**Univerzitet u Beogradu**

**Fakultet organizacionih nauka**

**Laboratorija za softversko inženjerstvo**

**Seminarski rad iz predmeta**

**Projektovanje softvera**

**Tema: Softverski sistem za praćenje rada kafića**

Profesor: Dr Siniša Vlajić Student: Elez Stefan

435/13

# Sadržaj:

[Sadržaj: 2](#_Toc492163212)

[1. Korisnički zahtevi 4](#_Toc492163213)

[1.1 Verbalni opis 4](#_Toc492163214)

[1.2 Slučajevi korišćenja 4](#_Toc492163215)

[**СК1: Случај коришћења - Унос новог конобара** 5](#_Toc492163216)

[**СК2: Случај коришћења - Измена конобара** 6](#_Toc492163217)

[**СК3: Случај коришћења - Брисанје података о конобару** 6](#_Toc492163218)

[**СК4: Случај коришћенја – Претрага конобара** 7](#_Toc492163219)

[**СК5: Случај коришћења – Унос података о пићу** 8](#_Toc492163220)

[**СК6: Случај коришћења - Измена података о пићу** 8](#_Toc492163221)

[**СК7: Случај коришћенја – Претрага пића** 9](#_Toc492163222)

[**СК8 Случај коришћења - Unos novog stola** 10](#_Toc492163223)

[**СК9 Случај коришћења - Унос задужења за сто** 10](#_Toc492163224)

[**СК10: Случај коришћења - Пресек стања** 11](#_Toc492163225)

[2. Analiza sistema 12](#_Toc492163226)

[2.1 Sistemski dijagram sekvenci 12](#_Toc492163227)

[DS1: Diagram sekvenci za slučaj korišćenja **unos novog konobara** 12](#_Toc492163228)

[DS2: Diagram sekvenci za slučaj korišćenja **izmena konobara** 13](#_Toc492163229)

[DS3: Dijagram sekvenci za slučaj korišćenja **brisanje konobara** 15](#_Toc492163230)

[DS4: Dijagram sekvenci za slučaj korišćenja **Preraga konobara** 17](#_Toc492163231)

[DS5:Dijagram sekvenci za slučaj korišćenja **Unos podataka o pića** 18](#_Toc492163232)

[DS6: Dijagram sekvenci za slučaj korišćenja **Izmena pića** 19](#_Toc492163233)

[DS7: Dijagram sekvenci za slučaj korišćenja **pretraga pića** 21](#_Toc492163234)

[DS8: Dijagram sekvenci za slučaj koršćenja **unos novog stola** 22](#_Toc492163235)

[DS9: Dijagram sekvenci za slučaj koršćenja **Unos zaduženja za sto** 23](#_Toc492163236)

[DS10: Dijagram sekvenci za slučaj koršćenja **Presek stanja** 26](#_Toc492163237)

[2.2 Definisanje ugovora o sistemskim operacijama 28](#_Toc492163238)

[*2.2.1* *Ugovor UG1: sacuvajKonobara* 28](#_Toc492163239)

[*2.2.2* *Ugovor UG2: pronadjiKonobara* 28](#_Toc492163240)

[*2.2.3* *Ugovor UG3: obirsiKonobara* 28](#_Toc492163241)

[*2.2.4* *Ugovor UG4: kreirajPice* 28](#_Toc492163242)

[*2.2.5* *Ugovor UG5: pronadjiPice* 29](#_Toc492163243)

[*2.2.6* *Ugovor UG6: zapamtiSto* 29](#_Toc492163244)

[*2.2.7* *Ugovor UG7: zapamtiRačun* 29](#_Toc492163245)

[*2.2.8* *Ugovor UG8: pretražiRačun* 29](#_Toc492163246)

[*2.2.9* *Ugovor UG9: obrisiRacun* 29](#_Toc492163247)

[*2.2.10* *Ugovor UG10: obrisiStavkuRacuna* 29](#_Toc492163248)

[2.3 Konceptualni dijagram klasa 30](#_Toc492163249)

[2.4 Relaconi model 30](#_Toc492163250)

[3. Projektovanje 33](#_Toc492163251)

[3.1 Arhitektura softverskog sistema 33](#_Toc492163252)

[3.2 Projektovanje korisničkog interfejsa 33](#_Toc492163253)

[3.2.1 Projektovanje ekranskih formi 33](#_Toc492163254)

[3.2.2. Projektovanje kontrolera korisničkog interfejsa 55](#_Toc492163255)

[3.3 Projektovanje aplikacione logike 55](#_Toc492163256)

[3.3.1 Komunikacija sa klijentima 55](#_Toc492163257)

[3.3.2 Kontroler aplikacione logike 56](#_Toc492163258)

[3.3.3 Projektovanje strukture softverskog sistema 59](#_Toc492163259)

[3.3.4 Projektovanje sistemskih operacija 60](#_Toc492163260)

[3.3.5 Broker baze podataka 65](#_Toc492163261)

[3.4 Projektovanje skladištom podataka 67](#_Toc492163262)

[4 Implementacija 69](#_Toc492163263)

[5 Testiranje 71](#_Toc492163264)

# Korisnički zahtevi

## Verbalni opis

Potrebno je napraviti aplikaciju koja omogućava vođenje evidencije o svim konobarima, o tome koji konobar je raspoređen za kojim stolom i o pićima koja lokal nudi mušterijama.

Aplikacija treba da vodi evidenciju o konobarima, da omogući pretragu prema određenim vrednostima, da omogući izmene, dodavanje i brisanje podataka o njima.

Aplikacija takođe, treba da omogući vođenje evidencije o stolovima za kojima sede mušterije, i o računima napravljenim za svakim stolom u toku dana, kao i o konobaru koji je zadužen za dati sto.

Pored toga, aplikacija treba da vodi računa o pićima lokala, odnosno da omogući nabavke nove vrste pića, kao i omogući izmenu svakog pica.

## Slučajevi korišćenja

Definisani su sledeći slučajevi koršćenja:

1. Unos novog konobara
2. Izmena podataka o konobaru
3. Brisanje podataka o konobaru
4. Pretraga konobara
5. Unos podataka o piću
6. Izmena podataka o piću
7. Pretraga pica
8. Unos novog stola
9. Unos zaduženja za sto
10. Presek dana



Slika 1 - Diagram slučajeva korišćenja

### **СК1: Случај коришћења - Унос новог конобара**

**Назив СК**

Креирање Конобара

**Актори СК**

Корисник

**Учесници СК**

Корисник и систем (програм)

**Предуслов**: Систем је укључен и корисник je улогован под својом шифром. Систем приказује форму за рад са конобаром.

**Основни сценарио СК**

1. Корисник **уноси** податке оконобару (АПУСО)

1. Корисник **контролише** да ли је коректно унео податке о конобару. (АНСО)
2. Корисник **позива** систем да систем да креира конобара
3. Систем **креира** конобара. (СО)
4. Систем **приказује** кориснику конобара и поруку: “Систем је креирао конобара“. (ИА)

Алтернативна сценарија

5.1 Уколико систем не може да креира конобара он приказује кориснику поруку: “Систем не може да креира конобара”. Прекида се извршење сценариа. (ИА)

### **СК2: Случај коришћења - Измена конобара**

**Назив СК**

Измена Конобара

**Актори СК**

Корисник

**Учесници СК**

Корисник и систем (програм)

**Предуслов**: Систем је укључен и корисник je улогован под својом шифром. Систем приказује форму за рад са конобаром.

**Основни сценарио СК**

1. Корисник **уноси** вредностпо којој претражује конобаре. (АПУСО)
2. Корисник **позива** систем да нађе конобаре по задатој вредности. (АПСО)
3. Систем **тражи** конобаре по задатој вредности. (СО)
4. Систем приказује кориснику конобаре и поруку: “Систем је нашао конобаре по задатој вредности”. (ИА)
5. Корисник бира конобара кога жели да измени (АПУСО)
6. Корисник **уноси** **(мења)** податке о конобару. (АПУСО)
7. Корисник **контролише** да ли је коректно унео податке о конобару. (АНСО)
8. Корисник **позива** систем да запамти податке о конобару. (АПСО)
9. Систем **памти** податке о конобару. (СО)
10. Систем **приказује** кориснику запамћени конобар и поруку: “Систем је запамтио конобара.” (ИА)

Алтернативна сценарија

4.1 Уколико систем не може да нађе конобара он приказује кориснику поруку: “Систем не може да нађе конобара по задатој вредности”. Прекида се извршење сценариа. (ИА)

10.1 Уколико систем не може да запамти податке о конобару он приказује кориснику поруку “Систем не може да запамти конобара”. Прекида се извршење сценариа. (ИА)

### **СК3: Случај коришћења - Брисанје података о конобару**

**Назив СК**

Брисање Конобара

**Актори СК**

Корисник

**Учесници СК**

Корисник и систем (програм)

**Предуслов**: Систем је укључен и корисник je улогован под својом шифром. Систем приказује форму за рад са конобаром.

**Основни сценарио СК**

1. Корисник **уноси** вредностпо којој претражује конобаре. (АПУСО)
2. Корисник **позива** систем да нађе конобаре по задатој вредности. (АПСО)
3. Систем **тражи** конобаре по задатој вредности. (СО)
4. Систем приказује кориснику конобаре и поруку: “Систем је нашао конобаре по задатој вредности”. (ИА)
5. Корисник бира конобара којег жели да обрише. (АПУСО)
6. Корисник **позива** систем да обрише конобара. (АПСО)
7. Систем **брише** конобара. (СО)
8. Систем **приказује** кориснику поруку: “Систем је обрисао конобара.” (ИА)

Алтернативна сценарија

4.1 Уколико систем не може да нађе конобаре он приказује кориснику поруку: “Систем не може да нађе конобаре по задатој вредности”. Прекида се извршење сценариа. (ИА)

8.1 Уколико систем не може да обрише конобара он приказује кориснику поруку “Систем не може да обрише конобара”. (ИА)

### **СК4: Случај коришћенја – Претрага конобара**

**Назив СК**

Претраживање конобара

**Актори СК**

Корисник

**Учесници СК**

Корисник и систем (програм)

**Предуслов**: Систем је укључен и корисник je улогован под својом шифром. Систем приказује форму за рад са конобарима. Листа конобара је учитана.

