Универзитет у Београду

Факултет организационих наука

Семинарски рад из предмета

Пројектовање софтвера

Тема: Софтверски систем за праћење рада гејминг центра у Јава окружењу

Ментор: Студент:

Проф. др Синиша Влајић Кристина Цветиновић 382/17

Београд, 2020

Садржај:

[1. Вербални опис 1](#_Toc76551112)

[2. Случајеви коришћења 2](#_Toc76551113)

[СК1: Случај коришћења – Пријављивање запосленог у систем 3](#_Toc76551114)

[СК2: Случај коришћења – Унос новог клијента 4](#_Toc76551115)

[СК3: Случај коришћења – Брисање клијента 5](#_Toc76551116)

[СК4: Случај коришћења – Претраживање клијента 6](#_Toc76551117)

[СК5: Случај коришћења – Унос новог пакета 7](#_Toc76551118)

[СК6: Случај коришћења – Измена пакета 8](#_Toc76551119)

[СК7: Случај коришћења – Брисање пакета 9](#_Toc76551120)

[СК8: Случај коришћења – Пријава клијента на пакет 10](#_Toc76551121)

[СК9: Случај коришћења – Одјава клијента са пакета 12](#_Toc76551122)

[3. Aнализа 13](#_Toc76551123)

[3.1 Понашање софверског система-дијаграм секвенци 13](#_Toc76551124)

[3.2 Дефинисање уговора о системским операцијама 34](#_Toc76551125)

[3.3 Структура софтверског система: Концептуални (доменски) модел 36](#_Toc76551126)

[3.4 Структура софтверског система: Релациони модел 37](#_Toc76551127)

[4. Пројектовање 41](#_Toc76551128)

[3.1. Архитектура софтверског система 41](#_Toc76551129)

[3.2 Пројектовање корисничког интерфејса 41](#_Toc76551130)

[3.2.1. Пројектовање екранских форми 42](#_Toc76551131)

[3.3. Пројектовање контролера корисничког интерфејса 74](#_Toc76551132)

[3.4. Пројектовање апликационе логике 74](#_Toc76551133)

[3.3.1. Комуникација са клијентима 74](#_Toc76551134)

[3.3.2. Контролер апликационе логике 80](#_Toc76551135)

[3.3.3. Пројектовање структуре софтверског система 81](#_Toc76551136)

[3.3.4. Пројектовање системских операција 85](#_Toc76551137)

[3.3.5. Брокер базе података 93](#_Toc76551138)

[3.5 Пројектовање складишта података 94](#_Toc76551139)

[5. Имплементација 96](#_Toc76551140)

[6. Тестирање 100](#_Toc76551141)

[7. Литература 101](#_Toc76551142)

# Вербални опис

Потребно је креирати софтверски систем који ће омогућити ефикасно вођење гејминг центра, односно вођење евиденције о клијентима и пакетима.

Софтверски систем би требало да обезбеди лакше регистровање нових, претраживање и уклањање постојећих клијената из система, као и чување података о њима.

Осим тога, потребно је омогућити једноставно креирање нових пакета услуга који обухватају задужени уређај и тачан број сати његовог задужења и сврставају се у понуду гејминг центра. Треба обезбедити и претраживање и измену, као и уклањање пакета услуга из система.

Софтверским системом треба обезбедити вршење пријаве као и одјаве клијената са пакета услуга.

Спољни корисник, односно актор, који би водио ову евиденцију био би запослени у гејминг центру. Неопходно је да запослени буде пријављен на систем под својом шифром и корисничким именом и адекватно обучен да користи софтверски систем.

# Случајеви коришћења

Случајеви коришћења наведени на UML дијаграму су:

1. Пријава запосленог у систем

2. Унос новог клијента

3. Брисање клијента

4. Претрага клијента

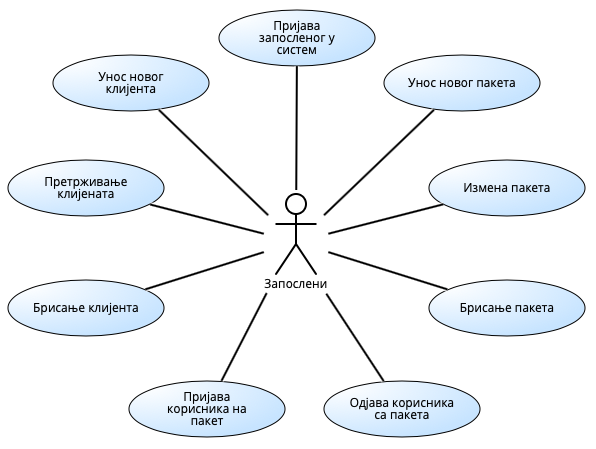
5. Унос новог пакета

6. Измена пакета

7. Брисање пакета

8. Пријава клијента на пакет

9. Одјава клијента са пакета



Слика : Случајеви коришћења

## СК1: Случај коришћења – Пријављивање запосленог у систем

**Назив СК**   
Пријављивање запосленог у систем

**Актори СК**   
Запослени

**Учесници СК**   
Запослени и систем (програм)

**Предуслов:** Систем је укључен и приказује форму за пријављивање запосленог.

**Основни сценарио СК**

1. Запослени **уноси** податке за пријаву запосленог. (АПУСО)
2. Запослени **контролише** да ли је коректно унео податке за пријаву запосленог. (АНСО)
3. Запослени **позива** систем да пронађе запосленог са задатим подацима. (АПСО)
4. Систем **претражује** запослене. (СО)
5. Систем **приказује** запосленом пронађеног запосленог и поруку: „Успешно сте се пријавили!“ и омогућава приступ систему. (ИА)

**Алтернативна сценација** 4.1 Уколико систем не може да пронађе запосленог приказује запосленом поруку: „Систем не може да пронађе запосленог на основу учитаних вредности.“ (ИА)

## СК2: Случај коришћења – Унос новог клијента

**Назив СК**

Креирање новог клијента

**Актори СК**

Запослени

**Учесници СК**

Запослени и систем (програм)

**Предуслов**: Систем је укључен и запослени je улогован под својом шифром. Систем приказује форму за рад са налогом клијента.

**Основни сценарио СК**

1. Запослени **уноси** податке о клијенту. (АПУСО)
2. Запослени **контролише** да ли је коректно унео податке у налог клијента. (АНСО)
3. Запослени **позива** систем да запамти податке о налогу клијента. (АПСО)
4. Систем **памти** податке о налогу клијента. (СО)
5. Систем **приказује** запосленом запамћени налог клијента и поруку: “Клијент успешно сачуван“. (ИА)

**Алтернативна сценарија**

5.1 Уколико систем не може да запамти податке о налогу клијента он приказује запосленом поруку “Систем не може да запамти   
налог клијента ”. (ИА)

## СК3: Случај коришћења – Брисање клијента

**Назив СК**

Брисање клијента

**Актори СК**

запослени

**Учесници СК**

запослени и систем (програм)

**Предуслов**: Систем је укључен и запослени je улогован под својом шифром. Учитана је листа налога клијената. Систем приказује форму за рад са налогом клијента.

**Основни сценарио СК**

1. Запослени **уноси** вредностпо којој претражује налоге клијената. (АПУСО)
2. Запослени **позива** систем да нађе налоге клијената по задатој вредности. (АПСО)
3. Систем **тражи** налоге клијената по задатој вредности. (СО)
4. Систем приказује запосленом налогe клијенaта и поруку: “Систем је нашао налогe клијенaта по задатој вредности”. (ИА)
5. Запослени **бира** налог клијента који жели да обрише. (АПУСО)
6. Запослени **позива** систем да обрише одабрани налог клијента. (АПСО)
7. Систем **брише** налог клијента. (СО)
8. Систем **приказује** запосленом поруку: “Систем је обрисао налог клијента.” (ИА)

**Алтернативна сценарија**

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

8.1 Уколико систем не може да обрише налог клијента он приказује запосленом поруку “Систем не може да обрише налог клијента ”. (ИА)

## СК4: Случај коришћења – Претраживање клијента

**Назив СК**

Претраживање клијента

**Актори СК**

запослени

**Учесници СК**

запослени и систем (програм)

**Предуслов**: Систем је укључен и запослени je улогован под својом шифром. Учитана је листа налога клијената. Систем приказује форму за рад са налогом клијента.

**Основни сценарио СК**

1. Запослени **уноси** вредностпо којој претражује налоге клијената. (АПУСО)
2. Запослени **позива** систем да нађе налоге клијената по задатој вредности. (АПСО)
3. Систем **тражи** налоге клијената по задатој вредности. (СО)
4. Систем приказује запосленом налоге клијената и поруку: “Систем је нашао налоге клијената по задатој вредности”. (ИА)
5. Запослени **бира** налог клијента који жели да учита. (АПУСО)
6. Запослени **позива** систем да учита податке о налогу клијента. (АПСО)
7. Систем **учитава** податке оодабраном налогу клијента. (СО)
8. Систем приказује запосленом податке о налогу клијента. (ИА)

**Алтернативна сценарија**

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

8.1 Уколико систем не може да учита податке о налогу клијента он приказује запосленом поруку: “Систем не може да учита налог клијента”.(ИА)

## СК5: Случај коришћења – Унос новог пакета

**Назив СК**

Креирање пакета

**Актори СК**

Запослени

**Учесници СК**

Запослени и систем (програм)

**Предуслов**: Систем је укључен и запослени je улогован под својом шифром. Систем приказује форму за рад са пакетом. Учитана је листа уређаја.

**Основни сценарио СК**

1. Запослени **уноси** податке о пакету. (АПУСО)
2. Запослени **контролише** да ли је коректно унео податке у пакет. (АНСО)
3. Запослени **позива** систем да запамти податке о пакету. (АПСО)
4. Систем **памти** податке о пакету. (СО)
5. Систем **приказује** запосленом запамћени пакет и поруку: “Систем је запамтио пакет“. (ИА)

**Алтернативна сценарија**

5.1 Уколико систем не може да запамти податке о пакету он приказује запосленом поруку “Систем не може да запамти пакет”. (ИА)

## СК6: Случај коришћења – Измена пакета

**Назив СК**

Промена пакета

**Актори СК**

Запослени

**Учесници СК**

Запослени и систем (програм)

**Предуслов**: Систем је укључен и запослени je улогован под својом шифром. Систем приказује форму за рад са пакетом. Учитана је листа пакета. Учитана је листа уређаја.

**Основни сценарио СК**

1. Запослени **уноси** вредностпо којој претражује пакете. (АПУСО)
2. Запослени **позива** систем да нађе пакете по задатој вредности. (АПСО)
3. Систем **тражи** пакете по задатој вредности. (СО)
4. Систем приказује запосленом пакете и поруку: “Систем је нашао пакете по задатој вредности”. (ИА)
5. Запослени **бира** пакет који жели да измени. (АПУСО)
6. Запослени **позива** систем да учита податке о пакету. (АПСО)
7. Систем **учитава** податке оодабраном пакету. (СО)
8. Систем приказује запосленом податке о пакету. (ИА)
9. Запослени **уноси** **(мења)** податке о пакету. (АПУСО)
10. Запослени **контролише** да ли је коректно унео податке о пакету. (АНСО)
11. Запослени **позива** систем да измени податке о пакету. (АПСО)
12. Систем **памти** податке о пакету. (СО)
13. Систем **приказује** запослеом запамћени пакет и поруку: “Систем је запамтио пакет.” (ИА)

**Алтернативна сценарија**

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

8.1 Уколико систем не може да учита податке о пакету он приказује запосленом поруку: “Систем не може да учита податке о пакету”. Прекида се извршење сценариа. (ИА)

13.1 Уколико систем не може да запамти податке о пакету он приказује запосленом поруку “Систем не може да запамти пакет”. (ИА)

## СК7: Случај коришћења – Брисање пакета

**Назив СК**

Брисање пакета

**Актори СК**

Запослени

**Учесници СК**

Запослени и систем (програм)

**Предуслов**: Систем је укључен и запослени je улогован под својом шифром. Систем приказује форму за рад са пакетом. Учитана је листа пакета.

**Основни сценарио СК**

1. Запослени **уноси** вредностпо којој претражује пакете. (АПУСО)
2. Запослени **позива** систем да нађе пакете по задатој вредности. (АПСО)
3. Систем **тражи** пакете по задатој вредности. (СО)
4. Систем приказује запосленом пакете и поруку: “Систем је нашао пакете по задатој вредности”. (ИА)
5. Запослени **бира** пакет који жели да обрише. (АПУСО)
6. Запослени **позива** систем да учита податке о пакету. (АПСО)
7. Систем **учитава** податке оодабраном пакету. (СО)
8. Систем приказује запосленом податке о пакету. (ИА)
9. Запослени **позива** систем да обрише пакет. (АПСО)
10. Систем **брише** пакет. (СО)
11. Систем **приказује** запосленом поруку: “Систем је обрисао пакет.” (ИА)

**Алтернативна сценарија**

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

8.1 Уколико систем не може да учита податке о пакету он приказује запосленом поруку: “Систем не може да учита податке о пакету”. Прекида се извршење сценариа. (ИА)

11.1 Уколико систем не може да обрише пакет он приказује запосленом поруку “Систем не може да обрише пакет”. (ИА)

## СК8: Случај коришћења – Пријава клијента на пакет

**Назив СК**

Пријава клијента на пакет

**Актори СК**

Запослени

**Учесници СК**

Запослени и систем (програм)

**Предуслов**: Систем је укључен и запослени је улогован под својом шифром. Учитана је листа пакета. Учитана је листа клијената. Клијент и пакет постоје у систему. Систем приказује форму за рад са пријавама на пакет.

**Основни сценарио СК**

1. Запослени **уноси** вредности по којима претражује клијенте. (АПУСО)
2. Запослени **позива** систем да нађе клијенте по задатој вредности. (АПСО)
3. Систем **тражи** клијенте по задатој вредности. (СО)
4. Систем **приказује** запосленом клијенте. (ИА)
5. Запослени **бира** клијента којег жели да пријави на пакет. (АПУСО)
6. Запослени **позива** систем да учита податке о клијенту којег жели да пријави на пакет. (АПСО)
7. Систем **учитава** податке о клијенту. (СО)
8. Систем **приказује** запосленом податке о клијенту. (ИА)
9. Запослени **уноси** вредности по којима претражује пакете. (АПУСО)
10. Запослени **позива** систем да нађе пакете по задатој вредности. (АПСО)
11. Систем **тражи** пакете по задатој вредности. (СО)
12. Систем **приказује** запосленом пакете и поруку: „Систем је нашао пакете по задатој вредности.“ (ИА)
13. Запослени **бира** пакет на који жели да пријави корисника. (АПУСО)
14. Запослени **позива** систем да учита податке о пакету на који жели да пријави корисника. (АПСО)
15. Систем **учитава** податке о пакету. (СО)
16. Систем **приказује** запосленом податке о пакету. (ИА)
17. Запослени **уноси** податке о пријави на пакет за који се клијент определио. (АПУСО)
18. Запослени **контролише** да ли је коректно унео податке о пријави на пакет. (АНСО)
19. Запослени **позива** систем да сачува пријаву на пакет за који се клијент определио. (АПСО)
20. Систем **чува** пријаву на пакет. (СО)
21. Систем **приказује** запосленом поруку: „Систем је сачувао додату пријаву на пакет“. (ИА)

**Алтернативна сценарија**

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

8.1 Уколико систем не може да учита податке о клијенту он приказује запосленом поруку: “Систем не може да учита податке о клијенту”. Прекида се извршење сценариа (ИА)

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

16.1 Уколико систем не може да учита податке о пакету он приказује запосленом поруку: “Систем не може да учита податке о пакету”. Прекида се извршење сценариа. (ИА)

21.1. Уколико систем не може да сачува нову пријаву на пакет он приказује запосленом поруку: „Систем не може да сачува додату пријаву на пакет “.(ИА)

## СК9: Случај коришћења – Одјава клијента са пакета

**Назив СК**

Одјава клијента са пакета

**Актори СК**

Запослени

**Учесници СК**

Запослени и систем (програм)

**Предуслов**: Систем је укључен и запослени је улогован под својом шифром. Учитана је листа пакета. Учитана је листа клијената. Клијент и пакет постоје у систему. Систем приказује форму за рад са одјавама.

**Основни сценарио СК**

1. Запослени **уноси** вредности по којима претражује пријаве на пакете. (АПУСО)
2. Запослени **позива** систем да нађе пријаве на пакете по задатој вредности. (АПСО)
3. Систем **тражи** пријаве на пакете по задатој вредности. (СО)
4. Систем **приказује** запосленом активне пријаве на пакете.(ИА)
5. Запослени **бира** пријаву на пакет са које жели да одјави клијента. (АПУСО)
6. Запослени **позива** систем да учита податке о пријави на пакет са које жели да одјави клијента. (АПСО)
7. Систем **учитава** податке о пријави на пакет. (СО)
8. Систем **приказује** запосленом податке о пријави на пакет.(ИА)
9. Запослени **позива** систем да одјави клијента са пакета. (АПСО)
10. Систем **чува** податке о одјави клијента са пакета. (СО)
11. Систем **приказује** запосленом поруку: „Одјава клијента је успешно евидентирана.“ (ИА)

**Алтернативна сценарија**

4.1 Уколико систем не може да нађе пријавe на пакетe по задатој вредности он приказује запосленом поруку: „Систем не може да нађе пријавe по задатој вредности“. Прекида се извршење сценариа. (ИА)

8.1 Уколико систем не може да учита податке о пријави на пакет он приказује запосленом поруку: „Систем не може да учита податке о пријави на пакет“. Прекида се извршење сценариа. (ИА)

11.1 Уколико систем не може да сачува податке о одјави клијента са пакета он приказује запосленом поруку: „Систем не може да сачува податке о одјави клијента са пакета “. (ИА)

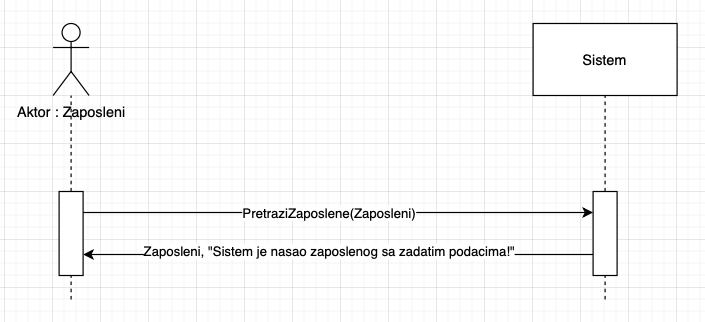
# Aнализа

## 3.1 Понашање софверског система-дијаграм секвенци

**ДС1: Дијаграми секвенци случаја коришћења: Пријава запосленог у систем**

*Oсновни сценарио:*

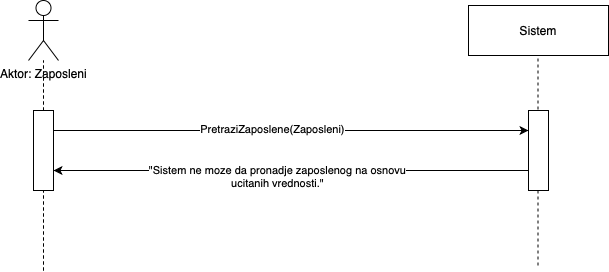
1. Запослени **позива** систем да пронађе запосленог са задатим подацима. (АПСО)
2. Систем **приказује** запосленом пронађеног запосленог и поруку: „Успешно сте се пријавили!“ и омогућава приступ систему. (ИА)

**

Слика 2: Основни сценарио пријаве запосленог на систем

*Алтернативна сценарија:*

2.1 Уколико систем не може да пронађе запосленог приказује запосленом поруку: „Систем не може да пронађе запосленог на основу учитаних вредности.“ (ИА)



Слика 3: Алтернативни сценарио пријаве запосленог на систем

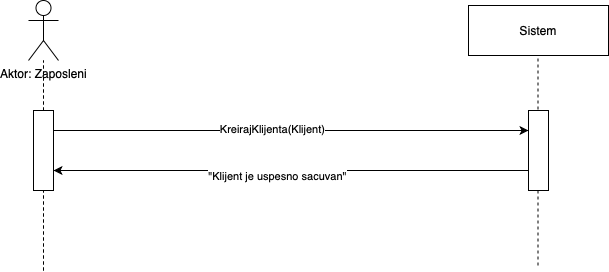
Уочена је једна системска операција:

1. *signal* **ZaposleniLogin**(*Zaposleni*);

**ДС2: Дијаграми секвенци случаја коришћења: Унос новог клијента**

*Oсновни сценарио:*

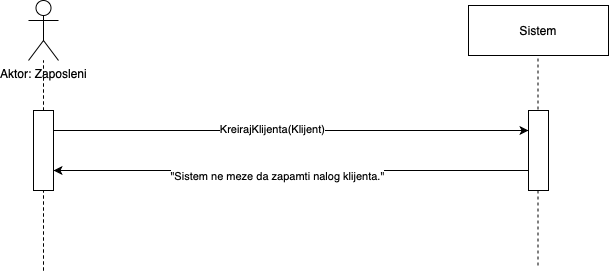
1. Запослени **позива** систем да запамти податке о налогу клијента. (АПСО)
2. Систем **приказује** запосленом запамћени налог клијента и поруку: “Клијент успешно сачуван“. (ИА)



Слика : Основни сценарио креирања клијента

*Алтернативна сценарија:*

2.1 Уколико систем не може да запамти податке о налогу клијента он приказује запосленом поруку “Систем не може да запамти   
налог клијента ”. (ИА)



Слика : Алтернативни сценарио креирања клијента

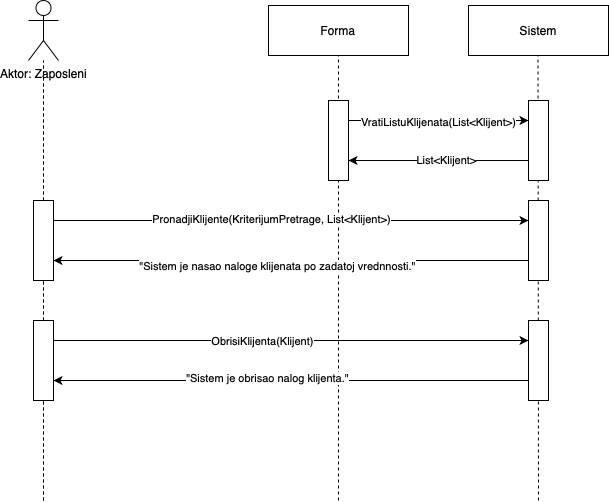
Уочена је једна системска операција:

1. *signal* **UnesiKlijenta** (*Klijent*);

**ДС3: Дијаграми секвенци случаја коришћења: Брисање клијента**

*Oсновни сценарио:*

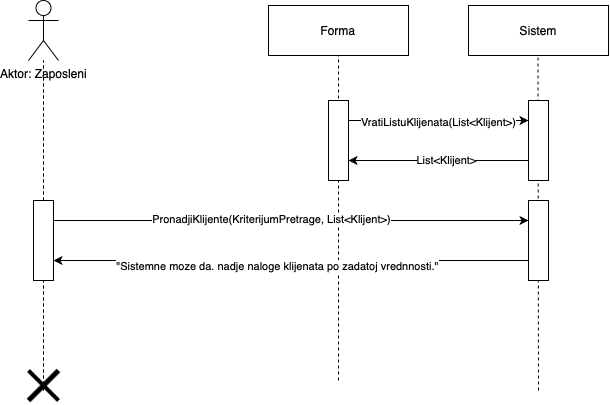
1. Форма позива систем да пронађе листу клијената. (АПСО)
2. Систем приказује на фроми листу клијената. (ИА)
3. Запослени **позива** систем да нађе налоге клијената по задатој вредности. (АПСО)
4. Систем приказује запосленом налогe клијенaта и поруку: “Систем је нашао налогe клијенaта по задатој вредности”. (ИА)
5. Запослени **позива** систем да обрише одабрани налог клијента. (АПСО)
6. Систем **приказује** запосленом поруку: “Систем је обрисао налог клијента.” (ИА)



