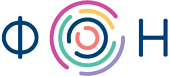
Универзитет у Београду

Факултет организационих наука



Семинарски рад

Назив предмета: Пројектовање софтвера, Катедра за софтверско инжењерство

тема:

Софтверски систем за здравствени систем у Јава развојном окружењу

Ментор: Проф.др. Синиша С. Влајић Студент: Жељана Грујић 2018/0045

Београд,Фебруар 2022.

Садржај

[1.Фаза прикупљања корисничких захтева 4](#_Toc96362421)

[1.1 Вербални опис 4](#_Toc96362422)

[1.2 Случајеви коришћења 5](#_Toc96362423)

[СК1: Случај коришћења- Пријављивање доктора 6](#_Toc96362424)

[СК2: Случај коришћења- Унос новог пацијента 7](#_Toc96362425)

[СК3: Случај коришћења- Претрага пацијента 8](#_Toc96362426)

[СК4: Случај коришћења- Измена података пацијента 9](#_Toc96362427)

[СК5: Случај коришћења- Брисање пацијента 10](#_Toc96362428)

[СК6: Случај коришћења- Унос термина (сложен СК) 11](#_Toc96362429)

[СК7: Случај коришћења- Претрага термина 12](#_Toc96362430)

[СК8: Случај коришћења- Измена термина (сложени СК) 13](#_Toc96362431)

[СК9: Случај коришћења- Отказивање термина 14](#_Toc96362432)

[СК10: Случај коришћења- Регистрација новог доктора 15](#_Toc96362433)

[2. Фаза анализе 16](#_Toc96362434)

[2.1 Понашање софтверског система 16](#_Toc96362435)

[ДС1: Дијаграм секвенци случаја коришћења – Пријављивање доктора 17](#_Toc96362436)

[ДС2: Дијаграм секвенци случаја коришћења –Унос новог пацијента 19](#_Toc96362437)

[ДС3: Дијаграм секвенци случаја коришћења –Претрага пацијента 20](#_Toc96362438)

[ДС4: Дијаграм секвенци случаја коришћења – Измена података пацијента 22](#_Toc96362439)

[ДС5: Дијаграм секвенци случаја коришћења – Брисање пацијента 25](#_Toc96362440)

[ДС6: Дијаграм секвенци случаја коришћења – Унос термина 28](#_Toc96362441)

[ДС7: Дијаграм секвенци случаја коришћења –Претрага термина 30](#_Toc96362442)

[ДС8: Дијаграм секвенци случаја коришћења – Измена термина 32](#_Toc96362443)

[ДС9: Дијаграм секвенци случаја коришћења – Отказивање термина 35](#_Toc96362444)

[ДС10: Дијаграм секвенци случаја коришћења –Регистрација новог доктора 38](#_Toc96362445)

[Резултирајуће системске операције 39](#_Toc96362446)

[2.2 Дефинисање уговора о системским операцијама 40](#_Toc96362447)

[Уговор UG1: Login 40](#_Toc96362448)

[Уговор UG2: kreirajPacijenta 40](#_Toc96362449)

[Уговор UG3: ucitajListuPacijenta 40](#_Toc96362450)

[Уговор UG4: nadjiPacijente 40](#_Toc96362451)

[Уговор UG5: ucitajPacijenta 40](#_Toc96362452)

[Уговор UG6: izmeniPacijenta 41](#_Toc96362453)

[Уговор UG7: obrisiPacijenta 41](#_Toc96362454)

[Уговор UG8: ucitajListuUsluga 41](#_Toc96362455)

[Уговор UG9: kreirajTermin 41](#_Toc96362456)

[Уговор UG10: ucitajListuTermina 41](#_Toc96362457)

[Уговор UG11: nadjiTermine 41](#_Toc96362458)

[Уговор UG12: ucitajTermin 41](#_Toc96362459)

[Уговор UG13: izmeniTermin 42](#_Toc96362460)

[Уговор UG14: obrisiTermin 42](#_Toc96362461)

[Уговор UG15: kreirajDoktora 42](#_Toc96362462)

[Структура софтверског система - Концептуални (доменски) модел 43](#_Toc96362463)

[Структура софтверског система- Релациони модел 44](#_Toc96362464)

[3.Фаза пројектовања 48](#_Toc96362465)

[3.1 Пројектовање корисничког интерфејса 49](#_Toc96362466)

[3.1.1 Пројектовање екранских форми 50](#_Toc96362467)

[СК1: Случај коришћења- Пријављивање доктора 50](#_Toc96362468)

[СК2: Случај коришћења- Унос новог пацијента 52](#_Toc96362469)

[СК3: Претрага пацијента 54](#_Toc96362470)

[СК4: Измена података пацијента 57](#_Toc96362471)

[СК5: Брисање пацијента 62](#_Toc96362472)

[СК6: Унос термина (сложен СК) 66](#_Toc96362473)

[СК7: Претрага термина 69](#_Toc96362474)

[СК8: Измена термина (сложени СК) 72](#_Toc96362475)

[СК9: Отказивање термина 76](#_Toc96362476)

[СК10: Регистрација новог доктора 80](#_Toc96362477)

[3.1.2 Пројектовање контролера корисничког интерфејса 82](#_Toc96362478)

[3.2 Пројектовање апликационе логике 82](#_Toc96362479)

[3.2.1 Конторлер апликационе логике 82](#_Toc96362480)

[3.2.2 Пословна логика 83](#_Toc96362481)

[Уговор UG15: kreirajDoktora 90](#_Toc96362482)

[3.2.3 Пројектовање складишта података 96](#_Toc96362483)

[4. Фаза имплементације 99](#_Toc96362484)

[5. Тестирање 101](#_Toc96362485)

[6. Закључак 101](#_Toc96362486)

[Коришћена литература 101](#_Toc96362487)

# 1.Фаза прикупљања корисничких захтева

## 1.1 Вербални опис

Потребно је направити софтверски систем, који ће моћи да се користи за потребе здравствених услуга.Систем треба да омогући пријаву администратора на систем (у овом случају, улога администратора припада доктору), затим евиденцију личних података пацијента (нпр. лични подаци, карактеристичне болести, алергијске реакције, примљене вакцине, преписане терапије), заказивање новог термина прегледа приликом којег је потребно назначити који пацијент врши заказиванје прегледа, као и ког датума и времена.Такође, за сваког пацијента је потребно водити евиденцију коришћене услуге .

Софтверски систем треба додатно да омогући администратору (доктор који има улогу актора), да поред креирања нових пацијената врши претрагу и измену података постојећих пацијената, претрагу по терминима и брисање заказаних термина прегледа, уколико пацијент одустане од заказаног термина.

## 1.2 Случајеви коришћења

Индентификовани случајеви коришћења:

1. [Пријављивање доктора](#_СК1:_Случај_коришћења-)
2. [Унос новог пацијента](#_СК2:_Случај_коришћења-)
3. [Претрага пацијента](#_СК3:_Претраживање_пацијента)
4. [Измена података пацијента](#_СК3:_Измена_података)
5. [Брисање пацијента](#_СК5:_Унос_заказаног)
6. [Унос термина](#_СК6:_Претрага_по) (сложен СК)
7. [Претрага термина](#_СК7:_Брисање_заказаног)
8. [Измена термина](#_СК8:_Унос_терапије) (сложен СК)
9. [Отказивање термина](#_СК9:_Претрага_преписаних)
10. Регистрација новог доктора



Слика 1. Случајеви коришћења

### СК1: Случај коришћења- Пријављивање доктора

**Назив СК**

Пријављивање **доктора**.

**Актори СК**

**Доктор**

**Учесници СК**

**Доктор и** систем (програм)

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

**Основни сценарио СК**

1. **Доктор уноси** податке за пријаву. (АПУСО)
2. **Доктор контролише** да ли је коректно унео податке за пријаву. (АНСО)
3. **Доктор позива** систем да пронађе **доктора** са задатим подацима.(АПСО)
4. Систем **претражује доктора**. (СО)
5. Систем **приказује** поруку „Успешно сте се пријавили на систем“. (ИА)

**Алтернативна сценарија**

* 1. Уколико систем није пронашао **доктора**, приказује **доктору** поруку

“Систем не може да пронађе **доктора** на основу унетих података”.(ИА)

* 1. Уколико **доктор** није успео да се пријави на систем,систем шаље **доктору** поруку “Неуспешна регистрација, молимо Вас покушајте поново”. (ИА)

### СК2: Случај коришћења- Унос новог пацијента

**Назив СК**

Унос новог пацијента.

**Актори СК**

**Доктор**

**Учесници СК**

**Доктор и** систем (програм)

**Предуслов**: Систем је укључен и **доктор** је улогован под својом шифром. **Доктор** поседује потребне информације о пацијенту. Систем приказује форму за рад са пацијентима.

**Основни сценарио СК**

1. **Доктор уноси** податке о пацијенту. (АПУСО)
2. **Доктор контролише** да ли је коректно унео податке о пацијенту. (АНСО)
3. **Доктор позива**систем да креира новог пацијента.(АПСО)
4. Систем **креира** новог пацијента. (СО)
5. Систем **приказује** поруку „Систем је успешно креирао новог пацијента“. (ИА)

**Алтернативна сценарија**

* 1. Уколико систем не може да креира новог пацијента, он приказује **доктору** поруку “Систем не може да креира новог пацијента”.(ИА)

### СК3: Случај коришћења- Претрага пацијента

**Назив СК**

Претрага пацијента

**Актори СК**

**Доктор**

**Учесници СК**

**Доктор и** систем (програм)

**Предуслов**: Систем је укључен и **доктору** je улогован под својом шифром. Систем приказује форму за претраживање пацијената. Учитана је листа пацијената.

**Основни сценарио СК**

* 1. **Доктор уноси** вредност по којој претражује пацијенте.(АПУСО)
  2. **Доктор позива** систем да нађе пацијенте по задатој вредности.(АПСО)
  3. Систем **тражи** пацијенте по задатој вредности.(СО)
  4. Систем **приказује доктору** податке о пацијентима и поруку:„Систем је нашао пацијенте по задатој вредности.”.(ИА)
  5. **Доктор** **бира** пацијента којег жели да му систем прикаже. (АПУСО)
  6. **Доктор** **позива** систем да учита пацијента. (АПСО)
  7. Систем **учитава** пацијента. (СО)
  8. Систем **приказује** **доктору** податке о пацијенту и поруку:

“Систем је учитао пацијента“. (ИА)

**Алтернативна сценарија**

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

8.1. Уколико систем не може да учита пацијента он приказује **доктору** поруку:

“Систем не може да учита клијента.”.(ИА)

### СК4: Случај коришћења- Измена података пацијента

**Назив СК**

Измена података пацијента

**Актори СК**

**Доктор**

**Учесници СК**

**Доктор** и систем (програм)

**Предуслов**: Систем је укључен и **доктор** je улогован под својом шифром. Систем приказује форму за рад са пацијентима. Учитана је листа пацијената.

**Основни сценарио СК**

* + - * 1. **Доктор уноси** вредност по којој претражује пацијенте.(АПУСО)
        2. **Доктор** **контролише** да ли је коректно унео податке за претрагу пацијената. (АНСО
        3. **Доктор позива** систем да пронађе пацијенте по задатој вредности.(АПСО)
        4. Систем **тражи** пацијенте по задатој вредности.(СО)
        5. Систем **приказује** податке о траженим пацијентима уз поруку: „Подаци о траженим пацијентима су успешно учитани”.(ИА)
        6. **Доктор** **бира** пацијента којег жели да измени. (АПУСО)
        7. **Доктор** **позива** систем да пронађе изабраног пацијента. (АПСО)
        8. Систем **проналази** изабраногпацијента. (СО)
        9. Систем **приказује** податке о траженом пацијенту уз поруку: „ Систем **је пронашао** изабраног пацијента”.(ИА)
        10. **Доктор мења** податке о пацијенту.(АПУСО)
        11. **Доктор контролише** да ли је коректно унео податке о пацијенту.(АНСО)
        12. **Доктор позива** систем да запамти податке о пацијену.(АПСО)
        13. Систем **памти** податке о пацијенту.(СО)
        14. Систем **приказује доктору** измењеног пацијента уз поруку: „Систем је изменио податке пацијента”.(ИА)

**Алтернативна сценарија**

5.1. Уколико систем не може да пронађе пацијенте, приказује следећу поруку **доктору**: „Систем не може да пронађе пацијенте на основу унетих података”. Прекида се извршење сценарија. (ИА)

9.1 Уколико систем не може да прикаже податке о пацијенту, приказује следећу поруку **доктору**: „Систем не може да пронађе пацијента“. Прекида се извршење сценарија. (ИА)

14.1. Уколико систем не може да измени пацијента, он приказује следећу поруку

**доктору**: „Систем не може да измени пацијента”. (ИА)

### СК5: Случај коришћења- Брисање пацијента

**Назив СК**

Брисање пацијента

**Актори СК**

**Доктор**

**Учесници СК**

**Доктор** и систем (програм)

**Предуслов**: Систем је укључен и **доктор** je улогован под својом шифром. Систем приказује форму за рад са пацијентима. Учитана је листа пацијената.

