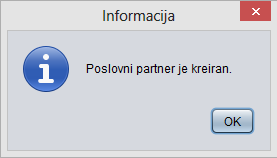
1. Korisnik **poziva** sistem da kreira novog poslovnog partnera. (APSO)

*Opis akcije*: Klikom na stavku menija Poslovni partner, *Unos novog partnera,* korisnik poziva sistemsku operaciju ***kreiraj(PoslovniPartner)*** koja pravi novog proizvod.

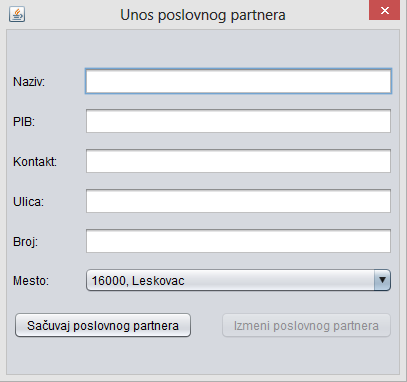


Slika 89 - Kreiranje novog partnera

1. Sistem **kreira** novog poslovnog partnera. (SO)
2. Sistem **prikazuje** korisniku novog poslovnog partnera i poruku: “Sistem je kreirao novog poslovnog partnera”. (IA)



Slika 90 – Poruka



Slika 91 - Kreiran poslovni partner

1. Korisnik **unosi** podatke o novom poslovnom partneru. (APUSO)



Slika 92 - Unos podataka

1. Korisnik **kontroliše** da li je korektno uneo podatke o poslovnom partneru. (ANSO)
2. Korisnik **poziva** sistem da zapamti podatke o poslovnom partneru. (APSO)

*Opis akcije*: Korisnik klikom na dugme „Sačuvaj“ poziva sistemsku operaciju ***Zapamti(PoslovniPartner)*** koja čuva novog poslovnog partnera.

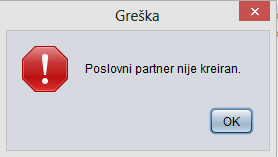
1. Sistem **pamti** podatke o poslovnom partneru. (SO)
2. Sistem **prikazuje** korisniku zapamćenog poslovnog partnera i poruku: “Sistem je sačuvao poslovnog partnera”. (IA)



Slika 93 - Partner je sačuvan

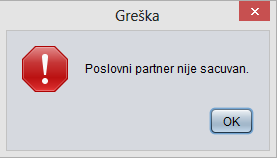
##### **Alternativna scenarija**

3.1 Ukoliko sistem ne može da kreira poslovnog partnera on prikazuje korisniku poruku: “Sistem ne može da kreira novog poslovnog partnera”. Prekida se izvršenje scenaria. (IA)



Slika 94 - Greška

8.1 Ukoliko sistem ne može da zapamti podatke o poslovnom partneru on prikazuje korisniku poruku “Sistem ne može da zapamti poslovnog partnera”. Prekida se izvršenje scenaria. (IA)



Slika 95 – Greška

#### Slučaj korišćenja 8: Izmena poslovnog partnera

##### **Naziv SK**

Izmena poslovnog partnera

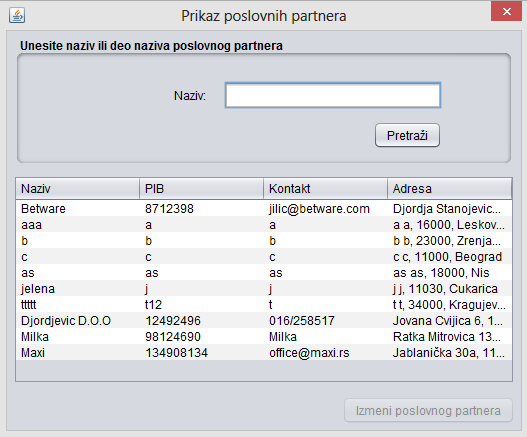
##### **Aktori SK**

Korisnik

##### **Učesnici SK**

Korisnik i sistem (program)

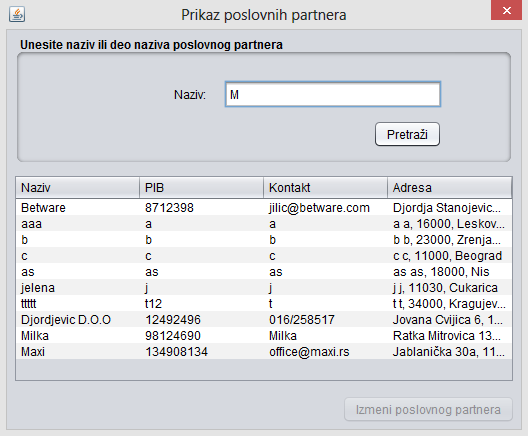
##### **Preduslov**: Sistem je uključen i korisnik je prijavljen pod svojom šifrom. Sistem prikazuje formu za pretragu poslovnih partnera.



Slika 96 - Forma za pretragu poslovnih partnera

##### **Osnovni scenario SK**

1. Korisnik **unosi** vrednost po kojoj pretražuje poslovne partnere. (APUSO)

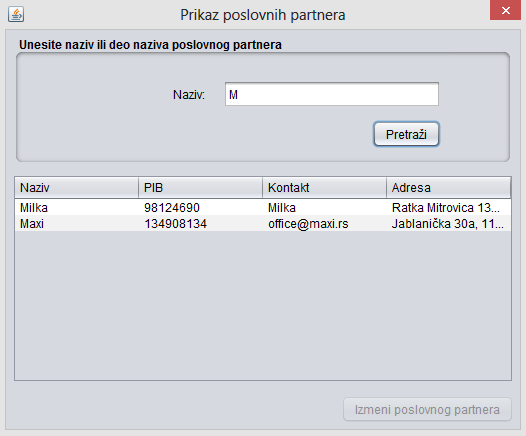


Slika 97 - Korisnik unosi kriterijum pretrage

1. Korisnik **poziva** sistem da nađe poslovne partnere po zadatoj vrednosti. (APSO)

Opis akcije: Korisnik klikom na dugme “Pretraži” poziva sistemsku operaciju Pretrazi (List<PoslovniPartner>, KriterijumPretrage) koja vraća listu poslovnih partnera.

1. Sistem **traži** poslovne partnere po zadatoj vrednosti. (SO)
2. Sistem **prikazuje** korisniku pronađene poslovne partnere. (IA)



Slika 98 - Pronađeni partneri

1. Korisnik **bira** poslovnog partnera. (APUSO)

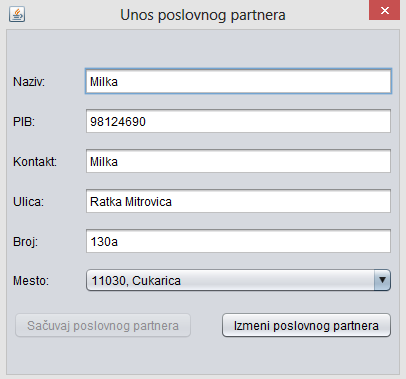


Slika 99 - Korisnik bira partnera

1. Korisnik **poziva** sistem da nađe sve podatke o poslovnom partneru. (APSO)

Opis akcije: Korisnik klikom na dugme “Izmeni poslovnog partnera” poziva sistemsku operaciju Nadji (PoslovniPartner, KriterijumPretrage) koja vraća podatke o izabranom poslovnom partneru.