Слика : Основни сценарио брисања клијента

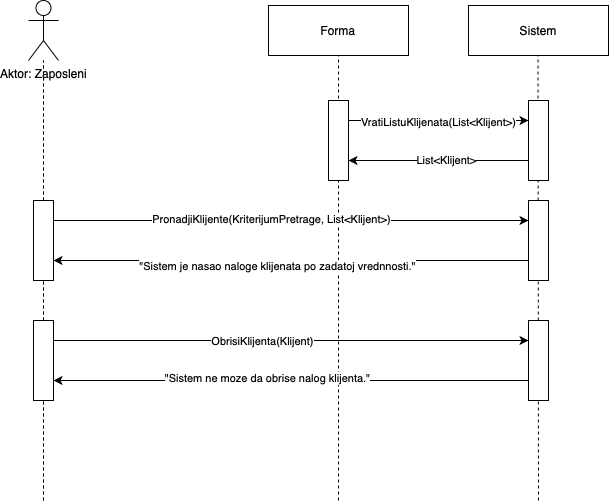
*Алтернативна сценарија:*

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



Слика : Алтернативни сценарио брисања клијента

6.1 Уколико систем не може да обрише налог клијента он приказује запосленом поруку “Систем не може да обрише налог клијента ”. (ИА)



Слика : Алтернатвни сценарио брисања клијента

Уоченe су три системске операције:

1. *signal* **VratiListuKlijenata**(*List<Klijent>*);

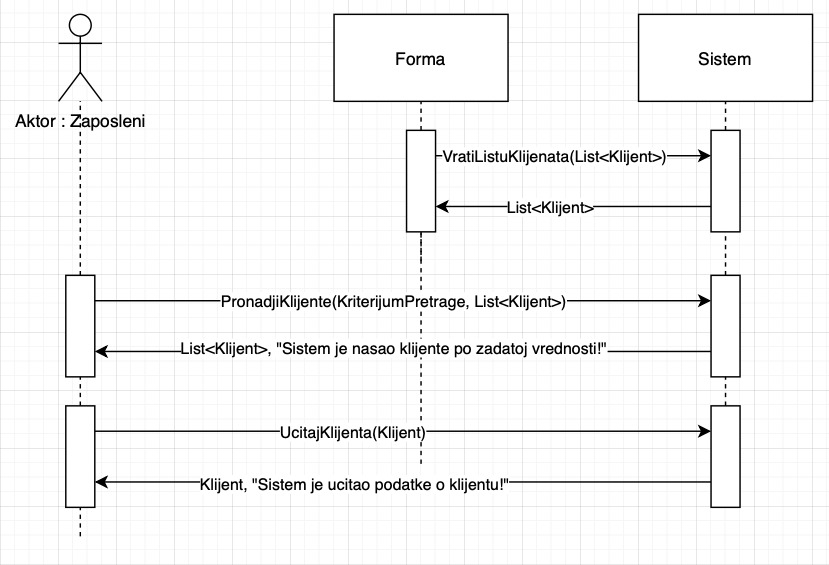
2. *signal* **PronadjiKljentе**(*KriterijumPretrage, List<Klijent>*);

3. *signal* **ObrisiKlijenta**(*Klijent*);

**ДС4: Дијаграми секвенци случаја коришћења: Претрага клијента**

*Oсновни сценарио:*

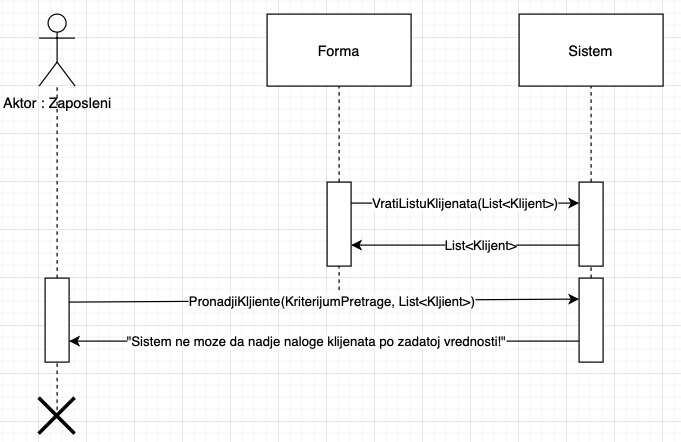
1. Форма позива систем да пронађе листу клијената. (АПСО)
2. Систем приказује на фроми листу клијената. (ИА)
3. Запослени **позива** систем да нађе налоге клијената по задатој вредности. (АПСО)
4. Систем приказује запосленом налоге клијената и поруку: “Систем је нашао налоге клијената по задатој вредности”. (ИА)
5. Запослени **позива** систем да учита податке о налогу клијента. (АПСО)
6. Систем приказује запосленом податке о налогу клијента. (ИА)



Слика :Основни сценарио претраге клијента

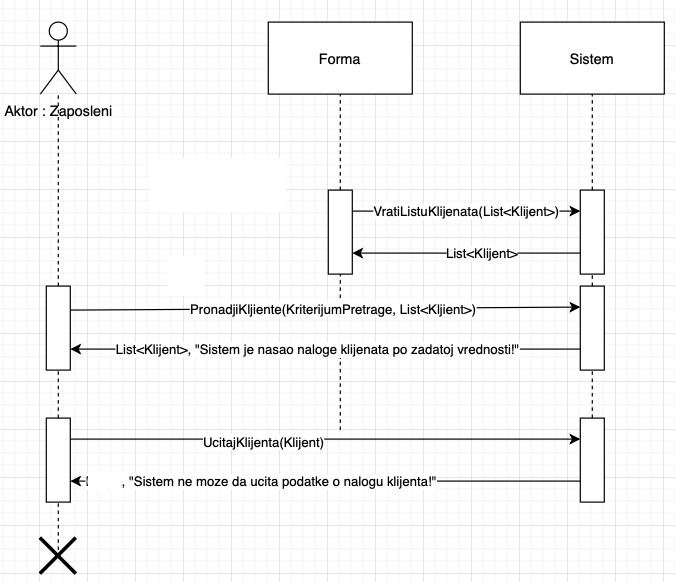
*Алтернативна сценарија:*

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



Слика : Алтернативни сценарио претраге клијента

6.1 Уколико систем не може да учита податке о налогу клијента он приказује запосленом поруку: “Систем не може да учита налог клијента”.(ИА)

****

Слика : Алтернативни сценарио претраге клијента

Уоченe су три системске операције:

1. *signal* **VratiListuKlijenata**(*List<Klijent>*);

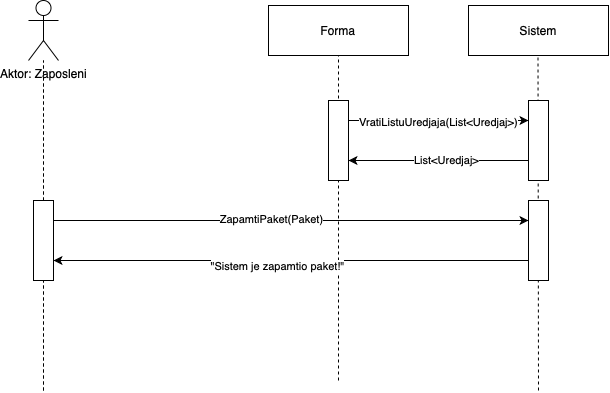
2. *signal* **PronadjiKljente**(*KriterijumPretrage, List<Klijent>*);

3. *signal* **UcitajKlijenta**(*Klijent*);

**ДС5: Дијаграми секвенци случаја коришћења: Унос новог пакета**

*Oсновни сценарио:*

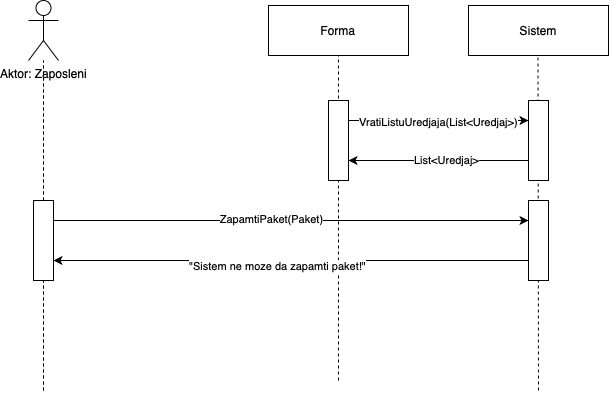
1. Форма позива систем да пронађе листу уређаја. (АПСО)
2. Систем приказује на форми листу уређаја. (ИА)
3. Запослени **позива** систем да запамти податке о пакету. (АПСО)
4. Систем **приказује** запосленом запамћени пакет и поруку: “Систем је запамтио пакет“. (ИА)



Слика : Основни сценарио уноса новог пакета

*Алтернативна сценарија:*

4.1 Уколико систем не може да запамти податке о пакету он приказује запосленом поруку “Систем не може да запамти пакет”. (ИА)



Слика : Алтернативни сценарио уноса новог пакета

Уоченe су две системске операције:

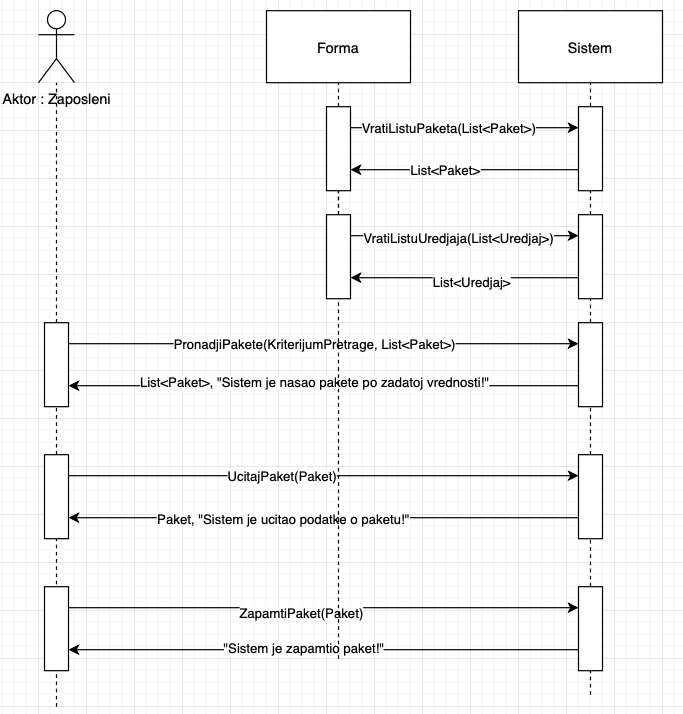
1. *signal* **VratiListuUredjaja**(*List<Uredjaj>*);

2. *signal* **ZapamtiPaket**(*Paket*);

**ДС6: Дијаграми секвенци случаја коришћења: Измена пакета**

*Oсновни сценарио:*

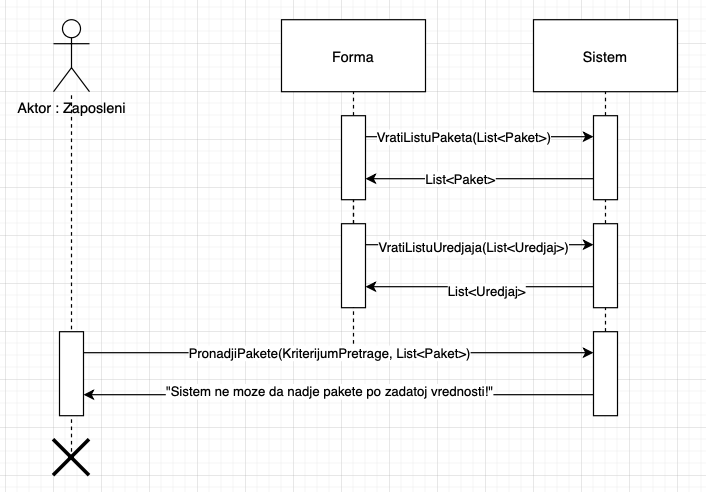
1. Форма позива систем да пронађе листу пакета. (АПСО)
2. Систем приказује на фроми листу клијената.(ИА)
3. Форма позива систем да пронађе листу уређаја. (АПСО)
4. Систем приказује на фроми листу уређаја.(ИА)
5. Запослени **позива** систем да нађе пакете по задатој вредности. (АПСО)
6. Систем приказује запосленом пакете и поруку: “Систем је нашао пакете по задатој вредности”. (ИА)
7. Запослени **позива** систем да учита податке о пакету. (АПСО)
8. Систем приказује запосленом податке о пакету и поруку: “Систем је учитао податке о пакету”. (ИА)
9. Запослени **позива** систем да измени податке о пакету. (АПСО)
10. Систем **приказује** запослеом запамћени пакет и поруку: “Систем је запамтио пакет.” (ИА)



Слика : Основни сценарио измене пакета

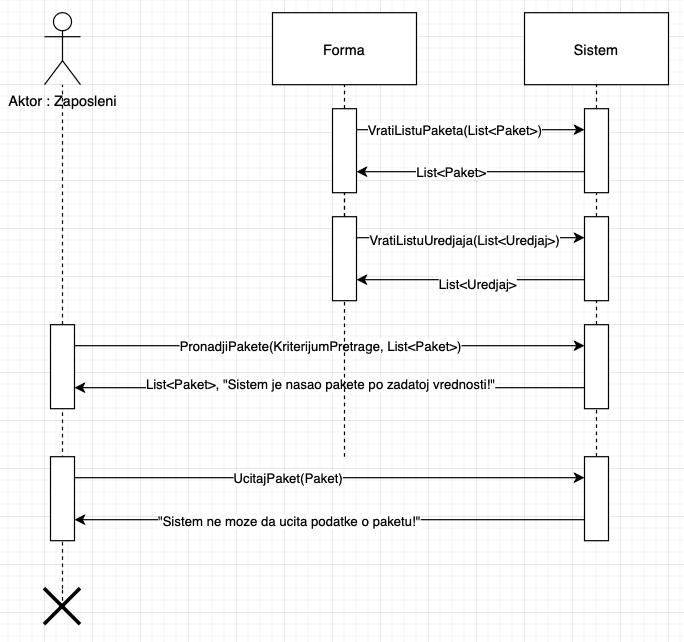
*Алтернативна сценарија:*

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



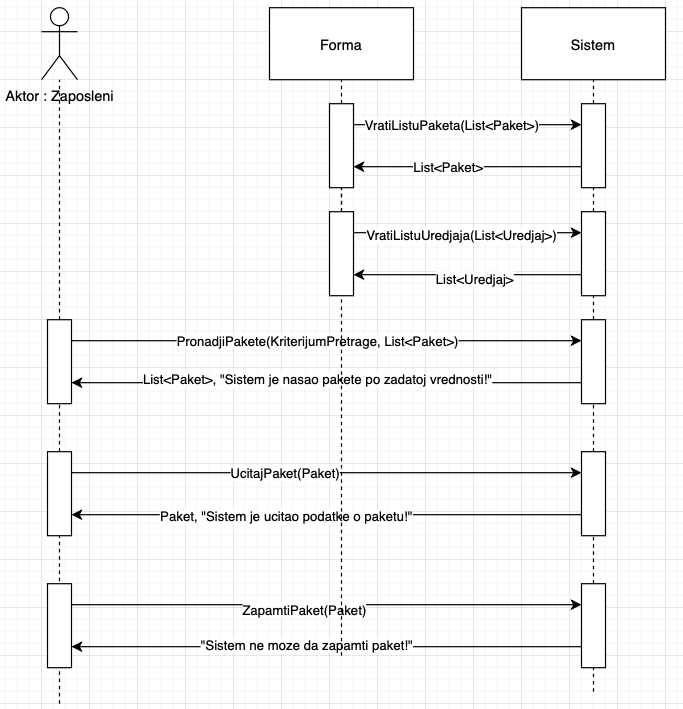
Слика : Алтернативни сценарио измене пакета

8.1 Уколико систем не може да учита податке о пакету он приказује запосленом поруку: “Систем не може да учита податке о пакету”. Прекида се извршење сценариа. (ИА)



Слика : Алтернативни сценарио измене пакета

10.1 Уколико систем не може да запамти податке о пакету он приказује запосленом поруку “Систем не може да запамти пакет”. (ИА)



Слика : Алтернативни сценарио измене пакета

Уочено је пет системских операција:

1. *signal* **VratiListuPaketa**(*List<Paket>*);

2. *signal* **VratiListuUredjaja**(*List<Uredjaj>*);

3. *signal* **PronadjiPaketе**(*KriterijumPretrage, List<Paket>*);

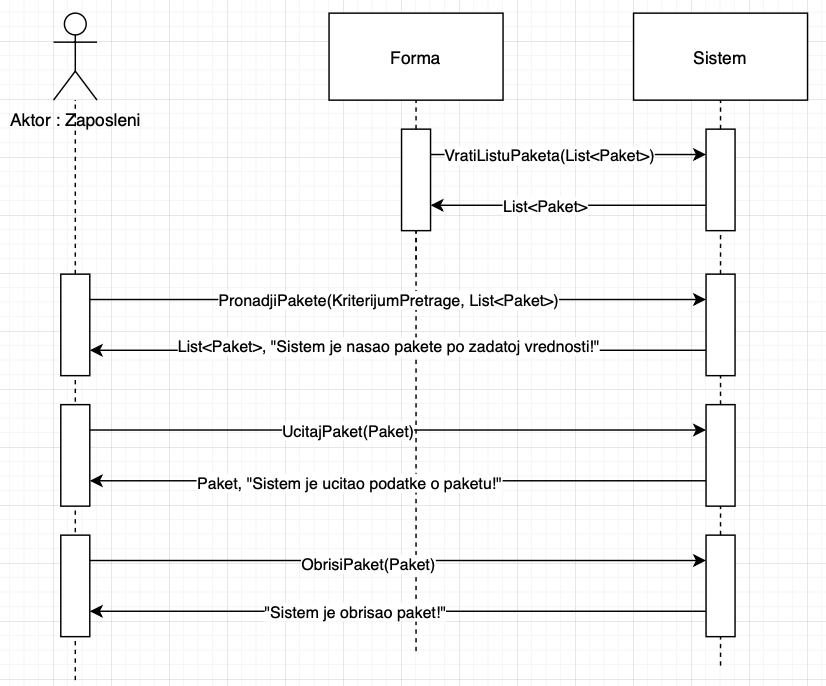
4. *siganl* **UcitajPaket**(*Paket*);

5. *signal* **IzmeniPaket**(*Paket*);

**ДС7: Дијаграми секвенци случаја коришћења: Брисање пакета**

*Oсновни сценарио:*

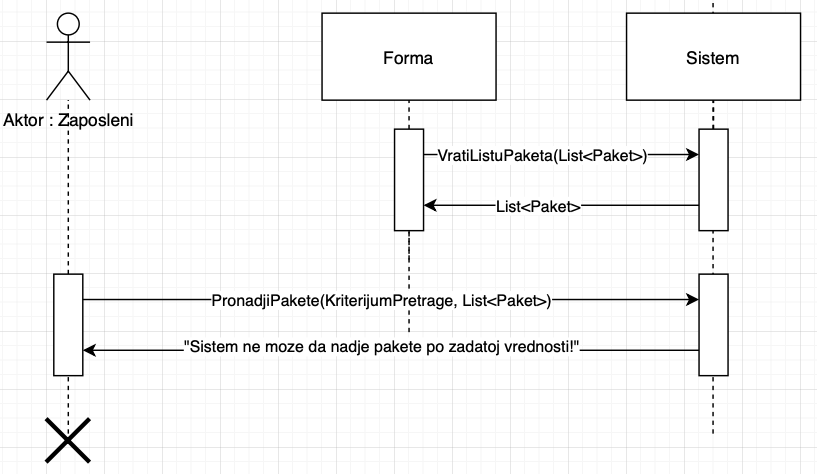
1. Форма позива систем да пронађе листу пакета. (АПСО)
2. Систем приказује на фроми листу пакета.(ИА)
3. Запослени **позива** систем да нађе пакете по задатој вредности. (АПСО)
4. Систем приказује запосленом пакете и поруку: “Систем је нашао пакете по задатој вредности”. (ИА)
5. Запослени **позива** систем да учита податке о пакету. (АПСО)
6. Систем приказује запосленом податке о пакету (ИА)
7. Запослени **позива** систем да обрише пакет. (АПСО)
8. Систем **приказује** запосленом поруку: “Систем је обрисао пакет.” (ИА)



Слика : Основни сценарио брисања пакета

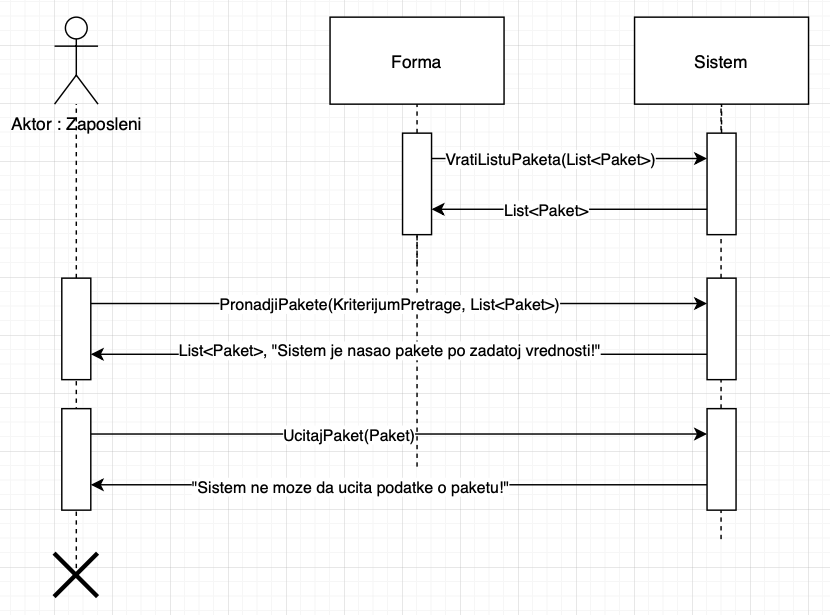
*Алтернативна сценарија:*

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



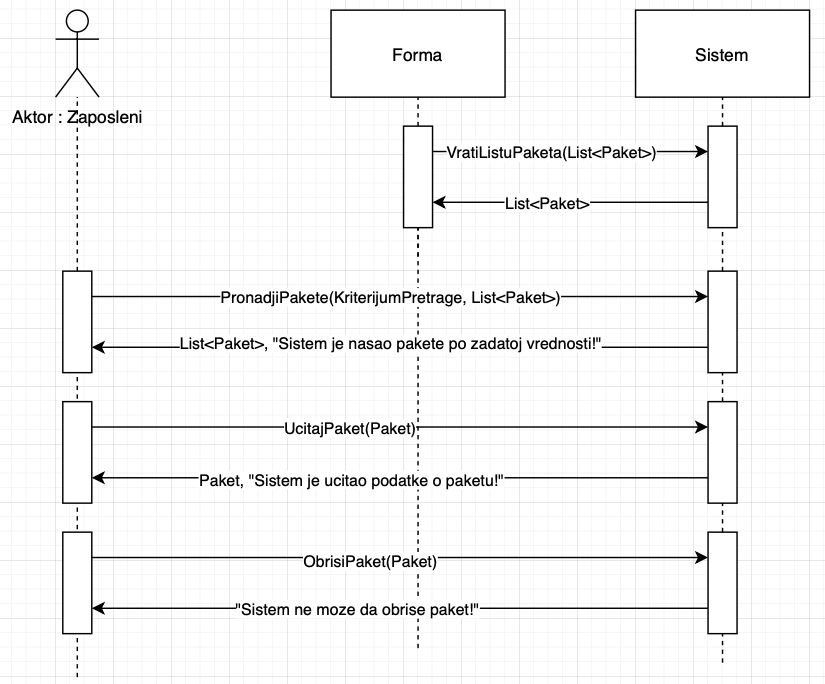
Слика : Алтернативни сценарио брисања пакета