**Основни сценарио СК**

1. Корисник **уноси** вредностпо којој претражује конобаре. (АПУСО)
2. Корисник **позива** систем да нађе конобаре по задатој вредности. (АПСО)
3. Систем **тражи** конобаре по задатој вредности. (СО)
4. Систем приказује Корисник нађене конобаре и поруку: “Систем је нашао конобара по задатој вредности”. (ИА)

Алтернативна сценарија

4.1 Уколико систем не може да нађе конобара он приказује кориснику поруку: “Систем не може да нађе конобара по задатој вредности”. (ИА)

### **СК5: Случај коришћења – Унос података о пићу**

**Назив СК**

Креирање Пића

**Актори СК**

Корисник

**Учесници СК**

Корисник и систем (програм)

**Предуслов**: Систем је укључен и корисник je улогован под својом шифром. Систем приказује форму за рад са пићем.

**Основни сценарио СК**

1. Корисник **уноси** податке у пиће. (АПУСО)
2. Корисник **контролише** да ли је коректно унео податке у пиће. (АНСО)
3. Корисник **позива** систем да креира пиће. (АПСО)
4. Систем **креира** пиће. (СО)
5. Систем **приказује** кориснику пиће и поруку: “Систем је креирао пиће“. (ИА)

Алтернативна сценарија

5.1 Уколико систем не може да креира пића он приказује кориснику поруку: “Систем не може да креира пића”. Прекида се извршење сценариа. (ИА)

### **СК6: Случај коришћења - Измена података о пићу**

**Назив СК**

Измена Пића

**Актори СК**

Корисник

**Учесници СК**

Корисник и систем (програм)

**Предуслов**: Систем је укључен и корисник je улогован под својом шифром. Систем приказује форму за рад са пићем.

**Основни сценарио СК**

1. Корисник **уноси** вредностпо којој претражује пића. (АПУСО)
2. Корисник **позива** систем да нађе пића по задатој вредности. (АПСО)
3. Систем **тражи** пића по задатој вредности. (СО)
4. Систем приказује кориснику пића и поруку: “Систем је нашао пића по задатој вредности”. (ИА)
5. Корисник бира пиће које жели да измени. (АПУСО)
6. Корисник **уноси** **(мења)** податке о пићу. (АПУСО)
7. Корисник **контролише** да ли је коректно унео податке о пићу. (АНСО)
8. Корисник **позива** систем да запамти податке о пићу. (АПСО)
9. Систем **памти** податке о пићу. (СО)
10. Систем **приказује** кориснику запамћени конобар и поруку: “Систем је запамтио пића.” (ИА)

Алтернативна сценарија

4.1 Уколико систем не може да нађе пића он приказује кориснику поруку: “Систем не може да нађе пића по задатој вредности”. Прекида се извршење сценариа. (ИА)

10.1 Уколико систем не може да запамти податке о пићу он приказује кориснику поруку “Систем не може да запамти пића”. Прекида се извршење сценариа. (ИА)

### **СК7: Случај коришћенја – Претрага пића**

**Назив СК**

Претраживање пића

**Актори СК**

Корисник

**Учесници СК**

Корисник и систем (програм)

**Предуслов**: Систем је укључен и корисник je улогован под својом шифром. Систем приказује форму за рад са пићима. Листа пића је учитана.

**Основни сценарио СК**

1. Корисник **уноси** вредностпо којој претражује конобаре. (АПУСО)
2. Корисник **позива** систем да нађе конобаре по задатој вредности. (АПСО)
3. Систем **тражи** конобаре по задатој вредности. (СО)
4. Систем приказује Корисник нађене конобаре и поруку: “Систем је нашао конобара по задатој вредности”. (ИА)

Алтернативна сценарија

4.1 Уколико систем не може да нађе конобара он приказује кориснику поруку: “Систем не може да нађе конобара по задатој вредности”. (ИА)

### **СК8 Случај коришћења - Unos novog stola**

**Назив СК**

Креирање Stola

**Актори СК**

Корисник

**Учесници СК**

Корисник и систем (програм)

**Предуслов**: Систем је укључен и корисник je улогован под својом шифром. Систем приказује форму за рад са stolom.

**Основни сценарио СК**

1. Корисник **позива** систем да креира сто. (АПСО)
2. Систем **креира** пиће. (СО)
3. Систем **приказује** кориснику пиће и поруку: “Систем је креирао пиће“. (ИА)

Алтернативна сценарија

3.1 Уколико систем не може да креира sto он приказује кориснику поруку: “Систем не може да креира sto”. Прекида се извршење сценариа. (ИА)

### **СК9 Случај коришћења - Унос задужења за сто**

**Назив СК**

Промена Стола

**Актори СК**

Корисник

**Учесници СК**

Корисник и систем (програм)

**Предуслов**: Систем је укључен и корисник je улогован под својом шифром. Систем приказује форму за рад са столом.

**Основни сценарио СК**

1. Корисник **позива** систем да нађе sve стolove. (АПСО)
2. Систем **тражи** сто по задатој вредности. (СО)
3. Систем приказује кориснику сто и поруку: “Систем је нашао стolove ”. (ИА)
4. Корисник **бира** сто у који жели да дода рачун(АПУСО)
5. Корисник **позива** систем да креиранови рачун (АПСО)
6. Корисник **контролише** да ли је коректно унео податке о рачуну. (АНСО)
7. Koрисник **позива** систем да креира нову ставку рачуна(АПСО)
8. Систем **kreira** novu ставку рачуна. (СО)
9. Систем **приказује** кориснику поруку: “Систем је запамтио ставку рачуна.” (ИА)
10. Систем **креира** рачун(СО)
11. Систем приказује кориснику рачун и поруку: „Систем је креирао рачун“ (ИА)
12. Корисник **позива** систем да сачува податке о столу. (АПСО)
13. Систем **памти** сто.(СО)

13. Систем **приказује** кориснику запамћен сто и поруку: “Систем je запамтио сто.(ИА)

Алтернативна сценарија

3.1 Уколико систем не може да нађе сто он приказује кориснику поруку: “Систем не може да нађе сто по задатој вредности”. Прекида се извршење сценариа. (ИА)

8.1 Уколико систем не може да запамти податке о ставци рачуна он приказује кориснику поруку “Систем не може да запамти ставку рачуна”. Прекида се извршење сценариа. (ИА)

10.1 Уколико систем не може да креира рачун, он приказује кориснику поруку: „Систем не може да креира рачун“. Прекида се извршење сценарија(ИА)

13.1 Уколико систем не може да запамти сто он приказује кориснику поруку: “Систем не може да запамти сто”. (ИА)

### **СК10: Случај коришћења - Пресек стања**

**Назив СК**

Пресек стања

**Актори СК**

Корисник

**Учесници СК**

Корисник и систем и контролер (програм)

**Предуслов**: Систем је укључен и корисник je улогован под својом шифром. Систем приказује форму за рад са столовима.

**Основни сценарио СК**

1. Корисник позива систем да врати све столове(АПСО)
2. Систем **тражи** све столове. (СО)
3. Систем vraća kontroleru stolove **приказује** кориснику поруку: “Sistem je našao stolove.” (ИА)
4. Корисник **позива** sistem да ресетује рачуне(АПСО)
5. Систем за сваки сто претражује рачуне(СО)
6. Систем **приказује** кориснику поруку: “Укупан дневни приход је: .” (ИА)

Алтернативна сценарија

3.1 Уколико систем не може да пронађе столове он приказује кориснику поруку “Систем не може да пронађе столове”. (ИА)

6.1 Уколико систем не може израчуна дневни приход он приказује кориснику поруку “Систем не може да израчуна дневни приход ”. (ИА)

# Analiza sistema

## Sistemski dijagram sekvenci

### DS1: Diagram sekvenci za slučaj korišćenja **unos novog konobara**

Glavni scenario:

1. Корисник **позива** систем да креира конобара. (АПСО)
2. Систем **приказује** кориснику конобара и поруку: “Систем је креирао конобара“. (ИА)



Alternativni scenario:

2.1 Уколико систем не може да креира конобара он приказује кориснику поруку: “Систем не може да креира конобара”. Прекида се извршење сценариа. (ИА)



* 1. Уколико систем не може да запамти податке о конобару он приказује кориснику поруку “Систем не може да запамти конобара”. (ИА)

### DS2: Diagram sekvenci za slučaj korišćenja **izmena konobara**

Glavni scenarijo:

1. Корисник **позива** систем да нађе конобара по задатој вредности. (АПСО)
2. Систем приказује кориснику конобара и поруку: “Систем је нашао конобаре по задатој вредности”. (ИА)
3. Корисник **позива** систем да запамти податке о конобару. (АПСО)
4. Систем **приказује** кориснику запамћени конобар и поруку: “Систем је запамтио конобара.” (ИА)



