

Универзитет у Београду

Факултет организационих наука

Лабораторија за софтверско инжењерство

Предмет: Пројектовање софтвера

Семинарски рад

Тема: Развој софтверског система за праћење рада хотела у Јава
окружењу

Ментор:

Mr Војислав Станојевић

Студент:

Бранислав Трајков 0229/2019

Београд, 2022.

Садржaj

1.	Прикупљање корисничких захтева	4
1.1.	Вербални опис	4
1.2.	Случајеви коришћења	4
СК1:	Случај коришћења – Креирање госта.....	6
СК2:	Случај коришћења – Претраживање госта	7
СК3:	Случај коришћења – Измена података госта.....	8
СК4:	Случај коришћења – Брисање госта.....	9
СК5:	Случај коришћења – Креирање собе.....	10
СК6:	Случај коришћења – Претраживање собе	11
СК7:	Случај коришћења – Измена података собе.....	12
СК8:	Случај коришћења – Оцена собе.....	13
СК9:	Случај коришћења – Креирање резервације (сложен СК)	14
СК10:	Случај коришћења – Претраживање резервације (сложен СК).....	15
2.	Анализа	16
2.1.	Системски дијаграми секвенци	16
ДС1:	Дијаграм секвенци случаја коришћења – Креирање госта	16
ДС2:	Дијаграм секвенци случаја коришћења – Претраживање госта.....	16
ДС3:	Дијаграм секвенци случаја коришћења – Измена података госта	18
ДС4:	Дијаграм секвенци случаја коришћења – Брисање госта	19
ДС5:	Дијаграм секвенци случаја коришћења – Креирање собе	21
ДС6:	Дијаграм секвенци случаја коришћења – Претраживање собе	22
ДС7:	Дијаграм секвенци случаја коришћења – Измена података собе	23
ДС8:	Дијаграм секвенци случаја коришћења – Оцена собе.....	25
ДС9:	Дијаграм секвенци случаја коришћења – Креирање резервације.....	27
ДС10:	Дијаграм секвенци случаја коришћења – Претраживање резервације	28
2.2.	Понашање софтверског система – Дефинисање уговора о системским операцијама	30
2.3.	Структура софтверског система – Концептуални(доменски) модел	33
2.4.	Структура софтверског система – релациони модел	33
3.	Пројектовање.....	38
	Архитектура софтверског система	38
3.1.	Пројектовање корисничког интерфејса	39
3.1.1.	Пројектовање екранских форми.....	39
СК1:	Случај коришћења – Креирање госта.....	42
СК2:	Случај коришћења – Претраживање госта	44
СК3:	Случај коришћења – Измена података госта.....	48
СК4:	Случај коришћења – Брисање госта.....	53
СК5:	Случај коришћења – Креирање собе.....	58
СК6:	Случај коришћења – Претраживање собе	60
СК7:	Случај коришћења – Измена података собе.....	64

СК8: Случај коришћења – Оцена собе.....	69
СК9: Случај коришћења – Креирање резервације (сложен СК)	74
СК10: Случај коришћења – Претраживање резервације (сложен СК).....	78
3.1.2. Пројектовање контролера корисничког интерфејса.....	82
3.2. Пројектовање апликационе логике	82
3.2.1. Контролер апликационе логике.....	82
3.2.2. Пословна логика	83
3.2.3. Брокер базе података	92
3.3. Пројектовање складишта података	93
4. Имплементација.....	96
5. Тестирање	98

1. Прикупљање корисничких захтева

1.1. Вербални опис

Семинарски рад се односи на израду апликације која ће водити евиденцију о раду хотелског смештаја. У оквиру апликације водиће се евиденција о гостима који посећују конкретан хотел, евиденција свих соба које хотел има на располагању као и евиденција уговорених резервација соба од стране гостију који планирају да посете и користе услуге хотелског смештаја.

Сама апликација омогућава функционисање на следећи начин. Дакле, апликација омогућава да рецепционер, приликом боравка нових гостију, те госте у апликацији креира и њихове податке унесе у систем. Уколико постоји потреба за тиме, рецепционер има могућност да податке гостију, у случају грешке, измени и те измене сачува у систем. За потребе стицања увида о гостима, рецепционер може вршити претраживање њихових података. У случају да одређени гост последњих година није боравио у хотелу, могуће је његово брисање од стране рецепционера. Што се тиче соба, уколико се хотел прошири и тиме увећа број расположивих јединица, нове собе рецепционер може додати у систем. Над свим собама се такође може вршити претраживање и, у случају реновирања постојећих соба, податке о њима рецепционер у систему може променити. Гост, поред тога што може вршити претраживање соба, може свој утисак о соби и боравку у истој изразити путем њеног оцењивања. Коначно, рецепционер, сваку нову резервацију соба од стране гостију, може унети у систем и, ако постоји накнадна потреба за подацима о некој резервацији, рецепционер до података може доћи претраживањем постојећих резервација.

1.2. Случајеви коришћења

У овој апликацији идентификовано је десет случајева коришћења:

1. Креирање госта
2. Претраживање госта
3. Измена података госта
4. Брисање госта
5. Креирање собе
6. Претраживање собе
7. Измена података собе
8. Оцена собе
9. Креирање резервације (сложен СК)
10. Претраживање резервације (сложен СК).



Слика 1. Модел случајева коришћења

СК1: Случај коришћења – Креирање госта

Назив СК

Креирање госта

Актори СК

Рецепционер

Учесници СК

Рецепционер и систем (програм)

Предуслов: Систем је укључен и рецепционер је пријављен под својом шифром. Систем приказује форму за рад са гостом. Учитана је листа градова.

Основни сценарио СК

1. Рецепционер уноси податке о госту. (АПУСО)
2. Рецепционер контролише да ли је коректно унео податке о госту. (АНСО)
3. Рецепционер позива систем да запамти податке о госту. (АПСО)
4. Систем памти податке о госту. (СО)
5. Систем приказује рецепционеру запамћеног госта и поруку: „Систем је запамтио госта“. (ИА)

Алтернативна сценарија

- 5.1 Уколико систем не може да запамти податке о госту он приказује рецепционеру поруку: „Систем не може да запамти госта“. (ИА)

СК2: Случај коришћења – Претраживање госта

Назив СК

Претраживање госта

Актори СК

Рецепционер

Учесници СК

Рецепционер и систем (програм)

Предуслов: Систем је укључен и **рецепционер** је пријављен под својом шифром. Систем приказује форму за рад са гостом.

Основни сценарио СК

1. Рецепционер уноси вредност по којој претражује госте. (АПУСО)
2. Рецепционер позива систем да нађе госте по задатој вредности. (АПСО)
3. Систем тражи госте по задатој вредности. (СО)
4. Систем приказује рецепционеру податке о гостима и поруку: "Систем је нашао госте по задатој вредности". (ИА)
5. Рецепционер бира госта. (АПУСО)
6. Рецепционер позива систем да учита госта. (АПСО)
7. Систем учитава госта. (СО)
8. Систем приказује рецепционеру податке о госту и поруку: „Систем је учитао госта“. (ИА)

Алтернативна сценарија

- 4.1. Уколико систем не може да нађе госте он приказује рецепционеру поруку: "Систем не може да нађе госте по задатој вредности". Прекида се извршење сценарија. (ИА)
- 8.1. Уколико систем не може да учита госта он приказује рецепционеру поруку: „Систем не може да учита госта“. (ИА)

СК3: Случај коришћења – Измена података госта

Назив СК

Измена података госта

Актори СК

Рецепционер

Учесници СК

Рецепционер и систем (програм)

Предуслов: Систем је укључен и рецепционер је пријављен под својом шифром. Систем приказује форму за рад са гостом. Учитана је листа градова.

Основни сценарио СК

1. Рецепционер уноси вредност по којој претражује госте. (АПУСО)
2. Рецепционер позива систем да нађе госте по задатој вредности. (АПСО)
3. Систем тражи госте по задатој вредности. (СО)
4. Систем приказује рецепционеру податке о гостима и поруку: "Систем је нашао госте по задатој вредности". (ИА)
5. Рецепционер бира госта. (АПУСО)
6. Рецепционер позива систем да учита госта. (АПСО)
7. Систем учитава госта. (СО)
8. Систем приказује рецепционеру податке о госту и поруку: „Систем је учитао госта“. (ИА)
9. Рецепционер уноси (мења) податке о госту. (АПУСО)
10. Рецепционер контролише да ли је исправно унео податке о госту. (АНСО)
11. Рецепционер позива систем да запамти податке о госту. (АПСО)
12. Систем памти податке о госту. (СО)
13. Систем приказује рецепционеру запамћеног госта и поруку: „Систем је запамтио госта“. (ИА)

Алтернативна сценарија

- 4.1. Уколико систем не може да нађе госте он приказује рецепционеру поруку: „Систем не може да нађе госте по задатој вредности“. Прекида се извршење сценарија. (ИА)
- 8.1. Уколико систем не може да учита госта он приказује рецепционеру поруку „Систем не може да учита госта“. Прекида се извршење сценарија. (ИА)
- 13.1. Уколико систем не може да запамти податке о госту он приказује рецепционеру поруку: „Систем не може да запамти госта“. (ИА)

СК4: Случај коришћења – Брисање госта

Назив СК

Брисање госта

Актори СК

Рецепционер

Учесници СК

Рецепционер и систем (програм)

Предуслов: Систем је укључен и рецепционер је пријављен под својом шифром. Систем приказује форму за рад са гостом.

Основни сценарио СК

1. Рецепционер уноси вредност по којој претражује госте. (АПУСО)
2. Рецепционер позива систем да нађе госте по задатој вредности. (АПСО)
3. Систем тражи госте по задатој вредности. (СО)
4. Систем приказује рецепционеру податке о гостима и поруку: „Систем је нашао госте по задатој вредности“. (ИА)
5. Рецепционер бира госта. (АПУСО)
6. Рецепционер позива систем да учита госта. (АПСО)
7. Систем учитава госта. (СО)
8. Систем приказује рецепционеру податке о госту и поруку: „Систем је учитао госта“. (ИА)
9. Рецепционер позива систем да обрише податке о госту. (АПСО)
10. Систем брише податке о госту. (СО)
11. Систем приказује рецепционеру поруку: „Систем је обрисао податке о госту.“ (ИА)

Алтернативна сценарија

- 4.1. Уколико систем не може да нађе госте он приказује рецепционеру поруку: „Систем не може да нађе госте по задатој вредности“. Прекида се извршење сценарија. (ИА)
- 8.1. Уколико систем не може да учита госта он приказује рецепционеру поруку: „Систем не може да учита госта“. Прекида се извршење сценарија. (ИА)
- 11.1. Уколико систем не може да обрише податке о госту он приказује рецепционеру поруку: „Систем не може да обрише госта“. (ИА)

СК5: Случај коришћења – Креирање собе

Назив СК

Креирање собе

Актори СК

Рецепционер

Учесници СК

Рецепционер и систем (програм)

Предуслов: Систем је укључен и рецепционер је пријављен под својом шифром. Систем приказује форму за рад са собом. Учитана је листа врста соба.

Основни сценарио СК

1. Рецепционер уноси податке о соби. (АПУСО)
2. Рецепционер контролише да ли је коректно унео податке о соби. (АНСО)
3. Рецепционер позива систем да запамти податке о соби. (АПСО)
4. Систем памти податке о соби. (СО)
5. Систем приказује рецепционеру запамћену собу и поруку: „Систем је запамтио собу“. (ИА)

Алтернативна сценарија

- 5.1. Уколико систем не може да запамти податке о соби он приказује рецепционеру поруку: „Систем не може да запамти собу“. (ИА)

СК6: Случај коришћења – Претраживање собе

Назив СК

Претраживање **собе**

Актори СК

Рецепционер/гост

Учесници СК

Рецепционер/гост и **систем** (програм)

Предуслов: **Систем** је укључен и **рецепционер/гост** је пријављен под својом шифром. Систем приказује форму за рад са **собом**.

Основни сценарио СК

1. **Рецепционер/гост уноси** вредност по којој претражује **собе**. (АПУСО)
2. **Рецепционер/гост позива** **систем** да нађе **собе** по задатој вредности. (АПСО)
3. **Систем тражи** **собе** по задатој вредности. (СО)
4. **Систем приказује** **рецепционеру/госту** податке о **собама** и поруку: „**Систем** је нашао **собе** по задатој вредности“. (ИА)
5. **Рецепционер/гост бира** **собу**. (АПУСО)
6. **Рецепционер/гост позива** **систем** да учита **собу**. (АПСО)
7. **Систем учитава** **собу**. (СО)
8. **Систем приказује** **рецепционеру/госту** податке о **соби** и поруку: „**Систем** је учитао **собу**“. (ИА)

Алтернативна сценарија

- 4.1. Уколико **систем** не може да нађе **собе** он приказује **рецепционеру/госту** поруку: „**Систем** не може да нађе **собе** по задатој вредности“. Прекида се извршење сценарија. (ИА)
- 8.1. Уколико **систем** не може да учита **собу** он приказује **рецепционеру/госту** поруку: „**Систем** не може да учита **собу**“. (ИА)

СК7: Случај коришћења – Измена података собе

Назив СК

Измена података собе

Актори СК

Рецепционер

Учесници СК

Рецепционер и систем (програм)

Предуслов: Систем је укључен и рецепционер је пријављен под својом шифром. Систем приказује форму за рад са собом. Учитана је листа врста соба.

Основни сценарио СК

1. Рецепционер уноси вредност по којој претражује собе. (АПУСО)
2. Рецепционер позива систем да нађе собе по задатој вредности. (АПСО)
3. Систем тражи собе по задатој вредности. (СО)
4. Систем приказује рецепционеру податке о собама и поруку: „Систем је нашао собе по задатој вредности“. (ИА)
5. Рецепционер бира собу. (АПУСО)
6. Рецепционер позива систем да учита собу. (АПСО)
7. Систем учитава собу. (СО)
8. Систем приказује рецепционеру податке о соби и поруку: „Систем је учитао собу“. (ИА)
9. Рецепционер уноси (мења) податке о соби. (АПУСО)
10. Рецепционер контролише да ли је исправно унео податке о соби. (АНСО)
11. Рецепционер позива систем да запамти податке о соби. (АПСО)
12. Систем памти податке о соби. (СО)
13. Систем приказује рецепционеру изменјену собу и поруку: „Систем је запамтио собу“. (ИА)

Алтернативна сценарија

- 4.1. Уколико систем не може да нађе собе он приказује рецепционеру поруку: „Систем не може да нађе собе по задатој вредности“. Прекида се извршење сценарија. (ИА)
- 8.1. Уколико систем не може да учита собу он приказује рецепционеру поруку: „Систем не може да учита собу“. Прекида се извршење сценарија. (ИА)
- 13.1. Уколико систем не може да запамти податке о соби он приказује рецепционеру поруку: „Систем не може да запамти собу“. (ИА)

СК8: Случај коришћења – Оцена собе

Назив СК

Оцена собе

Актори СК

Гост

Учесници СК

Гост и систем (програм)

Предуслов: Систем је укључен и гост је пријављен под својом шифром. Систем приказује форму за рад са собом.

Основни сценарио СК

1. Гост уноси вредност по којој претражује собе. (АПУСО)
2. Гост позива систем да нађе собе по задатој вредности. (АПСО)
3. Систем тражи собе по задатој вредности . (СО)
4. Систем приказује госту податке о собама и поруку : „Систем је нашао собе по задатој вредности“. (ИА)
5. Гост бира собу. (АПУСО)
6. Гост позива систем да учита собу. (АПСО)
7. Систем учитава собу. (СО)
8. Систем приказује госту податке о соби и поруку: „Систем је учитао собу“. (ИА)
9. Гост уноси (мења) оцену собе. (АПУСО)
10. Гост контролише да ли је исправно унео оцену собе. (АНСО)
11. Гост позива систем да запамти оцену собе. (АПСО)
12. Систем памти оцену собе. (СО)
13. Систем приказује госту запамћену оцену собе и поруку: „Систем је запамтио оцену собе“. (ИА)

Алтернативна сценарија

- 4.1. Уколико систем не може да нађе собе он приказује госту поруку: „Систем не може да нађе собе по задатој вредности“. Прекида се извршење сценарија. (ИА)
- 8.1. Уколико систем не може да учита собе он приказује госту поруку: „Систем не може да учита собу“. Прекида се извршење сценарија. (ИА)
- 13.1. Уколико систем не може да запамти оцену собе он приказује госту поруку: „Систем не може да запамти оцену собе“. (ИА)

СК9: Случај коришћења – Креирање резервације (сложен СК)

Назив СК

Креирање резервације

Актори СК

Рецепционер

Учесници СК

Рецепционер и систем (програм)

Предуслов: Систем је укључен и рецепционер је пријављен под својом шифром. Систем приказује форму за рад са резервацијом. Учитана је листа гостију и листа соба.

Основни сценарио СК

1. Рецепционер уноси податке о резервацији. (АПУСО)
2. Рецепционер контролише да ли је коректно унео податке о резервацији. (АНСО)
3. Рецепционер позива систем да запамти податке о резервацији. (АПСО)
4. Систем памти податке о резервацији. (СО)
5. Систем приказује рецепционеру запамћену резервацију и поруку: „Систем је запамтио резервацију“. (ИА)

Алтернативна сценарија

- 5.1. Уколико систем не може да запамти податке о резервацији он приказује рецепционеру поруку: „Систем не може да запамти резервацију“. (ИА)

СК10: Случај коришћења – Претраживање резервације (сложен СК)

Назив СК

Претраживање резервације

Актори СК

Рецепционер

Учесници СК

Рецепционер и систем (програм)

Предуслов: Систем је укључен и рецепционер је пријављен под својом шифром. Систем приказује форму за рад са резервацијом.

Основни сценарио СК

1. Рецепционер уноси вредност по којој претражује резервације. (АПУСО)
2. Рецепционер позива систем да нађе резервације по задатој вредности. (АПСО)
3. Систем тражи резервације по задатој вредности. (СО)
4. Систем приказује рецепционеру податке о резервацијама и поруку: „Систем је нашао резервације по задатој вредности“. (ИА)
5. Рецепционер бира резервацију. (АПУСО)
6. Рецепционер позива систем да учита резервацију. (АПСО)
7. Систем учитава резервацију. (СО)
8. Систем приказује рецепционеру податке о резервацији и поруку: „Систем је учитao резервацију“. (ИА)

Алтернативна сценарија

- 4.1. Уколико систем не може да нађе резервације он приказује рецепционеру поруку: „Систем не може да нађе резервације по задатој вредности“. Прекида се извршење сценарија. (ИА)
- 8.1. Уколико систем не може да учита резервацију он приказује рецепционеру поруку: „Систем не може да учита резервацију“. (ИА)

2. Анализа

2.1. Системски дијаграми секвенци

ДС1: Дијаграм секвенци случаја коришћења – Креирање госта

1. **Форма** позива **систем** да учита листу градова. (АПСО)
2. **Систем** враћа **форми** листу градова. (ИА)
3. **Рецепционер** позива **систем** да запамти податке о **госту**. (АПСО)
4. **Систем** приказује **рецепционеру** запамћеног **госта** и поруку: „**Систем** је запамтио **госта**“. (ИА)



Алтернативна сценарија

- 4.1 Уколико **систем** не може да запамти податке о **госту** он приказује **рецепционеру** поруку: „**Систем** не може да запамти **госта**“. (ИА)



Са наведених секвенцних дијаграма уочавају се 2 системске операције:

1. *signal* UcitajListuGradova(List<Grad>);
2. *signal* ZapamtiGosta(Gost).

ДС2: Дијаграм секвенци случаја коришћења – Претраживање госта

1. **Рецепционер** позива **систем** да нађе **госте** по задатој вредности. (АПСО)
2. **Систем** приказује **рецепционеру** податке о **гостима** и поруку: “**Систем** је нашао **госте** по задатој вредности”. (ИА)
3. **Рецепционер** позива **систем** да учита **госта**. (АПСО)

4. Систем приказује рецепционеру податке о госту и поруку: „Систем је учитао госта“. (ИА)



Алтернативна сценарија

2.1. Уколико систем не може да нађе госте он приказује рецепционеру поруку: „Систем не може да нађе госте по задатој вредности“. Прекида се извршење сценарија. (ИА)



4.1. Уколико систем не може да учита госта он приказује рецепционеру поруку: „Систем не може да учита госта“. (ИА)

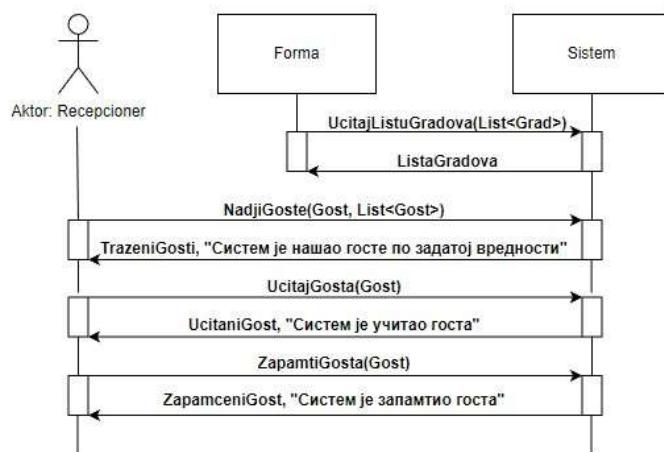


Са наведених секвенцних дијаграма уочавају се 2 системске операције:

1. *signal* NadjiGoste(Gost, List<Gost>);
2. *signal* UcitajGosta(Gost).

ДС3: Дијаграм секвенци случаја коришћења – Измена података госта

1. **Форма** позива **систем** да учита листу градова. (АПСО)
2. **Систем** враћа **форми** листу градова. (ИА)
3. **Рецепционер** позива **систем** да нађе **госте** по задатој вредности. (АПСО)
4. **Систем** приказује **рецепционеру** податке о **гостима** и поруку: „**Систем** је нашао госте по задатој вредности“. (ИА)
5. **Рецепционер** позива **систем** да учита **госта**. (АПСО)
6. **Систем** приказује **рецепционеру** податке о **госту** и поруку: „**Систем** је учитао **госта**“. (ИА)
7. **Рецепционер** позива **систем** да запамти податке о **госту**. (АПСО)
8. **Систем** приказује **рецепционеру** запамћеног **госта** и поруку: „**Систем** је запамтио **госта**“. (ИА)



Алтернативна сценарија

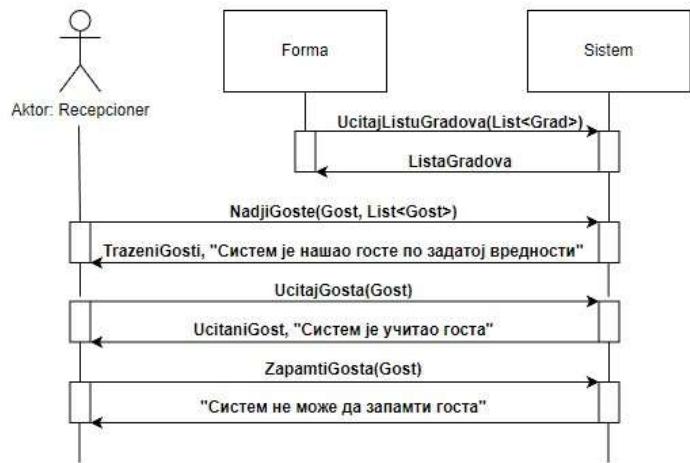
- 4.1. Уколико **систем** не може да нађе **госте** он приказује **рецепционеру** поруку: „**Систем** не може да нађе **госте** по задатој вредности“. Прекида се извршење сценарија. (ИА)



- 6.1. Уколико **систем** не може да учита **госта** он приказује **рецепционеру** поруку „**Систем** не може да учита **госта**“. Прекида се извршење сценарија. (ИА)



8.1. Уколико **систем** не може да запамти податке о **госту** он приказује **рецепционеру** поруку: „**Систем** не може да запамти **госта**“. (ИА)

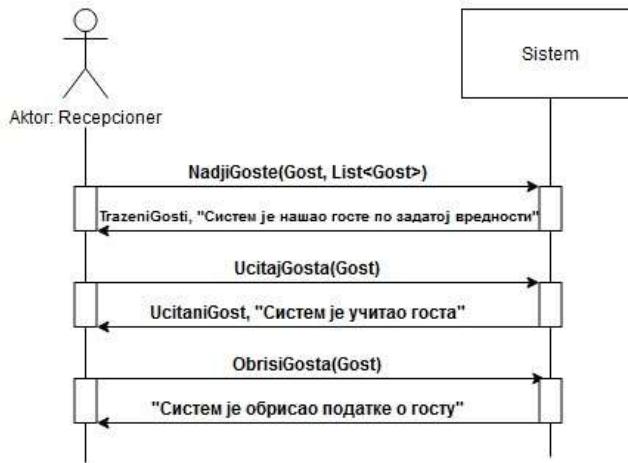


Са наведених секвенцних дијаграма уочавају се 4 системске операције:

1. *signal* UcitajListuGradova(List<Grad>);
2. *signal* NadjiGoste(Gost, List<Gost>);
3. *signal* UcitajGosta(Gost);
4. *signal* ZapamtiGosta(Gost).

ДС4: Дијаграм секвенци случаја коришћења – Брисање госта

1. **Рецепционер** **позива** **систем** да нађе **госте** по задатој вредности. (АПСО)
2. **Систем** **приказује** **рецепционеру** податке о **гостима** и поруку: „**Систем** је нашао **госте** по задатој вредности“. (ИА)
3. **Рецепционер** **позива** **систем** да учита **госта**. (АПСО)
4. **Систем** **приказује** **рецепционеру** податке о **госту** и поруку: „**Систем** је учитао **госта**“. (ИА)
5. **Рецепционер** **позива** **систем** да обрише податке о **госту**. (АПСО)
6. **Систем** **приказује** **рецепционеру** поруку: “**Систем** је обрисао податке о **госту**.” (ИА)



Алтернативна сценарија

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



4.1. Уколико **систем** не може да учита **госта** он приказује **рецепционеру** поруку: „**Систем** не може да учита **госта**“. Прекида се извршење сценарија. (ИА)



6.1. Уколико **систем** не може да обрише податке о **госту** он приказује **рецепционеру** поруку: „**Систем** не може да обрише **госта**“. (ИА)



Са наведених секвенцних дијаграма уочавају се 3 системске операције:

1. *signal* NadjiGoste(Gost, List<Gost>);
2. *signal* UcitajGosta(Gost);
3. *signal* ObrisGosta(Gost).

