

Универзитет у Београду
Факултет организационих наука

Катедра за софтверско инжењерство
Предмет: Пројектовање софтвера

Семинарски рад

Тема: Развој софтверског система за управљање
аранжманима и путничким уговорима туристичке агенције у
Јава окружењу

Професор:
Проф. др Синиша Влајић

Ментор:
Доц. др Милош Милић

Студент:
Филип Гогих 1055/2016

Београд, 2021.

Садржај

1. Прикупљање корисничких захтева.....	4
1.1. Вербални опис	4
1.2. Случајеви коришћења.....	4
СК1: Случај коришћења - Креирање дестинације	5
СК2: Случај коришћења - Претраживање дестинација.....	6
СК3: Случај коришћења - Измена дестинације.....	7
СК4: Случај коришћења - Креирање аранжмана.....	8
СК5: Случај коришћења - Претраживање аранжмана	9
СК6: Случај коришћења - Измена аранжмана	10
СК7: Случај коришћења - Брисање аранжмана	11
СК8: Случај коришћења - Креирање уговора.....	12
СК9: Случај коришћења - Претраживање уговора	13
СК10: Случај коришћења - Сторнирање уговора	14
СК11: Случај коришћења - Претраживање клијената	15
2. Анализа	16
2.1. Системски дијаграми секвенци	16
ДС1: Дијаграм секвенце случаја коришћења - Креирање дестинације	16
ДС2: Дијаграм секвенце случаја коришћења - Претраживање дестинација.....	18
ДС3: Дијаграм секвенце случаја коришћења - Измена дестинације.....	19
ДС4: Дијаграм секвенце случаја коришћења - Креирање аранжмана.....	21
ДС5: Дијаграм секвенце случаја коришћења - Претраживање аранжмана	23
ДС6: Дијаграм секвенце случаја коришћења - Измена аранжмана	24
ДС7: Дијаграм секвенце случаја коришћења - Брисање аранжмана	26
ДС8: Дијаграм секвенце случаја коришћења - Креирање уговора.....	28
ДС9: Дијаграм секвенце случаја коришћења - Претраживање уговора	30
ДС10: Дијаграм секвенце случаја коришћења - Сторнирање уговора	31
ДС11: Дијаграм секвенце случаја коришћења - Претраживање клијената	33
2.2. Дефинисање уговора о системским операцијама	35
2.3. Структура софтверског система - Концептуални (доменски) модел	38
2.4. Структура софтверског система - Релациони модел.....	38
3. Пројектовање.....	42
3.1. Архитектура софтверског система	42

3.2.	Пројектовање корисничког интерфејса	42
3.2.1	Пројектовање екранских форми	43
СК1:	Случај коришћења - Креирање дестинације	45
СК2:	Случај коришћења - Претраживање дестинација	47
СК3:	Случај коришћења - Измена дестинације.....	49
СК4:	Случај коришћења - Креирање аранжмана.....	52
СК5:	Случај коришћења - Претраживање аранжмана	54
СК6:	Случај коришћења - Измена аранжмана	56
СК7:	Случај коришћења - Брисање аранжмана	59
СК8:	Случај коришћења - Креирање уговора.....	61
СК9:	Случај коришћења - Претраживање уговора	63
СК10:	Случај коришћења - Сторнирање уговора	65
СК11:	Случај коришћења - Претраживање клијената	67
3.3.	Пројектовање апликационе логике.....	69
3.3.1.	Комуникација са клијентима	69
3.3.2.	Контролер апликационе логике.....	70
3.3.3.	Пословна логика	71
3.3.4.	Брокер базе података	79
3.4.	Пројектовање складишта података	81
4.	Имплементација.....	84
5.	Тестирање	87
6.	Литература	87

1. Прикупљање корисничких захтева

1.1. Вербални опис

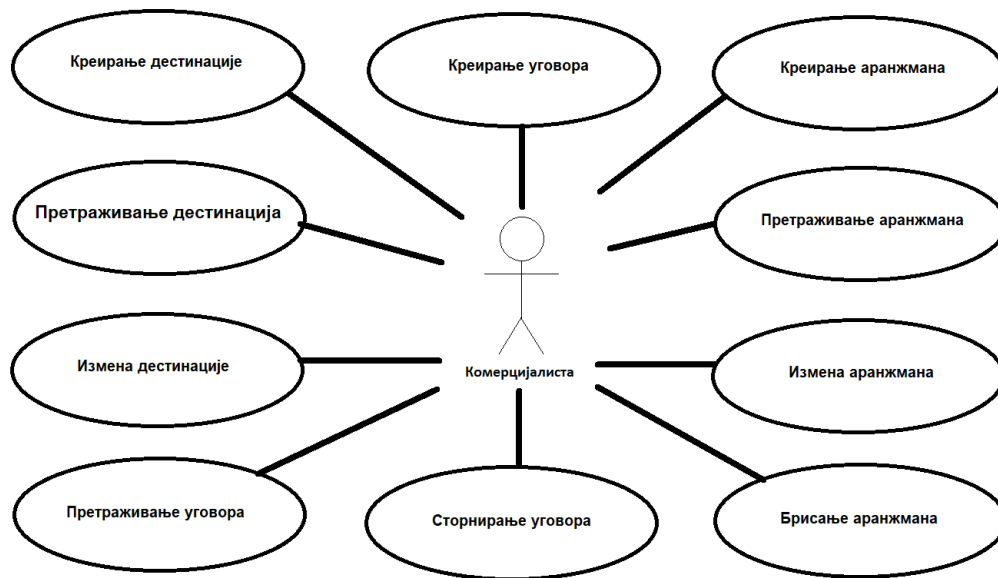
Потребно је направити апликацију која ће омогућити управљање аранжманима туристичке агенције за дестинације које се налазе у понуди, као и управљање уговорима пријављених путника за исте те аранжмане.

Приликом ширења пословања, нове дестинације је могуће **креирати** (уносити), док је ради прегледа целокупне понуде могуће **претражити** све дестинације. Сваке сезоне је потребно одредити смештајни капацитет дестинације, тако да је могуће и **изменити** дестинацију. Такође, омогућено је **креирање** и унос података нових аранжмана, као и **креирање** и унос нових уговора приликом склапања истих са путницима, за одговарајући аранжман. Аранжмане је по потреби могуће **претражити**, **изменити** њихове податке и **обрисати**. Путничке уговоре могуће је **претражити** и **сторнирати** - уколико се уговор раскине или је неопходно променити га (у том случају се постојећи уговор сторнира, а креира се нови, измењени).

1.2. Случајеви коришћења

У овој апликацији је идентификовано десет случајева коришћења:

1. Креирање дестинације
2. Претраживање дестинација
3. Измена дестинације
4. Креирање аранжмана
5. Претраживање аранжмана
6. Измена аранжмана
7. Брисање аранжмана
8. Креирање уговора (сложени СК)
9. Претраживање уговора
10. Сторнирање уговора
11. Претраживање клијената



Слика 1 Случајеви коришћења

СК1: Случај коришћења - Креирање дестинације

Назив СК

Креирање **дестинације**

Актери СК

Комерцијалиста

Учесници СК

Комерцијалиста и **систем** (програм)

Предуслов: Систем је укључен и **комерцијалиста** је улогован под својом шифром. Систем приказује форму за рад са **дестинацијом**.

Основни сценарио СК

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

Алтернативна сценарија

- 5.1 Уколико **систем** не може да креира **дестинацију** он приказује **комерцијалисти** поруку “**Систем** не може да креира **дестинацију**”. (ИА)

СК2: Случај коришћења - Претраживање дестинација

Назив СК

Претраживање дестинација

Актори СК

Комерцијалиста

Учесници СК

Комерцијалиста и систем (програм)

Предуслов: Систем је укључен и комерцијалиста је улогован под својом шифром. Систем приказује форму за рад са дестинацијом. Учитана је листа дестинација.

Основни сценарио СК

1. Комерцијалиста уноси вредност по којој претражује дестинације. (АПУСО)
2. Комерцијалиста позива систем да нађе дестинације по задатој вредности. (АПСО)
3. Систем тражи дестинације по задатој вредности. (СО)
4. Систем приказује комерцијалисти нађене дестинације. (ИА)

Алтернативна сценарија

- 4.1 Уколико систем не може да нађе дестинацију он приказује комерцијалисти поруку: "Систем не може да нађе дестинацију по задатој вредности". (ИА)

СК3: Случај коришћења - Измена дестинације

Назив СК

Измена дестинације

Актори СК

Комерцијалиста

Учесници СК

Комерцијалиста и систем (програм)

Предуслов: Систем је укључен и комерцијалиста је улогован под својом шифром. Систем приказује форму за рад са дестинацијом. Учитана је листа дестинација.

Основни сценарио СК

1. Комерцијалиста уноси вредност по којој претражује дестинацију. (АПУСО)
2. Комерцијалиста позива систем да нађе дестинацију по задатој вредности. (АПСО)
3. Систем тражи дестинације по задатој вредности. (СО)
4. Систем приказује комерцијалисти дестинације. (ИА)
5. Комерцијалиста бира дестинацију чије податке жели да измени. (АПУСО)
6. Комерцијалиста позива систем да учита податке о дестинацији. (АПСО)
7. Систем учитава изабрану дестинацију. (СО)
8. Систем приказује комерцијалисти изабрану дестинацију. (ИА)
9. Комерцијалиста уноси (мења) податке о дестинацији. (АПУСО)
10. Комерцијалиста контролише да ли је коректно унео податке о дестинацији. (АНСО)
11. Комерцијалиста позива систем да запамти податке о дестинацији. (АПСО)
12. Систем памти податке о дестинацији. (СО)
13. Систем приказује комерцијалисти запамћену дестинацију и поруку: "Систем је запамтио дестинацију." (ИА)

Алтернативна сценарија

- 4.1 Уколико систем не може да нађе дестинацију он приказује комерцијалисти поруку: "Систем не може да нађе дестинацију по задатој вредности". Прекида се извршење сценарија. (ИА)
- 13.1 Уколико систем не може да запамти податке о дестинацији он приказује комерцијалисти поруку "Систем не може да запамти дестинацију". (ИА)

СК4: Случај коришћења - Креирање аранжмана

Назив СК

Креирање аранжмана

Актери СК

Комерцијалиста

Учесници СК

Комерцијалиста и систем (програм)

Предуслов: Систем је укључен и комерцијалиста је улогован под својом шифром. Систем приказује форму за рад са аранжманом.

Основни сценарио СК

1. Комерцијалиста уноси податке у аранжман. (АПУСО)
2. Комерцијалиста контролише да ли је коректно унео податке у аранжман. (АНСО)
3. Комерцијалиста позива систем да креира аранжман. (АПСО)
4. Систем креира аранжман. (СО)
5. Систем приказује комерцијалисти креиран аранжман и поруку: "Систем је креирао аранжман". (ИА)

Алтернативна сценарија

5.1 Уколико систем не може да креира аранжман он приказује комерцијалисти поруку: "Систем не може да креира аранжман". Прекида се извршење сценарија. (ИА)

СК5: Случај коришћења - Претраживање аранжмана

Назив СК

Претраживање аранжмана

Актери СК

Комерцијалиста

Учесници СК

Комерцијалиста и систем (програм)

Предуслов: Систем је укључен и комерцијалиста је улогован под својом шифром. Систем приказује форму за рад са аранжманом. Учитана је листа аранжмана.

Основни сценарио СК

1. Комерцијалиста уноси вредност по којој претражује аранжмане. (АПУСО)
2. Комерцијалиста позива систем да нађе аранжмане по задатој вредности. (АПСО)
3. Систем тражи аранжмане по задатој вредности. (СО)
4. Систем приказује комерцијалисти нађене аранжмане. (ИА)

Алтернативна сценарија

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

СК6: Случај коришћења - Измена аранжмана

Назив СК

Измена аранжмана

Актери СК

Комерцијалиста

Учесници СК

Комерцијалиста и систем (програм)

Предуслов: Систем је укључен и комерцијалиста је улогован под својом шифром. Систем приказује форму за рад са аранжманом. Учитана је листа аранжмана.

Основни сценарио СК

1. Комерцијалиста уноси вредност по којој претражује аранжмане. (АПУСО)
2. Комерцијалиста позива систем да нађе аранжмане по задатој вредности. (АПСО)
3. Систем тражи аранжмане по задатој вредности. (СО)
4. Систем приказује комерцијалисти нађене аранжмане. (ИА)
5. Комерцијалиста бира аранжман чије податке жели да измени. (АПУСО)
6. Комерцијалиста позива систем да учита податке о аранжману. (АПСО)
7. Систем учитава изабран аранжман. (СО)
8. Систем приказује комерцијалисти изабран аранжман. (ИА)
9. Комерцијалиста уноси (мења) податке о аранжману. (АПУСО)
10. Комерцијалиста контролише да ли је коректно унео податке о аранжману. (АНСО)
11. Комерцијалиста позива систем да запамти податке о аранжману. (АПСО)
12. Систем памти податке о аранжману. (СО)
13. Систем приказује комерцијалисти запамћени аранжман и поруку: "Систем је запамтио аранжман." (ИА)

Алтернативна сценарија

- 4.1 Уколико систем не може да нађе аранжман он приказује комерцијалисти поруку: "Систем не може да нађе аранжман по задатој вредности". Прекида се извршење сценарија. (ИА)
- 13.1 Уколико систем не може да запамти податке о аранжману он приказује комерцијалисти поруку "Систем не може да запамти аранжман". (ИА)

СК7: Случај коришћења - Брисање аранжмана

Назив СК

Брисање аранжмана

Актори СК

Комерцијалиста

Учесници СК

Комерцијалиста и систем (програм)

Предуслов: Систем је укључен и комерцијалиста је улогован под својом шифром. Систем приказује форму за рад са аранжманом. Учитана је листа аранжмана.

Основни сценарио СК

1. Комерцијалиста уноси вредност по којој претражује аранжмане. (АПУСО)
2. Комерцијалиста позива систем да нађе аранжмане по задатој вредности. (АПСО)
3. Систем тражи аранжмане по задатој вредности. (СО)
4. Систем приказује комерцијалисти нађене аранжман. (ИА)
5. Комерцијалиста бира аранжман чије податке жели да обрише. (АПУСО)
6. Комерцијалиста позива систем да обрише аранжман. (АПСО)
7. Систем брише аранжман. (СО)
8. Систем приказује комерцијалисти поруку: "Систем је обрисао аранжман." (ИА)

Алтернативна сценарија

- 4.1 Уколико систем не може да нађе аранжман он приказује комерцијалисти поруку: "Систем не може да нађе аранжман по задатој вредности". Прекида се извршење сценарија. (ИА)
- 8.1 Уколико систем не може да обрише аранжман он приказује комерцијалисти поруку "Систем не може да обрише аранжман". (ИА)

СК8: Случај коришћења - Креирање уговора

Назив СК

Креирање уговора

Актори СК

Комерцијалиста

Учесници СК

Комерцијалиста и систем (програм)

Предуслов: Систем је укључен и комерцијалиста је улогован под својом шифром. Систем приказује форму за рад са уговором. Учитана је листа аранжмана.

Основни сценарио СК

1. Комерцијалиста уноси податке у уговор. (АПУСО)
2. Комерцијалиста контролише да ли је коректно унео податке у уговор. (АНСО)
3. Комерцијалиста позива систем да обради уговор. (АПСО)
4. Систем обрађује уговор. (СО)
5. Систем приказује комерцијалисти обрађен уговор и поруку: "Систем је обрадио уговор". (ИА)

Алтернативна сценарија

- 5.1 Уколико систем не може да обради уговор он приказује комерцијалисти поруку: "Систем не може да обради уговор". (ИА)

СК9: Случај коришћења - Претраживање уговора

Назив СК

Претраживање уговора

Актори СК

Комерцијалиста

Учесници СК

Комерцијалиста и систем (програм)

Предуслов: Систем је укључен и комерцијалиста је улогован под својом шифром. Систем приказује форму за рад са уговором.

Основни сценарио СК

1. Комерцијалиста уноси вредност по којој претражује уговор. (АПУСО)
2. Комерцијалиста позива систем да нађе уговор по задатој вредности. (АПСО)
3. Систем тражи уговор по задатој вредности. (СО)
4. Систем приказује комерцијалисти податке о уговору. (ИА)

Алтернативна сценарија

- 4.1 Уколико систем не може да нађе уговор он приказује комерцијалисти поруку: "Систем не може да нађе уговор по задатој вредности". (ИА)

СК10: Случај коришћења - Сторнирање уговора

Назив СК

Сторнирање уговора

Актори СК

Комерцијалиста

Учесници СК

Комерцијалиста и систем (програм)

Предуслов: Систем је укључен и комерцијалиста је улогован под својом шифром. Систем приказује форму за рад са уговором.

Основни сценарио СК

1. Комерцијалиста уноси вредност по којој претражује уговор. (АПУСО)
2. Комерцијалиста позива систем да нађе уговор по задатој вредности. (АПСО)
3. Систем тражи уговор по задатој вредности. (СО)
4. Систем приказује комерцијалисти уговор. (ИА)
5. Комерцијалиста позива систем да сторнира задати уговор. (АПСО)
6. Систем сторнира уговор. (СО)
7. Систем приказује комерцијалисти сторниран уговор и поруку: "Систем је сторнирао уговор". (ИА)

Алтернативна сценарија

- 4.1 Уколико систем не може да нађе уговор он приказује комерцијалисти поруку: "Систем не може да нађе уговор по задатој вредности". Прекида се извршење сценарија. (ИА)
- 7.1 Уколико систем не може да сторнира уговор он приказује комерцијалисти поруку: "Систем не може да сторнира уговор".

СК11: Случај коришћења - Претраживање клијената

Назив СК

Претраживање клијената

Актори СК

Комерцијалиста

Учесници СК

Комерцијалиста и систем (програм)

Предуслов: Систем је укључен и комерцијалиста је улогован под својом шифром. Систем приказује форму за рад са клијентима.

Основни сценарио СК

1. Комерцијалиста уноси вредност по којој претражује клијента. (АПУСО)
2. Комерцијалиста позива систем да нађе клијента по задатој вредности. (АПСО)
3. Систем тражи клијента по задатој вредности. (СО)
4. Систем приказује комерцијалисти податке о клијенту и поруку: “Систем је нашао клијента по задатој вредности”. (ИА)

Алтернативна сценарија

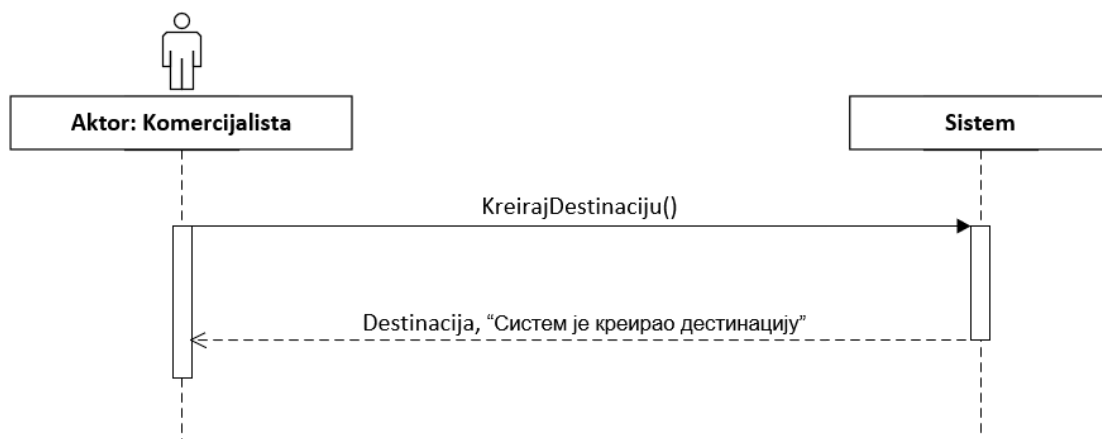
- 4.1 Уколико систем не може да нађе клијента он приказује комерцијалисти поруку: “Систем не може да нађе клијента по задатој вредности”. (ИА)

2. Анализа

2.1. Системски дијаграми секвенци

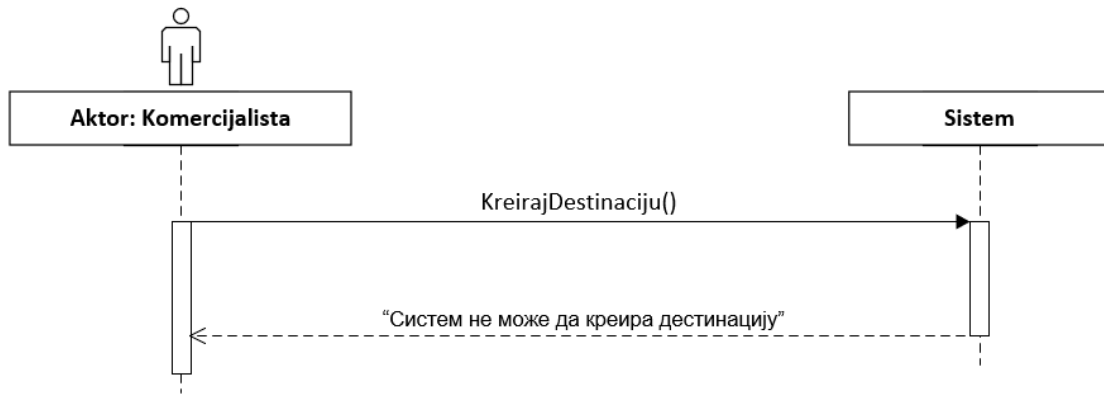
ДС1: Дијаграм секвенце случаја коришћења - Креирање дестинације

1. **Комерцијалиста** **позива** **систем** да креира **дестинацију**. (АПСО)
2. **Систем** **приказује** **комерцијалисти** **дестинацију** и поруку: **"Систем је креирао дестинацију"**. (ИА)



Алтернативна сценарија

2.1 Уколико **систем** не може да креира **дестинацију** он приказује **комерцијалисти** поруку: “**Систем** не може да креира **дестинацију**”. Прекида се извршење сценарија. (ИА)