**Основни сценарио СК**

1. **Доктор** **уноси** вредност по којој претражује пацијенте. (АПУСО)
2. **Доктор** **контролише** да ли је коректно унео податке за претрагу пацијената. (АНСО)
3. **Доктор** **позива** систем да пронађе пацијенте по задатој вредности. (АПСО)
4. Систем **тражи** пацијенте по задатој вредности. (СО)
5. Систем **приказује** пацијенте **доктор** и поруку: „Систем је пронашао пацијенте по задатој вредности“. (ИА)
6. **Доктор** **бира** пацијента кога жели да избрише. (АПУСО)
7. **Доктор** **позива** систем да нађе изабраног пацијента. (АПСО)
8. Систем **проналази** изабраног пацијента. (СО)
9. Систем **приказује** пацијента и поруку: „Систем је пронашао пацијента“. (ИА)
10. **Доктор** **позива** систем да обрише пацијента. (АПСО)
11. Систем **брише** пацијента. (СО)
12. Систем **приказује** **доктор** поруку: „Систем је обрисао пацијента“. (ИА)

**Алтернативна сценарија:**

5.1 Уколико систем не може да нађе пацијенте, он приказује **доктор** поруку: “Систем не може да пронађе пацијенте по задатој вредности”. Прекида се извршење сценарија. (ИА)

9.1 Уколико систем не може да нађе пацијенте, он приказује **доктору** поруку: “Систем не може да учита пацијента.”. Прекида се извршење сценарија. (ИА)

12.1 Уколико систем не може да обрише пацијента он приказује **доктору** поруку: „Систем не може да обрише пацијента “. (ИА)

### СК6: Случај коришћења- Унос термина (сложен СК)

**Назив СК**

Унос термина

**Актори СК**

**Доктор**

**Учесници СК**

**Доктор** и систем (програм)

**Предуслов**: Систем је укључен и **доктор** је улогован под својом шифром.Систем приказује форму за рад са терминима. Учитане су листа пацијената и листа услуга.

**Основни сценарио СК**

* + - * 1. **Доктор** **уноси** податке о термину.(АПУСО)
        2. **Доктор** **контролише** да ли је коректно унео податке о термину.(АНСО)
        3. **Доктор позива** систем да креира термин.(АПСО)
        4. Систем **креира** термин.(СО)
        5. Систем **приказује** **доктору** креирани термин и поруку:„ Систем је креирао термин.”. (ИА)

**Алтернативна сценарија:**

Уколико систем не може да креира рачунар он приказује **доктору** поруку: “Систем не може да креира термин“.(ИА)

### СК7: Случај коришћења- Претрага термина

**Назив СК**

Претрага термин

**Актори СК**

**Доктор**

**Учесници СК**

**Доктор** и систем (програм)

**Предуслов**:Систем је укључен и **доктор** je улогован под својом шифром. Систем приказује форму за претраживање рачунара. Учитана је листа термина.

**Основни сценарио СК**

1. **Доктор** **уноси** вредност по којој претражује термине. (АПУСО)
2. **Доктор позива** систем да нађе термине по задатој вредности. (АПСО)
3. Систем **тражи** термине по задатој вредности. (СО)
4. Систем **приказује** **доктору** податке о терминима и поруку:„Систем је нашао термине по задатој вредности.”. (ИА)
5. **Доктор** **бир** термин који жели да му систем прикаже. (АПУСО)
6. **Доктор** **позива** систем да учита термин. (АПСО)
7. Систем **учитава** термин. (СО)
8. Систем **приказује** **доктору** податке о термину и поруку:“Систем је учитао термин.“. (ИА)

**Алтернативна сценарија:**

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

8.1 Уколико систем не може да учита термин приказује **доктор** поруку: “Систем не може да учита термин.”.(ИА)

### СК8: Случај коришћења- Измена термина (сложени СК)

**Назив СК**

Измена термина

**Актори СК**

**Доктор**

**Учесници СК**

**Доктор** и систем (програм)

**Предуслов:** Систем је укључен и **доктор** je улогован под својом шифром. Систем приказује форму за рад са термина. Учитане су листе термина, пацијената и услуга.

**Основни сценарио СК**

* + - * 1. **Доктор** **уноси** вредност по којој претражује термине. (АПУСО)
        2. **Доктор контролише** да ли је коректно унео податке за претрагу термина. (АНСО)
        3. **Доктор позива** систем да нађе термине по задатој вредности. (АПСО)
        4. Систем **тражи** термине по задатој вредности. (СО)
        5. Систем **приказује** **доктору** термине и поруку: „Систем је нашао термине по задатој вредности.“. (ИА)
        6. **Доктор** **бира** термин који жели да измени. (АПУСО)
        7. **Доктор позива** систем да нађе изабрани термин. (АПСО)
        8. Систем **проналази** изабрани термин. (СО)
        9. Систем **приказује** **доктору** термин и поруку: „Систем је учитао термин “. (ИА)
        10. **Доктор** **мења** податке о термину. (АПУСО)
        11. **Доктор контролише** да ли је коректно унео податке о термину. (АНСО)
        12. **Доктор** **позива** систем да запамти податке о термину. (АПСО)
        13. Систем **памти** податке о термину. (СО)
        14. Систем **приказује доктору** поруку: „Систем је изменио термин.“. (ИА)

**Алтернатинвна сценарија:**

5.1 Уколико систем не може да пронађе термине он приказује **доктору** поруку: “Систем не може да пронађе термине по задатој вредности”. Прекида се извршење сценарија. (ИА)

9.1. Уколико систем не може да прикаже податке о термину, приказује следећу поруку **доктору**: „ Систем не може учита термин.”. Прекида се извршење сценарија. (ИА)

14.1 Уколико систем не може да измени податке о термину он приказује **доктору** поруку: „Систем не може да измени термин“. (ИА)

### СК9: Случај коришћења- Отказивање термина

**Назив СК**

Брисање термина

**Актори СК**

**Доктор**

**Учесници СК**

**Доктор** и систем (програм)

**Предуслов**: Систем је укључен и **доктор** je улогован под својом шифром.Систем приказује форму за рад са терминима. Учитана је листа термина.

**Основни сценарио СК**

1. **Доктор** **уноси** вредност по којој претражује термине. (АПУСО)
2. **Доктор контролише** да ли је коректно унео податке за претрагу термина. (АНСО)
3. **Доктор позива** систем да нађе термине по задатој вредности. (АПСО)
4. Систем **тражи** термине по задатој вредности. (СО)
5. Систем **приказује** термине **доктору** и поруку: „Систем је нашао термине по задатој вредности“. (ИА)
6. **Доктор бира** термин који жели да избрише. (АПУСО)
7. **Доктор позива** систем да нађе изабрани термин. (АПСО)
8. Систем **проналази** изабрани термин. (СО)
9. Систем **приказује** **доктору** термин и поруку: „Систем је учитао термин“. (ИА)
10. **Доктор** позива систем да обрише термин. (АПСО)
11. Систем **брише** термин. (СО)
12. Систем **приказује** **доктору** поруку: „Систем је обрисао термин“. (ИА)

**Алтернативна сценарија:**

5.1 Уколико систем не може да нађе термине, он приказује **доктору** поруку: “Систем не може да нађе термине по задатој вредности”. Прекида се извршење сценарија. (ИА)

9.1 Уколико систем не може да прикаже податке о термину, он приказује **доктору** поруку: “Систем не може да учита термин.”. Прекида се извршење сценарија. (ИА)

12.1 Уколико систем не може да обрише термин он приказује **доктору** поруку: „Систем не може да обрише термин.“. (ИА)

### СК10: Случај коришћења- Регистрација новог доктора

**Назив СК**

Регистрација новог доктора.

**Актори СК**

**Доктор**

**Учесници СК**

**Доктор и** систем (програм)

**Предуслов**: Систем је укључен и **доктор** је улогован под својом шифром. **Доктор** поседује потребне информације о новом доктору. Систем приказује форму за рад са докторима.

**Основни сценарио СК**

1. **Доктор уноси** податке о новом доктору. (АПУСО)
2. **Доктор контролише** да ли је коректно унео податке о доктору. (АНСО)
3. **Доктор позива** систем да креира новог доктора.(АПСО)
4. Систем **креира** новог доктора. (СО)
5. Систем **приказује** поруку „Систем је успешно креирао новог доктора“. (ИА)

**Алтернативна сценарија**

5.1 Уколико систем не може да креира новог доктора, он приказује **доктору** поруку “Систем не може да креира новог доктора”.(ИА)

# 2. Фаза анализе

У фази анализе описујемо логичку структуру и понашање софтвера. Заправо, резултат фазе анализе јесте *пословна логика* софтверског система. Понашање софтверског система се описује помоћу *дијаграма секвенци* и *системских операција*, док се структура система описује помоћу *концептуалног* и *релационог модела*.

## 2.1 Понашање софтверског система

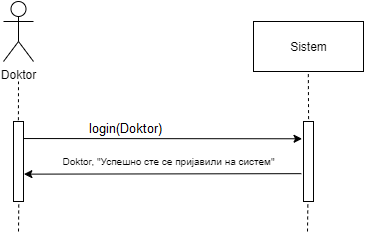
Понашање софтверског система приказујемо путем системских дијаграма секвенци, где ћемо за сваки *случај коришћења*, уочен у *фази прикупљања захтева*, дати дијаграм секвенци. Системски дијаграм секвенци треба да прикаже интеракцију између актора и система, путем активности у одређеном редоследу.

\*На дијаграму секвенци, актор не комуницира са системом директо, већ преко посредника (форма). Приликом цртања секвенцних дијаграма, изоставићемо приказ форме, али се њено присуство у комуникацији подразумева.

### ДС1: Дијаграм секвенци случаја коришћења – Пријављивање доктора

**Основни сценарио СК**

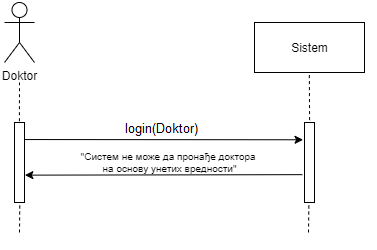
1. **Доктор позива** систем да пронађе **доктора** са задатим подацима.(АПСО)
2. Систем **приказује** поруку „Успешно сте се пријавили на систем“. (ИА)



Дијаграм : ДС Пријављивање доктора (основни сценарио)

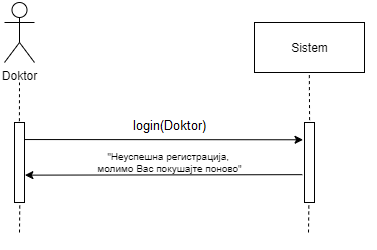
**Алтернативна сценарија**

* 1. Уколико систем није пронашао доктора, приказује доктору поруку “Систем не може да нађе доктора на основу унетих вредности”. (ИА)



Дијаграм 2: ДС Пријављивање доктора (алтернативни сценарио 1)

2.2 Уколико доктор није успео да се пријави на систем,систем шаље доктору поруку: .„Неуспешна регистрација, молимо Вас покушајте поново.“. (ИА)



Дијаграм 3: ДС Пријављивање доктора (алтернативни сценарио 2)

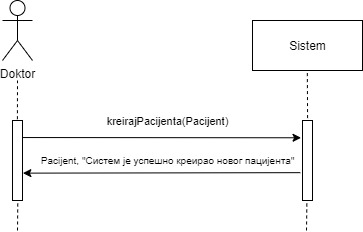
Са наведених секвенцних дијаграма уочава се једна системска операција коју треба пројектовати:

1. *signal* **login(Doktor)**

### ДС2: Дијаграм секвенци случаја коришћења –Унос новог пацијента

**Основни сценарио СК**

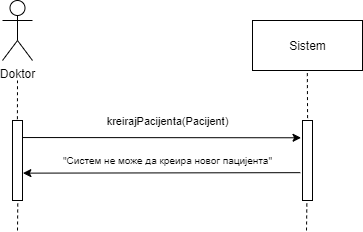
* + - * 1. **Доктор позива** систем да креира новог пацијента.(АПСО)
        2. **Систем приказује доктору** креираног пацијента и поруку:„Систем је успешно креирао новог пацијента.”. (ИА)



Дијаграм 4: ДС Унос новог пацијента (основни сценарио)

**Алтернативна сценарија СК**

Уколико систем не може да креира новог пацијента он приказује **доктору** поруку:„Систем не може да креира новог пацијента.”. (ИА)



Дијаграм 5: ДС Унос новог пацијента (алтернативни сценарио)

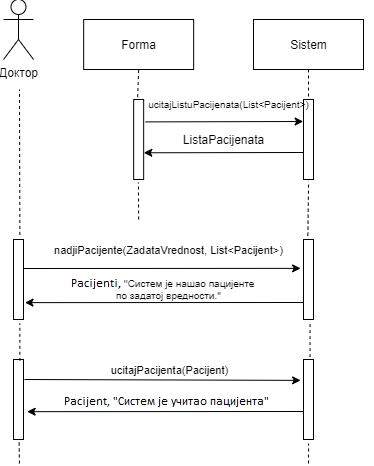
Са наведених секвенцних дијаграма уочава се једна системска операција коју треба пројектовати:

* + - * 1. *signal* **kreirajPacijenta(Pacijent)**

### ДС3: Дијаграм секвенци случаја коришћења –Претрага пацијента

**Основни сценарио СК**

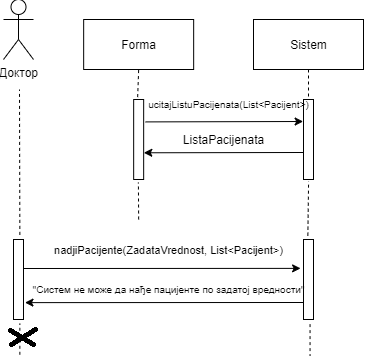
1. Форма **позива** систем да врати листу пацијената. (АПСО)
2. Систем **враћа** форми листу пацијената. (ИА)
3. **Доктор** **позива** систем да нађе пацијенате по задатој вредности.(АПСО)
4. Систем **приказује доктору** податке о пацијентима и поруку:„Систем је нашао пацијенте по задатој вредности.”.(ИА)
5. **Доктор позива** систем да учита пацијента. (АПСО)
6. Систем **приказује доктору** податке о пацијенту и поруку: “Систем је учитао клијента“. (ИА)



Дијаграм 6: ДС Претрага пацијента (основни сценарио)

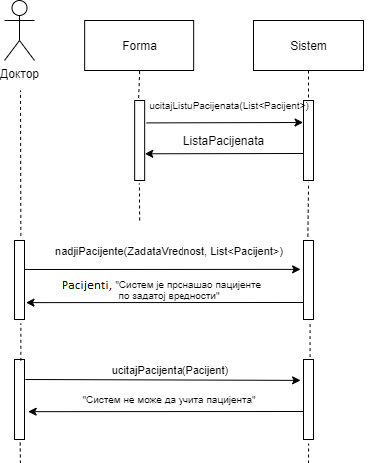
**Алтернативна сценарија**

* 1. Уколико систем не може да нађе пацијенте по задатој вредности, он приказује доктору поруку: “Систем не може да нађе пацијенте по задатој вредности.” Прекида се извршење сценарија. (ИА)



Дијаграм 7: ДС Претраживање пацијената (алтернативни сценарио 1)

* 1. Уколико систем не може да учита пацијенте он приказује доктору поруку: “Систем не може да учита клијента.”.(ИА)



Дијаграм 8 : ДС Претраживање пацијената (алтернативни сценарио 2)

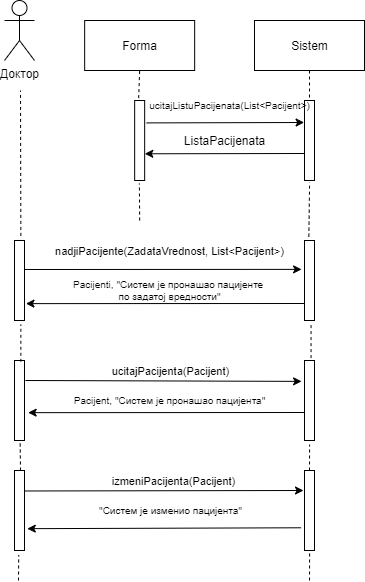
Са наведених секвенцних дијаграма уочавају се три системске операције које треба пројектовати:

* + - * 1. *signal* **ucitajListuPacijenata(List<Pacijent>)**
        2. *signal* **nadjiPacijente(String zadataVrednost, List<Pacijent>)**
        3. *signal* **ucitajPacijenta(Pacijent)**

### ДС4: Дијаграм секвенци случаја коришћења – Измена података пацијента

**Основни сценарио СК**

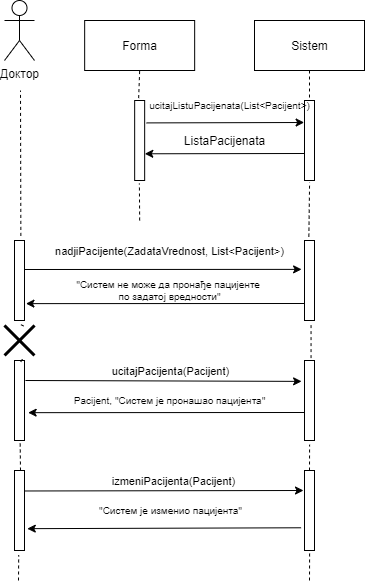
* 1. Форма **позива** систем да врати листу пацијената. (АПСО)
  2. Систем **враћа** форми листу пацијената. (ИА)
  3. **Доктор** **позива** систем да нађе пацијенте по задатој вредности. (АПСО)
  4. Систем **приказује доктору** пацијенте и поруку: „Систем је нашао пацијенте по задатој вредности.“. (ИА)
  5. **Доктор** **позива** систем да нађе изабраног пацијента. (АПСО)
  6. Систем **приказује** пацијента и поруку: „Систем је учитао пацијента“. (ИА)
  7. **Доктор** **позива** систем да запамти податке о пацијенту. (АПСО)
  8. Систем **приказује** **доктору** поруку: „Систем је изменио пацијента“. (ИА)



Дијаграм 9: ДС Измена података пацијента (основни сценарио)

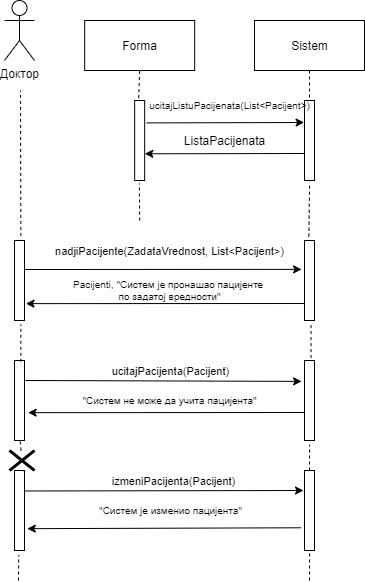
**Алтернативна сценарија СК**

* 1. Уколико систем не може да прикаже податке о пацијентима, приказује следећу поруку **доктору**:„Систем не може да пронађе пацијенте на основу унетих података”.Прекида се извршење сценарија. (ИА)



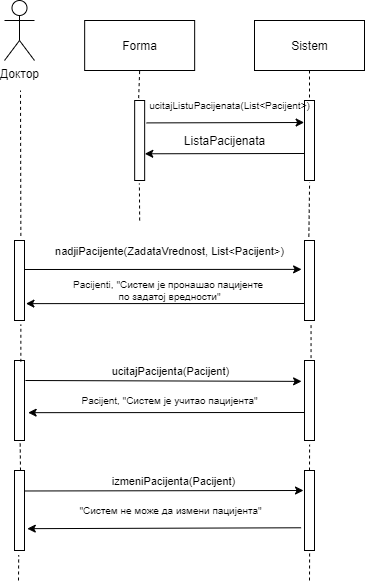
Дијаграм 10: ДС Измена података пацијента (алтернативни сценарио 1)

6.1. Уколико систем не може да прикаже податке о пацијенту, приказује следећу поруку докору: „Систем не може да учита пацијента.”. Прекида се извршење сценарија. (ИА)



Дијаграм 11 ДС: Измена података пацијента (алтернативни сценарио 2)

* 1. Уколико систем не може да измени податке пацијента, приказује следећу поруку **доктору**:„Систем не може да измени пацијента.”. (ИА)



Дијаграм 12: ДС Измена података пацијента (алтернативни сценарио 3)

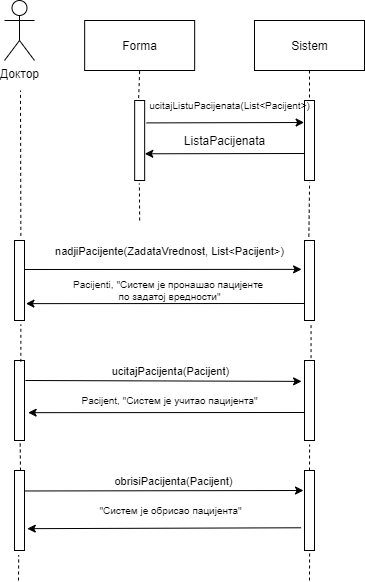
Са наведених секвенцних дијаграма уочавају се три системске операције које треба пројектовати:

* + - * 1. *signal* **ucitajListuPacijenata(List<Pacijent>)**
        2. *signal* **nadjiPacijente(String zadataVrednost, List<Pacijent>)**
        3. *signal* **ucitajPacijenta(Pacijent)**
        4. *signal* **izmeniPacijenta(Pacijent)**

### ДС5: Дијаграм секвенци случаја коришћења – Брисање пацијента

**Основни сценарио СК**

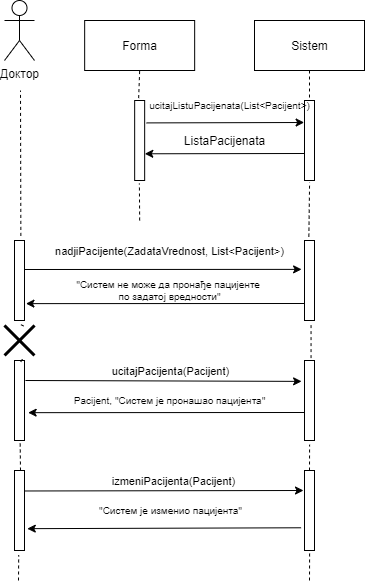
1. Форма **позива** систем да врати листу пацијената. (АПСО)
2. Систем **враћа** форми листу пацијената. (ИА)
3. **Доктор позива** систем да нађе пацијенте по задатој вредности. (АПСО)
4. Систем **приказује** клијенте **доктору** и поруку: „Систем је нашао пацијенте по задатој вредности“. (ИА)
5. **Доктор позива** систем да нађе изабраног пацијента. (АПСО)
6. Систем **приказује** пацијента и поруку: „Систем је учитао пацијента“. (ИА)
7. **Доктор** **позива** систем да обрише пацијента. (АПСО)
8. Систем **приказује** **доктору** поруку: „Систем је обрисао пацијента“. (ИА)



Дијаграм 13: ДС Брисање пацијента (основни сценарио)

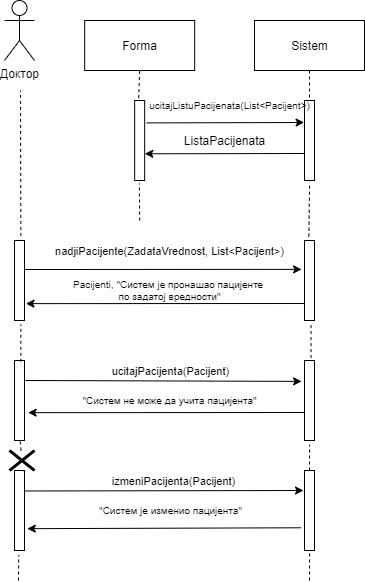
**Алтернативна сценарија СК**

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



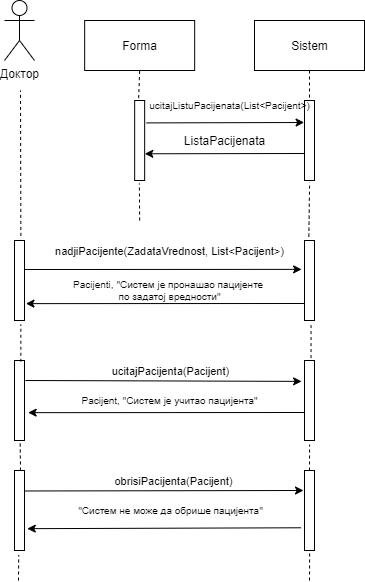
Дијаграм 14: ДС Брисање пацијента (алтернативни сценарио 1)

6.1. Уколико систем не може да прикаже податке о пацијенту, приказује следећу поруку докору: „Систем не може да учита пацијента.”. Прекида се извршење сценарија. (ИА)



Дијаграм 15 ДС: Брисање пацијента (алтернативни сценарио 2)

* 1. Уколико систем не може да обрише пацијента, приказује следећу поруку **доктору**:„Систем не може да измени пацијента.”. (ИА)



Дијаграм 16 ДС: Брисање пацијента (алтернативни сценарио 2)

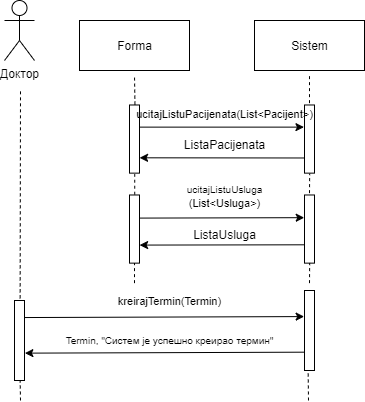
Са наведених секвенцних дијаграма уочавају се четири системска операција које треба пројектовати:

1. *signal* **ucitajListuPacijenata(List<Pacijent>)**
2. *signal* **nadjiPacijente(ZadataVrednost, List<Pacijent>)**
3. *signal* **ucitajPacijenta(Pacijent)**
4. *signal* **obrisiPacijenta(Pacijent)**

### ДС6: Дијаграм секвенци случаја коришћења – Унос термина

**Основни сценарио СК**

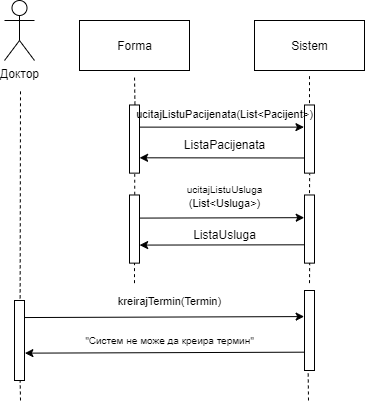
* + - * 1. Форма **позива** систем да врати листу пацијената. (АПСО)
        2. Систем **враћа** форми листу пацијената. (ИА)
        3. Форма **позива** систем да врати листу услуга. (АПСО)
        4. Систем **враћа** форми листу услуга
        5. **Доктор позива** систем да креира термин.(АПСО)
        6. Систем **приказује** **доктору** креирани термин и поруку: „Систем је креирао термин.”. (ИА)