1. Sistem **traži** podatke o izabranom poslovnom partneru. (SO)
2. Sistem **prikazuje** korisniku sve podatke o poslovnom partneru. (IA)



Slika 100 - Podaci o poslovnom partneru

1. Korisnik **unosi (menja)** podatke o poslovnom partneru. (APUSO)
2. Korisnik **kontroliše** da li je korektno uneo podatke o poslovnom partneru. (ANSO)
3. Korisnik **poziva** sistem da zapamti podatke o poslovnom partneru. (APSO)

*Opis akcije*: Korisnik klikom na dugme „Izmeni poslovnog partnera“ poziva sistemsku operaciju ***Zapamti(PoslovniPartner)*** koja čuva poslovnog partnera.

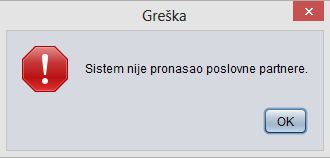
1. Sistem **pamti** podatke o poslovnom partneru. (SO)
2. Sistem **prikazuje** korisniku zapamćenog poslovnog partnera i poruku: “Sistem je zapamtio poslovnog partnera.” (IA)



Slika 101 - Poslovni partner je izmenjen

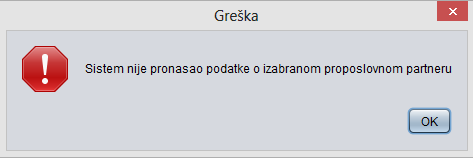
##### **Alternativna scenarija**

4.1 Ukoliko sistem ne može da nađe nijedanog poslovnog partnera on prikazuje korisniku poruku: “Sistem ne može da nađe poslovnog partnera po zadatoj vrednosti”. (IA)



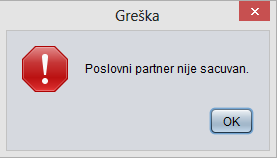
Slika 102 - Greška

8.1 Ukoliko sistem ne može da nađe podatke o poslovnom partneru on prikazuje korisniku poruku: “Sistem ne može da nađe poslovnog partnera po zadatoj vrednosti”. (IA)



Slika 103 - Greška

13.1 Ukoliko sistem ne može da zapamti podatke o izmenjenom poslovnom partneru on prikazuje korisniku poruku “Sistem ne može da zapamti izmenjenog poslovnog partnera ”. Prekida se izvršenje scenaria. (IA)



Slika 104 - Greška

#### Slučaj korišćenja 9: Prijavljivanje korisnika

##### **Naziv SK:**

Prijavljivanje korisnika

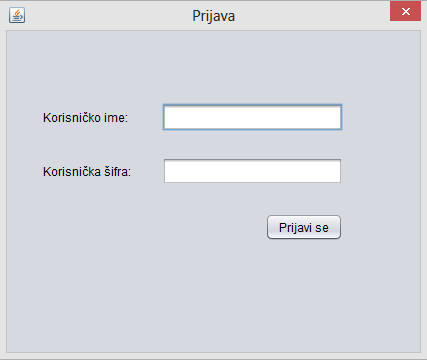
##### **Aktori SK:**

Korisnik

##### **Učesnici SK:**

Korisnik i sistem (program)

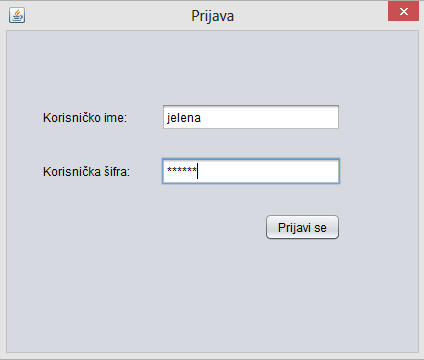
##### **Preduslov:** Sistem je uključen i prikazuje formu za prijavu.



Slika 105 - Forma za prijavu

##### **Osnovni scenario SK:**

1. Korisnik **unosi** podatke za autentifikaciju korisnika. (APUSO)

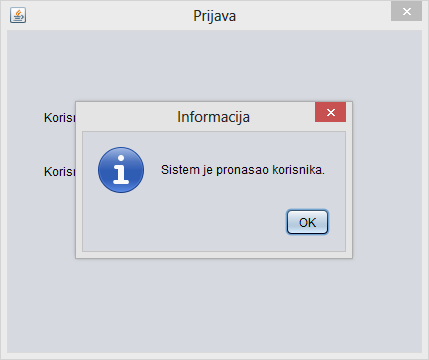


Slika 106 - Unos podataka za autentifikaciju

1. Korisnik **poziva** sistem da pronađe korisnika sa zadatim podacima. (APSO)

*Opis akcije*: Korisnik klikom na dugme „Prijavi se“ poziva sistemsku operaciju ***NadjiKorisnika(Korisnik)***.

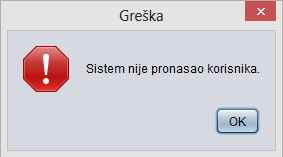
1. Sistem **pretražuje** korisnike. (SO)
2. Sistem **prikazuje** poruku da je korisnik uspešno prijavljen i početnu formu . (IA)



Slika 107 - Uspešna prijava

##### **Alternativna scenarija:**

4.1 Ukoliko sistem ne može da nađe korisnika prikazuje poruku da korisnik nije uneo tačne podatke. (IA)



Slika 108 - Greška

### Projektovanje kontrolera korisničkog interfejsa

Kontroler korisničkog interfejsa predstavlja sponu između ekranske forme i aplikacione logike.

#### KontrolerKIUnosFakture

public class KontrolerKIUnosFakture {

private static Faktura faktura;

public static void kreirajFakturu(FmUnosFakture f) throws Exception {

faktura = new Faktura();

int id = KontrolerKI.generisiID(faktura);

faktura.setIdFakture(id);

TransferObjekat toZahtev = new TransferObjekat();

toZahtev.setOperacija(konstante.Operacije.KREIRAJ\_FAKTURU);

toZahtev.setParametar(faktura);

Komunikacija.vratiObjekat().posalji(toZahtev);

TransferObjekat toOdgovor = Komunikacija.vratiObjekat().procitaj();

String poruka = null;

if (toOdgovor.getIzuzetak() != null) {

poruka = ((Exception) toOdgovor.getIzuzetak()).getMessage();

JOptionPane.showMessageDialog(f, poruka, "Greška", JOptionPane.ERROR\_MESSAGE);

} else {

poruka = (String) toOdgovor.getPoruka();

f.inicijalizuj();

JOptionPane.showMessageDialog(f, poruka, "Informacija", JOptionPane.INFORMATION\_MESSAGE);

}

}

static void zapamtiFakturu(FmUnosFakture aThis, JTextField txtBrojFakture, JTextField txtBrojOtpremnice,

JTextField txtDatum, JTextField txtOsnovica, JTextField txtPdv, JTextField txtUkupnaVrednost,