6.1 Уколико систем не може да учита податке о пакету он приказује запосленом поруку: “Систем не може да учита податке о пакету”. Прекида се извршење сценариа. (ИА)



Слика : Алтернативни сценарио брисања пакета

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



Слика : : Алтернативни сценарио брисања пакета

Уоченe су четири системске операције:

1. *signal* **VratiListuPaketa**(*List<Paket>*);

2. *signal* **PronadjiPaketе**(*KriterijumPretrage, List<Paket>*);

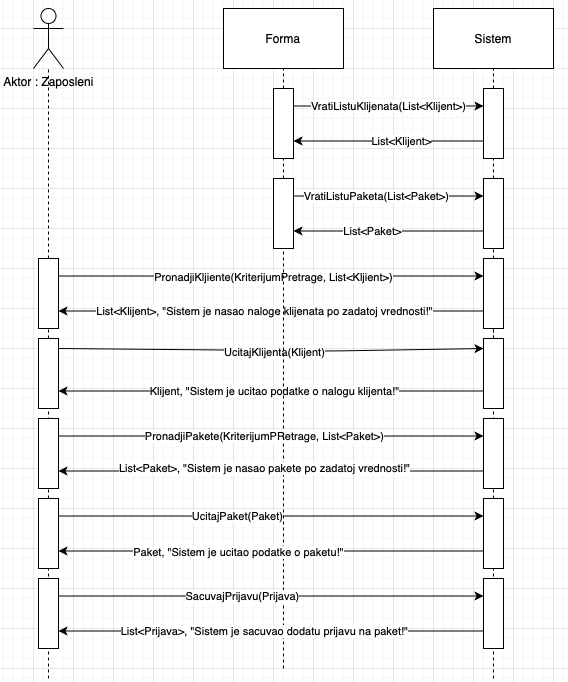
3. *siganl* **UcitajPaket**(*Paket*);

4. *signal* **ObrisiPaket**(*Paket*);

**ДС8: Дијаграми секвенци случаја коришћења: Пријава клијента на пакет**

*Oсновни сценарио:*

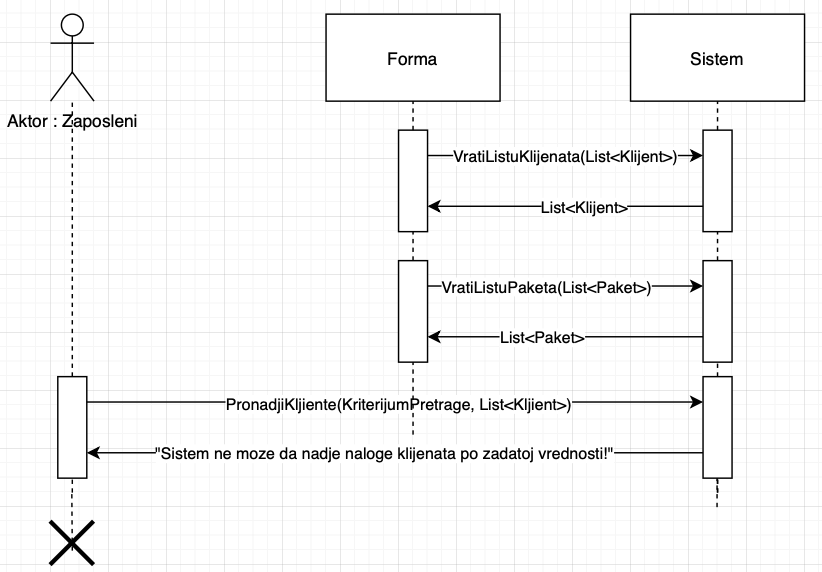
1. Форма позива систем да пронађе листу клијената.(АПСО)
2. Систем приказује на форми листу клијената.(ИА)
3. Форма позива ситем да пронађе листу пакета.(АПСО)
4. Систем приказује на форми листу пакета.(ИА)
5. Запослени **позива** систем да нађе клијенте по задатој вредности. (АПСО)
6. Систем **приказује** запосленом клијенте. (ИА)
7. Запослени **позива** систем да учита податке о клијенту којег жели да пријави на пакет. (АПСО)
8. Систем **приказује** запосленом податке о клијенту. (ИА)
9. Запослени **позива** систем да нађе пакете по задатој вредности. (АПСО)
10. Систем **приказује** запосленом пакете и поруку: „Систем је нашао пакете по задатој вредности.“ (ИА)
11. Запослени **позива** систем да учита податке о пакету на који жели да пријави корисника. (АПСО)
12. Систем **приказује** запосленом податке о пакету. (ИА)
13. Запослени **позива** систем да сачува пријаву на пакет за који се клијент определио. (АПСО)
14. Систем **приказује** запосленом поруку: „Систем је сачувао додату пријаву на пакет“. (ИА)



Слика : : Основни сценарио пријаве клијента на пакет

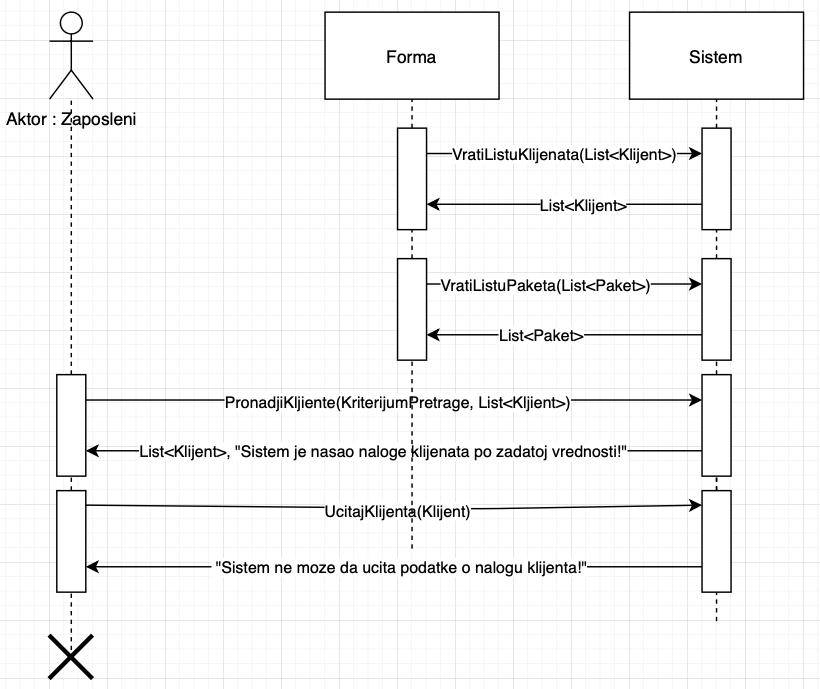
*Алтернативна сценарија:*

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



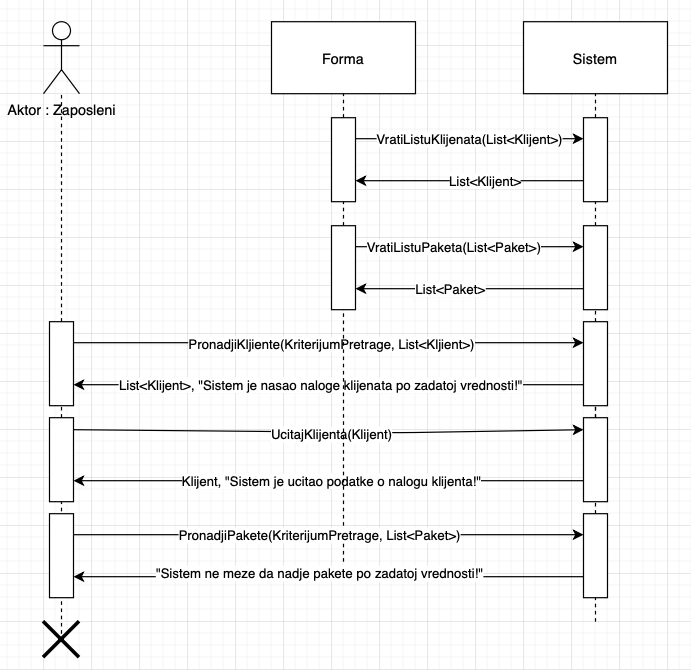
Слика : Алтернативни сценарио пријаве клијента на пакет

8.1 Уколико систем не може да учита податке о кориснику он приказује запосленом поруку: “Систем не може да учита податке о налогу клијента”. Прекида се извршење сценариа (ИА)



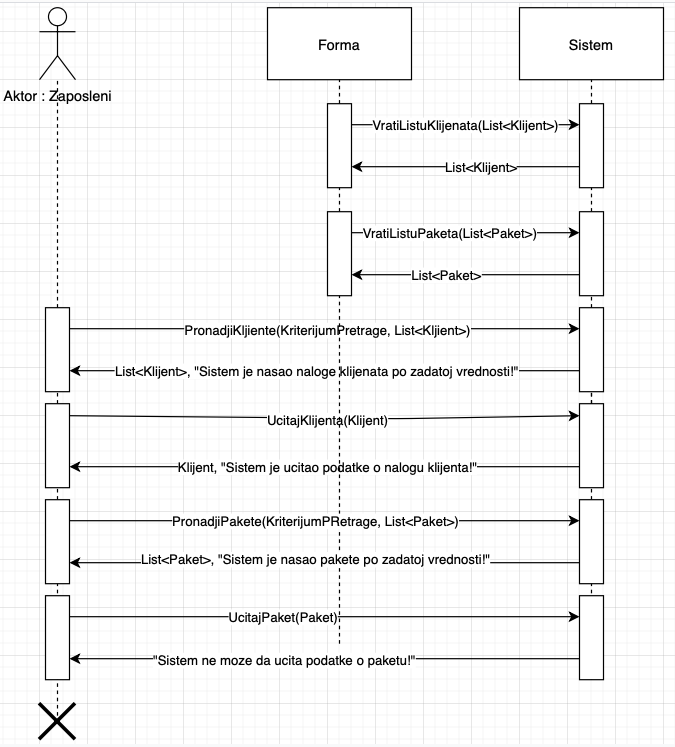
Слика : Алтернативни сценарио пријаве клијента на пакет

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



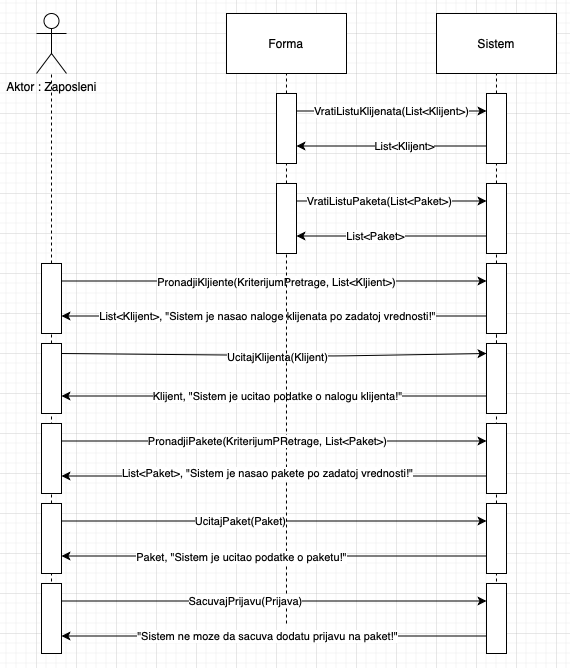
Слика : Алтернативни сценарио пријаве клијента на пакет

12.1 Уколико систем не може да учита податке о пакету он приказује запосленом поруку: “Систем не може да учита податке о пакету”. Прекида се извршење сценариа. (ИА)



Слика : Алтернативни сценарио пријаве клијента на пакет

14.1. Уколико систем не може да сачува нову пријаву на пакет он приказује запосленом поруку: „Систем не може да сачува додату пријаву на пакет “.(ИА)



Слика : Алтернативни сценарио пријаве клијента на пакет

Уочено је седам системских операција:

1. *signal* **VratiListuKlijenata**(*List<Klijent>*);

2. *signal* **VratiListuPaketa**(*List<Paket>*);

3. *signal* **PronadjiKljentе**(*KriterijumPretrage, List<Klijent>*);

4. *signal* **UcitajKlijenta**(*Klijent*);

5. *signal* **PronadjiPaketе**(*KriterijumPretrage, List<Paket>*);

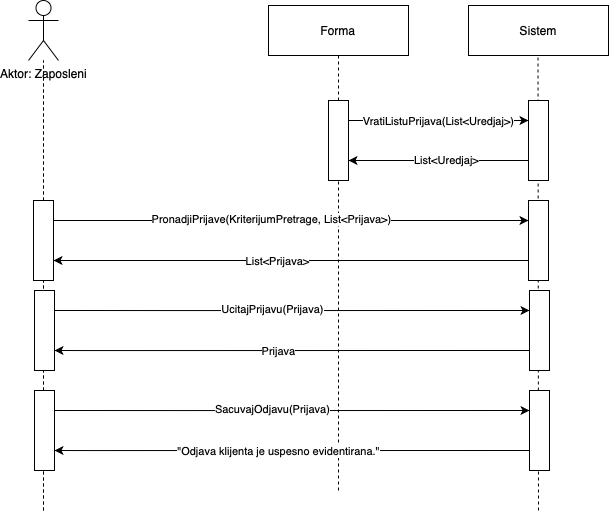
6. *siganl* **UcitajPaket**(*Paket*);

7. *signal* **SacuvajPrijavu**(*Prijava*);

**ДС9: Дијаграми секвенци случаја коришћења: Одјава клијента са пакета**

*Oсновни сценарио:*

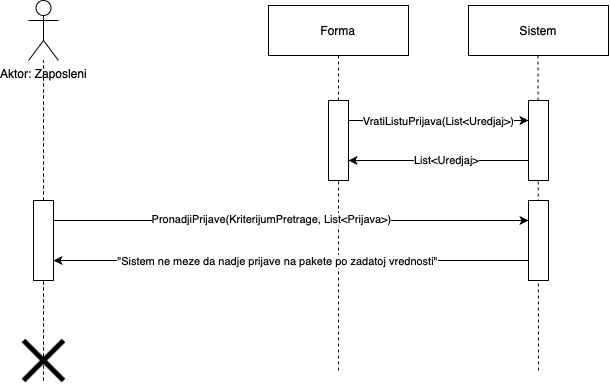
1. Форма позива систем да пронађе листу пријава.(АПСО)
2. Систем приказује на форми листу пријава.(ИА)
3. Запослени **позива** систем да нађе пријаве на пакете по задатој вредности. (АПСО)
4. Систем **приказује** запосленом активне пријаве на пакете.(ИА)
5. Запослени **позива** систем да учита податке о пријави на пакет са које жели да одјави клијента. (АПСО)
6. Систем **приказује** запосленом податке о пријави на пакет.(ИА)
7. Запослени **позива** систем да одјави клијента са пакета. (АПСО)
8. Систем **приказује** запосленом поруку: „Одјава клијента је успешно евидентирана.“ (ИА)



Слика : Основни сценарио одјаве клијента са пакета

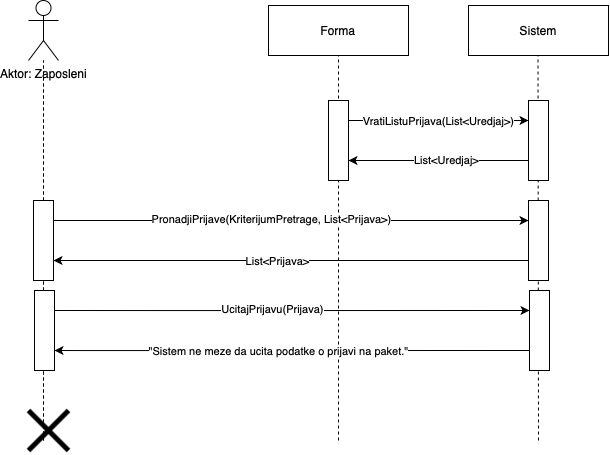
*Алтернативна сценарија:*

4.1 Уколико систем не може да нађе пријавe на пакетe по задатој вредности он приказује запосленом поруку: „Систем не може да нађе пријавe на пакетe по задатој вредности“. Прекида се извршење сценариа. (ИА)



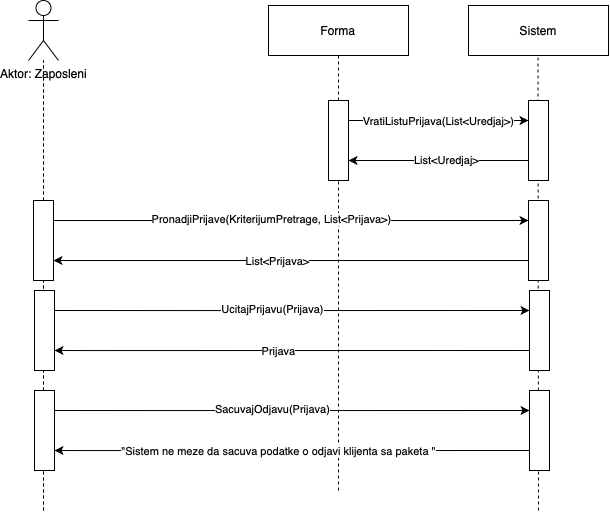
Слика : Алтернативни сценарио одјаве корисника са пакета

6.1 Уколико систем не може да учита податке о пријави на пакет он приказује запосленом поруку: „Систем не може да учита податке о пријави на пакет“. Прекида се извршење сценариа. (ИА)



Слика : Алтернативни сценарио одјаве корисника са пакета

8.1 Уколико систем не може да сачува податке о одјави клијента са пакета он приказује запосленом поруку: „Систем не може да сачува податке о одјави клијента са пакета “. (ИА)



Слика : Алтернативни сценарио одјаве корисника са пакета

Уочене су три системске операције:

1. *signal* **VratiListuPrijava**(*List<Prijava>*);

2. *signal* **PronadjiPrijave**(*KriterijumPretrage, List<Prijava>)*;

3. *signal* **SacuvajOdjavu**(*Prijava*);

**Као резултат анализе система уочено је седамнаест системских операција:**

1. *signal* **ZaposleniLogin**(*Zaposleni*);

2. *signal* **UnesiKlijenta** (*Klijent*);

3. *signal* **PronadjiKljentе**(*KriterijumPretrage, List<Klijent>*);

4. *signal* **UcitajKlijenta**(*Klijent*);

5. *signal* **ObrisiKlijenta**(*Klijent*);

6. *signal* **ZapamtiPaket**(*Paket*);

7. *signal* **PronadjiPaketе**(*KriterijumPretrage, List<Paket>*);

8. *siganl* **UcitajPaket**(*Paket*);

9. *signal* **ObrisiPaket**(*Paket*);

10. *signal* **SacuvajPrijavu**(*Prijava*);

11. *signal* **PronadjiPrijave**(*KriterijumPretrage, List<Prijava>)*;

12. *signal* **IzmeniPaket**(*Paket*) ;

13. *signal* **SacuvajOdjavu**(*Prijava*);

14. *signal* **VratiListuPaketa**(*List<Paket>*);

15. *signal* **VratiListuKlijenata**(*List<Klijent>*);

16. *signal* **VratiListuUredjaja**(*List<Uredjaj>*);

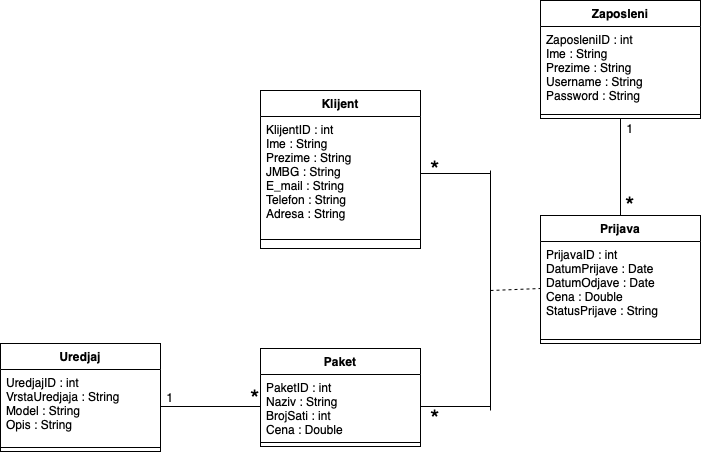
17. *signal* **VratiListuPrijava**(*List<Prijava>*);