Alternativni scenariji:

* 1. Уколико систем не може да нађе конобара он приказује кориснику поруку: “Систем не може да нађе конобара по задатој вредности”. Прекида се извршење сценариа. (ИА)



6.1 Уколико систем не може да запамти податке о конобару он приказује кориснику поруку “Систем не може да запамти конобара”. Прекида се извршење сценариа. (ИА)



### DS3: Dijagram sekvenci za slučaj korišćenja **brisanje konobara**

Glavni scenarijo:

1. Корисник **позива** систем да нађе конобара по задатој вредности. (АПСО)
2. Систем приказује кориснику конобара и поруку: “Систем је нашао конобара по задатој вредности”. (ИА)
3. Корисник **позива** систем да обрише конобара. (АПСО)
4. Систем **приказује** кориснику поруку: “Систем је обрисао конобара.” (ИА)



Alternativni scenariji:

2.1 Уколико систем не може да нађе конобара он приказује кориснику поруку: “Систем не може да нађе конобара по задатој вредности”. Прекида се извршење сценариа. (ИА)



4.1 Уколико систем не може да обрише конобара он приказује кориснику поруку “Систем не може да обрише конобара”. (ИА)



### DS4: Dijagram sekvenci za slučaj korišćenja **Preraga konobara**

Glavni scenarijo:

1. Корисник **позива** систем да нађе конобаре по задатој вредности. (АПСО)
2. Систем приказује Корисник нађене конобаре и поруку: “Систем је нашао конобара по задатој вредности”. (ИА)



Alternativni scenariji:

2.1 Уколико систем не може да нађе конобара он приказује кориснику поруку: “Систем не може да нађе конобара”. Прекида се извршење сценариа. (ИА)



### DS5:Dijagram sekvenci za slučaj korišćenja **Unos podataka o pića**

Glavni scenarijo:

1. Корисник **позива** систем да креира пића. (АПСО)
2. Систем **приказује** кориснику пића и поруку: “Систем је креирао пића“. (ИА)
3. Корисник **позива** систем да запамти податке о пићу. (АПСО)
4. Систем **приказује** кориснику запамћени конобар и поруку: “Систем је запамтио конобар“. (ИА)



Alternativni scenariji:

2.1 Уколико систем не може да креира пића он приказује кориснику поруку: “Систем не може да креира пића”. Прекида се извршење сценариа. (ИА)



### DS6: Dijagram sekvenci za slučaj korišćenja **Izmena pića**

Glavni scenarijo:

1. Корисник **позива** систем да нађе пића по задатој вредности. (АПСО)
2. Систем приказује кориснику пића и поруку: “Систем је нашао пића по задатој вредности”. (ИА)
3. Корисник **позива** систем да запамти податке о пићу. (АПСО)
4. Систем **приказује** кориснику запамћени конобар и поруку: “Систем је запамтио пића.” (ИА)



Alternativni scenariji:

* 1. Уколико систем не може да нађе пића он приказује кориснику поруку: “Систем не може да нађе пића по задатој вредности”. Прекида се извршење сценариа. (ИА)



4.1 Уколико систем не може да запамти податке о пићу он приказује кориснику поруку “Систем не може да запамти пића”. Прекида се извршење сценариа. (ИА)



### DS7: Dijagram sekvenci za slučaj korišćenja **pretraga pića**

Glavni scenarijo:

1. Корисник **уноси** вредностпо којој претражује pića. (АПУСО)
2. Систем приказује Корисник нађенa pića и поруку: “Систем је нашао piće по задатој вредности”. (ИА)



Alternativni scenariji:

2.1 Уколико систем не може да нађе пића он приказује кориснику поруку: “Систем не може да нађе пића по задатој вредности”. Прекида се извршење сценариа. (ИА)



### DS8: Dijagram sekvenci za slučaj koršćenja **unos novog stola**

Glavni scenarijo:

1. Корисник **позива** систем да креира sto. (АПСО)
2. Систем **приказује** кориснику sto и поруку: “Систем је креирао sto“. (ИА)



Alternativni scenariji:

2.1 Уколико систем не може да креира sto он приказује кориснику поруку: “Систем не може креира sto”. Прекида се извршење сценариа. (ИА)



### DS9: Dijagram sekvenci za slučaj koršćenja **Unos zaduženja za sto**

Glavni scenarijo:

1. Корисник **позива** систем да нађе sve стolove. (АПСО)
2. Систем приказује кориснику сто и поруку: “Систем је нашао стolove ”. (ИА)
3. Корисник **позива** систем да креиранови рачун (АПСО)
4. Систем **приказује** кориснику поруку: “Систем је запамтио ставку рачуна.” (ИА)
5. Систем приказује кориснику рачун и поруку: „Систем је креирао рачун“ (ИА)
6. Корисник **позива** систем да сачува податке о столу. (АПСО)

7. Систем **приказује** кориснику запамћен сто и поруку: “Систем je запамтио сто.(ИА)



Alternativni scenariji:

2.1 Уколико систем не може да нађе сто он приказује кориснику поруку: “Систем не може да нађе сто по задатој вредности”. Прекида се извршење сценариа. (ИА)



4.1 Уколико систем не може да креира ставку рачуна, он приказује кориснику поруку: „Систем не може да креира ставку рачуна“. Прекида се извршење сценарија(ИА)



5.1 Уколико систем не може да креира рачун, он приказује кориснику поруку: „Систем не може да креира рачун“. Прекида се извршење сценарија(ИА)



7.1 Уколико систем не може да запамти сто он приказује кориснику поруку: “Систем не може да запамти сто”. (ИА)



### DS10: Dijagram sekvenci za slučaj koršćenja **Presek stanja**

Glavni scenarijo:

1. Корисник позива систем да врати све столове(АПСО)
2. Систем vraća kontroleru stolove **приказује** кориснику поруку: “Sistem je našao stolove.” (ИА)
3. Корисник **позива** sistem да направи пресек стања(АПСО)
4. Систем **приказује** кориснику поруку: “Укупан дневни приход је: .” (ИА)



Alternativni scenariji:

2.1 Уколико систем не може да нађе столове он приказује кориснику поруку: “Систем не може да нађе столове”. (ИА)



6.1 Уколико систем не може да израчуна дневни приход он приказује кориснику поруку: “Систем не може да израчуна дневни приход ”. (ИА)



## Definisanje ugovora o sistemskim operacijama

### *Ugovor UG1: sacuvajKonobara*

* **Operacija:** saveWaiter():signal
* **Veza sa SK:** SK1
* **Preduslov:** Vrednosna i strukturna ograničenja nad objektom Konobar moraju biti zadovoljena
* **Postuslov:** Napravljen je novi Konobar

### *Ugovor UG2: pronadjiKonobara*

* **Operacija:** searchWaiter(Konobar k): signal
* **Veza sa SK:** SK2, SK3, SK4
* **Preduslov:**
* **Postuslov:**

### *Ugovor UG3: obirsiKonobara*

* **Operacija:** deleteWaiter(Konobar k): signal
* **Veza sa SK:** SK3
* **Preduslov:** Strukturna ograničenja nad objektom Pice moraju biti zadovoljena
* **Postuslov:** Konobar je obrisan

### *Ugovor UG4: kreirajPice*

* **Operacija:**  saveDrink (): signal
* **Veza sa SK:** SK5
* **Preduslov:** Vrednosna i strukturna ograničenja nad objektom Pice moraju biti zadovoljena
* **Postuslov:** Napravljeno je novo Pice

### *Ugovor UG5: pronadjiPice*

* **Operacija:** searchDrink (Pice p): signal
* **Veza sa SK:** SK7, SK6
* **Preduslov:**
* **Postuslov:**

### *Ugovor UG6: zapamtiSto*

* **Operacija:** saveTable (): signal
* **Veza sa SK:** SK8
* **Preduslov:** Vrednosna i strukturna ograničenja nad objektom Sto moraju biti zadovoljena
* **Postuslov:** Sto je zapamćen

### *Ugovor UG7: zapamtiRačun*

* **Operacija:** saveBill (): signal
* **Veza sa SK:** SK9
* **Preduslov:** Vrednosna i strukturna ograničenja nad objektom Racun moraju biti zadovoljena
* **Postuslov:** Racun je zapamćen

### *Ugovor UG8: pretražiRačun*

* **Operacija:** searchBill(String s): signal
* **Veza sa SK:** SK9
* **Preduslov:**
* **Postuslov:**

### *Ugovor UG9: obrisiRacun*

* **Operacija:** deleteRacun(Racun r): signal
* **Veza sa SK:** SK10
* **Preduslov:** Strukturna ograničenja nad objektom Racun moraju biti zadovoljena
* **Postuslov:** Racun je obrisan

### *Ugovor UG10: obrisiStavkuRacuna*

* **Operacija:** deleteStavkaRacuna(StavkaRacuna s): signal
* **Veza sa SK:** SK10
* **Preduslov:** Strukturna ograničenja nad objektom StavkaRacuna moraju biti zadovoljena
* **Postuslov:** StavkaRacuna je obrisana

## Konceptualni dijagram klasa



## Relaconi model

Konobar(konobarID, JMBG, ime, username, password)

Sto(stoID, ukupanDnevniIznos, zauzet)

Pice(piceID, nazivPica, cena)

StavkaRacuna(racunID, brStavkeRacuna, piceID)