JTable tblStavkeFakture, JComboBox cbPoslovniPartner, JCheckBox checkObradjena, JCheckBox checkStornirana) {

Faktura f = null;

try {

String brojFakture = txtBrojFakture.getText().trim();

String brojOtpremnice = txtBrojOtpremnice.getText().trim();

String datumS = txtDatum.getText().trim();

if (brojFakture.isEmpty() || brojOtpremnice.isEmpty() || datumS.isEmpty()) {

throw new Exception("Sva polja su obavezna.");

}

Date d = null;

try {

d = new SimpleDateFormat("dd/MM/yyyy").parse(datumS);

} catch (Exception e) {

throw new Exception("Unesite datum u formatu dd/MM/yyyy");

}

Date datum = new java.sql.Date(d.getTime());

double osnovica = Double.parseDouble(txtOsnovica.getText().trim());

double pdv = Double.parseDouble(txtPdv.getText().trim());

double ukupnaVrednost = Double.parseDouble(txtUkupnaVrednost.getText().trim());

boolean obradjena = checkObradjena.isSelected();

boolean stornirana = checkStornirana.isSelected();

PoslovniPartner partner = (PoslovniPartner) cbPoslovniPartner.getSelectedItem();

List<StavkaFakture> stavke = ((ModelTabeleStavkaFakture) tblStavkeFakture.getModel()).vratiStavke();

if (stavke.isEmpty()) {

throw new Exception("Faktura mora imati bar jednu stavku.");

}

for (StavkaFakture stavkaFakture : stavke) {

if (stavkaFakture.getProizvod() == null) {

throw new Exception("Morate uneti proizvod za svaku stavku");

}

if (stavkaFakture.getKolicina() <= 0) {

throw new Exception("Količina mora biti broj veći od nule.");

}

if (stavkaFakture.getRabat() < 0 || stavkaFakture.getRabat() > 1) {

throw new Exception("Rabat mora biti broj između 0 i 1.");

}

if (stavkaFakture.getPdv()< 0 || stavkaFakture.getPdv() > 1) {

throw new Exception("PDV mora biti broj između 0 i 1.");

}

}

f = new Faktura(faktura.getIdFakture(), brojFakture, partner,

ukupnaVrednost, obradjena, stornirana, stavke, osnovica, pdv, brojOtpremnice, datum);

TransferObjekat toZahtev = new TransferObjekat();

toZahtev.setOperacija(Operacije.SACUVAJ\_FAKTURU);

toZahtev.setParametar(f);

Komunikacija.vratiObjekat().posalji(toZahtev);

TransferObjekat toOdogovor = Komunikacija.vratiObjekat().procitaj();

String poruka = null;

if (toOdogovor.getIzuzetak() != null) {

poruka = ((Exception) toOdogovor.getIzuzetak()).getMessage();

JOptionPane.showMessageDialog(aThis, poruka, "Greška", JOptionPane.ERROR\_MESSAGE);

} else {

poruka = (String) toOdogovor.getPoruka();

aThis.setFaktura(f);

aThis.setSacuvana();

JOptionPane.showMessageDialog(aThis, poruka, "Informacija", JOptionPane.INFORMATION\_MESSAGE);

}

} catch (Exception ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(aThis, ex.getMessage(), "Greška", JOptionPane.ERROR\_MESSAGE);

}

}

public static Faktura getFaktura() {

return faktura;

}

}

#### KontrolerKIUnosProizvoda

public class KontrolerKIUnosProizvoda {

private static Proizvod proizvod;

public static void kreirajProizvod(FmUnosProizvoda f) throws Exception {

proizvod = new Proizvod();

int proizvodId = KontrolerKI.generisiID(proizvod);

proizvod.setId(proizvodId);

TransferObjekat toZahtev = new TransferObjekat();

toZahtev.setOperacija(Operacije.KREIRAJ\_PROIZVOD);

toZahtev.setParametar(proizvod);

Komunikacija.vratiObjekat().posalji(toZahtev);

TransferObjekat toOdgovor = Komunikacija.vratiObjekat().procitaj();

String poruka = null;

if (toOdgovor.getIzuzetak() != null) {

poruka = ((Exception) toOdgovor.getIzuzetak()).getMessage();

} else {

poruka = (String) toOdgovor.getPoruka();

}

JOptionPane.showMessageDialog(f, poruka);

}

static void sacuvajProizvod(JTextField txtNaziv, JTextField txtCena,

JComboBox cbJedMere, FmUnosProizvoda f) {

try {

String naziv = txtNaziv.getText();

if (naziv.isEmpty() || txtNaziv.getText().isEmpty()) {

throw new Exception("Sva polja su obavezna.");

}

double cena = 0;

try {

cena = Double.parseDouble(txtCena.getText());

} catch (Exception e) {

throw new Exception("Cena mora biti broj.");

}

if (cena <= 0) {

throw new Exception("Cena mora biti broj veći od nule.");

}

JedMere jedMere = (JedMere) cbJedMere.getSelectedItem();

proizvod = new Proizvod(proizvod.getId(), naziv, cena, jedMere);

TransferObjekat toZahtev = new TransferObjekat();

toZahtev.setOperacija(Operacije.SACUVAJ\_PROIZVOD);

toZahtev.setParametar(proizvod);

Komunikacija.vratiObjekat().posalji(toZahtev);

TransferObjekat toOdogovor = Komunikacija.vratiObjekat().procitaj();

if (toOdogovor.getIzuzetak() != null) {

JOptionPane.showMessageDialog(f, toOdogovor.getPoruka(), "Greška", JOptionPane.ERROR\_MESSAGE);

} else {

JOptionPane.showMessageDialog(f, toOdogovor.getPoruka(), "Informacija", JOptionPane.INFORMATION\_MESSAGE);

}

} catch (Exception ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(f, ex.getMessage(), "Greška", JOptionPane.ERROR\_MESSAGE);

}

}

}

#### KontorlerKIUnosPoslovnogPartnera

public class KontorlerKIUnosPoslovnogPartnera {

private static PoslovniPartner poslovniPartner;

public static void zapamtiPoslovnogPartnera(JTextField txtNaziv, JTextField txtPib,

JTextField txtKontakt, JTextField txtUlica, JTextField txtBroj,

JComboBox cbMesto, FmUnosPoslovnogPartnera fmUnosMesta) {

PoslovniPartner partner = null;

try {

String naziv = txtNaziv.getText().trim();

String pib = txtPib.getText().trim();

String kontakt = txtKontakt.getText().trim();

String ulica = txtUlica.getText().trim();

String broj = txtBroj.getText().trim();

if (naziv.isEmpty() || pib.isEmpty() || kontakt.isEmpty() || ulica.isEmpty() ||

broj.isEmpty()) {

throw new Exception("Sva polja su obavezna.");

}

Mesto mesto = (Mesto) cbMesto.getSelectedItem();

Adresa adresa = new Adresa(ulica, broj, mesto);

partner = new PoslovniPartner(poslovniPartner.getId(), pib, naziv, kontakt, adresa);