## 3.2 Дефинисање уговора о системским операцијама

1. Уговор УГ1: *ZaposleniLogin*  
   Операција: **ZaposleniLogin** (*Zaposleni*) : *signal*;  
   Веза са СК: СК1  
   Предуслови: /  
   Постуслови: /
2. Уговор УГ2: *ZapamtiKlijenta*  
   Операција: **UnesiKlijenta**(*Klijent*) : *signal*;  
   Веза са СК: СК2  
   Предуслови:  
    -Вредносна ограничења над објектом *Klijent* морају бити задовољена.  
    -Структурна ограничења над објектом *Klijent* морају бити задовољена.  
   Постуслови:  
    -Запамћени су подаци о клијенту.
3. Уговор УГ3: *PronadjiKljente*  
   Операција:*signal* **PronadjiKljentе**(*KriterijumPretrage, List<Klijent>*);  
   Веза са СК: СК3, СК4, СК8, СК9  
   Предуслови: /  
   Постуслови: /
4. Уговор УГ4: *UcitajKlijenta*  
   Операција: **UcitajKlijenta**(*Klijent*) : *signal*;  
   Веза са СК: СК3, СК4, СК8, СК9  
   Предуслови: /  
   Постуслови: /
5. Уговор УГ5: *ObrisiKlijenta*  
   Операција: **ObrisiKlijenta**(*Klijent*) : *signal*;  
   Веза са СК: СК3  
   Предуслови:   
    -Структурна ограничења над објектом *Klijent* морају бити задовољена.  
   Постуслови:   
    -Клијент је обрисан.
6. Уговор УГ6: *ZapamtiPaket*  
   Операција: **ZapamtiPaket**(*Paket)* : *signal*;  
   Веза са СК: СК5, СК6  
   Предуслови:  
    -Вредносна ограничења над објектом *Paket* морају бити задовољена.  
    -Структурна ограничења над објектом *Paket* морају бити задовољена.  
   Постуслови:  
    -Сачувани су подаци о пакету.
7. Уговор УГ7: *PronadjiPaket*  
   Операција: **PronadjiPaketе**(*KriterijumPretrage, List<Paket>*) : *signal*;  
   Веза са СК: СК6, СК7, СК8  
   Предуслови: /  
   Постуслови: /
8. Уговор УГ8: *UcitajPaket*  
   Операција: **UcitajPaket**(*Paket*) : *signal*;  
   Веза са СК: СК6, СК7, СК8  
   Предуслови: /  
   Постуслови: /
9. Уговор УГ9: *ObrisiPaket*  
   Операција: **ObrisiPaket**(*Paket*) : *signal*;  
   Веза са СК: СК7  
   Предуслови:   
    -Структурна ограничења над објектом *Paket* морају бити задовољена.  
   Постуслови:   
    -Пакет је обрисан.
10. Уговор УГ10: *SacuvajPrijavu*  
    Операција: **SacuvajPrijavu**(*Prijava*) : *signal*;  
    Веза са СК: СК8  
    Предуслови:  
     -Вредносна ограничења над објектом *Prijava* морају бити задовољена.  
     -Структурна ограничења над објектом *Prijava* морају бити задовољена.   
    Постуслови:  
     -Сачувани су подаци о новој пријави.
11. Уговор УГ11: *PronadjiPrijavu*  
    Операција: **PronadjiPrijave**(*KriterijumPretrage, List<Prijava>)*: *signal*;  
    Веза са СК: СК9  
    Предуслови: /  
    Постуслови: /
12. **Уговор УГ12: *IzmeniPaket***  
    Операција: **IzmeniPaket**(*Paket*) : *signal*;  
    Веза са СК: СК6, СК7, СК8  
    Предуслови:   
    -Структурна ограничења над објектом *Paket* морају бити задовољена.  
    Постуслови:   
    -Пакет је измењен.
13. Уговор УГ13: *SacuvajOdjavu*  
    Операција: **SacuvajOdjavu**(*Prijava*) : *signal*;  
    Веза са СК: СК9  
    Предуслови:  
     -Вредносна ограничења над објектом *Prijava* морају бити задовољена.  
     -Структурна ограничења над објектом *Prijava* морају бити задовољена.   
    Постуслови:  
     -Подаци о одјави су сачувани.
14. Уговор УГ14: *VratiListuPaketa*  
     Операција: **VratiListuPaketa**(*List<Paket>*): *signal*;  
    Веза са СК: СК6, СК7, СК8, СК9  
    Предуслови: /  
    Постуслови: /
15. Уговор УГ15:*VratiListuKlijenata*  
    Операција: **VratiListuKlijenata**(*List<Klijent>*): *signal*;  
    Веза са СК: СК3, СК4, СК8, СК9  
    Предуслови: /  
    Постуслови: /
16. Уговор УГ16:*VratiListuUredjaja*  
    Операција: **VratiListuUredjaja**(*List<Uredjaj>*): *signal*;  
    Веза са СК: СК5, СК6  
    Предуслови: /  
    Постуслови: /
17. Уговор УГ17: *VratiListuPrijava*

Операција: **VratiListuPrijava**(*List<Prijava>*): *signal*;  
Веза са СК: СК9   
Предуслови: /  
Постуслови: /

## 3.3 Структура софтверског система: Концептуални (доменски) модел



Слика : Концептуални модел софтверског система

## 3.4 Структура софтверског система: Релациони модел

Zaposleni ( ZaposleniID, Ime, Prezime, Username, Password )

Klijent(KlijentID, Ime, Prezime, JMBG, E\_mail, Telefon, Adresa)

Paket(PaketID, Naziv, BrojSati, Cena, *UredjajID*)

Uredjaj(UredjajID, Opis, VrstaUredjaja, Model)

Prijava(PrijavaID, PaketID, KlijentID, DatumPrijave, DatumOdjave, Cena, StatusPrijave, *ZaposleniID*)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Табела:  Zaposleni | | Просто Вредносно Ограничење | | Сложено Вредносно Ограничење | | Структурно ограничење |
| Атрибути | Име | Тип | Вредност атрибута | Међузав. атрибута једне табеле | Међузав. атрибута више табела | INSERT /  UPDATE CASCADES Prijava  DELETE RESTRICTED Prijava |
| ZaposleniID | Integer | Not null |  |  |
| Ime | String | Not null |  |  |
| Prezime | String | Not null |  |  |
| Username | String | Not null |  |  |
| Password | String | Not null |  |  |

Табела : Запослени

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Табела:  Klijent | | Просто Вредносно Ограничење | | Сложено Вредносно Ограничење | | Структурно ограничење |
| Атрибути | Име | Тип | Вредност атрибута | Међузав. атрибута једне табеле | Међузав. атрибута више табела | INSERT /  UPDATE CASCADES Prijava  DELETE RESTRICTED Prijava |
| KlijentID | Integer | Not null |  |  |
| Ime | String | Not null |  |  |
| Prezime | String | Not null |  |  |
| JMBG | String | Not null |  |  |
| E\_mail | String | Not null |  |  |
| Telefon | String | Not null |  |  |
| Adresa | String | Not null |  |  |

Табела : Клијент

Табела : Пакет

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Табела:  Paket | | Просто Вредносно Ограничење | | Сложено Вредносно Ограничење | | Структурно ограничење |
| Атрибути | Име | Тип | Вредност атрибута | Међузав. атрибута једне табеле | Међузав. атрибута више табела | INSERT RESTRICTED Uredjaj  UPDATE CASCADES Prijava, RESTRICTED Uredjaj  DELETE RESTRICTED Prijava |
| PaketID | Integer | Not null |  |  |
| Naziv | String | Not null |  |  |
| BrojSati | Integer | Not null and >0 |  |  |
| Cena | Double | Not null and >0 |  |  |
| UredjajID | Integer | Not null |  |  |

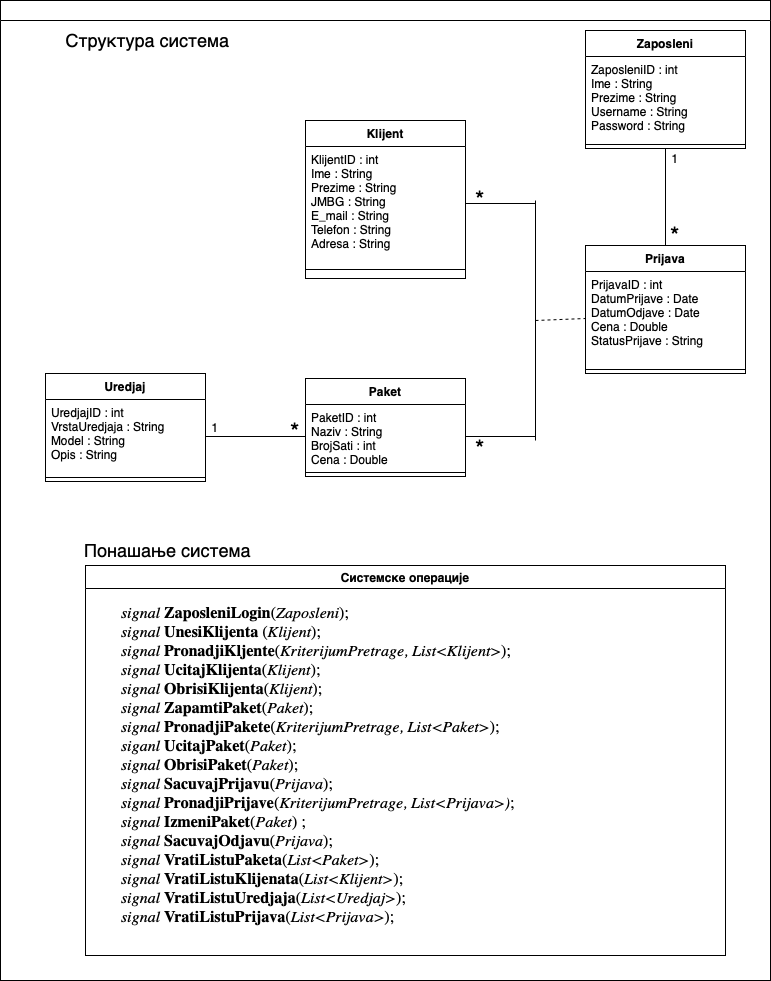
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Табела:  Uredjaj | | Просто Вредносно Ограничење | | Сложено Вредносно Ограничење | | Структурно ограничење |
| Атрибути | Име | Тип | Вредност атрибута | Међузав. атрибута једне табеле | Међузав. атрибута више табела | INSERT /  UPDATE CASCADES Paket  DELETE RESTRICTED Paket |
| UredjajID | Integer | Not null |  |  |
| Opis | String | Not null |  |  |
| VrstaUredjaja | String | Not null |  |  |
| Model | String | Not null |  |  |

Табела : Уређај

Табела : Пријава

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Табела:  Prijava | | Просто Вредносно Ограничење | | Сложено Вредносно Ограничење | | Структурно ограничење |
| Атрибути | Име | Тип | Вредност атрибута | Међузав. атрибута једне табеле | Међузав. атрибута више табела | INSERT RESTRICTED Paket, Klijent, Zaposleni  UPDATE RESTRICTED Paket, Klijent, Zaposleni  DELETE / |
| PrijavaID | Integer | Not null |  |  |
| KlijentID | Integer | Not null |  |  |
| PaketID | Integer | Not null |  |  |
| DatumPrijave | Date | Not null and default: CurrentDate |  |  |
| DatumOdjave | Date | Not null and default: CurrentDate |  |  |
| Cena | Double | Not null and >0 |  |  |
| StatusPrijave | String | Not null and in(‘PRIJAVLJEN’, ‘ODJAVLJEN’) |  |  |
| ZaposleniID | Integer | Not null |  |  |

Као резултат анализе сценарија СК и прављења концептуалног модела добија се логичка структура и понашање софтверског система:



Слика : Структура система

# Пројектовање

Фаза пројектовања описује физичку структуру и понашање софтверског система (архитектуру софтверског система).

## 3.1. Архитектура софтверског система

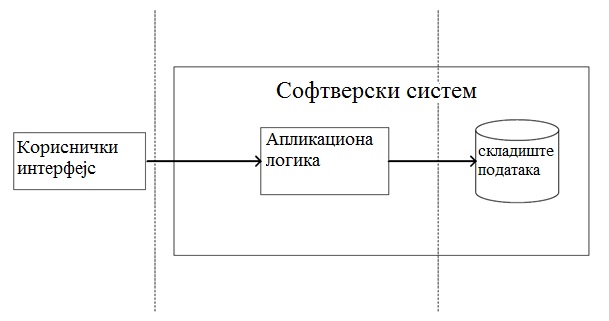
Архитектура софтверског систем је тронивојска и састоји се од следећих нивоа:

• Кориснички интерфејс

• Апликациона логика

• Складиште података

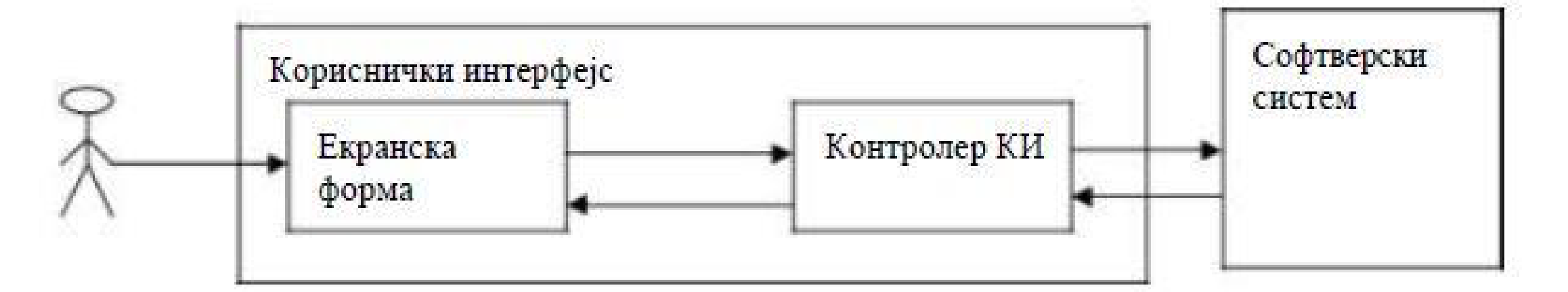
Ниво корисничког интерфејса је на страни клијента, а апликациона логика и складиште података на страни сервера. Сви они заједно учествују у функционисању софтверског система.



Слика

## 3.2 Пројектовање корисничког интерфејса

Кориснички интерфејс представља реализацију улаза и/или излаза софтверског система. Кориснички интерфејс се састоји од екранских форми и контролера корисничког интерфејса. Екранска форма има улогу да прихвата податке које корисник уноси, прихвата догађаје које корисник прави, позива контролера корисничког интерфејса и приказује податке које добија као резултат од контролера.



Слика 33

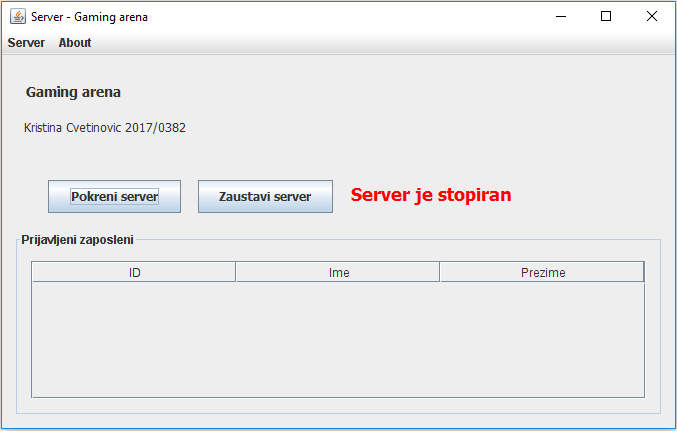
Сценарији коришћења екранских форми су директно повезани са сценаријима случајева коришћења.

### 3.2.1. Пројектовање екранских форми

Кориснички интерфејс је дефинисан преко скупа екранских форми. Сценарија коришћења екранских форми су директно повезани са сценаријима случајева коришћења. Екранска форма има улогу да прихвати податке које уноси актор, прихвата догађаје које прави актор, позива контролера корисничког интерфејса како би му проследио те податке и приказује податке добијене од контролера корисничког интерфејса.

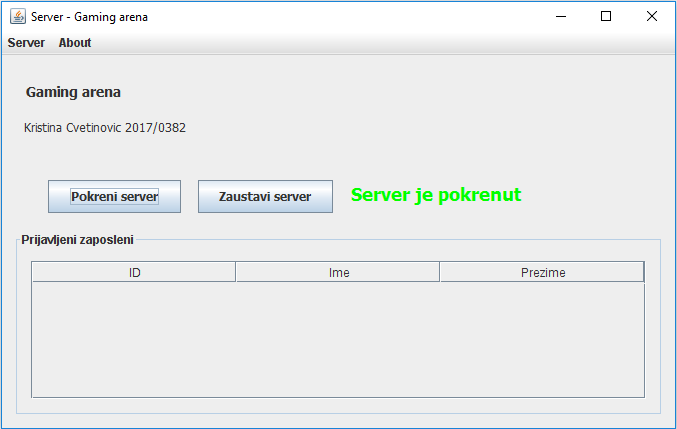
#### Екранска форма - Сервер

Главна екранска форма серверске стране апликације на почетку изгледа овако:



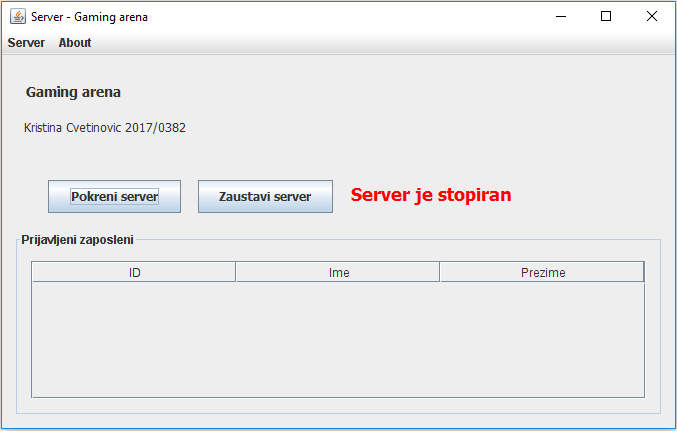
Слика

Кликом на дугме Покрени сервер, сервер се покреће након чега форма изгледа овако:



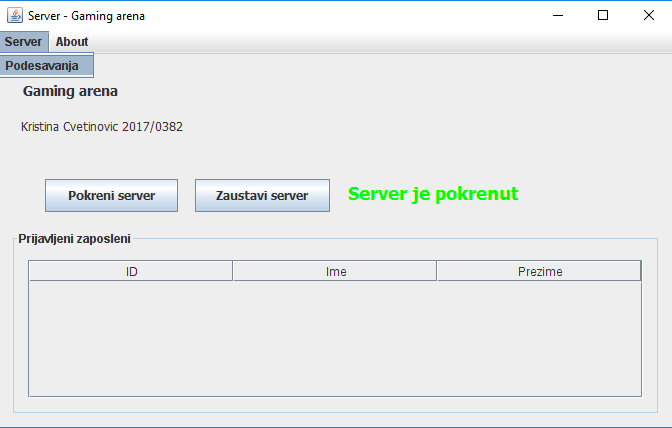
Слика

Кликом на дугме Заустави сервер, сервер се зауставља.

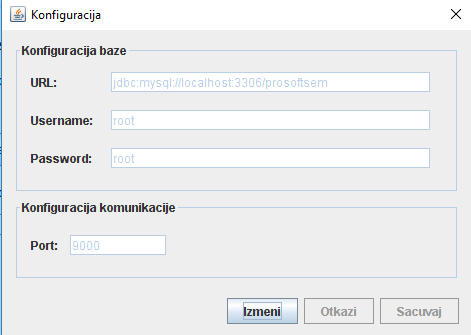


Слика

Омогућен је увид и подешавање параметара конекције за базу података, као и параметара конекције, односно избор броја порта на коме ће сервер бити покренут, избором из менија "Podešavanjа".

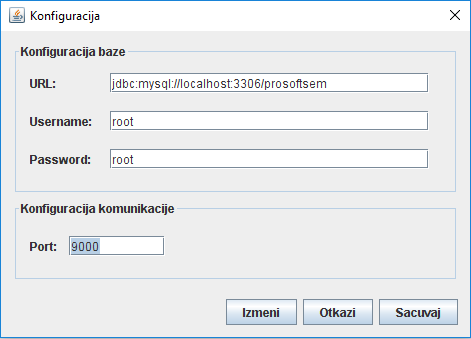


Слика



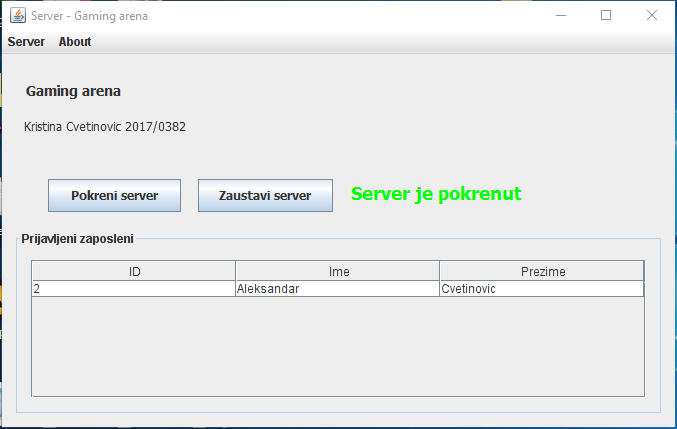
Слика

Кликом на дугме измени омогућена је измена параметара:



Слика

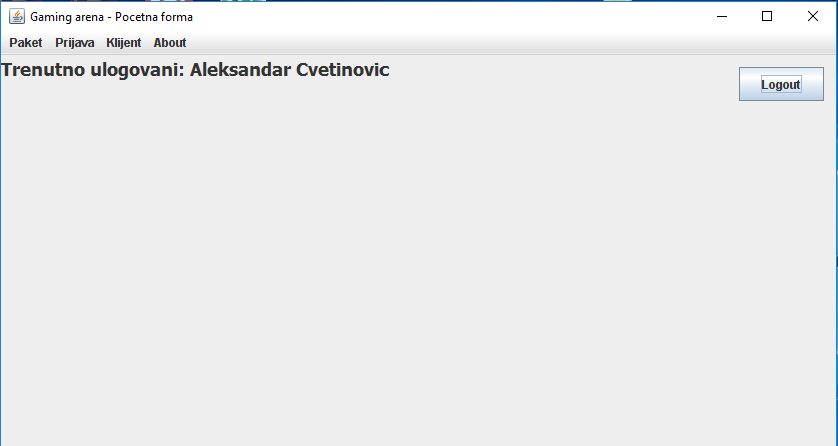
Такође омогућен је и преглед свих тренутно пријављених запослених.