Дијаграм 17: ДС Унос термина (основни сценарио)

**Алтернативна сценарија**

6.1 Уколико систем не може да креира термине прегледа, он приказује доктору поруку: “Систем не може да нађе зкреира термин прегледа.”(ИА)



Дијаграм 18: ДС Унос термина (алтернативни сценарио)

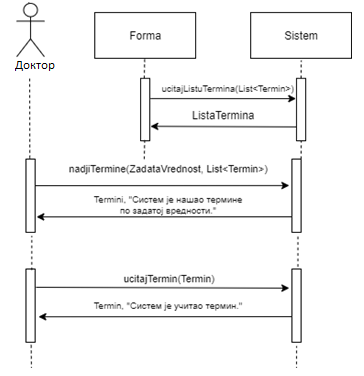
Са наведених секвенцних дијаграма уочавају се три системске операције које треба пројектовати:

* + - * 1. *signal* **ucitajListuPacijenata(List<Pacijent>)**
        2. *signal* **ucitajListuUsluga(List<Usluga>)**
        3. *signal* **kreirajTermin(Termin)**

### ДС7: Дијаграм секвенци случаја коришћења –Претрага термина

**Основни сценарио СК**

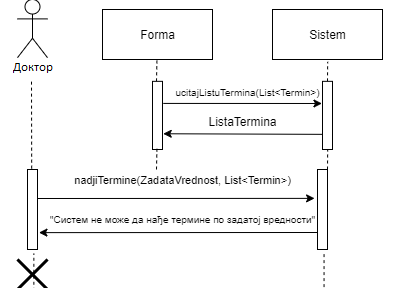
1. Форма **позива** систем да врати листу термина. (АПСО)
2. Систем **враћа** форми листу термина. (ИА)
3. **Доктор позива** систем да нађе термине по задатој вредности. (АПСО)
4. Систем **приказује доктору** податке о терминима и поруку: „Систем је нашао термине по задатој вредности.”. (ИА)
5. **Доктор** **позива** систем да учита термин. (АПСО)
6. Систем **приказује доктору** податке о термину и поруку: “Систем је учитао термин.“. (ИА)



Дијаграм 19: ДС Претрага заказаног термина прегледа ( основни сценарио)

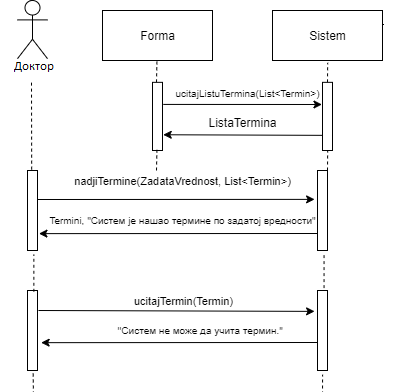
**Алтернативна сценарија**

* 1. Уколико систем не може да нађе термине прегледа, приказује следећу поруку **докотру**:„Систем не може да нађе термине прегледа по унетој вредности”. Прекида се извршење сценарија. (ИА)



Дијаграм 20: ДС Претрага заказаног термина прегледа ( алтернативни сценарио 1)

* 1. Уколико систем не може да учита термин прегледа он приказује доктору поруку “Систем не може да учита заказани термин прегледа.”. (ИА)



Дијаграм 21: ДС Претрага заказаног термина прегледа ( алтернативни сценарио 2)

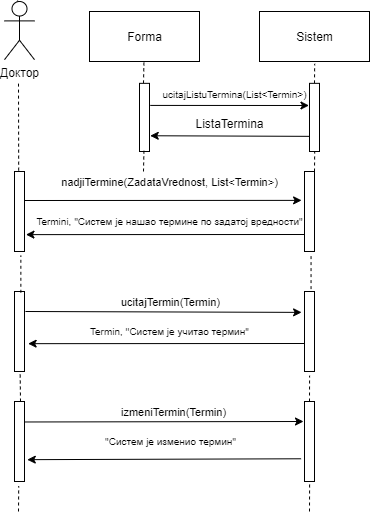
Са наведених секвенцних дијаграма уочавају се три системске операције које треба пројектовати:

* + - * 1. *signal* **ucitajListuTermina(List<Termin >)**
        2. *signal* **nadjiTermine(String zadataVrednost, List<Termin >)**
        3. *signal* **ucitajTermin(Termin)**

### ДС8: Дијаграм секвенци случаја коришћења – Измена термина

**Основни сценарио СК**

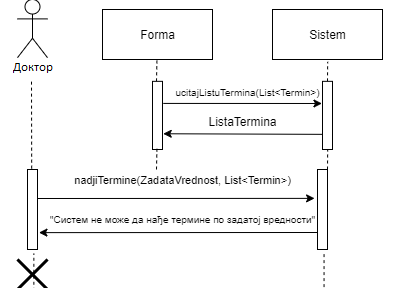
1. Форма **позива** систем да врати листу термина. (АПСО)
2. Систем **враћа** форми листу термина. (ИА)
3. **Доктор позива** систем да нађе термине по задатој вредности. (АПСО)
4. Систем **приказује доктору** термине и поруку: „Систем је нашао термине по задатој вредности.“. (ИА)
5. **Доктор позива** систем да нађе изабрани термин. (АПСО)
6. Систем **приказује** **доктору** термин и поруку: „Систем је учитао термин “. (ИА)
7. **Доктор** **позива** систем да запамти податке о термину. (АПСО)
8. Систем **приказује** **доктору** поруку: „Систем је изменио термин.“. (ИА)



Дијаграм 22: ДС Измена термина(основни сценарио)

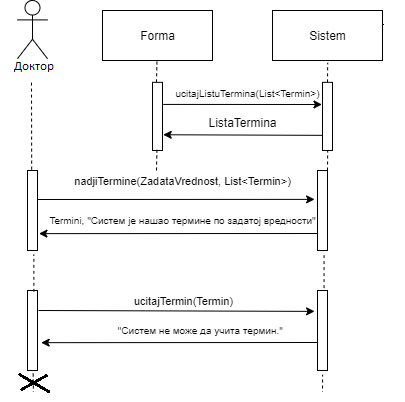
**Алтернативна сценарија**

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



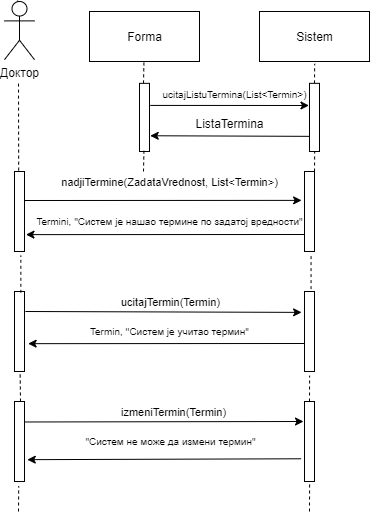
Дијаграм 23: ДС Унос термина (алтернативни сценарио 1)

6.1 Уколико систем не може прикаже податке о термину он приказује доктоу поруку “Систем не може да учита термин ”.Прекида се извршење сценарија (ИА)



Дијаграм 24: ДС Измена термина (алтернативни сценарио 2)

* 1. Уколико систем не може да измени податке о термину он приказује **доктору** поруку: „Систем не може да измени термин“. (ИА)



Дијаграм 25:ДС Измена термина (алтернативни сценарио 3)

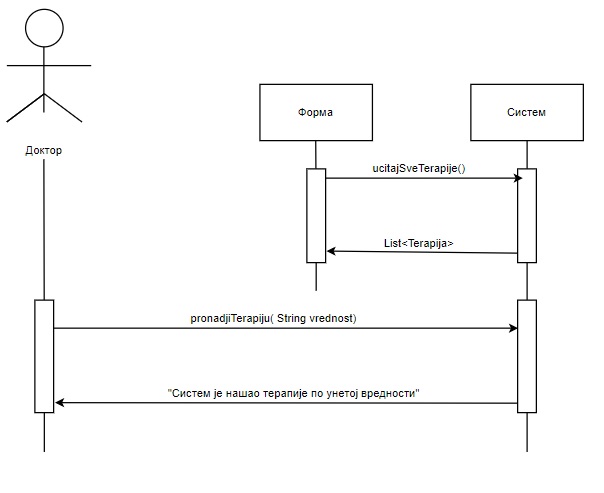
Са наведених секвенцних дијаграма уочавају се четири системске операције које треба пројектовати:

* + - * 1. *signal* **ucitajListuTermina(List<Termin >)**
        2. *signal* **nadjiTermine(String zadataVrednost, List<Termin >)**
        3. *signal* **ucitajTermin(Termin)**
        4. *signal* **izmeniTermin(Termin)**

### ДС9: Дијаграм секвенци случаја коришћења – Отказивање термина

**Основни сценарио СК**

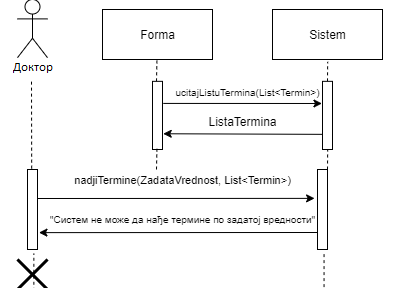
1. Форма **позива** систем да врати листу термина. (АПСО)
2. Систем **враћа** форми листу термина. (ИА)
3. **Доктор позива** систем да нађе термине по задатој вредности. (АПСО)
4. Систем **приказује** термине **доктору** и поруку: „Систем је нашао термине по задатој вредности“. (ИА)
5. **Доктор позива** систем да нађе изабрани термин. (АПСО)
6. Систем **приказује доктору** термин и поруку: „Систем је учитао термин“. (ИА)
7. **Доктор** позива систем да обрише термин. (АПСО)
8. Систем **приказује** **доктору** поруку: „Систем је обрисао термин“.(ИА)



Дијаграм 26: ДС Отказивање термина (основни сценарио)

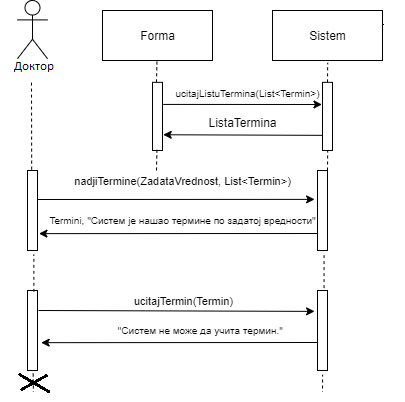
**Алтернативна сценарија**

4.1 . Уколико систем не може да нађе термине он приказује **доктору** поруку:„Систем не може да нађе термине по унетој вредности”.Прекида се извршење сценарија. (ИА)



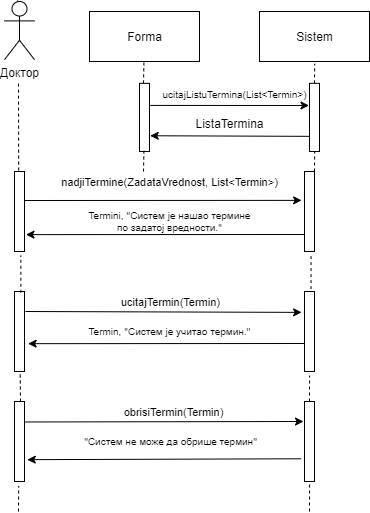
Дијаграм 27: ДС Отказивање термина (алтернативни сценарио 1)

6.1 . Уколико систем не може да нађе податке о термину он приказује **доктору** поруку:„Систем не може да учита термин”. Прекида се извршење сценарија.(ИА)



Дијаграм 28: ДС Измена термина (алтернативни сценарио 2)

8.1 Уколико систем не може да обрише термин он приказује **доктору** поруку: „Систем не може да обрише термин.“. (ИА)



Дијаграм 29: ДС Измена термина (алтернативни сценарио 3)

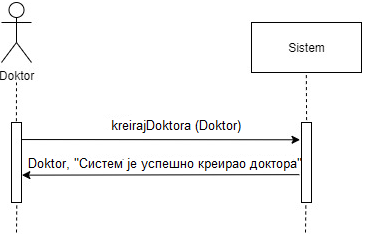
Са наведених секвенцних дијаграма уочавају се четири системске операције које треба пројектовати:

* + - * 1. *signal* **ucitajListuTermina(List<Termin >)**
        2. *signal* **nadjiTermine(String zadataVrednost, List<Termin >)**
        3. *signal* **ucitajTermin(Termin)**
        4. *signal* **obrisiTermin(Termin)**

### ДС10: Дијаграм секвенци случаја коришћења –Регистрација новог доктора

**Основни сценарио СК**

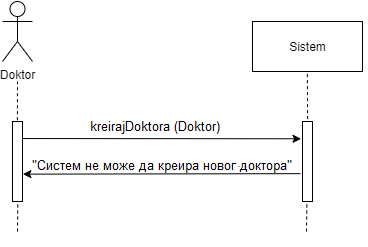
* + - * 1. **Доктор позива** систем да креира новог доктора.(АПСО)
        2. **Систем приказује доктору** доктора пацијента и поруку:„Систем је успешно креирао новог доктора.”. (ИА)