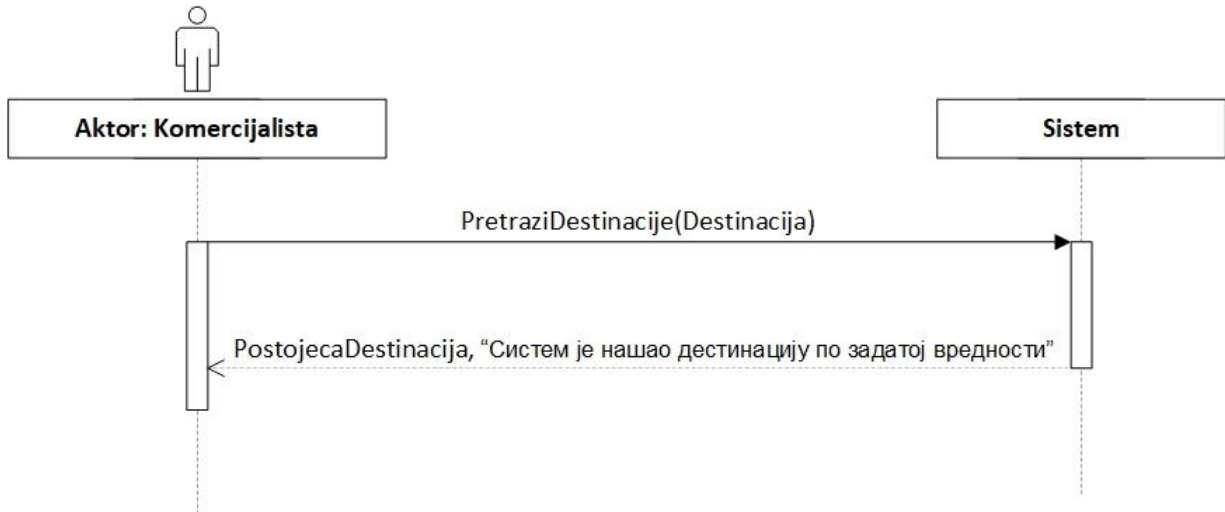


Са наведених секвенцих дијаграма уочава се 1 системска операција које треба пројектовати:

1. signal **KreirajDestinaciju** (Destinacija);

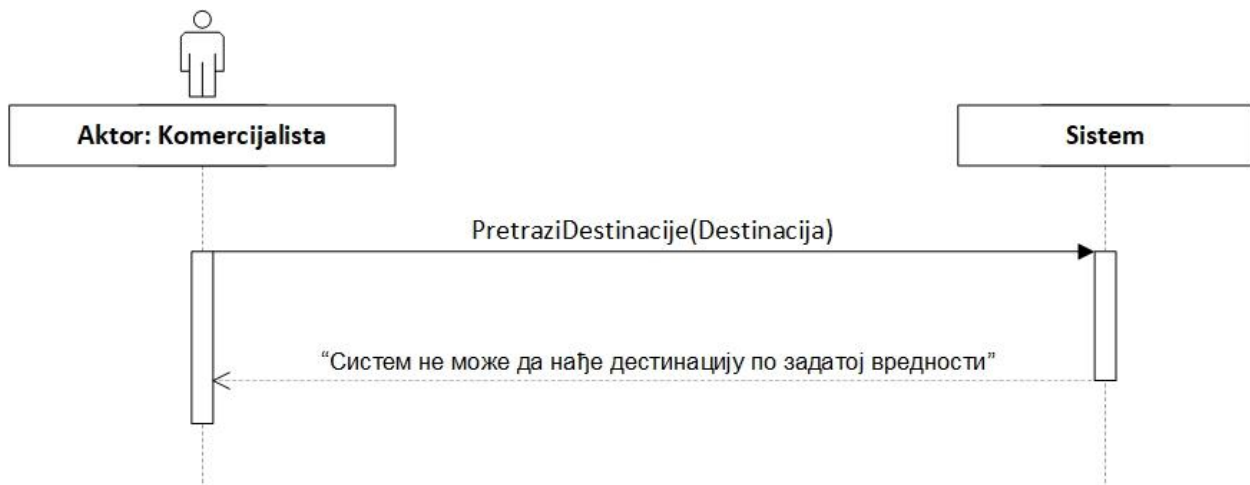
ДС2: Дијаграм секвенце случаја коришћења - Претраживање дестинација

1. **Комерцијалиста** **позива** **систем** да нађе **дестинацију** по задатој вредности. (АПСО)
2. **Систем** приказује **комерцијалисти** податке о **дестинацији** и поруку: "**Систем** је нашао **дестинацију** по задатој вредности". (ИА)



Алтернативна сценарија

- 2.1 Уколико **систем** не може да нађе **дестинацију** он приказује **комерцијалисти** поруку: "**Систем** не може да нађе **дестинацију** по задатој вредности". (ИА)

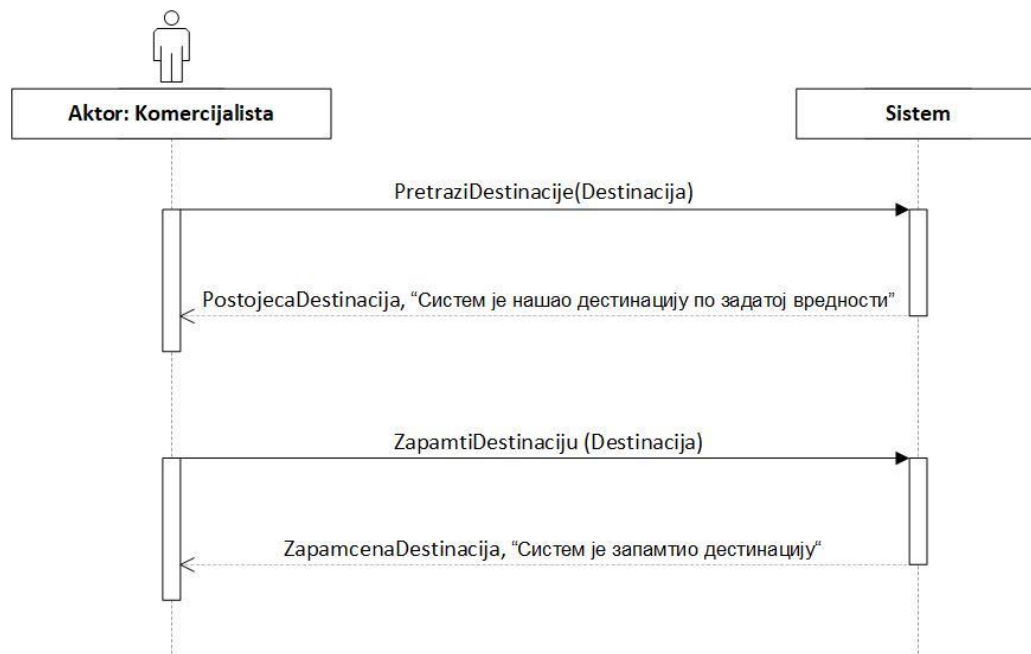


Са наведених секвенцих дијаграма уочава се 1 системска операција коју треба пројектовати:

1. signal **PretraziDestinacije** (Destinacija);

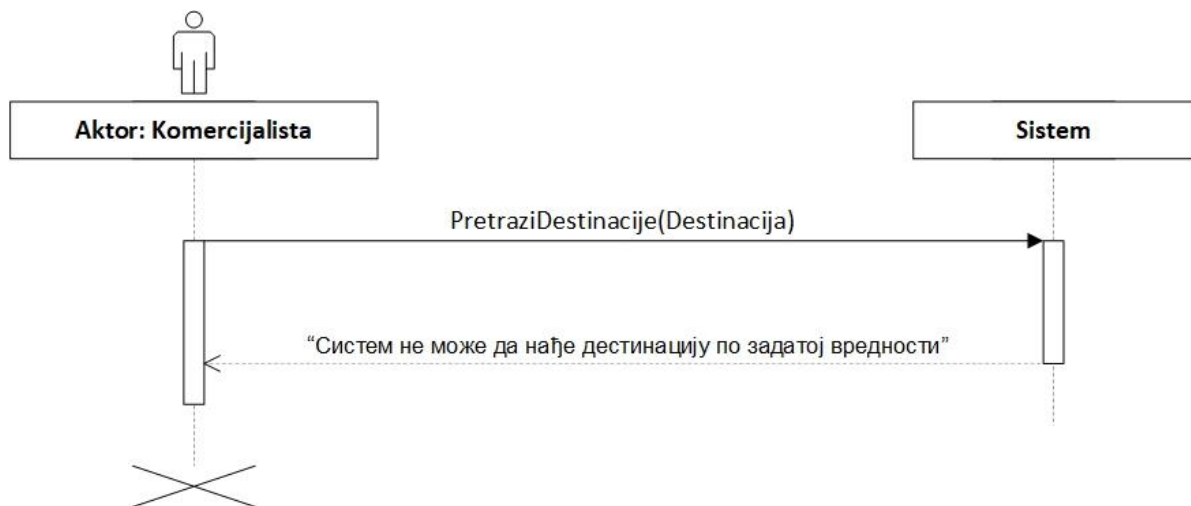
ДСЗ: Дијаграм секвенце случаја коришћења - Измена дестинације

1. **Комерцијалиста** **позива** **систем** да нађе **дестинацију** по задатој вредности. (АПСО)
2. **Систем** приказује **комерцијалисти** **дестинацију** и поруку: “**Систем** је нашао **дестинацију** по задатој вредности”. (ИА)
3. **Комерцијалиста** **позива** **систем** да запамти податке о **дестинацији**. (АПСО)
4. **Систем** **приказује** **комерцијалисти** запамћену **дестинацију** и поруку: “**Систем** је запамтио **дестинацију**.” (ИА)

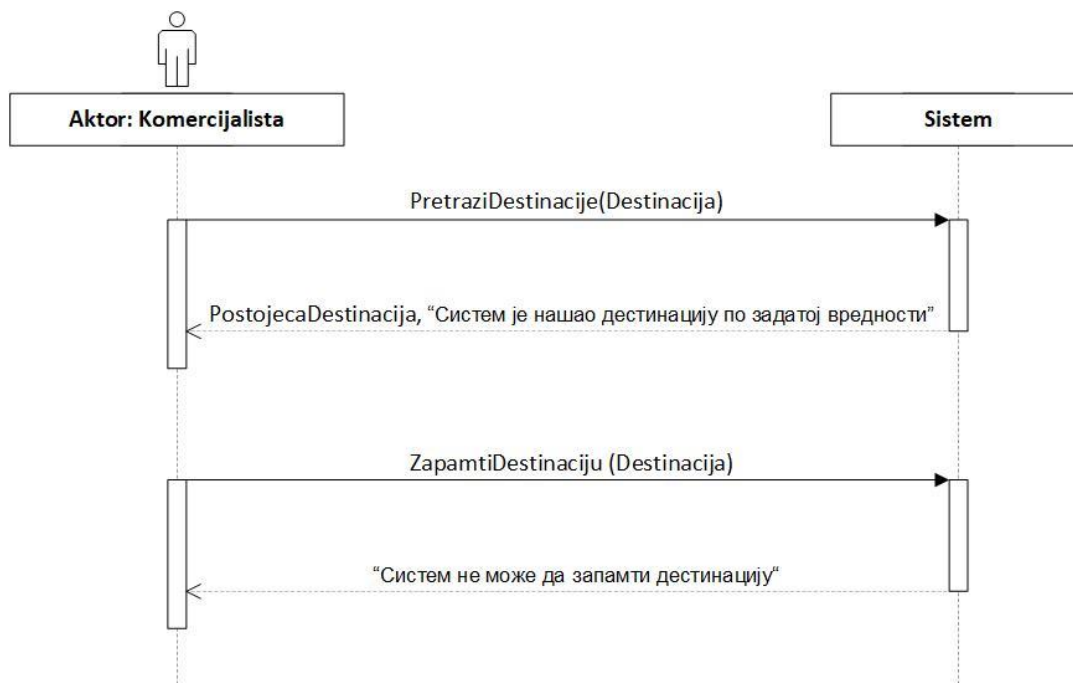


Алтернативна сценарија

- 2.1 Уколико **систем** не може да нађе **дестинацију** он приказује **комерцијалисти** поруку: “**Систем** не може да нађе **дестинацију** по задатој вредности”. Прекида се извршење сценарија. (ИА)



4.1 Уколико **систем** не може да запамти податке о **дестинацији** он приказује **комерцијалисти** поруку “**Систем** не може да запамти **дестинацију**”. (ИА)

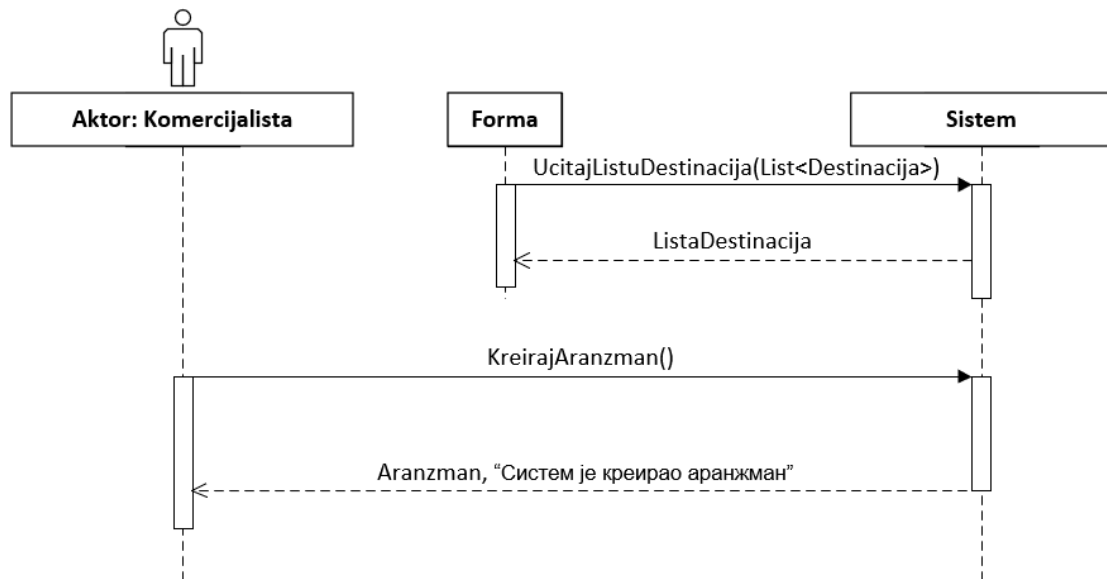


Са наведених секвенцих дијаграма уочавају се 2 системске операције које треба пројектовати:

1. signal **PretraziDestinacije** (Destinacija);
2. signal **ZapamtiDestinaciju** (Destinacija);

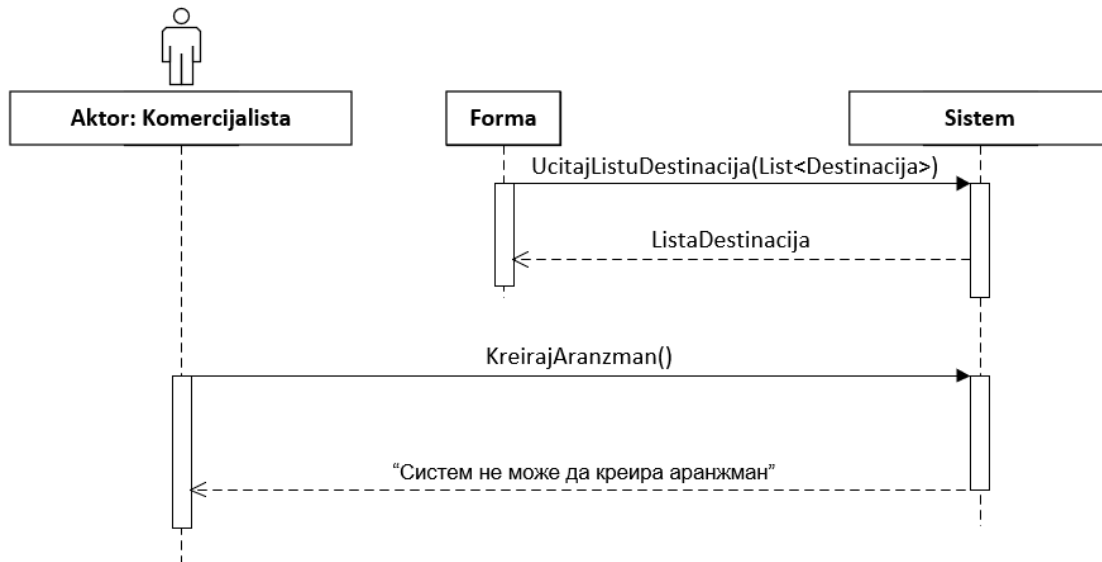
ДС4: Дијаграм секвенце случаја коришћења - Креирање аранжмана

1. **Форма** **позива** **систем** да учита **листу дестинација**. (АПСО)
2. **Систем** **враћа** **форми** **листу дестинација**. (ИА)
3. **Комерцијалиста** **позива** **систем** да креира **аранжман**. (АПСО)
4. **Систем** **приказује** **комерцијалисти** **аранжман** и поруку: **"Систем је креирао аранжман"**. (ИА)



Алтернативна сценарија

- 4.1 Уколико **систем** не може да креира **аранжман** он приказује **комерцијалисти** поруку: “**Систем** не може да креира **аранжман**”. (ИА)

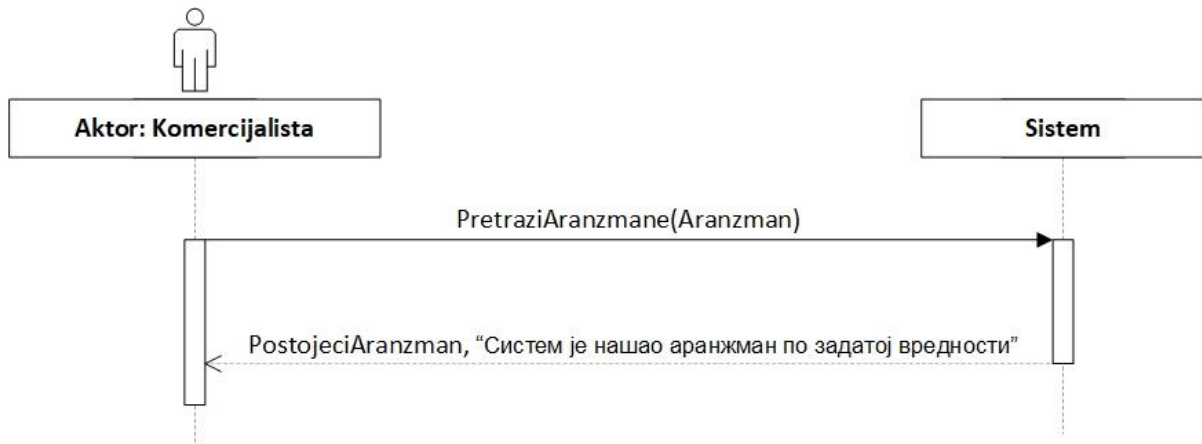


Са наведених секвенчних дијаграма уочавају се 2 системске операције које треба пројектовати:

1. signal **UcitajListuDestinacija** (List <Destinacija>)
2. signal **KreirajAranzman** (Aranzman);

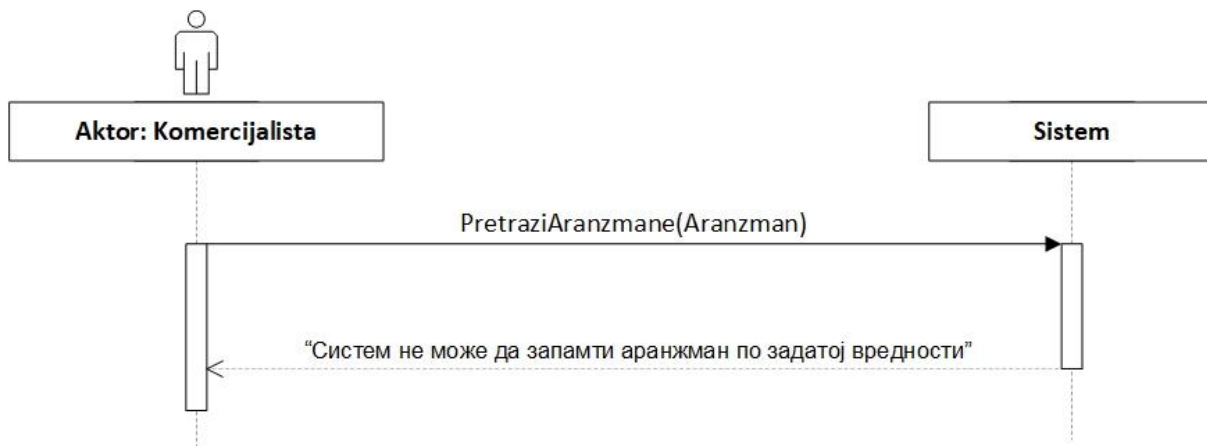
ДС5: Дијаграм секвенце случаја коришћења - Претраживање аранжмана

1. **Комерцијалиста** **позива** **систем** да нађе **аранжман** по задатој вредности. (АПСО)
2. **Систем** приказује **комерцијалисти** податке о **аранжману** и поруку: "**Систем** је нашао **аранжман** по задатој вредности". (ИА)



Алтернативна сценарија

- 2.1 Уколико **систем** не може да нађе **аранжман** он приказује **комерцијалисти** поруку: "**Систем** не може да нађе **аранжман** по задатој вредности". (ИА)