ДС5: Дијаграм секвенци случаја коришћења – Креирање собе

1. **Форма** **позива** **систем** да учита листу врста соба. (АПСО)
2. **Систем** **враћа** **форми** листу врста соба. (ИА)
3. **Рецепционер** **позива** **систем** да запамти податке о **соби**. (АПСО)
4. **Систем** **приказује** **рецепционеру** запамћену **собу** и поруку: „**Систем** је запамтио **собу**“. (ИА)



Алтернативна сценарија

- 4.1. Уколико **систем** не може да запамти податке о **соби** он приказује **рецепционеру** поруку: „**Систем** не може да запамти **собу**“. (ИА)

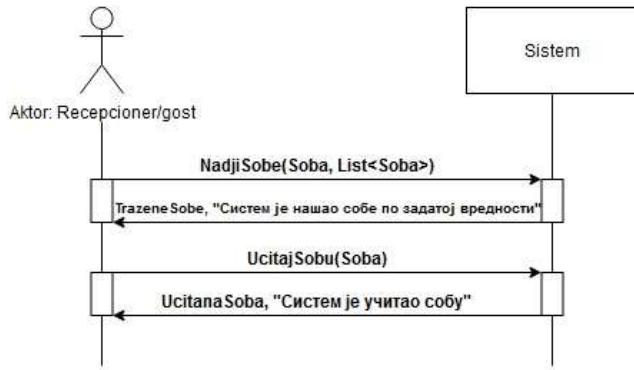


Са наведених секвенцних дијаграма уочавају се 2 системске операције:

1. *signal* `UcitajListuVrstaSoba(List<VrstaSobe>)`;
2. *signal* `ZapamtiSobu(Soba)`.

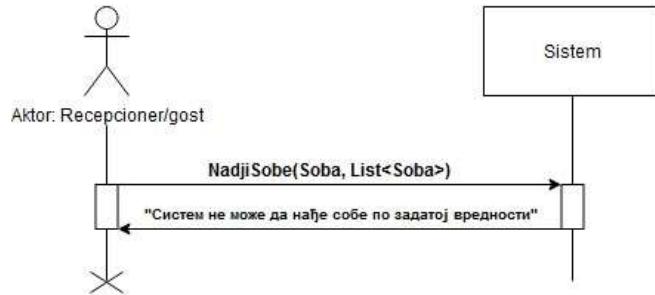
ДС6: Дијаграм секвенци случаја коришћења – Претраживање собе

1. **Рецепционер/гост позива систем** да нађе собе по задатој вредности. (АПСО)
2. **Систем приказује рецепционеру/госту** податке о **собама** и поруку: „**Систем** је нашао **собе** по задатој вредности“. (ИА)
3. **Рецепционер/гост позива систем** да учита **собу**. (АПСО)
4. **Систем приказује рецепционеру/госту** податке о **соби** и поруку: „**Систем** је учитао **собу**“ (ИА)



Алтернативна сценарија

- 2.1. Уколико **систем** не може да нађе **собе** он приказује **рецепционеру/госту** поруку: „**Систем** не може да нађе **собе** по задатој вредности“. Прекида се извршење сценарија. (ИА)



4.1. Уколико **систем** не може да учита **собу** он приказује **рецепционеру/госту** поруку: „**Систем** не може да учита **собу**“. (ИА)

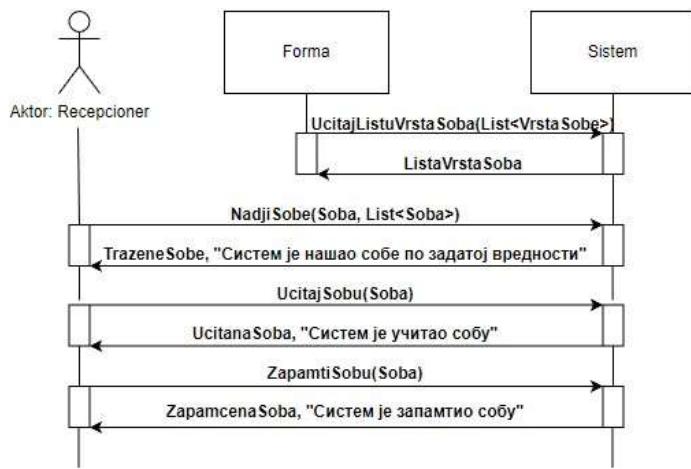


Са наведених секвенцијалних дијаграма уочавају се 2 системске операције:

1. *signal* NadjiSobe(Soba, List<Soba>);
2. *signal* UcitajSobu(Soba).

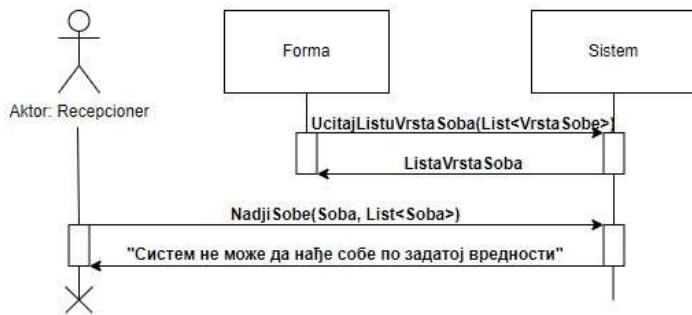
ДС7: Дијаграм секвенци случаја коришћења – Измена података собе

1. **Форма позива** **систем** да учита листу врста соба. (АПСО)
2. **Систем враћа** **форми** листу врста соба. (ИА)
3. **Рецепционер позива** **систем** да нађе **собе** по задатој вредности. (АПСО)
4. **Систем приказује** **рецепционеру** податке о **собама** и поруку: „**Систем** је нашао **собе** по задатој вредности“. (ИА)
5. **Рецепционер позива** **систем** да учита **собу**. (АПСО)
6. **Систем приказује** **рецепционеру** податке о **соби** и поруку: „**Систем** је учитао **собу**“. (ИА)
7. **Рецепционер позива** **систем** да запамти податке о **соби**. (АПСО)
8. **Систем приказује** **рецепционеру** изменјену **собу** и поруку: „**Систем** је запамтио **собу**“. (ИА)

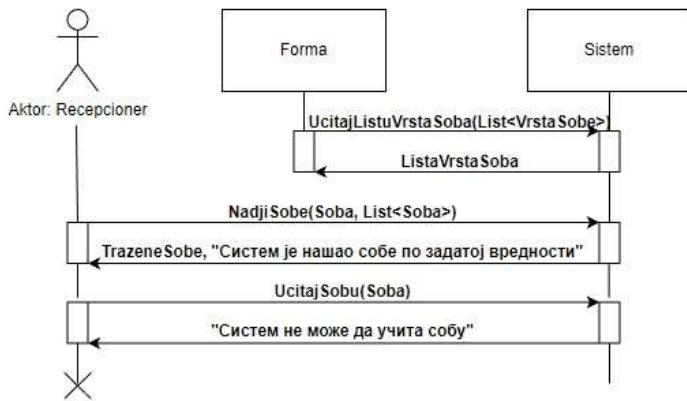


Алтернативна сценарија

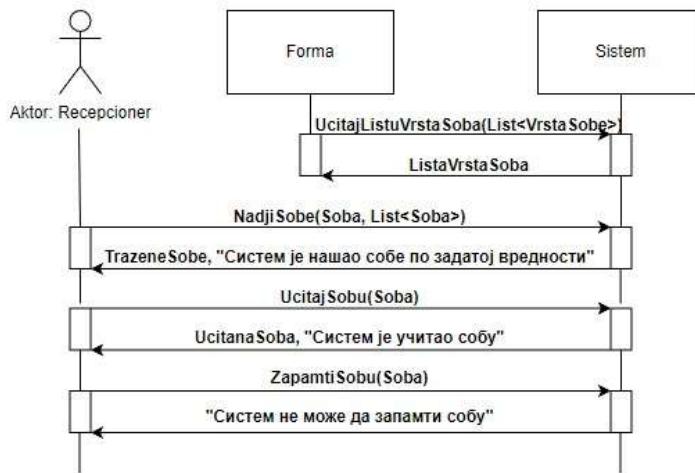
4.1. Уколико **систем** не може да нађе **собе** он приказује **рецепционеру** поруку: „**Систем** не може да нађе **собе** по задатој вредности“. Прекида се извршење сценарија. (ИА)



6.1. Уколико **систем** не може да учита **собу** он приказује **рецепционеру** поруку: „**Систем** не може да учита **собу**“. Прекида се извршење сценарија. (ИА)



8.1. Уколико **систем** не може да запамти податке о **соби** он приказује **рецепционеру** поруку: „**Систем** не може да запамти **собу**“. (ИА)

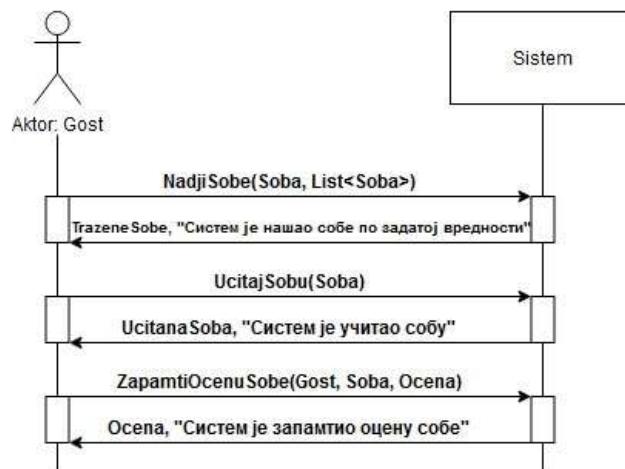


Са наведених секвенцних дијаграма уочавају се 4 системске операције:

1. *signal* / UcitajListuVrstaSoba(List<VrstaSobe>);
2. *signal* NadjiSobe(Soba, List<Soba>);
3. *signal* UcitajSobu(Soba);
4. *signal* ZapamtiSobu(Soba).

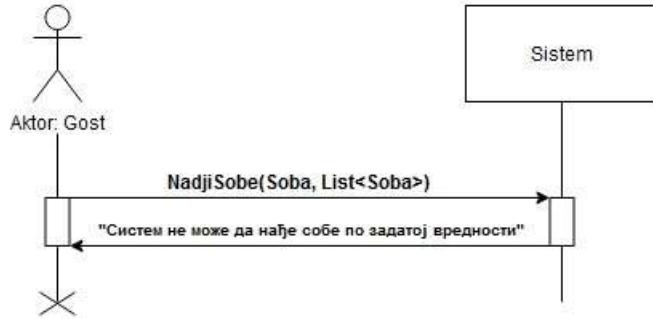
ДС8: Дијаграм секвенци случаја коришћења – Оцена собе

1. Гост позива систем да нађе собе по задатој вредности. (АПСО)
2. Систем приказује госту податке о собама и поруку : „Систем је нашао собе по задатој вредности“. (ИА)
3. Гост позива систем да учита собу. (АПСО)
4. Систем приказује госту податке о соби и поруку: „Систем је учитао собу“. (ИА)
5. Гост позива систем да запамти оцену собе. (АПСО)
6. Систем приказује госту запамћену оцену собе и поруку: „Систем је запамтио оцену собе“. (ИА)

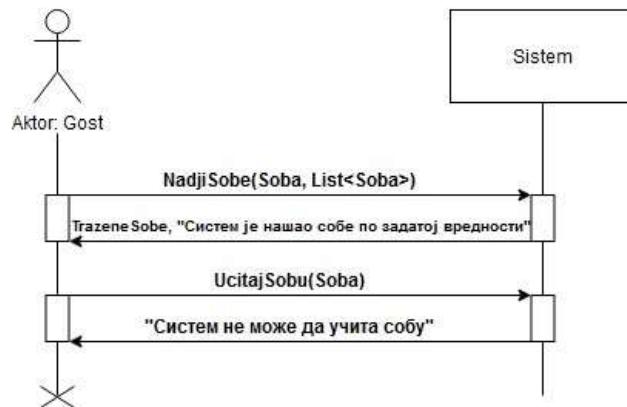


Алтернативна сценарија

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



4.1. Уколико **систем** не може да учита **собе** он приказује **госту** поруку: „**Систем** не може да учита **собу**“. Прекида се извршење сценарија. (ИА)



6.1. Уколико **систем** не може да запамти оцену **собе** он приказује **госту** поруку: „**Систем** не може да запамти оцену **собе**“. (ИА)

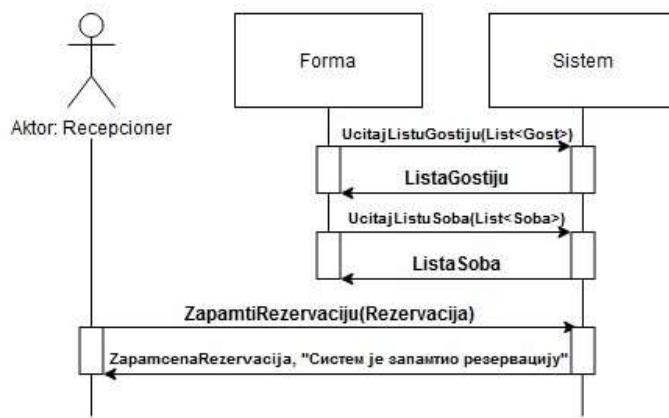


Са наведених секвенцних дијаграма уочавају се 3 системске операције:

1. *signal NadjiSobe(Soba, List<Soba>);*
2. *signal UcitajSobu(Soba);*
3. *signal ZapamtiOcenuSobe(Gost, Soba, Ocena).*

ДС9: Дијаграм секвенци случаја коришћења – Креирање резервације

1. **Форма** **позива** **систем** да учита листу гостију. (АПСО)
2. **Систем** **враћа** **форми** листу гостију. (ИА)
3. **Форма** **позива** **систем** да учита листу соба. (АПСО)
4. **Систем** **враћа** **форми** листу соба. (ИА)
5. **Рецепционер** **позива** **систем** да запамти податке о резервацији. (АПСО)
6. **Систем** **приказује** **рецепционеру** запамћену резервацију и поруку: „**Систем** је запамтио резервацију“. (ИА)



Алтернативна сценарија

- 6.1. Уколико **систем** не може да запамти податке о **резервацији** он приказује **рецепционеру** поруку: „**Систем** не може да запамти **резервацију**“. (ИА)

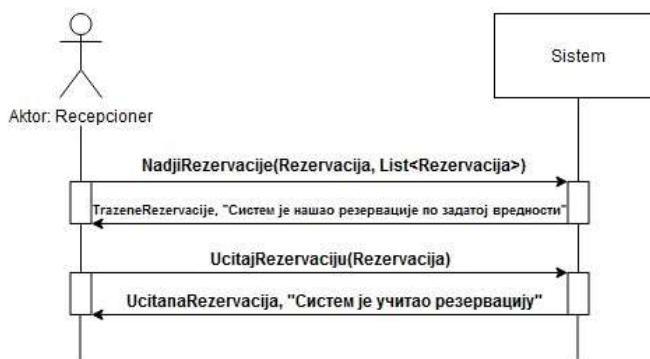


Са наведених секвенцних дијаграма уочавају се 3 системске операције:

1. *signal* UcitajListuGostiju(List<Gost>);
2. *signal* UcitajListuSoba(List<Soba>);
3. *signal* ZapamtiRezervaciju(Rezervacija).

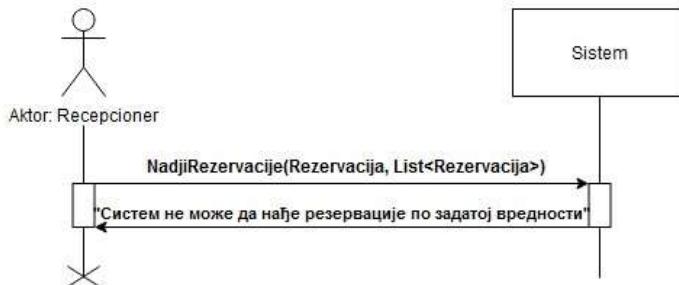
ДС10: Дијаграм секвенци случаја коришћења – Претраживање резервације

1. **Рецепционер** **позива** **систем** да нађе резервације по задатој вредности. (АПСО)
2. **Систем** **приказује** **рецепционеру** податке о резервацијама и поруку: „**Систем** је нашао резервације по задатој вредности”. (ИА)
3. **Рецепционер** **позива** **систем** да учита резервацију. (АПСО)
4. **Систем** **приказује** **рецепционеру** податке о резервацији и поруку: „**Систем** је учитао резервацију“. (ИА)

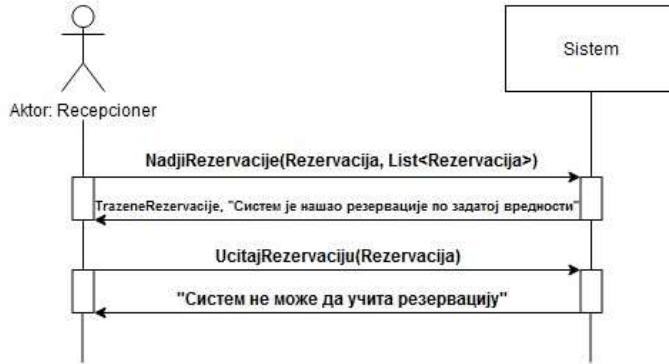


Алтернативна сценарија

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



- 4.1. Уколико **систем** не може да учита **резервацију** он приказује **рецепционеру** поруку: „**Систем** не може да учита **резервацију**“. (ИА)



Са наведених секвенцних дијаграма уочавају се 2 системске операције:

1. *signal* NadjiRezervacije(Rezervacija, List<Rezervacija>);
2. *signal* UcitajRezervaciju(Rezervacija).

На основу анализе сценарија добијено је 15 системских операција:

1. *signal* UcitajListuGradova(List<Grad>);
2. *signal* ZapamtiGosta(Gost);
3. *signal* NadjiGoste(Gost, List<Gost>);
4. *signal* UcitajGosta(Gost);
5. *signal* ObrisGosta(Gost);
6. *signal* UcitajListuVrstaSoba(List<VrstaSobe>);
7. *signal* ZapamtiSobu(Soba);
8. *signal* NadjiSobe(Soba, List<Soba>);
9. *signal* UcitajSobu(Soba);
10. *signal* ZapamtiOcenuSobe(Gost, Soba, Ocena);
11. *signal* UcitajListuGostiju(List<Gost>);
12. *signal* UcitajListuSoba(List<Soba>);
13. *signal* ZapamtiRezervaciju(Rezervacija);
14. *signal* NadjiRezervacije(Rezervacija, List<Rezervacija>);
15. *signal* UcitajRezervaciju(Rezervacija).

2.2. Понашање софтверског система – Дефинисање уговора о системским операцијама

Уговор УГ1: UcitajListuGradova(List<Grad>) signal;

Веза са СК: СК1, СК3

Предуслови:

Постуслови:

Уговор УГ2: ZapamtiGosta(Gost) signal;

Веза са СК: СК1, СК3

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

Постуслови: Подаци о госту су запамћени.

Уговор УГ3: NadjiGoste(Gost, List<Gost>) signal;

Веза са СК: СК2, СК3, СК4

Предуслови:

Постуслови:

Уговор УГ4: UcitajGosta(Gost) signal;

Веза са СК: СК2, СК3, СК4

Предуслови:

Постуслови:

Уговор УГ5: ObrisGosta(Gost) signal;

Веза са СК: СК4

Предуслови: Структурна ограничења над објектом **Gost** морају бити задовољена.

Постуслови: Гост је обрисан.

Уговор УГ6: UcitajListuVrstaSoba(List<VrstaSobe>) signal;

Веза са СК: СК5, СК7

Предуслови:

Постуслови:

Уговор УГ7: ZapamtiSobu(Soba) signal;

Веза са СК: СК5, СК7

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

Постуслови: Подаци о соби су запамћени.

Уговор УГ8: NadjiSobe(Soba, List<Soba>) signal;

Веза са СК: СК6, СК7, СК8

Предуслови:

Постуслови:

Уговор УГ9: UcitajSobu(Soba) signal;

Веза са СК: СК6, СК7, СК8

Предуслови:

Постуслови:

Уговор УГ10: ZapamtiOcenuSobe(Gost, Soba, Ocena) signal;

Веза са СК: СК8

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

Постуслови: Подаци о оцени су запамћени.

Уговор УГ11: UcitajListuGostiju(List<Gost>) signal;

Веза са СК: СК9

Предуслови:

Постуслови:

Уговор УГ12: UcitajListuSoba(List<Soba>) signal;

Веза са СК: СК9

Предуслови:

Постуслови:

Уговор УГ13: ZapamtiRezervaciju(Rezervacija) signal;

Веза са СК: СК9

Предуслови: Вредносна и структурна ограничења над објектима **Rezervacija** и **StavkaRezervacije** морају бити задовољена.

Постуслови: Подаци о резервацији су запамћени.

Уговор УГ14: NadjiRezervacije(Rezervacija, List<Rezervacija>) signal;

Веза са СК: СК10

Предуслови:

Постуслови:

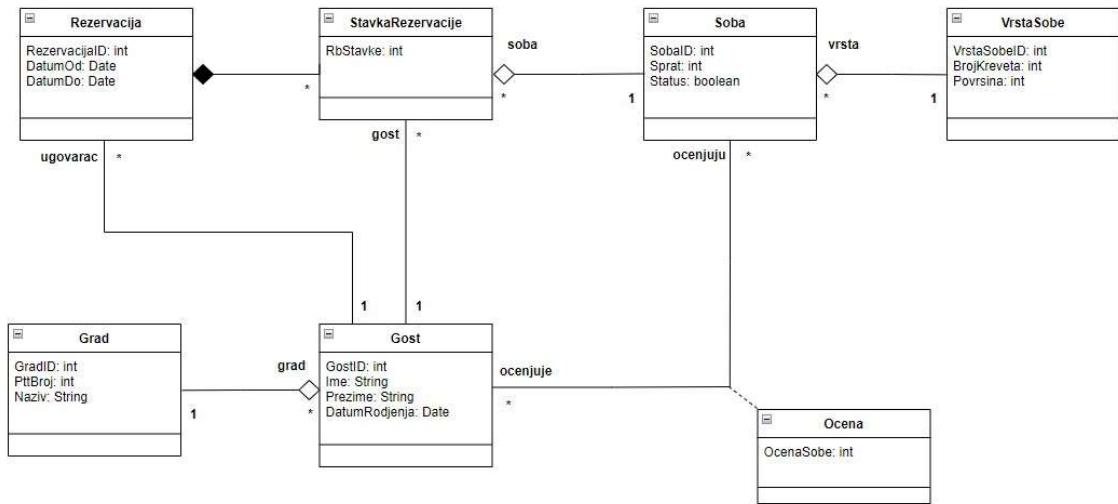
Уговор УГ15: UcitajRezervaciju(Rezervacija) signal;

Веза са СК: СК10

Предуслови:

Постуслови:

2.3. Структура софтверског система – Концептуални(доменски) модел



2.4. Структура софтверског система – релациони модел

Grad(GradID, PttBroj, Naziv)

VrstaSobe(VrstaSobelID, BrojKreveta, Povrsina)

Gost(GostID, Ime, Prezime, DatumRodjenja, GradID)

Soba(SobalID, Sprat, Status, VrstaSobelID)

Rezervacija(RezervacijalID, DatumOd, DatumDo, GostID)

StavkaRezervacije(RezervacijalID, RbStavke, GostID, SobalID)

Ocena(GostID, SobalID, OcenaSobe)

Tabela Grad		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 Gost DELETE RESTRICTED Gost
	GradID	Integer	not null and >0			
	PttBroj	Integer	not null and >0			
	Naziv	String	not null			

Tabela VrstaSobe		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 Soba DELETE RESTRICTED Soba
	VrstaSobelID	Integer	not null and >0			
	BrojKreveta	Integer	not null and >0			
	Povrsina	Integer	not null and >0			

Tabela Soba		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	INSERT RESTRICTED VrstaSobe UPDATE RESTRICTED VrstaSobe UPDATE CASCADES Rezervacija, Ocena DELETE RESTRICTED Rezervacija, Ocena
	SobelID	Integer	not null and >0			
	Sprat	Integer	not null and >0			
	Status	Boolean	not null (default: false)			
	VrstaSobelID	Integer	not null and >0			

Tabela Gost		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	INSERT RESTRICTED Grad UPDATE RESTRICTED Grad UPDATE CASCADES Rezervacija, Ocena DELETE RESTRICTED Rezervacija, Ocena
	GostID	Integer	not null and >0			
	Ime	String	not null			
	Prezime	String	not null			
	DatumRodjenja	Date	not null			
	GradID	Integer	not null and >0			