Дијаграм 30: ДС Унос новог доктора (основни сценарио)

**Алтернативна сценарија СК**

Уколико систем не може да креира новог доктора он приказује **доктору** поруку: „Систем не може да креира новог доктора.”. (ИА)



Дијаграм 31: ДС Унос новог доктора (алтернативни сценарио)

Са наведених секвенцних дијаграма уочава се једна системска операција коју треба пројектовати:

* + - * 1. *signal* **kreirajDoktora(Doktor)**

## Резултирајуће системске операције

Као резултат анализе сценарија добијамо укупно 14 системских операција које треба пројектовати:

1. *signal* **login(Doktor)**
2. *signal***kreirajPacijenta(Pacijent)**
3. *signal* **ucitajListuPacijenata(List<Pacijent>)**

*у пројекту ће бити имплементирана као getAllPacijent(List<Pacijent>)*

1. *signal* **nadjiPacijente(ZadataVrednost, List<Pacijent>)**
2. *signal* **ucitajPacijenta(Pacijent)**
3. *signal* **izmeniPacijenta(Pacijent)**
4. *signal* **obrisiPacijenta(Pacijent)**
5. *signal* **ucitajListuUsluga(List<Usluga>)**

*у пројекту ће бити имплементирана као getAllUsluga(List<Usluga>)*

1. *signal* **kreirajTermin(Termin)**
2. *signal* **ucitajListuTermina(List<Termin>)**

*у пројекту ће бити имплементирана као getAllTermin(List<Termin>)*

1. *signal* **nadjiTermine(ZadataVrednost, List<Termin>)**
2. *signal* **ucitajTermin(Termin)**
3. *signal* **izmeniTermin(Termin)**
4. *signal* **obrisiTermin(Termin)**
5. *signal* **kreirajDoktora (Doktor)**

## 2.2 Дефинисање уговора о системским операцијама

Понашање софтверског система се описује преко системских операција, а за сваку системску операцију се прави уговор. Уговор описују понашање системске операције, то јест описује се оно **шта** та системска опеација треба да одради (али не и како то треба да одради).

Један уговор везује се за једну системску операцију, и састоји се од следећих секција:

* *операција*
* *веза са СК*
* *предуслов*
* *постуслов*

### Уговор UG1: Login

**Операција: login(Doktor):** signal;

**Веза са СК:** СК1

**Предуслови:** */*

**Постуслови:** */*

### Уговор UG2: kreirajPacijenta

**Операција: kreirajPacijenta(Pacijent):**signal;

**Веза са СК:** СК2

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

**Постуслови:** *Kreiran je novi pacijent.*

### Уговор UG3: ucitajListuPacijenta

**Операција: getAllPacijent(List<Pacijent>):**signal;

**Веза са СК:** СК3, СК4, CK5, СК6

**Предуслови:** */*

**Постуслови:** */*

\*ne vrsi se nikakvo azuriranje pa mi pred I postsulovi ne trebaju

### Уговор UG4: nadjiPacijente

**Операција: nadjiPacijente(ZadataVrednost, List<Pacijent>):**signal;

**Веза са СК:** СК3, СК4, СК5

**Предуслови:** */*

**Постуслови:** */*

### Уговор UG5: ucitajPacijenta

**Операција: ucitajPacijenta(Pacijent):**signal;

**Веза са СК:** СК3, СК4, СК5

**Предуслови:** */*

**Постуслови:** */*

### Уговор UG6: izmeniPacijenta

**Операција: izmeniPacijenta(Pacijent)**:signal;

**Веза са СК:** СК4

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

**Постуслови:** *Подаци о пацијенту су измењени.*

### Уговор UG7: obrisiPacijenta

**Операција: obrisiPacijenta(Pacijent):**signal;

**Веза са СК:** СК5

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

**Постуслови:** *Пацијент је обрисана.*

### Уговор UG8: ucitajListuUsluga

**Операција: getAllUsluga(List<Usluga>):**signal;

**Веза са СК:** СК6

**Предуслови:** */*

**Постуслови:** */*

### Уговор UG9: kreirajTermin

**Операција: kreirajTermin(Termin):**signal;

**Веза са СК:** СК6

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

**Постуслови:** *Унети подаци о заказаном термину прегледа су запамћени.*

### Уговор UG10: ucitajListuTermina

**Операција: getAllTermin(List<Termin>):**signal;

**Веза са СК:** СК7, СК8, СК9

**Предуслови:** */*

**Постуслови:** */*

### Уговор UG11: nadjiTermine

**Операција: nadjiTermine(ZadataVrednost, List<Termin>)**signal;

**Веза са СК:** СК7, СК8, СК9

**Предуслови:** */*

**Постуслови:** */*

### Уговор UG12: ucitajTermin

**Операција: ucitajTermin(Termin) :**signal;

**Веза са СК:** СК7, СК8, СК9

**Предуслови:** */*

**Постуслови:** */*

### Уговор UG13: izmeniTermin

**Операција: izmeniTermin(Termin):**signal;

**Веза са СК:** СК8

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

**Постуслови:** *Заказани термин прегледа је измењен.*

### Уговор UG14: obrisiTermin

**Операција: obrisiTermin(Termin):**signal;

**Веза са СК:** СК9

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

**Постуслови:** *Заказани термин прегледа је обрисана.*

### Уговор UG15: kreirajDoktora

**Операција: kreirajDoktora(Doktor):**signal;

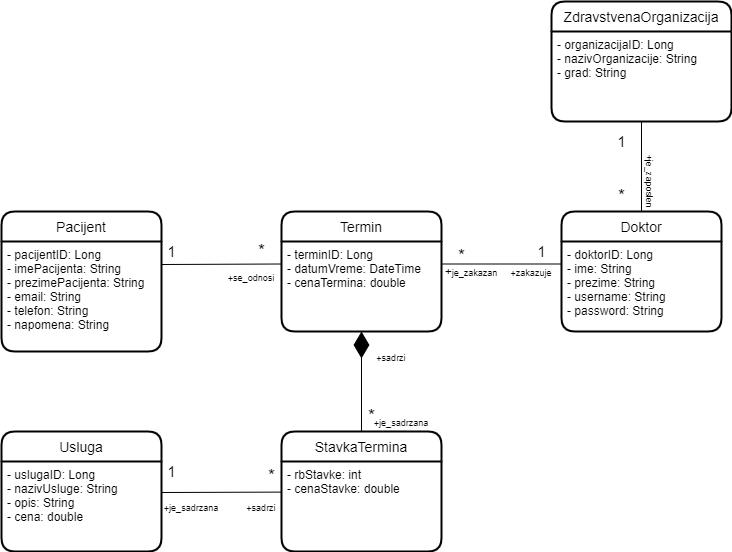
**Веза са СК:** СК10

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

**Постуслови:** *Kreiran je novi доктор.*

## Структура софтверског система - Концептуални (доменски) модел

Помоћу концептуалног модела описујемо структуру система. Концептуални модел садржи концептуалне класе (доменске објекте) и асоцијације између концептуалних класа.



Слика Концептуални модел

## Структура софтверског система- Релациони модел

На основу концептуалног модела, прави се релациони модел, а на основу њега се пројектује релациона база података.

У концептуалном моделу се могу индентификовати следеће класе: ZdravstvenaOrganizacija, Doktor, Pacijent, Usluga,Termin i StavkaTermina. Свака релација ће бити представљена као једна табела у релационом моделу.

ZdravstvenaOrganizacija(organizacijaID, nazivOrganizacije, grad)

Pacijent(pacijentID, imePacijenta, prezimePacijenta, email, telefon, napomena)

Usluga(uslugaID, nazivUsluge, opis, cena)

Doktor(doktorID, ime, prezime, username,password, *organizacijaID*)

Termin(terminID, datumVreme, cenaTermina, *doktorID, pacijentID*)

StavkaTermina(*terminID*,rbStavke, cenaStavke, *uslugID* )

**Табеле ограничења**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela ZdravstvenaOrganizacija | | Prosto vrednosno ograničenje | | Složeno vrednosno ograničenje | | Strukturno ograničenje |
| Atributi |  | Tip atributa | Vrednost atributa | Međuzavisnost atributa jedne tabele | Međuzavisnost atributa više tabela | INSERT /  UPDATE CASCADES Doktor  DELETE  RESTRICTED Doktor |
| organizacijaID | Long | Not null and >0 |  |  |
| nazivOrganizacije | String | Not null |  |  |
| grad | String | Not null |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela Pacijent | | Prosto vrednosno ograničenje | | Složeno vrednosno ograničenje | | Strukturno ograničenje |
| Atributi |  | Tip atributa | Vrednost atributa | Međuzavisnost atributa jedne tabele | Međuzavisnost atributa više tabela | INSERT /  UPDATE CASCADES  Termin  DELETE RESTRICTED Termin |
| pacijentID | Long | Not null and > 0 |  |  |
| imePacijenta | String | Not null |  |  |
| prezimePacijenta | String | Not null |  |  |
| email | String |  |  |  |
| brTelefona | String |  |  |  |
| napomena | String |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

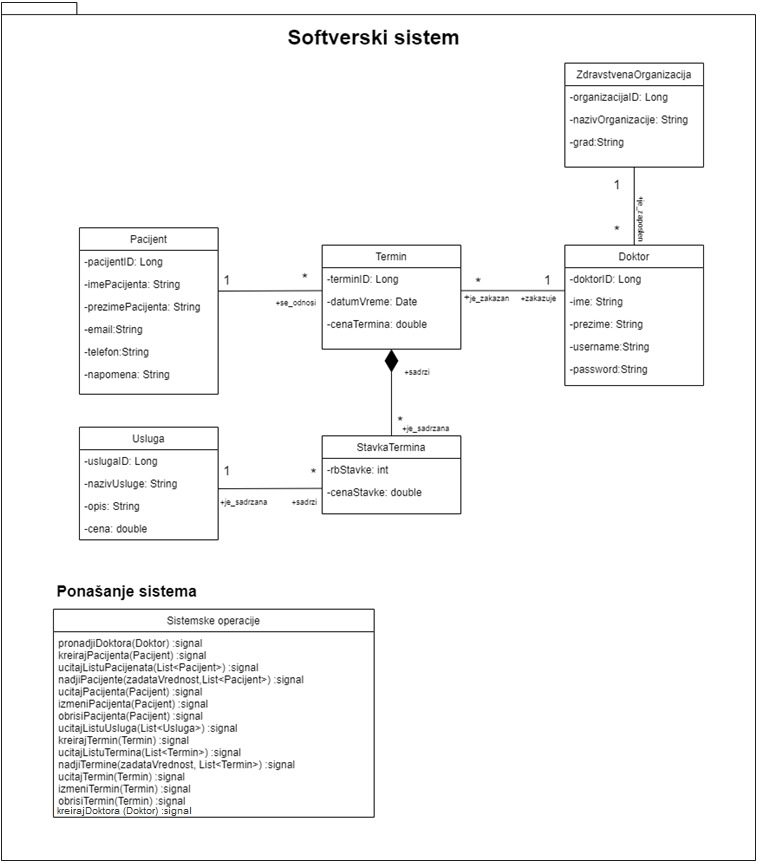
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela Usluga | | Prosto vrednosno ograničenje | | Složeno vrednosno ograničenje | | Strukturno ograničenje |
| Atributi |  | Tip atributa | Vrednost atributa | Međuzavisnost atributa jedne tabele | Međuzavisnost atributa više tabela | INSERT /  UPDATE CASCADES  StavkaTermina  DELETE RESTRICTEDStavkaTermina |
| uslugaID | Long | Not null and >0 |  |  |
| nazivUsluge | String |  |  |  |
| opis | String |  |  |  |
| cena | double |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela Doktor | | Prosto vrednosno ograničenje | | Složeno vrednosno ograničenje | | Strukturno ograničenje |
| Atributi |  | Tip atributa | Vrednost atributa | Međuzavisnost atributa jedne tabele | Međuzavisnost atributa više tabela | INSERT RESTRICTED Zdravstvena  Organizacija  UPDATE RESTRICTED Zdravstvena  Organizacija  , CASCADES  Termin  DELETE RESTRICTED  Termin |
| doktorID | Long | Not null and >0 |  |  |
| ime | String | Not null |  |  |
| prezime | String | Not null |  |  |
| username | String | Not null |  |  |
| password | String | Not null |  |  |
| organizacijaID | Long | Not null and >0 |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela  Termin | | Prosto vrednosno ograničenje | | Složeno vrednosno ograničenje | | Strukturno ograničenje |
| Atributi |  | Tip atributa | Vrednost atributa | Međuzavisnost atributa jedne tabele | Međuzavisnost atributa više tabela | INSERT RESTRICTED Doktor, Pacijent  UPDATE RESTRICTED Doktor, Pacijent,  CASCADE  StavkaTermina  DELETE  RESTRICTED  StavkaTermina |
| terminID | Long | Not null  and >0 |  |  |
| datumVreme | DateTime | Not null  and >0 |  |  |
| cenaTermina | Date | double |  | cenaTermina  =SUM(StavkaTermina.cenaStavke) |
| vrstaPrioriteta | String |  |  |  |
| doktorID | Long | Not null  and >0 |  |  |
| pacijentID | Long | Not null and >0 |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela StavkaTermina | | Prosto vrednosno ograničenje | | Složeno vrednosno ograničenje | | Strukturno ograničenje |
| Atributi |  | Tip atributa | Vrednost atributa | Međuzavisnost atributa jedne tabele | Međuzavisnost atributa više tabela | INSERT RESTRICTED Termin, Usluga  UPDATE RESTRICTED Termin, Usluga  DELETE/ |
| terminID | Long | Not null and >0 |  |  |
| rbStavke | Long | Not null and >0 |  |  |
| cenaStavke | double |  |  |  |
| uslugaID | Long | Not null and >0 |  |  |

Као резултат анализе сценарија случајева коришћења и прављења концептуалног модела добија се логичка структура и понашање софтверског система.