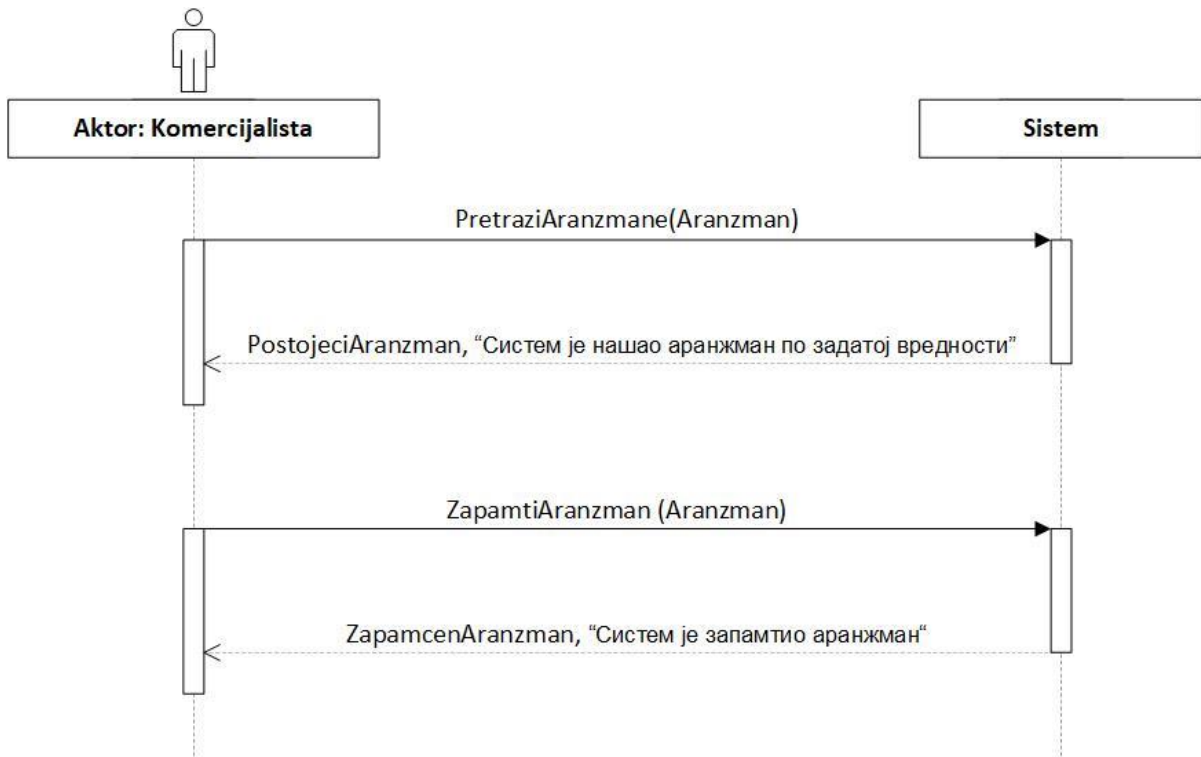


Са наведених секвенцих дијаграма уочава се 1 системска операција коју треба пројектовати:

1. signal **PretraziAranzmane** (Aranzman);

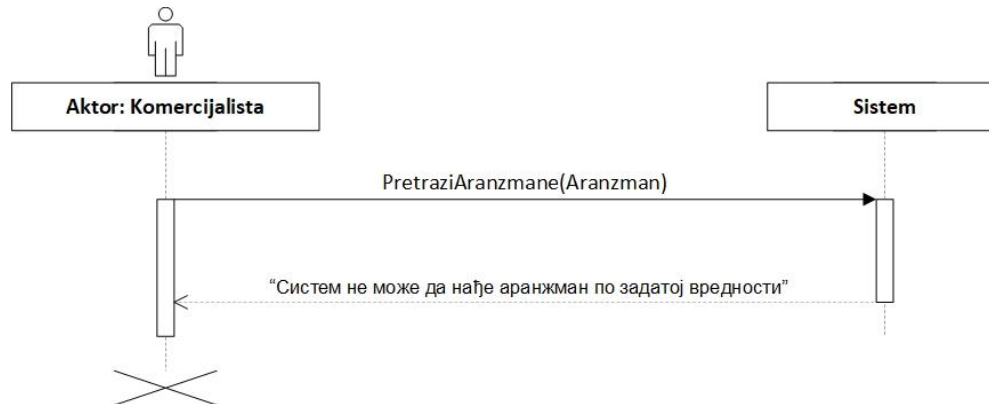
ДС6: Дијаграм секвенце случаја коришћења - Измена аранжмана

1. **Комерцијалиста** **позива** **систем** да нађе **аранжман** по задатој вредности. (АПСО)
2. **Систем** приказује **комерцијалисти** **аранжман** и поруку: “**Систем** је нашао **аранжман** по задатој вредности”. (ИА)
3. **Комерцијалиста** **позива** **систем** да запамти податке о **аранжману**. (АПСО)
4. **Систем** **приказује** **комерцијалисти** запамћени **аранжман** и поруку: “**Систем** је запамтио **аранжман**.” (ИА)

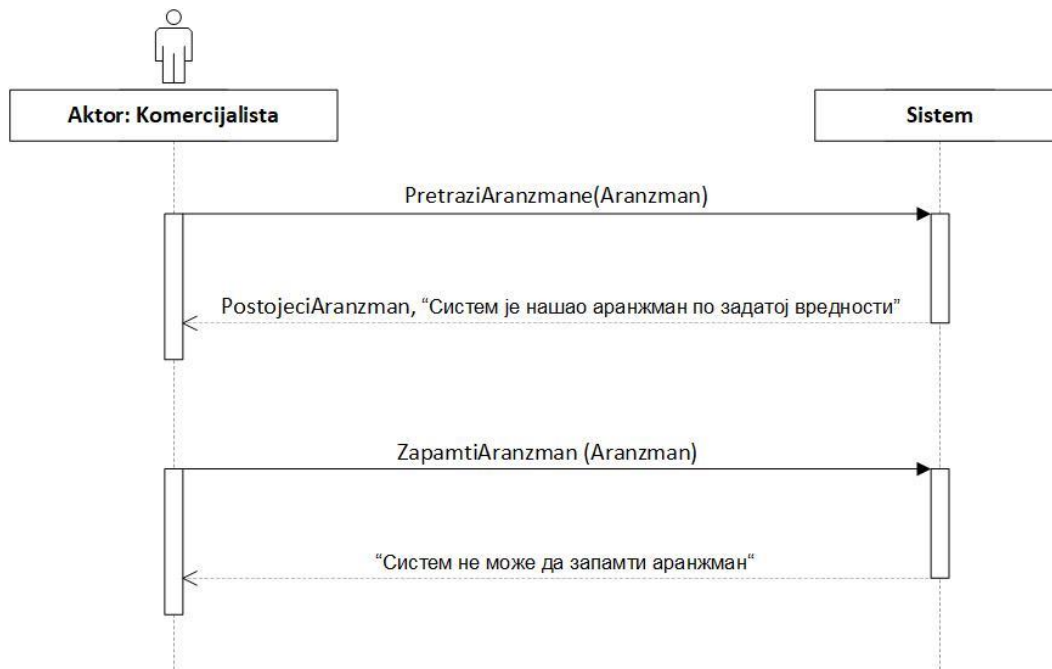


Алтернативна сценарија

2.1 Уколико **систем** не може да нађе **аранжман** он приказује **комерцијалисти** поруку: “**Систем** не може да нађе **аранжман** по задатој вредности”. Прекида се извршење сценарија. (ИА)



4.1 Уколико **систем** не може да запамти податке о **аранжману** он приказује **комерцијалисти** поруку “**Систем** не може да запамти **аранжман**”. (ИА)

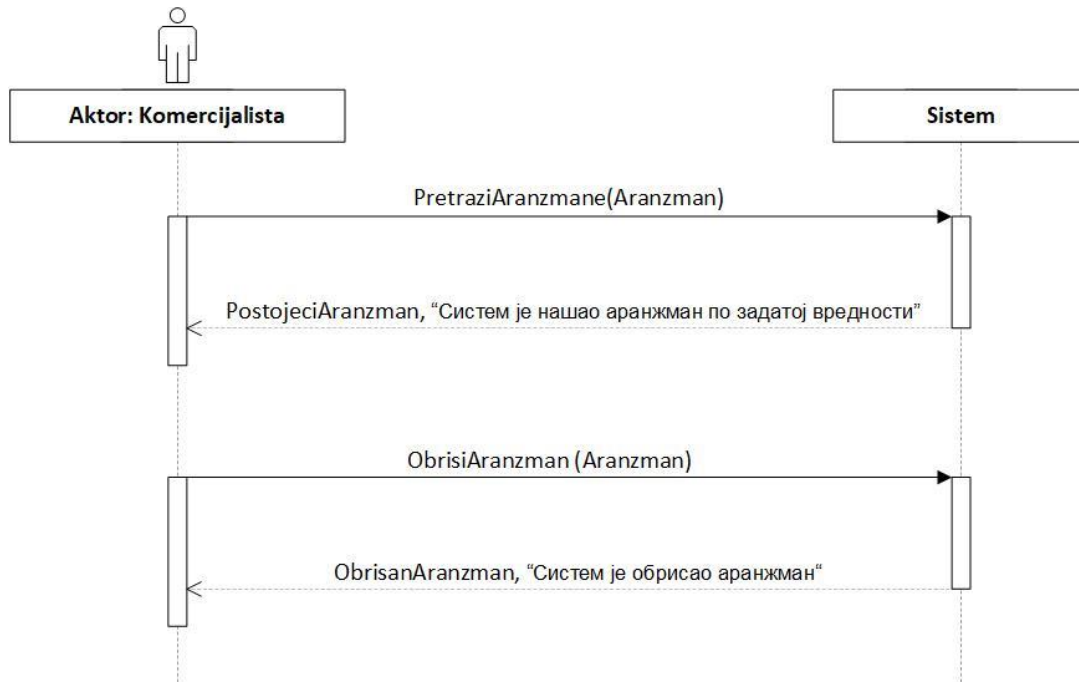


Са наведених секвенцих дијаграма уочавају се 2 системске операције које треба пројектовати:

1. signal **PretraziAranzmane** (Aranzman);
2. signal **ZapamtiAranzman** (Aranzman);

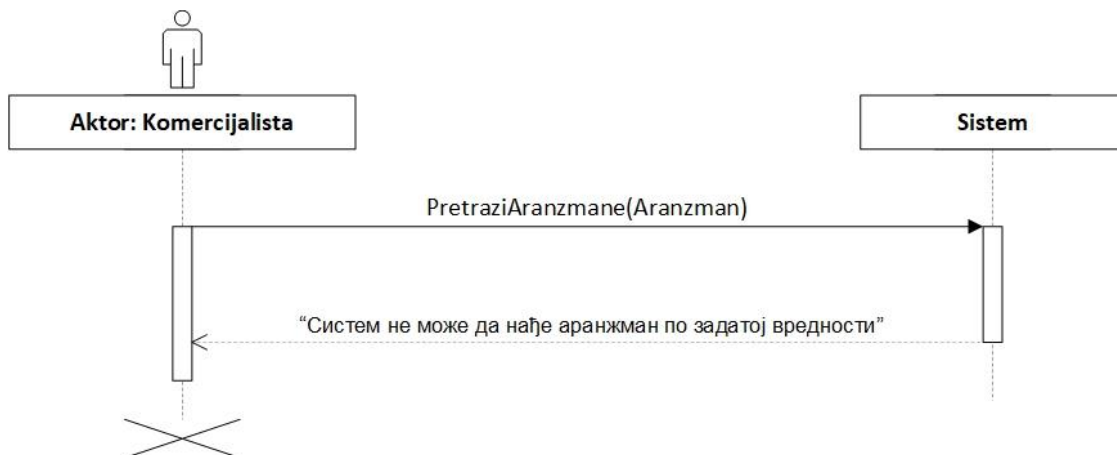
ДС7: Дијаграм секвенце случаја коришћења - Брисање аранжмана

1. **Комерцијалиста** **позива** **систем** да нађе **аранжман** по задатој вредности. (АПСО)
2. **Систем** приказује **комерцијалисти** **аранжман** и поруку: "**Систем** је нашао **аранжман** по задатој вредности". (ИА)
3. **Комерцијалиста** **позива** **систем** да обрише **аранжман**. (АПСО)
4. **Систем** приказује **комерцијалисти** поруку: "**Систем** је обрисао **аранжман**." (ИА)

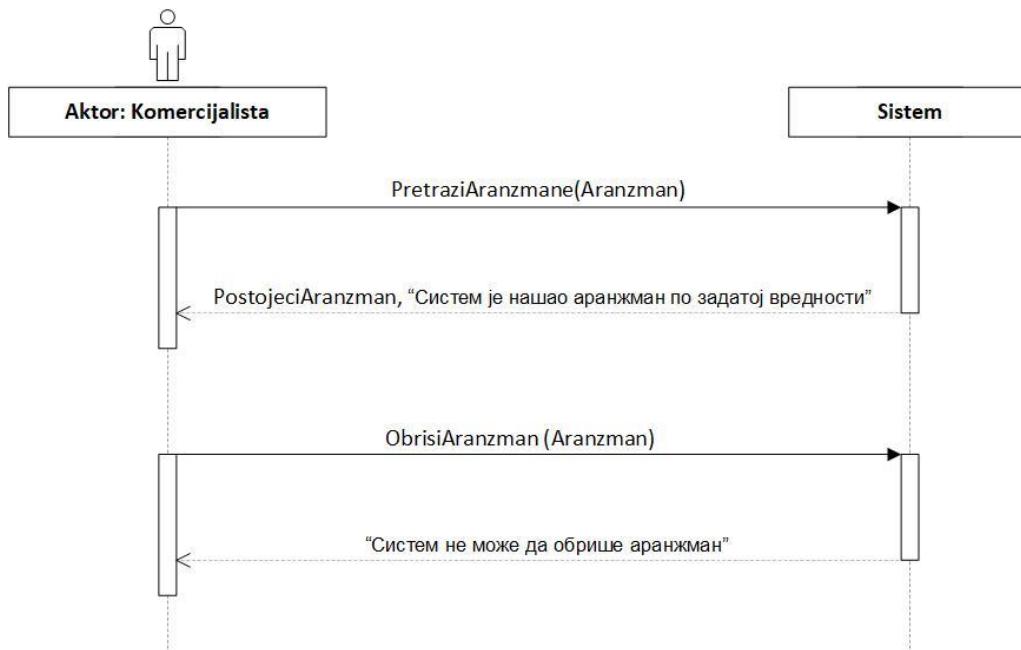


Алтернативна сценарија

2.1 Уколико **систем** не може да нађе **аранжман** он приказује **комерцијалисти** поруку: "**Систем** не може да нађе **аранжман** по задатој вредности". Прекида се извршење сценарија. (ИА)



4.1 Уколико **систем** не може да обрише **аранжман** он приказује **комерцијалисти** поруку “**Систем** не може да обрише **аранжман**”. (ИА)

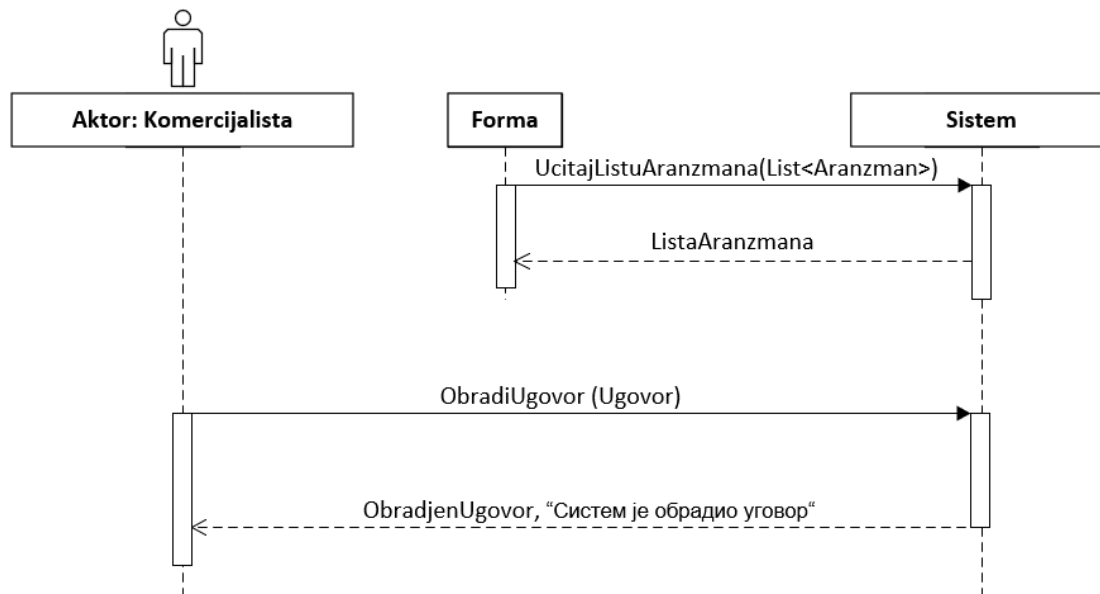


Са наведених секвенцих дијаграма уочавају се 2 системске операције које треба пројектовати:

1. signal **PretraziAranzmane** (Aranzman);
2. signal **ObrisiAranzman** (Aranzman);

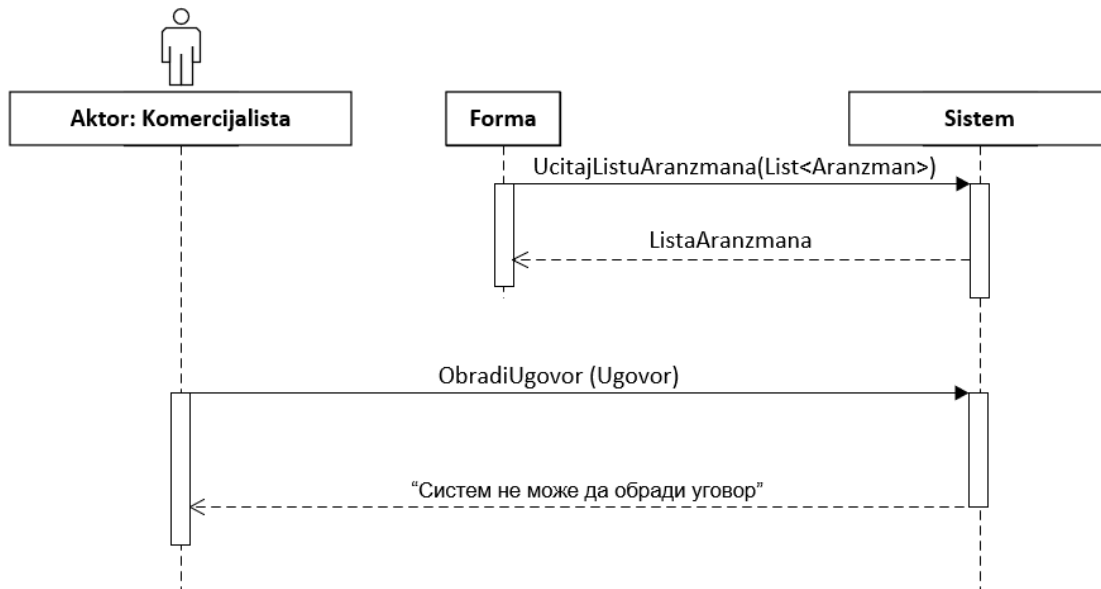
ДС8: Дијаграм секвенце случаја коришћења - Креирање уговора

1. **Форма** **позива** **систем** да учита **листу аранжмана**. (АПСО)
2. **Систем** **враћа** **форми** **листу аранжмана**. (ИА)
3. **Комерцијалиста** **позива** **систем** да обради **уговор**. (АПСО)
4. **Систем** **приказује** **комерцијалисти** **обрађен уговор** и поруку: "**Систем** је обрадио **уговор**". (ИА)



Алтернативна сценарија

- 4.1 Уколико **систем** не може да обради **уговор** он приказује **комерцијалисти** поруку: “**Систем** не може да обради **уговор**”. (ИА)

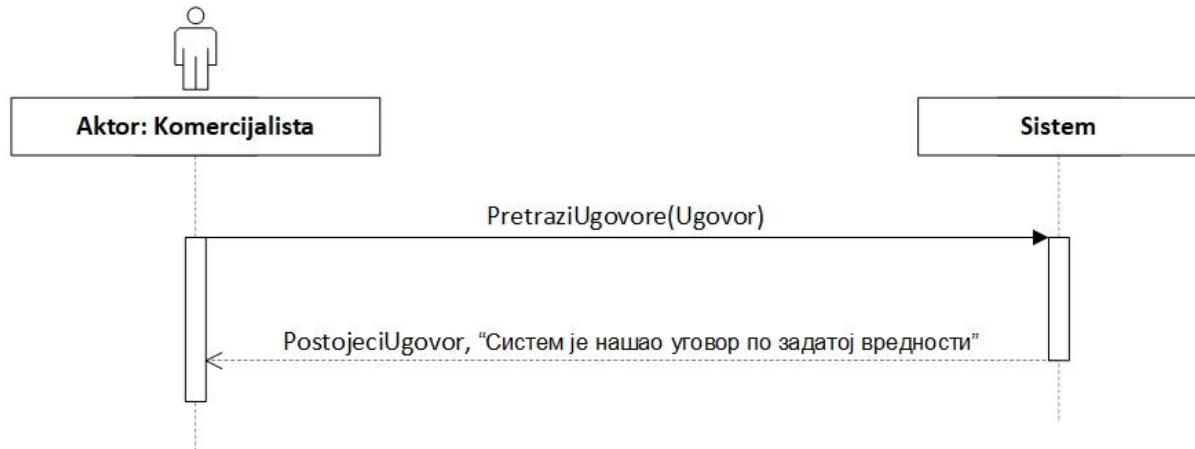


Са наведених секвенцих дијаграма уочавају се 2 системске операције које треба пројектовати:

1. signal **UcitajListuAranzmana** (List <Aranzman>)
2. signal **ObradiUgovor** (Ugovor);

ДС9: Дијаграм секвенце случаја коришћења - Претраживање уговора

1. **Комерцијалиста** **позива** **систем** да нађе **уговор** по задатој вредности. (АПСО)
2. **Систем** приказује **комерцијалисти** податке о **уговору** и поруку: "**Систем** је нашао **уговор** по задатој вредности". (ИА)



Алтернативна сценарија

- 2.1 Уколико **систем** не може да нађе **уговор** он приказује **комерцијалисти** поруку: "**Систем** не може да нађе **уговор** по задатој вредности". (ИА)