TransferObjekat toZahtev = new TransferObjekat();

toZahtev.setOperacija(Operacije.SACUVAJ\_POSLOVNOG\_PARTNERA);

toZahtev.setParametar(partner);

Komunikacija.vratiObjekat().posalji(toZahtev);

TransferObjekat toOdogovor = Komunikacija.vratiObjekat().procitaj();

String poruka = null;

if (toOdogovor.getIzuzetak() != null) {

poruka = ((Exception) toOdogovor.getIzuzetak()).getMessage();

JOptionPane.showMessageDialog(fmUnosMesta, poruka, "Greška", JOptionPane.ERROR\_MESSAGE);

} else {

poruka = (String) toOdogovor.getPoruka();

JOptionPane.showMessageDialog(fmUnosMesta, poruka, "Informacija", JOptionPane.INFORMATION\_MESSAGE);

}

} catch (Exception ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(fmUnosMesta, ex.getMessage(), "Greška", JOptionPane.ERROR\_MESSAGE);

}

}

public static void kreirajPoslovnogPartnera(FmUnosPoslovnogPartnera f) throws ClassNotFoundException, IOException, Exception {

poslovniPartner = new PoslovniPartner();

int poslovniPartnerId = KontrolerKI.generisiID(poslovniPartner);

poslovniPartner.setId(poslovniPartnerId);

TransferObjekat toZahtev = new TransferObjekat();

toZahtev.setOperacija(konstante.Operacije.KREIRAJ\_POSLOVNOG\_PARTNERA);

toZahtev.setParametar(poslovniPartner);

Komunikacija.vratiObjekat().posalji(toZahtev);

TransferObjekat toOdgovor = Komunikacija.vratiObjekat().procitaj();

String poruka = null;

if (toOdgovor.getIzuzetak() != null) {

poruka = ((Exception) toOdgovor.getIzuzetak()).getMessage();

JOptionPane.showMessageDialog(f, poruka, "Greška", JOptionPane.ERROR\_MESSAGE);

} else {

poruka = (String) toOdgovor.getPoruka();

JOptionPane.showMessageDialog(f, poruka, "Informacija", JOptionPane.INFORMATION\_MESSAGE);

}

}

}

## Projektovanje aplikacione logike

### Kontroler aplikacione logike

public class Kontroler {

private static Kontroler k;

private OpstaSistemskaOperacija oso;

private Kontroler() {

}

public static Kontroler vratiObjekat() {

if (k == null) {

k = new Kontroler();

}

return k;

}

public void sacuvajObjekat(OpstiDomenskiObjekat odo) throws Exception{

oso = new SacuvajObjekat(odo);

oso.izvrsenjeSO();

}

public List<OpstiDomenskiObjekat> vratiSveObjekte(OpstiDomenskiObjekat odo) throws Exception {

oso = new VratiSveObjekte(odo);

oso.izvrsenjeSO();

return ((VratiSveObjekte)oso).getLista();

}

public List<OpstiDomenskiObjekat> vratiObjekteSaVezanimObjektom(OpstiDomenskiObjekat odo)throws Exception{

oso = new VratiObjekteSaVezanimObjektom(odo);

oso.izvrsenjeSO();

List<OpstiDomenskiObjekat> lista = ((VratiObjekteSaVezanimObjektom)oso).getLista();

return lista;

}

public OpstiDomenskiObjekat kreirajObjekat(OpstiDomenskiObjekat odo) throws Exception {

oso = new KreirajNoviObjekat(odo);

oso.izvrsenjeSO();

return ((KreirajNoviObjekat)oso).getObjekat();

}

public int generisiID(OpstiDomenskiObjekat odo) throws Exception {

oso = new GenerisiId(odo);

oso.izvrsenjeSO();

return ((GenerisiId)oso).getId();

}

public void kreirajISacuvajObjekat(OpstiDomenskiObjekat odo) throws Exception {

oso = new KreirajIsacuvajObjekat(odo);

oso.izvrsenjeSO();

}

public List<OpstiDomenskiObjekat> pretraziObjekte(OpstiDomenskiObjekat odo) throws Exception {

oso = new PretraziObjekte(odo);

oso.izvrsenjeSO();

return ((PretraziObjekte)oso).getLista();

}

public List<OpstiDomenskiObjekat> pretraziSaVezanimObjektom(OpstiDomenskiObjekat odo) throws Exception {

oso = new PretraziObjekteSaVezanimObjektom(odo);

oso.izvrsenjeSO();

return ((PretraziObjekteSaVezanimObjektom)oso).getLista();

}

public OpstiDomenskiObjekat prikaziPodatke(OpstiDomenskiObjekat odo) throws Exception {

oso = new PrikaziPodatke(odo);

oso.izvrsenjeSO();

return ((PrikaziPodatke)oso).getObjekat();

}

public OpstiDomenskiObjekat prikaziPodatkeSaVezanimObjektom(OpstiDomenskiObjekat odo) throws Exception {

oso = new PrikaziPodatkeSaVezanimObjektom(odo);

oso.izvrsenjeSO();

return ((PrikaziPodatkeSaVezanimObjektom)oso).getObjekat();

}

public void sacuvajObjekatSaStavkama(OpstiDomenskiObjekat odo) throws Exception{

oso = new SacuvajFakturuSaStavkama(odo);

oso.izvrsenjeSO();

}

public OpstiDomenskiObjekat prikaziPodatkeSaVezanimObjektomIStavkama(OpstiDomenskiObjekat odo) throws Exception{

oso = new PrikaziPodatkeSaVezanimObjektomIStavkama(odo);

oso.izvrsenjeSO();

return ((PrikaziPodatkeSaVezanimObjektomIStavkama)oso).getObjekat();

}

public void obradi(OpstiDomenskiObjekat odo) throws Exception {

oso = new ObradiFakturu(odo);

oso.izvrsenjeSO();

}

public void storniraj(OpstiDomenskiObjekat odo) throws Exception {

oso = new StornirajFakturu(odo);

oso.izvrsenjeSO();

}

}

### Poslovna logika

Za svaki od ugovora o sistemskim operacijama koje smo prethodno definisali, projektuje se konceptualno rešenje.

#### Opšta sistemska operacija

public abstract class OpstaSistemskaOperacija {

protected DatabaseBroker dbbr;

protected OpstiDomenskiObjekat odo;

public OpstaSistemskaOperacija(OpstiDomenskiObjekat odo) {

dbbr = new DatabaseBroker();

this.odo = odo;

}

public void izvrsenjeSO() throws Exception{

otvoriBazu();

try{

izvrsiOperaciju();

potvrdi();

}catch(Exception e){

odbaci();

throw e;

}

zatvoriBazu();

}

private void otvoriBazu() throws Exception {

dbbr.ucitajDriver();

dbbr.otvoriKonekciju();

}

public abstract void izvrsiOperaciju() throws Exception;

private void potvrdi() throws Exception {

dbbr.commitTransakcije();

}

private void odbaci() throws Exception {

dbbr.rollbackTransakcije();

}

private void zatvoriBazu() throws Exception {

dbbr.zatvoriKonekciju();

}

}

#### KreirajNoviObjekat

public class KreirajNoviObjekat extends OpstaSistemskaOperacija{

OpstiDomenskiObjekat objekat;

public KreirajNoviObjekat(OpstiDomenskiObjekat odo) {

super(odo);