Слика Софтверски систем

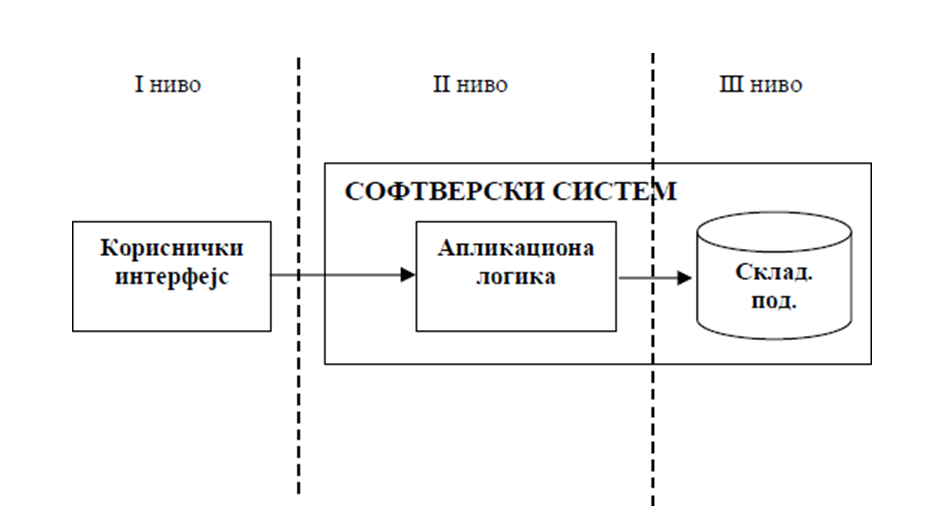
# 3.Фаза пројектовања

Фаза пројектовања описује физичку структуру и понашање софтверског система. Пројектовање архитектуре софтверског система обухвата пројектовање корисничког интерфејса (пројектовање контролера корисничког интерфејса и екранских форми), апликационе логике (пријектовање контролера апликационе логике и пословне логике) и складишта података (брокер базе података).

Архитектура система је тронивојска и састоји се од следећих нивоа:

* корисничкиинтерфејс
* апликационалогика
* складиштеподатака

Ниво корисничког интерфејса ја на страни клијента, док су апликациона логика и складиште на страни сервера.

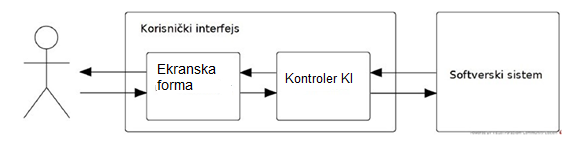


Слика Тронивојска архитектура

## 3.1 Пројектовање корисничког интерфејса

Кориснички интерфејс представља улазно-излазну реализацију софтверског система. Састоји се од:

1. Екранске форме
2. Контролера корисничког интерфејса



Слика Структура корисничког интерфејса

### 3.1.1 Пројектовање екранских форми

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

Постоје два аспекта пројектовања екранске форме:

1. Пројектовање сценарија СК који се изводе преко екранске форме

2. Пројектовање метода екранске форме

### СК1: Случај коришћења- Пријављивање доктора

**Назив СК**

Пријављивање **доктора**.

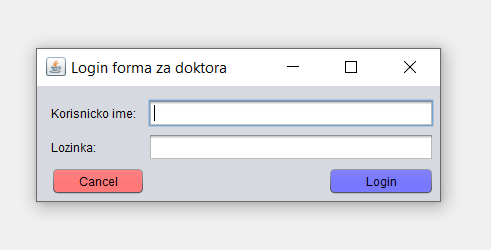
**Актори СК**

**Доктор**

**Учесници СК**

**Доктор и** систем (програм)

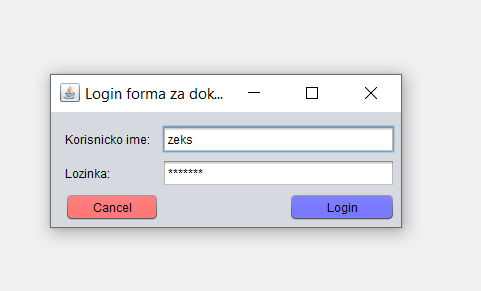
**Предуслов**: Систем је укључен и приказује форму за пријављивање **доктора**.



Слика Форма за пријавлјивање доктора на систем

**Основни сценарио СК**

1. **.Доктор уноси** податке за пријаву. (АПУСО)
2. **Доктор контролише** да ли је коректно унео податке за пријаву. (АНСО)



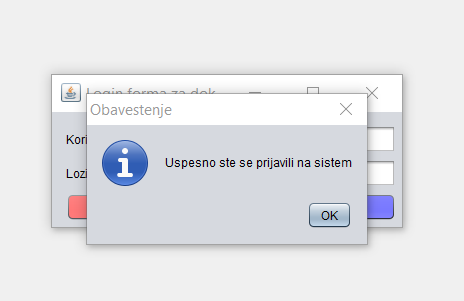
Слика Пријављивање доктора на систем

1. **Доктор позива** систем да пронађе **доктора** са задатим подацима.(АПСО)

Опис акције: Доктор кликом на дугме „login“ позива системску операцију **login(Doktor)**

4. Систем **претражује доктора**. (СО)

5. Систем **приказује** поруку „Успешно сте се пријавили на систем“. (ИА)

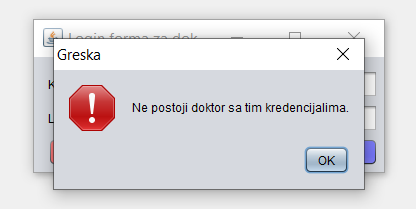


Слика Успешно пријављивање доктора

**Алтернативна сценарија**

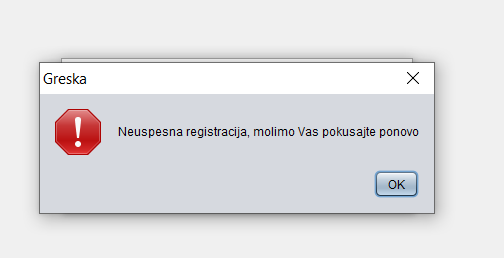
* 1. Уколико систем није пронашао **доктора**, приказује **доктору** поруку

“Систем не може да пронађе **доктора** на основу унетих података”.(ИА)



Слика Неуспешно пријављивање доктора

* 1. Уколико **доктор** није успео да се пријави на систем,систем шаље **доктору** поруку “Неуспешна регистрација, молимо Вас покушајте поново”. (ИА)



Слика Систем приказује поруку о неуспешној регистрацији

### СК2: Случај коришћења- Унос новог пацијента

**Назив СК**

Унос новог пацијента.

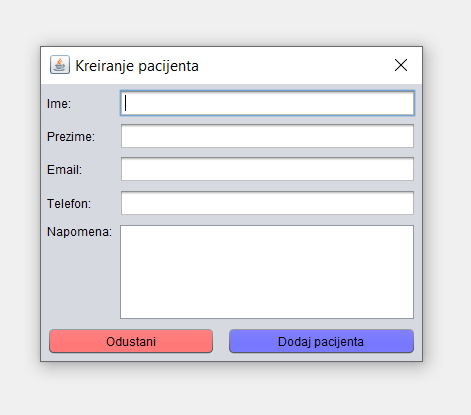
**Актори СК**

**Доктор**

**Учесници СК**

**Доктор и** систем (програм)

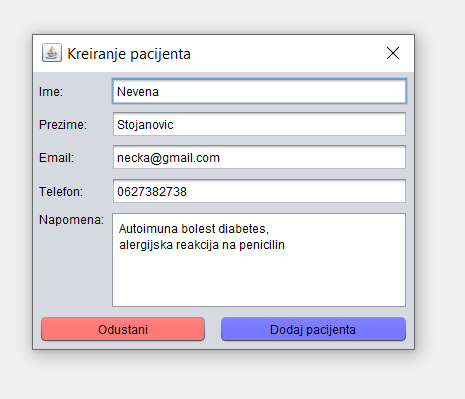
**Предуслов**: Систем је укључен и **доктор** је улогован под својом шифром. **Доктор** поседује потребне информације о пацијенту. Систем приказује форму за рад са пацијентима.



Слика Форма за уснос новог пацијента

**Основни сценарио СК**

1. **Доктор уноси** податке о пацијенту. (АПУСО)
2. **Доктор контролише** да ли је коректно унео податке о пацијенту. (АНСО)

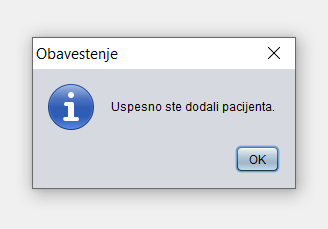


Слика Креирање новог пацијента од стране доктора

1. **Доктор позива** систем да креира новог пацијента.(АПСО)

Опис акције: Доктор кликом на дугме „Dodaj pacijenta“ позива системску операцију **kreirajPacijenta(Pacijent)**

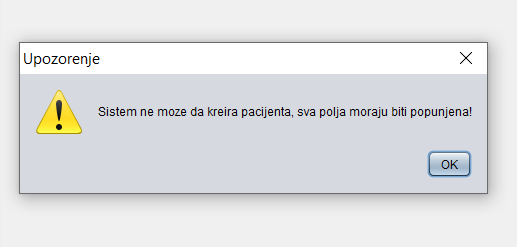
1. Систем **креира** новог пацијента. (СО)
2. Систем **приказује** поруку „Систем је успешно креирао новог пацијента“. (ИА)



Слика Систем приказује поруку о успешности додавања пацијента

**Алтернативна сценарија**

* 1. Уколико систем не може да креира новог пацијента, он приказује **доктору** поруку “Систем не може да креира новог пацијента”.(ИА)



Слика Систем приказује поруку о неуспешности креирања пацијента

### СК3: Претрага пацијента

**Назив СК**

Претрага пацијента

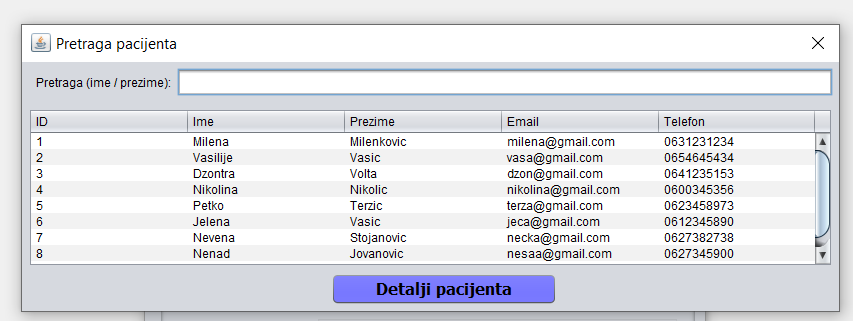
**Актори СК**

**Доктор**

**Учесници СК**

**Доктор и** систем (програм)

**Предуслов**: Систем је укључен и **доктору** je улогован под својом шифром. Систем приказује форму за претраживање пацијената. Учитана је листа пацијената.