Racun(racunID, iznos, konobarID, stoID, placen)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Konobar** | | **Prosto vrednosno ograničenje** | | **Složeno vrednosno ograničenje** | | **Strukturno ograničenje** |
| **Atributi** | **Ime** | **Tip atributa** | **Vrednost atributa** | **Međuzav. atributa jedne tabele** | **Međuzav. atributa više tabela** | INSERT /  UPDATE CASCADES  Racun  DELETE CASCADES  Racun |
| konobarID | Integer | not null and >0 |  |  |
| JMBG | String | Not null |  |  |
| ime | String | Not null |  |  |
| username | String | Not null |  |  |
| password | String | Not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Pice** | | **Prosto vrednosno ograničenje** | | **Složeno vrednosno ograničenje** | | **Strukturno ograničenje** |
| **Atributi** | **Ime** | **Tip atributa** | **Vrednost atributa** | **Međuzav. atributa jedne tabele** | **Međuzav. atributa više tabela** | INSERT /  UPDATE /  DELETE  CASCADES  StavkaRacuna |
| piceID | Integer | not null and >0 |  |  |
| nazivPica | String | Not null |  |  |
| cena | double | Not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Sto** | | **Prosto vrednosno ograničenje** | | **Složeno vrednosno ograničenje** | | **Strukturno ograničenje** |
| **Atributi** | **Ime** | **Tip atributa** | **Vrednost atributa** | **Međuzav. atributa jedne tabele** | **Međuzav. atributa više tabela** | INSERT /  UPDATE /  DELETE CASCADES Racun |
| stoID | Integer | not null and >0 |  |  |
| ukupanDnevniIznos | double | Not null |  |  |
| Zauzet | int | 0 or 1 |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Racun** | | **Prosto vrednosno ograničenje** | | **Složeno vrednosno ograničenje** | | **Strukturno ograničenje** |
| **Atributi** | **Ime** | **Tip atributa** | **Vrednost atributa** | **Međuzav. atributa jedne tabele** | **Međuzav. atributa više tabela** | INSERT RESTRICTED  StavkaRacuna  UPDATE CASCADES  StavkaRacuna DELETE CASCADES  StavkaRacuna |
| racunID | Integer | not null and >0 |  |  |
| iznos | double | Not null |  |  |
| konobarID | int | Not null |  |  |
| stoID | String | Not null |  |  |
| Placen | int | 0 or 1 |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela StavkaRacuna** | | **Prosto vrednosno ograničenje** | | **Složeno vrednosno ograničenje** | | **Strukturno ograničenje** |
| **Atributi** | **Ime** | **Tip atributa** | **Vrednost atributa** | **Međuzav. atributa jedne tabele** | **Međuzav. atributa više tabela** | INSERT /  UPDATE CASCADES  Racun  DELETE  CASCADES  StavkaRacuna |
| racunID | Integer | not null and >0 |  |  |
| brStavkeRacuna | Integer | Not null and > 0 |  |  |
| cena | double | Not null |  |  |

# Projektovanje

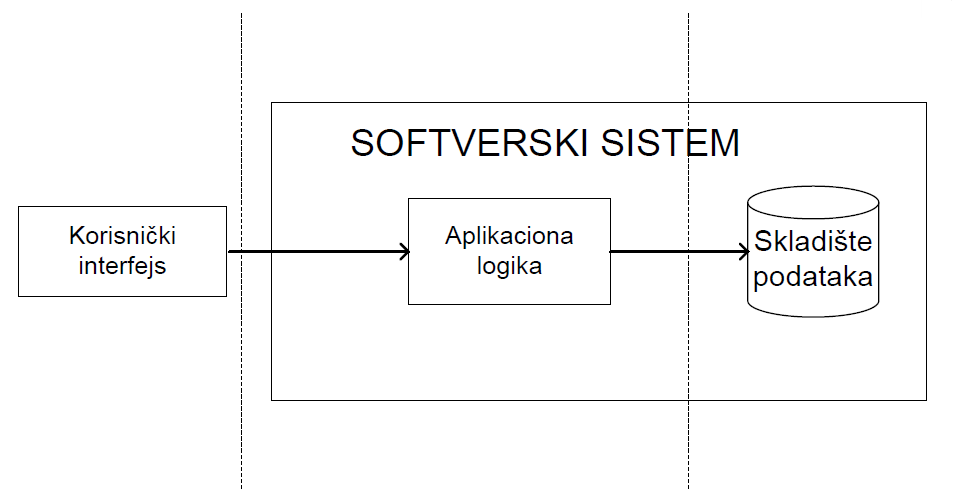
U fazi projektovanja, opisujemo fizičku strukturu softverskog sistema, kao i njegovo ponašanje.

## 3.1 Arhitektura softverskog sistema

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

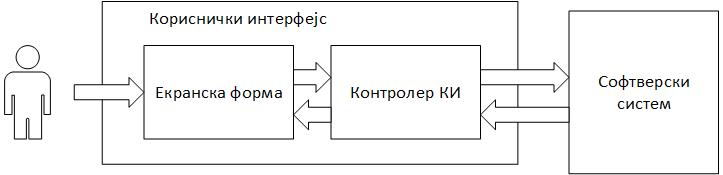
* korisnički interfejs
* aplikaciona logika
* skladište podataka

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



## 3.2 Projektovanje korisničkog interfejsa

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



Ekranska forma ima ulogu da prihvati podatke koje unosi korisnik, prihvata događaje koje pravi korisnik, prima i šalje podatke kontroleru KI za komunikaciju sa serverom.

### 3.2.1 Projektovanje ekranskih formi

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

#### 3.2.1.1 Slučaj korišćenja - Unos novog konobara

**Назив СК**

Креирање Конобара

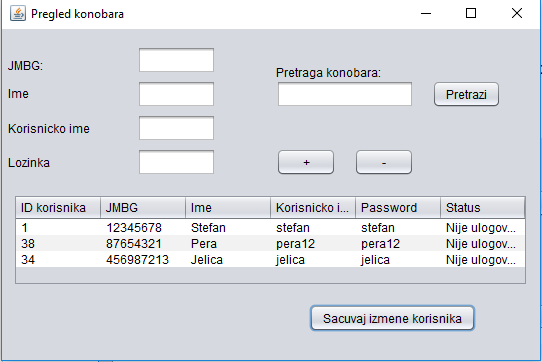
**Актори СК**

Корисник

**Учесници СК**

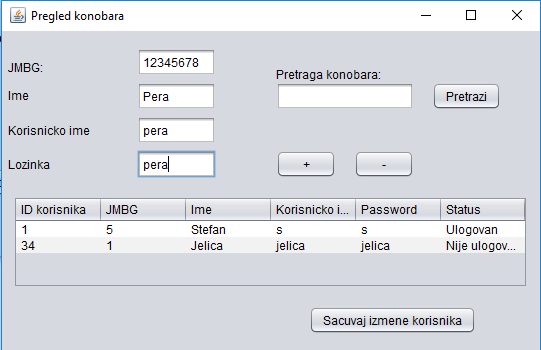
Корисник и систем (програм)

**Предуслов**: Систем је укључен и корисник je улогован под својом шифром. Систем приказује форму за рад са конобаром.



**Основни сценарио СК**

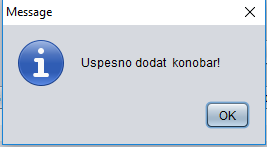
1. Корисник **уноси** податке оконобару (АПУСО)



1. Корисник **контролише** да ли је коректно унео податке о конобару. (АНСО)
2. Корисник **позива** систем да систем да креира конобара

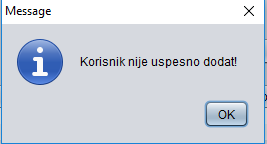
Opis akcije: Korisnik klikom na dugme „+“ poziva sistemsku operaciju SOSaveWaiter() koja pamti konobara.

1. Систем **креира** конобара. (СО)
2. Систем **приказује** кориснику конобара и поруку: “Систем је креирао конобара“. (ИА)



Алтернативна сценарија

5.1 Уколико систем не може да креира конобара он приказује кориснику поруку: “Систем не може да креира конобара”. Прекида се извршење сценариа. (ИА)



#### 3.2.1.2 Slučaj korišćenja - Izmena konobara

**Назив СК**

Измена Конобара

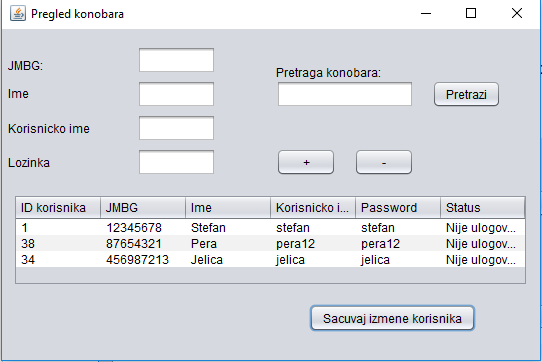
**Актори СК**

Корисник

**Учесници СК**

Корисник и систем (програм)

**Предуслов**: Систем је укључен и корисник je улогован под својом шифром. Систем приказује форму за рад са конобаром.

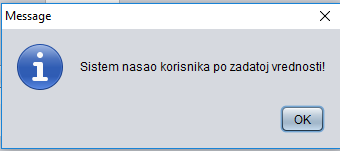


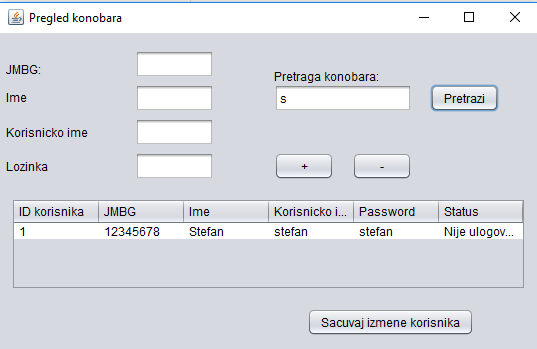
**Основни сценарио СК**

1. Корисник **уноси** вредностпо којој претражује конобаре. (АПУСО)