}

@Override

public void izvrsiOperaciju() throws Exception {

dbbr.kreiraj(odo);

}

public OpstiDomenskiObjekat getObjekat() {

return objekat;

}

}

#### SacuvajObjekat

public class SacuvajObjekat extends OpstaSistemskaOperacija{

public SacuvajObjekat(OpstiDomenskiObjekat odo) {

super(odo);

}

@Override

public void izvrsiOperaciju() throws Exception {

if (!odo.vrednosnaOgranicenja()) {

throw new Exception("Nisu zadovoljena vrednosna ograničenja!");

}

dbbr.sacuvaj(odo);

}

}

#### PretraziObjekte

public class PretraziObjekte extends OpstaSistemskaOperacija{

List<OpstiDomenskiObjekat> lista;

public PretraziObjekte(OpstiDomenskiObjekat odo) {

super(odo);

}

@Override

public void izvrsiOperaciju() throws Exception {

lista = dbbr.pretraziObjekte(odo);

}

public List<OpstiDomenskiObjekat> getLista() {

return lista;

}

}

#### PrikaziPodatke

public class PrikaziPodatke extends OpstaSistemskaOperacija{

OpstiDomenskiObjekat objekat;

public PrikaziPodatke(OpstiDomenskiObjekat odo) {

super(odo);

}

@Override

public void izvrsiOperaciju() throws Exception {

objekat = dbbr.dajPodatke(odo);

}

public OpstiDomenskiObjekat getObjekat() {

return objekat;

}

}

#### SacuvajFakturuSaStavkama

public class SacuvajFakturuSaStavkama extends OpstaSistemskaOperacija {

public SacuvajFakturuSaStavkama(OpstiDomenskiObjekat odo) {

super(odo);

}

@Override

public void izvrsiOperaciju() throws Exception {

if (!preduslov()) {

throw new Exception("Nisu ispiunjeni preduslovi!");

}

if (!odo.vrednosnaOgranicenja()) {

throw new Exception("Nisu zadovoljena vrednosna ograničenja!");

}

dbbr.brisiStavke(odo);

for (int i = 0; i < odo.vratiBrojStavki(); i++) {

dbbr.brisiSlabObjekat(odo.vratiStavku(i), odo);

}

dbbr.sacuvaj(odo);

for (int i = 0; i < odo.vratiBrojStavki(); i++) {

dbbr.sacuvajSlabObjekat(odo.vratiStavku(i), odo);

}

}

private boolean preduslov() throws Exception {

return !(dbbr.vratiLogickuVrednostAtributa("obradjena", odo)

|| dbbr.vratiLogickuVrednostAtributa("stornirana", odo));

}

}

#### PrikaziPodatkeSaVezanimObjektomIStavkama

public class PrikaziPodatkeSaVezanimObjektomIStavkama extends OpstaSistemskaOperacija{

OpstiDomenskiObjekat objekat;

public PrikaziPodatkeSaVezanimObjektomIStavkama(OpstiDomenskiObjekat odo) {

super(odo);

}

@Override

public void izvrsiOperaciju() throws Exception {

objekat = dbbr.dajPodatke(odo);

objekat.setVezaniObjekat(dbbr.dajPodatke(odo.vratiVezaniObjekat()));

List<OpstiDomenskiObjekat> lista = dbbr.vratiSlabeObjekte(odo);

for (OpstiDomenskiObjekat opstiDomenskiObjekat : lista) {

opstiDomenskiObjekat.setVezaniObjekat(dbbr.dajPodatke(opstiDomenskiObjekat.vratiVezaniObjekat()));

}

objekat.setujStavke(lista);

}

public OpstiDomenskiObjekat getObjekat() {

return objekat;

}

}

#### ObradiFakturu

public class ObradiFakturu extends OpstaSistemskaOperacija {

public ObradiFakturu(OpstiDomenskiObjekat odo) {

super(odo);

}

@Override

public void izvrsiOperaciju() throws Exception {

if (!preduslov()) {

throw new Exception("Nisu ispiunjeni preduslovi!");

}

if (!odo.vrednosnaOgranicenja()) {

throw new Exception("Nisu zadovoljena vrednosna ograničenja!");

}

dbbr.brisiStavke(odo);

for (int i = 0; i < odo.vratiBrojStavki(); i++) {

dbbr.brisiSlabObjekat(odo.vratiStavku(i), odo);

}

dbbr.sacuvaj(odo);

for (int i = 0; i < odo.vratiBrojStavki(); i++) {

dbbr.sacuvajSlabObjekat(odo.vratiStavku(i), odo);

}

}

private boolean preduslov() throws Exception {

return !(dbbr.vratiLogickuVrednostAtributa("obradjena", odo)

|| dbbr.vratiLogickuVrednostAtributa("stornirana", odo));

}

}

#### StornirajFakturu

public class StornirajFakturu extends OpstaSistemskaOperacija {

public StornirajFakturu(OpstiDomenskiObjekat odo) {

super(odo);

}

@Override

public void izvrsiOperaciju() throws Exception {

if (!preduslov()) {

throw new Exception("Nisu ispiunjeni preduslovi!");

}

if (!odo.vrednosnaOgranicenja()) {

throw new Exception("Nisu zadovoljena vrednosna ograničenja!");

}

dbbr.brisiStavke(odo);

for (int i = 0; i < odo.vratiBrojStavki(); i++) {

dbbr.brisiSlabObjekat(odo.vratiStavku(i), odo);

}

dbbr.sacuvaj(odo);

for (int i = 0; i < odo.vratiBrojStavki(); i++) {

dbbr.sacuvajSlabObjekat(odo.vratiStavku(i), odo);