Слика

#### Екранска форма - Клијент

При покретању корисничког дела програма приказује се следећи кориснички интерфејс:



Слика

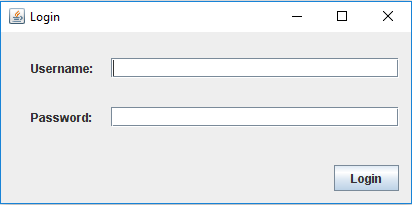
**СК1: Случај коришћења – Пријављивање запосленог у систем**

**Назив СК**   
Пријављивање запосленог у систем

**Актори СК**   
Запослени

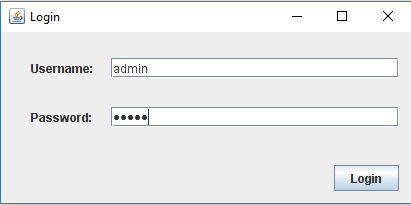
**Учесници СК**   
Запослени и систем (програм)

**Предуслов:** Систем је укључен и приказује форму за пријављивање запосленог.

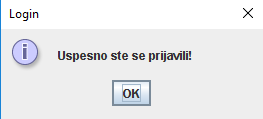


Слика 42

**Основни сценарио СК**

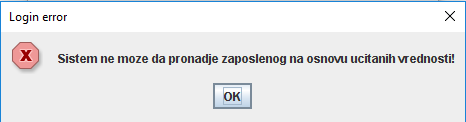
1. Запослени **уноси** податке за пријаву запосленог. (АПУСО)  
   

Слика 43

1. Запослени **контролише** да ли је коректно унео податке за пријаву запосленог. (АНСО)
2. Запослени **позива** систем да пронађе запосленог са задатим подацима. (АПСО)
3. Систем **претражује** запослене. (СО)
4. Систем **приказује** запосленом пронађеног запосленог и поруку: „Успешно сте се пријавили!“ и омогућава приступ систему. (ИА)  
     
   

Слика 44

**Алтернативна сценација** 4.1 Уколико систем не може да пронађе запосленог приказује запосленом поруку: „Систем не може да пронађе запосленог на основу учитаних вредности.“ (ИА)



Слика 45

**СК2: Случај коришћења – Унос новог клијента**

**Назив СК**

Креирање новог клијента

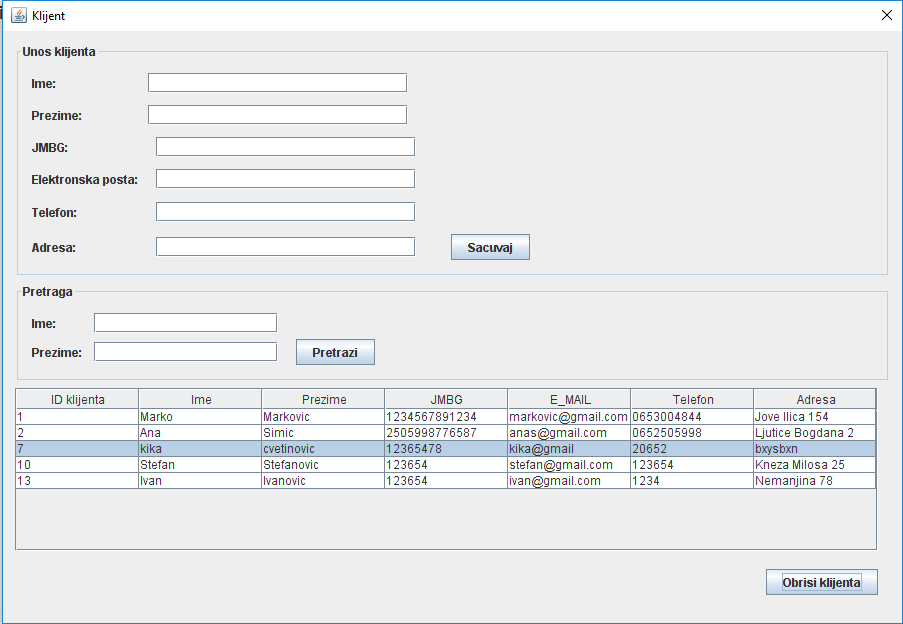
**Актори СК**

Запослени

**Учесници СК**

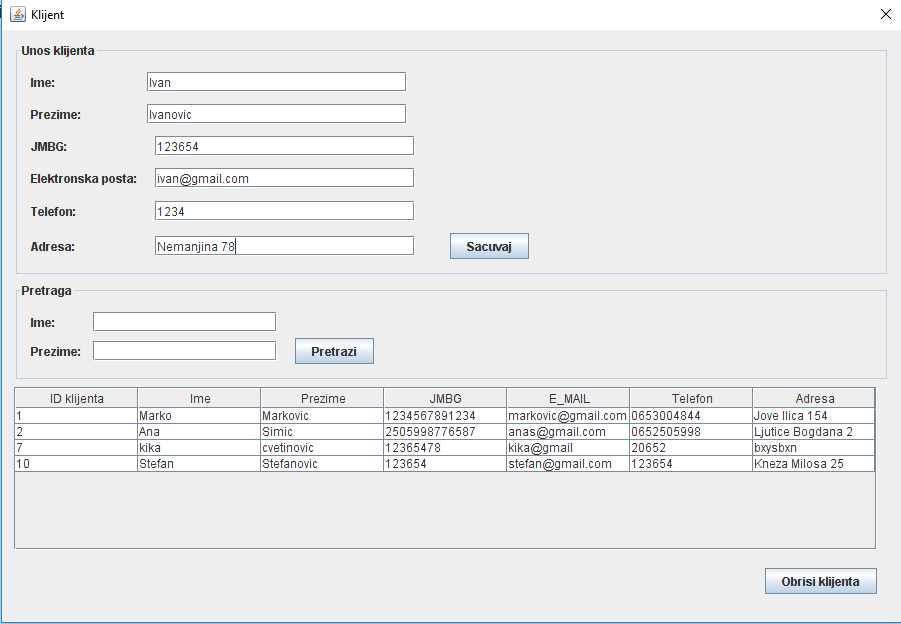
Запослени и систем (програм)

**Предуслов**: Систем је укључен и запослени je улогован под својом шифром. Систем приказује форму за рад са налогом клијента.

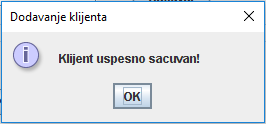


Слика

**Основни сценарио СК**

1. Запослени **уноси** податке о клијенту. (АПУСО)  
   

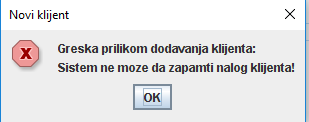
Слика

1. Запослени **контролише** да ли је коректно унео податке у налог клијента. (АНСО)
2. Запослени **позива** систем да запамти податке о налогу клијента. (АПСО)
3. Систем **памти** податке о налогу клијента. (СО)
4. Систем **приказује** запосленом запамћени налог клијента и поруку: “Клијент успешно сачуван“. (ИА)  
     
   

Слика

**Алтернативна сценарија**

5.1 Уколико систем не може да запамти податке о налогу клијента он приказује запосленом поруку “Систем не може да запамти   
налог клијента ”. (ИА)



Слика

**СК3: Случај коришћења – Брисање клијента**

**Назив СК**

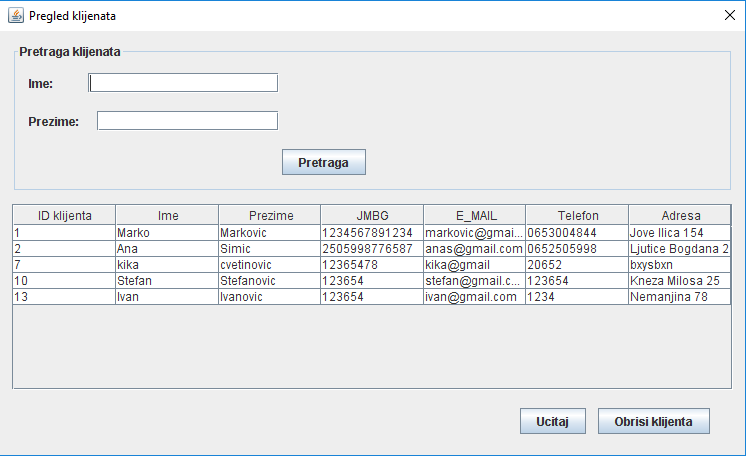
Брисање клијента

**Актори СК**

запослени

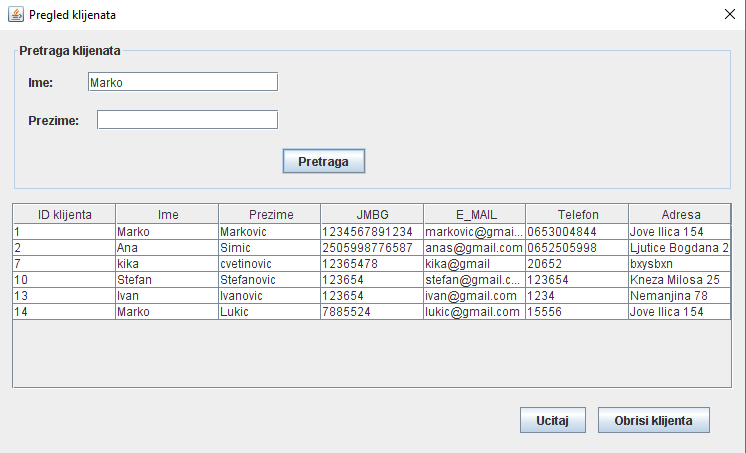
**Учесници СК**

запослени и систем (програм)

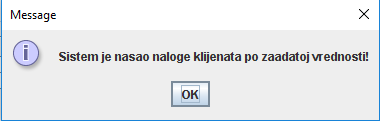
**Предуслов**: Систем је укључен и запослени je улогован под својом шифром. Учитана је листа налога клијената. Систем приказује форму за рад са налогом клијента.   
  


Слика

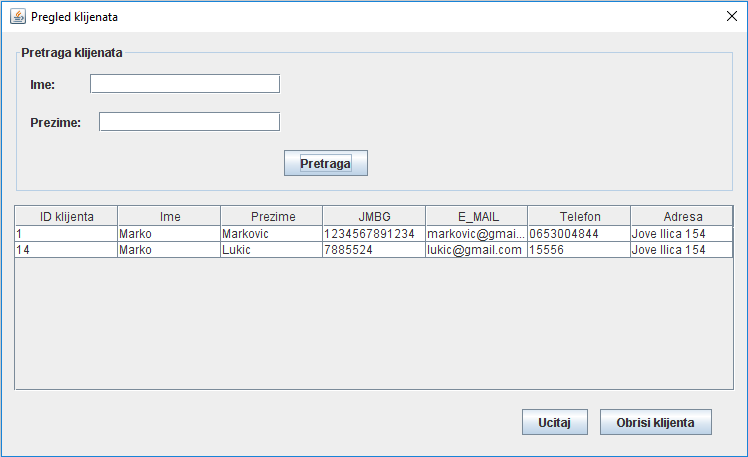
**Основни сценарио СК**

1. Запослени **уноси** вредностпо којој претражује налоге клијената. (АПУСО)  
   

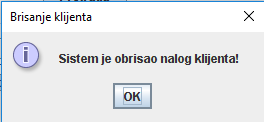
Слика

1. Запослени **позива** систем да нађе налоге клијената по задатој вредности. (АПСО)
2. Систем **тражи** налоге клијената по задатој вредности. (СО)
3. Систем приказује запосленом налогe клијенaта и поруку: “Систем је нашао налогe клијенaта по задатој вредности”. (ИА)

Слика

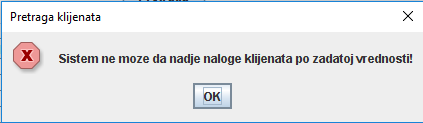


Слика

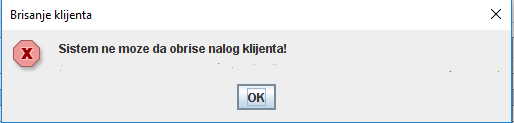
1. Запослени **бира** налог клијента који жели да обрише. (АПУСО)
2. Запослени **позива** систем да обрише одабрани налог клијента. (АПСО)
3. Систем **брише** налог клијента. (СО)
4. Систем **приказује** запосленом поруку: “Систем је обрисао налог клијента.”(ИА)  
   

Слика

**Алтернативна сценарија**

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


Слика

8.1 Уколико систем не може да обрише налог клијента он приказује запосленом поруку “Систем не може да обрише налог клијента ”. (ИА)  


Слика

**СК4: Случај коришћења – Претраживање клијента**

**Назив СК**

Претраживање клијента

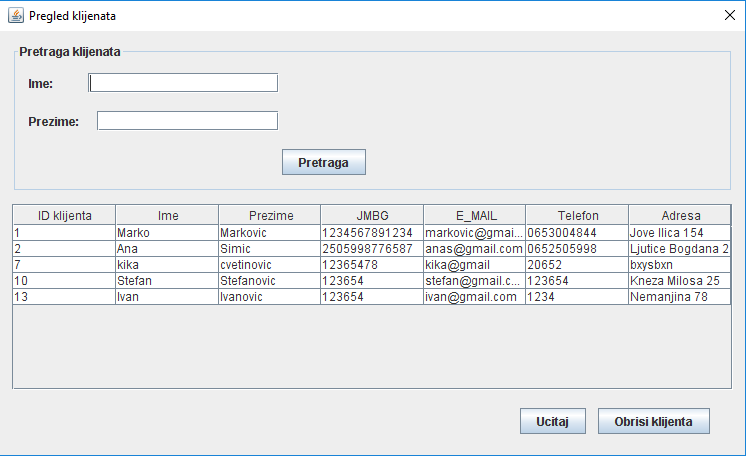
**Актори СК**

запослени

**Учесници СК**

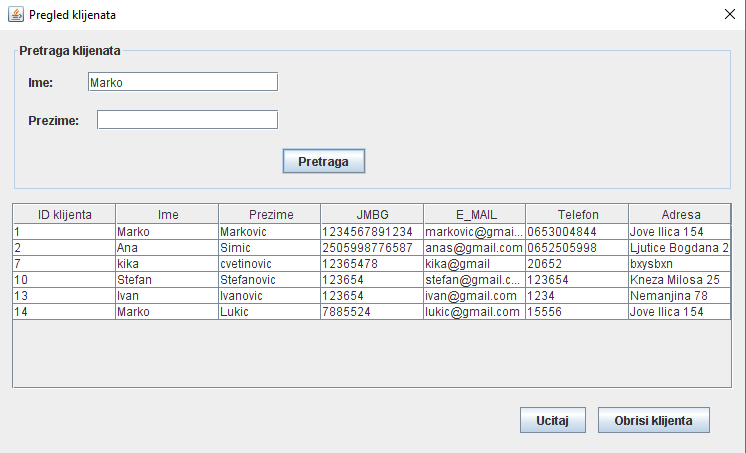
запослени и систем (програм)

**Предуслов**: Систем је укључен и запослени je улогован под својом шифром. Учитана је листа налога клијената. Систем приказује форму за рад са налогом клијента.

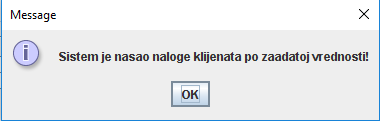


Слика

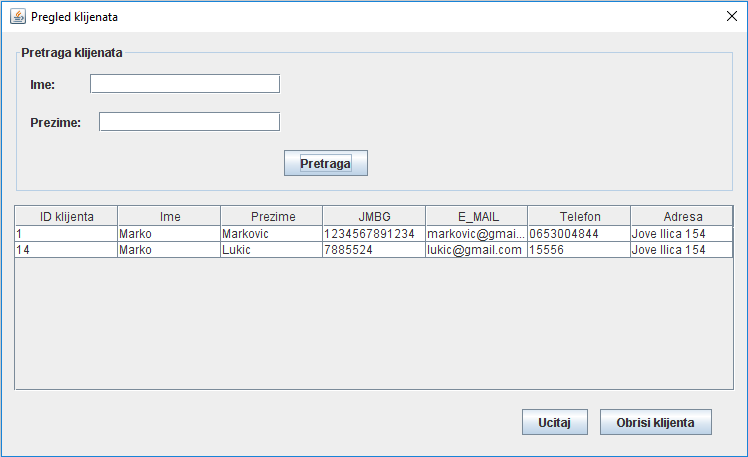
**Основни сценарио СК**

1. Запослени **уноси** вредностпо којој претражује налоге клијената. (АПУСО)  
   

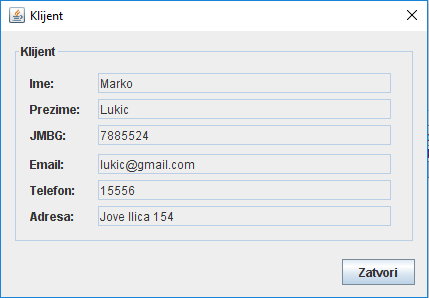
Слика

1. Запослени **позива** систем да нађе налоге клијената по задатој вредности. (АПСО)
2. Систем **тражи** налоге клијената по задатој вредности. (СО)
3. Систем приказује запосленом налоге клијената и поруку: “Систем је нашао налоге клијената по задатој вредности”. (ИА)  
   

Слика

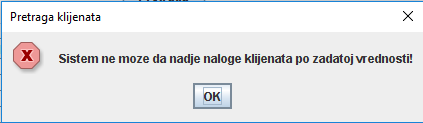


Слика

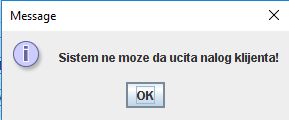
1. Запослени **бира** налог клијента који жели да учита. (АПУСО)
2. Запослени **позива** систем да учита податке о налогу клијента. (АПСО)
3. Систем **учитава** податке оодабраном налогу клијента. (СО)
4. Систем приказује запосленом податке о налогу клијента. (ИА)  
   

Слика

**Алтернативна сценарија**

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


Слика

8.1 Уколико систем не може да учита податке о налогу клијента он приказује запосленом поруку: “Систем не може да учита налог клијента”.(ИА)   
  


Слика

**СК5: Случај коришћења – Унос новог пакета**

**Назив СК**

Креирање пакета

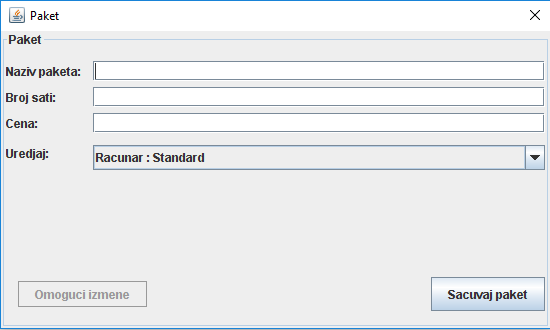
**Актори СК**

Запослени

**Учесници СК**

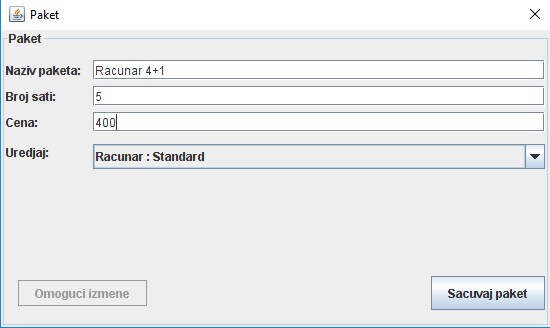
Запослени и систем (програм)

**Предуслов**: Систем је укључен и запослени je улогован под својом шифром. Систем приказује форму за рад са пакетом. Учитана је листа уређаја.



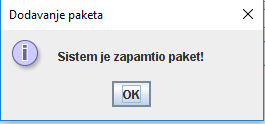
Слика

**Основни сценарио СК**

1. Запослени **уноси** податке о пакету. (АПУСО)  
     
   

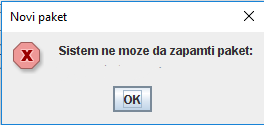
Слика

1. Запослени **контролише** да ли је коректно унео податке у пакет. (АНСО)
2. Запослени **позива** систем да запамти податке о пакету. (АПСО)
3. Систем **памти** податке о пакету. (СО)
4. Систем **приказује** запосленом запамћени пакет и поруку: “Систем је запамтио пакет“. (ИА)



Слика

**Алтернативна сценарија**

5.1 Уколико систем не може да запамти податке о пакету он приказује запосленом поруку “Систем не може да запамти пакет”. (ИА)  


Слика

**СК6: Случај коришћења – Измена пакета**

**Назив СК**

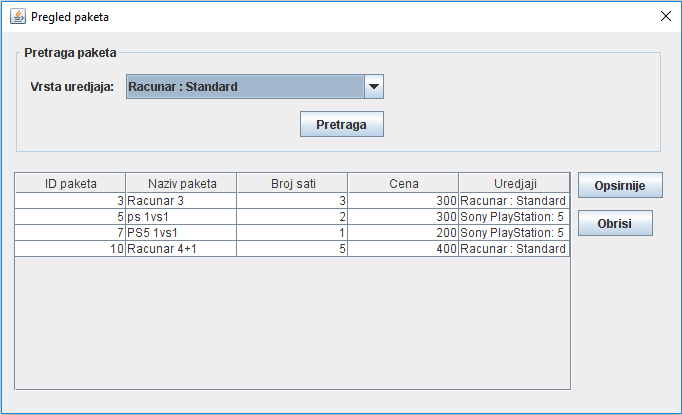
Промена пакета

**Актори СК**

Запослени

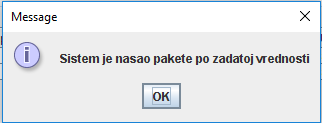
**Учесници СК**

Запослени и систем (програм)

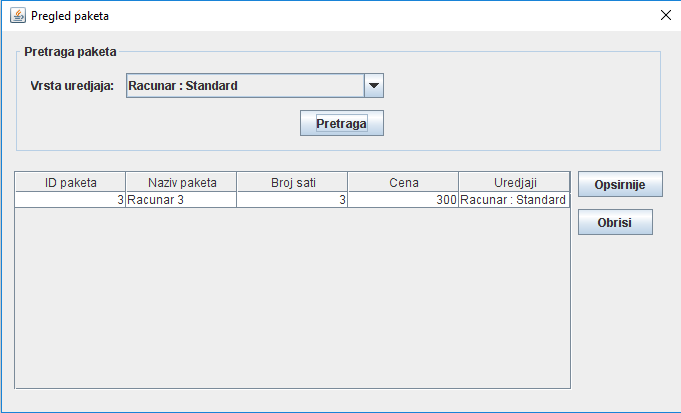
**Предуслов**: Систем је укључен и запослени je улогован под својом шифром. Систем приказује форму за рад са пакетом. Учитана је листа пакета. Учитана је листа уређаја.  


Слика

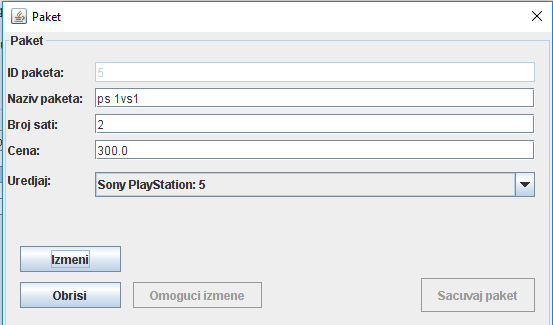
**Основни сценарио СК**

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

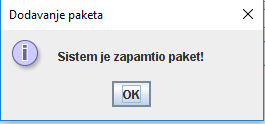
Слика



Слика

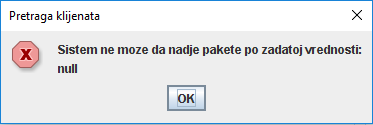
1. Запослени **бира** пакет који жели да измени. (АПУСО)
2. Запослени **позива** систем да учита податке о пакету. (АПСО)
3. Систем **учитава** податке оодабраном пакету. (СО)
4. Систем приказује запосленом податке о пакету. (ИА)  
   

Слика

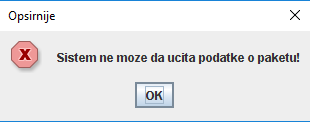
1. Запослени **уноси** **(мења)** податке о пакету. (АПУСО)
2. Запослени **контролише** да ли је коректно унео податке о пакету. (АНСО)
3. Запослени **позива** систем да измени податке о пакету. (АПСО)
4. Систем **памти** податке о пакету. (СО)
5. Систем **приказује** запослеом запамћени пакет и поруку: “Систем је запамтио пакет.” (ИА)  
   

Слика

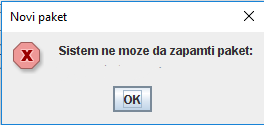
**Алтернативна сценарија**

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


Слика

8.1 Уколико систем не може да учита податке о пакету он приказује запосленом поруку: “Систем не може да учита податке о пакету”. Прекида се извршење сценариа. (ИА)  


Слика

13.1 Уколико систем не може да запамти податке о пакету он приказује запосленом поруку “Систем не може да запамти пакет”. (ИА)  


Слика

**СК7: Случај коришћења – Брисање пакета**

**Назив СК**

Брисање пакета

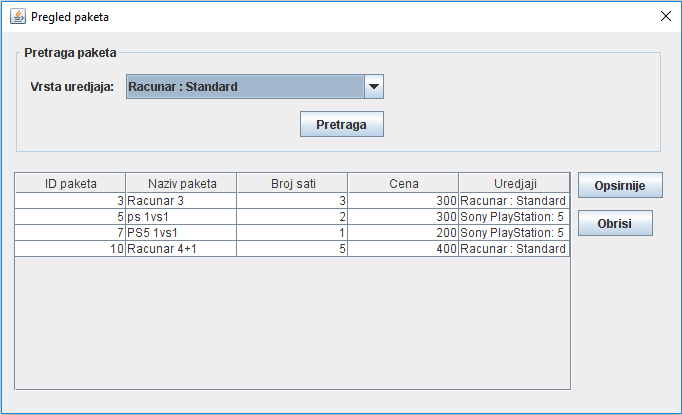
**Актори СК**

Запослени

**Учесници СК**

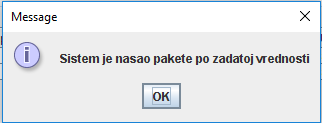
Запослени и систем (програм)

**Предуслов**: Систем је укључен и запослени je улогован под својом шифром. Систем приказује форму за рад са пакетом. Учитана је листа пакета.