Opis akcije: korisnik unosi reč za pretragu u polje

1. Корисник **позива** систем да нађе конобаре по задатој вредности. (АПСО)
2. Opis akcije: Klikkom na dugme „pretrazi“ korisnik poziva sistemsku operaciju SOSearchWaiter() koja pretražuje konobare
3. Систем **тражи** конобаре по задатој вредности. (СО)
4. Систем приказује кориснику конобаре и поруку: “Систем је нашао конобаре по задатој вредности”. (ИА)



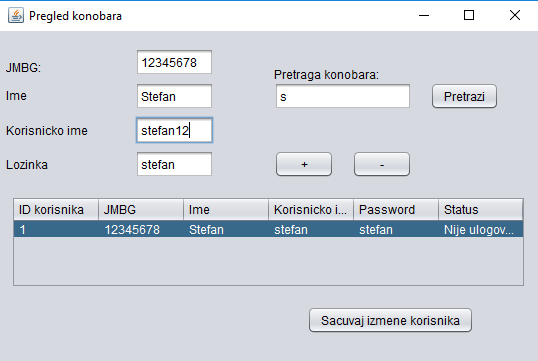


1. Корисник бира конобара кога жели да измени (АПУСО)

Opis akcije: klikom na red u tabeli, u polja se učitavaju trenutne vrednosti izabranog konobara

1. Корисник **уноси** **(мења)** податке о конобару. (АПУСО)

Opis akcije: Korisnik menja vrednosti u poljima



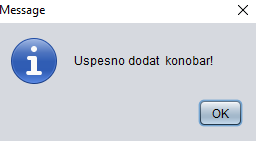
1. Корисник **контролише** да ли је коректно унео податке о конобару. (АНСО)

Unete vrednosti moraju ispuniti određene kriterijume

1. Корисник **позива** систем да запамти податке о конобару. (АПСО)

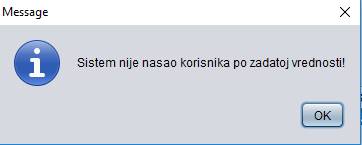
Opis akcije: Klikom na dugme „Sačuvaj izmene korisnika“ poziva se sistemska operacija SOSaveUser() koja update-uje konobara

1. Систем **памти** податке о конобару. (СО)
2. Систем **приказује** кориснику запамћени конобар и поруку: “Систем је запамтио конобара.” (ИА)

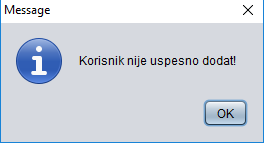


Алтернативна сценарија

4.1 Уколико систем не може да нађе конобара он приказује кориснику поруку: “Систем не може да нађе конобара по задатој вредности”. Прекида се извршење сценариа. (ИА)



10.1 Уколико систем не може да запамти податке о конобару он приказује кориснику поруку “Систем не може да запамти конобара”. Прекида се извршење сценариа. (ИА)



#### 3.2.1.3 Slučaj korišćenja – Brisanje podataka o konobaru

**Назив СК**

Брисање Конобара

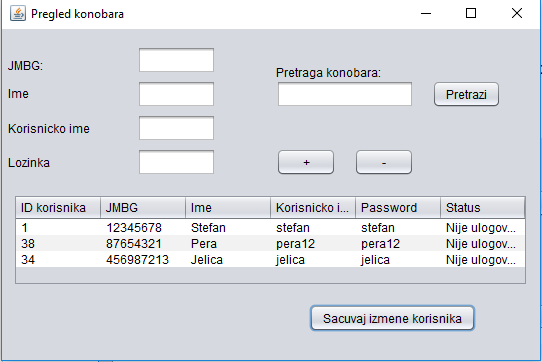
**Актори СК**

Корисник

**Учесници СК**

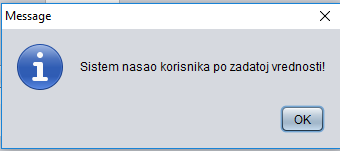
Корисник и систем (програм)

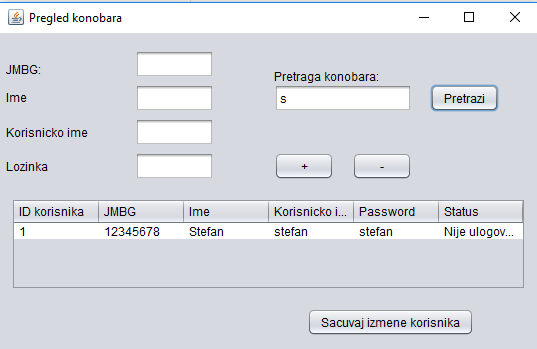
**Предуслов**: Систем је укључен и корисник je улогован под својом шифром. Систем приказује форму за рад са конобаром.



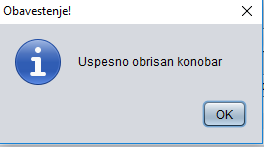
**Основни сценарио СК**

1. Корисник **уноси** вредностпо којој претражује конобаре. (АПУСО)
2. Корисник **позива** систем да нађе конобаре по задатој вредности. (АПСО)
3. Систем **тражи** конобаре по задатој вредности. (СО)
4. Систем приказује кориснику конобаре и поруку: “Систем је нашао конобаре по задатој вредности”. (ИА)



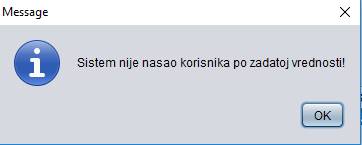


1. Корисник бира конобара којег жели да обрише. (АПУСО)
2. Корисник **позива** систем да обрише конобара. (АПСО)
3. Систем **брише** конобара. (СО)
4. Систем **приказује** кориснику поруку: “Систем је обрисао конобара.” (ИА)

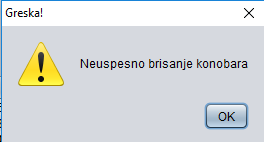


Алтернативна сценарија

4.1 Уколико систем не може да нађе конобаре он приказује кориснику поруку: “Систем не може да нађе конобаре по задатој вредности”. Прекида се извршење сценариа. (ИА)



8.1 Уколико систем не може да обрише конобара он приказује кориснику поруку “Систем не може да обрише конобара”. (ИА)



#### 3.2.1.4 Slučaj korišćenja – Pretraga konobara

**Назив СК**

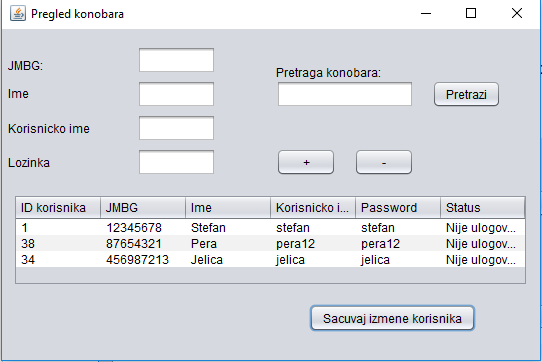
Претраживање конобара

**Актори СК**

Корисник

**Учесници СК**

Корисник и систем (програм)



**Предуслов**: Систем је укључен и корисник je улогован под својом шифром. Систем приказује форму за рад са конобарима. Листа конобара је учитана.

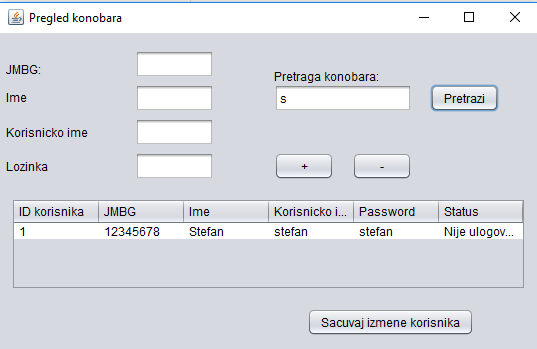
**Основни сценарио СК**

1. Корисник **уноси** вредностпо којој претражује конобаре. (АПУСО)

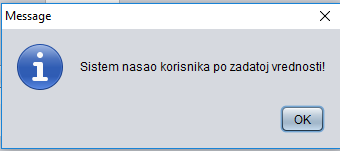
Opis akcije: korisnik unosi reč za pretragu u polje

1. Корисник **позива** систем да нађе конобаре по задатој вредности. (АПСО)

Opis akcije: Klikkom na dugme „pretrazi“ korisnik poziva sistemsku operaciju SOSearchWaiter() koja pretražuje konobare

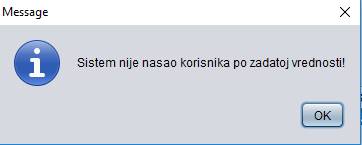


1. Систем **тражи** конобаре по задатој вредности. (СО)
2. Систем приказује кориснику конобаре и поруку: “Систем је нашао конобаре по задатој вредности”. (ИА)



Алтернативна сценарија

4.1 Уколико систем не може да нађе конобара он приказује кориснику поруку: “Систем не може да нађе конобара по задатој вредности”. (ИА)