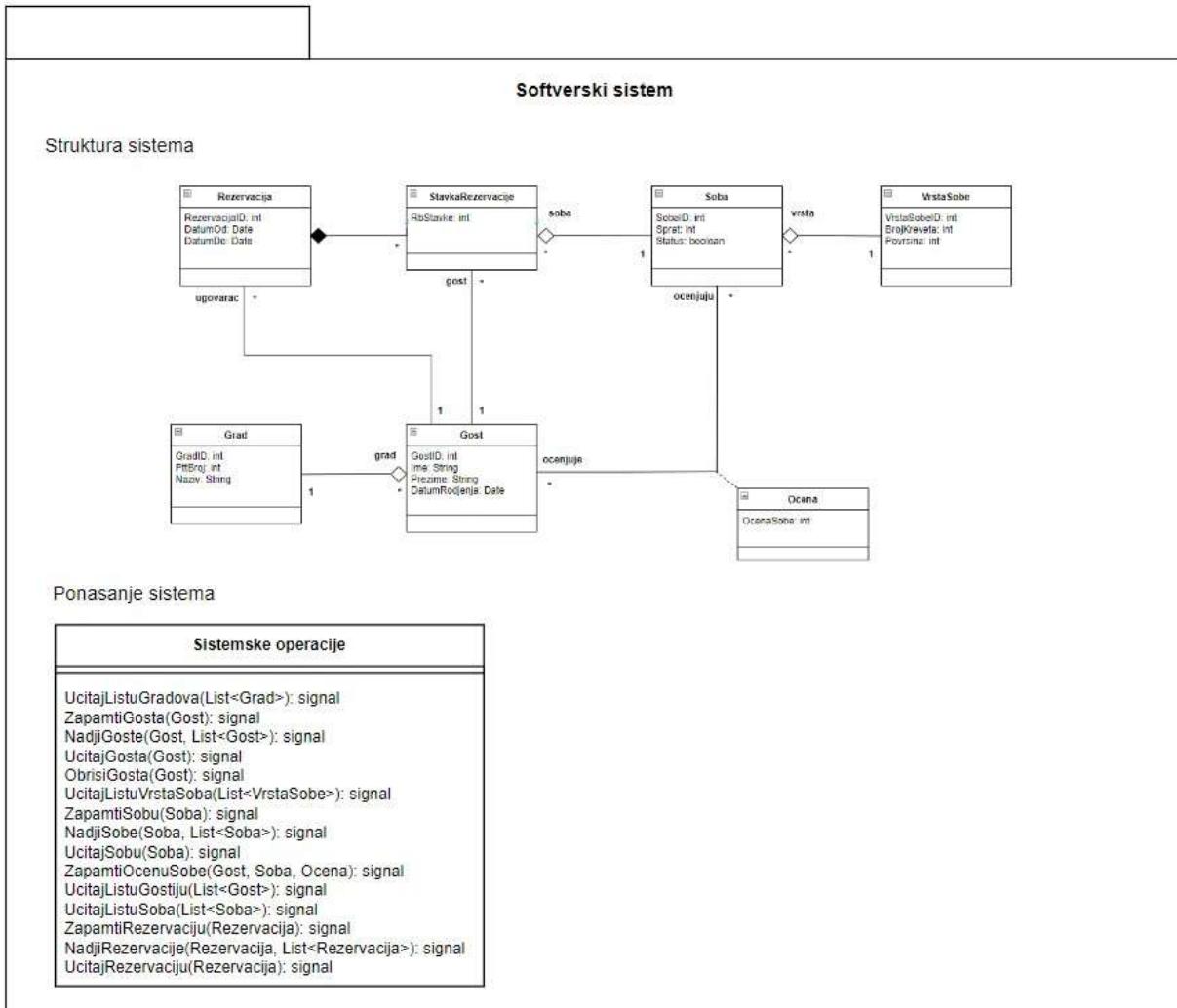
Tabela Rezervacija		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	INSERT RESTRICTED Gost UPDATE RESTRICTED Gost UPDATE CASCADES StavkaRezervacije DELETE RESTRICTED StavkaRezervacije
	RezervacijaID	Integer	not null and >0			
	DatumOd	Date	not null			
	DatumDo	Date	not null			
	GostID	Integer	not null and >0			

Tabela Ocena		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	INSERT RESTRICTED Gost, Soba UPDATE RESTRICTED Gost, Soba DELETE /
	GostID	Integer	not null and >0			
	SobaID	Integer	not null and >0			

	OcenaSobe	Integer	>0 and <6 (default:0)			
--	-----------	---------	--------------------------	--	--	--

Tabela StavkaRezervacije		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	INSERT RESTRICTED Rezervacija, Gost, Soba UPDATE RESTRICTED Rezervacija, Gost, Soba DELETE /
	RezervacijaID	Integer	not null and >0			
	RbStavke	Integer	not null and >0			
	GostID	Integer	not null and >0			
	SobaID	Integer	not null and >0			

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



3. Пројектовање

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

Архитектура софтверског система

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

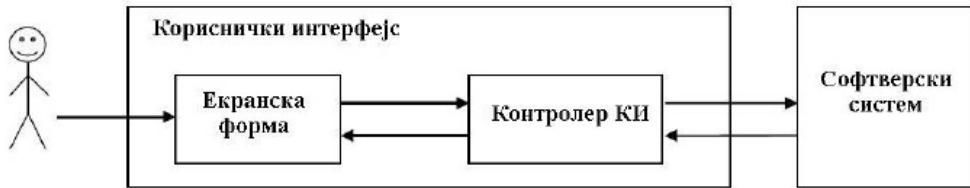
- Кориснички интерфејс
- Апликациона логика
- Складиште података

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



3.1. Пројектовање корисничког интерфејса

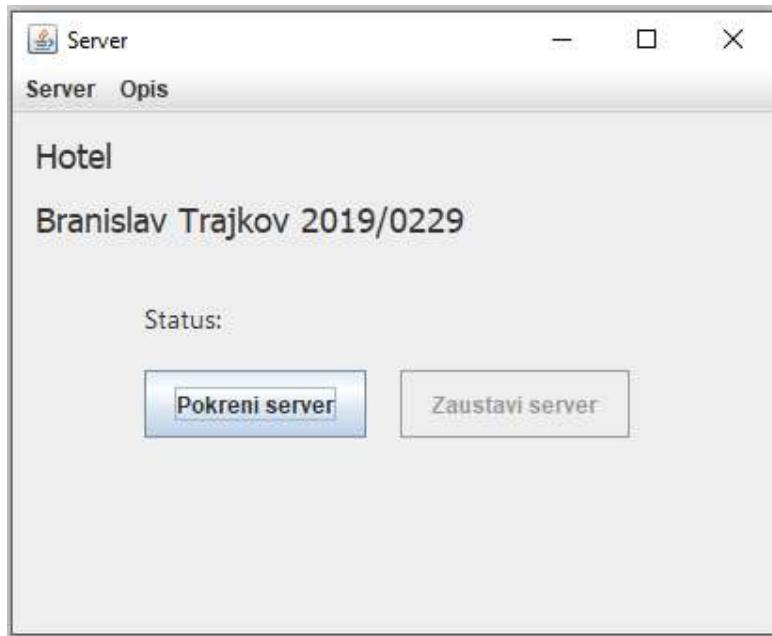
Кориснички интерфејс представља реализацију улаза и/или излаза софтверског система и његову структуру чине екранска форма и контролер корисничког интерфејса.



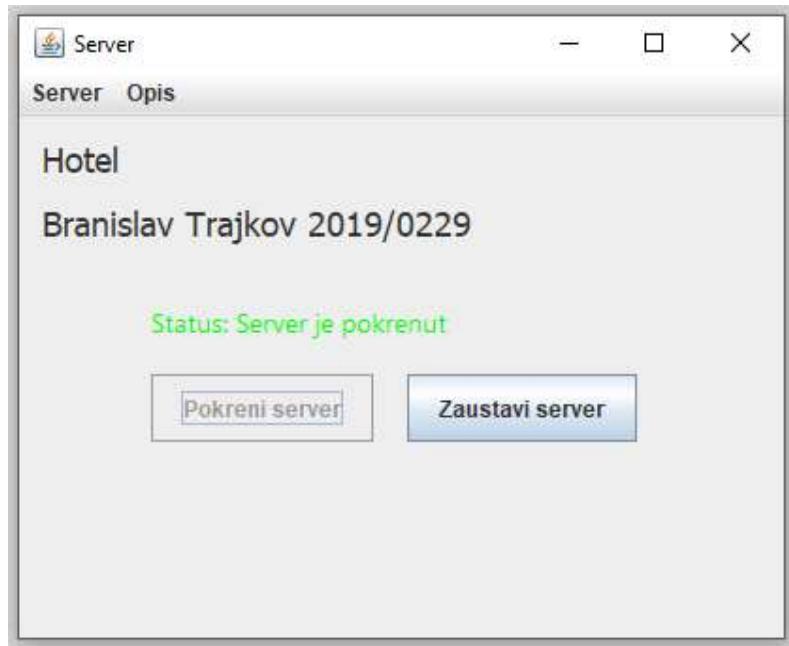
3.1.1. Пројектовање екранских форми

Кориснички интерфејс је дефинисан преко скупа екранских форми. Сценарија коришћења екранских форми је директно повезан са сценаријима случајева коришћења.

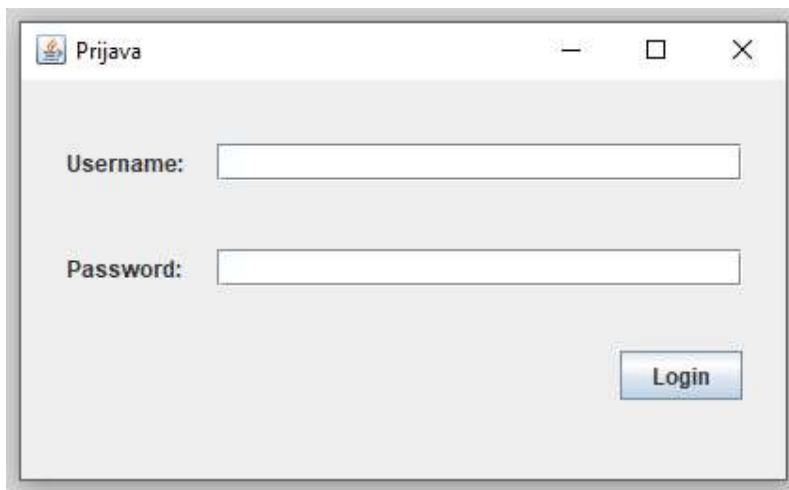
На серверској страни програма пројектована је корисничка форма која изгледа пре активације овако:



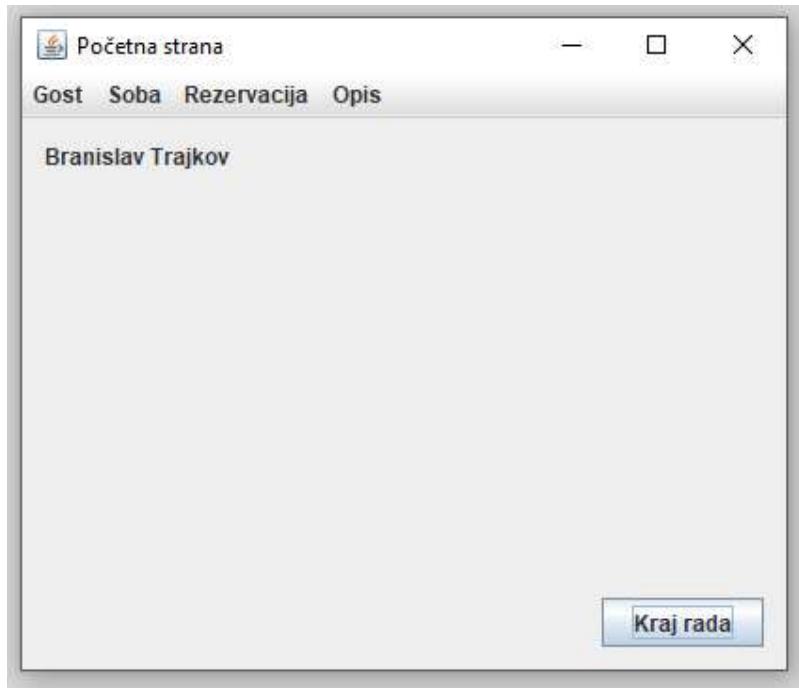
Након активације, серверска форма изгледа овако:



На клијентској страни прво је потребно улоговати се како би се доспело у могућност коришћења апликације. Форма за логовање изгледа овако:



Након логовања, систем приказује клијенту главну екранску форму из које се може доћи до осталих екранских форми.



СК1: Случај коришћења – Креирање госта

Назив СК

Креирање госта

Актори СК

Рецепционер

Учесници СК

Рецепционер и систем (програм)

Предуслов: Систем је укључен и **рецепционер** је пријављен под својом шифром. Систем приказује форму за рад са **гостом**. Учитана је листа градова.

The screenshot shows a window titled "Gost". Inside, there are five input fields: "ID:" (empty), "Ime:" (empty), "Prezime:" (empty), "Datum rođenja (dd.MM.yyyy):" (empty), and "Grad:" with a dropdown menu showing "Beograd". Below the fields are five buttons: "Obriši", "Izmeni", "Otkaži", "Omogući izmene", and "Sačuvaj".

Основни сценарио СК

1. Рецепционер уноси податке о госту. (АПУСО)

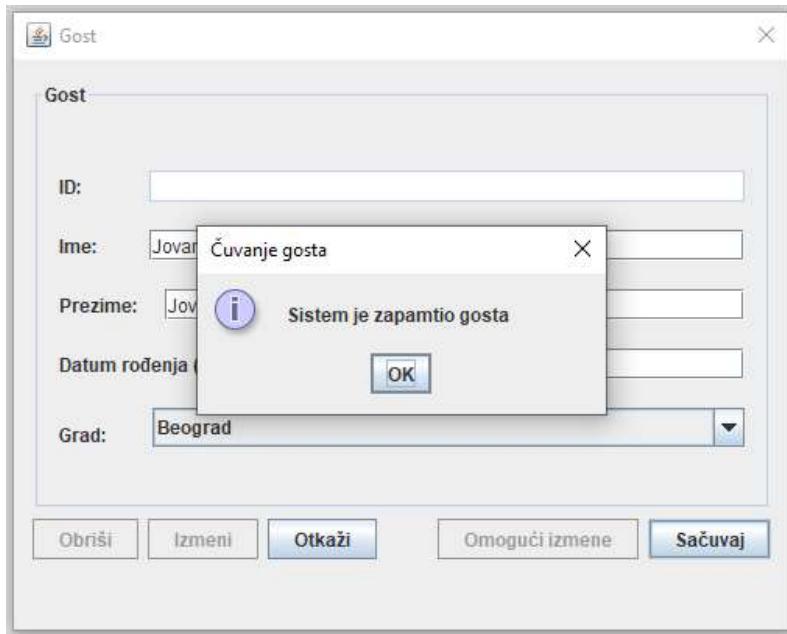
2. **Рецепционер контролише** да ли је коректно унео податке о **госту**. (АНСО)

3. **Рецепционер позива** **систем** да запамти податке о **госту**. (АПСО)

Опис акције: Рецепционер кликом на дугме „Сачувај“ позива системску операцију **ZapamtiGosta(Gost)**

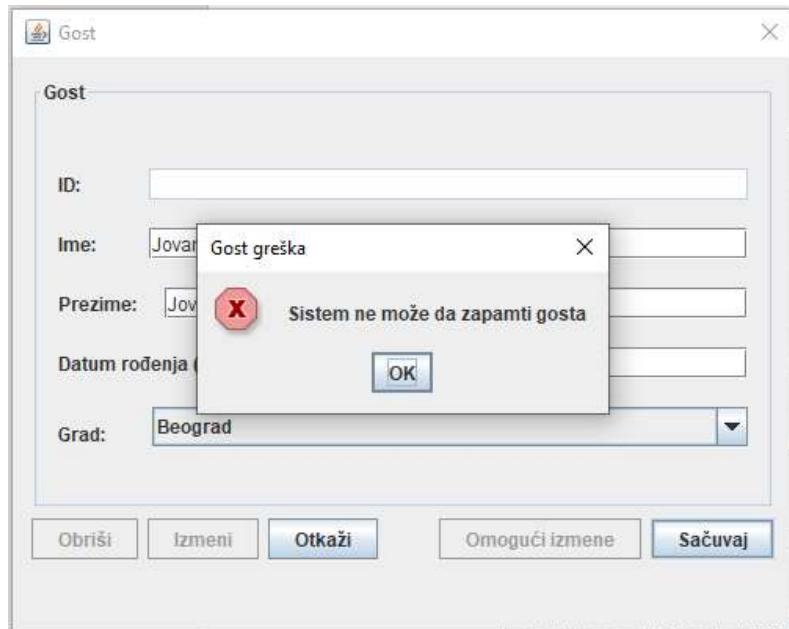
4. **Систем памти** податке о **госту**. (СО)

5. **Систем приказује** **рецепционеру** запамћеног **госта** и поруку: „**Систем је запамтио госта**“. (ИА)



Алтернативна сценарија

5.1 Уколико **систем** не може да запамти податке о **госту** он приказује **рецепционеру** поруку: „**Систем не може да запамти госта**“. (ИА)



СК2: Случај коришћења – Претраживање госта

Назив СК

Претраживање [госта](#)

Актори СК

[Рецепционер](#)

Учесници СК

[Рецепционер](#) и [систем](#) (програм)

Предуслов: [Систем](#) је укључен и [рецепционер](#) је пријављен под својом шифром. Систем приказује форму за рад са [гостом](#).

Pretraga gostiju

ID	Ime	Prezime	Datum rodjenja	Grad
2	Ivana	Ivanovic	02.09.1972	Zajecar
6	Ivana	Trajkov	02.09.1972	Zajecar
9	Jovan	Jovanovic	05.06.2000	Beograd
5	Luka	Lukic	05.05.2005	Vranje

Основни сценарио СК

- Рецепционер уноси вредност по којој претражује госте. (АПУСО)

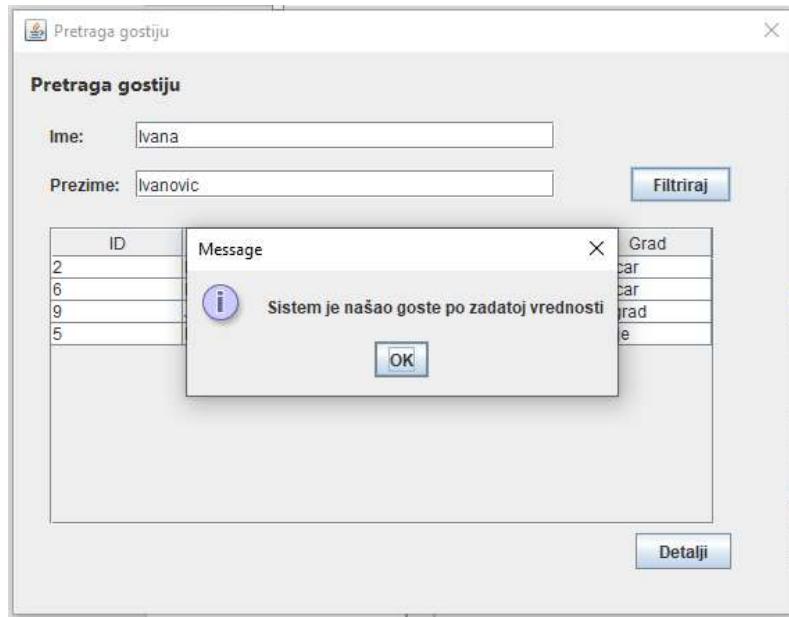
Pretraga gostiju

ID	Ime	Prezime	Datum rodjenja	Grad
2	Ivana	Ivanovic	02.09.1972	Zajecar
6	Ivana	Trajkov	02.09.1972	Zajecar
10	Ivana	Ivanovic	05.05.1995	Beograd
9	Jovan	Jovanovic	05.06.2000	Beograd
5	Luka	Lukic	05.05.2005	Vranje

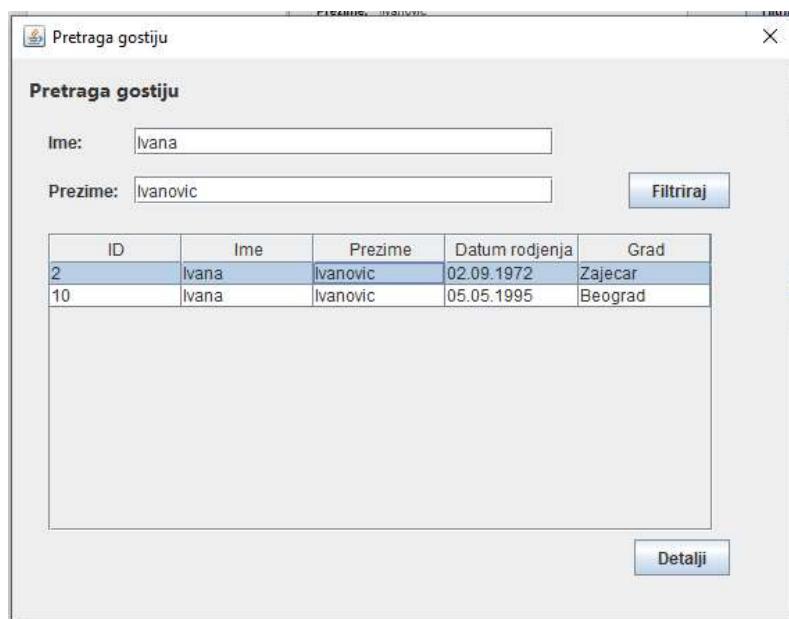
- Рецепционер позива систем да нађе госте по задатој вредности. (АПСО)

Опис акције: Рецепционер кликом на дугме „Филтрирај“ позива системску операцију NadjiGoste(Gost,List<Gost>)

- Систем тражи госте по задатој вредности. (СО)
- Систем приказује рецепционеру податке о гостима и поруку: “Систем је нашао госте по задатој вредности”. (ИА)



5. Рецепционер бира госта. (АПУСО)

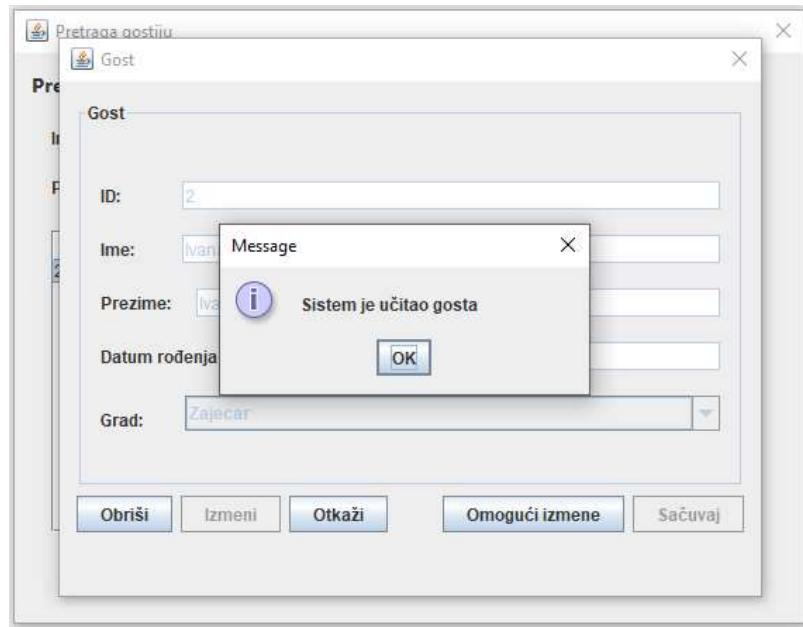


6. Рецепционер позива систем да учита госта. (АПСО)

Опис акције: Рецепционер кликом на дугме „Детаљи“ позива системску операцију **UcitajGosta(Gost)**

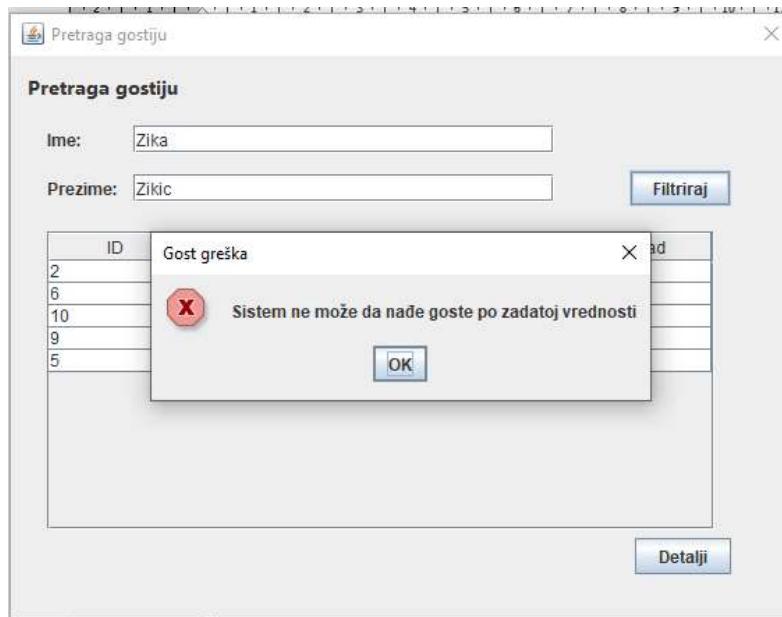
7. Систем учитава госта. (СО)

8. Систем приказује рецепционеру податке о госту и поруку: „Систем је учитао госта“. (ИА)

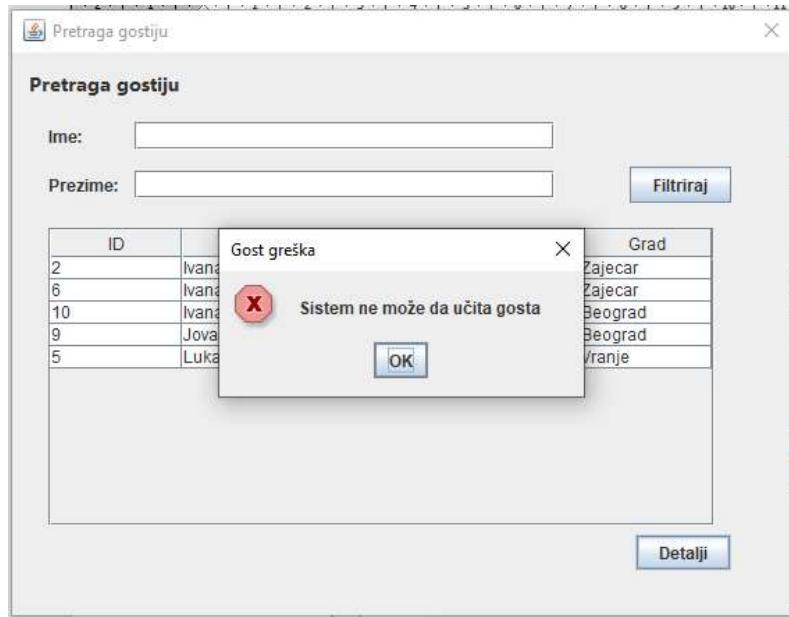


Алтернативна сценарија

4.1. Уколико систем не може да нађе госте он приказује рецепционеру поруку: „Систем не може да нађе госте по задатој вредности“. Прекида се извршење сценарија. (ИА)



8.1. Уколико систем не може да учита госта он приказује рецепционеру поруку: „Систем не може да учита госта“. (ИА)



СК3: Случај коришћења – Измена података госта

Назив СК

Измена података госта

Актори СК

Рецепционер

Учесници СК

Рецепционер и систем (програм)

Предуслов: Систем је укључен и **рецепционер** је пријављен под својом шифром. Систем приказује форму за рад са **гостом**. Учитана је листа градова.