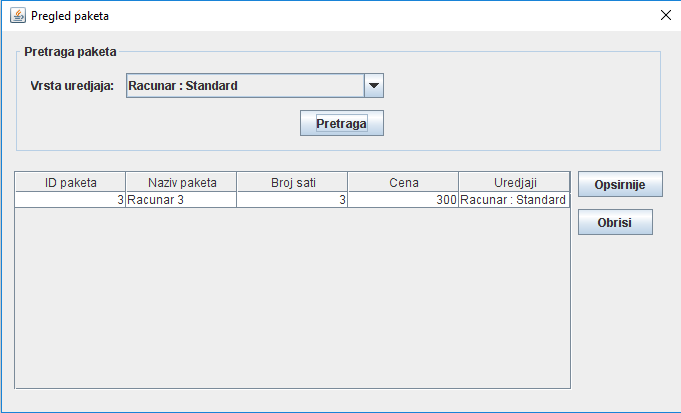


Слика

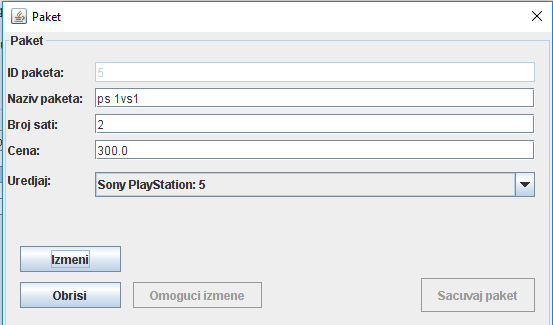
**Основни сценарио СК**

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

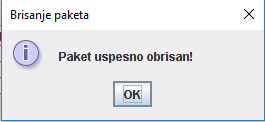
Слика



Слика

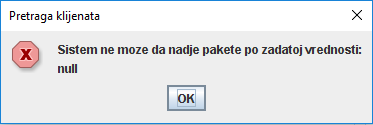
1. Запослени **бира** пакет који жели да обрише. (АПУСО)
2. Запослени **позива** систем да учита податке о пакету. (АПСО)
3. Систем **учитава** податке оодабраном пакету. (СО)
4. Систем приказује запосленом податке о пакету (ИА)  
   

Слика

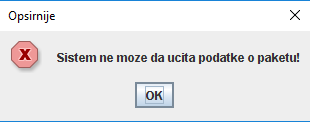
1. Запослени **позива** систем да обрише пакет. (АПСО)
2. Систем **брише** пакет. (СО)
3. Систем **приказује** запосленом поруку: “Систем је обрисао пакет.” (ИА)  
   

Слика

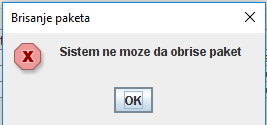
**Алтернативна сценарија**

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


Слика

8.1 Уколико систем не може да учита податке о пакету он приказује запосленом поруку: “Систем не може да учита податке о пакету”. Прекида се извршење сценариа. (ИА)  


Слика

11.1 Уколико систем не може да обрише пакет он приказује запосленом поруку “Систем не може да обрише пакет”. (ИА)  


Слика

**СК8: Случај коришћења – Пријава клијента на пакет**

**Назив СК**

Пријава клијента на пакет

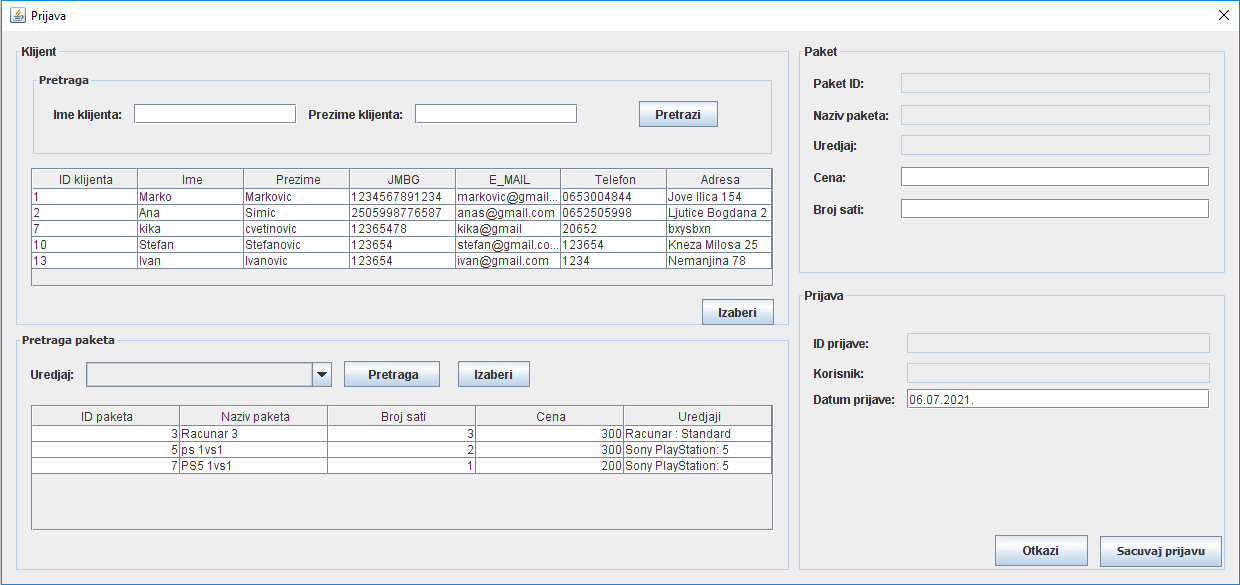
**Актори СК**

Запослени

**Учесници СК**

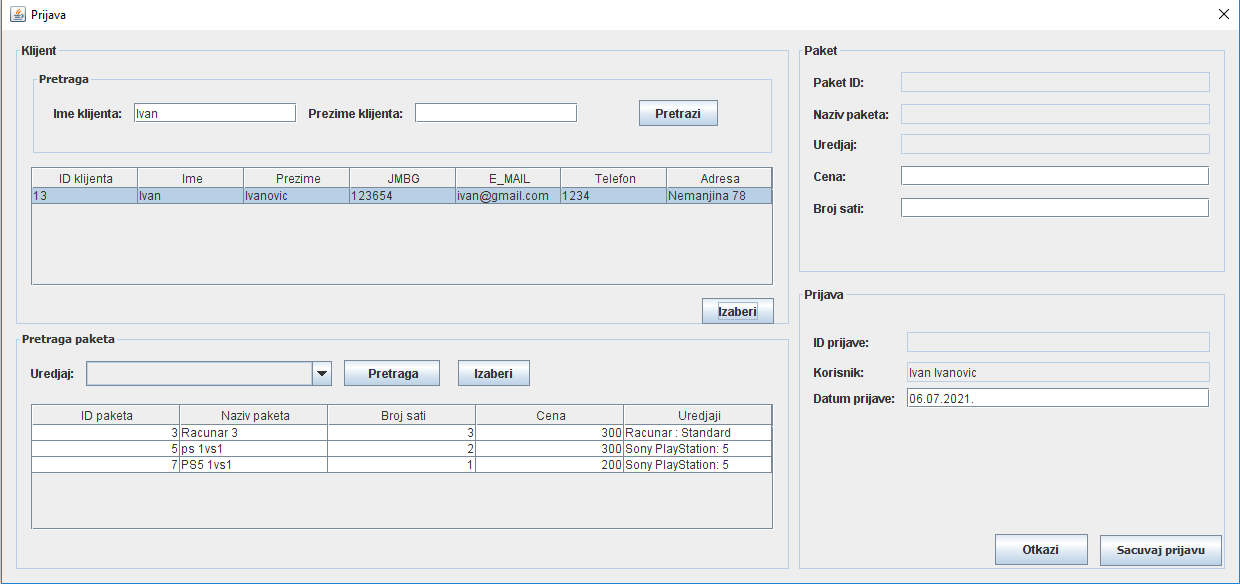
Запослени и систем (програм)

**Предуслов**: Систем је укључен и запослени је улогован под својом шифром. Учитана је листа пакета. Учитана је листа клијената. Клијент и пакет постоје у систему. Систем приказује форму за рад са пријавама на пакет.

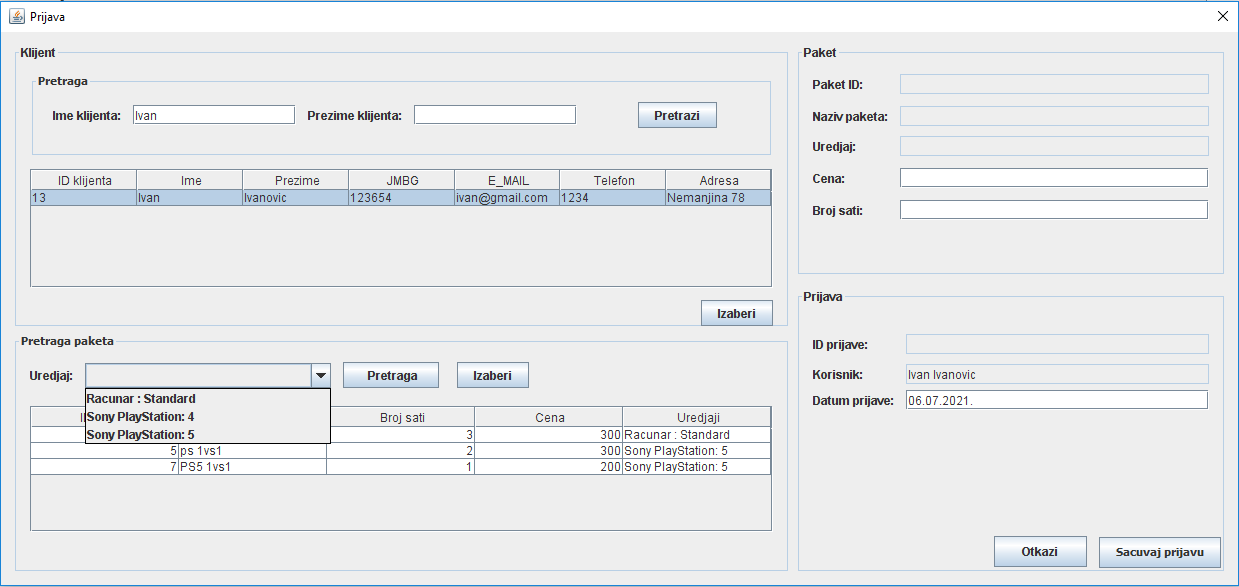


Слика

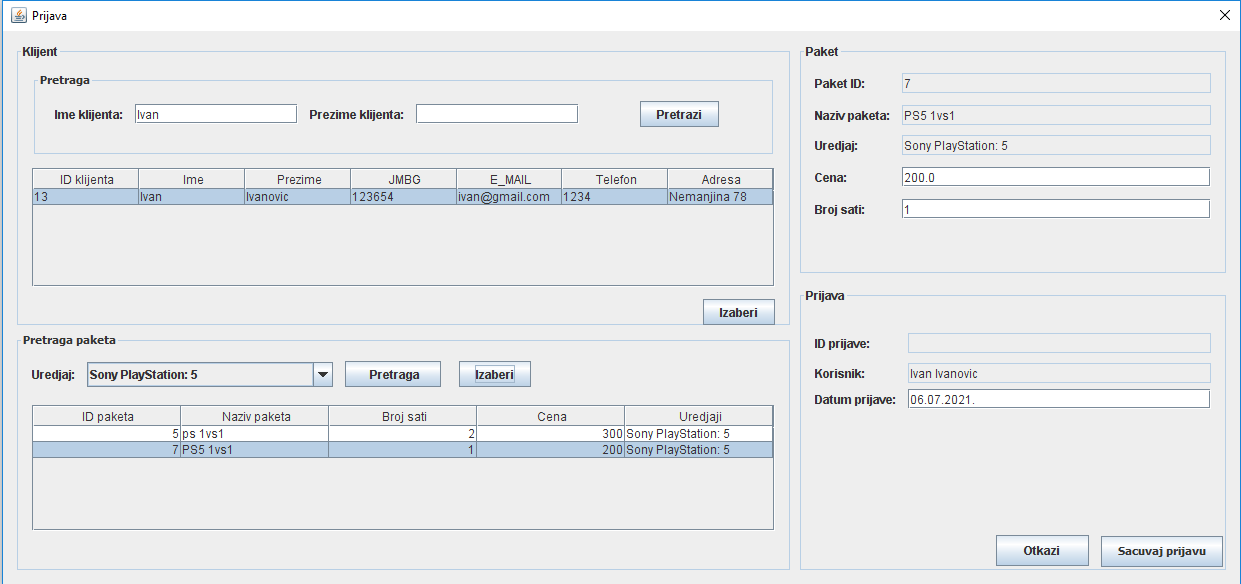
**Основни сценарио СК**

1. Запослени **уноси** вредности по којима претражује клијенте. (АПУСО)
2. Запослени **позива** систем да нађе клијенте по задатој вредности. (АПСО)
3. Систем **тражи** клијенте по задатој вредности. (СО)
4. Систем **приказује** запосленом клијенте. (ИА)
5. Запослени **бира** клијента којег жели да пријави на пакет. (АПУСО)
6. Запослени **позива** систем да учита податке о клијенту којег жели да пријави на пакет. (АПСО)
7. Систем **учитава** податке о клијенту. (СО)
8. Систем **приказује** запосленом име и презиме клијента на форми. (ИА)  
   

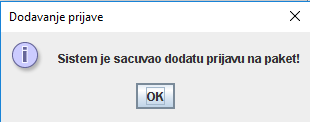
Слика

1. Запослени **уноси** вредности по којима претражује пакете. (АПУСО)  
   

Слика

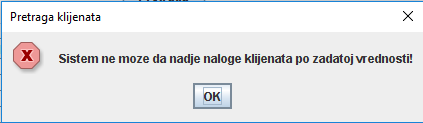
1. Запослени **позива** систем да нађе пакете по задатој вредности. (АПСО)
2. Систем **тражи** пакете по задатој вредности. (СО)
3. Систем **приказује** запосленом пакете.(ИА)
4. Запослени **бира** пакет на који жели да пријави корисника. (АПУСО)
5. Запослени **позива** систем да учита податке о пакету на који жели да пријави корисника. (АПСО)
6. Систем **учитава** податке о пакету. (СО)
7. Систем **приказује** запосленом податке о пакету. (ИА)  
   

Слика

1. Запослени **уноси** податке о пријави на пакет за који се клијент определио. (АПУСО)
2. Запослени **контролише** да ли је коректно унео податке о пријави на пакет. (АНСО)
3. Запослени **позива** систем да сачува пријаву на пакет за који се клијент определио. (АПСО)
4. Систем **чува** пријаву на пакет. (СО)
5. Систем **приказује** запосленом поруку: „Систем је сачувао додату пријаву на пакет“. (ИА)  
   

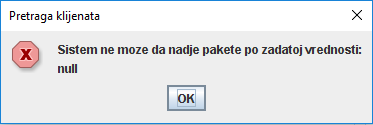
Слика

**Алтернативна сценарија**

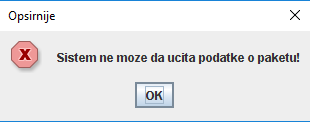
4.1 Уколико систем не може да нађе клијенте он приказује запосленом поруку: “Систем не може да нађе клијенте по задатој вредности”. Прекида се извршење сценариа. (ИА)  


Слика

8.1 Уколико систем не може да учита податке о клијенту он приказује запосленом поруку: “Систем не може да учита име и презиме клијента”. Прекида се извршење сценариа (ИА)

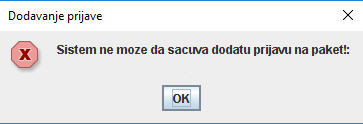
12.1 Уколико систем не може да нађе пакетe он приказује запосленом поруку: “Систем не може да нађе пакетe по задатој вредности”. Прекида се извршење сценариа. (ИА)   


Слика

16.1 Уколико систем не може да учита податке о пакету он приказује запосленом поруку: “Систем не може да учита податке о пакету”. Прекида се извршење сценариа. (ИА)  


Слика

21.1. Уколико систем не може да сачува нову пријаву на пакет он приказује запосленом поруку: „Систем не може да сачува додату пријаву на пакет “.(ИА)



**СК9: Случај коришћења – Одјава клијента са пакета**

**Назив СК**

Одјава клијента са пакета

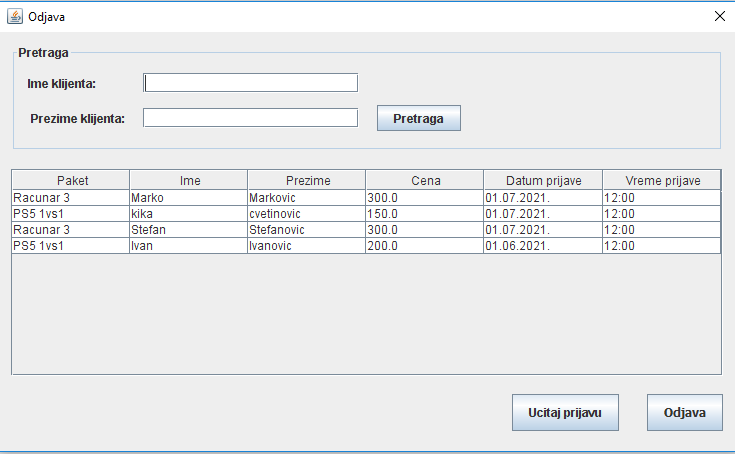
**Актори СК**

Запослени

**Учесници СК**

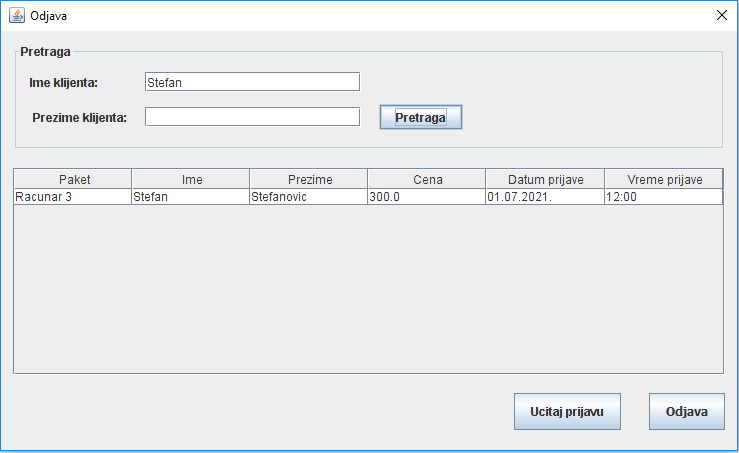
Запослени и систем (програм)

**Предуслов**: Систем је укључен и запослени је улогован под својом шифром. Учитана је листа пакета. Учитана је листа клијената. Клијент и пакет постоје у систему. Систем приказује форму за рад са одјавама.

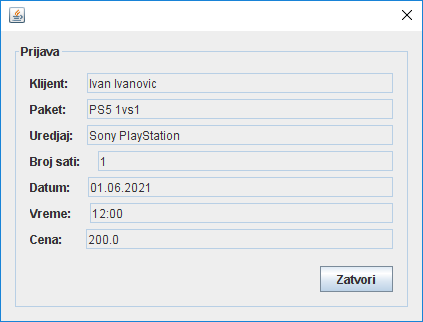


Слика

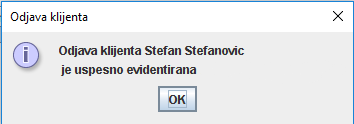
**Основни сценарио СК**

1. Запослени **уноси** вредности по којима претражује пријаве на пакете. (АПУСО)
2. Запослени **позива** систем да нађе пријаве на пакете по задатој вредности. (АПСО)
3. Систем **тражи** пријаве на пакете по задатој вредности. (СО)
4. Систем **приказује** запосленом активне пријаве на пакете.(ИА)  
   

Слика

1. Запослени **бира** пријаву на пакет са које жели да одјави клијента. (АПУСО)
2. Запослени **позива** систем да учита податке о пријави на пакет са које жели да одјави клијента. (АПСО)
3. Систем **учитава** податке о пријави на пакет. (СО)
4. Систем **приказује** запосленом податке о пријави на пакет.(ИА)  
   

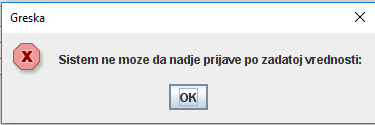
Слика

1. Запослени **позива** систем да одјави клијента са пакета. (АПСО)
2. Систем **чува** податке о одјави клијента са пакета. (СО)
3. Систем **приказује** запосленом поруку: „Одјава клијента је успешно евидентирана.“ (ИА)  
   

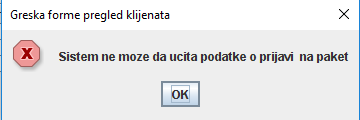
Слика



**Алтернативна сценарија**

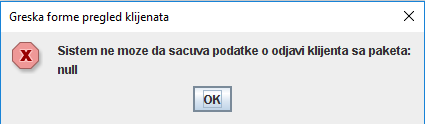
4.1 Уколико систем не може да нађе пријавe на пакетe по задатој вредности он приказује запосленом поруку: „Систем не може да нађе пријавe по задатој вредности“. Прекида се извршење сценариа. (ИА)  


Слика

8.1 Уколико систем не може да учита податке о пријави на пакет он приказује запосленом поруку: „Систем не може да учита податке о пријави на пакет“. Прекида се извршење сценариа. (ИА)  


Слика

11.1 Уколико систем не може да сачува податке о одјави клијента са пакета он приказује запосленом поруку: „Систем не може да сачува податке о одјави клијента са пакета “. (ИА)



Слика

## 3.3. Пројектовање контролера корисничког интерфејса

Контролер корисничког интерфејса је одговоран за:

• прихватање графичких објеката од екранске форме

• конвертовање података који се налазе у графичким објектима у доменске објекте који ће бити прослеђени преко мреже до апликационог сервера

• конвертовање доменских објеката у графичке објекте и прослеђује их до екранске форме.

Контролер има улогу посредника између софтверског система и екранских форми.

Класа “Communication” дефинисана је као Singleton јер је користе све екранске форме током извршавања програма, и није потребно инстанцирати га више пута. Он представља контролера Има по једну методу за сваки захтев који клијент жели да пошаље серверу, која је задужена за обликовање, односно креирање захтева клијента, примање одговора и враћање резултата системске операције до екранске форме.

Пример методе:



Слика

## 3.4. Пројектовање апликационе логике

Апликациони сервери су одговорни да обезбеде сервисе који ће да омогуће реализацију апликационе логике софтверског система.

Пројектовани апликациони сервер садржи:

* део за комуникацију са клијентима,
* контролер апликационе логике,
* део за комуникацију са складиштем података (Брокер базе података),
* део који садржи пословну логику.

### 3.3.1. Комуникација са клијентима

Део за комуникацију, односно нит "Server" подиже серверски сокет који ослушкује мрежу.Када клијент успостави конекцију, сервер генерише нит која ће бити одговорна за двосмерну везу са клијентом**,** "ProcessClientsRequests"**.**

Клијент шаље захтев за извршење неке од системских операција, одговарајућа нит која је додељена клијенту прихвата захтев и позива контролера апликационе логике да изврши одговарајуће системске операције. Након извршења системске операције, контролер враћа резултат извршења операције до нити која је задужена за тог клијента.Резултат се затим прослеђује клијенту.

Комуникација се обавља разменом објеката класе “Request” и класе “Response”.