#### 3.2.1.5 Slučaj korišćenja – Unos podataka o piću

**Назив СК**

Креирање Пића

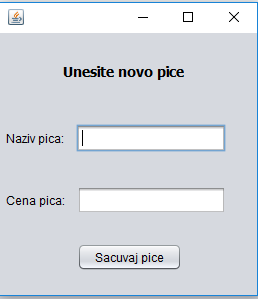
**Актори СК**

Корисник

**Учесници СК**

Корисник и систем (програм)

**Предуслов**: Систем је укључен и корисник je улогован под својом шифром. Систем приказује форму за рад са пићем.



**Основни сценарио СК**

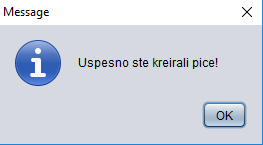
1. Корисник **уноси** податке у пиће. (АПУСО)

Opis akcije: korisnik popunjava polja forme

1. Корисник **контролише** да ли је коректно унео податке у пиће. (АНСО)
2. Корисник **позива** систем да креира пиће. (АПСО)

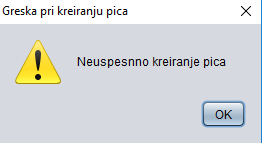
Opis akcije: Klikom na dugme „Sacuvaj pice“ korisnik poziva sistemsku operaciju SONewDrink() koja kreira piće

1. Систем **креира** пиће. (СО)
2. Систем **приказује** кориснику пиће и поруку: “Систем је креирао пиће“. (ИА)



Алтернативна сценарија

5.1 Уколико систем не може да креира пића он приказује кориснику поруку: “Систем не може да креира пића”. Прекида се извршење сценариа. (ИА)



#### 3.2.1.6 Slučaj korišćenja – Izmena podataka o piću

**Назив СК**

Измена Пића

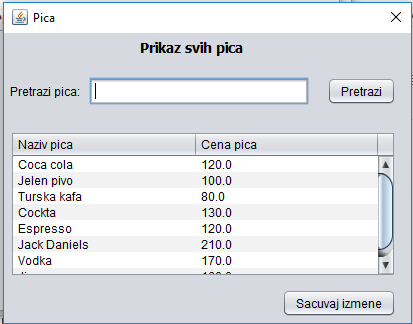
**Актори СК**

Корисник

**Учесници СК**

Корисник и систем (програм)

**Предуслов**: Систем је укључен и корисник je улогован под својом шифром. Систем приказује форму за рад са пићем.



**Основни сценарио СК**

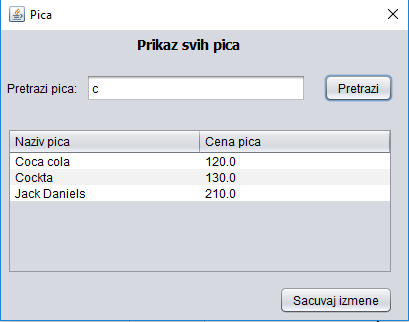
1. Корисник **уноси** вредностпо којој претражује пића. (АПУСО)

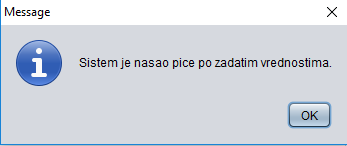
Opis akcije: Korisnik popunjava polje za pretragu

1. Корисник **позива** систем да нађе пића по задатој вредности. (АПСО)

Opis akcije: Klikom na dugme „Pretrazi“ korisnik zove sistemsku operaciju SOSearchDrinks() koja pretražuje pića

1. Систем **тражи** пића по задатој вредности. (СО)
2. Систем приказује кориснику пића и поруку: “Систем је нашао пића по задатој вредности”. (ИА)



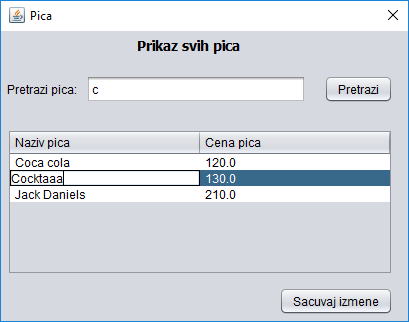


1. Корисник бира пиће које жели да измени. (АПУСО)

Opis akcije: klikom na red u tabeli, korisnik bira piće koje će menjati

1. Корисник **уноси** **(мења)** податке о пићу. (АПУСО)

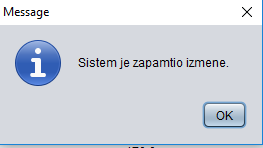
Opis akcije: duplim klikom na polje tabele, korisnik unosi vrednosti



1. Корисник **контролише** да ли је коректно унео податке о пићу. (АНСО)
2. Корисник **позива** систем да запамти податке о пићу. (АПСО)

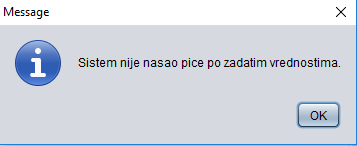
Opis akcije: Klikom na dugme „Sacuvaj izmene“ korisnik poziva sistemsku operaciju SONewDrink(), koja pamti izmene

1. Систем **памти** податке о пићу. (СО)
2. Систем **приказује** кориснику запамћени конобар и поруку: “Систем је запамтио пића.” (ИА)

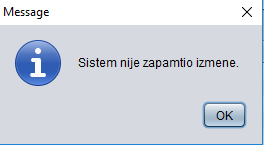


Алтернативна сценарија

4.1 Уколико систем не може да нађе пића он приказује кориснику поруку: “Систем не може да нађе пића по задатој вредности”. Прекида се извршење сценариа. (ИА)



10.1 Уколико систем не може да запамти податке о пићу он приказује кориснику поруку “Систем не може да запамти пића”. Прекида се извршење сценариа. (ИА)



#### 3.2.1.7 Slučaj korišćenja – Pretraga pića

**Назив СК**

Претраживање пића

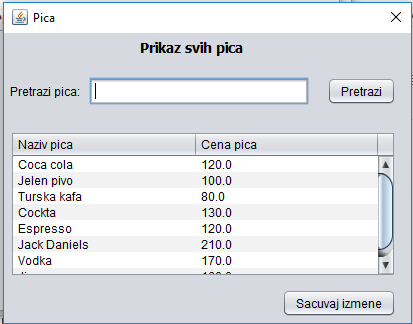
**Актори СК**

Корисник

**Учесници СК**

Корисник и систем (програм)

**Предуслов**: Систем је укључен и корисник je улогован под својом шифром. Систем приказује форму за рад са пићима. Листа пића је учитана.



**Основни сценарио СК**

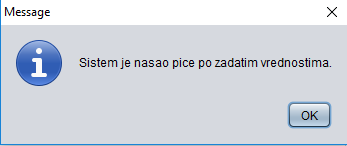
1. Корисник **уноси** вредностпо којој претражује пића. (АПУСО)

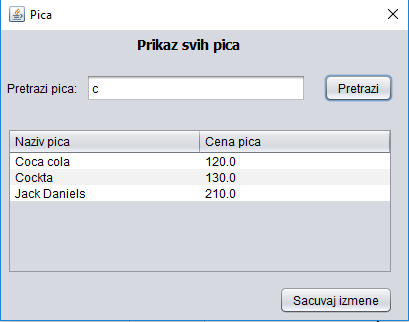
Opis akcije: Korisnik popunjava polje za pretragu

1. Корисник **позива** систем да нађе пића по задатој вредности. (АПСО)

Opis akcije: Klikom na dugme „Pretrazi“ korisnik zove sistemsku operaciju SOSearchDrinks() koja pretražuje pića

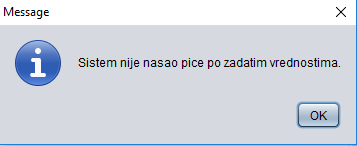
1. Систем **тражи** пића по задатој вредности. (СО)
2. Систем приказује кориснику пића и поруку: “Систем је нашао пића по задатој вредности”. (ИА)





Алтернативна сценарија

4.1 Уколико систем не може да нађе конобара он приказује кориснику поруку: “Систем не може да нађе конобара по задатој вредности”. (ИА)



#### 3.2.1.8 Slučaj korišćenja – Unos novog stola

**Назив СК**

Креирање Stola

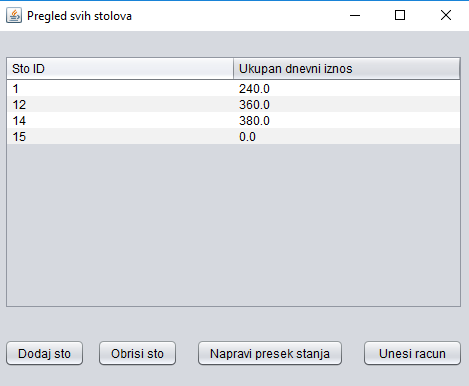
**Актори СК**

Корисник

**Учесници СК**

Корисник и систем (програм)

**Предуслов**: Систем је укључен и корисник je улогован под својом шифром. Систем приказује форму за рад са stolom.

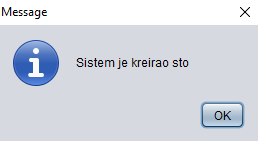


**Основни сценарио СК**

1. Корисник **позива** систем да креира сто. (АПСО)

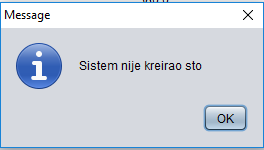
Opis akcije: Klikom na dugme „Dodaj sto“ poziva se sistemska operacije SONewTable() koja kreira sto