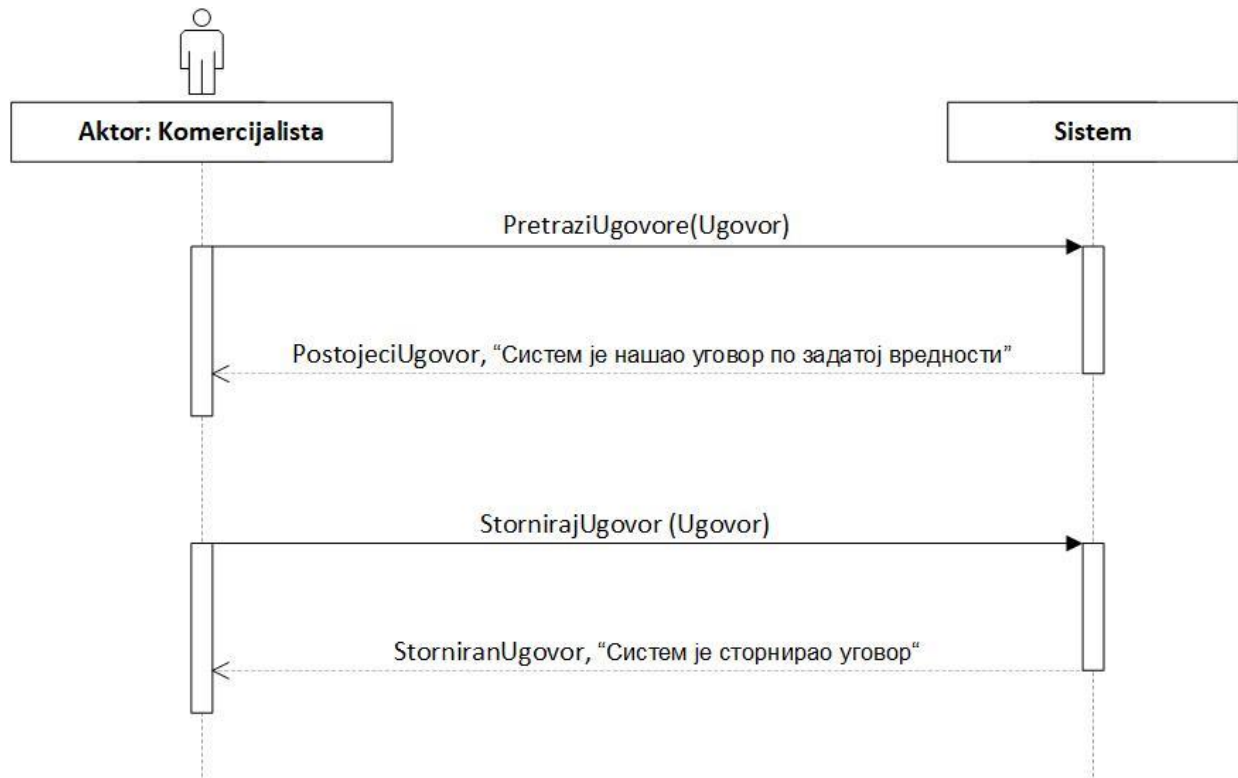


Са наведених секвенцих дијаграма уочава се 1 системска операција коју треба пројектовати:

1. signal **PretraziUgovore** (Ugovor);

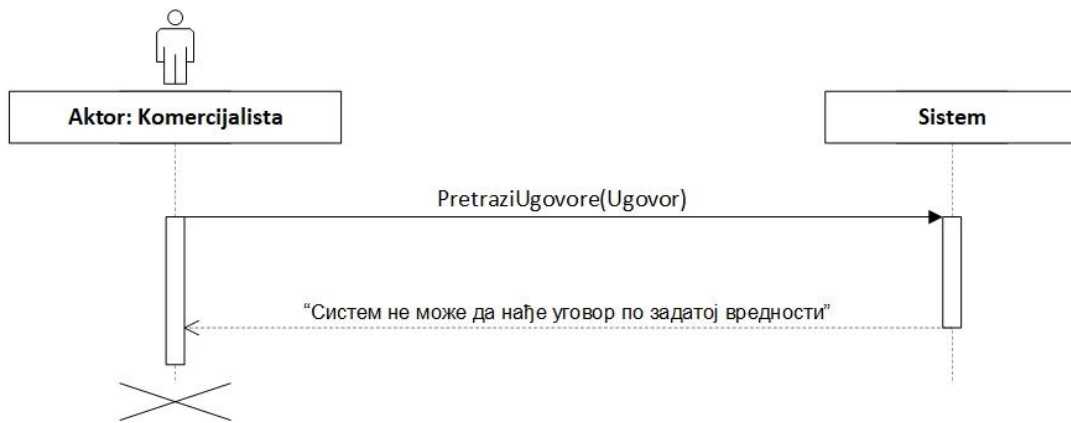
ДС10: Дијаграм секвенце случаја коришћења - Сторнирање уговора

1. **Комерцијалиста** **позива** **систем** да нађе **уговор** по задатој вредности. (АПСО)
2. **Систем** приказује **комерцијалисти** **уговор** и поруку: "**Систем** је нашао **уговор** по задатој вредности". (ИА)
3. **Комерцијалиста** **позива** **систем** да сторнира задати **уговор**. (АПСО)
4. **Систем** приказује **комерцијалисти** сторниран **уговор** и поруку: "**Систем** је сторнирао **уговор**". (ИА)

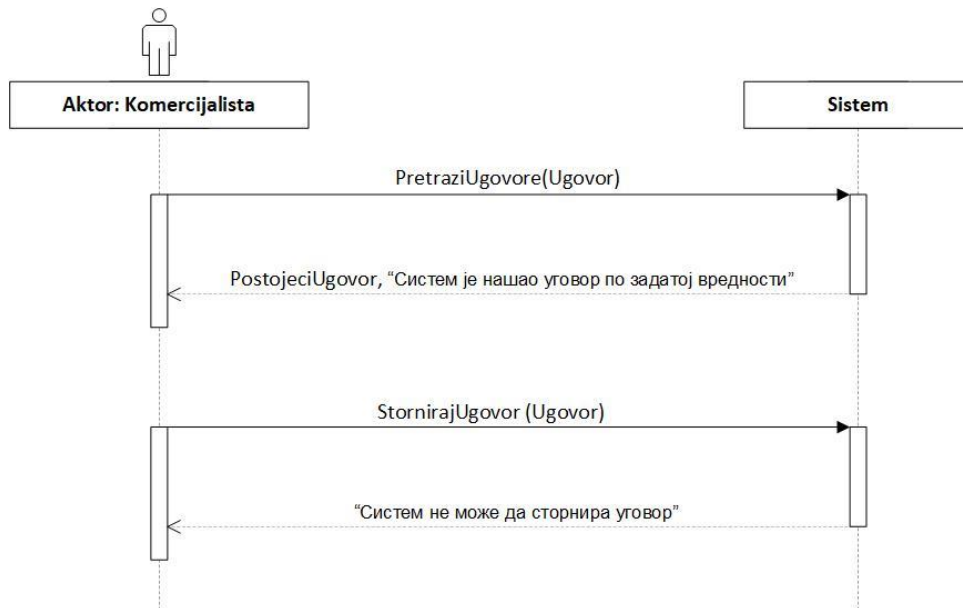


Алтернативна сценарија

2.1 Уколико **систем** не може да нађе **уговор** он приказује **комерцијалисти** поруку: “**Систем** не може да нађе **уговор** по задатој вредности”. Прекида се извршење сценарија. (ИА)



4.1 Уколико **систем** не може да сторнира **уговор** он приказује **комерцијалисти** поруку: “**Систем** не може да сторнира **уговор**”.

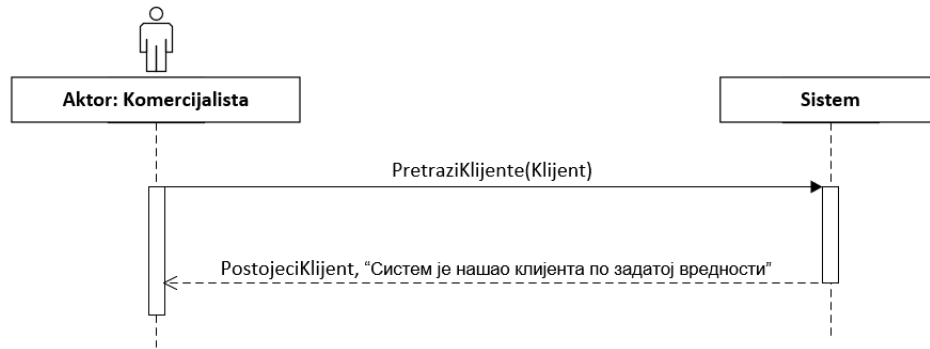


Са наведених секвенцих дијаграма уочавају се 2 системске операције које треба пројектовати:

1. signal **PretraziUgovore** (Ugovor);
2. signal **StornirajUgovor** (Ugovor);

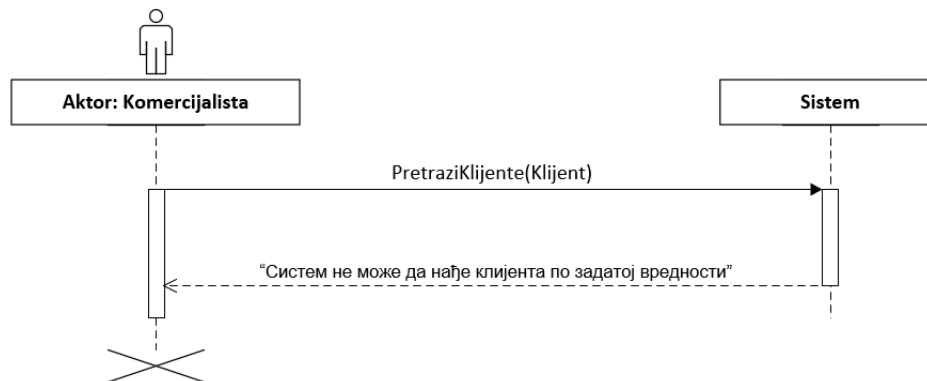
ДС11: Дијаграм секвенце случаја коришћења - Претраживање клијената

1. **Комерцијалиста** **позива** **систем** да нађе **клијента** по задатој вредности. (АПСО)
2. **Систем** приказује **комерцијалисти** податке о **клијенту** и поруку: "**Систем** је нашао **клијента** по задатој вредности". (ИА)



Алтернативна сценарија

- 1.1 Уколико **систем** не може да нађе **клијента** он приказује **комерцијалисти** поруку: "**Систем** не може да нађе **клијента** по задатој вредности". (ИА)



Са наведених секвенцих дијаграма уочава се 1 системска операција коју треба пројектовати:

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

Kao rezultat analize scenarija добијено је укупно 13 системских операција које треба пројектовати:

1. *signal* **KreirajDestinaciju** (Destinacija);
2. *signal* **ZapamtiDestinaciju** (Destinacija);
3. *signal* **PretraziDestinacije** (Destinacija);
4. *signal* **UcitajListuDestinacija** (List <Destinacija>)
5. *signal* **KreirajAranzman** (Aranzman);
6. *signal* **ZapamtiAranzman** (Aranzman);
7. *signal* **PretraziAranzmane** (Aranzman);
8. *signal* **ObrisiAranzman** (Aranzman);
9. *signal* **UcitajListuAranzmana** (List <Aranzman>)
10. *signal* **ObradiUgovor** (Ugovor);
11. *signal* **PretraziUgovore** (Ugovor);
12. *signal* **StornirajUgovor** (Ugovor);
13. *signal* **PretraziKlijente** (Klijent);

2.2. Дефинисање уговора о системским операцијама

1. Уговор UG1: *KreirajDestinaciju*

Операција: **KreirajDestinaciju** (*Destinacija*):signal;

Веза са СК: СК1

Предуслови: Вредносна и структурна ограничења над објектом Destinacija морају бити задовољена.

Постуслови: Направљена је нова дестинација.

2. Уговор UG2: *ZapamtiDestinaciju*

Операција: **ZapamtiDestinaciju** (*Destinacija*):signal;

Веза са СК: СК3

Предуслови: Вредносна и структурна ограничења над објектом Destinacija морају бити задовољена.

Постуслови: Дестинација је запамћена.

3. Уговор UG3: *PretraziDestinacije*

Операција: **PretraziDestinacije** (*Destinacija*):signal;

Веза са СК: СК2, СК3

Предуслови:

Постуслови: Пронађена је тражена дестинација.

4. Уговор UG4: *UcitajListuDestinacija*

Операција: **UcitajListuDestinacija** (*List<Destinacija>*):signal;

Веза са СК: СК4

Предуслови:

Постуслови: Учитана је листа дестинација.

5. Уговор UG5: *KreirajAranzman*

Операција: **KreirajAranzman** (*Aranzman*):signal;

Веза са СК: СК4

Предуслови: Вредносна и структурна ограничења над објектом Aranzman морају бити задовољена.

Постуслови: Направљен је нови аранжман.

6. Уговор UG6: *ZapamtiAranzman*

Операција: *ZapamtiAranzman* (*Aranzman*):signal;

Веза са СК: СК6

Предуслови: Вредносна и структурна ограничења над објектом Aranzman морају бити задовољена.

Постуслови: Аранжман је запамћен.

7. Уговор UG7: *PretraziAranzmane*

Операција: *PretraziAranzmane* (*Aranzman*):signal;

Веза са СК: СК5, СК6, СК7

Предуслови:

Постуслови: Пронађен је тражени аранжман.

8. Уговор UG8: *ObrisiAranzman*

Операција: *ObrisiAranzman* (*Aranzman*):signal;

Веза са СК: СК7

Предуслови: Вредносна и структурна ограничења над објектом Aranzman морају бити задовољена.

Постуслови: Аранжман је обрисан.

9. Уговор UG9: *UcitajListuAranzmana*

Операција: *UcitajListuAranzmana* (*List<Aranzman>*):signal;

Веза са СК: СК8

Предуслови:

Постуслови: Учитана је листа аранжмана.

10. Уговор UG10: *ObradiUgovor*

Операција: *ObradiUgovor* (*Ugovor*):signal;

Веза са СК: СК8

Предуслови: Вредносна и структурна ограничења над објектом Ugovor морају бити задовољена. Ако је уговор обрађен или сторниран не може се извршити системска операција.

Постуслови: Уговор је обрађен.

11. Уговор UG11: *PretraziUgovore*

Операција: *PretraziUgovore* (*Ugovor*):signal;

Веза са СК: СК9, СК10

Предуслови:

Постуслови: Пронађен је тражени уговор.

12. Уговор UG12: *StornirajUgovor*

Операција: *StornirajUgovor (Ugovor):signal;*

Веза са СК: СК10

Предуслови: Вредносна и структурна ограничења над објектом Ugovor морају бити задовољена. Ако је уговор сторниран не може се извршити системска операција.

Постуслови: Уговор је сторниран.

13. Уговор UG13: *PretraziKlijente*

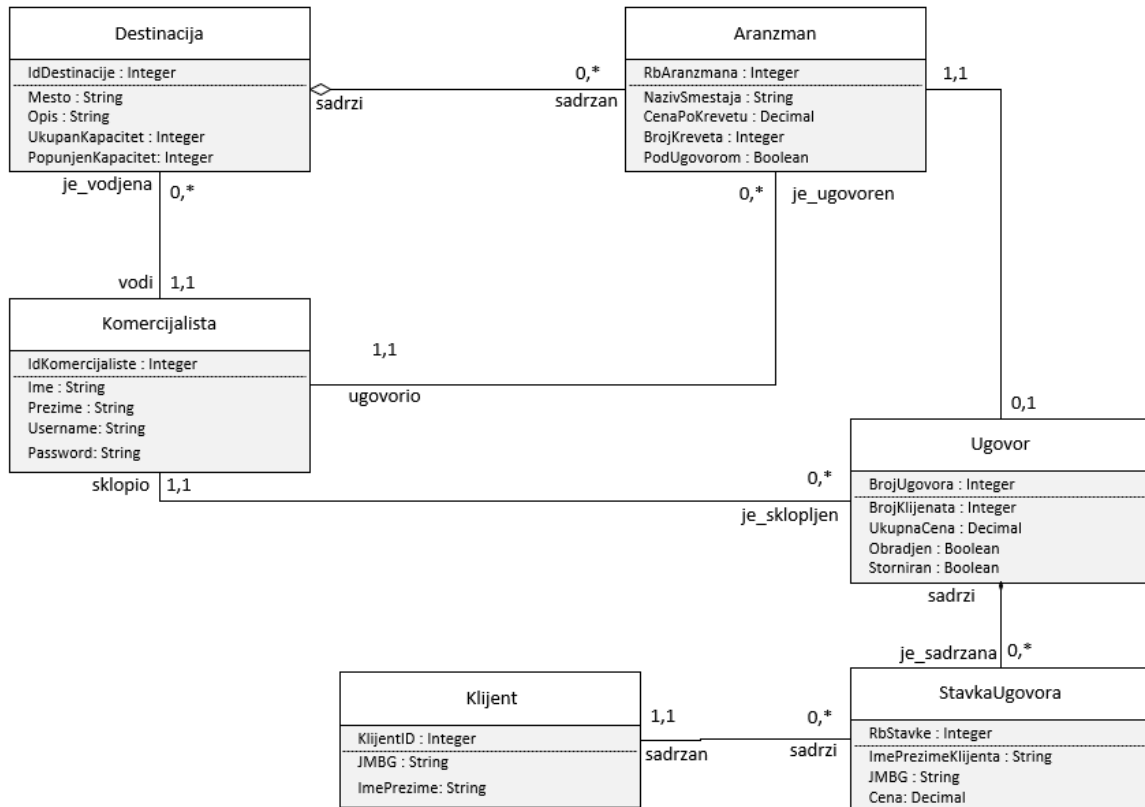
Операција: *PretraziKlijente (Klijent):signal;*

Веза са СК: СК11

Предуслови:

Постуслови: Пронађен је тражени клијент.

2.3. Структура софтверског система - Концептуални (доменски) модел



2.4. Структура софтверског система - Релациони модел

Komercijalista (IdKomercijaliste, Ime, Prezime, Username, Password)

Destinacija (IdDestinacije, Mesto, Opis,, IdKomercijaliste)

Aranzman (IdDestinacije,RbAranzmana, NazivSmestaja, CenaPoKrevetu, BrojKreveta, PodUgovorom, IdKomercijaliste)

Ugovor (BrojUgovora, BrojKlijenata, UkupnaCena, Obradjen, Storniran, IdDestinacije,RbAranzmana , IdKomercijaliste)

StavkaUgovora(BrojUgovora,RbStavke, ImePrezimeKlijenta, JMBG, Cena)

Klijent(KlijentID,JMBG, ImePrezime)

Tabela Komerijalista		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	INSERT / UPDATE CASCADES Destinacija, Aranzman, Ugovor DELETE RESTRICTED Destinacija, Aranzman, Ugovor
	IdKomerijaliste	Integer	Not null and >0			
	Ime	String	Not null			
	Prezime	String	Not null			
	Username	String	Not null			
	Password	String	Not null			

Tabela Destinacija		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	INSERT RESTRICTED Komerijalista UPDATE RESTRICTED Komerijalista CASCADES Aranzman DELETE RESTRICTED Aranzman
	IdDestinacije	Integer	Not null and >0			
	Mesto	String	Not null			
	Opis	String	Not null			
	IdKomerijaliste	Integer	Not null and >0			

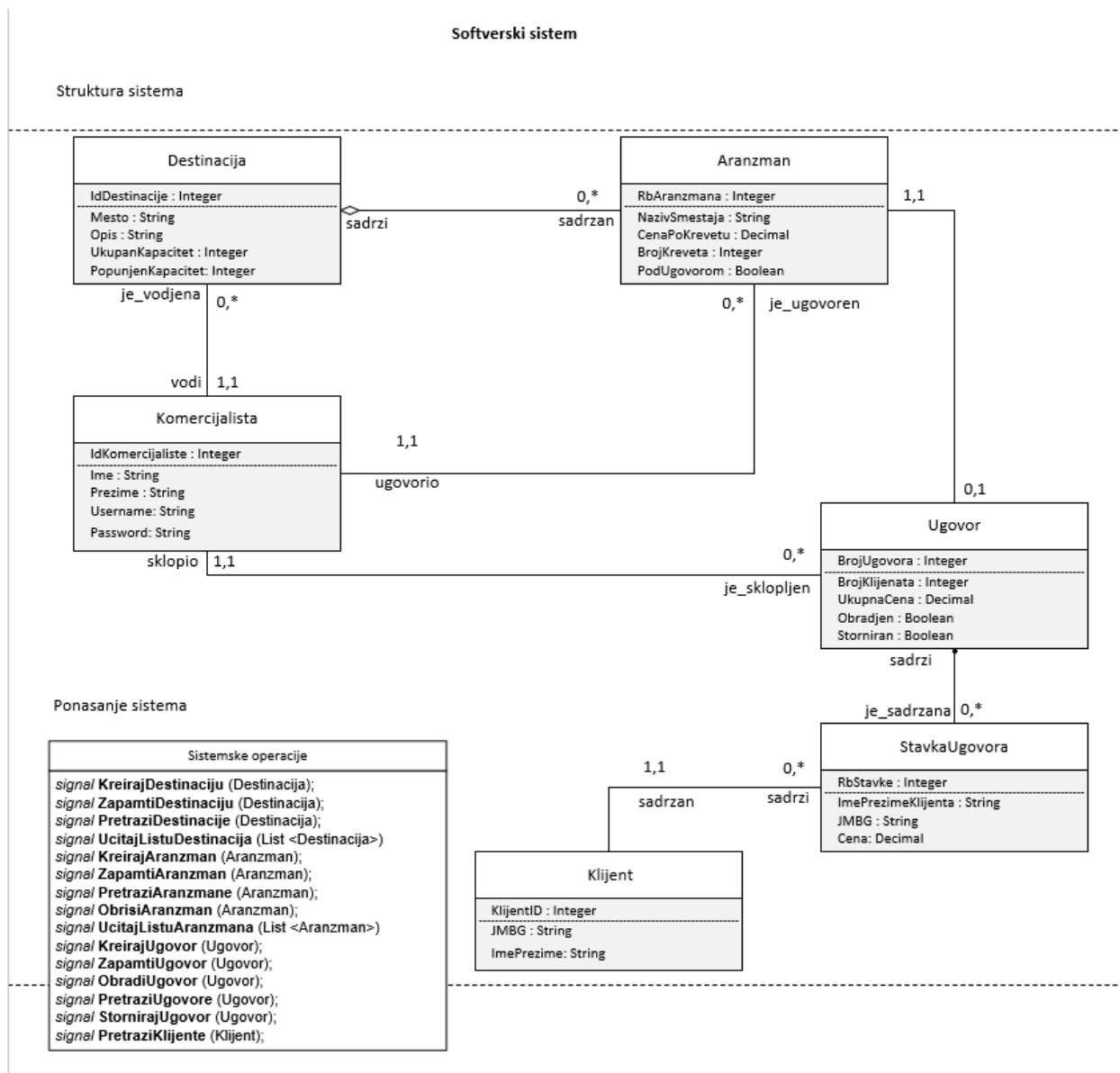
Tabela Aranzman		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	INSERT RESTRICTED Komerijalista, Destinacija UPDATE RESTRICTED Komerijalista, Destinacija, Ugovor DELETE RESTRICTED Ugovor
	IdDestinacije	Integer	Not null and >0			
	RbAranzmana	Integer	Not null and >0			
	NazivSmestaja	String	Not null			
	CenaPoKrevetu	Decimal	Not null and >0			
	BrojKreveta	Integer	Not null and >= 0			
	PodUgovorom	Boolean	Not null (default:false)			
	IdKomerijaliste	Integer	Not null and >0			