Pretraga gostiju

ID	Ime	Prezime	Datum rođenja	Grad
2	Ivana	Ivanovic	02.09.1972	Zajecar
6	Ivana	Trajkov	02.09.1972	Zajecar
9	Jovan	Jovanovic	05.06.2000	Beograd
5	Luka	Lukic	05.05.2005	Vranje

Основни сценарио СК

- Рецепционер уноси вредност по којој претражује госте. (АПУСО)

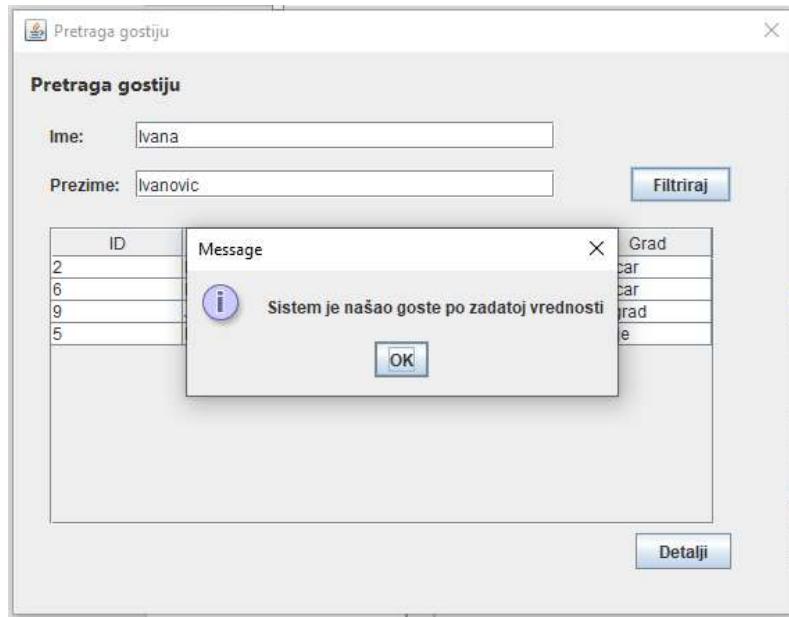
Pretraga gostiju

ID	Ime	Prezime	Datum rođenja	Grad
2	Ivana	Ivic	02.09.1972	Zajecar
6	Ivana	Trajkov	02.09.1972	Zajecar
10	Ivana	Ivanovic	05.05.1995	Beograd
9	Jovan	Jovanovic	05.06.2000	Beograd
5	Luka	Lukic	05.05.2005	Vranje

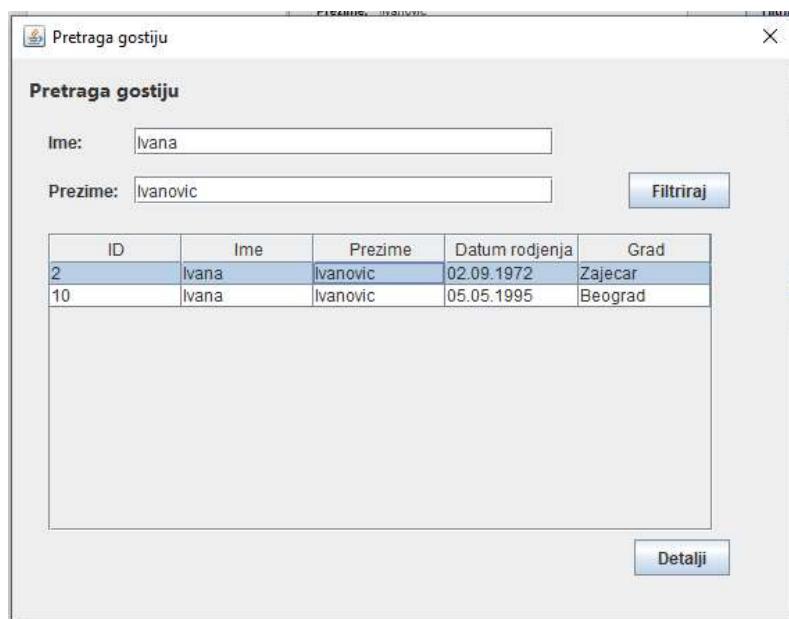
- Рецепционер позива систем да нађе госте по задатој вредности. (АПСО)

Опис акције: Рецепционер кликом на дугме „Филтрирај“ позива системску операцију **NadjiGoste(Gost,List<Gost>)**

- Систем тражи госте по задатој вредности. (СО)
- Систем приказује рецепционеру податке о гостима и поруку: “Систем је нашао госте по задатој вредности”. (ИА)



5. Рецепционер бира госта. (АПУСО)

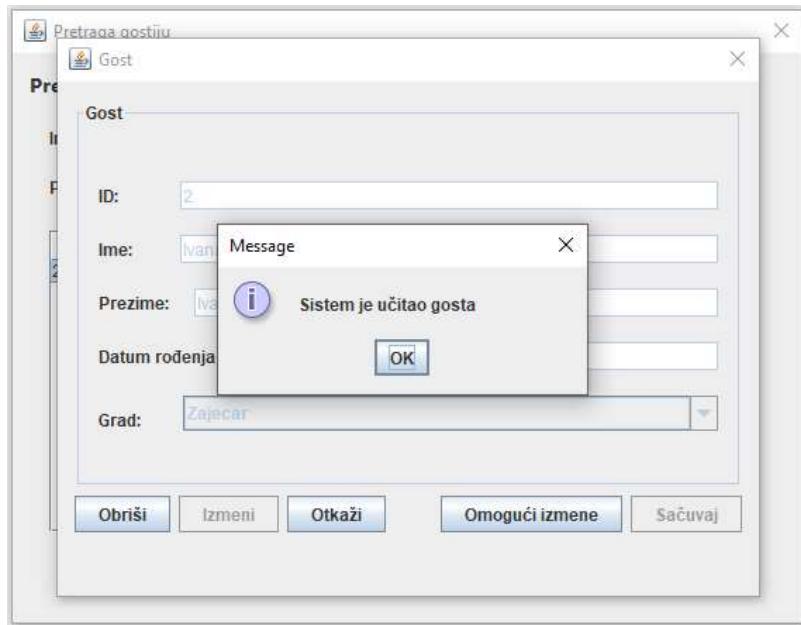


6. Рецепционер позива систем да учита госта. (АПСО)

Опис акције: Рецепционер кликом на дугме „Детаљи“ позива системску операцију **UcitajGosta(Gost)**

7. Систем учитава госта. (СО)

8. Систем приказује рецепционеру податке о госту и поруку: „Систем је учитао госта“. (ИА)



9. Рецепционер уноси (менја) податке о госту. (АПУСО)

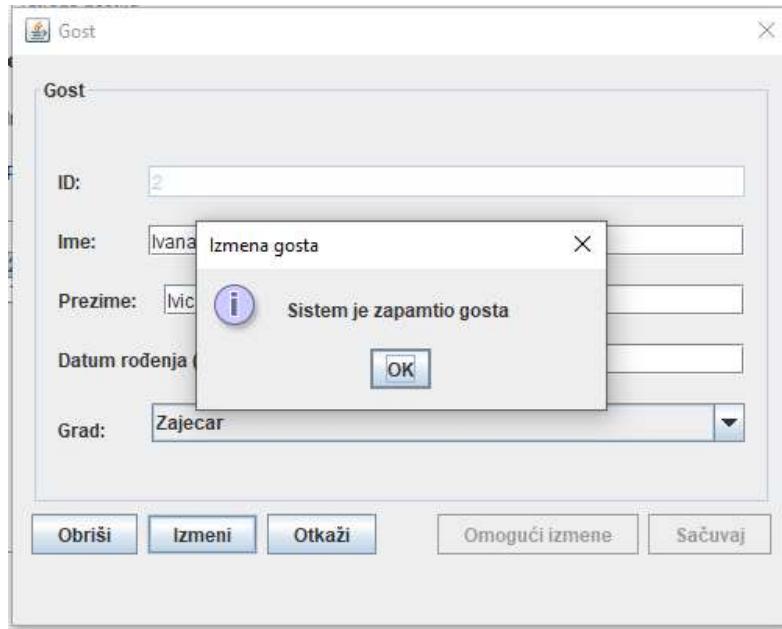
10. Рецепционер контролише да ли је исправно унео податке о госту. (АНСО)

11. Рецепционер позива систем да запамти податке о госту. (АПСО)

Опис акције: Рецепционер кликом на дугме „Измени“ позива системску операцију **ZapamtiGosta(Gost)**

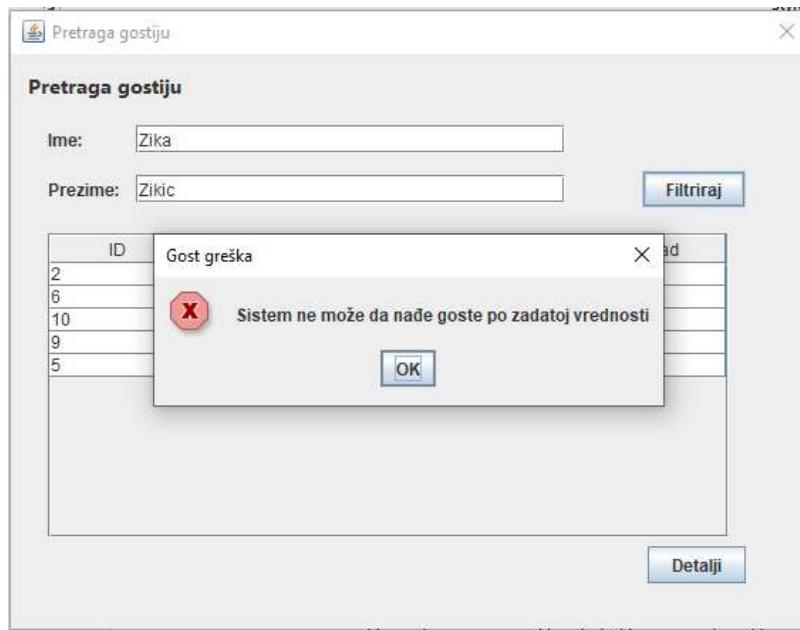
12. Систем памти податке о госту. (СО)

13. Систем приказује рецепционеру запамћеног госта и поруку: „Систем је запамтио госта“. (ИА)

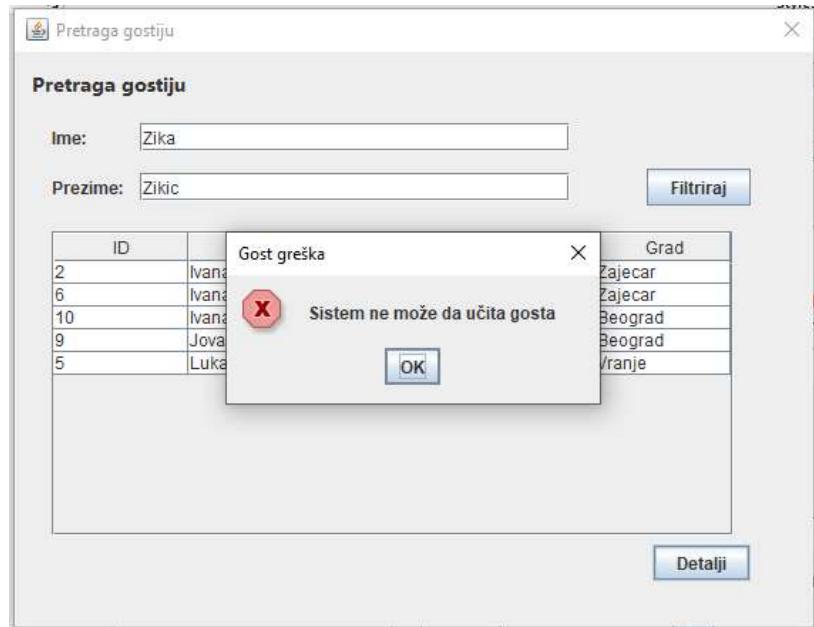


Алтернативна сценарија

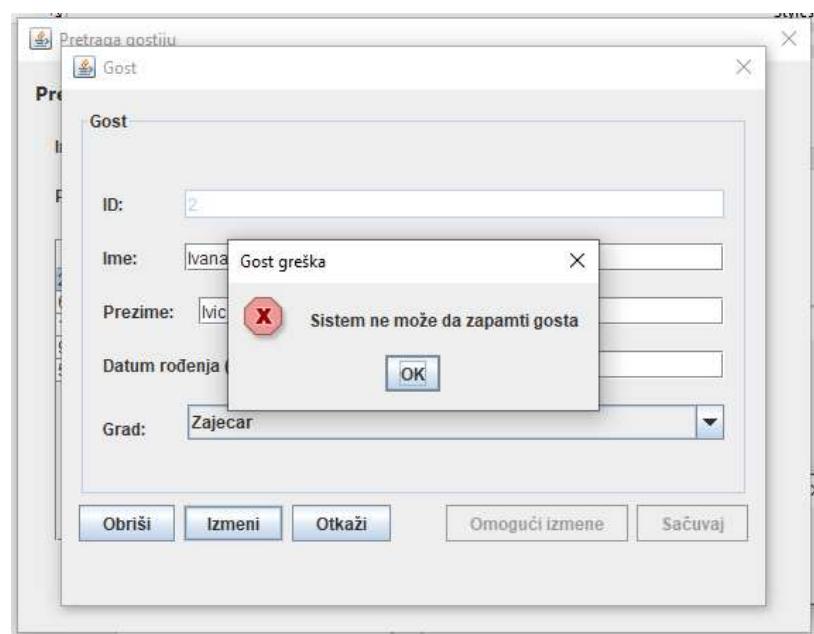
4.1. Уколико систем не може да нађе госте он приказује рецепционеру поруку: „Систем не може да нађе госте по задатој вредности“. Прекида се извршење сценарија. (ИА)



8.1. Уколико систем не може да учита госта он приказује рецепционеру поруку „Систем не може да учита госта“. Прекида се извршење сценарија. (ИА)



13.1. Уколико систем не може да запамти податке о госту он приказује рецепционеру поруку: „Систем не може да запамти госта“.



СК4: Случај коришћења – Брисање госта

Назив СК

Брисање госта

Актори СК

Рецепционер

Учесници СК

Рецепционер и систем (програм)

Предуслов: Систем је укључен и **рецепционер** је пријављен под својом шифром. Систем приказује форму за рад са **гостом**.

The screenshot shows a window titled "Pretraga gostiju". At the top, there are two input fields: "Ime:" and "Prezime:", both empty. To the right of these fields is a blue "Filtriraj" (Filter) button. Below the fields is a table with five columns: ID, Ime, Prezime, Datum rodjenja, and Grad. The table contains five rows of data:

ID	Ime	Prezime	Datum rodjenja	Grad
2	Ivana	Ivanovic	02.09.1972	Zajecar
6	Ivana	Trajkov	02.09.1972	Zajecar
9	Jovan	Jovanovic	05.06.2000	Beograd
5	Luka	Lukic	05.05.2005	Vranje

At the bottom right of the window is a blue "Detalji" (Details) button.

Основни сценарио СК

1. Рецепционер уноси вредност по којој претражује госте. (АПУСО)

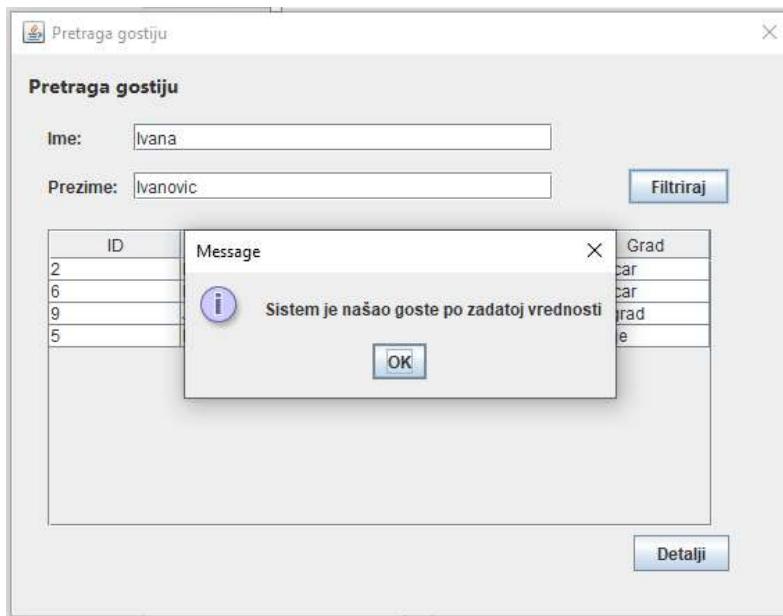
The screenshot shows the same "Pretraga gostiju" window as before, but with search criteria entered. The "Ime:" field contains "Ivana" and the "Prezime:" field contains "Ivanovic". The "Filtriraj" button is visible to the right of the surname field. Below the search fields is the same table of guest records. The table now shows only one result, which is the fourth row from the original list:

ID	Ime	Prezime	Datum rodjenja	Grad
9	Jovan	Jovanovic	05.06.2000	Beograd

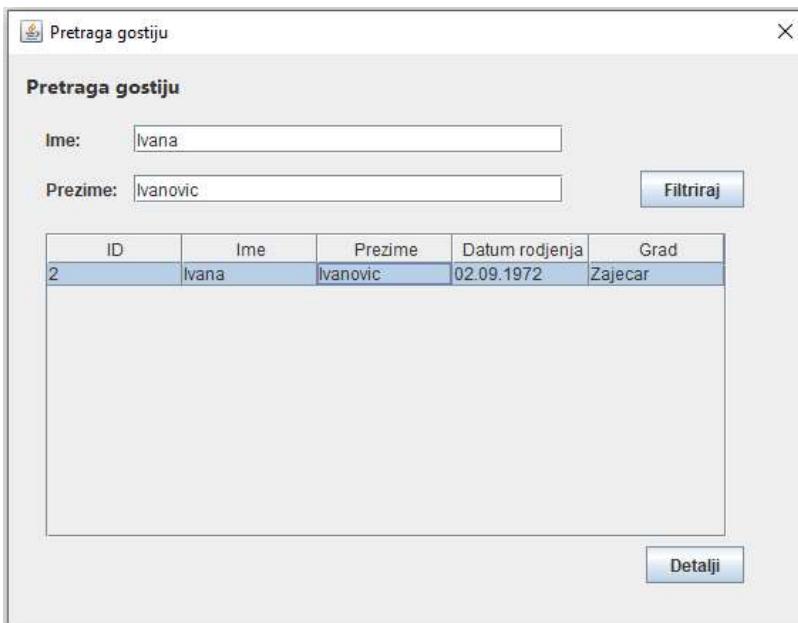
2. Рецепционер позива систем да нађе госте по задатој вредности. (АПСО)

Опис акције: Рецепционер кликом на дугме „Филтрирај“ позива системску операцију **NadjiGoste(Gost,List<Gost>)**

3. Систем тражи госте по задатој вредности. (СО)
 4. Систем приказује рецепционеру податке о гостима и поруку: “Систем је нашао госте по задатој вредности”. (ИА)



5. Рецепционер бира госта. (АПУСО)

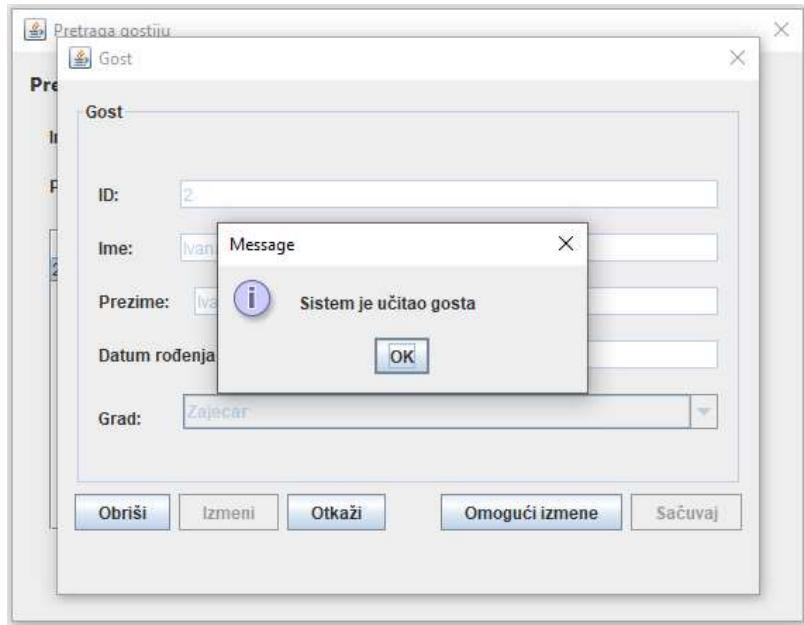


6. Рецепционер позива систем да учита госта. (АПСО)

Опис акције: Рецепционер кликом на дугме „Детаљи“ позива системску операцију **UcitajGosta(Gost)**

7. Систем учитава госта. (СО)

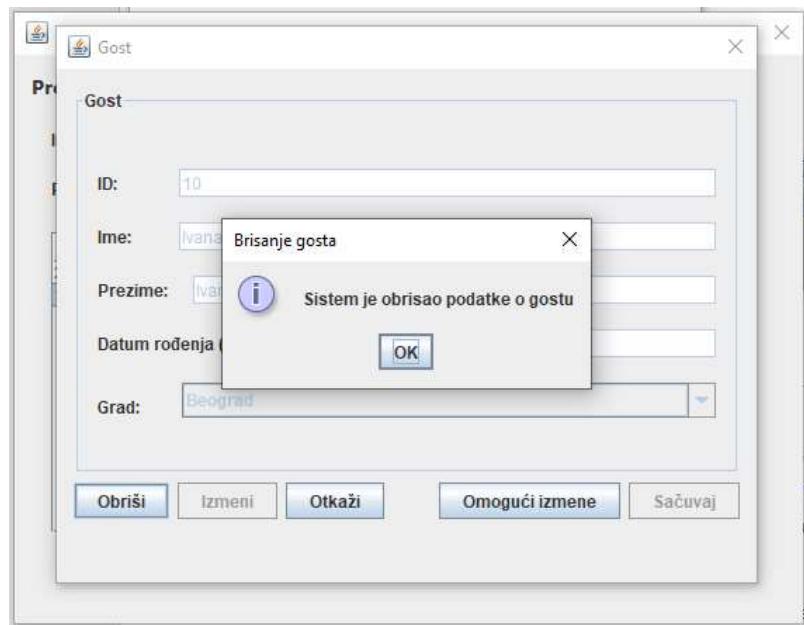
8. Систем приказује рецепционеру податке о госту и поруку: „Систем је учитао госта“. (ИА)



9. Рецепционер позива систем да обрише податке о госту. (АПСО)

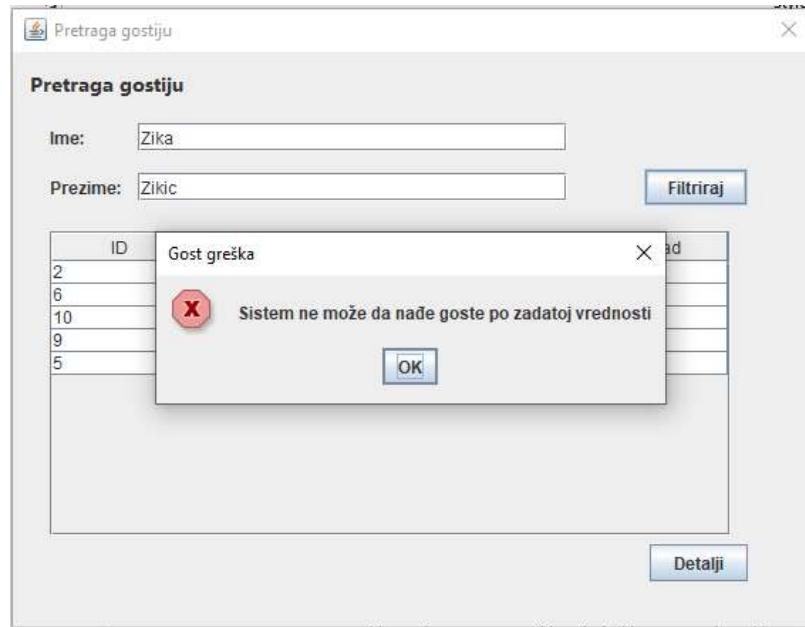
Опис акције: Рецепционер кликом на дугме „Обриши“ позива системску операцију **ObrisiGosta(Gost)**

10. Систем брише податке о госту. (СО)
11. Систем приказује рецепционеру поруку: „Систем је обрисао податке о госту.“ (ИА)