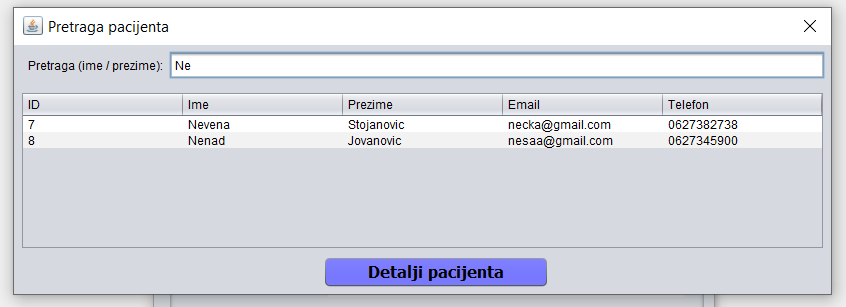
Слика Систем приказује форму за претраживање пацијената

**Основни сценарио СК**

* 1. **Доктор уноси** вредност по којој претражује пацијенте.(АПУСО)
  2. **Доктор позива** систем да нађе пацијенте по задатој вредности.(АПСО)

Опис акције: При самом отварању форме за претрагу пацијената, иницијално се позива системска операција **ucitajListuPacijenata(),** а учитани пацијенти се приказују у табели. Доктор у поље претраге уноси критеријум по коме претражује пацијенте, при чему се позива системска операција **nadjiPacijente(ZadataVrednost, List<Pacijent>)**

* 1. Систем **тражи** пацијенте по задатој вредности.(СО)
  2. Систем **приказује доктору** податке о пацијентима и поруку:„Систем је нашао пацијенте по задатој вредности.”.(ИА)



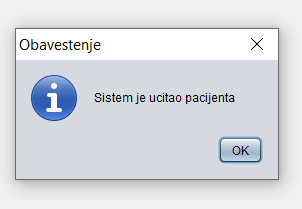
Слика Систем тражи пацијенте по задатој вредности

* 1. **Доктор** **бира** пацијента којег жели да му систем прикаже. (АПУСО)
  2. **Доктор** **позива** систем да учита пацијента. (АПСО)

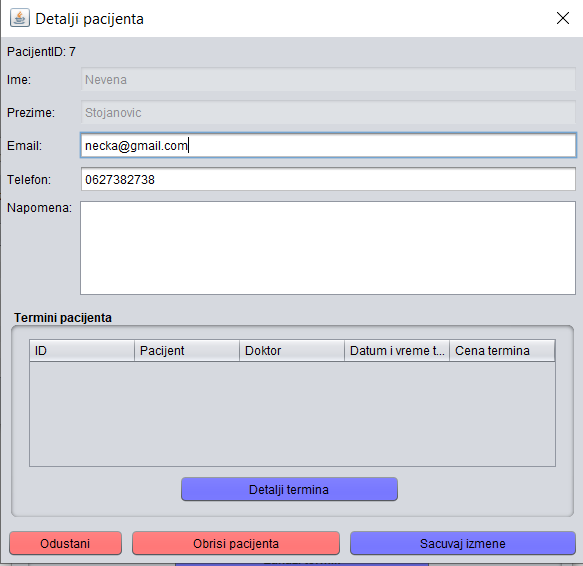
Опис акције: Доктор кликом на дугме „Detalji pacijenta“ позива системску операцију **ucitajPacijenta(Pacijent)**

* 1. Систем **учитава** пацијента. (СО)
  2. Систем **приказује** **доктору** податке о пацијенту и поруку:

“Систем је учитао пацијента“. (ИА)



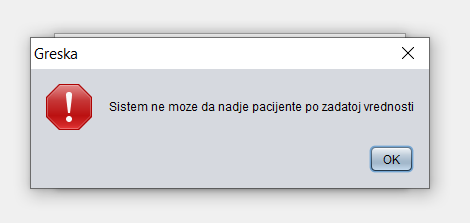
Слика Систем приказује поруку о успешности учитавања пацијента



Слика Систем приказује податке пацијента

**Алтернативна сценарија**

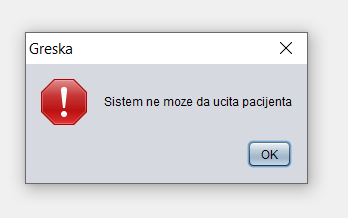
4.1 Уколико систем не може да нађе пацијенте по задатој вредности, он приказује **доктору** поруку: “Систем не може да нађе пацијенте по задатој вредности.”.(ИА)



Слика Систем приказује поруку о неупшеном учитавању пацијената

8.1. Уколико систем не може да учита пацијента он приказује **доктору** поруку:

“Систем не може да учита пацијента.”.(ИА)



Слика Систем приказује поруку о неуспешности учитавања пацијента

### СК4: Измена података пацијента

**Назив СК**

Измена података пацијента

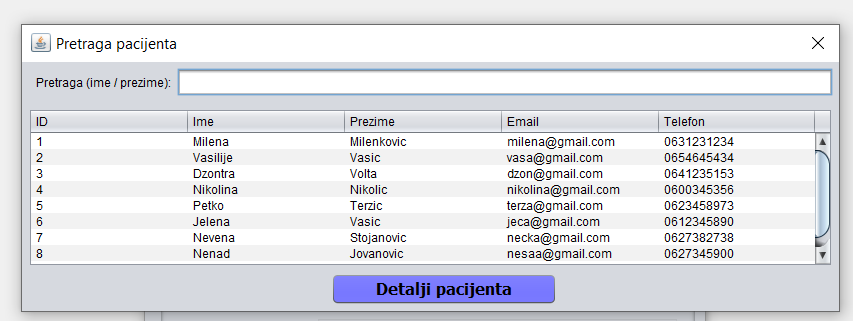
**Актори СК**

**Доктор**

**Учесници СК**

**Доктор** и систем (програм)

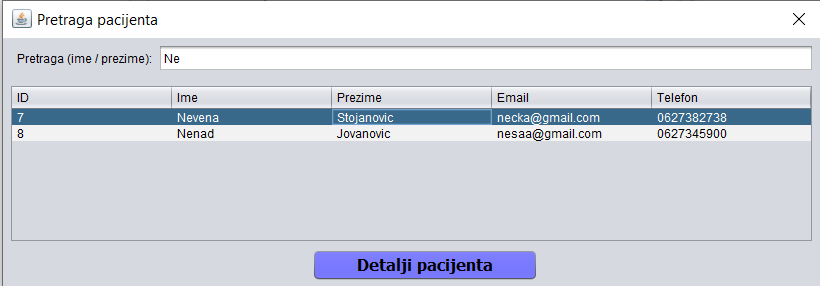
**Предуслов**: Систем је укључен и **доктор** je улогован под својом шифром. Систем приказује форму за рад са пацијентима. Учитана је листа пацијената.



Слика Форма за рад са пацијентима

**Основни сценарио СК**

* + - * 1. **Доктор уноси** вредност по којој претражује пацијенте.(АПУСО)

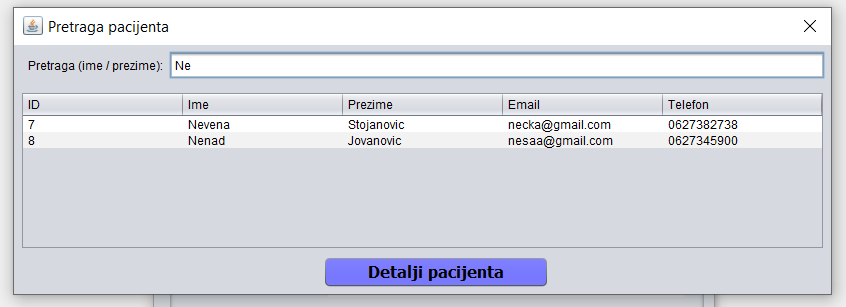
****

Слика Избор пацијента за измену

* + - * 1. **Доктор** **контролише** да ли је коректно унео податке за претрагу пацијената. (АНСО
        2. **Доктор позива** систем да пронађе пацијенте по задатој вредности.(АПСО)

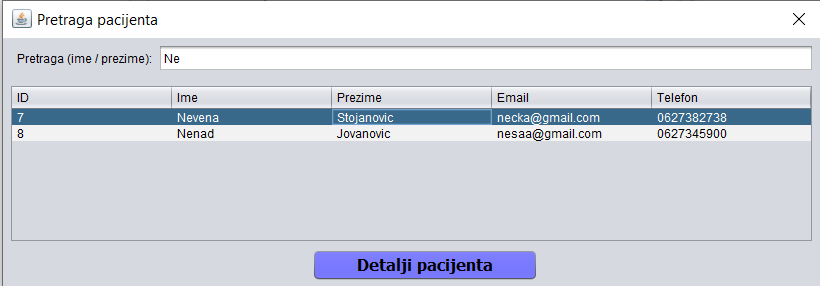
Опис акције: При самом отварању форме за претрагу пацијената, иницијално се позива системска операција **ucitajListuPacijenata(),** а учитани пацијенти се приказују у табели. Доктор у поље претраге уноси критеријум по коме претражује пацијенте, при чему се позива системска операција **nadjiPacijente(ZadataVrednost, List<Pacijent>)**

* + - * 1. Систем **тражи** пацијенте по задатој вредности.(СО)
        2. Систем **приказује** податке о траженим пацијентима уз поруку: „Подаци о траженим пацијентима су успешно учитани”.(ИА)



Слика Систем приказује пацијенте по задатој вредности

* + - * 1. **Доктор** **бира** пацијента којег жели да измени. (АПУСО)

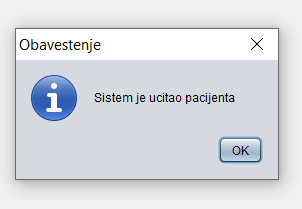


Слика Доктор бира пацијента за измену

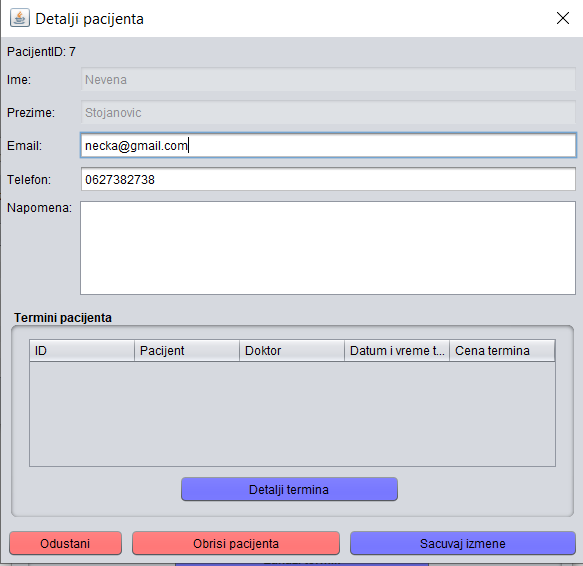
* + - * 1. **Доктор** **позива** систем да пронађе изабраног пацијента. (АПСО)

Опис акције: Доктор кликом на дугме „Detalji pacijenta“ позива системску операцију **ucitajPacijenta(Pacijent)**

* + - * 1. Систем **проналази** изабраногпацијента. (СО)
        2. Систем **приказује** податке о траженом пацијенту уз поруку: „ Систем **је пронашао** изабраног пацијента”.(ИА)

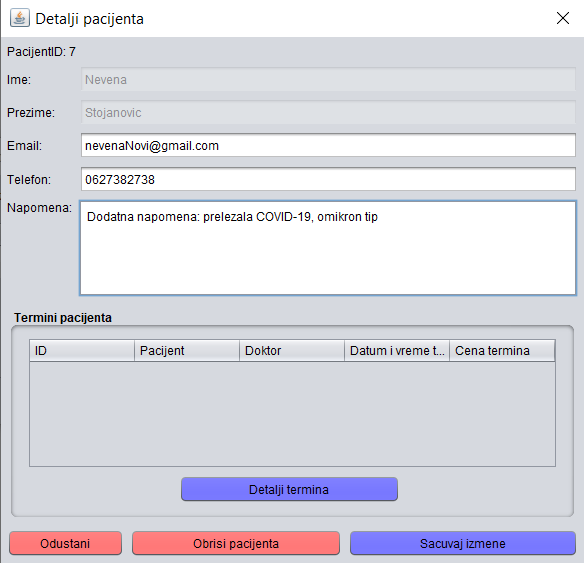


Слика Порука о успешности учитавања траженог пацијента



Слика Приказивање података изабраног пацијента

* + - * 1. **Доктор мења** податке о пацијенту.(АПУСО)
        2. **Доктор контролише** да ли је коректно унео податке о пацијенту.(АНСО)

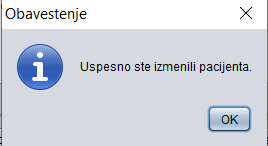


Слика Доктор мења податке пацијента

* + - * 1. **Доктор позива** систем да запамти податке о пацијену.(АПСО)

Опис акције: Доктор кликом на дугме „Sacuvaj izmene“ позива системску операцију **izmeniPacijenta(Pacijent)**

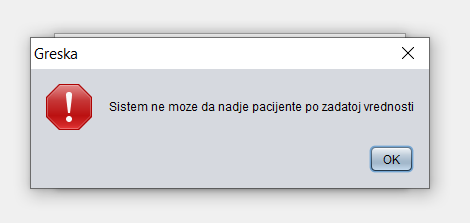
* + - * 1. Систем **памти** податке о пацијенту.(СО)
        2. Систем **приказује доктору** измењеног пацијента уз поруку: „Систем је изменио податке пацијента”.(ИА)



Слика Систем приказује поруку да је пацијент успешно измењен

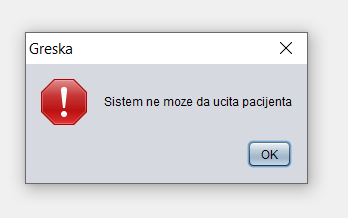
**Алтернативна сценарија**

Уколико систем не може да пронађе пацијенте, приказује следећу поруку **доктору**: „Систем не може да пронађе пацијенте на основу унетих података”. Прекида се извршење сценарија. (ИА)



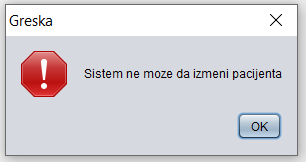
Слика Порука о неуспешном учитавању пацијента

* 1. Уколико систем не може да прикаже податке о пацијенту, приказује следећу поруку **доктору**: „Систем не може да пронађе пацијента“. Прекида се извршење сценарија. (ИА)



Слика Неуспешан проналазак пацијента

14.1. Уколико систем не може да измени пацијента, он приказује следећу поруку **доктору**: „Систем не може да измени пацијента”. (ИА)



Слика Порука о неуспешној измени пацијента

### СК5: Брисање пацијента

**Назив СК**

Брисање пацијента

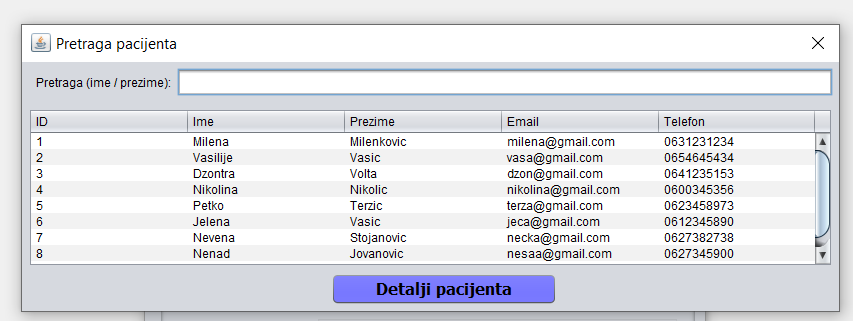
**Актори СК**

**Доктор**

**Учесници СК**

**Доктор** и систем (програм)

**Предуслов**: Систем је укључен и **доктор** je улогован под својом шифром. Систем приказује форму за рад са пацијентима. Учитана је листа пацијената.