}

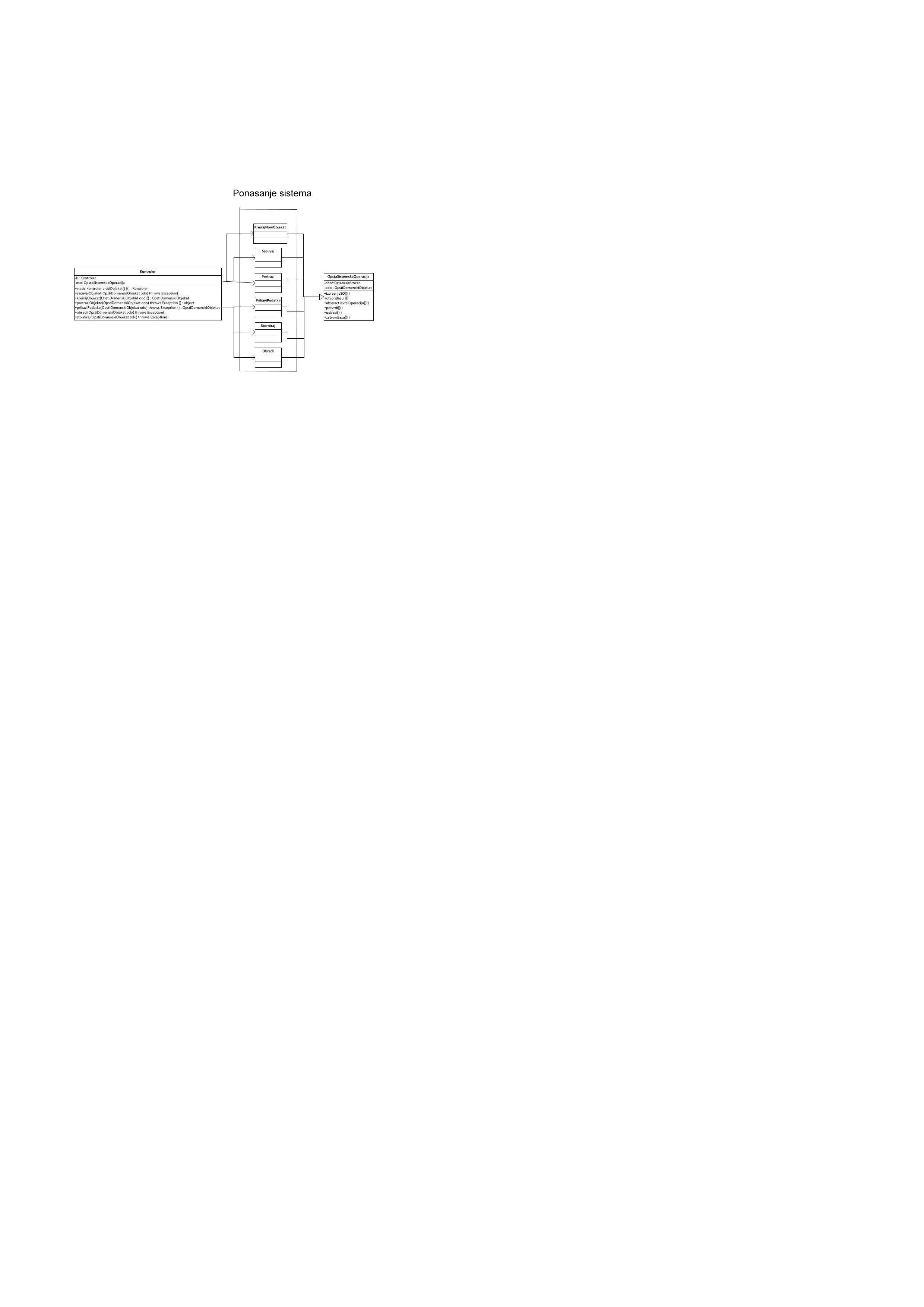
}

private boolean preduslov() throws Exception {

return !(dbbr.vratiLogickuVrednostAtributa("stornirana", odo));

}

}



Slika 109 - Klase koje su odgovorne za SO nasledjuju SistemskuOperaciju

#### Projektovanje strukture softverskog sistema

Na osnuvu konceptualnih klasa (slika 34), prave se softverske klase strukture:

##### Faktura:

public class Faktura {

private int idFakture;

private String brojFakture;

private PoslovniPartner poslovniPartner;

private double ukupnaVrednost;

private boolean obradjena;

private boolean stornirana;

private List<StavkaFakture> stavke;

private double osnovica;

private double pdv;

private String brojOtpremnice;

private Date datum;

public Faktura(int idFakture, String brojFakture, PoslovniPartner poslovniPartnerId, double ukupnaVrednost, boolean obradjena, boolean stornirana, List<StavkaFakture> stavke, double osnovica, double pdv, String brojOtpremnice, Date datum) {

this.idFakture = idFakture;

this.brojFakture = brojFakture;

this.poslovniPartner = poslovniPartnerId;

this.ukupnaVrednost = ukupnaVrednost;

this.obradjena = obradjena;

this.stornirana = stornirana;

this.stavke = stavke;

this.osnovica = osnovica;

this.pdv = pdv;

this.brojOtpremnice = brojOtpremnice;

this.datum = datum;

}

}

##### StavkaFakture

public class StavkaFakture {

private int rb;

private double kolicina;

private double rabat;

private double iznosRabata;

private double poreskaOsnovica;

private double pdv;

private double iznosPdva;

private double vrednostSaPdvom;

private Proizvod proizvod;

public StavkaFakture(int rb, double kolicina, double rabat, double iznosRabata, double poreskaOsnovica, double pdv, double iznosPdva, double vrednostSaPdvom, Proizvod proizvod) {

this.rb = rb;

this.kolicina = kolicina;

this.rabat = rabat;

this.iznosRabata = iznosRabata;

this.poreskaOsnovica = poreskaOsnovica;

this.pdv = pdv;

this.iznosPdva = iznosPdva;

this.vrednostSaPdvom = vrednostSaPdvom;

this.proizvod = proizvod;

}

}

##### Proizvod

public class Proizvod {

private int id;

private String naziv;

private double cena;

private JedMere jedinicaMere;

public Proizvod(int id, String naziv, double cena, JedMere jedinicaMere) {

this.id = id;

this.naziv = naziv;

this.cena = cena;

this.jedinicaMere = jedinicaMere;

}

}

##### PoslovniPartner

public class PoslovniPartner {

private int id;

private String pib;

private String naziv;

private String kontakt;

private Adresa adresa;

public PoslovniPartner(int id, String pib, String naziv, String kontakt, Adresa adresa) {

this.id = id;

this.pib = pib;

this.naziv = naziv;

this.kontakt = kontakt;

this.adresa = adresa;

}

}

##### Mesto

public class Mesto{

private int ptt;

private String naziv;

public Mesto(int ptt, String naziv) {

this.ptt = ptt;

this.naziv = naziv;

}

}

##### Adresa

public class Adresa {

private String ulica;

private String broj;

private Mesto mesto;

public Adresa(String ulica, String broj, Mesto mesto) {

this.ulica = ulica;

this.broj = broj;

this.mesto = mesto;

}

}

##### JedMere

public enum JedMere {

kom, m, l, kg, m2, m3;

}

### Broker baze podataka

public class DatabaseBroker {

private Connection konekcija;

public void ucitajDriver() throws Exception {

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

}

public void otvoriKonekciju() throws Exception {

konekcija = DriverManager.getConnection("jdbc:odbc:bazaProSoftSem");

konekcija.setAutoCommit(false); // Zahteva eksplicitnu potvrdu transakcije

}

public void zatvoriKonekciju() throws Exception {

konekcija.close();

}

public void commitTransakcije() throws Exception {

try {

konekcija.commit();

} catch (SQLException ex) {

ex.printStackTrace();

throw new Exception("Transakcija nije potvrđena!");

}

}

public void rollbackTransakcije() throws Exception {

try {

konekcija.rollback();

} catch (SQLException ex) {

ex.printStackTrace();

throw new Exception("Transakcija nije potvrđena!");

}

}

public void sacuvaj(OpstiDomenskiObjekat odo) throws Exception {

try {

String sql = "UPDATE " + odo.vratiNazivTabele()

+ " SET " + odo.vratiAtributeSaVrednostima() + " WHERE "

+ odo.vratiNazivKljuca() + " = " + odo.vratiVrednostKljuca();

System.out.println(sql);

Statement sqlNaredba = konekcija.createStatement();

sqlNaredba.executeUpdate(sql);

sqlNaredba.close();