Tabela Ugovor		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	INSERT RESTRICTED Komerijalista, Aranzman UPDATE RESTRICTED Komerijalista, Aranzman CASCADES StavkaUgovora DELETE CASCADES StavkaUgovora
	BrojUgovora	String	Not null			
	BrojKlijenata	Integer	Not null and >0		BrojKlijenata=SUM (StavkaUgovora)	
	UkupnaCena	Decimal	Not null and >0		UkupnaCena=SUM(StavkaUgovora.Cena)	
	Obradjen	Boolean	Not null (default:false)			
	Storniran	Boolean	Not null (default:false)			
	IdDestinacije	Integer	Not null and >0			
	RbAranzmana	Integer	Not null and >0			
	IdKomerijaliste	Integer	Not null and >0			

Tabela StavkaUgovora		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	INSERT RESTRICTED Ugovor UPDATE RESTRICTED Ugovor DELETE /
	BrojUgovora	Integer	Not null and > 0			
	RbStavke	Integer	Not null and >0			
	ImePrezimeKlijenta	String	Not null			
	JMBG	String	Not null			
	Cena	Decimal	Not null and >0			

Tabela Klijent		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	INSERT / UPDATE / DELETE /
	KlijentId	Integer	Not null			
	JMBG	String	Not null			
	ImePrezime	String	Not null			

Kao rezultat analize scenarija SK i pravљења konceptualnog modela добија се логичка структура и понашање софтверског система:



3. Пројектовање

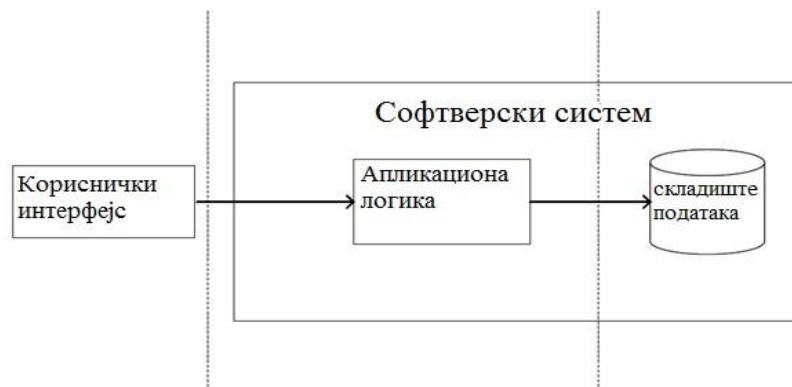
Фаза пројектовања описује физичку структуру и понашање софтверског система (архитектуру софтверског система).

3.1. Архитектура софтверског система

Тронивојска архитектура софтверског система се састоји од следећих нивоа:

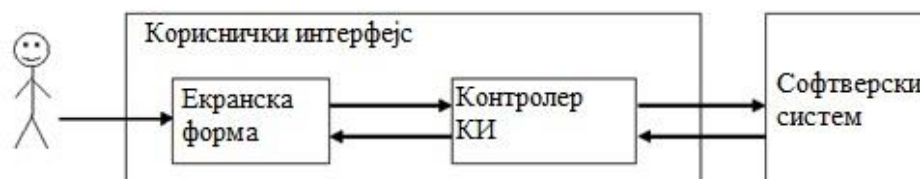
- Кориснички интерфејс
- Апликациона логика
- Складиште података

Ниво корисничког интерфејса је на страни клијента, а апликациона логика и складиште података су на страни сервера.



3.2. Пројектовање корисничког интерфејса

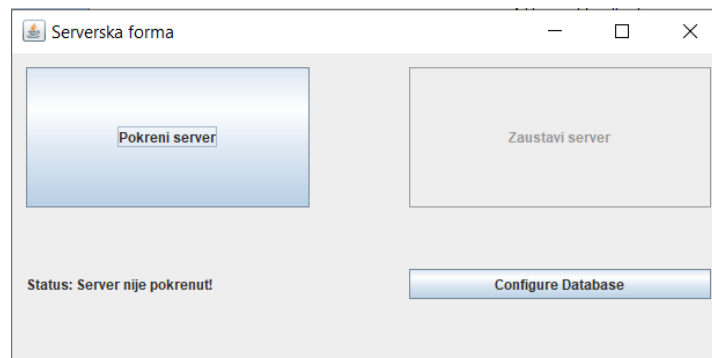
Кориснички интерфејс представља реализацију улаза и/или излаза софтверског система и састоји се од екранских форми и контролера корисничког интерфејса.



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 он прави, позива контролера корисничког интерфејса како би му проследио те податке и приказује податке добијене од контролера корисничког интерфејса.

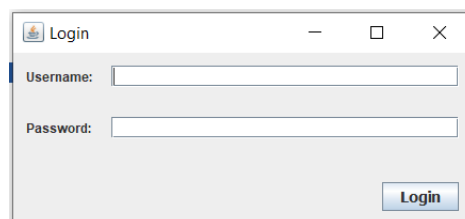
Изглед корисничког интерфејса серверског дела апликације (пре покретања сервера):



Изглед корисничког интерфејса серверског дела апликације (након покретања сервера):



Изглед корисничког интерфејса клијентског дела апликације (форма за пријављивање комерцијалисте на систем):



Изглед корисничког интерфејса клијентског дела апликације (главна форма):

Komercijalista: Filip Gogic

Destinacije

Naziv	Opis	Ukupan kapacitet	Popunjen kapacitet
Lefkada	Najlepse grcko ostrvo!	22	12
Kavos	Svakodnevne zurke!	13	5
Kefalonija	Prelepe plaze sa kornjacama!	6	4
Kusadasi	Najveca party destinacija u Turskoj	3	3
Suncev Breg	Povoljno letovanje u Bugarskoj	5	5
Hurgada	Letovaliste u Egiptu	4	0
Evia	Banja u Grčkoj	7	0
Majorka	Luksuzno u Spaniji	5	0
Krakov	Grad u Poljskoj	0	0
Pariz	Francuska	0	0
Barselona	Spanija	0	0
Madrid	Spanija 2	0	0
Valensija	Spanija 3	4	0
Mona	posljed	2	0

Pretraži Resetuj tabelu

Otvori aranžmane
Dodaj destinaciju
Izmeni destinaciju

Кроз случајеве коришћења пројектоване су и остале екранске форме које ће апликација поседовати.

СК1: Случај коришћења - Креирање дестинације

Назив СК

Креирање дестинације

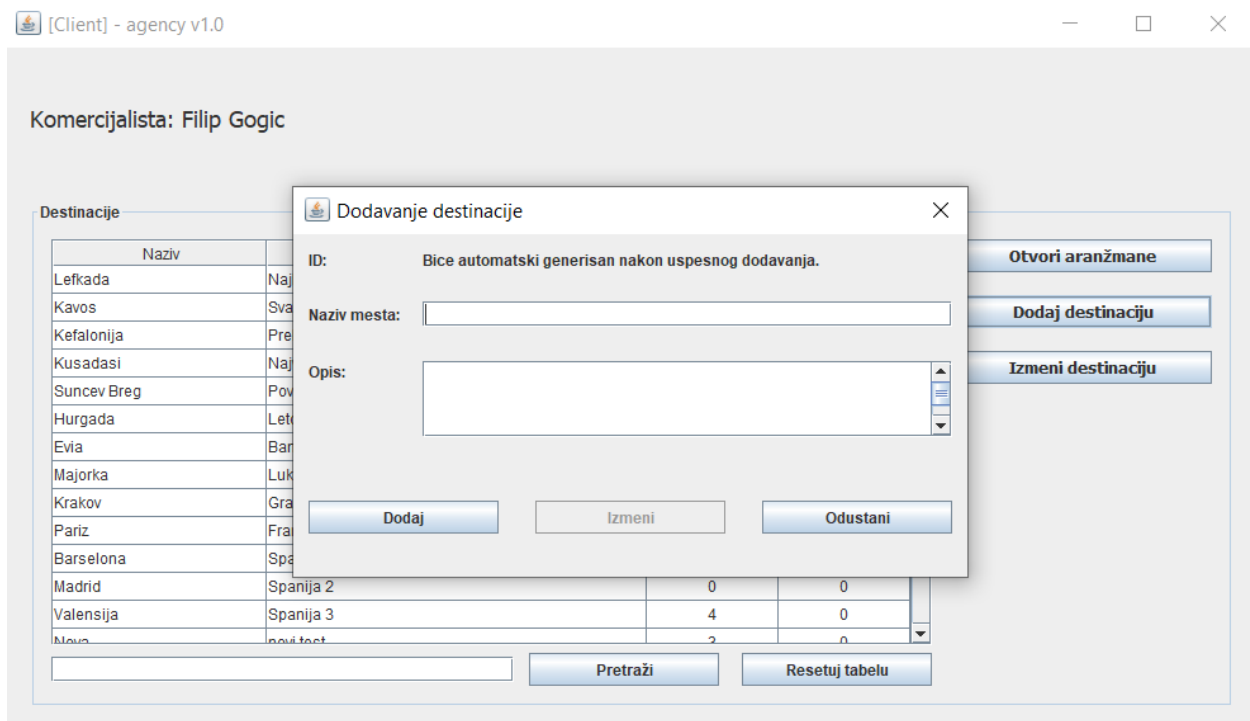
Актори СК

Комерцијалиста

Учесници СК

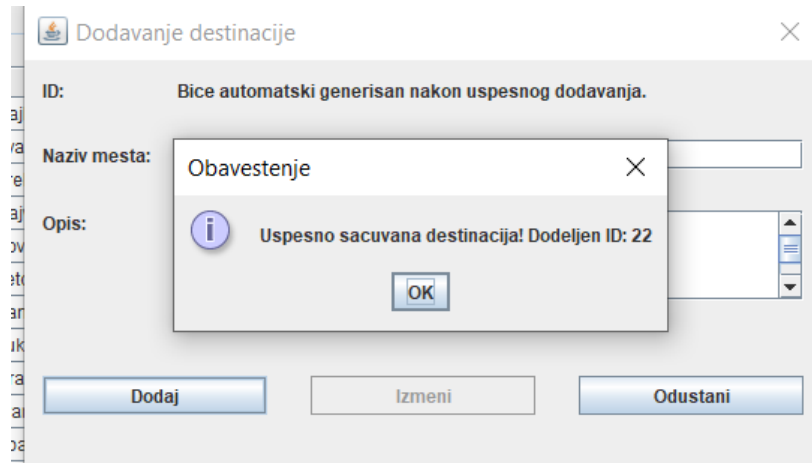
Комерцијалиста и систем (програм)

Предуслов: Систем је укључен и комерцијалиста је улогован под својом шифром. Систем приказује форму за рад са дестинацијом.



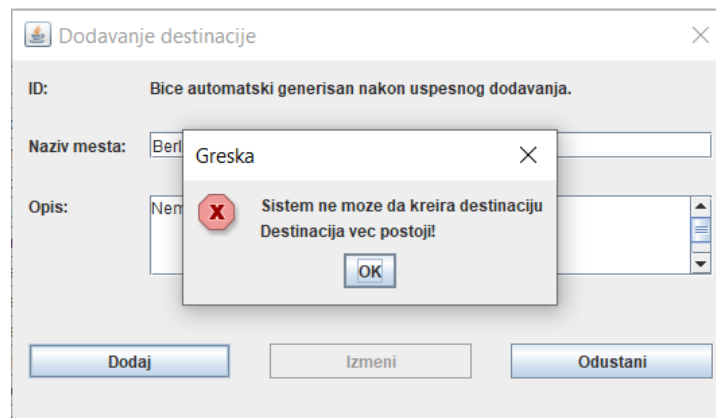
Основни сценарио СК

1. Комерцијалиста уноси податке у дестинацију. (АПУСО)
2. Комерцијалиста контролише да ли је коректно унео податке у дестинацију. (АНСО)
3. Комерцијалиста позива систем да креира дестинацију. (АПСО)
4. Систем креира дестинацију. (СО)
5. Систем приказује комерцијалисти креирану дестинацију и поруку: "Систем је креирао дестинацију". (ИА)



Алтернативна сценарија

5.1 Уколико **систем** не може да креира **дестинацију** он приказује **комерцијалисти** поруку “**Систем** не може да креира **дестинацију**”. (ИА)



СК2: Случај коришћења - Претраживање дестинација

Назив СК

Претраживање дестинације

Актори СК

Комерцијалиста

Учесници СК

Комерцијалиста и систем (програм)

Предуслов: Систем је укључен и комерцијалиста је улогован под својом шифром. Систем приказује форму за рад са дестинацијом. Учитана је листа дестинација.

[Client] - agency v1.0

Komercijalista: Filip Gogic

Destinacije

Naziv	Opis	Ukupan kapacitet	Popunjen kapacitet
Lefkada	Najlepse grcko ostrvo!	12	3
Kavos	Svakodnevne zurke!	10	0
Kefalonija	Prelepe plaze sa kornjacama!	4	0
Kusadasi	Najveca party destinacija u Turskoj	3	0
Suncev Breg	Povoljno letovanje u Bugarskoj	5	0
Hurgada	Letovaliste u Egiptu	4	0
Evia	Banja u Grčkoj	7	0
Majorka	Luksuzno u Spaniji	5	0
Krakov	Grad u Poljskoj	0	0
Pariz	Francuska	0	0
Barselona	Spanija	0	0
Madrid	Spanija 2	0	0
Valensija	Spanija 3	4	0
Moskva	poslednja	2	0

Pretraži Resetuj tabelu

Otvori aranžmane
Dodaj destinaciju
Izmeni destinaciju

Основни сценарио СК

1. Комерцијалиста уноси вредност по којој претражује дестинације. (АПУСО)
2. Комерцијалиста позива систем да нађе дестинације по задатој вредности. (АПСО)
3. Систем тражи дестинације по задатој вредности. (СО)
4. Систем приказује комерцијалисти нађене дестинације. (ИА)

Destinacije

Naziv	Opis	Ukupan kapacitet	Popunjen kapacitet
Lefkada	Najlepse grcko ostrvo!	12	3

Алтернативна сценарија

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

Destinacije

Naziv	Opis	Ukupan kapacitet	Popunjen kapacitet
Lefkada	Najlepse grcko ostrvo!	12	3
Kavos	Svakodnevne zurke!	12	2
Kefalonija	Prelepe plaze		
Kusadasi	Najveca party		
Suncev Breg	Povoljno letovanje		
Hurgada	Letovaliste u Egiptu		
Evia	Banja u Grckoj		
Majorka	Luksuzno u Spaniji	5	0
Krakov	Grad u Poljskoj	0	0
Pariz	Francuska	0	0
Barselona	Spanija	0	0
Madrid	Spanija 2	0	0
Valensija	Spanija 3	4	0
Moskva	novi test	2	0

Greska prilikom pretrage

X Sistem ne moze da pronadje destinaciju po zadatoj vrednosti!

OK

СК3: Случај коришћења - Измена дестинације

Назив СК

Измена дестинације

Актори СК

Комерцијалиста

Учесници СК

Комерцијалиста и систем (програм)

Предуслов: Систем је укључен и комерцијалиста је улогован под својом шифром. Систем приказује форму за рад са дестинацијом. Учитана је листа дестинација.

[Client] - agency v1.0

Komercijalista: Filip Gogic

Destinacije

Naziv	Opis	Ukupan kapacitet	Popunjen kapaci...
Lefkada	Najlepse grcko ostrvo!	12	3
Kavos	Svakodnevne zurke!	10	0
Kefalonija	Prelepe plaze sa kornjacama!	4	0
Kusadasi	Najveca party destinacija u Turskoj	3	0
Suncev Breg	Povoljno letovanje u Bugarskoj	5	0
Hurgada	Letovaliste u Egiptu	4	0
Evia	Banja u Grčkoj	7	0
Majorka	Luksuzno u Spaniji	5	0
Krakov	Grad u Poljskoj	0	0
Pariz	Francuska	0	0
Barselona	Spanija	0	0
Madrid	Spanija 2	0	0
Valensija	Spanija 3	4	0
Moskva	posred...	2	0

Pretraži Resetuj tabelu

Otvori aranžmane
Dodaj destinaciju
Izmeni destinaciju

Основни сценарио СК

1. Комерцијалиста уноси вредност по којој претражује дестинацију. (АПУСО)
2. Комерцијалиста позива систем да нађе дестинацију по задатој вредности. (АПСО)
3. Систем тражи дестинације по задатој вредности. (СО)
4. Систем приказује комерцијалисти нађене дестинације. (ИА)

Destinacije

Naziv	Opis	Ukupan kapacitet	Popunjen kapacitet
Kavos	Svakodnevne zurke!	13	2
Kazablanka	test	0	0

5. **Комерцијалиста** бира **дестинацију** чије податке жели да измени.(АПУСО)
6. **Комерцијалиста** **позива систем** да учита податке о **дестинацији**. (АПСО)
7. **Систем** **учитава** изабрану **дестинацију**. (СО)
8. **Систем** **приказује** **комерцијалисти** изабрану **дестинацију**. (ИА)

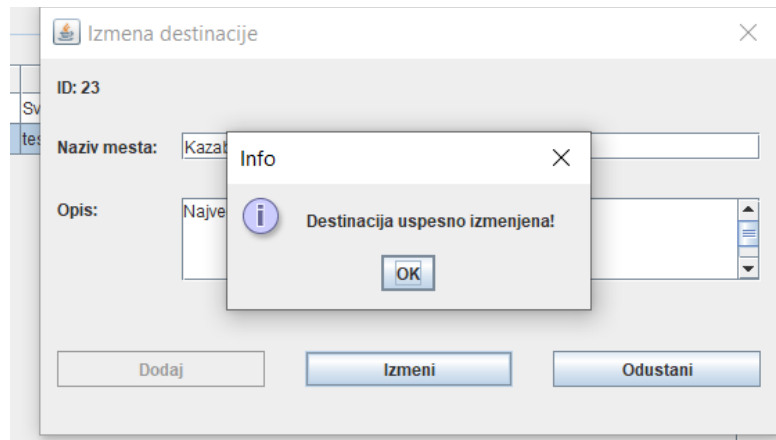
Izmena destinacije

ID: 23

Naziv mesta:

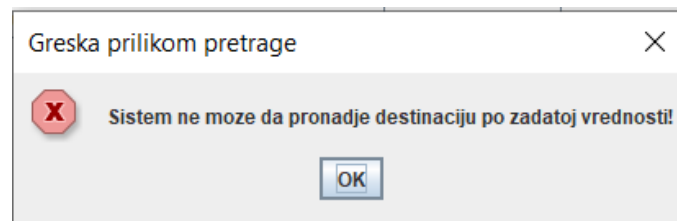
Opis:

9. **Комерцијалиста** **уноси (мења)** податке о **дестинацији**. (АПУСО)
10. **Комерцијалиста** **контролише** да ли је коректно унео податке о **дестинацији**. (АНСО)
11. **Комерцијалиста** **позива систем** да запамти податке о **дестинацији**. (АПСО)
12. **Систем** **памти** податке о **дестинацији**. (СО)
13. **Систем** **приказује** **комерцијалисти** запамћену **дестинацију** и поруку: “**Систем** је запамтио **дестинацију**.” (ИА)



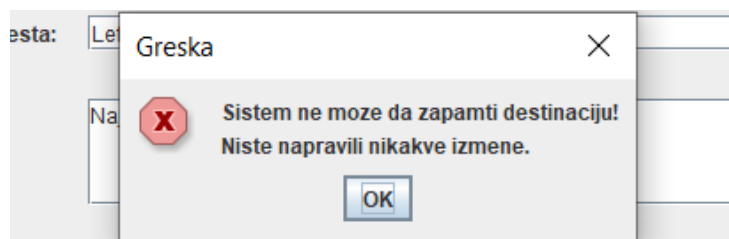
Алтернативна сценарија

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



8.1 Уколико **систем** не може да нађе **дестинацију** он приказује **комерцијалисти** поруку “**Систем** не може да нађе **дестинацију**”. (ИА)

13.1 Уколико **систем** не може да запамти податке о **дестинацији** он приказује **комерцијалисти** поруку “**Систем** не може да запамти **дестинацију**”. (ИА)



СК4: Случај коришћења - Креирање аранжмана

Назив СК

Креирање аранжмана


Актори СК

Комерцијалиста

Учесници СК

Комерцијалиста и систем (програм)

Предуслов: Систем је укључен и комерцијалиста је улогован под својом шифром. Систем приказује форму за рад са аранжманом.

 Dodavanje aranzmana ×

Destinacija: Lefkada

RB Aranzmana: Bice automatski generisan nakon uspesnog dodavanja aranzmana.