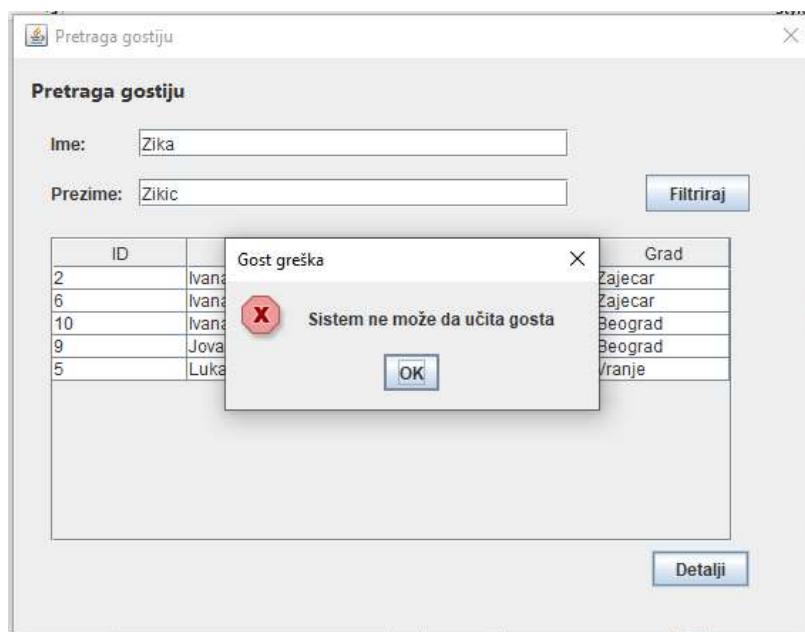


Алтернативна сценарија

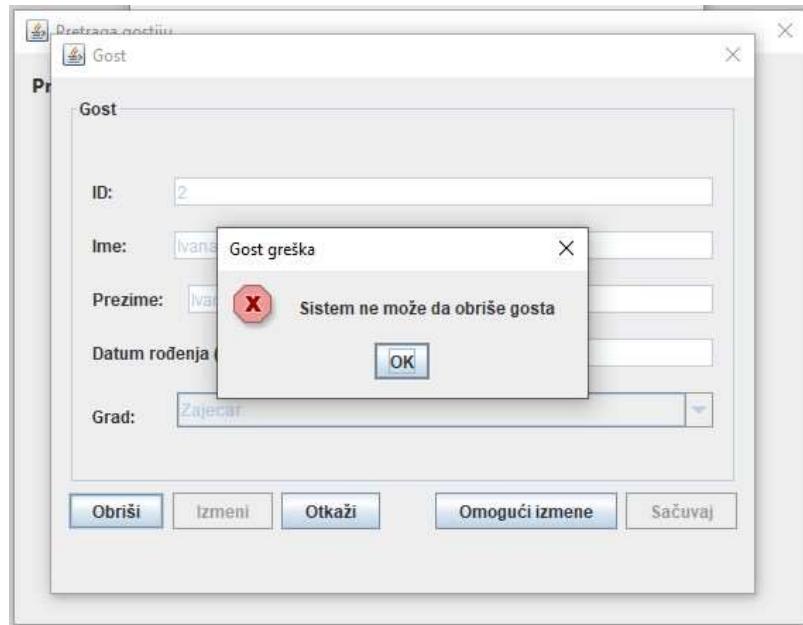
- 4.1. Уколико систем не може да нађе госте он приказује рецепционеру поруку: „Систем не може да нађе госте по задатој вредности“. Прекида се извршење сценарија. (ИА)



8.1. Уколико систем не може да учита госта он приказује рецепционеру поруку: „Систем не може да учита госта“. Прекида се извршење сценарија. (ИА)



11.1. Уколико систем не може да обрише податке о госту он приказује рецепционеру поруку: „Систем не може да обрише госта“. (ИА)



СК5: Случај коришћења – Креирање собе

Назив СК

Креирање собе

Актори СК

Рецепционер

Учесници СК

Рецепционер и систем (програм)

Предуслов: Систем је укључен и **рецепционер** је пријављен под својом шифром. Систем приказује форму за рад са **собом**. Учитана је листа врста соба.

Soba

Sprat:

Status: zauzeta

Vrsta sobe: Jednokrevetna

Oceni Izmeni Otkaži Omogući izmene Sačuvaj

Основни сценарио СК

1. Рецепционер уноси податке о соби. (АПУСО)

Soba

Sprat:

Status: zauzeta

Vrsta sobe: Trokrevetna

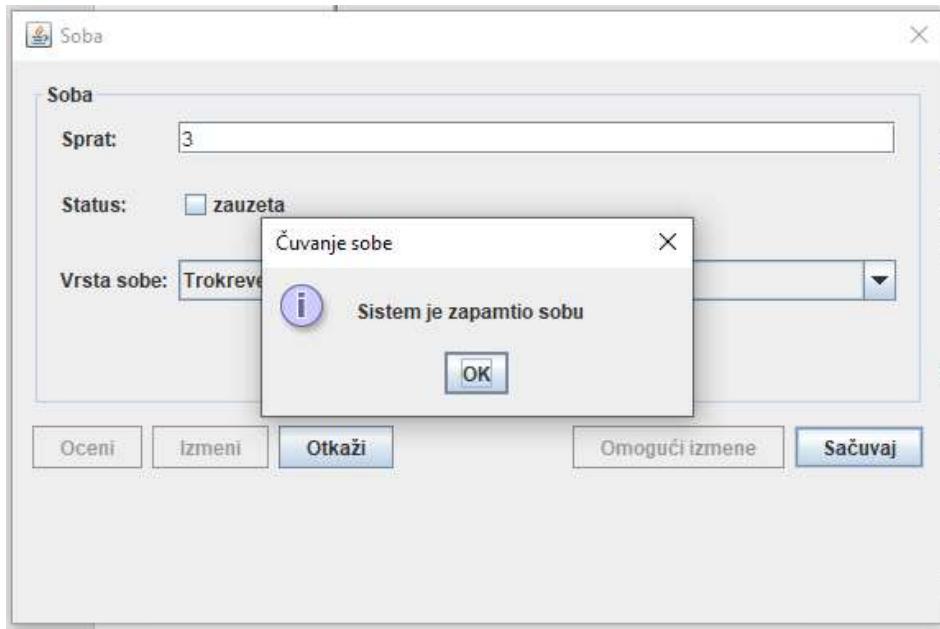
Oceni Izmeni Otkaži Omogući izmene Sačuvaj

2. Рецепционер контролише да ли је коректно унео податке о соби. (АНСО)

3. Рецепционер позива систем да запамти податке о соби. (АПСО)

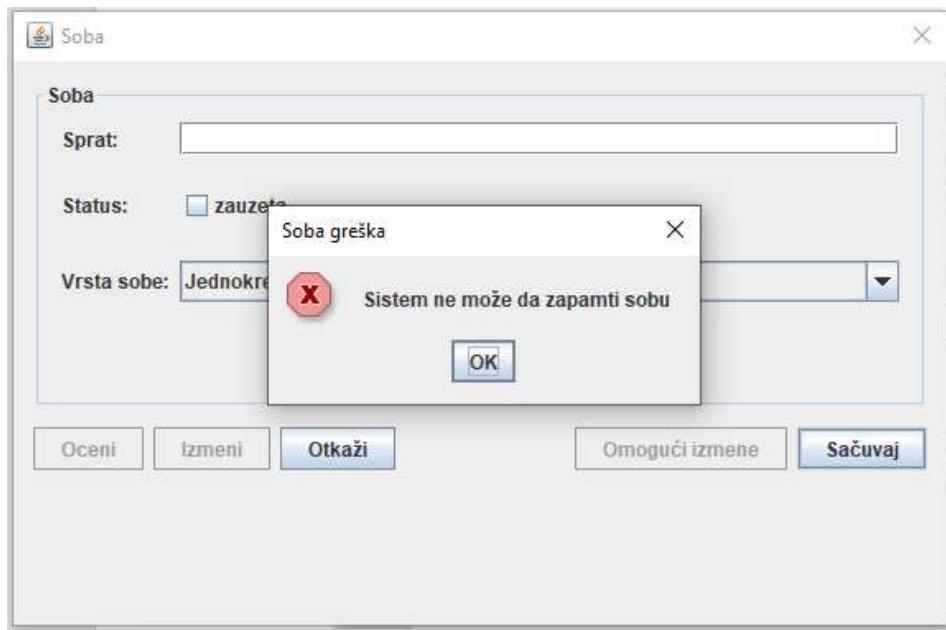
Опис акције: Рецепционер кликом на дугме „Сачувај“ позива системску операцију **ZapamtiSobu(Soba)**

4. Систем памти податке о соби. (СО)
5. Систем приказује рецепционеру запамћену собу и поруку: „Систем је запамтио собу“. (ИА)



Алтернативна сценарија

- 5.1. Уколико систем не може да запамти податке о соби он приказује **рецепционеру** поруку: „Систем не може да запамти собу“. (ИА)



СК6: Случај коришћења – Претраживање собе

Назив СК

Претраживање собе

Актори СК

Рецепционер/гост

Учесници СК

Рецепционер/гост и систем (програм)

Предуслов: Систем је укључен и **рецепционер/гост** је пријављен под својом шифром. Систем приказује форму за рад са **собом**.

The screenshot shows a window titled 'Pretraga soba'. At the top, there is a search bar labeled 'Sprat:' with a placeholder and a 'Filtriraj' button. Below the search bar is a table with the following data:

ID	Sprat	Status	Vrsta sobe
1	1	Zauzeta	Jednokrevetna
7	1	Slobodna	Jednokrevetna
6	1	Slobodna	Jednokrevetna
4	1	Zauzeta	Jednokrevetna
3	1	Slobodna	Jednokrevetna
5	2	Slobodna	Dvokrevetna
2	2	Slobodna	Trokrevetna
8	2	Slobodna	Jednokrevetna
10	3	Slobodna	Trokrevetna
9	4	Slobodna	Dvokrevetna

At the bottom right of the table area is a 'Detalji' button.

Основни сценарио СК

1. Рецепционер/гост уноси вредност по којој претражује собе. (АПУСО)

Pretraga soba

Sprat:	2	Filtriraj	
1	1	Zauzeta	Jednokrevetna
7	1	Slobodna	Jednokrevetna
6	1	Slobodna	Jednokrevetna
4	1	Zauzeta	Jednokrevetna
3	1	Slobodna	Jednokrevetna
5	2	Slobodna	Dvokrevetna
2	2	Slobodna	Trokrevetna
8	2	Slobodna	Jednokrevetna
10	3	Slobodna	Trokrevetna
9	4	Slobodna	Dvokrevetna

Detalji

2. Рецепционер/гост **позива** систем да нађе **собе** по задатој вредности. (АПСО)

Опис акције: Рецепционер/гост кликом на дугме „Филтрирај“ позива системску операцију **NadjiSobe(Soba,List<Soba>)**

3. Систем тражи **собе** по задатој вредности. (СО)
4. Систем **приказује** рецепционеру/госту податке о **собама** и поруку: „Систем је нашао **собе** по задатој вредности“. (ИА)

Pretraga soba

Sprat:	2	Filtriraj	
1	1	Zauzeta	Jednokrevetna
7	1	Slobodna	Jednokrevetna
6	1	Slobodna	Jednokrevetna
4	1	Zauzeta	Jednokrevetna
3	1	Slobodna	Jednokrevetna
5	2	Slobodna	Dvokrevetna
2	2	Slobodna	Trokrevetna
8	2	Slobodna	Jednokrevetna
10	3	Slobodna	Trokrevetna
9	4	Slobodna	Dvokrevetna

Message

i Sistem je našao sobe po zadatoj vrednosti

OK

Detalji

5. Рецепционер/гост **бира** собу. (АПУСО)

Pretraga soba

ID	Sprat	Status	Vrsta sobe
2	2	Slobodna	Trokrevetna
5	2	Slobodna	Dvakrevetna
8	2	Slobodna	Jednokrevetna

Detalji

6. Рецепционер/гост **позива** систем да учита собу. (АПСО)

Опис акције: Рецепционер/гост кликом на дугме „Детаљи“ позива системску операцију **UcitajSobu(Soba)**

7. Систем **учитава** собу. (СО)
8. Систем **приказује** рецепционеру/госту податке о соби и поруку: „Систем је учитао собу“. (ИА)

Soba

Sprat:	2
Status:	<input type="checkbox"/> zauzeta
Vrsta sobe:	Jednokrevetna

Message

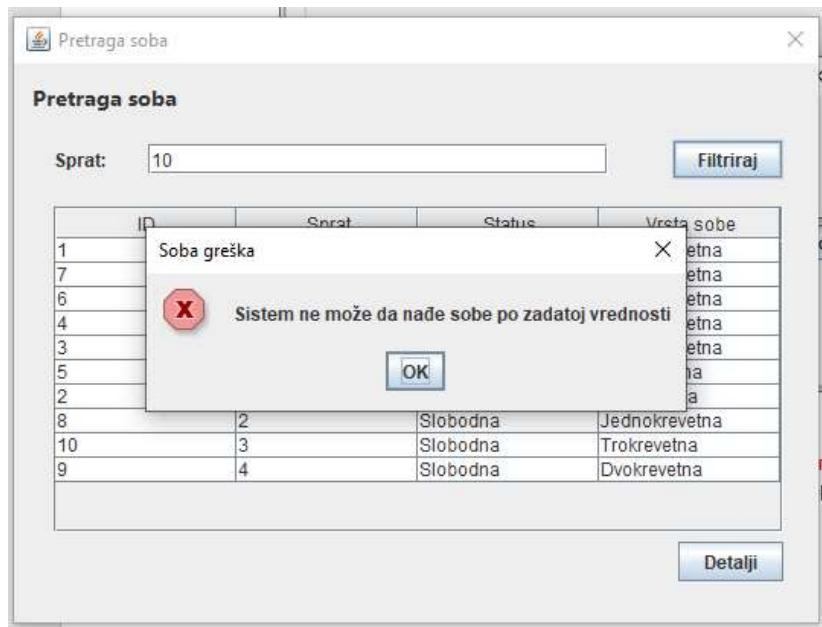
i Sistem je učitao sobu

OK

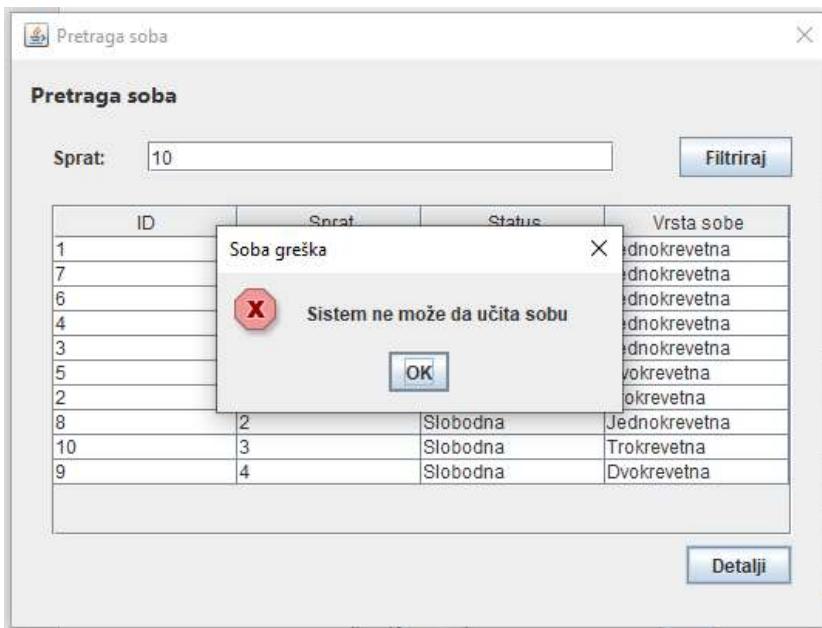
Ocenij Izmeni Otkaži Omogući izmene Sačuvaj

Алтернативна сценарија

- 4.1. Уколико систем не може да нађе собе он приказује рецепционеру/госту поруку: „Систем не може да нађе собе по задатој вредности“. Прекида се извршење сценарија. (ИА)



8.1. Уколико **систем** не може да учита **собу** он приказује **рецепционеру/госту** поруку:
„Систем не може да учита **собу**“. (ИА)



СК7: Случај коришћења – Измена података собе

Назив СК

Измена података собе

Актори СК

Рецепционер

Учесници СК

Рецепционер и систем (програм)

Предуслов: Систем је укључен и **рецепционер** је пријављен под својом шифром. Систем приказује форму за рад са **собом**. Учитана је листа врста соба.

ID	Sprat	Status	Vrsta sobe
1	1	Zauzeta	Jednokrevetna
7	1	Slobodna	Jednokrevetna
6	1	Slobodna	Jednokrevetna
4	1	Zauzeta	Jednokrevetna
3	1	Slobodna	Jednokrevetna
5	2	Slobodna	Dvokrevetna
2	2	Slobodna	Trokrevetna
8	2	Slobodna	Jednokrevetna
10	3	Slobodna	Trokrevetna
9	4	Slobodna	Dvokrevetna

Detalji

Основни сценарио СК

1. Рецепционер **уноси** вредност по којој претражује **собе**. (АПУСО)

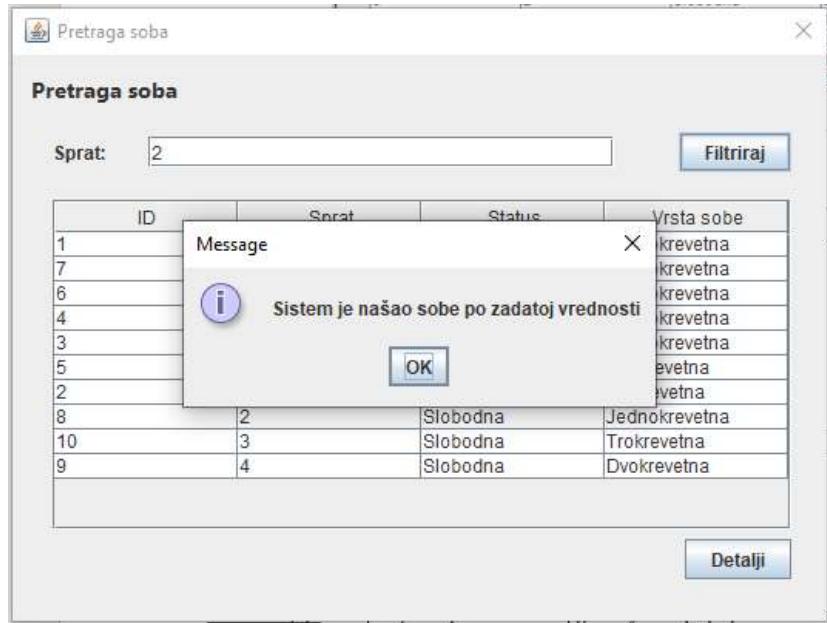
ID	Sprat	Status	Vrsta sobe
1	1	Zauzeta	Jednokrevetna
7	1	Slobodna	Jednokrevetna
6	1	Slobodna	Jednokrevetna
4	1	Zauzeta	Jednokrevetna
3	1	Slobodna	Jednokrevetna
5	2	Slobodna	Dvokrevetna
2	2	Slobodna	Trokrevetna
8	2	Slobodna	Jednokrevetna
10	3	Slobodna	Trokrevetna
9	4	Slobodna	Dvokrevetna

Detalji

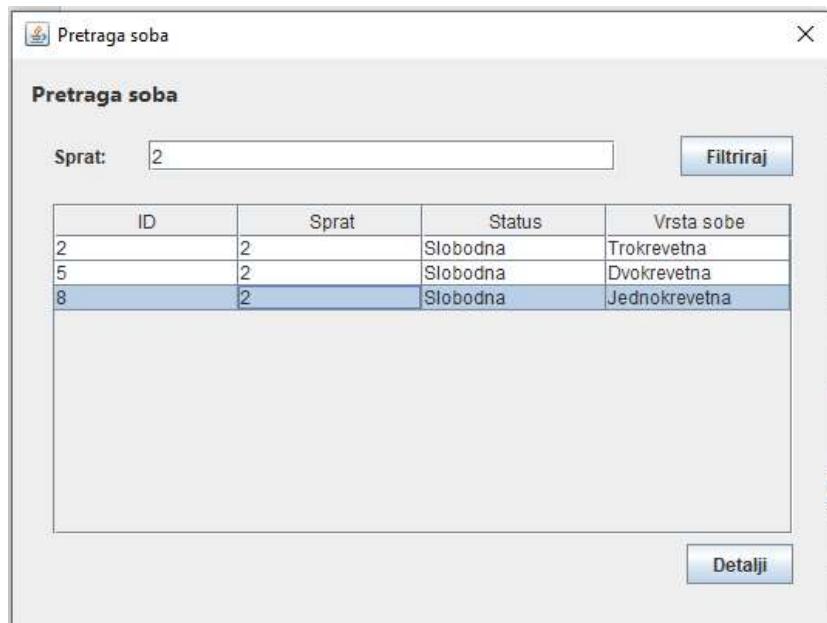
2. Рецепционер **позива** систем да нађе **собе** по задатој вредности. (АПСО)

Опис акције: Рецепционер кликом на дугме „Филтрирај“ позива системску операцију **NadjiSobe(Soba,List<Soba>)**

3. Систем тражи собе по задатој вредности. (СО)
4. Систем приказује рецепционеру податке о собама и поруку: „Систем је нашао собе по задатој вредности“. (ИА)



5. Рецепционер бира собу. (АПУСО)

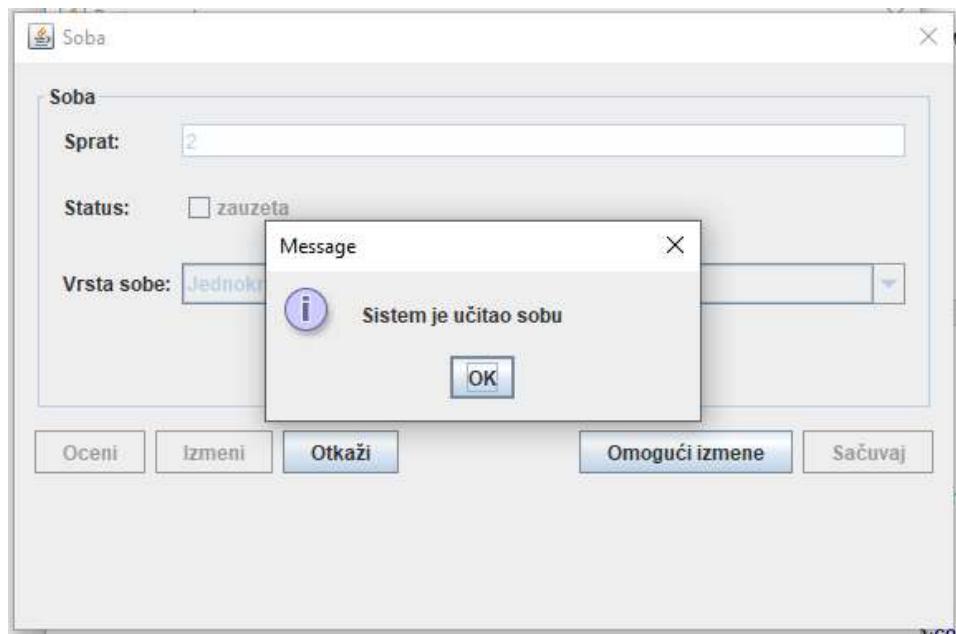


6. Рецепционер позива систем да учита собу. (АПСО)

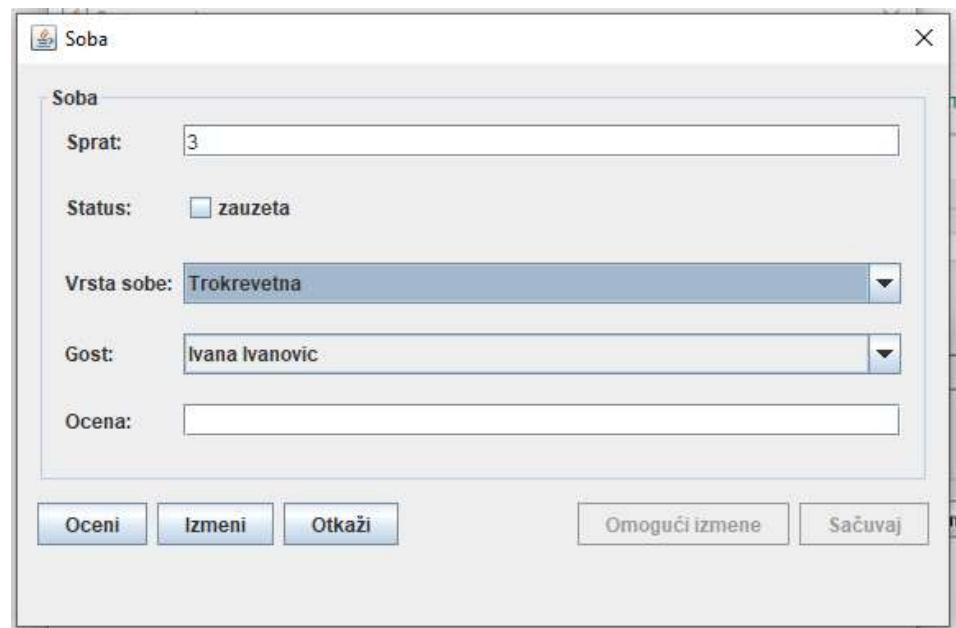
Опис акције: Рецепционер кликом на дугме „Детаљи“ позива системску операцију **UcitajSobu(Soba)**

7. Систем учитава собу. (СО)

8. Систем приказује рецепционеру податке о соби и поруку: „Систем је учитао собу“. (ИА)



9. Рецепционер уноси (менја) податке о соби. (АПУСО)