} catch (SQLException ex) {

ex.printStackTrace();

throw new Exception("Greška pri čuvanju!");

}

}

public List<OpstiDomenskiObjekat> vratiSve(OpstiDomenskiObjekat odo) throws Exception {

try {

String sql = "SELECT \* FROM " + odo.vratiNazivTabele();

System.out.println(sql);

Statement sqlNaredba = konekcija.createStatement();

ResultSet rs = sqlNaredba.executeQuery(sql);

List<OpstiDomenskiObjekat> lista = odo.napuni(rs);

return lista;

} catch (SQLException ex) {

ex.printStackTrace();

throw new Exception("Greška pri učitavanju!");

}

}

public OpstiDomenskiObjekat kreiraj(OpstiDomenskiObjekat odo) throws Exception {

try {

String upit = "INSERT INTO " + odo.vratiNazivTabele() + "("

+ odo.vratiNazivKljuca() + ")" + " VALUES (" + odo.vratiVrednostKljuca() + ")";

System.out.println(upit);

Statement sqlNaredba = konekcija.createStatement();

sqlNaredba.executeUpdate(upit);

sqlNaredba.close();

return odo;

} catch (SQLException ex) {

ex.printStackTrace();

throw new Exception("Grepka pri kreiranju objekta!");

}

}

public int generisiId(OpstiDomenskiObjekat odo) throws Exception {

try {

int broj = 0;

String upit = "SELECT LAST(" + odo.vratiNazivKljuca() + ") AS poslednjiId FROM " + odo.vratiNazivTabele();

System.out.println(upit);

Statement sqlNaredba = konekcija.createStatement();

ResultSet rs = sqlNaredba.executeQuery(upit);

if (rs.next()) {

broj = rs.getInt("poslednjiId") + 1;

}

return broj;

} catch (SQLException ex) {

ex.printStackTrace();

throw new Exception("Greška pri generisanju ID-a!");

}

}

public void kreirajISacuvaj(OpstiDomenskiObjekat odo) throws Exception {

try {

String upit = "INSERT INTO " + odo.vratiNazivTabele() + " VALUES (" + odo.vratiVrednostiAtributa() + ")";

System.out.println(upit);

Statement sqlNaredba = konekcija.createStatement();

sqlNaredba.executeUpdate(upit);

sqlNaredba.close();

} catch (SQLException ex) {

ex.printStackTrace();

throw new Exception("Greška pri kreiranju i čuvanju objekta!");

}

}

public List<OpstiDomenskiObjekat> pretraziObjekte(OpstiDomenskiObjekat odo) throws Exception {

try {

String sql = "SELECT \* FROM " + odo.vratiNazivTabele() + " WHERE " + odo.vratiKriterijumPretrage();

System.out.println(sql);

Statement sqlNaredba = konekcija.createStatement();

ResultSet rs = sqlNaredba.executeQuery(sql);

return odo.napuni(rs);

} catch (SQLException ex) {

ex.printStackTrace();

throw new Exception("Greška pri pretraživanju!");

}

}

public OpstiDomenskiObjekat dajPodatke(OpstiDomenskiObjekat odo) throws Exception {

try {

String upit = "SELECT \* FROM " + odo.vratiNazivTabele() + " WHERE " + odo.vratiNazivKljuca()

+ "=" + odo.vratiVrednostKljuca();

System.out.println(upit);

Statement sqlNaredba = konekcija.createStatement();

ResultSet rs = sqlNaredba.executeQuery(upit);

return odo.napuni(rs).get(0);

} catch (Exception ex) {

ex.printStackTrace();

throw new Exception("Greška pri izvlačenju podataka!");

}

}

public void brisiStavke(OpstiDomenskiObjekat odo) throws Exception {

try {

String upit;

OpstiDomenskiObjekat odo2;

for (int i = 0; i < odo.vratiBrojStavki(); i++) {

odo2 = odo.vratiStavku(i);

upit = "DELETE FROM " + odo2.vratiNazivTabele() + " WHERE " + odo2.vratiNazivKljuca()

+ "=" + odo.vratiVrednostKljuca() + " AND " + odo.vratiNazivKljuca() + "="

+ odo.vratiVrednostKljuca();

System.out.println(upit);

Statement sqlNaredba = konekcija.createStatement();

sqlNaredba.executeUpdate(upit);

}

} catch (Exception ex) {

ex.printStackTrace();

throw new Exception("Greška pri brisanju!");

}

}

public void brisiSlabObjekat(OpstiDomenskiObjekat slab, OpstiDomenskiObjekat jak) throws Exception {

try {

String upit = "DELETE FROM " + slab.vratiNazivTabele() + " WHERE " + slab.vratiNazivKljuca()

+ "=" + slab.vratiVrednostKljuca() + " AND " + jak.vratiNazivKljuca() + "="

+ jak.vratiVrednostKljuca();

System.out.println(upit);

Statement sqlNaredba = konekcija.createStatement();

sqlNaredba.executeUpdate(upit);

} catch (Exception ex) {

ex.printStackTrace();

throw new Exception("Greška pri brisanju!");

}

}

public void sacuvajSlabObjekat(OpstiDomenskiObjekat slab, OpstiDomenskiObjekat jak) throws Exception {

try {

String upit = "INSERT INTO " + slab.vratiNazivTabele() + " VALUES ("

+ slab.vratiVrednostiAtributa() + ", " + jak.vratiVrednostKljuca() + ")";

System.out.println(upit);

Statement sqlNaredba = konekcija.createStatement();

sqlNaredba.executeUpdate(upit);

sqlNaredba.close();

} catch (SQLException ex) {

ex.printStackTrace();

throw new Exception("Greška pri kreiranju i čuvanju objekta!");

}

}

public List<OpstiDomenskiObjekat> vratiSlabeObjekte(OpstiDomenskiObjekat odo) throws Exception {

try {

String upit = "SELECT \* FROM " + odo.vratiTabeluStavke() + " WHERE " + odo.vratiNazivKljuca()

+ "=" + odo.vratiVrednostKljuca();

System.out.println(upit);

Statement sqlNaredba = konekcija.createStatement();

ResultSet rs = sqlNaredba.executeQuery(upit);

return odo.vratiJednuStavku().napuni(rs);

} catch (Exception ex) {

ex.printStackTrace();

throw new Exception("Greška pri izvlačenju podataka!");

}

}

public boolean vratiLogickuVrednostAtributa(String atribut, OpstiDomenskiObjekat odo) throws Exception {

try {

String upit = " SELECT \*"

+ " FROM " + odo.vratiNazivTabele()

+ " WHERE " + odo.vratiNazivKljuca() + "=" + odo.vratiVrednostKljuca();

System.out.println(upit);

boolean s = false;

Statement sqlNaredba = konekcija.createStatement();

ResultSet rs = sqlNaredba.executeQuery(upit);

rs.next();

s = rs.getBoolean(atribut);

return s;

} catch (Exception ex) {

ex.printStackTrace();

throw new Exception("Greška pri izvlačenju podataka!");