Naziv smestaja:

Cena po krevetu:

Broj kreveta:

Komercijalista: Filip Gogic


Sacuvaj aranzman

Izmeni aranzman

Odustani

Основни сценарио СК

1. Комерцијалиста уноси податке у аранжман. (АПУСО)
2. Комерцијалиста контролише да ли је коректно унео податке у аранжман. (АНСО)
3. Комерцијалиста позива систем да креира аранжман. (АПСО)
4. Систем креира аранжман. (СО)
5. Систем приказује комерцијалисти креиран аранжман и поруку: “Систем је креирао аранжман”. (ИА)

 Dodavanje aranzmana ×

Destinacija: Kefalonija

RB Aranzmana: Bice automatski generisan nakon uspesnog dodavanja aranzmana.


Naziv smestaja:

Cena po krevetu:

Broj kreveta:

Komercijalista: Filip Gogic


Obavestenje ×

 Sistem je zapamtio aranzman! Dodeljen RB: 35

Алтернативна сценарија

- 5.1 Уколико **систем** не може да креира **аранжман** он приказује **комерцијалисти** поруку: “**Систем** не може да креира **аранжман**”. Прекида се извршење сценарија. (ИА)

Greska ×

 Sistem ne moze da zapamti aranzman!

СК5: Случај коришћења - Претраживање аранжмана

Назив СК

Претраживање аранжмана

Актори СК

Комерцијалиста

Учесници СК

Комерцијалиста и систем (програм)

Предуслов: Систем је укључен и комерцијалиста је улогован под својом шифром. Систем приказује форму за рад са аранжманом. Учитана је листа аранжмана.

[Client] - agency v1.0 - Aranzman

Destinacija

Naziv: Lefkada

Opis: Najlepse grcko ostrvo!

ID Destinacije: 1

Komercijalista: Filip Gogic

Ukupan kapacitet: 15

Popunjen kapacitet: 3

RB aranzmana	Naziv smestaja	Cena po krevetu	Broj kreveta	Pod ugovorom
1	Leferis	150.0	3	<input checked="" type="checkbox"/>
2	Nikki	189.0	4	<input type="checkbox"/>
21	Apostolos	154.55	5	<input type="checkbox"/>
34	New Star hotel	333.69	3	<input type="checkbox"/>

Pretrazi aranzmane

Resetuj tabelu

Dodaj aranzman

Izmeni aranzman

Obrisi aranzman

Kreiraj ugovor

Pogledaj ugovor

Pretrazi po broju ugovora

Основни сценарио СК

1. Комерцијалиста уноси вредност по којој претражује аранжмане. (АПУСО)
2. Комерцијалиста позива систем да нађе аранжмане по задатој вредности. (АПСО)
3. Систем тражи аранжмане по задатој вредности. (СО)
4. Систем приказује комерцијалисти нађене аранжмане. (ИА)

СК6: Случај коришћења - Измена аранжмана

Назив СК

Измена аранжмана

Актори СК

Комерцијалиста

Учесници СК

Комерцијалиста и систем (програм)

Предуслов: Систем је укључен и комерцијалиста је улогован под својом шифром. Систем приказује форму за рад са аранжманом. Учитана је листа аранжмана.

[Client] - agency v1.0 - Aranzman

Destinacija

Naziv: Lefkada Ukupan kapacitet: 15

Opis: Najlepse grcko ostrvo! Popunjen kapacitet: 3

ID Destinacije: 1

Komercijalista: Filip Gogic

RB aranzmana	Naziv smestaja	Cena po krevetu	Broj kreveta	Pod ugovorom
1	Letteris	150.0	3	<input checked="" type="checkbox"/>
2	Nikki	189.0	4	<input type="checkbox"/>
21	Apostolos	154.55	5	<input type="checkbox"/>
34	New Star hotel	333.69	3	<input type="checkbox"/>

Dodaj aranzman

Izmeni aranzman

Obrisi aranzman

Kreiraj ugovor

Pogledaj ugovor

Pretrazi po broju ugovora

Pretrazi aranzmane

Resetuj tabelu

Основни сценарио СК

1. Комерцијалиста уноси вредност по којој претражује аранжмане. (АПУСО)
2. Комерцијалиста позива систем да нађе аранжмане по задатој вредности. (АПСО)
3. Систем тражи аранжмане по задатој вредности. (СО)
4. Систем приказује комерцијалисти нађене аранжмане. (ИА)

[Client] - agency v1.0 - Aranzman

Destinacija

Naziv: Lefkada Ukupan kapacitet: 15
 Opis: Najlepse grcko ostrvo! Popunjen kapacitet: 3
 ID Destinacije: 1
 Komercijalista: Filip Gogic

RB aranzmana	Naziv smestaja	Cena po krevetu	Broj kreveta	Pod ugovorom
2	Nikki	189.0	4	<input type="checkbox"/>

Pretrazi aranzmane Resetuj tabelu

Dodaj aranzman
 Izmeni aranzman
 Obrisi aranzman
 Kreiraj ugovor
 Pogledaj ugovor
 Pretrazi po broju ugovora

5. **Комерцијалиста бира аранжман** чије податке жели да измени.(АПУСО)
6. **Комерцијалиста позива систем** да учита податке о **аранжману**. (АПСО)
7. **Систем учитава** изабран **аранжман**. (СО)
8. **Систем приказује комерцијалисти** изабран **аранжман**. (ИА)

Izmena aranzmana

Destinacija: Lefkada
 RB Aranzmana: 2

Naziv smestaja:

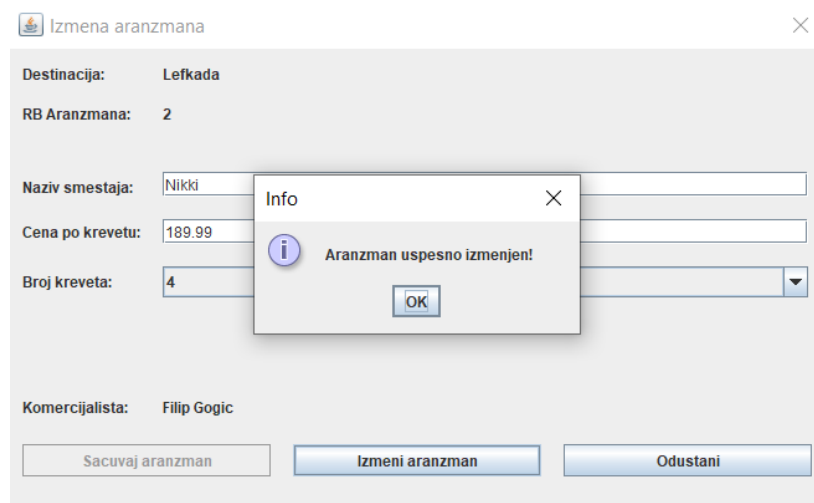
Cena po krevetu:

Broj kreveta:

Komercijalista: Filip Gogic

Sacuvaj aranzman Izmeni aranzman Odustani

9. **Комерцијалиста уноси (мења)** податке о **аранжману**. (АПУСО)
10. **Комерцијалиста контролише** да ли је коректно унео податке о **аранжману**. (АНСО)
11. **Комерцијалиста позива систем** да запамти податке о **аранжману**. (АПСО)
12. **Систем памти** податке о **аранжману**. (СО)
13. **Систем приказује комерцијалисти** запамћени **аранжман** и поруку: "Систем је запамтио **аранжман**." (ИА)



Izmena aranzmana

Destinacija: Lefkada

RB Aranzmana: 2

Naziv smestaja: Nikki

Cena po krevetu: 189.99

Broj kreveta: 4

Komercijalista: Filip Gogic

Sacuvaj aranzman Izmeni aranzman Odustani

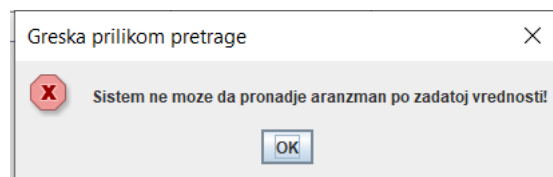
Info

Aranzman uspesno izmenjen!

OK

Алтернативна сценарија

4.1 Уколико **систем** не може да нађе **аранжман** он приказује **комерцијалисти** поруку: “**Систем** не може да нађе **аранжман** по задатој вредности”. Прекида се извршење сценарија. (ИА)

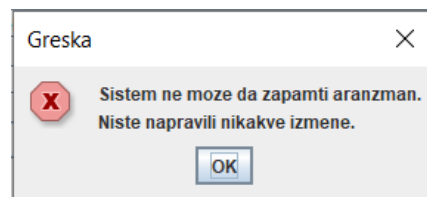


Greska prilikom pretrage

Sistem ne moze da pronadje aranzman po zadatoj vrednosti!

OK

13.1 Уколико **систем** не може да запамти податке о **аранжману** он приказује **комерцијалисти** поруку “**Систем** не може да запамти **аранжман**”. (ИА)



Greska

Sistem ne moze da zapamti aranzman.
Niste napravili nikakve izmene.

OK

СК7: Случај коришћења - Брисање аранжмана

Назив СК

Брисање аранжмана

Актори СК

Комерцијалиста

Учесници СК

Комерцијалиста и систем (програм)

Предуслов: Систем је укључен и комерцијалиста је улогован под својом шифром. Систем приказује форму за рад са аранжманом. Учитана је листа аранжмана.

[Client] - agency v1.0 - Aranzman

Destinacija

Naziv: Lefkada Ukupan kapacitet: 15

Opis: Najlepse grcko ostrvo! Popunjen kapacitet: 3

ID Destinacije: 1

Komercijalista: Filip Gogic

RB aranzmana	Naziv smestaja	Cena po krevetu	Broj kreveta	Pod ugovorom
1	Letteris	150.0	3	<input checked="" type="checkbox"/>
2	Nikki	189.0	4	<input type="checkbox"/>
21	Apostolos	154.55	5	<input type="checkbox"/>
34	New Star hotel	333.69	3	<input type="checkbox"/>

Dodaj aranzman

Izmeni aranzman

Obrisi aranzman

Kreiraj ugovor

Pogledaj ugovor

Pretrazi po broju ugovora

Pretrazi aranzmane

Resetuj tabelu

Основни сценарио СК

1. Комерцијалиста уноси вредност по којој претражује аранжмане. (АПУСО)
2. Комерцијалиста позива систем да нађе аранжмане по задатој вредности. (АПСО)
3. Систем тражи аранжмане по задатој вредности. (СО)
4. Систем приказује комерцијалисти нађене аранжман. (ИА)

[Client] - agency v1.0 - Aranzman

Destinacija

Naziv: Lefkada Ukupan kapacitet: 15
 Opis: Najlepse grcko ostrvo! Popunjen kapacitet: 3
 ID Destinacije: 1

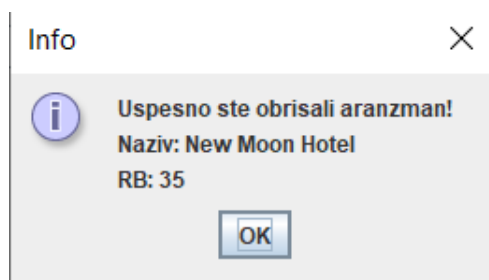
Komercijalista: Filip Gogic

RB aranzmana	Naziv smestaja	Cena po krevetu	Broj kreveta	Pod ugovorom
2	Nikki	189.0	4	<input type="checkbox"/>

Pretrazi aranzmane Resetuj tabelu

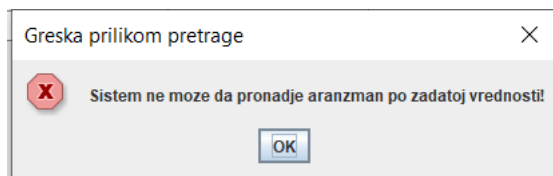
Dodaj aranzman
 Izmeni aranzman
 Obrisi aranzman
 Kreiraj ugovor
 Pogledaj ugovor
 Pretrazi po broju ugovora

5. **Комерцијалиста бира аранжман** чије податке жели да обрише. (АПУСО)
6. **Комерцијалиста позива систем** да обрише **аранжман**. (АПСО)
7. **Систем брише аранжман**. (СО)
8. **Систем приказује комерцијалисти** поруку: “**Систем** је обрисао **аранжман**.” (ИА)

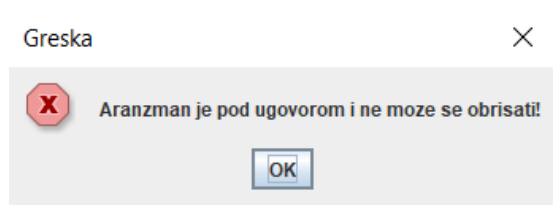


Алтернативна сценарија

- 4.1 Уколико **систем** не може да нађе **аранжман** он приказује **комерцијалисти** поруку: “**Систем** не може да нађе **аранжман** по задатој вредности”. Прекида се извршење сценарија. (ИА)



- 8.1 Уколико **систем** не може да обрише **аранжман** он приказује **комерцијалисти** поруку “**Систем** не може да обрише **аранжман**”. (ИА)



СК8: Случај коришћења - Креирање уговора

Назив СК

Креирање **уговора**

Актори СК

Комерцијалиста

Учесници СК

Комерцијалиста и **систем** (програм)

Предуслов: Систем је укључен и **комерцијалиста** је улогован под својом шифром. Систем приказује форму за рад са **уговором**. Учитана је листа аранжмана.

Ugovor

Ugovor

Destinacija: Lefkada

Smestaj: Nikki

Broj kreveta: 4

Komercijalista: Filip Gogic

Broj ugovora: Broj ugovora će biti kreiran nakon obrade

Broj klijenata: 0 Ukupna cena: 0

RB	Ime i Prezime klijenta	JMBG	Cena
----	------------------------	------	------

Obradi ugovor

Ime i prezime klijenta:

JMBG:

Cena:

Storniraj ugovor

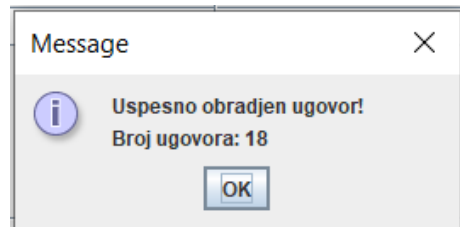
Unesi klijenta

Obrisi klijenta

Odustani

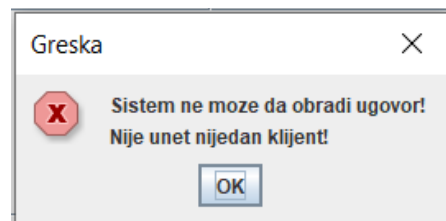
Основни сценарио СК

1. **Комерцијалиста** **уноси** податке у **уговор**. (АПУСО)
2. **Комерцијалиста** **контролише** да ли је коректно унео податке у **уговор**. (АНСО)
3. **Комерцијалиста** **позива** **систем** да обради **уговор**. (АПСО)
4. **Систем** **обрађује** **уговор**. (СО)
5. **Систем** **приказује** **комерцијалисти** обрађен **уговор** и поруку: “**Систем** је обрадио **уговор**”. (ИА)



Алтернативна сценарија

5.1 Уколико **систем** не може да обради **уговор** он приказује **комерцијалисти** поруку: "**Систем** не може да обради **уговор**". (ИА)



СК9: Случај коришћења - Претраживање уговора

Назив СК

Претраживање **уговора**

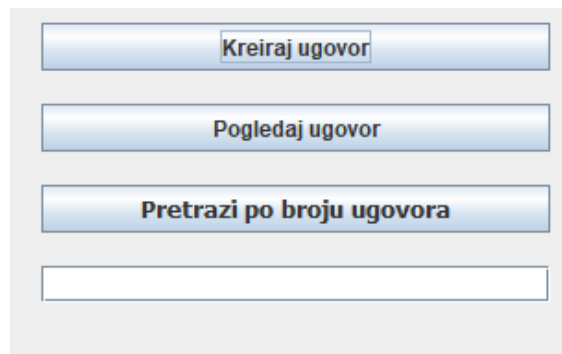
Актори СК

Комерцијалиста

Учесници СК

Комерцијалиста и **систем** (програм)

Предуслов: Систем је укључен и **комерцијалиста** је улогован под својом шифром. Систем приказује форму за рад са **уговором**.



The image shows a web interface with a light gray background. It contains three blue buttons with white text, stacked vertically. The first button is labeled 'Kreiraj ugovor', the second 'Pogledaj ugovor', and the third 'Pretrazi po broju ugovora'. Below these buttons is a white rectangular text input field with a thin gray border.

Основни сценарио СК

1. **Комерцијалиста** **уноси** вредност по којој претражује **уговор**. (АПУСО)
2. **Комерцијалиста** **позива** **систем** да нађе **уговор** по задатој вредности. (АПСО)
3. **Систем** **тражи** **уговор** по задатој вредности. (СО)
4. **Систем** приказује **комерцијалисти** податке о **уговору**. (ИА)

Ugovor

Destinacija: Lefkada

Smestaj: Letteris

Broj kreveta: 3

Komercijalista: Filip Gogic

Broj ugovora: 6

Broj klijenata: 2 Ukupna cena: 298.0

RB	Ime i Prezime klijenta	JMBG	Cena
1	Filip Filipovic	1234567897894	149.0
2	Bla bla	1234657894659	149.0

Obradi ugovor

Storniraj ugovor

Odustani

Ime i prezime klijenta:

JMBG:

Cena:

Unesi klijenta

Obrisi klijenta

Pretrazi aranzmane

Resetuj tabelu

6

Dodaj aranzman

Izmeni aranzman

Obrisi aranzman

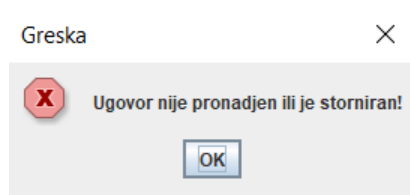
Kreiraj ugovor

Pogledaj ugovor

Pretrazi po broju ugovora

Алтернативна сценарија

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



СК10: Случај коришћења - Сторнирање уговора

Назив СК

Сторнирање **уговора**

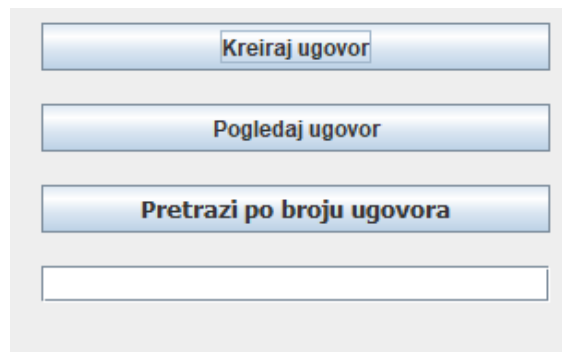
Актори СК

Комерцијалиста

Учесници СК

Комерцијалиста и **систем** (програм)

Предуслов: **Систем** је укључен и **комерцијалиста** је улогован под својом шифром. Систем приказује форму за рад са **уговором**.



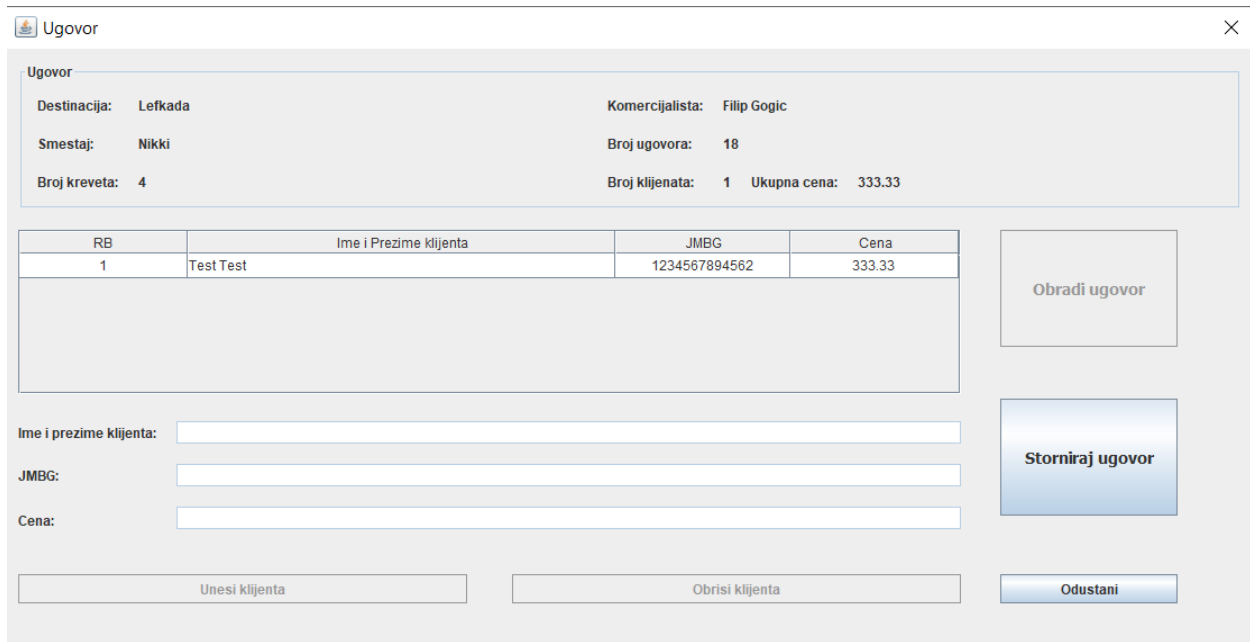
Kreiraj ugovor

Pogledaj ugovor

Pretrazi po broju ugovora

Основни сценарио СК

1. **Комерцијалиста** **уноси** вредност по којој претражује **уговор**. (АПУСО)
2. **Комерцијалиста** **позива** **систем** да нађе **уговор** по задатој вредности. (АПСО)
3. **Систем** **тражи** **уговор** по задатој вредности. (СО)
4. **Систем** приказује **комерцијалисти** **уговор**. (ИА)



Ugovor

Ugovor

Destinacija: Lefkada Komercijalista: Filip Gagic

Smestaj: Nikki Broj ugovora: 18

Broj kreveta: 4 Broj klijenata: 1 Ukupna cena: 333.33

RB	Ime i Prezime klijenta	JMBG	Cena
1	Test Test	1234567894562	333.33

Obradi ugovor

Ime i prezime klijenta:

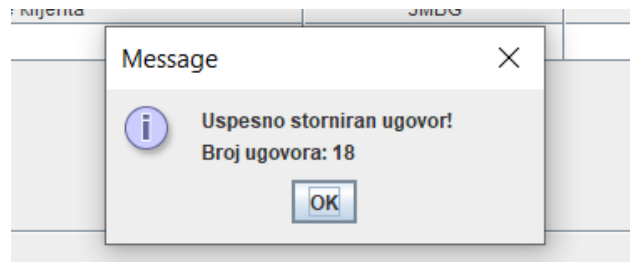
JMBG:

Cena:

Storniraj ugovor

Unesi klijenta Obrisi klijenta Odustani

5. **Комерцијалиста** **позива** **систем** да сторнира задати **уговор**. (АПСО)
6. **Систем** **сторнира** **уговор**. (СО)
7. **Систем** **приказује** **комерцијалисти** сторниран **уговор** и поруку: “**Систем** је сторнирао **уговор**”. (ИА)

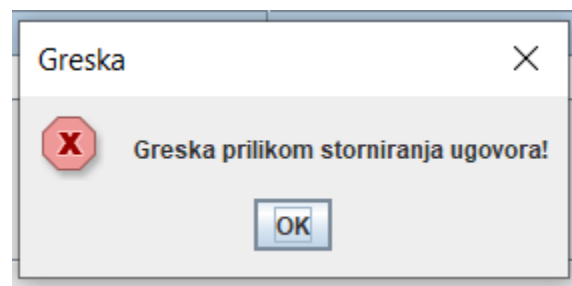


Алтернативна сценарија

- 4.1 Уколико **систем** не може да нађе **уговор** он приказује **комерцијалисти** поруку: “**Систем** не може да нађе **уговор** по задатој вредности”. Прекида се извршење сценарија. (ИА)



- 7.1 Уколико **систем** не може да сторнира **уговор** он приказује **комерцијалисти** поруку: “**Систем** не може да сторнира **уговор**”.