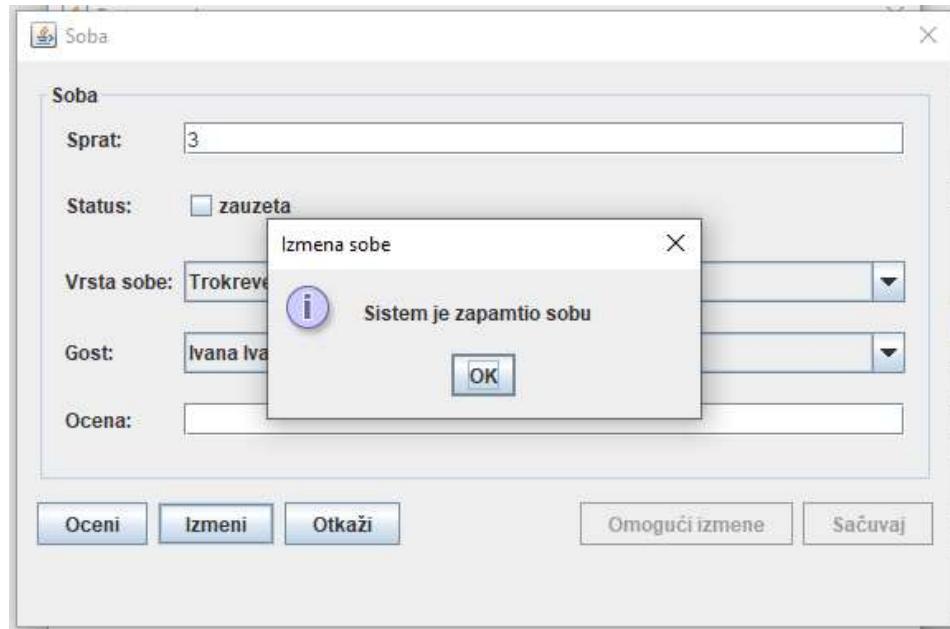
10. Рецепционер контролише да ли је исправно унео податке о соби. (АНСО)

11. Рецепционер позива систем да запамти податке о соби. (АПСО)

Опис акције: Рецепционер кликом на дугме „Измени“ позива системску операцију **ZapamtiSobu(Soba)**

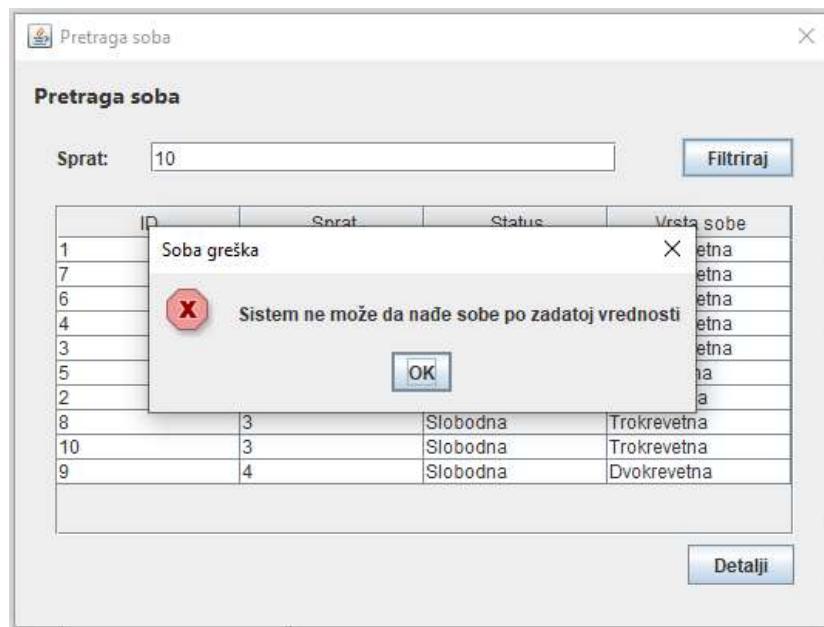
12. Систем памти податке о соби. (СО)

13. Систем приказује рецепционеру изменјену собу и поруку: „Систем је запамтио собу“. (ИА)

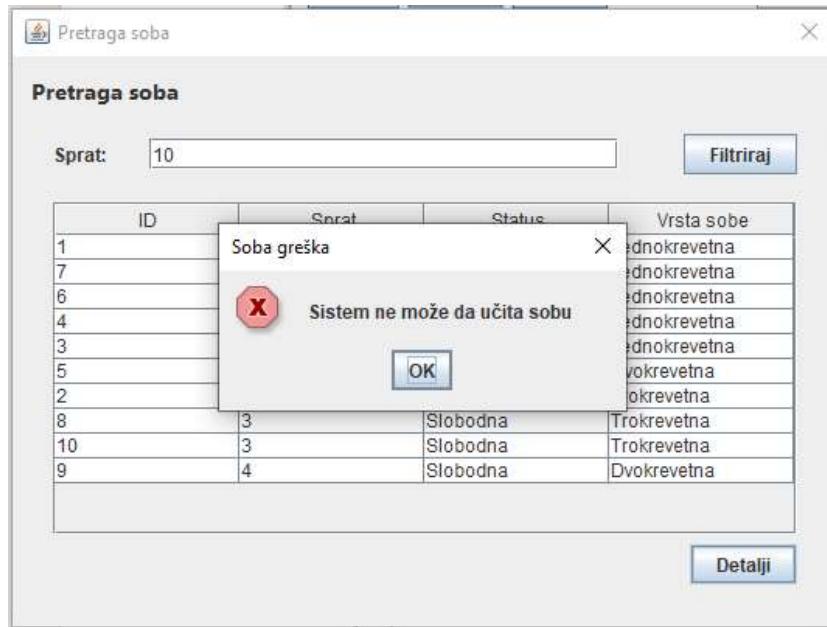


Алтернативна сценарија

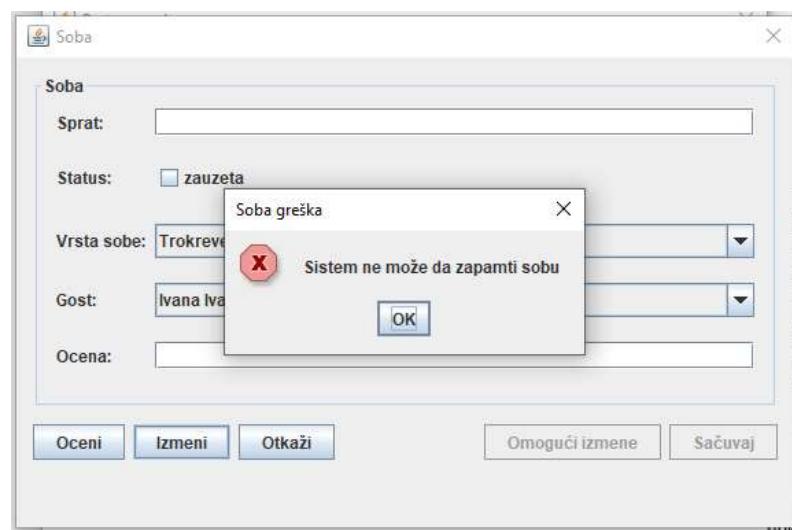
4.1. Уколико систем не може да нађе собе он приказује рецепционеру поруку: „Систем не може да нађе собе по задатој вредности“. Прекида се извршење сценарија. (ИА)



8.1. Уколико систем не може да учита собу он приказује рецепционеру поруку: „Систем не може да учита собу“. Прекида се извршење сценарија. (ИА)



13.1. Уколико **систем** не може да запамти податке о **соби** он приказује **рецепционеру** поруку: „**Систем не може да запамти собу**“. (ИА)



СК8: Случај коришћења – Оцена собе

Назив СК

Оцена собе

Актори СК

Гост

Учесници СК

Гост и **систем** (програм)

Предуслов: Систем је укључен и гост је пријављен под својом шифром. Систем приказује форму за рад са собом.

ID	Sprat	Status	Vrsta sobe
1	1	Zauzeta	Jednokrevetna
7	1	Slobodna	Jednokrevetna
6	1	Slobodna	Jednokrevetna
4	1	Zauzeta	Jednokrevetna
3	1	Slobodna	Jednokrevetna
5	2	Slobodna	Dvokrevetna
2	2	Slobodna	Trokrevetna
8	3	Slobodna	Trokrevetna
10	3	Slobodna	Trokrevetna
9	4	Slobodna	Dvokrevetna

Основни сценарио СК

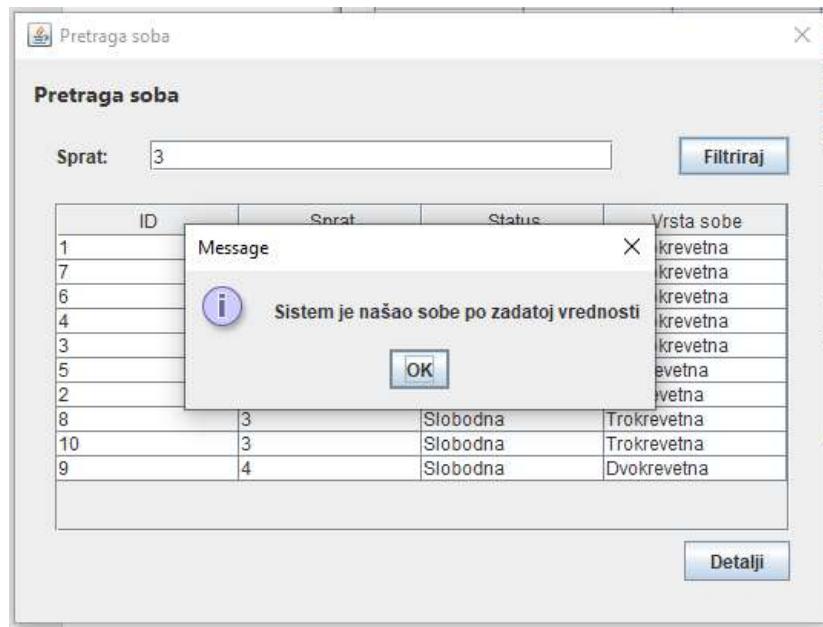
- Гост уноси вредност по којој претражује собе. (АПУСО)

ID	Sprat	Status	Vrsta sobe
1	1	Zauzeta	Jednokrevetna
7	1	Slobodna	Jednokrevetna
6	1	Slobodna	Jednokrevetna
4	1	Zauzeta	Jednokrevetna
3	1	Slobodna	Jednokrevetna
5	2	Slobodna	Dvokrevetna
2	2	Slobodna	Trokrevetna
8	3	Slobodna	Trokrevetna
10	3	Slobodna	Trokrevetna
9	4	Slobodna	Dvokrevetna

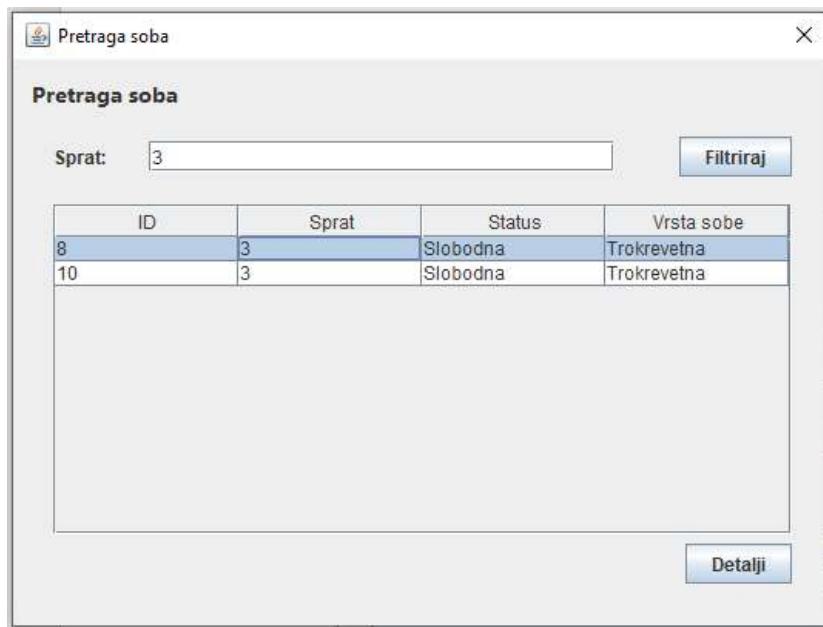
- Гост позива систем да нађе собе по задатој вредности. (АПСО)

Опис акције: Рецепционер кликом на дугме „Филтрирај“ позива системску операцију NadjiSobe(Soba,List<Soba>)

- Систем тражи собе по задатој вредности . (СО)
- Систем приказује госту податке о собама и поруку : „Систем је нашао собе по задатој вредности“. (ИА)



5. Гост бира собу. (АПУСО)

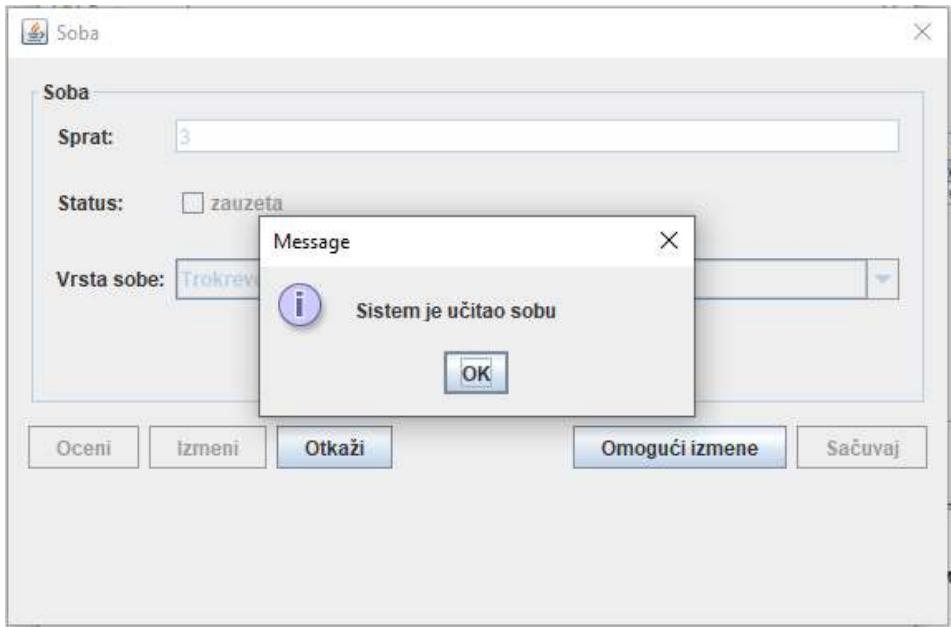


6. Гост позива систем да учита собу. (АПСО)

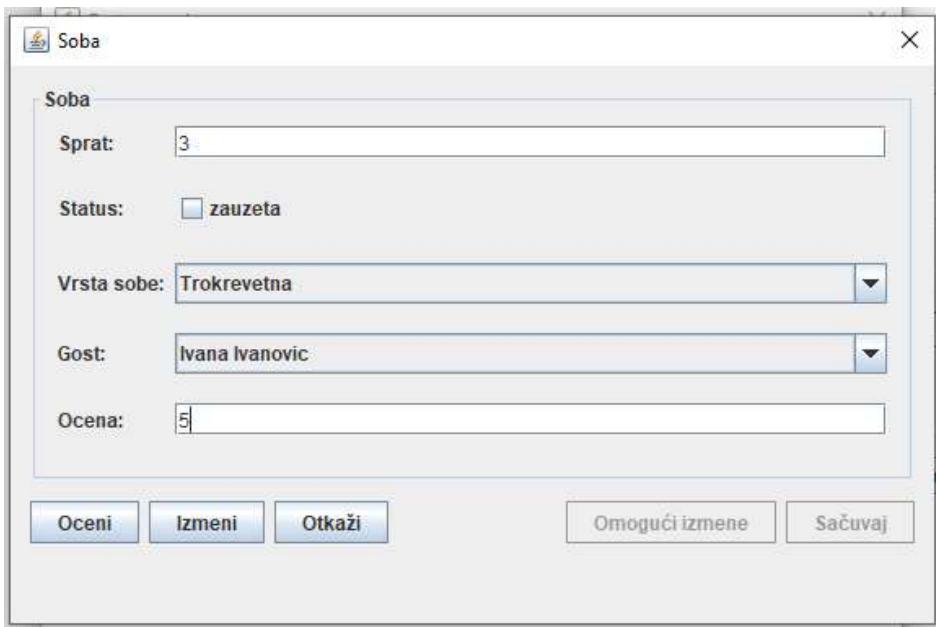
Опис акције: Рецепционер кликом на дугме „Детаљи“ позива системску операцију **UcitajSobu(Soba)**

7. Систем учитава собу. (СО)

8. Систем приказује госту податке о соби и поруку: „Систем је учитao собу“. (ИА)



9. Гост уноси (мења) оцену собе. (АПУСО)



10. Гост контролише да ли је исправно унео оцену собе. (АНСО)

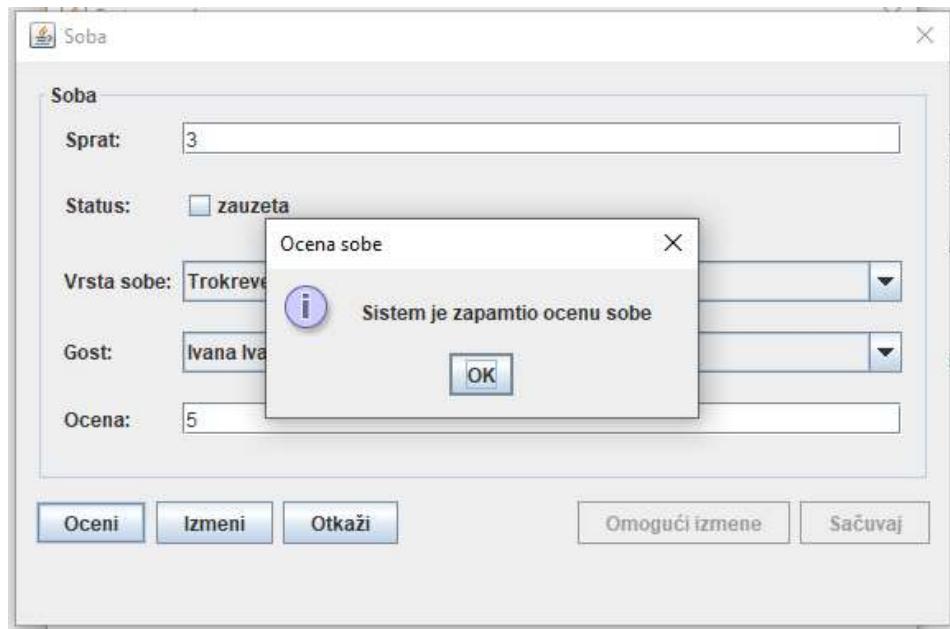
11. Гост позива систем да запамти оцену собе. (АПСО)

Опис акције: Гост кликом на дугме „Оцени“ позива системску операцију

ZapamtiOcenuSobe(Gost, Soba, Ocena)

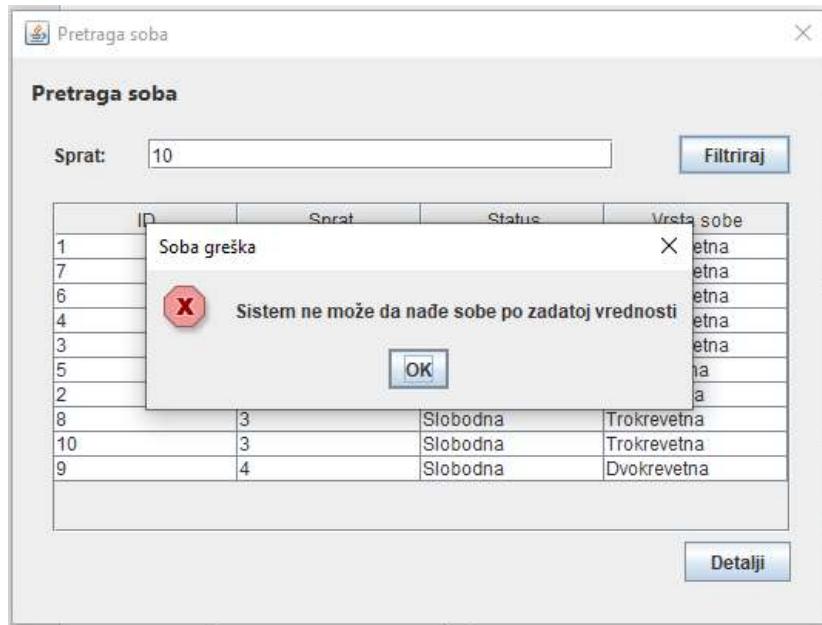
12. Систем памти оцену собе. (СО)

13. Систем приказује госту запамћену оцену собе и поруку: „Систем је запамтио оцену собе“. (ИА)

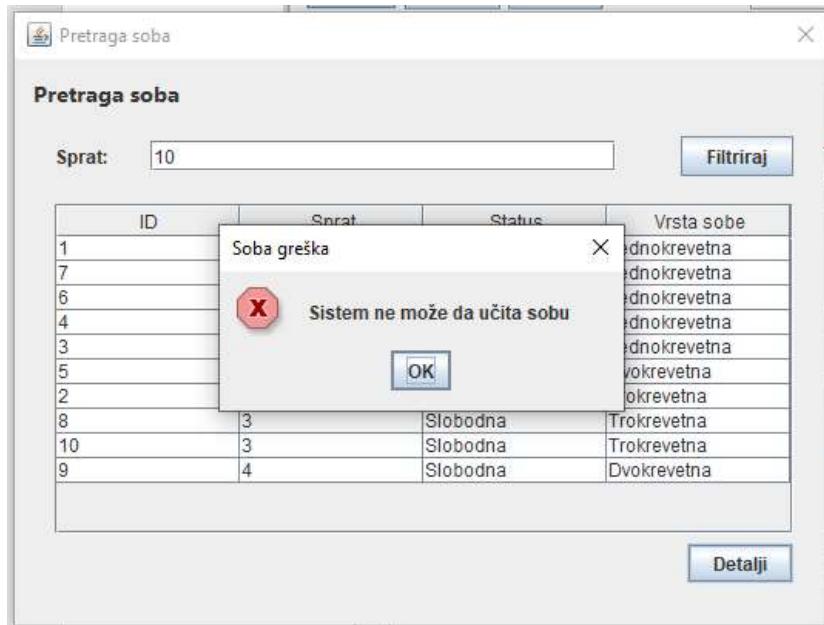


Алтернативна сценарија

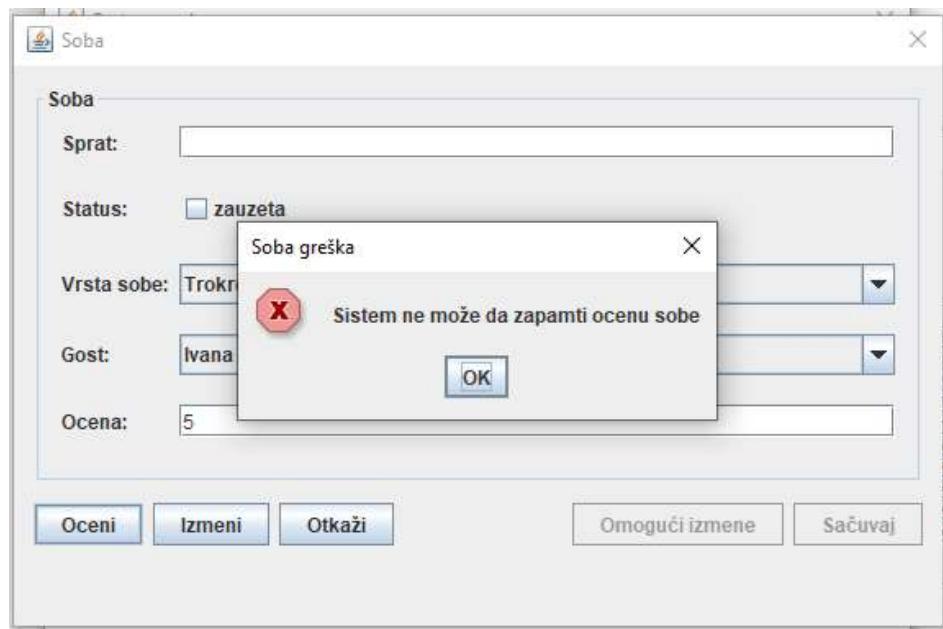
4.1. Уколико систем не може да нађе собе он приказује госту поруку: „Систем не може да нађе собе по задатој вредности“. Прекида се извршење сценарија. (ИА)



8.1. Уколико систем не може да учита собе он приказује госту поруку: „Систем не може да учита собу“. Прекида се извршење сценарија. (ИА)



13.1. Уколико **систем** не може да запамти оцену **собе** он приказује **госту** поруку: „Систем не може да запамти оцену **собе**“. (ИА)



СК9: Случај коришћења – Креирање резервације (сложен СК)

Назив СК

Креирање резервације

Актори СК

Рецепционер

Учесници СК

Рецепционер и систем (програм)

Предуслов: Систем је укључен и **рецепционер** је пријављен под својом шифром. Систем приказује форму за рад са **резервацијом**. Учитана је листа гостију и листа соба.