1. Систем **креира** сто. (СО)
2. Систем **приказује** кориснику пиће и поруку: “Систем је креирао сто“. (ИА)



Алтернативна сценарија

3.1 Уколико систем не може да креира sto он приказује кориснику поруку: “Систем не може да креира sto”. Прекида се извршење сценариа. (ИА)



#### 3.2.1.9 – Slučaj korišćenja – Unos zaduženja za sto

**Назив СК**

Промена Стола

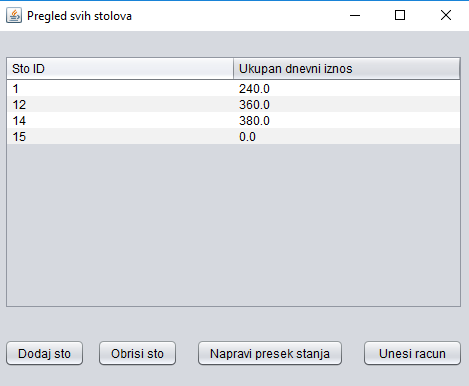
**Актори СК**

Корисник

**Учесници СК**

Корисник и систем (програм)

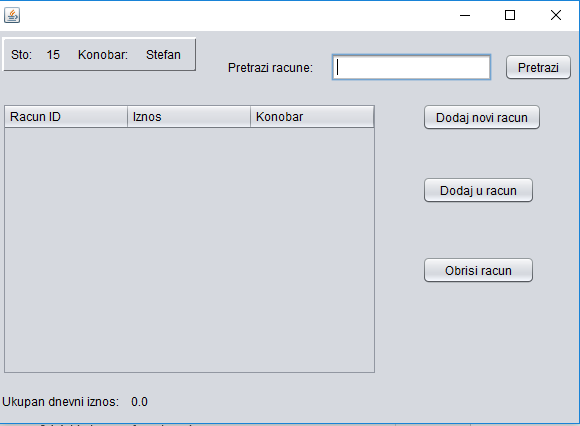
**Предуслов**: Систем је укључен и корисник je улогован под својом шифром. Систем приказује форму за рад са столом.



**Основни сценарио СК**

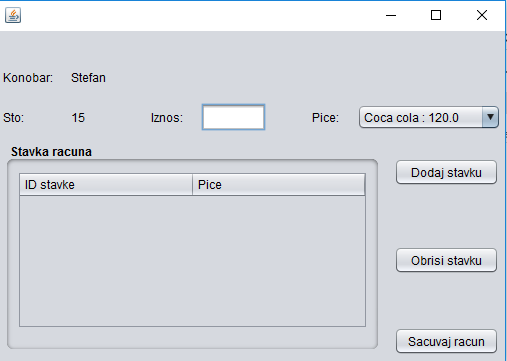
1. Корисник **позива** систем да нађе sve стolove. (АПСО)
2. Систем **тражи** сто по задатој вредности. (СО)
3. Систем приказује кориснику сто и поруку: “Систем је нашао стolove ”. (ИА)
4. Корисник **бира** сто у који жели да дода рачун(АПУСО)
5. Корисник **позива** систем да креиранови рачун (АПСО)

Opis akcije: klikom na dugme „Unesi racun“ otvara se forma za unos računa

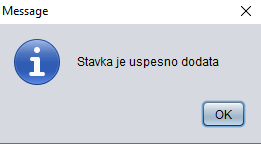


1. Корисник **контролише** да ли је коректно унео податке о рачуну. (АНСО)
2. Koрисник **позива** систем да креира нову ставку рачуна

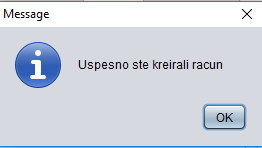
Opis akcije: Klikom na dugme „Dodaj novi racun“ otvara se forma za unos stavke računa



1. Систем **kreira** novu ставку рачуна. (СО)
2. Систем **приказује** кориснику поруку: “Систем је запамтио ставку рачуна.” (ИА)

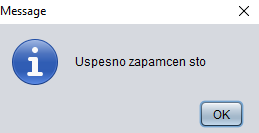


1. Систем **креира** рачун(СО)
2. Систем приказује кориснику рачун и поруку: „Систем је креирао рачун“ (ИА)



1. Корисник **позива** систем да сачува податке о столу. (АПСО)
2. Систем **памти** сто.(СО)

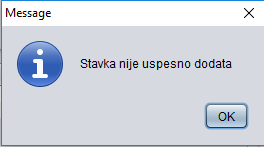
14. Систем **приказује** кориснику запамћен сто и поруку: “Систем je запамтио сто.(ИА)



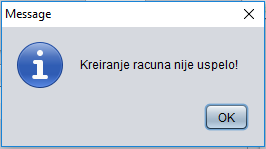
Алтернативна сценарија

3.1 Уколико систем не може да нађе сто он приказује кориснику поруку: “Систем не може да нађе сто по задатој вредности”. Прекида се извршење сценариа. (ИА)

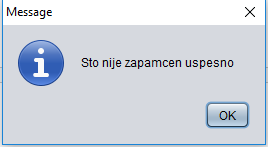
8.1 Уколико систем не може да запамти податке о ставци рачуна он приказује кориснику поруку “Систем не може да запамти ставку рачуна”. Прекида се извршење сценариа. (ИА)



10.1 Уколико систем не може да креира рачун, он приказује кориснику поруку: „Систем не може да креира рачун“. Прекида се извршење сценарија(ИА)



13.1 Уколико систем не може да запамти сто он приказује кориснику поруку: “Систем не може да запамти сто”. (ИА)



#### 3.2.1.10 Slučaj korišćenja – Presek stanja

**Назив СК**

Presek stanja

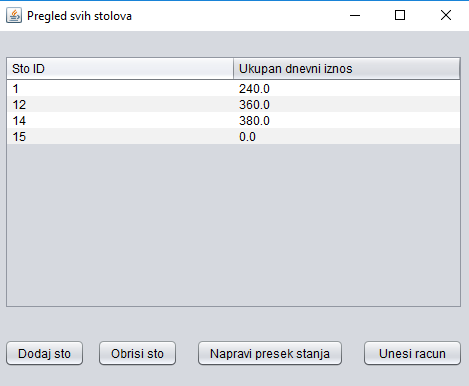
**Актори СК**

Корисник

**Учесници СК**

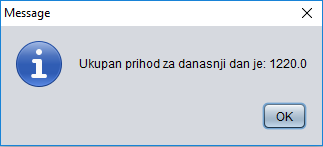
Корисник и систем и контролер (програм)

**Предуслов**: Систем је укључен и корисник je улогован под својом шифром. Систем приказује форму за рад са столовима.



**Основни сценарио СК**

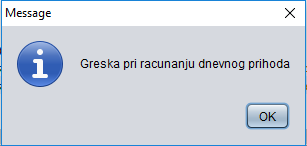
1. Корисник позива систем да врати све столове(АПСО)
2. Систем **тражи** све столове. (СО)
3. Систем vraća kontroleru stolove **приказује** кориснику поруку: “Sistem je našao stolove.” (ИА)
4. Корисник **позива** sistem да направи пресек(АПСО)
5. Систем **приказује** кориснику поруку: “Укупан дневни приход је: .” (ИА)



Алтернативна сценарија

3.1 Уколико систем не може да пронађе столове он приказује кориснику поруку “Систем не може да пронађе столове”. (ИА)

5.1 Уколико систем не може израчуна дневни приход он приказује кориснику поруку “Систем не може да израчуна дневни приход ”. (ИА)



### 3.2.2. Projektovanje kontrolera korisničkog interfejsa

Kontroler korisničkog interfejsa je odgovoran za:

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

## 3.3 Projektovanje aplikacione logike

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

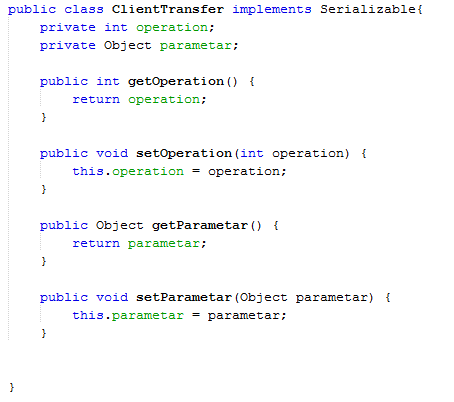
* deo za komunikaciju s klijentima
* kontrolera aplikacione logike
* deo koji sadrži poslovnu logiku
* deo za komunikaciju sa skladištem podataka (broker baze podataka)

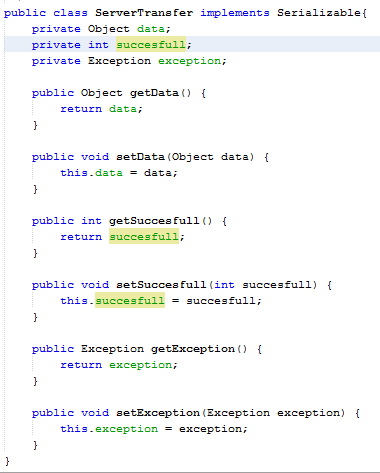
### 3.3.1 Komunikacija sa klijentima

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

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

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





### 3.3.2 Kontroler aplikacione logike

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

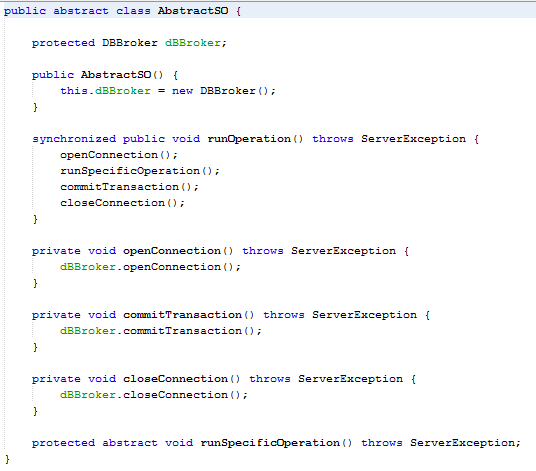
Kako je u fazama prikupljanja zahteva i analize data specifikacija strukture i ponašanja softverskog sistema, odnosno specifikacija poslovne logike softverskog sistema, sledeća slika daje opis sistema nakon faze projektovanja komunikacije sa klijentima i kontrolera aplikacione logike:



Klasa kontroler prima zahtevi od niti klijenta za izvršavanje sistemskih operacija, koji se zatim prosleđuju klasama odgovornim za izvršenje sistemskih operacija. Za svaku sistemsku operaciju prave se softverske klase koje treba da realizuju sistemsku operaciju. Sve klase za realizaciju SO, nasleđuju AbstractSO. Njena metoda runOperation() odrađuje konekciju na bazu, commit i zatvaranje konekcije. Ostale klase implementiraju samo runSpecificOperation().