СК11: Случај коришћења - Претраживање клијената

Назив СК

Претраживање клијената

Актори СК

Комерцијалиста

Учесници СК

Комерцијалиста и систем (програм)

Предуслов: Систем је укључен и комерцијалиста је улогован под својом шифром. Систем приказује форму за рад са клијентима.

The screenshot shows a software window titled "Ugovor". It contains a form with the following fields and values:

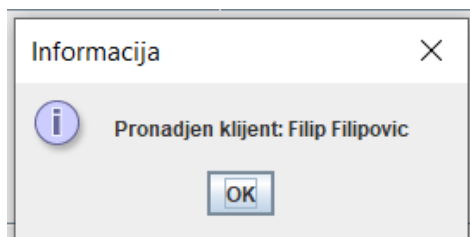
- Destinacija: Lefkada
- Komercijalista: Filip Gogic
- Smestaj: Nikki
- Broj ugovora: Broj ugovora će biti kreiran nakon obrade
- Datum ugovora: 12.09.2021.
- Broj kreveta: 4
- Broj klijenata: 0
- Ukupna cena: 0

Below these fields is a table with columns: RB, Ime i Prezime klijenta, JMBG, and Cena. To the right of the table are two buttons: "Obradi ugovor" and "Storniraj ugovor".

At the bottom, there are three input fields: "Ime i prezime klijenta:", "JMBG:", and "Cena:". To the right of the "JMBG:" field is a button labeled "Pretrazi u bazi", which is highlighted with a red rectangle. Below these fields are three buttons: "Unesi klijenta", "Obrisi klijenta", and "Odustani".

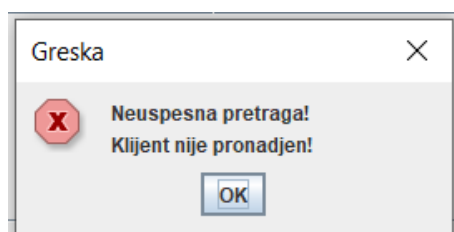
Основни сценарио СК

1. Комерцијалиста уноси вредност по којој претражује клијента. (АПУСО)
2. Комерцијалиста позива систем да нађе клијента по задатој вредности. (АПСО)
3. Систем тражи клијента по задатој вредности. (СО)
4. Систем приказује комерцијалисти податке о клијенту и поруку: "Систем је нашао клијента по задатој вредности". (ИА)



Алтернативна сценарија

4.1 Уколико **систем** не може да нађе **клијента** он приказује **комерцијалисти** поруку: "**Систем** не може да нађе **клијента** по задатој вредности". (ИА)



3.3. Пројектовање апликационе логике

Апликациони сервери треба да обезбеде сервисе који ће омогућити реализацију апликационе логике софтверског система. Пројектовани апликациони сервер садржи:

- Део за комуникацију са клијентом
- Контролер апликационе логике
- Део који садржи пословну логику
- Део за комуникацију са складиштем података (брокер базе података)

3.3.1. Комуникација са клијентима

Део за комуникацију подиже серверски сокет (класа `ServerThread`) који ће да ослушкује мрежу. Када клијентски сокет успостави конекцију са серверским сокетом, тада сервер генерише нит која ће успоставити двосмерну везу са клијентом (класа `ProcessRequests`).

Слање и примање података од клијента се обавља разменом објекта класа `Request` и `Response` и остварује се преко сокета.

```
7
8  import java.io.Serializable;
9
10 /**
11  *
12  * @author Gogic
13  */
14 public class Request implements Serializable{
15     private Operations operation;
16     private Object argument;
17
18     public Request() {
19     }
20
21     public Request(Operations operation, Object argument) {
22         this.operation = operation;
23         this.argument = argument;
24     }
25
26     public Object getArgument() {
27         return argument;
28     }
29
30     public void setArgument(Object argument) {
31         this.argument = argument;
32     }
33
34     public Operations getOperation() {
35         return operation;
36     }
37
38     public void setOperation(Operations operation) {
39         this.operation = operation;
40     }
41
42 }
```

Слика 2 Request

```

8  import java.io.Serializable;
9
10 /**
11  *
12  * @author Gogic
13  */
14 public class Response implements Serializable{
15     private Object result;
16     private Exception exception;
17
18     public Response() {
19     }
20
21     public Response(Object result, Exception exception) {
22         this.result = result;
23         this.exception = exception;
24     }
25
26     public Exception getException() {
27         return exception;
28     }
29
30     public void setException(Exception exception) {
31         this.exception = exception;
32     }
33
34     public Object getResult() {
35         return result;
36     }
37
38     public void setResult(Object result) {
39         this.result = result;
40     }
41
42 }
43

```

Слика 3 Response

Клијент шаље захтев за извршење неке од системских операција до одговарајуће нити која је повезана са тим клијентом. Та нит прихвата захтев и прослеђује га до контролера апликационе логике. Након извршења системске операције, резултат се преко контролера апликационе логике враћа до нити клијента која тај резултат шаље назад до клијента.

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).

3.3.3. Пословна логика

Класе одговорне за системске операције наслеђују класу `GenericOperation`, у којој је дефинисана веза са базом података и омогућује праћење извршења СО као трансакција. Метода `execute(Object param)` садржи позив три методе, од којих једна проверава предуслове, друга отвара конекцију са базом, а трећа извршава операцију. Успешно – потврђивање трансакције, изузетак – поништавање трансакције.

```
public abstract class GenericOperation {

    protected final Repository repositoryDB;

    public GenericOperation() {
        this.repositoryDB = new RepositoryDbGeneric();
    }

    public final Object execute(Object param) throws Exception{
        try {
            preconditions(param);
            startTransaction();
            Object result = executeOperation(param);
            commitTransaction();
            return result;
        } catch (Exception e) {
            e.printStackTrace();
            rollbackTransaction();
            throw e;
        } finally{
            disconnect();
        }
    }

    private void startTransaction() throws Exception {
        ((DbRepository) repositoryDB).connect();
    }

    protected abstract void preconditions(Object param) throws Exception;

    protected abstract Object executeOperation(Object param) throws Exception;

    private void commitTransaction() throws Exception {
        ((DbRepository) repositoryDB).commit();
    }

    private void rollbackTransaction() throws Exception {
        ((DbRepository) repositoryDB).rollback();
    }

    private void disconnect() throws Exception {
        ((DbRepository) repositoryDB).disconnect();
    }
}
```

Слика 4 Generic Operation

За сваку системску операцију треба направити концептуална решења која су директно повезана са логиком проблема. За сваки од уговора се пројектује концептуално решење.

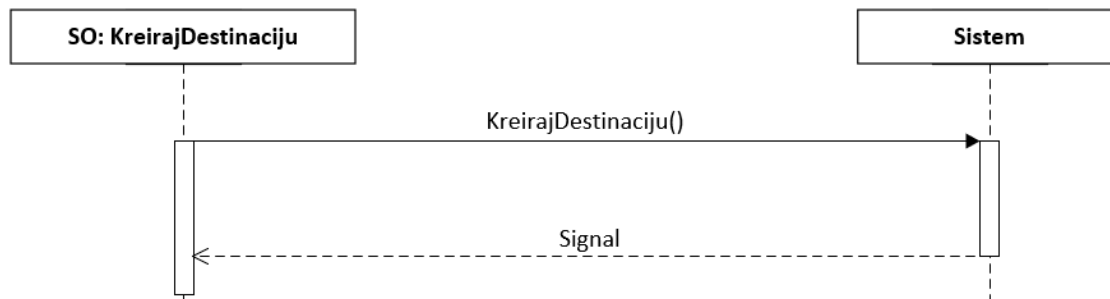
1. Уговор UG1: *KreirajDestinaciju*

Операција: *KreirajDestinaciju* (*Destinacija*):signal;

Веза са СК: СК1

Предуслови: Вредносна и структурна ограничења над објектом *Destinacija* морају бити задовољена.

Постуслови: Направљена је нова дестинација.



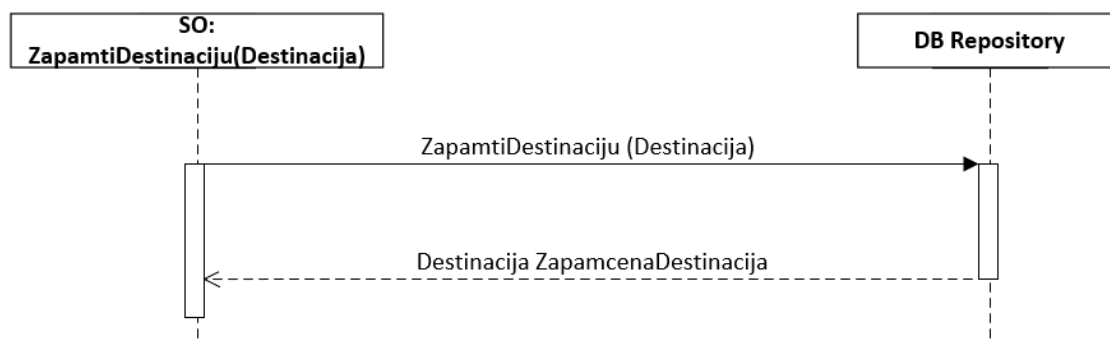
2. Уговор UG2: *ZapamtiDestinaciju*

Операција: *ZapamtiDestinaciju* (*Destinacija*):signal;

Веза са СК: СК3

Предуслови: Вредносна и структурна ограничења над објектом *Destinacija* морају бити задовољена.

Постуслови: Дестинација је запамћена.



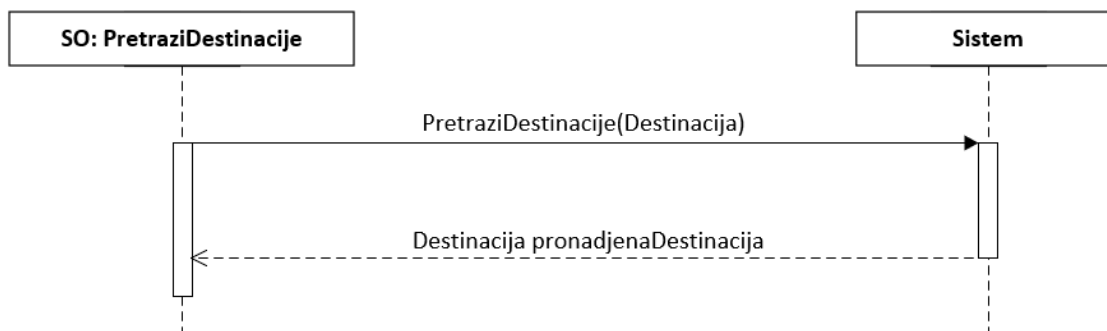
3. Уговор UG3: *PretraziDestinacije*

Операција: **PretraziDestinacije** (*Destinacija*):signal;

Веза са СК: CK2, CK3

Предуслови:

Постуслови: Пронађена је тражена дестинација.



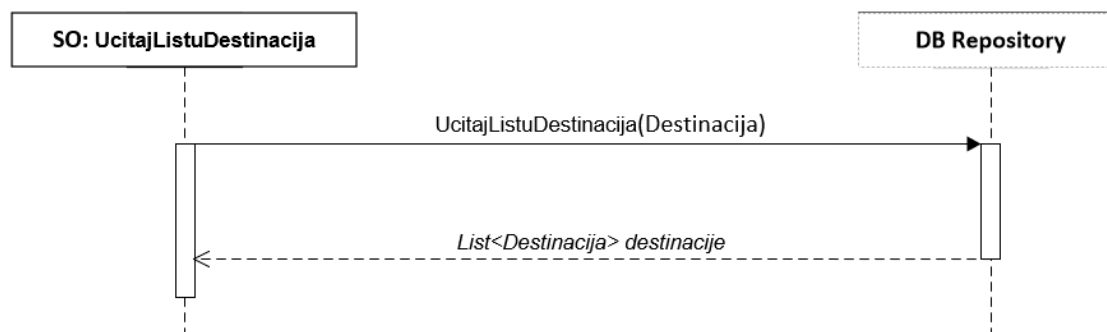
4. Уговор UG4: *UcitajListuDestinacija*

Операција: **UcitajListuDestinacija** (*List<Destinacija>*):signal;

Веза са СК: CK4

Предуслови:

Постуслови: Учитана је листа дестинација.



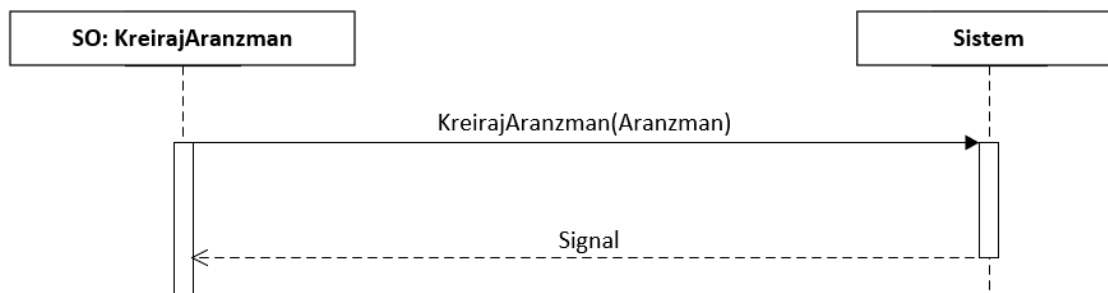
5. Уговор UG5: *KreirajAranzman*

Операција: *KreirajAranzman* (*Aranzman*):signal;

Веза са СК: СК4

Предуслови: Вредносна и структурна ограничења над објектом *Aranzman* морају бити задовољена.

Постуслови: Направљен је нови аранжман.



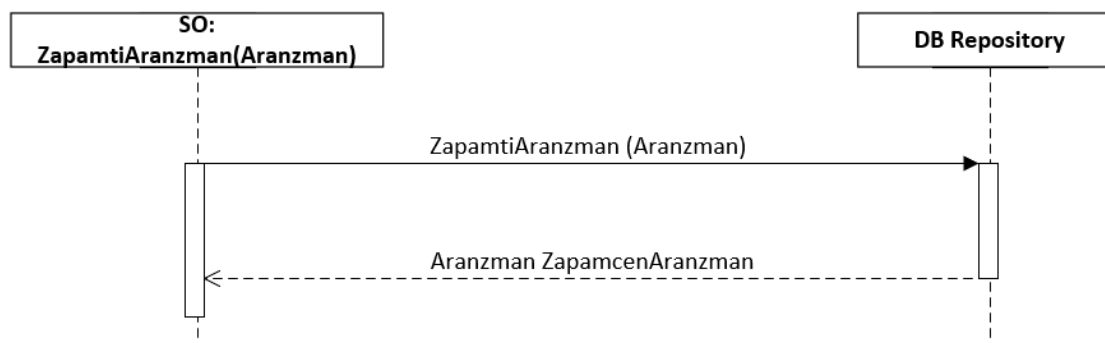
6. Уговор UG6: *ZapamtiAranzman*

Операција: *ZapamtiAranzman* (*Aranzman*):signal;

Веза са СК: СК6

Предуслови: Вредносна и структурна ограничења над објектом *Aranzman* морају бити задовољена.

Постуслови: Аранжман је запамћен.



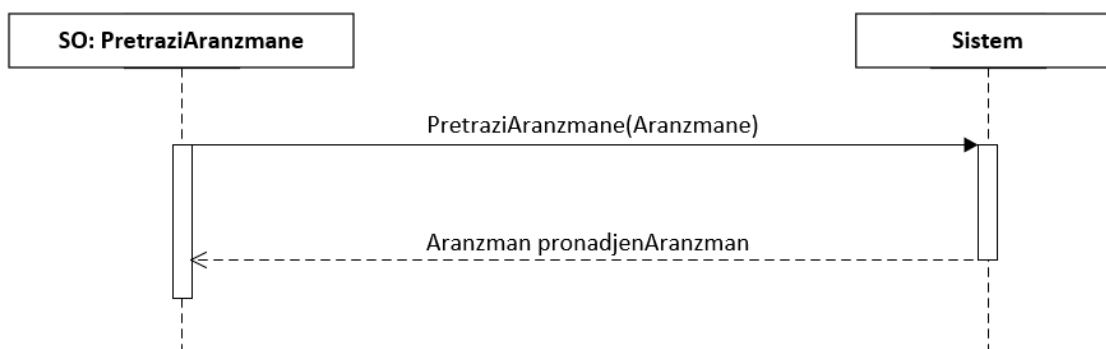
7. Уговор UG7: *PretraziAranzmane*

Операција: **PretraziAranzmane** (*Aranzman*):signal;

Веза са СК: СК5, СК6, СК7

Предуслови:

Постуслови: Пронађен је тражени аранжман.



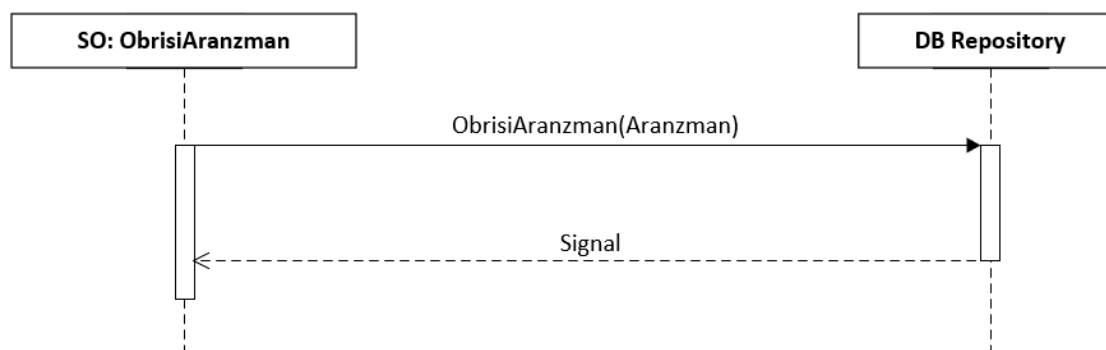
8. Уговор UG8: *ObrisiAranzman*

Операција: **ObrisiAranzman** (*Aranzman*):signal;

Веза са СК: СК7

Предуслови: Вредносна и структурна ограничења над објектом *Aranzman* морају бити задовољена. Ако постоји обрађен или сторниран уговор који одговара објекту *Aranzman* не може се извршити системска операција.

Постуслови: Аранжман је обрисан.



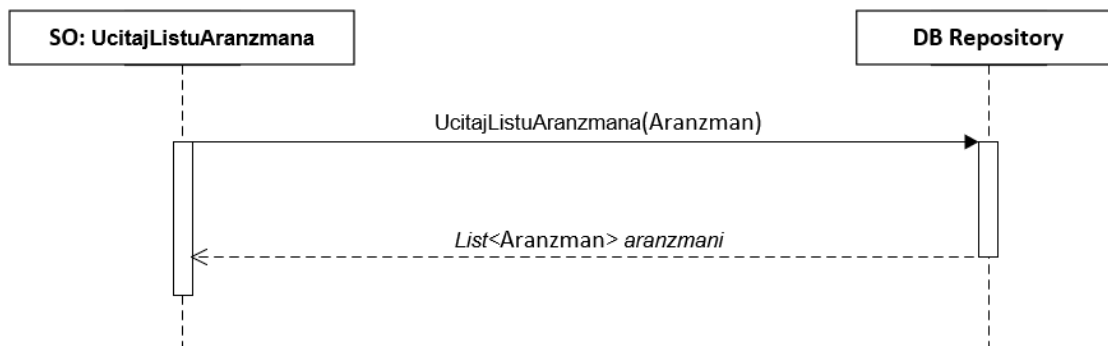
9. Уговор UG9: *UcitajListuAranzmana*

Операција: *UcitajListuAranzmana* (*List<Aranzman>*):signal;

Веза са СК: СК8

Предуслови:

Постуслови: Учитана је листа аранжмана.