The screenshot shows a Windows application window titled "Rezervacija". The window is divided into two main sections: "Rezervacija" and "Stavka rezervacije".
In the "Rezervacija" section, there are three input fields:

- "Datum od (dd.MM.yyyy)" (Date from (dd.MM.yyyy))
- "Datum do (dd.MM.yyyy)" (Date to (dd.MM.yyyy))
- "Ugovarač:" (Arranger) with the value "Ivana Ivanovic"

A table below these fields has three columns: "Redni broj" (Row number), "Gost" (Guest), and "Soba" (Room).
In the "Stavka rezervacije" section, there are two input fields:

- "Gost:" (Guest) with the value "Ivana Ivanovic"
- "Soba:" (Room) with the value "Broj sobe: 1"

At the bottom of the window, there are several buttons:

- "Obrisi" (Delete)
- "Dodaj" (Add)
- "Otkaži" (Cancel)
- "Sačuvaj" (Save)

Основни сценарио СК

1. Рецепционер уноси податке о резервацији. (АПУСО)

Rezervacija

Redni broj	Gost	Soba
1	Ivana Ivanovic	Broj sobe: 2
2	Jovan Jovanovic	Broj sobe: 2

Stavka rezervacije

Gost: Jovan Jovanovic

Soba: Broj sobe: 2

Dodaj

Obriši

Otkaži

Sačuvaj

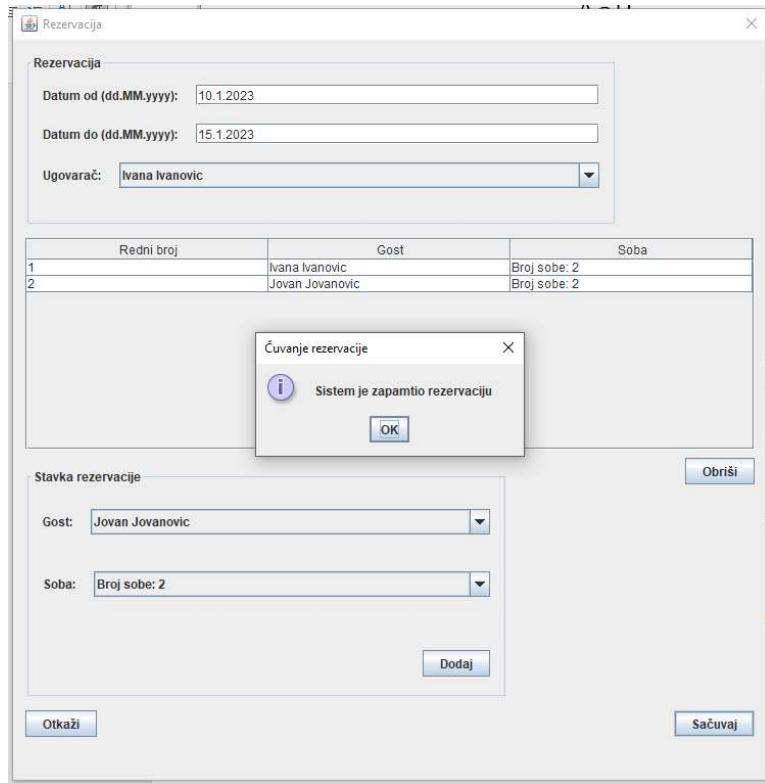
2. **Рецепционер контролише** да ли је коректно унео податке о **резервацији**. (АНСО)

3. **Рецепционер позива систем** да запамти податке о **резервацији**. (АПСО)

Опис акције: Рецепционер кликом на дугме „Сачувај“ позива системску операцију **ZapamtiRezervaciju(Rezervacija)**

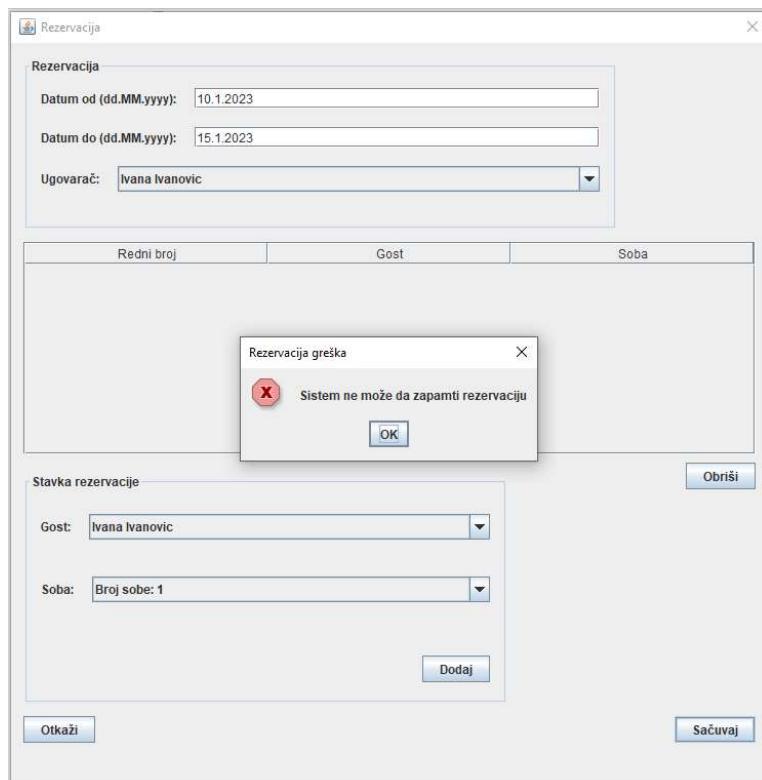
4. **Систем памти** податке о **резервацији**. (СО)

5. **Систем приказује** **рецепционеру** запамћену **резервацију** и поруку: „**Систем** је запамтио **резервацију**“. (ИА)



Алтернативна сценарија

5.1. Уколико систем не може да запамти податке о резервацији он приказује рецепционеру поруку: „Систем не може да запамти резервацију“. (ИА)



СК10: Случај коришћења – Претраживање резервације (сложен СК)

Назив СК

Претраживање резервације

Актори СК

Рецепционер

Учесници СК

Рецепционер и систем (програм)

Предуслов: Систем је укључен и **рецепционер** је пријављен под својом шифром. Систем приказује форму за рад са резервацијом.

ID	Datum od	Datum do	Ugovarač
1	25.01.2023	30.01.2023	Ivana Ivanovic
2	25.01.2023	30.01.2023	Ivana Ivanovic
3	01.01.2023	05.01.2023	Luka Lukic
4	01.02.2023	15.02.2023	Ivana Ivanovic
6	05.05.2023	10.05.2023	Ivana Trajkov
7	13.12.2022	23.01.2023	Ivana Trajkov
8	05.01.2023	25.01.2023	Luka Lukic
9	10.01.2023	15.01.2023	Ivana Ivanovic

Detalji

Основни сценарио СК

1. Рецепционер уноси вредност по којој претражује резервације. (АПУСО)

Pretraga rezervacija

Datum početka (dd.MM.yyyy): **Filtriraj**

ID	Datum od	Datum do	Ugovarač
1	25.01.2023	30.01.2023	Ivana Ivanovic
2	25.01.2023	30.01.2023	Ivana Ivanovic
3	01.01.2023	05.01.2023	Luka Lukic
4	01.02.2023	15.02.2023	Ivana Ivanovic
6	05.05.2023	10.05.2023	Ivana Trajkov
7	13.12.2022	23.01.2023	Ivana Trajkov
8	05.01.2023	25.01.2023	Luka Lukic
9	10.01.2023	15.01.2023	Ivana Ivanovic

Detalji

2. Рецепционер **позива систем** да нађе **резервације** по задатој вредности. (АПСО)

Опис акције: Рецепционер кликом на дугме „Филтрирај“ позива системску операцију **NadjiRezervacije(Rezervacija,List<Rezervacija>)**

3. Систем **тражи резервације** по задатој вредности. (СО)
4. Систем **приказује** **рецепционеру** податке о резервацијама и поруку: „Систем је нашао резервације по задатој вредности“. (ИА)

Pretraga rezervacija

Datum početka (dd.MM.yyyy): **Filtriraj**

ID	Datum od	Datum do	Ugovarač
1	Message		Xovic
2			Xovic
3			Xovic
4			Xovic
6			XOV
7			XOV
8			XOV
9	10.01.2023	15.01.2023	Ivana Ivanovic

Message

Sistem je našao rezervacije po zadatoj vrednosti

OK

Detalji

5. Рецепционер **бира** резервацију. (АПУСО)

Pretraga rezervacija

ID	Datum od	Datum do	Ugovarač
1	25.01.2023	30.01.2023	Ivana Ivanovic
2	25.01.2023	30.01.2023	Ivana Ivanovic

Detalji

6. **Рецепционер позива систем да учита резервацију.** (АПСО)

Опис акције: Рецепционер кликом на дугме „Детаљи“ позива системску операцију **UcitajRezervaciju(Rezervacija)**

7. **Систем учитава резервацију.** (СО)

8. **Систем приказује рецепционеру податке о резервацији и поруку:** „**Систем је учитао резервацију**“. (ИА)

Rezervacija

Redni broj	Gost	Soba
1	Ivana Ivanovic	Broj sobe: 1
2	Luka Lukic	Broj sobe: 1

Message
Sistem je učitao rezervaciju
OK

Stavka rezervacije

Gost: Ivana Ivanovic

Soba: Broj sobe: 1

Dodaj

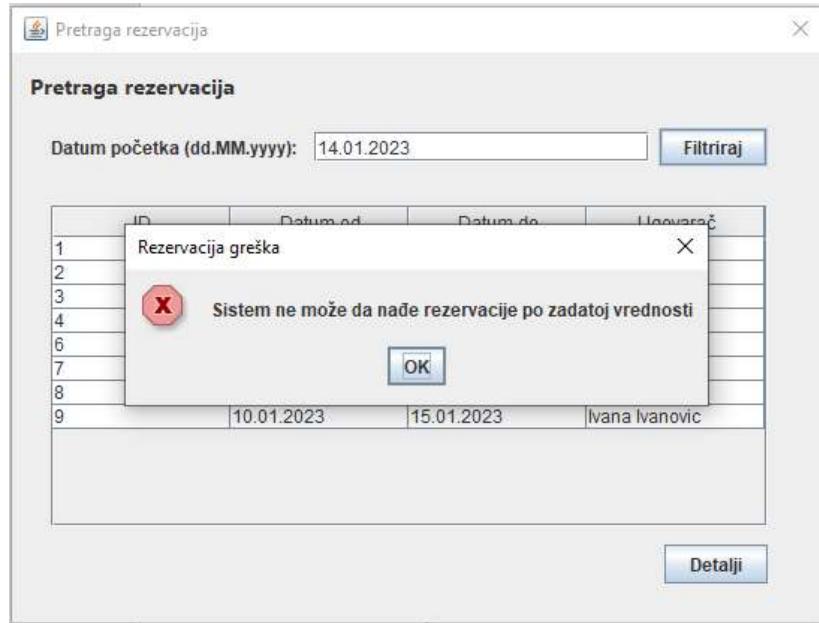
Obriši

Otkaži

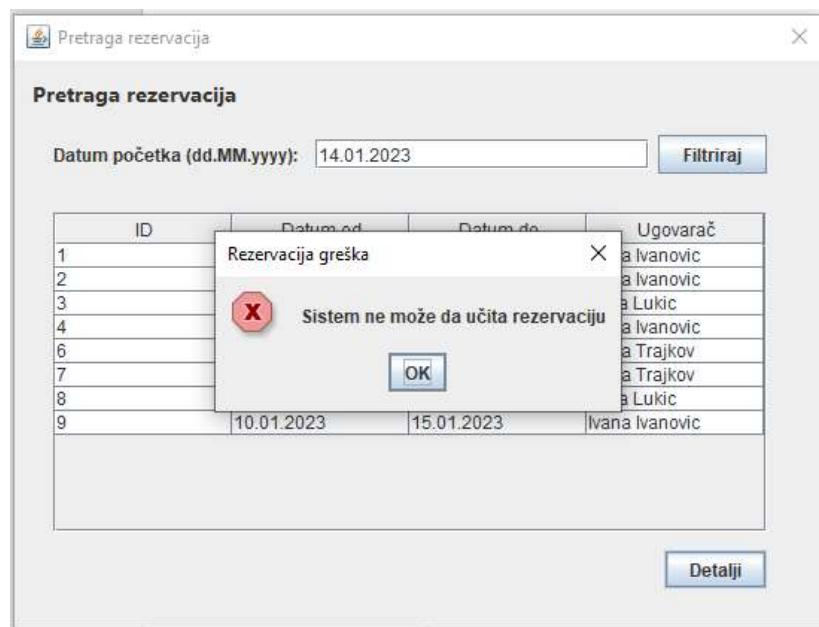
Sačuvaj

Алтернативна сценарија

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



- 8.1. Уколико **систем** не може да учита **резервацију** он приказује **рецепционеру** поруку:
„**Систем** не може да учита **резервацију**“. (ИА)



3.1.2. 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.

3.2. 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 који садржи пословну логику,
- Део за комуникацију са складиштем података (брокер базе података).

Комуникација са клијентима

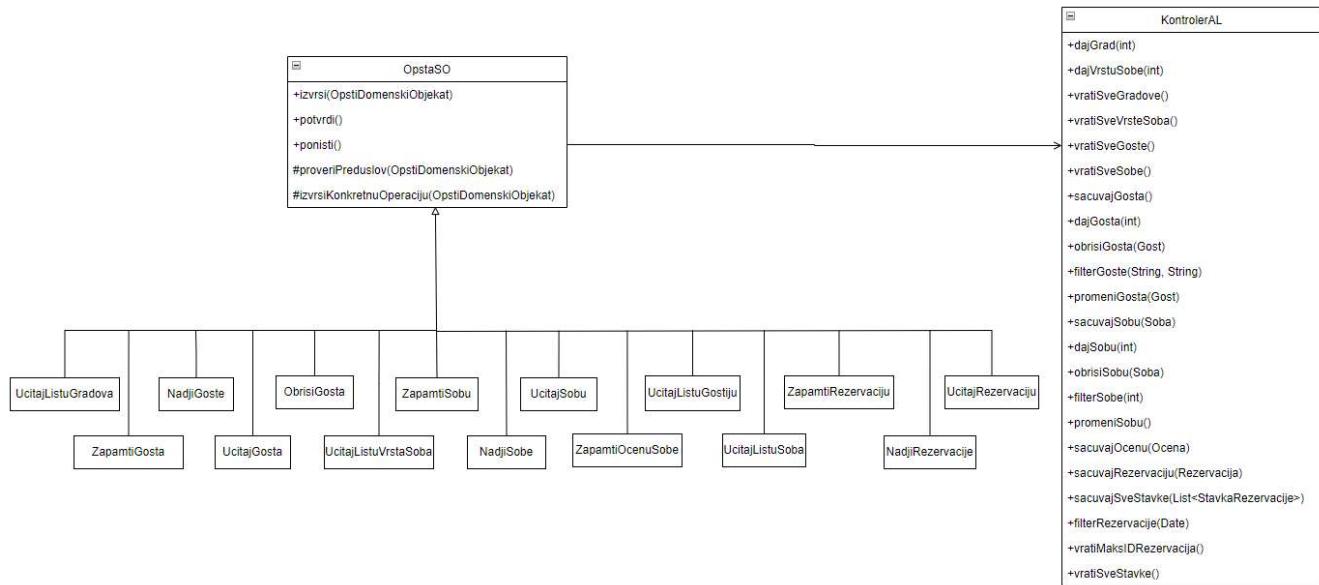
Део за комуникацију подиже серверски сокет који ослушкује мрежу. Када клијентски сокет успостави конекцију са серверским сокетом, тада сервер генерише нит која ће успоставити двосмерну комуникацију са клијентом.

Слање и примање података од клијента се обавља разменом објеката класе Request i Response и остварује се преко сокета.

Клијент шаље захтев за извршење неке од системских операција до одговарајуће нити која је повезана са тим клијентом. Та нит приhvата 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.

3.2.1. Контролер апликационе логике

Контролер апликационе логике приhvата захтев за извршење системске операције од нити клијента и даље га преусмерава до класа које су одговорне за извршење системских операција. Након извршења системске операције контролер апликационе логике приhvата резултат и прослеђује га позиваоцу (нити клијента).



3.2.2. Пословна логика

Пројектовање понашања софтверског система – системске операције

За сваку системску операцију треба направити концептуална решења која су директно повезана са логиком проблема.

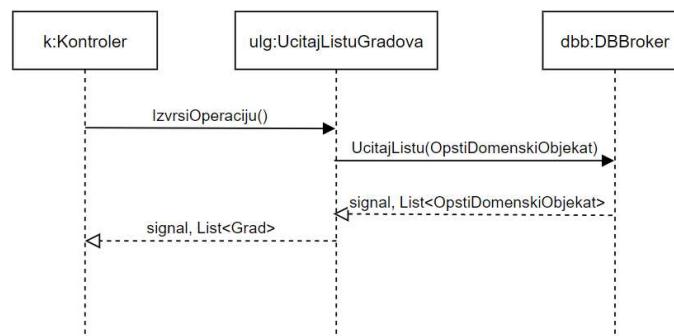
За сваки уговор пројектује се концептуално решење.

Уговор УГ1: UcitajListuGradova(List<Grad>) signal;

Веза са СК: СК1, СК3

Предуслови:

Постуслови:

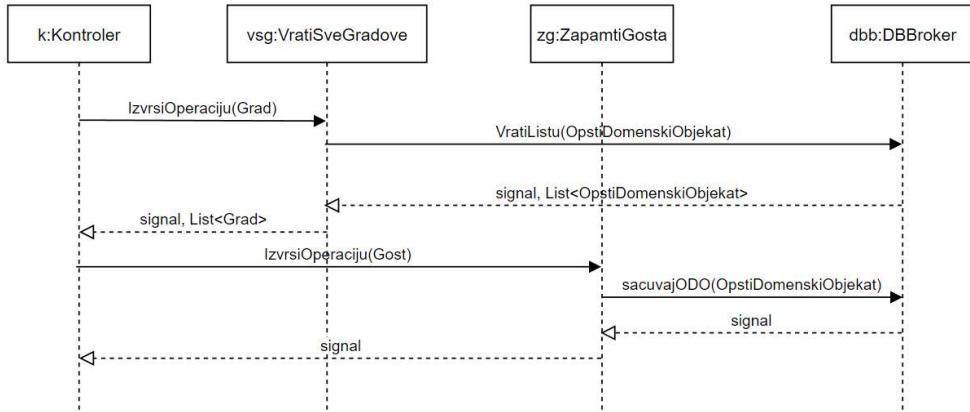


Уговор УГ2: ZapamtiGosta(Gost) signal;

Веза са СК: СК1, СК3

Предуслови: Вредносна и структурна ограничења над објектом **Gost** морају бити задовољена.
Учитана листа градова из базе података у комбо боксу.

Постуслови: Подаци о госту су запамћени.

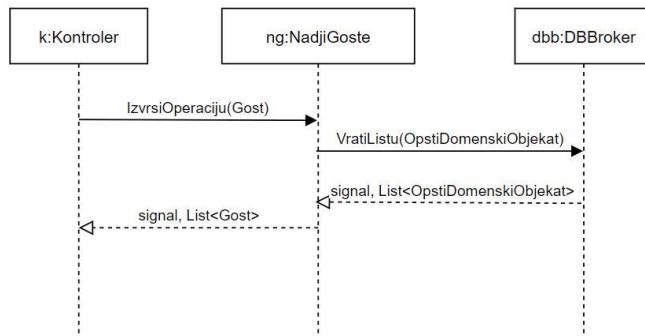


Уговор УГ3: NadjiGoste(Gost, List<Gost>) signal;

Веза са СК: СК2, СК3, СК4

Предуслови:

Постуслови:

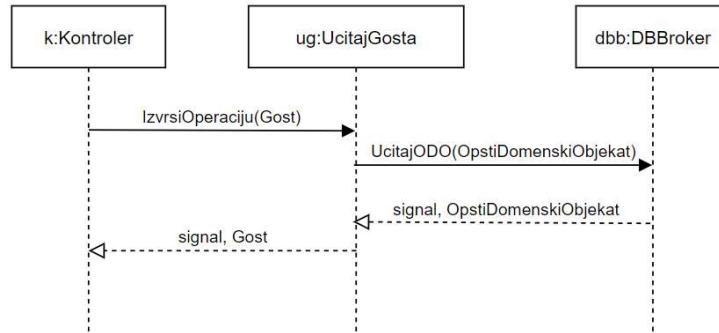


Уговор УГ4: UcitajGosta(Gost) signal;

Веза са СК: СК2, СК3, СК4

Предуслови:

Постуслови:

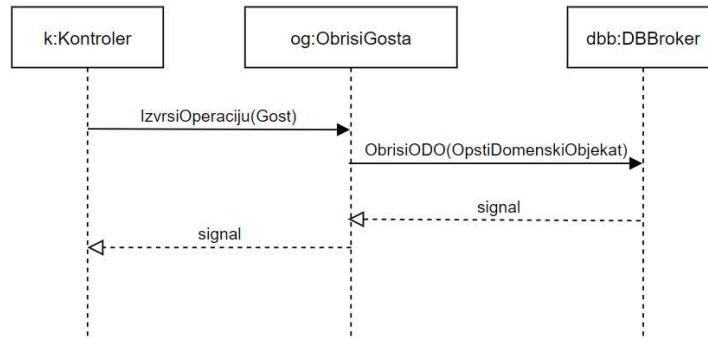


Уговор УГ5: ObrisGosta(Gost) signal;

Веза са СК: СК4

Предуслови: Структурна ограничења над објектом **Gost** морају бити задовољена.

Постуслови: Гост је обрисан.

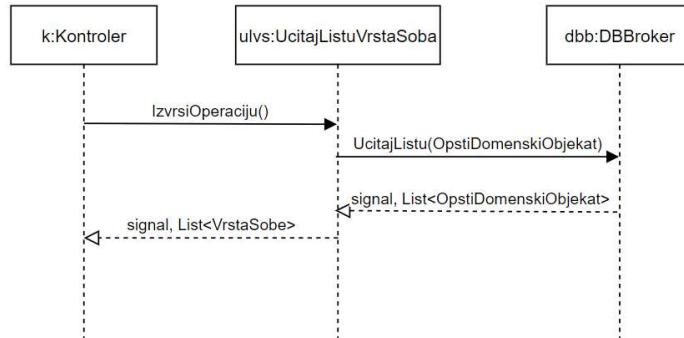


Уговор УГ6: UcitajListuVrstaSoba(List<VrstaSobe>) signal;

Веза са СК: СК5, СК7

Предуслови:

Постуслови:

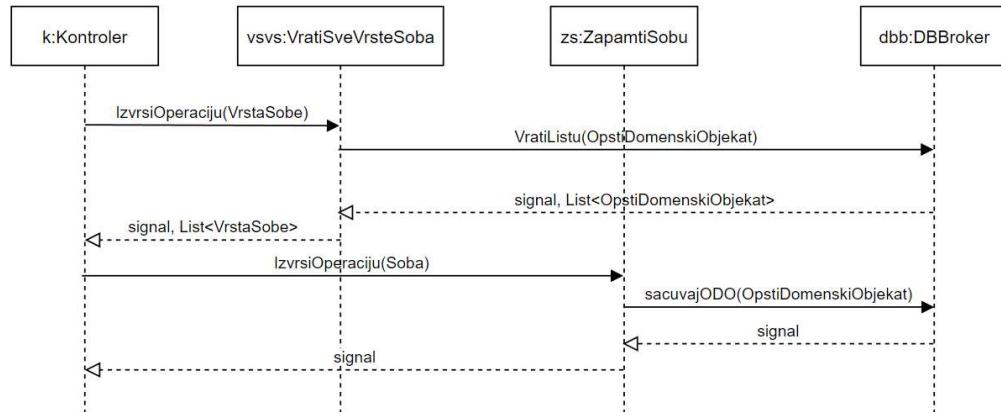


Уговор УГ7: ZapamtiSobu(Soba) signal;

Веза са СК: СК5, СК7

Предуслови: Вредносна и структурна ограничења над објектом **Soba** морају бити задовољена.
Учитана листа врста соба из базе података у комбо бокс.

Постуслови: Подаци о соби су запамћени.

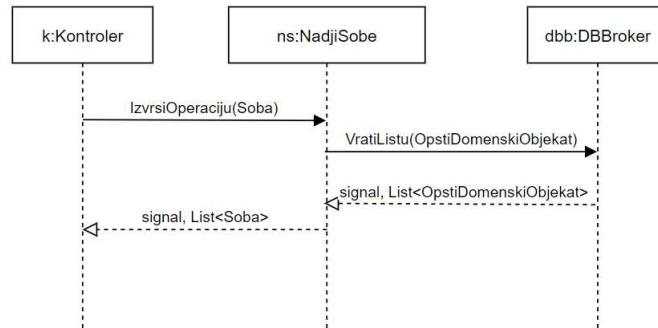


Уговор УГ8: NadjiSobe(Soba, List<Soba>) signal;

Веза са СК: СК6, СК7, СК8

Предуслови:

Постуслови:

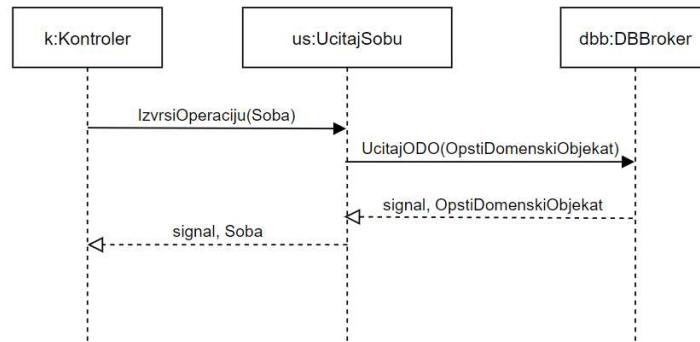


Уговор УГ9: UcitajSobu(Soba) signal;

Веза са СК: СК6, СК7, СК8

Предуслови:

Постуслови:

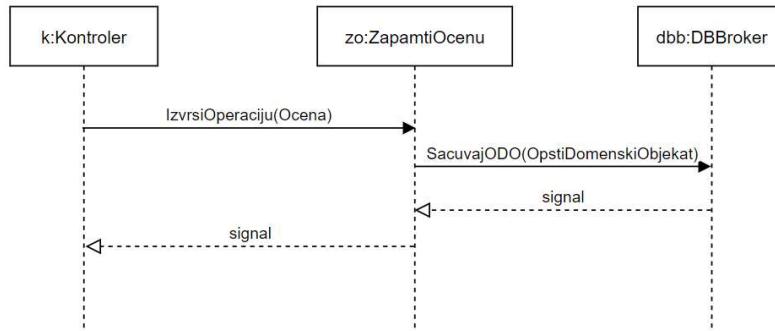


Уговор УГ10: ZapamtiOcenuSobe(Gost, Soba, Ocena) signal;

Веза са СК: СК8

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

Постуслови: Подаци о оцени су запамћени.