***Request***

package rs.ac.bg.fon.ps.communication;

import java.io.Serializable;

/\*\*  
\*  
\* @author Lenovo  
\*/  
public class Request implements Serializable{  
private Operation operation;  
private Object argument;

public Request() {  
}

public Request(Operation operation, Object argument) {  
this.operation = operation;  
this.argument = argument;  
}

public Operation getOperation() {  
return operation;  
}

public void setOperation(Operation operation) {  
this.operation = operation;  
}

public Object getArgument() {  
return argument;  
}

public void setArgument(Object argument) {  
this.argument = argument;  
}

***Resposnse***

package rs.ac.bg.fon.ps.communication;

import java.io.Serializable;

/\*\*  
\*  
\* @author Lenovo  
\*/  
public class Response implements Serializable{  
private Object result;  
private Exception exception;

public Response() {  
}

public Response(Object result, Exception exception) {  
this.result = result;  
this.exception = exception;  
}

public Object getResult() {  
return result;  
}

public void setResult(Object result) {  
this.result = result;  
}

public Exception getException() {  
return exception;  
}

public void setException(Exception exception) {  
this.exception = exception;  
}  
  
  
}

Захтеви, као и одговори се шаљу помоћу класе Sender, а прихватају се помоћу класе Receiver.

***Sender:***

public class Sender implements Serializable{  
private Socket socket;

public Sender(Socket socket) {  
this.socket = socket;  
}  
public void send(Object object) throws Exception{  
try {  
ObjectOutputStream out=new ObjectOutputStream(socket.getOutputStream());  
out.writeObject(object);  
out.flush();  
} catch (IOException ex) {  
ex.printStackTrace();  
throw new Exception("Greska pri slanju objeka! \n"+ex.getMessage());  
}  
}  
}

***Receiver:***

package rs.ac.bg.fon.ps.communication;

import java.io.IOException;  
import java.io.ObjectInputStream;  
import java.io.Serializable;  
import java.net.Socket;  
import java.util.logging.Level;  
import java.util.logging.Logger;

/\*\*  
\*  
\* @author Lenovo  
\*/  
public class Receiver implements Serializable{  
private Socket socket;

public Receiver(Socket socket) {  
this.socket = socket;  
}  
  
public Object receive() throws Exception{  
try {  
ObjectInputStream in=new ObjectInputStream(socket.getInputStream());  
return in.readObject();  
} catch (IOException ex) {  
ex.printStackTrace();  
throw new Exception("Greska pri prihvatanju objeka! \n"+ex.getMessage());  
}  
}  
}

***ProcessCilentsRequests****:*

package rs.ac.bg.fon.ps.thread;

import java.io.IOException;  
import java.net.Socket;  
import java.util.List;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
import rs.ac.bg.fon.ps.communication.Receiver;  
import rs.ac.bg.fon.ps.communication.Request;  
import rs.ac.bg.fon.ps.communication.Response;  
import rs.ac.bg.fon.ps.communication.Sender;  
import rs.ac.bg.fon.ps.controller.Controller;  
import rs.ac.bg.fon.ps.domain.Klijent;  
import rs.ac.bg.fon.ps.domain.Paket;  
import rs.ac.bg.fon.ps.domain.Prijava;  
import rs.ac.bg.fon.ps.domain.Zaposleni;  
import rs.ac.bg.fon.ps.view.controller.MainController;

/\*\*  
\*  
\* @author Lenovo  
\*/  
public class ProcessClientsRequests extends Thread {

Socket socket;  
private boolean state=false;

public ProcessClientsRequests(Socket socket) {  
this.socket = socket;  
}

@Override  
public void run() {  
Sender sender = new Sender(socket);  
Receiver receiver = new Receiver(socket);  
while (!false) {  
try {  
Request request = (Request) receiver.receive();  
Response response = new Response();  
try {  
switch (request.getOperation()) {

case ZAPOSLENI\_LOGIN:  
Zaposleni zaposleni = (Zaposleni) request.getArgument();  
Zaposleni ulogovani=Controller.getInstance().zaposleniLogin(zaposleni.getUsername(), zaposleni.getPassword(), this);  
MainController.getInstance().dodajUlogovanog(ulogovani);  
response.setResult(ulogovani);  
break;  
case GET\_ALL\_UREDJAJI:  
response.setResult(Controller.getInstance().getAllUredjaji());  
break;  
case GET\_ALL\_KLIJENTI:  
response.setResult(Controller.getInstance().getAllKlijenti());  
break;  
case GET\_ALL\_PAKETI:  
response.setResult(Controller.getInstance().getAllPaketi());  
break;  
case ADD\_PAKET:  
Paket paket = (Paket) request.getArgument();  
Controller.getInstance().addPaket(paket);  
break;  
case ADD\_PRIJAVA:  
Prijava prijava = (Prijava) request.getArgument();  
Controller.getInstance().addPrijava(prijava);  
response.setResult(prijava);  
break;  
case IZMENI\_PAKET:  
Paket paket1 = (Paket) request.getArgument();  
Controller.getInstance().izmeniPaket(paket1);  
break;  
case OBRISI\_PAKET:  
Paket paket2 = (Paket) request.getArgument();  
Controller.getInstance().obrisiPaket(paket2);  
break;  
case VRATI\_PAKET:  
Paket paketZahtev = (Paket) request.getArgument();  
Paket paket3 = Controller.getInstance().vratiPaket(paketZahtev);  
response.setResult(paket3);  
break;  
case ADD\_KLIJENT:  
Klijent noviKlijent=(Klijent) request.getArgument();  
Controller.getInstance().unesiKlijenta(noviKlijent);  
break;  
case OBRISI\_KLIJENTA:  
Klijent klijentBrisanje=(Klijent) request.getArgument();  
Controller.getInstance().obrisiKlijenta(klijentBrisanje);  
break;  
case PRETRAGA\_KLIJENATA:  
Klijent klijent=(Klijent) request.getArgument();  
List<Klijent> listaPretragaKlijenata=Controller.getInstance().vratiKlijentePoUslovu(klijent);  
response.setResult(listaPretragaKlijenata);  
break;  
case VRATI\_AKTIVNE\_PRIJAVE:  
Prijava prijavaUslov=(Prijava) request.getArgument();  
List<Prijava> listaAktivnihPrijava=Controller.getInstance().vratiAktivnePrijave(prijavaUslov);  
response.setResult(listaAktivnihPrijava);  
break;  
case ODJAVI\_KORISNIKA:  
Prijava prijavaOdj=(Prijava) request.getArgument();  
Controller.getInstance().odjaviSaPaketa(prijavaOdj);  
break;  
case PRETRAGA\_PAKETA:  
Paket paketPretraga=(Paket) request.getArgument();  
List<Paket> listaPaketaIzPretrage=Controller.getInstance().pretraziPakete(paketPretraga);  
response.setResult(listaPaketaIzPretrage);  
break;  
case ODJAVI\_ZAPOSLENOG:  
Zaposleni zaposleniLogout=(Zaposleni) request.getArgument();  
MainController.getInstance().izbaciOdjavljenog(zaposleniLogout);  
Controller.getInstance().odjaviZaposlenog(this);  
break;  
}  
} catch (Exception ex) {  
ex.printStackTrace();  
response.setException(ex);  
}  
sender.send(response);  
} catch (Exception ex) {  
Logger.getLogger(ProcessClientsRequests.class.getName()).log(Level.SEVERE, null, ex);  
}System.out.println("PREKIDA SE KLIJENTSKA NIT");  
}  
}

public void ugasiNit() {  
state=true;  
try {  
socket.close();  
} catch (IOException ex) {  
System.out.println("Gasi se klijentska nit!");  
}  
}

}

Класа “Operation” садржи следеће операције:

package rs.ac.bg.fon.ps.communication;

import java.io.Serializable;

/\*\*  
\*  
\* @author Lenovo  
\*/  
public enum Operation implements Serializable{  
ZAPOSLENI\_LOGIN,  
ADD\_PAKET,  
GET\_ALL\_PAKETI,  
OBRISI\_PAKET,  
IZMENI\_PAKET,  
VRATI\_PAKET,  
ADD\_PRIJAVA,  
GET\_ALL\_UREDJAJI,  
GET\_ALL\_KLIJENTI,  
ADD\_KLIJENT,  
OBRISI\_KLIJENTA,  
PRETRAGA\_KLIJENATA,  
VRATI\_AKTIVNE\_PRIJAVE,  
ODJAVI\_KORISNIKA,  
PRETRAGA\_PAKETA,  
ODJAVI\_ZAPOSLENOG;  
}

### 3.3.2. Контролер апликационе логике

Контролер апликационе логике прихвата захтеве за извршење системских операција и исте прослеђује до конкретне системске операције. Након извршења системске операције, контролер прихвата одговор и враћа назад позиваоцу (нити клијента).

Класа “Controller” као што је већ наведено, прихвата захтеве од нити клијената и прослеђује их до одговарајућих класа које су задужене за извршавање системских операција. За сваку системску операцију креирана је посебна класа која је задужена за реализацију самих операција.

Свака од системских операција, треба да наследи класу "АbstractGenericOperation" како би могла да се повеже са базом и како би се њено извршавање пратило као трансакција. Метода која обезбеђује наведене захтеве је метода *execute()* која даје кораке алгоритма извршавања сваке од системских операција.

### 3.3.3. Пројектовање структуре софтверског система

Свака класа има приватна поља атрибута, гетере и сетере за те атрибуте, беспараметарски и параметарски конструктор. Свака од ових класа имплементира интерфејс GenericEntity и његове методе, чиме је омогућено њихово чување у бази података помоћу генеричких метода класе “RepositoryDBGeneric”.

Приказан је интерфејс GenericEntity:

package rs.ac.bg.fon.ps.domain;

import java.io.Serializable;  
import java.sql.ResultSet;  
import java.util.List;

/\*\*  
\*  
\* @author Lenovo  
\*/  
public interface GenericEntity extends Serializable{

String getTableName();

String getColumnNamesForInsert();

String getInsertValues();

void setID(int id);

String getColumns();

String tableNameForGetAll();

public List<GenericEntity> getList(ResultSet rs) throws Exception;

String getUslovBrisanja();

String getPoljaIZmene();

GenericEntity getEntity(ResultSet rs) throws Exception;

String getUslovPretrage();  
}

Приказан је пример класе Paket:

package rs.ac.bg.fon.ps.domain;

import java.io.Serializable;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Objects;

/\*\*  
\*  
\* @author Lenovo  
\*/  
public class Paket implements GenericEntity {

private int paketID;  
private String nazivPaketa;  
private int brojSati;  
private double cena;  
private Uredjaj uredjaj;

public Paket() {  
}

public Paket(int paketID, String nazivPaketa, int brojSati, double cena, Uredjaj uredjaj) {  
this.paketID = paketID;  
this.nazivPaketa = nazivPaketa;  
this.brojSati = brojSati;  
this.cena = cena;

this.uredjaj = uredjaj;  
}

public int getPaketID() {  
return paketID;  
}

public void setPaketID(int paketID) {  
this.paketID = paketID;  
}

public String getNazivPaketa() {  
return nazivPaketa;  
}

public void setNazivPaketa(String nazivPaketa) {  
this.nazivPaketa = nazivPaketa;  
}

public int getBrojSati() {  
return brojSati;  
}

public void setBrojSati(int brojSati) {  
this.brojSati = brojSati;  
}

public double getCena() {  
return cena;  
}

public void setCena(double cena) {  
this.cena = cena;  
}

public Uredjaj getUredjaj() {  
return uredjaj;  
}

public void setUredjaj(Uredjaj uredjaj) {  
this.uredjaj = uredjaj;  
}

@Override  
public int hashCode() {  
int hash = 7;  
hash = 67 \* hash + this.paketID;  
hash = 67 \* hash + Objects.hashCode(this.nazivPaketa);  
hash = 67 \* hash + this.brojSati;  
hash = 67 \* hash + (int) (Double.doubleToLongBits(this.cena) ^ (Double.doubleToLongBits(this.cena) >>> 32));  
hash = 67 \* hash + Objects.hashCode(this.uredjaj);  
return hash;  
}

@Override  
public boolean equals(Object obj) {  
if (this == obj) {  
return true;  
}  
if (obj == null) {  
return false;  
}  
if (getClass() != obj.getClass()) {  
return false;  
}  
final Paket other = (Paket) obj;  
if (this.paketID != other.paketID) {  
return false;  
}  
return true;  
}

@Override  
public String toString() {  
return nazivPaketa;  
}

@Override  
public String getTableName() {  
return "paket";  
}

@Override  
public String getColumnNamesForInsert() {  
return "paketID, naziv, broj\_sati, cena, uredjajID";  
}

@Override  
public String getInsertValues() {  
StringBuilder sb = new StringBuilder();  
sb.append(paketID).append(", ").append("'").append(nazivPaketa).append("', ").append(brojSati)  
.append(", ").append(cena).append(", ")  
.append(uredjaj.getUredjajID());  
return sb.toString();  
}

@Override  
public void setID(int id) {  
paketID = id;  
}

@Override  
public String getColumns() {  
return "p.paketID as paketID,p.naziv as naziv, p.broj\_sati, p.cena, "  
+ " u.uredjajID as uredjajID, u.vrstaUredjaja as vrstaUredjaja, u.model, u.opis";  
}

@Override  
public String tableNameForGetAll() {  
return " paket p inner join uredjaj u on (p.uredjajID = u.uredjajID)";  
}

@Override  
public List<GenericEntity> getList(ResultSet rs) throws Exception {  
List<GenericEntity> listaPaketa = new ArrayList<>();  
while (rs.next()) {  
Paket paket = new Paket();  
paket.setPaketID(rs.getInt("paketID"));  
paket.setNazivPaketa(rs.getString("naziv"));  
paket.setBrojSati(rs.getInt("broj\_sati"));  
paket.setCena(rs.getDouble("cena"));  
Uredjaj uredjaj = new Uredjaj(rs.getInt("uredjajID"), rs.getString("vrstaUredjaja"), rs.getString("model"), rs.getString("opis"));  
paket.setUredjaj(uredjaj);  
listaPaketa.add(paket);  
}  
return listaPaketa;  
}

@Override  
public String getUslovBrisanja() {  
return "paketID=" + paketID;  
}

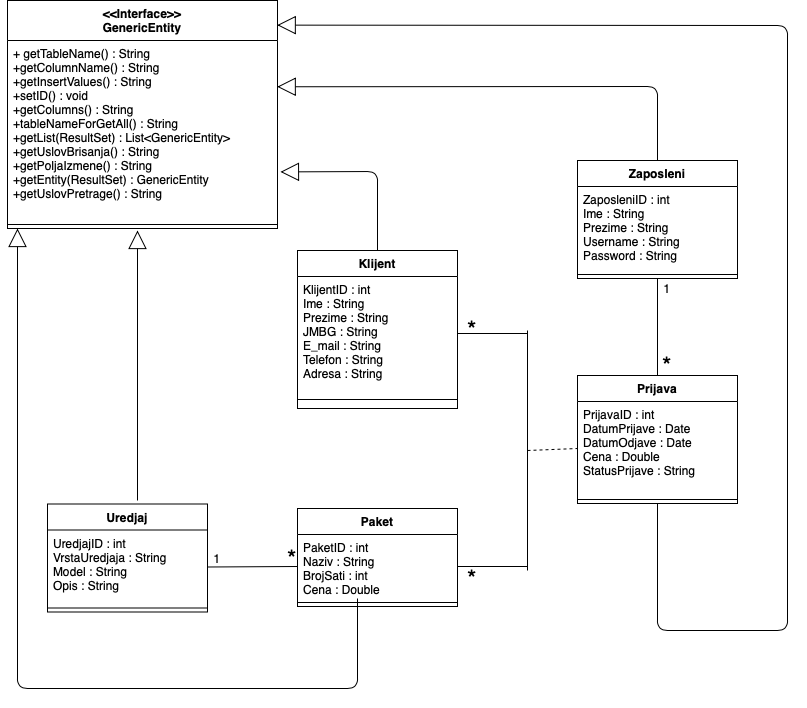
@Override  
public String getPoljaIZmene() {  
return "naziv= '" + getNazivPaketa() + "' ,"  
+ "broj\_sati=" + getBrojSati() + ", "  
+ "cena=" + getCena() + ", "  
+ "uredjajID=" + uredjaj.getUredjajID();  
}

@Override  
public GenericEntity getEntity(ResultSet rs) throws SQLException {  
Paket paket = new Paket();  
while (rs.next()) {  
paket.setPaketID(rs.getInt("paketID"));  
paket.setNazivPaketa(rs.getString("naziv"));  
paket.setBrojSati(rs.getInt("broj\_sati"));  
paket.setCena(rs.getDouble("cena"));  
Uredjaj uredjaj = new Uredjaj(rs.getInt("uredjajID"), rs.getString("vrstaUredjaja"), rs.getString("model"), rs.getString("opis"));  
paket.setUredjaj(uredjaj);  
}  
return paket;  
}

@Override  
public String getUslovPretrage() {  
return " p.uredjajID="+uredjaj.getUredjajID();  
}

}

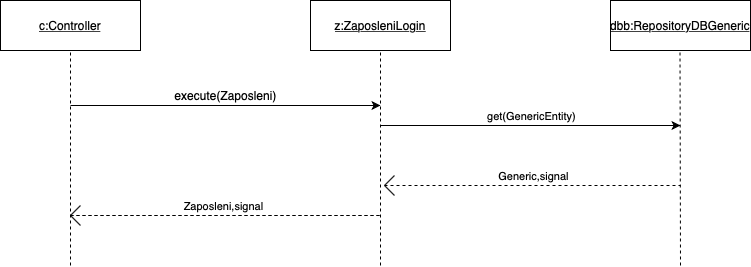
Концептуалне класе:



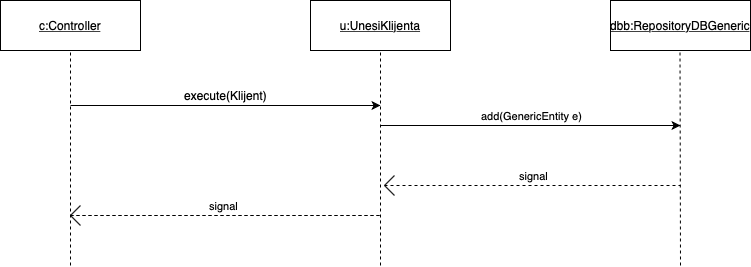
Слика

### 3.3.4. Пројектовање системских операција

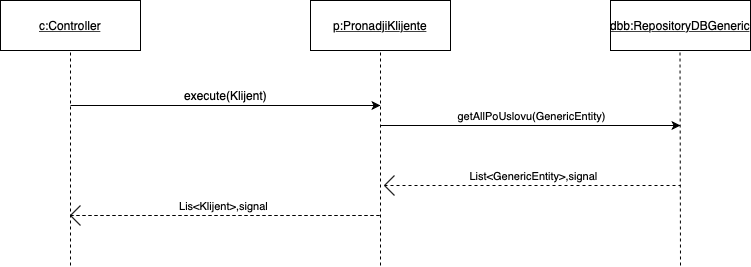
Свака од системских операција, треба да наследи класу "АbstractGenericOperation" како би могла да се повеже са базом и како би се њено извршавање пратило као трансакција. Метода која обезбеђује наведене захтеве је метода *execute()* која даје кораке алгоритма извршавања сваке од системских операција.

1. **Уговор УГ1: *ZaposleniLogin***  
   Операција: **ZaposleniLogin**(*Zaposleni*) : *signal*;  
   Веза са СК: СК1  
   Предуслови: /  
   Постуслови: /  
     
   

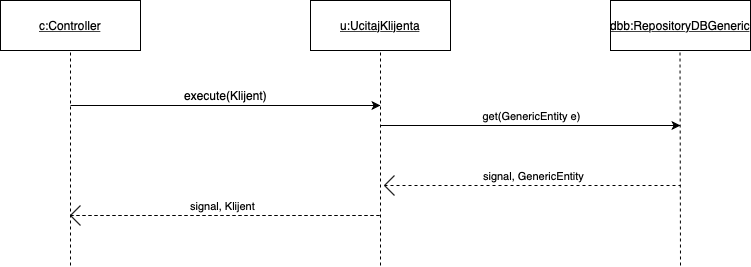
Слика

1. **Уговор УГ2: UnesiKlijenta**  
   Операција: **UnesiKlijenta**(*Klijent*) : *signal*;  
   Веза са СК: СК2  
   Предуслови:  
    -Вредносна ограничења над објектом *Klijent* морају бити задовољена.  
    -Структурна ограничења над објектом *Klijent* морају бити задовољена.  
   Постуслови:  
    -Запамћени су подаци о клијенту.  
     
   

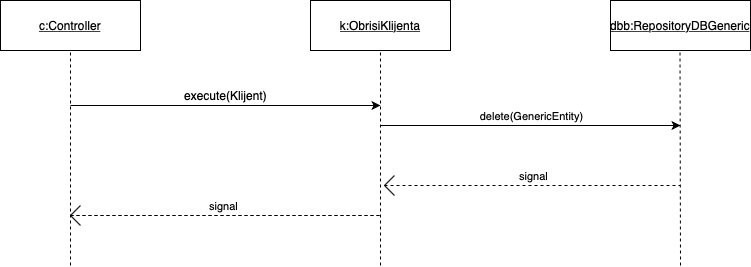
Слика

1. **Уговор УГ3: PronadjiKljentе**Операција: **PronadjiKljentе**(*KriterijumPretrage, List<Klijent)*: *signal*;  
   Веза са СК: СК3, СК4, СК8, СК9  
   Предуслови: /  
   Постуслови: /  
     
   

Слика

1. **Уговор УГ4: *UcitajKlijenta***  
   Операција: **UcitajKlijenta**(*Klijent*) : *signal*;  
   Веза са СК: СК3, СК4, СК8, СК9  
   Предуслови: /  
   Постуслови: /  
     
   

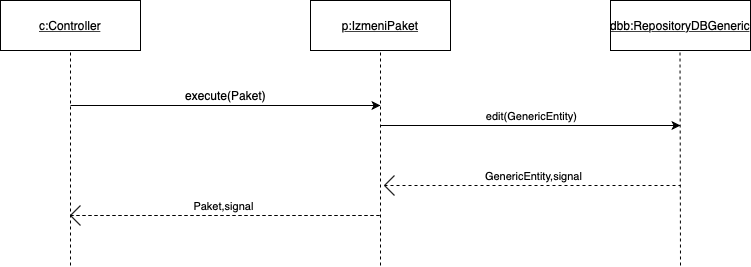
Слика

1. **Уговор УГ5: *ObrisiKlijenta***  
   Операција: **ObrisiKlijenta**(*Klijent*) : *signal*;  
   Веза са СК: СК3  
   Предуслови:   
    -Структурна ограничења над објектом *Klijent* морају бити задовољена.  
   Постуслови:   
    -Клијент је обрисан.  
     
   

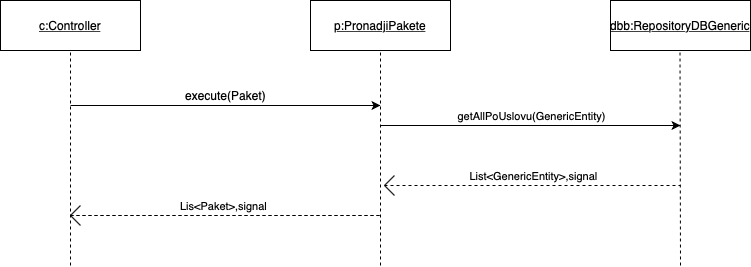
Слика

1. **Уговор УГ6: *ZapamtiPaket***  
   Операција: **ZapamtiPaket**(*Paket)* : *signal*;  
   Веза са СК: СК5, СК6  
   Предуслови:  
    -Вредносна ограничења над објектом *Paket* морају бити задовољена.  
    -Структурна ограничења над објектом *Paket* морају бити задовољена.  
   Постуслови:  
    -Сачувани су подаци о пакету.  
     