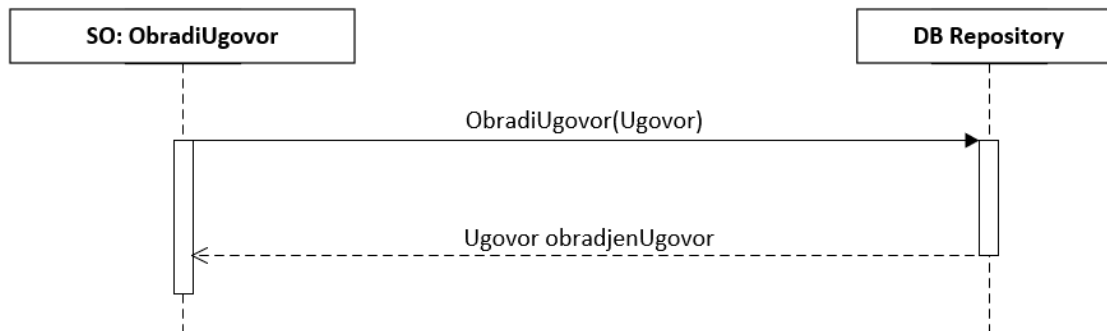
10. Уговор UG10: *ObradiUgovor*

Операција: *ObradiUgovor* (*Ugovor*):signal;

Веза са СК: СК8

Предуслови: Вредносна и структурна ограничења над објектом *Ugovor* морају бити задовољена. Ако је уговор обрађен или сторниран не може се извршити системска операција.

Постуслови: Уговор је обрађен.



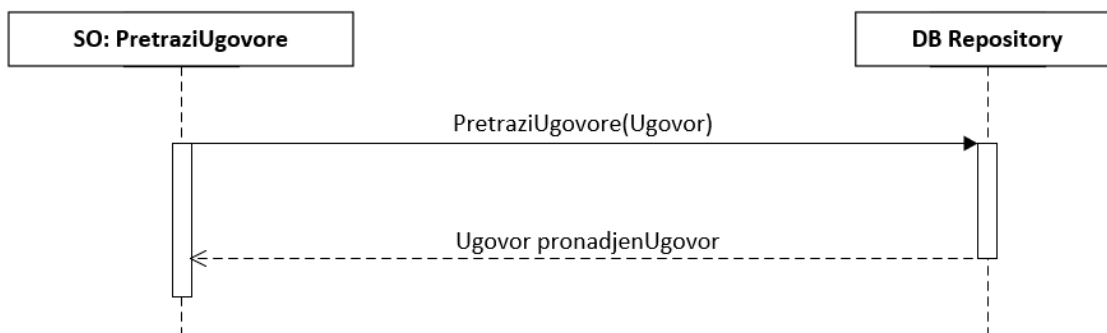
11. Уговор UG11: *PretraziUgovore*

Операција: **PretraziUgovore** (*Ugovor*):signal;

Веза са СК: СК9, СК10

Предуслови:

Постуслови: Пронађен је тражени уговор.



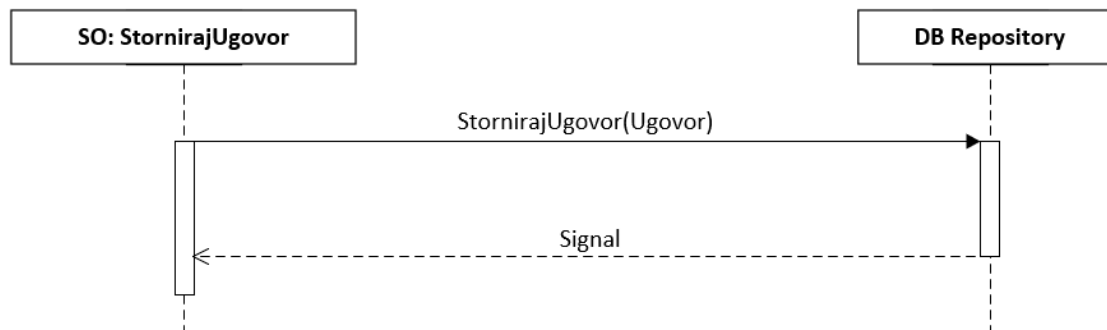
12. Уговор UG12: *StornirajUgovor*

Операција: **StornirajUgovor** (*Ugovor*):signal;

Веза са СК: СК10

Предуслови: Вредносна и структурна ограничења над објектом *Ugovor* морају бити задовољена. Ако је уговор сторниран не може се извршити системска операција.

Постуслови: Уговор је сторниран.



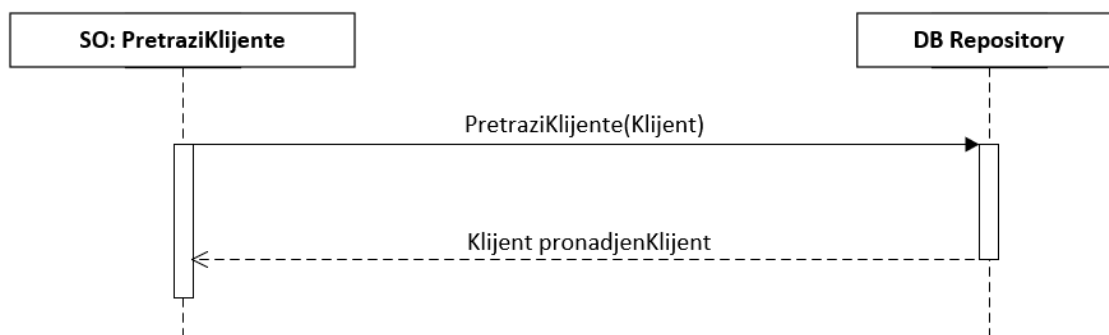
13. Уговор UG13: *PretraziKlijente*

Операција: **PretraziKlijente** (*Klijent*):signal;

Веза са СК: СК11

Предуслови:

Постуслови: Пронађен је тражени клијент.



3.3.4. Брокер базе података

Класе Repository и DbRepository представљају перзистентни оквир који посредује у свим операцијама над базом података и реализује касније наведене методе.

```
package repository;

import java.util.List;

/**
 *
 * @author Gogic
 * @param <T>
 */
public interface Repository <T>{

    public T insert(T parameter) throws Exception;
    public T update(T parameter) throws Exception;
    public boolean delete(T parameter) throws Exception;

    public List<T> getAll(T parameter) throws Exception;
    public List<T> getBy(T parameter, String column, String value) throws Exception;

    public T get(T parameter) throws Exception;
}
```

Слика 5 Repository

```
package repository.db;

import repository.Repository;

/**
 *
 * @author Gogic
 * @param <T>
 */
public interface DbRepository <T> extends Repository<T>{

    default public void connect() throws Exception{
        DbConnectionFactory.getInstance().getConnection();
    }

    default public void disconnect() throws Exception{
        DbConnectionFactory.getInstance().getConnection().close();
    }

    default public void commit() throws Exception{
        DbConnectionFactory.getInstance().getConnection().commit();
    }

    default public void rollback() throws Exception{
        DbConnectionFactory.getInstance().getConnection().rollback();
    }
}
```

Слика 6 DbRepository

Све методе су пројектоване као генеричке – могу да прихвате различите доменске објекте преко параметра. На тај начин избегавамо потребу за имплементацијом појединачних метода за сваку доменску класу. Ово је остварено помоћу интерфејса `GenericObject` кога имплементирају све доменске класе.

```
* @author Gogic
*/
public interface GenericObject extends Serializable{

    public String getTableName();
    public String getColumnNamesForInsert();
    public String getInsertValues();
    public void setID(int ID);

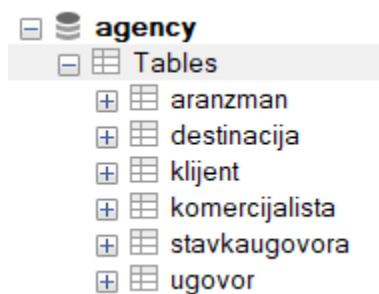
    public GenericObject getObject(ResultSet rs) throws SQLException;
    public String getWhereCase();
    public String getUpdateValues();

    public List<GenericObject> getList(ResultSet rs) throws SQLException;



    public Integer getId();
}
```


3.4. Пројектовање складишта података

На основу концептуалног модела, складиште података је пројектовано коришћењем SQLyog Community 64 – MySQL GUI v13.1.6.




Слика 7 Табеле у бази

	Field	Type
	destinacijaid	int NOT NULL
	rbaranzmana	int NOT NULL
	nazivsmestaja	varchar(50) NULL
	cenapokrevetu	decimal(10,2) NULL
	brojkreveta	int NULL
	podugovorom	tinyint NULL
	komercijalistaid	int NULL


Слика 8 Табела аранжман

	Field	Type
	destinacijaid	int NOT NULL
	mesto	varchar(50) NULL
	opis	varchar(255) NULL
	komercijalistaid	int NULL



Слика 9 Табела дестинација

	Field	Type
	klijentid	int NOT NULL
	jmbg	varchar(13) NOT NULL
	imeprezime	varchar(100) NULL


Слика 10 Табела клијент

	Field	Type
	komercijalistaid	int NOT NULL
	ime	varchar(50) NULL
	prezime	varchar(50) NULL
	username	varchar(20) NULL
	password	varchar(20) NULL

Слика 11 Табела комерцијалиста

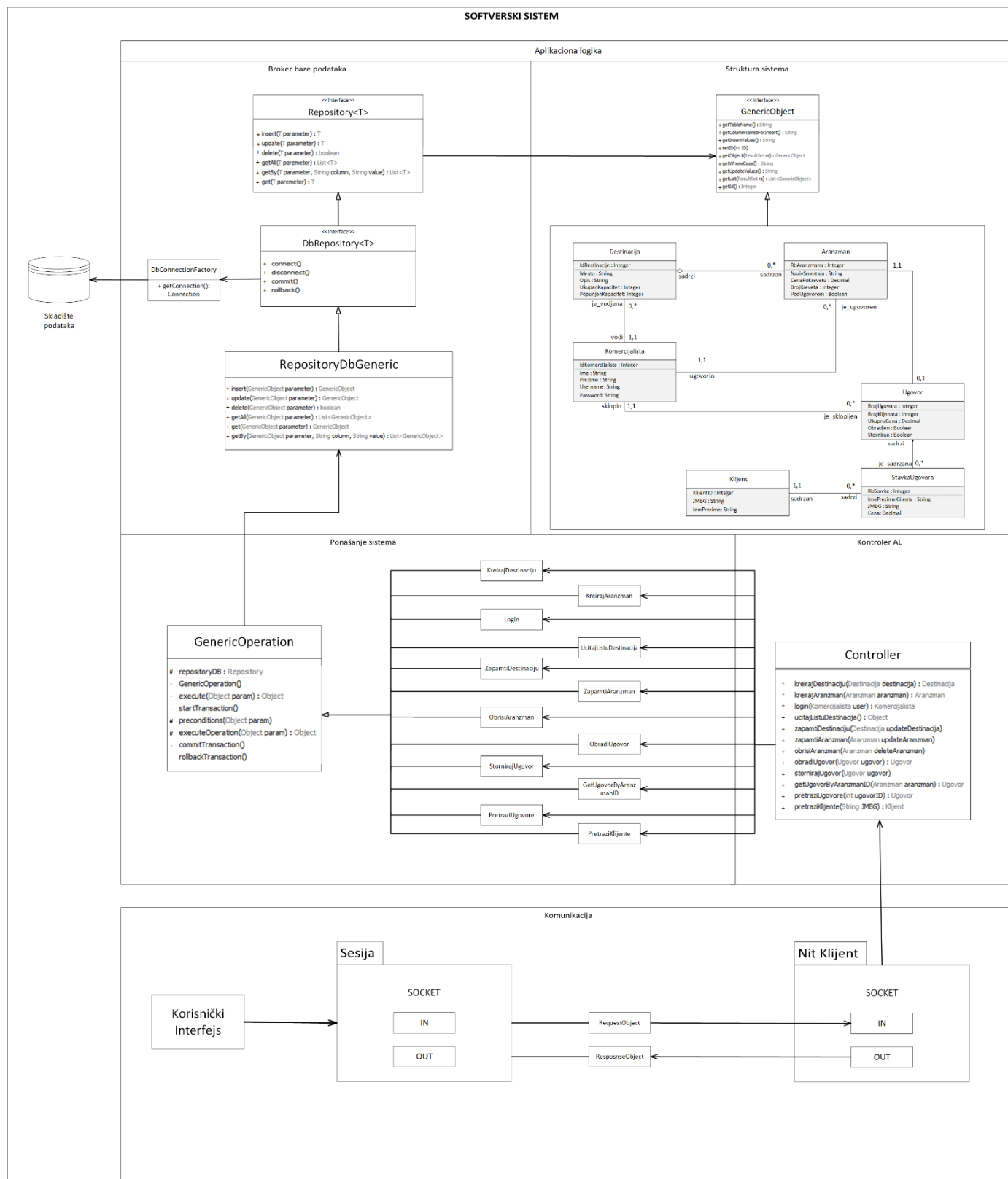
	Field	Type
	brojugovora	int NOT NULL
	rbstavke	int NOT NULL
	imeprezimeklijenta	varchar(100) NULL
	jmbg	varchar(13) NULL
	cena	decimal(10,2) NULL

Слика 12 Табела ставка уговора

	Field	Type
	brojugovora	int NOT NULL
	brojklijenata	int NULL
	ukupnacena	decimal(10,2) NULL
	obradjen	tinyint NULL
	storniran	tinyint NULL
	destinacijaid	int NULL
	rbaranzmana	int NULL
	komercijalistaid	int NULL
	datumsklapanjaugovora	date NULL

Слика 13 Табела уговор

На основу претходних целина, може се саставити цела архитектура софтверског система:



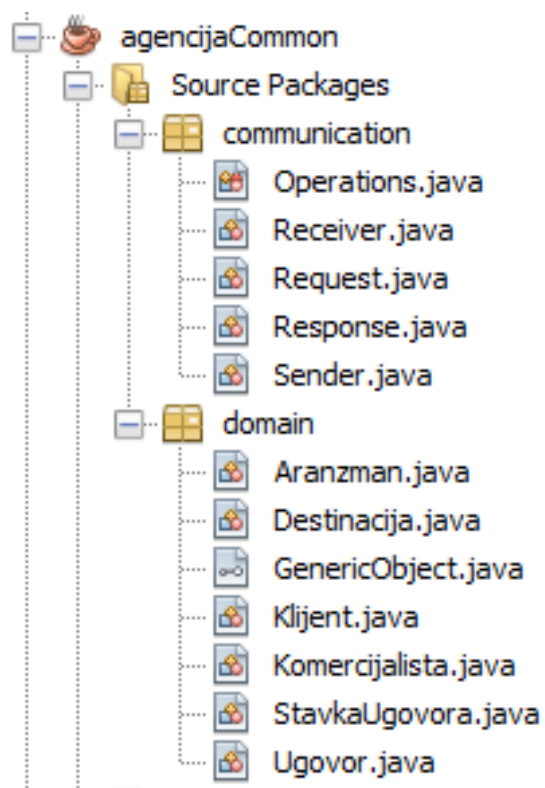
Слика 14 Архитектура софтверског система

4. Имплементација

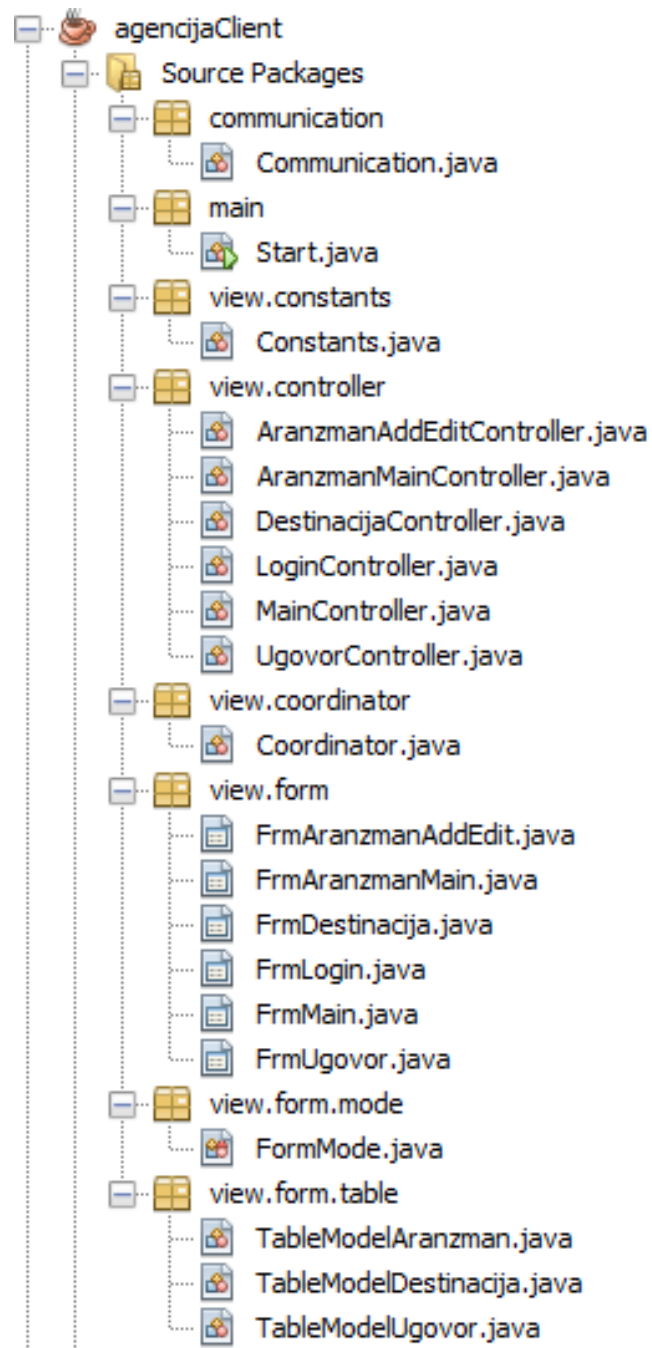
Софтверски систем развијен је у програмском језику Јава и пројектован је као клијент-сервер апликација. Као развојно окружење коришћен је NetBeans IDE 12.1, а као систем за управљање базом података коришћен је SQLyog. Читав систем је реализован у три пројекта:

- `agencijaCommon`
- `agencijaClient`
- `agencijaServer`

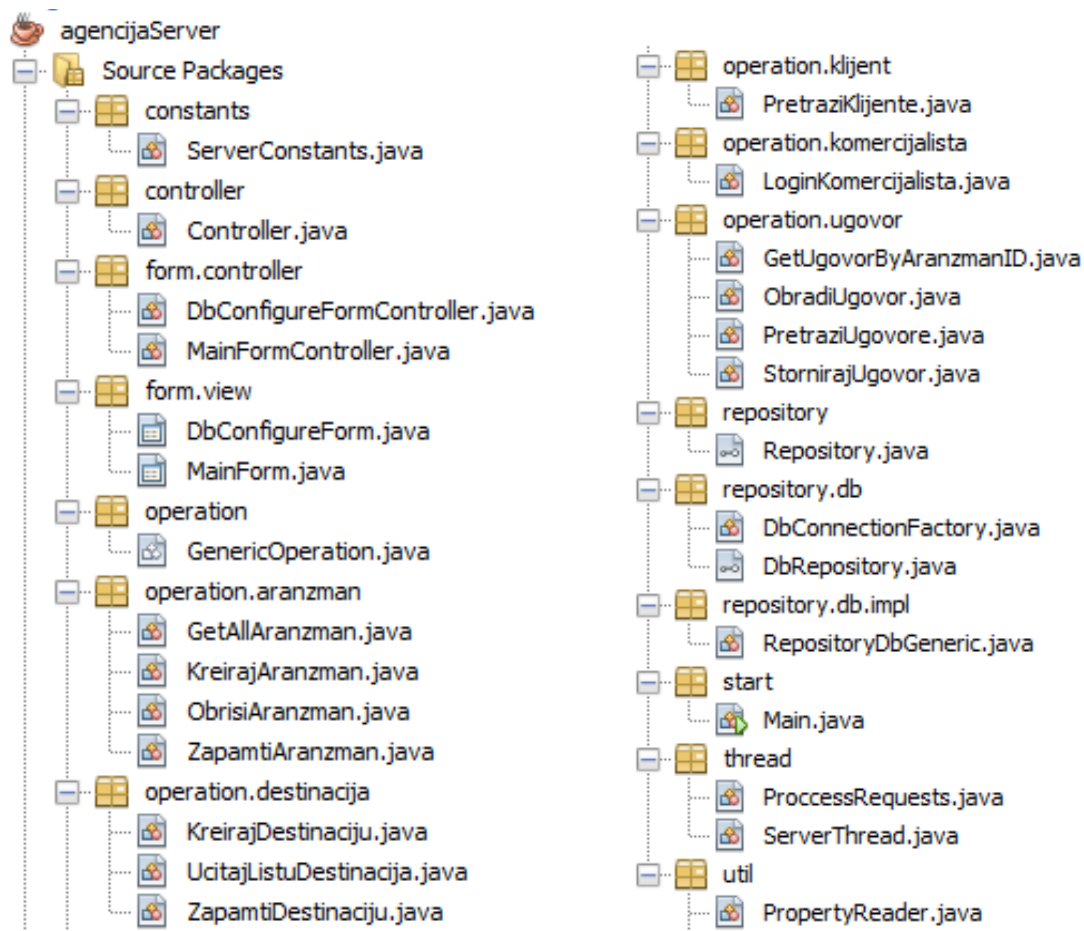
На основу архитектуре софтверског система добијене су следеће софтверске класе:



Слика 15 `agencijaCommon`



Слика 16 agencijaClient



Слика 17 agencijaServer

5. Тестирање

У оквиру пројекта нису коришћени алати за аутоматско тестирање. Тестирање се свело на мануелно тестирање сваког од случаја коришћења, са циљем испитивања како основних, тако и алтернативних сценарија. Тестирани су сценарији у којима се уносе неправилни подаци, прекида веза са сервером, прекида веза са базом података. На основу извршених тестирања су отклоњени уочени недостаци.

6. Литература

[1] др Синиша Влајић (2020): Пројектовање софтвера, Факултет организационих наука, Београд