}

}

}

U procesu pravljenja generičkih metoda DatabaseBroker klase dobili smo metode interfejsa OpstiDomenskiObjekat:

public interface OpstiDomenskiObjekat {

public String vratiNazivTabele();

public String vratiAtributeSaVrednostima();

public List<OpstiDomenskiObjekat> napuni(ResultSet rs) throws Exception;

public String vratiNazivKljuca();

public Object vratiVrednostKljuca();

public String vratiVrednostiAtributa();

public String vratiKriterijumPretrage()

public int vratiBrojStavki();

public OpstiDomenskiObjekat vratiStavku(int rb);

public OpstiDomenskiObjekat vratiVezaniObjekat();

public void setVezaniObjekat(OpstiDomenskiObjekat odo)

public String vratiTabeluStavke();

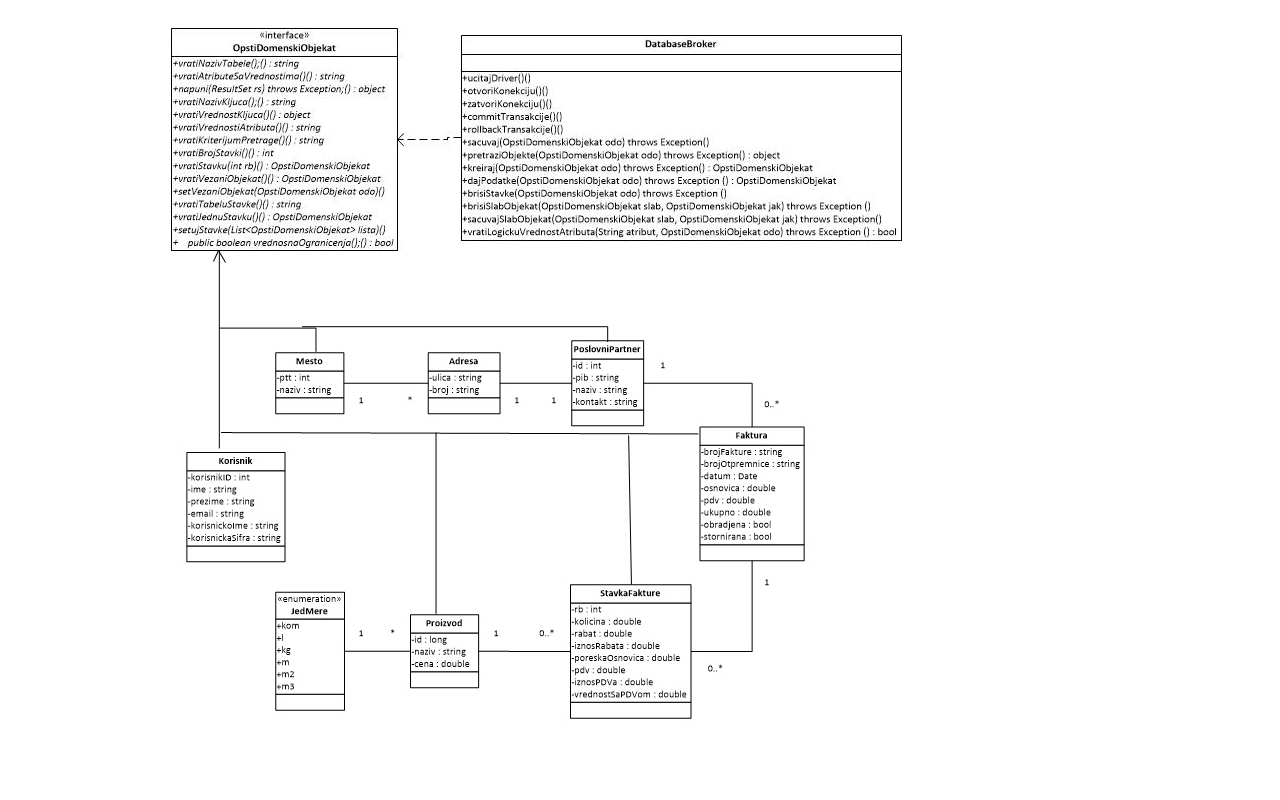
public OpstiDomenskiObjekat vratiJednuStavku();

public void setujStavke(List<OpstiDomenskiObjekat> lista);

public boolean vrednosnaOgranicenja();

}

Kao rezultat projektovanja klase DatabaseBrokera i interfejsa OpstiDomenskiObjekat dobijaju se sleći dijagrami:



Slika 110 - DatabaseBroker se povezuje sa klasom OpstiDomenskiObjekat

## Projektovanje skladišta podataka

Na osnovu softverskih klasa strukture projektovali smo tabele relacionog sistema za upravljanje bazom podataka.

#### Table: Faktura

**Columns**

Name Type Size

idFakture Long Integer 4

brojFakture Text 15

brojOtpremnice Text 15

datum Date/Time 8

osnovica Double 8

ukupnoPdv Double 8

ukupnaVrednost Double 8

obradjena Yes/No 1

stornirana Yes/No 1

poslovniPartnerId Long Integer 4

**Table Indexes**

Name Number of Fields

idFakture 1

Fields:

idFakture Ascending

PoslovniPartnerFaktura 1

Fields:

poslovniPartnerId Ascending

poslovniPartnerId 1

Fields:

poslovniPartnerId Ascending

PrimaryKey 1

Fields:

idFakture Ascending

#### Table: StavkaFakture

**Columns**

Name Type Size

rb Long Integer 4

kolicina Double 8

rabat Double 8

iznosRabata Double 8

poreskaOsnovica Double 8

pdv Double 8

iznosPdva Double 8

vrednostSaPdvom Double 8

proizvodId Long Integer 4

idFakture Long Integer 4

**Table Indexes**

Name Number of Fields

FakturaStavkaFakture 1

Fields:

idFakture Ascending

PrimaryKey 2

Fields:

rb Ascending

idFakture Ascending

proizvodId 1

Fields:

proizvodId Ascending

ProizvodStavkaFakture 1

Fields:

proizvodId Ascending

#### Table: PoslovniPartner

**Columns**

Name Type Size

poslovniPartnerId Long Integer 4

pib Text 9

naziv Text 50

kontakt Text 50

ulica Text 50

broj Text 10

ptt Long Integer 4

**Table Indexes**

Name Number of Fields

MestoPoslovniPartner 1

Fields:

ptt Ascending

PrimaryKey 1

Fields:

poslovniPartnerId Ascending

#### Table: Proizvod

**Columns**

Name Type Size

proizvodId Long Integer 4

naziv Text 50

cena Double 8

jedinicaMere Text 20

**Table Indexes**

Name Number of Fields

PrimaryKey 1

Fields:

proizvodId Ascending

#### Table: Mesto

**Columns**

Name Type Size

ptt Long Integer 4

naziv Text 50

**Table Indexes**

Name Number of Fields

PrimaryKey 1

Fields:

ptt Ascending

#### Table: Korisnik

**Columns**

Name Type Size

ID Long Integer 4

ime Text 30

prezime Text 30

korisnickoIme Text 15

korisnickaSifra Text 15

email Text 50

**Table Indexes**

Name Number of Fields

PrimaryKey 1

Fields:

ID Ascending