   

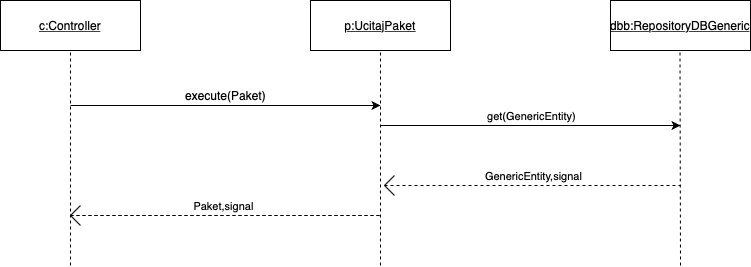
Слика

1. **Уговор УГ7: *IzmeniPaket***  
   Операција: **IzmeniPaket**(*Paket*) : *signal*;  
   Веза са СК: СК6, СК7, СК8  
   Предуслови: /  
   Постуслови: /  
   

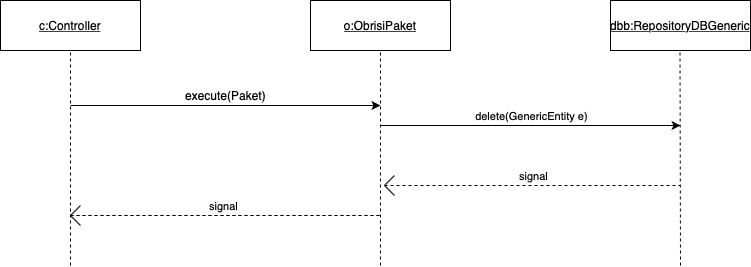
Слика

1. **Уговор УГ8: PronadjiPaketе**  
   Операција: **PronadjiPaketе**(*KriterijumPretrage, List<Paket>*): *signal*;  
   Веза са СК: СК3, СК4, СК8, СК9  
   Предуслови: /  
   Постуслови: /  
     
   

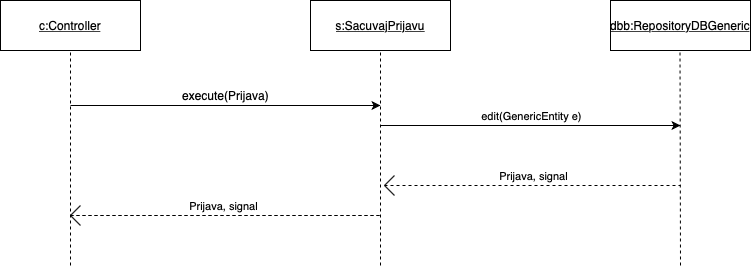
Слика

1. **Уговор УГ9: *UcitajPaket***Операција: **UcitajPaket**(*Paket*) : *signal*;  
   Веза са СК: СК6, СК7, СК8  
   Предуслови: /  
   Постуслови: /  
   

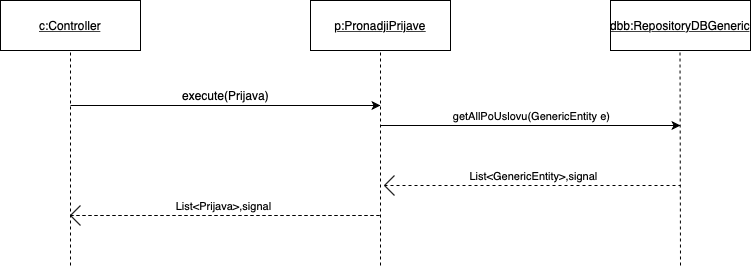
Слика

1. **Уговор УГ10: *ObrisiPaket***  
   Операција: **ObrisiPaket**(*Paket*) : *signal*;  
   Веза са СК: СК7  
   Предуслови:   
    -Структурна ограничења над објектом *Paket* морају бити задовољена.  
   Постуслови:   
    -Пакет је обрисан.  
     
   

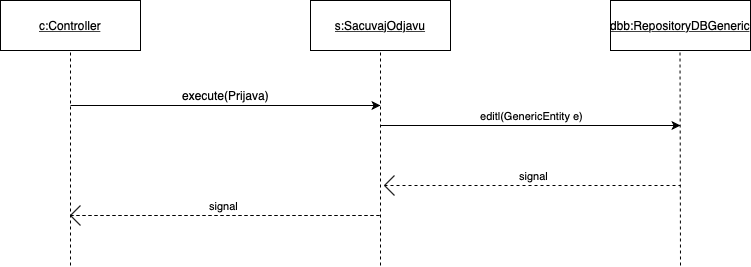
Слика

1. **Уговор УГ11: *SacuvajPrijavu***  
   Операција: **SacuvajPrijavu**(*Prijava*) : *signal*;  
   Веза са СК: СК8  
   Предуслови:  
    -Вредносна ограничења над објектом *Prijava* морају бити задовољена.  
    -Структурна ограничења над објектом *Prijava* морају бити задовољена.   
   Постуслови:  
    -Сачувани су подаци о новој пријави.  
     
   

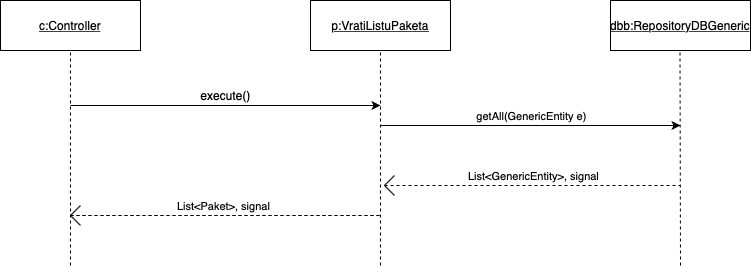
Слика

1. **Уговор УГ12: *PronadjiPrijave***Операција: **PronadjiPrijave**(*List<Prijava>*) : *signal*;  
   Веза са СК: СК9  
   Предуслови: /  
   Постуслови: /  
     
     
   

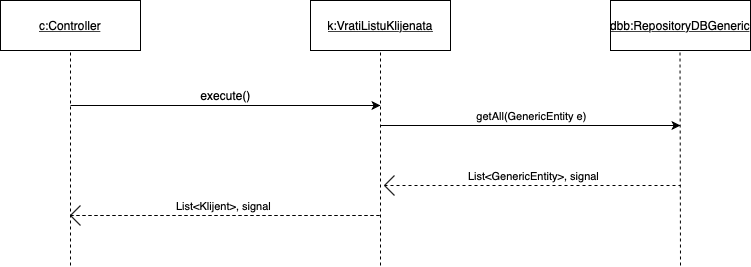
Слика

1. **Уговор УГ13: *SacuvajOdjavu***  
   Операција: **SacuvajOdjavu**(*Prijava*) : *signal*;  
   Веза са СК: СК9  
   Предуслови:  
    -Вредносна ограничења над објектом *Prijava* морају бити задовољена.  
    -Структурна ограничења над објектом *Prijava* морају бити задовољена.   
   Постуслови:  
    -Подаци о одјави су сачувани.  
     
   

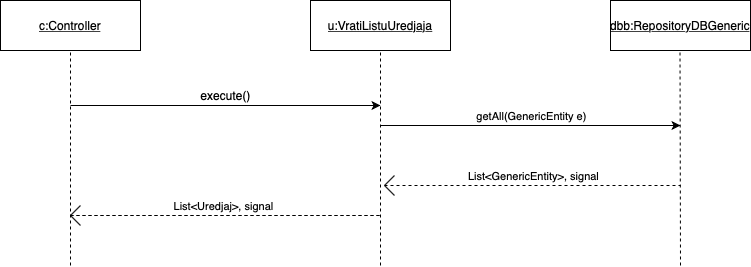
Слика

1. **Уговор УГ14: *VratiListuPaketa***  
    Операција: **VratiListuPaketa**(*List<Paket>*): *signal*;  
   Веза са СК: СК6, СК7, СК8, СК9  
   Предуслови: /  
   Постуслови: /  
     
   

Слика

1. **Уговор УГ15: *VratiListuKlijenata***Операција: **VratiListuKlijenata**(*List<Klijent>*): *signal*;  
   Веза са СК: СК3, СК4, СК8, СК9  
   Предуслови: /  
   Постуслови: /  
     
   

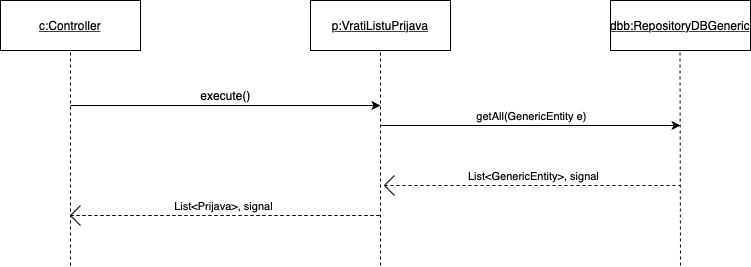
Слика

1. **Уговор УГ16: *VratiListuUredjaja***  
   Операција: **VratiListuUredjaja**(*List<Uredjaj>*): *signal*;  
   Веза са СК: СК5, СК6  
   Предуслови: /  
   Постуслови: /  
   

Слика

1. **Уговор УГ17: *VratiListuPrijava***

Операција: **VratiListuPrijava**(*List<Prijava>*): *signal*;  
Веза са СК: СК9   
Предуслови: /  
Постуслови: /

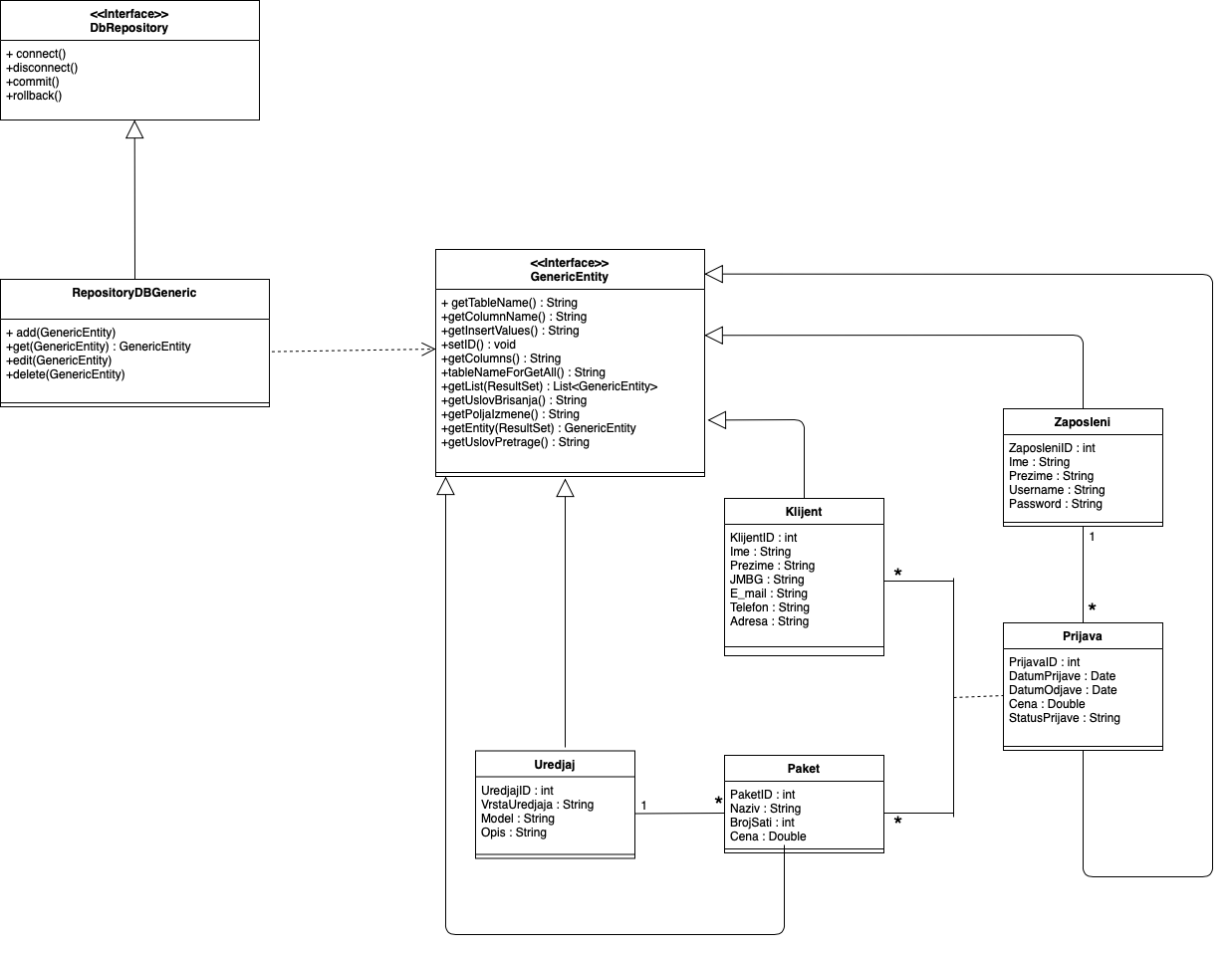


Слика

### 3.3.5. Брокер базе података

Класа "RepositoryDBGeneric" имплементира интерфејс DBRepository, који имплементира интерфејс “Repository”, као и све операције неопходне за рад са базом података и обезбеђује перзистентни сервис објектима доменских класа који се чувају у бази података.

Све методе “ RepositoryDBGeneric” класе су пројектоване као генеричке, што значи да могу да прихвате различите доменске објекте.

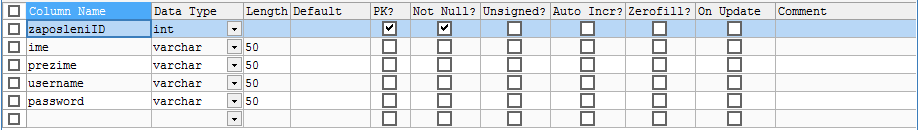


Слика : Концептуалне класе

## 3.5 Пројектовање складишта података

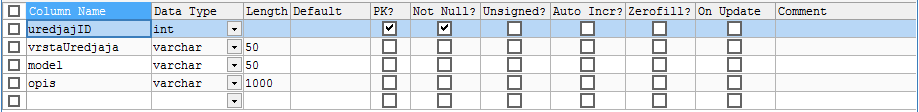
На основу доменских класа софтвера пројектоване су табеле (складишта података) релационог система за управљање базом података. У овом раду коришћен је MySQL систем за управљање базом података.

Табела запослени:



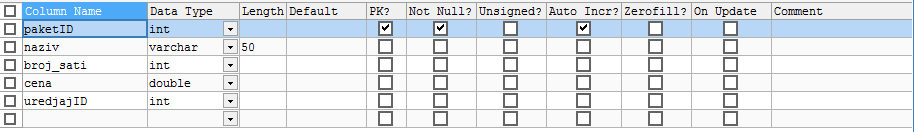
Слика

Табела уређај:



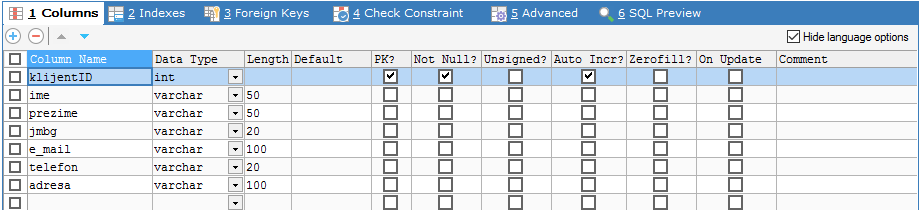
Слика

Табела пакет:



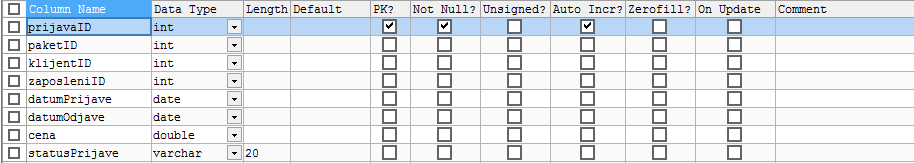
Слика

Табела клијент:



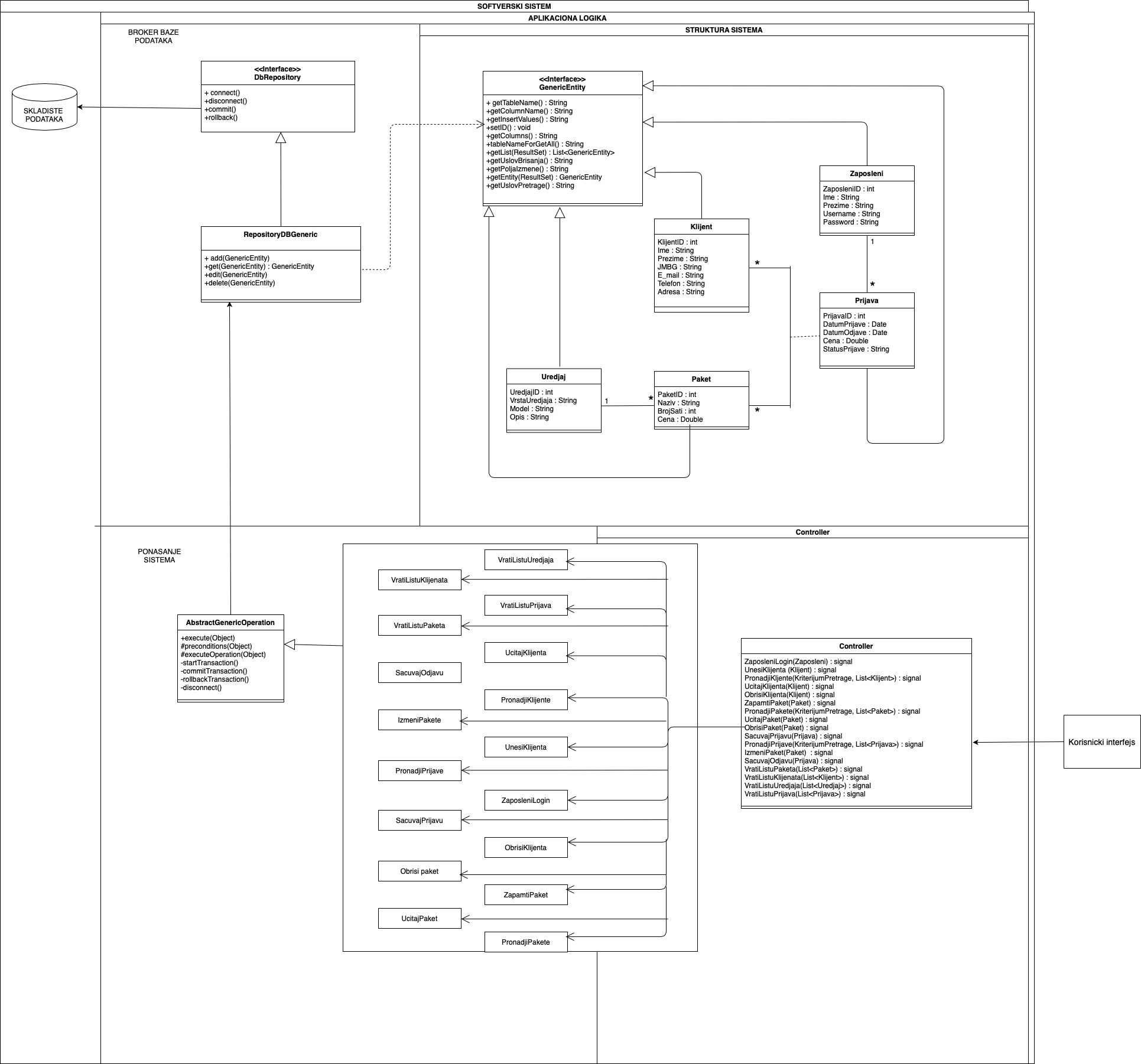
Слика

Табела пријава:



Слика

На основу претходних целина може се саставити цела архитектура софтверског система за праћење рада у гејминг центру.



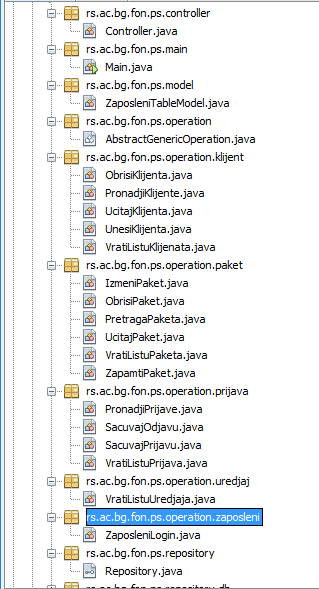
Слика : Архитектура софтверског система

# Имплементација

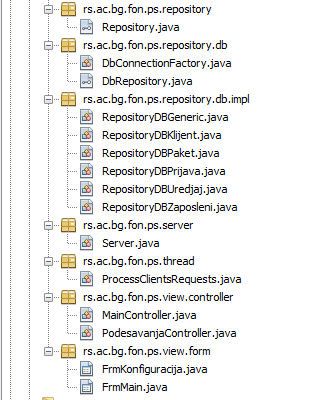
Систем је реализован у програмском језику Јava и пројектован је као клијент-сервер апликација. Развојно окружење које је коришћено је NetBeans IDE 8.2, а као систем за управљање базом података коришћен је MySQL. Софтверски систем је реализован у три пројекта: Projekat, ProjekatKlijent и ProjekatCommon.

Добијене су следеће класе:

Projekat:

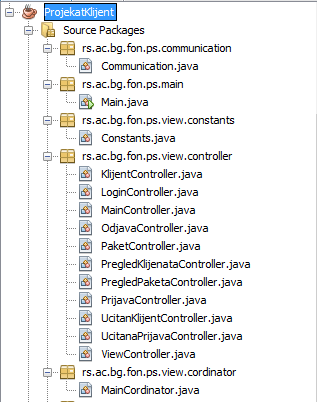


Слика

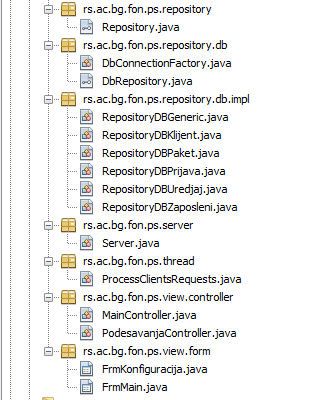


Слика

ProjekatKlijent:

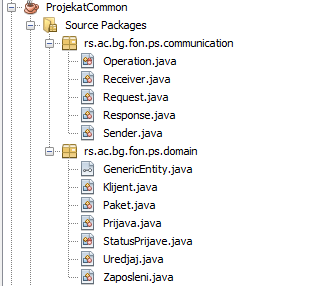


Слика



Слика

ProjekatCommon:



Слика

# Тестирање

Сваки од имплементираних случајева коришћења је тестиран. Покретањем апликације и уносом неисправних података, вршене су провере исправности имплементираних валидација. Поред неправилних, уношени су и правилни подаци, како би се тестирали сви приказани случајеви коришћења и потврдили представљени исходи. Уколико је дошло до одређеног недостатка приликом тестирања, они су уклоњени и исправљени.

# Литература

Др Синиша Влајић, Пројектовање софтвера, Београд, 2020