Opšta sistemska operacija je predstavljena klasom AbstractSO:

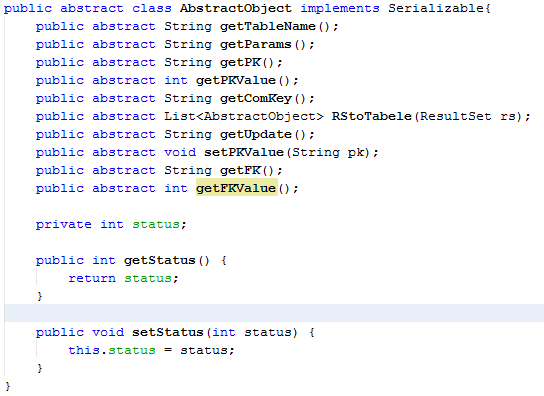


### 3.3.3 Projektovanje strukture softverskog sistema



Domenske klase su: Konobar, Piće, Sto, Račun, StavkaRacuna. Svaka od domenskih klasa, ima svoje privatne atribute, javne get i set metode, kao i parametarski i neparametarski konstruktor. Sve domenske klase nasleđuju AbstractObject klasu, koja sadrži dodatna polja i metode, kojima se postiže apstrakcija.

Klasa AbstractObject:



### 3.3.4 Projektovanje sistemskih operacija

#### Ugovor UG1: sacuvajKonobara

* **Operacija:** saveWaiter():signal
* **Veza sa SK:** SK1
* **Preduslov:** Vrednosna i strukturna ograničenja nad objektom Konobar moraju biti zadovoljena
* **Postuslov:** Napravljen je novi Konobar



#### Ugovor UG2: pronadjiKonobara

* **Operacija:** searchWaiter(Konobar k): signal
* **Veza sa SK:** SK2, SK3, SK4
* **Preduslov:**
* **Postuslov:**



#### Ugovor UG3: obirsiKonobara

* **Operacija:** deleteWaiter(Konobar k): signal
* **Veza sa SK:** SK3
* **Preduslov:** Strukturna ograničenja nad objektom Pice moraju biti zadovoljena
* **Postuslov:** Konobar je obrisan



#### Ugovor UG4: kreirajPice

* **Operacija:**  saveDrink (): signal
* **Veza sa SK:** SK5
* **Preduslov:** Vrednosna i strukturna ograničenja nad objektom Pice moraju biti zadovoljena
* **Postuslov:** Napravljeno je novo Pice



#### Ugovor UG5: pronadjiPice

* **Operacija:** searchDrink (Pice p): signal
* **Veza sa SK:** SK7, SK6
* **Preduslov:**
* **Postuslov:**



#### Ugovor UG6: zapamtiSto

* **Operacija:** saveTable (): signal
* **Veza sa SK:** SK8
* **Preduslov:** Vrednosna i strukturna ograničenja nad objektom Sto moraju biti zadovoljena
* **Postuslov:** Sto je zapamćen



#### Ugovor UG7: zapamtiRačun

* **Operacija:** saveBill (): signal
* **Veza sa SK:** SK9
* **Preduslov:** Vrednosna i strukturna ograničenja nad objektom Racun moraju biti zadovoljena
* **Postuslov:** Racun je zapamćen



#### Ugovor UG8: pretražiRačune

* **Operacija:** searchBill(String s): signal
* **Veza sa SK:** SK9
* **Preduslov:** Vrednosna i strukturna ograničenja nad objektom StavkaRacuna moraju biti zadovoljena
* **Postuslov:** Stavka računa je zapamćena



#### Ugovor UG9: obrisiRacun

* **Operacija:** deleteRacun(Racun r): signal
* **Veza sa SK:** SK10
* **Preduslov:** Strukturna ograničenja nad objektom Racun moraju biti zadovoljena
* **Postuslov:** Racun je obrisan



#### Ugovor UG10: obrisiStavkuRacuna

* **Operacija:** deleteStavkaRacuna(StavkaRacuna s): signal
* **Veza sa SK:** SK10
* **Preduslov:** Strukturna ograničenja nad objektom StavkaRacuna moraju biti zadovoljena
* **Postuslov:** StavkaRacuna je obrisana



### 3.3.5 Broker baze podataka

Klasa DBBroker predstavlja perzistentni okvir koji posreduje u svim operacijama nad bazom podataka i realizuje sledeće metode (otvaranje konekcije se izvršava u konstruktoru):

* public void openConnection()
* public void commitTransaction()
* public void closeConnection()
* public List<AbstractObject> getAllObjects(AbstractObject o)
* public AbstractObject saveOrUpdateObject(AbstractObject object)
* public void deleteObject(AbstractObject object)
* public List<AbstractObject> getObjectByPK (AbstractObject o, String ID, int tip)
* public List<AbstractObject> getObjectByFK(AbstractObject o, String ID)
* public AbstractObject saveObject(AbstractObject o)

Da bismo izbegli implementaciju pojedinačnih metoda za svaku domensku klasu, sve metode klase DBBroker su generičke. Rade sa AbstractObject i preko parametara dobijaju sve potrebne informacije.



## 3.4 Projektovanje skladištom podataka

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

|  |  |
| --- | --- |
| **Tabela: Konobari** | |
| *Ime polja* | *Tip* |
| konobarID | int |
| JMBG | varchar |
| ime | varchar |
| username | varchar |
| password | varchar |

|  |  |
| --- | --- |
| **Tabela: Sto** | |
| *Ime polja* | *Tip* |
| stoID | int |
| ukupanDnevniIznos | double |

|  |  |
| --- | --- |
| **Tabela: Pice** | |
| *Ime polja* | *Tip* |
| piceID | int |
| nazivPica | varchar |
| cena | double |

|  |  |
| --- | --- |
| **Tabela: Racun** | |
| *Ime polja* | *Tip* |
| racunID | int |
| iznos | double |
| ime | varchar |
| konobarID | int |
| stoID | int |

|  |  |
| --- | --- |
| **Tabela: StavkaRacuna** | |
| *Ime polja* | *Tip* |
| racunID | int |
| brStavkeRacuna | int |
| piceID | int |

# 4 Implementacija

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

Čitav sistem je realizovan u tri projekta: kaficClient, kaficServer i kaficCommon. Na osnovu arhitekture softverskog sistema dobijene su sledeće softverske klase:

KaficClient:

* communication/Communication
* controller/Controller
* model/BillsTableModel
* model/DrinksTableModel
* model/TablesTableModel
* model/StavkeTableModel
* view/BillsFrm
* view/DrinksFrm
* view/LoginFrm
* view/NewBillFrm
* view/NewDrinkFrm
* view/TablesFrm
* view/UserPageFrm

KaficCommon:

* constances/Constances
* domen/AbstractObject
* domen/Konobar
* domen/Sto
* domen/Racun
* domen/Pice
* domen/StavkaRacuna
* exception/ServerException
* exception/ValidationException
* transfer/ClientTransfer
* transfer/ServerTransfer

KaficServer:

* controller/ServerControllere
* dbbroker/DBBroker
* model/KonobariTableModel
* threads/Server
* threads/ClientThread
* view/KonobariFrm
* view/ServerFrm
* system\_operation/AbstractSO
* system\_operation/SODeleteWaiter
* system\_operation/SODeleteStavka
* system\_operation/SODeleteBill
* system\_operation/SODeleteTable
* system\_operation/SOLoadBills
* system\_operation/SOLoadDrinks
* system\_operation/SOLoadStavke
* system\_operation/SOLoadTables
* system\_operation/SOLoadWaiterList
* system\_operation/SOLogin
* system\_operation/SONewBill
* system\_operation/SONewDrink
* system\_operation/SONewTable
* system\_operation/SOSaveWaiter
* system\_operation/SOSearchBills
* system\_operation/SOSearchDrinks
* system\_operation/SOSearchWaiter

# 5 Testiranje

Svaki od implementiranih slučajeva korišćenja je testiran. Prilikom testiranja svakog slučaja korišćenja, pored unetih pravilnih podataka, unošeni su i nepravilni podaci da bi se utvrdilo kakav će rezultat izvršenja biti. Posmatrano je ponašanje sistema sa ciljem otklanjanja grešaka i poboljšanja kvaliteta sistema.