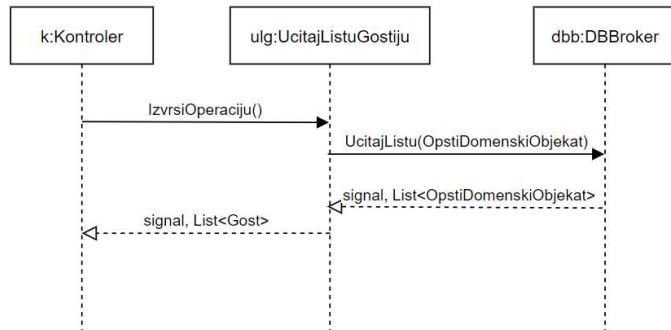


Уговор УГ11: UcitajListuGostiju(List<Gost>) signal;

Веза са СК: СК9

Предуслови:

Постуслови:

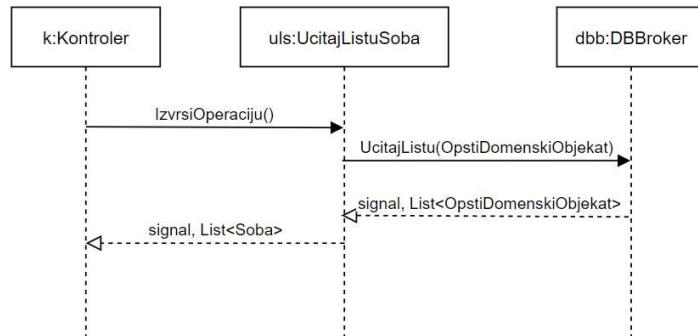


Уговор УГ12: UcitajListuSoba(List<Soba>) signal;

Веза са СК: СК9

Предуслови:

Постуслови:

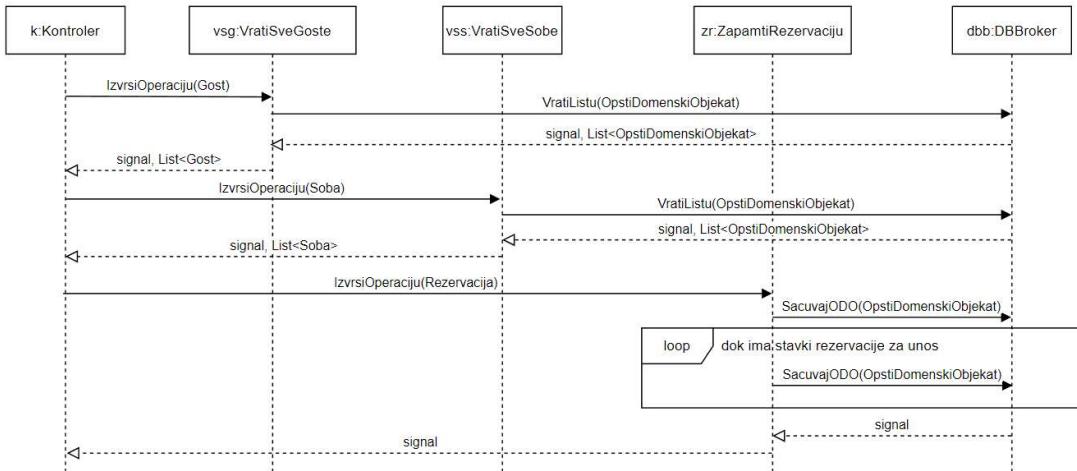


Уговор УГ13: ZapamtiRezervaciju(Rezervacija) signal;

Веза са СК: СК9

Предуслови: Вредносна и структурна ограничења над објектима **Rezervacija** и **StavkaRezervacije** морају бити задовољена. Учитана листа гостију и листа соба из базе података у комбо бокс.

Постуслови: Подаци о резервацији су запамћени.

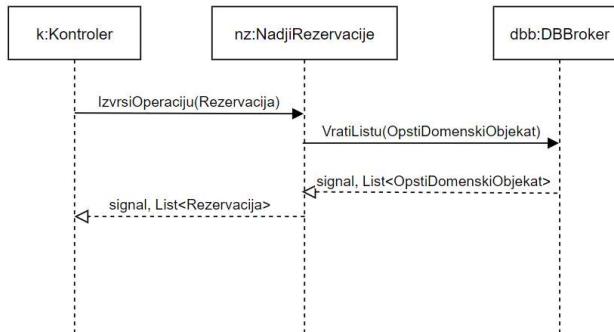


Уговор УГ14: NadjiRezervacije(Rezervacija, List<Rezervacija>) signal;

Веза са СК: СК10

Предуслови:

Постуслови:

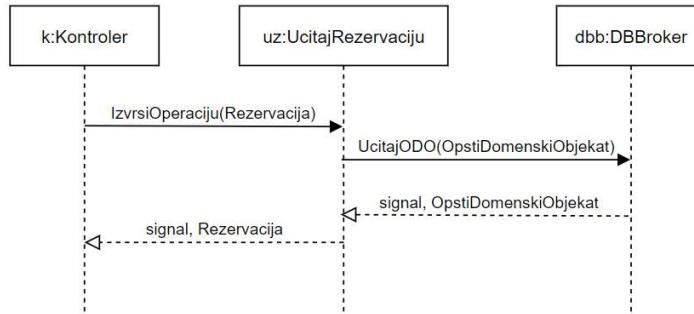


Уговор УГ15: UcitajRezervaciju(Rezervacija) signal;

Веза са СК: СК10

Предуслови:

Постуслови:

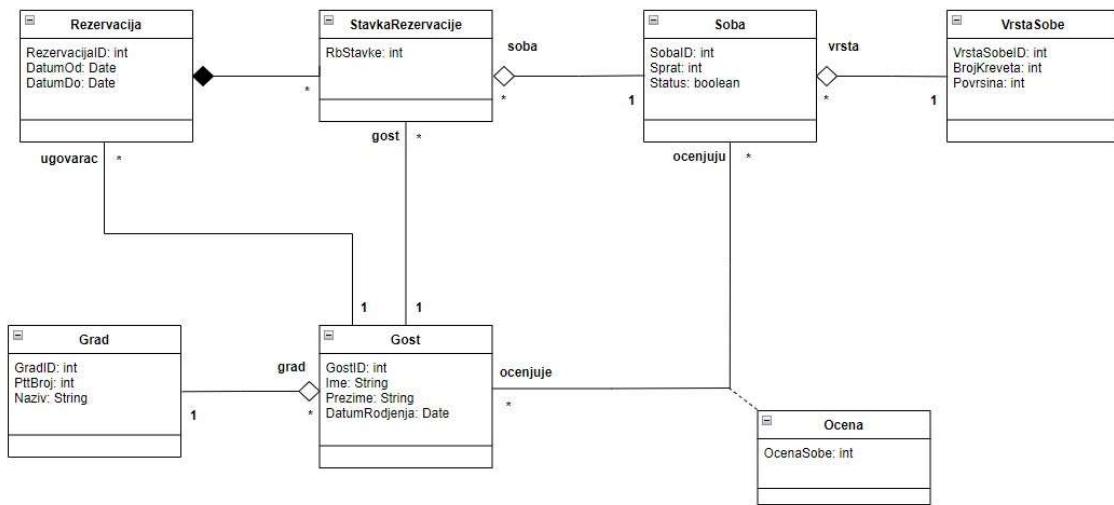


Класе које су одговорне за извршење системских операција наслеђују класу OpstaSO како би могле да се повежу са базом. OpstaSO представља апстрактну класу чија главна метода (izvrsi()) у себи садржи отварање конекције са базом, валидацију, проверу предуслова, извршење операције, потврду у бази уколико је извршење операције успешно, поништавање уколико извршење операције није било успешно и затварање конекције. Свака од системских операција даје своју имплементацију методе за проверу предуслова, уколико постоји, и методе за извршење конкретне системске операције.

Пројектовање структуре софтверског система

На основу концептуалних класа праве се софтверске класе структуре.

Концептуалне класе:



Софтверске класе структуре:

```

public class Gost implements OpstiDomenskiObjekat {

    private int gostID;
    private String ime;
    private String prezime;
    private Date datumRodjenja;
    private Grad grad;

    public Gost() {
    }

    public Gost(int gostID, String ime, String prezime, Date datumRodjenja, Grad grad) {
        this.gostID = gostID;
        this.ime = ime;
        this.prezime = prezime;
        this.datumRodjenja = datumRodjenja;
        this.grad = grad;
    }
}
  
```

```
public class Grad implements OpstiDomenskiObjekat{
    private int gradID;
    private int pttBroj;
    private String naziv;

    public Grad() {
    }

    public Grad(int gradID, int pttBroj, String naziv) {
        this.gradID = gradID;
        this.pttBroj = pttBroj;
        this.naziv = naziv;
    }
}

public class Ocena implements OpstiDomenskiObjekat{
    private Gost gost;
    private Soba soba;
    private int ocenaSobe;

    public Ocena() {
    }

    public Ocena(Gost gost, Soba soba, int ocenaSobe) {
        this.gost = gost;
        this.soba = soba;
        this.ocenaSobe = ocenaSobe;
    }
}

public class Rezervacija implements OpstiDomenskiObjekat{
    private int rezervacijaID;
    private Date datumOd;
    private Date datumDo;
    private Gost gost;

    private List<StavkaRezervacije> stavkeRezervacije;

    public Rezervacija() {
        stavkeRezervacije = new ArrayList<>();
    }

    public Rezervacija(int rezervacijaID, Date datumOd, Date datumDo, Gost gost, List<StavkaRezervacije> stavkeRezervacije) {
        this.rezervacijaID = rezervacijaID;
        this.datumOd = datumOd;
        this.datumDo = datumDo;
        this.gost = gost;
        this.stavkeRezervacije = stavkeRezervacije;
    }
}
```

```
public class Soba implements OpstiDomenskiObjekat{
    private int sobaID;
    private int sprat;
    private boolean status;
    private VrstaSobe vrstaSobe;

    public Soba() {
    }

    public Soba(int sobaID, int sprat, boolean status, VrstaSobe vrstaSobe) {
        this.sobaID = sobaID;
        this.sprat = sprat;
        this.status = status;
        this.vrstaSobe = vrstaSobe;
    }
}

public class StavkaRezervacije implements OpstiDomenskiObjekat{
    private Rezervacija rezervacija;
    private int rbStavke;
    private Gost gost;
    private Soba soba;

    public StavkaRezervacije() {
    }

    public StavkaRezervacije(Rezervacija rezervacija, int rbStavke, Gost gost, Soba soba) {
        this.rezervacija = rezervacija;
        this.rbStavke = rbStavke;
        this.gost = gost;
        this.soba = soba;
    }
}

public class VrstaSobe implements OpstiDomenskiObjekat{
    private int vrstaSobeID;
    private int brojKreveta;
    private int povrsina;

    public VrstaSobe() {
    }

    public VrstaSobe(int vrstaSobeID, int brojKreveta, int povrsina) {
        this.vrstaSobeID = vrstaSobeID;
        this.brojKreveta = brojKreveta;
        this.povrsina = povrsina;
    }
}
```

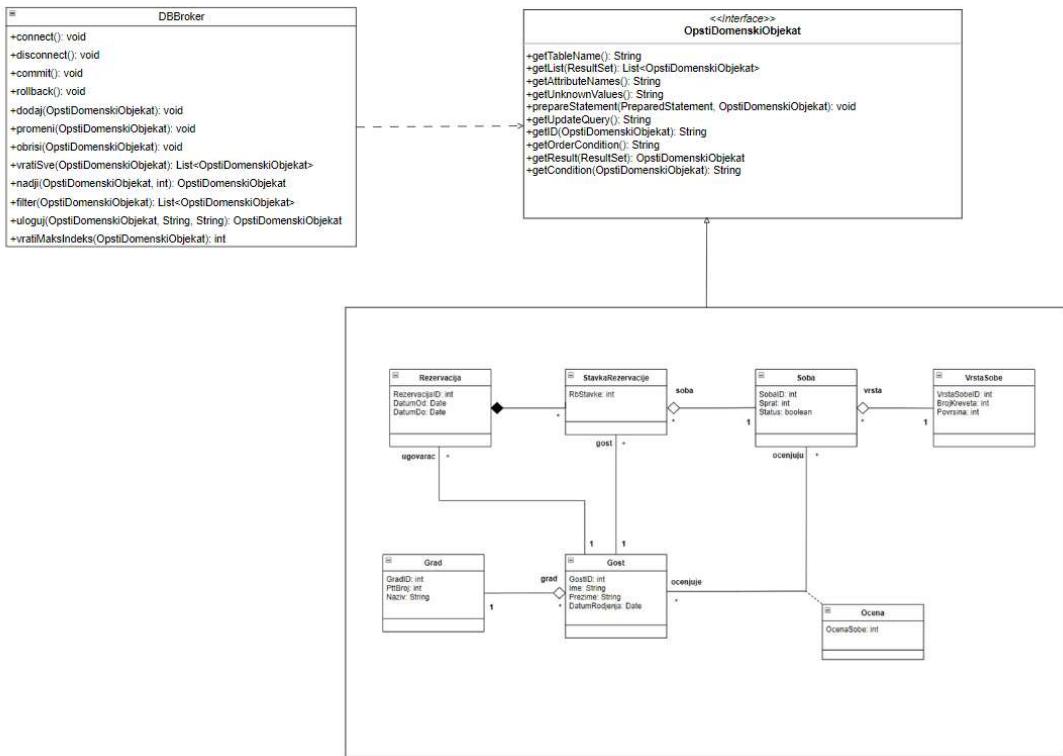
3.2.3. Брокер базе података

Класа DBBroker представља перзистентан оквир који посредује у свим операцијама над базом података и реализације следеће методе:

- public void connect();
- public void disconnect();
- public void commit();
- public void rollback();
- public void dodaj(OpstiDomenskiObjekat param);
- public void promeni(OpstiDomenskiObjekat param);
- public void obrisi(OpstiDomenskiObjekat param);
- public List< OpstiDomenskiObjekat > vratiSve(OpstiDomenskiObjekat param);
- public OpstiDomenskiObjekat nadji(OpstiDomenskiObjekat entity, int id);
- public List<OpstiDomenskiObjekat> filter(OpstiDomenskiObjekat entity);
- public OpstiDomenskiObjekat uloguj(OpstiDomenskiObjekat entity, String username, String password);
- public int vratiMaksIndeks(OpstiDomenskiObjekat entity).

Све методе класе DBBroker су пројектоване као генеричке, што значи да могу да прихватају различите доменске објекте преко параметара. Ово је остварено дефинисањем интерфејса OpstiDomenskiObjekat кога имплементирају све доменске класе.

- public String getTableName();
- public List<OpstiDomenskiObjekat> getList(ResultSet resultSet);
- public String getAttributeNames();
- public String getUnknownValues();
- public void prepareStatement(PreparedStatement ps, OpstiDomenskiObjekat entity);
- public String getUpdateQuery();
- public String getID(OpstiDomenskiObjekat entity);
- public String getOrderCondition();
- public OpstiDomenskiObjekat getResult(ResultSet resultSet);
- public String getCondition(OpstiDomenskiObjekat entity).



3.3. Пројектовање складишта података

На основу доменских класа софтвера пројектоване су табеле (складишта података) релационог система за управљање базом података. Систем за управљање базом података који је коришћен у студијском примеру је MySQL.

Табела Gost

Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update	Comment	Virtuality	Expression	Check Constraint
id	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	(none)	<input type="checkbox"/>		
ime	varchar	255		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	(none)	<input type="checkbox"/>		
prezime	varchar	255		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	(none)	<input type="checkbox"/>		
datumRodjenja	date			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	(none)	<input type="checkbox"/>		
gradID	int	11		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	(none)	<input type="checkbox"/>		
				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>		

Табела Grad

Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update	Comment	Virtuality	Expression	Check Constraint
id	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
pttBroj	int	11		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
naziv	varchar	255		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				<input type="checkbox"/>

Табела Осена

Табела Rezervacija

Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update	Comment	Virtuality	Expression	Check Constraint
<input type="checkbox"/> id	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/> datumOd	date			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/> datumDo	date			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/> gestID	int	11		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				

Табела Soba

Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update	Comment	Virtuality	Expression	Check Constraint
id	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)	<input type="button" value="▼"/>	
sprat	int	11		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)	<input type="button" value="▼"/>	
status	tinyint	1		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)	<input type="button" value="▼"/>	
vrstaSobeID	int	11		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)	<input type="button" value="▼"/>	
				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="button" value="▼"/>	<input type="button" value="▼"/>	

Табела StavkaRezervacije

Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update	Comment	Virtuality	Expression	Check Constraint
rezervacijaID	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)	<input type="button" value="▼"/>	
rbStavke	int	11		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)	<input type="button" value="▼"/>	
gostID	int	11		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)	<input type="button" value="▼"/>	
sobaID	int	11		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)	<input type="button" value="▼"/>	
				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="button" value="▼"/>	<input type="button" value="▼"/>	

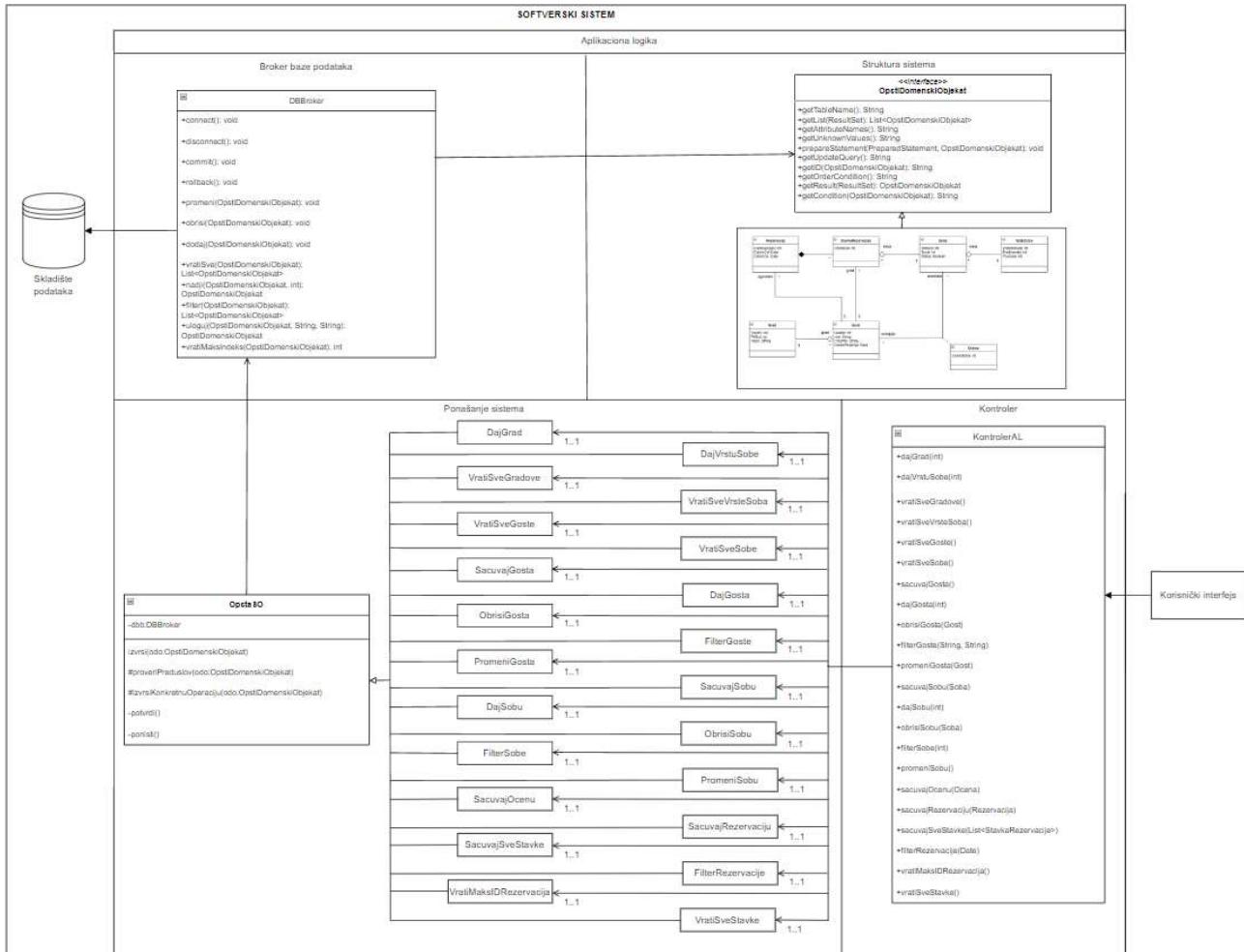
Табела VrstaSobe

Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update	Comment	Virtuality	Expression	Check Constraint
id	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)	<input type="button" value="▼"/>	
brojKreveta	int	11		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)	<input type="button" value="▼"/>	
povrsina	int	11		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)	<input type="button" value="▼"/>	
				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="button" value="▼"/>	<input type="button" value="▼"/>	

Додата је и табела Korisnik са креденцијалима за почетно логовање на систем.

Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update	Comment	Virtuality	Expression	Check Constraint
id	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)	<input type="button" value="▼"/>	
ime	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)	<input type="button" value="▼"/>	
prezime	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)	<input type="button" value="▼"/>	
username	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)	<input type="button" value="▼"/>	
password	varchar	255		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)	<input type="button" value="▼"/>	
				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="button" value="▼"/>	<input type="button" value="▼"/>	

На основу претходних целина, може се саставити цела архитектура софтверског система за праћење рада хотелског смештаја.



4. Имплементација

Софтверски систем је развијан у програмском језику “Java”. Систем је пројектован као клијент-сервер апликација. Као систем за управљање базом података коришћен је MySQL, док је развојно окружење “NetBeans IDE 14”. На основу архитектуре софтверског система добијене су следеће софтверске класе:

- HotelPSKlijent
 - komunikacija/Komunikacija
 - main/MainKlijent
 - view/forme/FrmGost
 - view/forme/FrmLogin
 - view/forme/FrmMain
 - view/forme/FrmPretragaGostiju
 - view/forme/FrmPretragaRezervacija
 - view/forme/FrmPretragaSoba
 - view/forme/FrmRezervacija
 - view/forme/FrmSoba
 - view/forme/komponenta/tabele/GostTableModel
 - view/forme/komponenta/tabele/RezervacijaTableModel
 - view/forme/komponenta/tabele/SobaTableModel
 - view/forme/komponenta/tabele/StavkaRezervacijeTableModel
 - view/forme/util/FormMode
 - view/konstante/Konstante
 - view/kontroler/GostKontroler
 - view/kontroler/LoginKontroler
 - view/kontroler/MainKontroler
 - view/kontroler/PretragaGostijuKontroler
 - view/kontroler/PretragaRezervacijaKontroler
 - view/kontroler/PretragaSobaKontroler
 - view/kontroler/RezervacijaKontroler
 - view/kontroler/SobaKontroler
 - view/koordinator/MainKoordinator
- HotelPSServer
 - forme/FrmMain
 - forme/FrmPodesavanja
 - forme/kontroler/MainKontroler
 - forme/kontroler/PodesavanjaKontroler
 - forme/koordinator/MainKoordinator
 - kontroler/Kontroler
 - main/MainServer
 - repository/Repository
 - repository/db/DbConnectionFactory
 - repository/db/DbRepository
 - repository/db/impl/DBBroker
 - server/ServerNit
 - server/Settings
 - server/thread/ProcessClientRequests
 - so/OpstaSO
 - so/gost/IzmeniGosta
 - so/gost/NadjiGoste
 - so/gost/ObrisniGosta
 - so/gost/UcitajGosta
 - so/gost/UcitajListuGostiju
 - so/gost/ZapamtiGosta

- so/grad/UcitajGrad
 - so/grad/UcitajListuGradova
 - so/korisnik/UlogujKorisnika
 - so/ocena/ZapamtiOcenuSobe
 - so/rezervacija/NadjiRezervacije
 - so/rezervacija/UcitajListuRezervacija
 - so/rezervacija/UcitajMaksimalniID
 - so/rezervacija/UcitajRezervaciju
 - so/rezervacija/ZapamtiRezervaciju
 - so/soba/IzmeniSobu
 - so/soba/NadjiSobe
 - so/soba/ObrisniSobu
 - so/soba/UcitajListuSoba
 - so/soba/UcitajSobu
 - so/soba/ZapamtiSobu
 - so/stavkaRezervacije/UcitajListuStavki
 - so/stavkaRezervacije/ZapamtiStavkuRezervacije
 - so/vrstaSobe/UcitajListuVrstaSoba
 - so/vrstaSobe/UcitajVrstuSobe
- HotelPSZajednicki
 - domen/Gost
 - domen/Grad
 - domen/Korisnik
 - domen/Ocena
 - domen/OpstiDomenskiObjekat
 - domen/Rezervacija
 - domen/Soba
 - domen/StavkaRezervacije
 - domen/VrstaSobe
 - komunikacija/Operation
 - komunikacija/Receiver
 - komunikacija/Request
 - komunikacija/Response
 - komunikacija/Sender.

5. Тестирање

Сваки од имплементираних случајева коришћења је тестиран. Приликом тестирања сваког случаја коришћења, поред унетих правилних података, уношени су и неправилни подаци да би се утврдило какав ће бити резултат извршења. Након фазе тестирања, софтвер је спреман за коришћење од стране крајњег корисника.