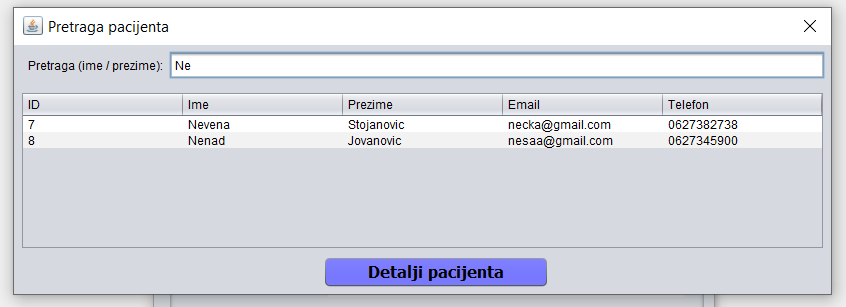
Слика Форма за рад са пацијентима

**Основни сценарио СК**

1. **Доктор** **уноси** вредност по којој претражује пацијенте. (АПУСО)
2. **Доктор** **контролише** да ли је коректно унео податке за претрагу пацијената. (АНСО)
3. **Доктор** **позива** систем да пронађе пацијенте по задатој вредности. (АПСО)

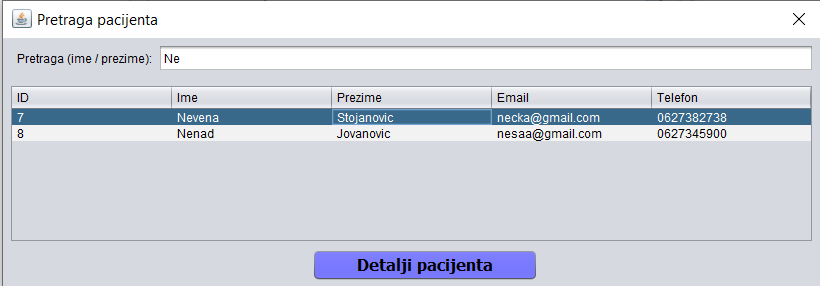
Опис акције: При самом отварању форме за претрагу пацијената, иницијално се позива системска операција **ucitajListuPacijenata(),** а учитани пацијенти се приказују у табели. Доктор у поље претраге уноси критеријум по коме претражује пацијенте, при чему се позива системска операција **nadjiPacijente(ZadataVrednost, List<Pacijent>)**

1. Систем **тражи** пацијенте по задатој вредности. (СО)
2. Систем **приказује** пацијенте **доктор** и поруку: „Систем је пронашао пацијенте по задатој вредности“. (ИА)



Слика Систем учитава пацијенте по задатој вредности

1. **Доктор** **бира** пацијента кога жели да избрише. (АПУСО)

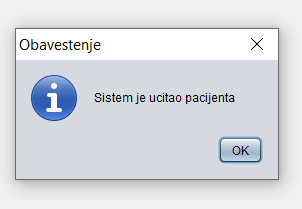
****

Слика Доктор бира пацијента за брисање

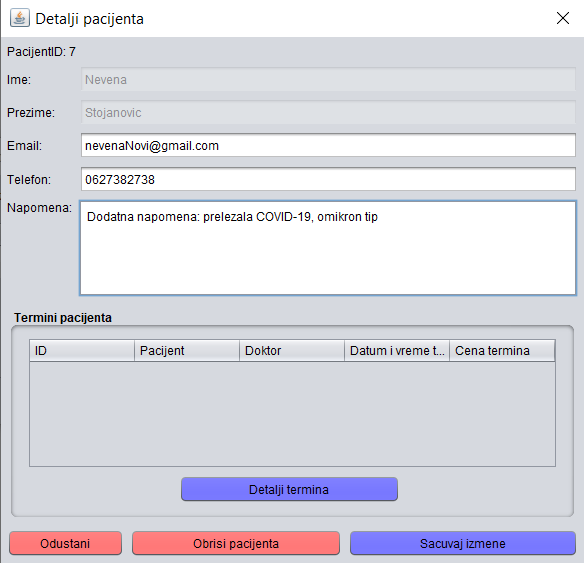
1. **Доктор** **позива** систем да нађе изабраног пацијента. (АПСО)

Опис акције: Доктор кликом на дугме „Detalji pacijenta“ позива системску операцију **ucitajPacijenta(Pacijent)**

1. Систем **проналази** изабраног пацијента. (СО)
2. Систем **приказује** пацијента и поруку: „Систем је пронашао пацијента“. (ИА)



Слика Систем приказује поруку да је успешно учитао пацијента

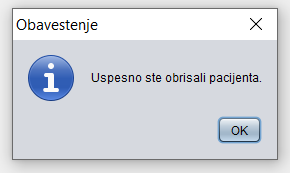


Слика Систем приказује податке изабраног пацијента

1. **Доктор** **позива** систем да обрише пацијента. (АПСО)

Опис акције: Доктор кликом на дугме „Obrisi pacijenta“ позива системску операцију **obrisiPacijenta(Pacijent)**

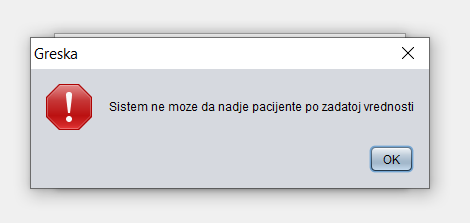
1. Систем **брише** пацијента. (СО)
2. Систем **приказује** **доктор** поруку: „Систем је обрисао пацијента“. (ИА)



Слика Систем приказује поруку о успешности брисања пацијента

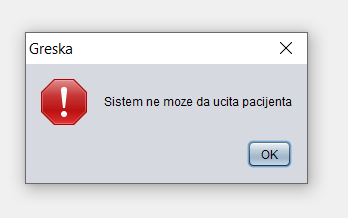
**Алтернативна сценарија:**

5.1 Уколико систем не може да нађе пацијенте, он приказује **доктор** поруку: “Систем не може да пронађе пацијенте по задатој вредности”. Прекида се извршење сценарија. (ИА)



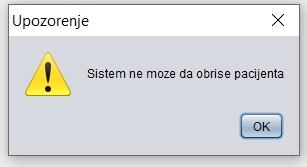
Слика Порука о неуспешном учитавању пацијената

9.1 Уколико систем не може да нађе пацијенте, он приказује **доктору** поруку: “Систем не може да учита пацијента.”. Прекида се извршење сценарија. (ИА)



Слика Порука о неуспешном учитавању пацијента

12.1 Уколико систем не може да обрише пацијента он приказује **доктору** поруку: „Систем не може да обрише пацијента “. (ИА)



Слика Порука о неуспешном брисању пацијента

### СК6: Унос термина (сложен СК)

**Назив СК**

Унос термина

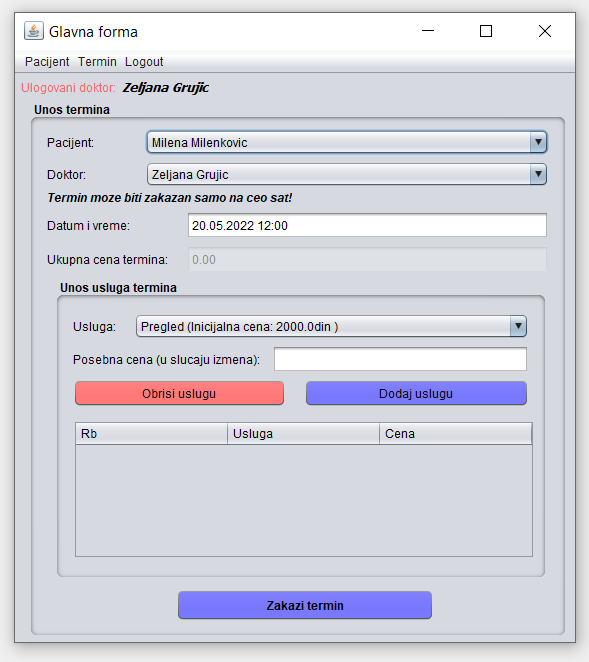
**Актори СК**

**Доктор**

**Учесници СК**

**Доктор** и систем (програм)

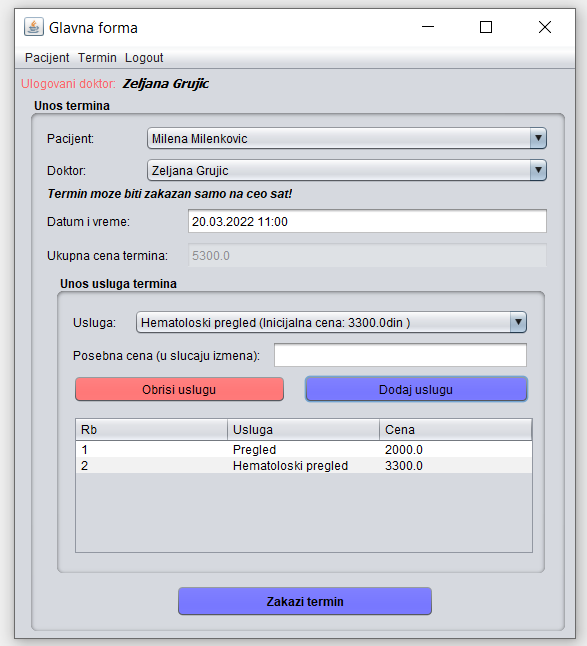
**Предуслов**: Систем је укључен и **доктор** је улогован под својом шифром.Систем приказује форму за рад са терминима. Учитане су листа пацијената и листа услуга.



Слика Форма за рад са терминима

**Основни сценарио СК**

* + - * 1. **Доктор** **уноси** податке о термину.(АПУСО)
        2. **Доктор** **контролише** да ли је коректно унео податке о термину.(АНСО)

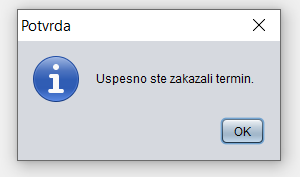


Слика Доктор уноси податке о новом темрину

* + - * 1. **Доктор позива** систем да креира термин.(АПСО)

Опис акције: Доктор кликом на дугме „Zakazi termin“ позива системску операцију **kreirajTermin(Termin)**

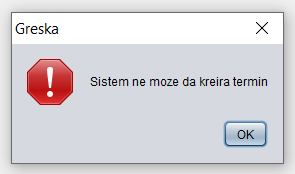
* + - * 1. Систем **креира** термин.(СО)
        2. Систем **приказује** **доктору** креирани термин и поруку:„ Систем је креирао термин.”. (ИА)



Слика Систем обавештава доктора о успешности заказивања термина

**Алтернативна сценарија:**

Уколико систем не може да креира термин он приказује **доктору** поруку: “Систем не може да креира термин“.(ИА)



Слика Систем обавештава доктора о неуспешном заказивању термина

### СК7: Претрага термина

**Назив СК**

Претрага термин

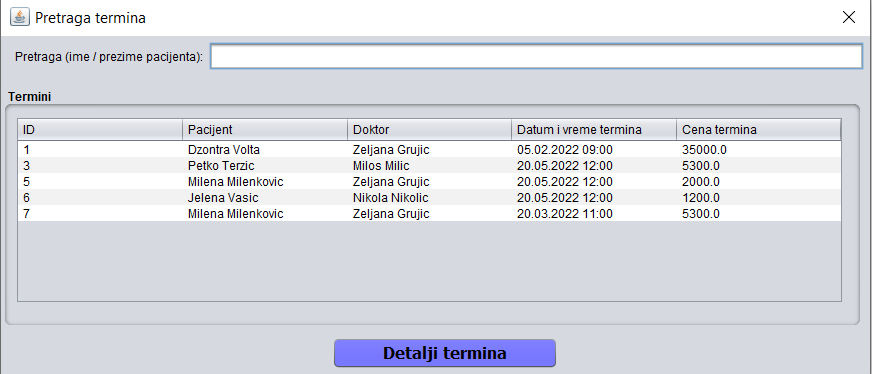
**Актори СК**

**Доктор**

**Учесници СК**

**Доктор** и систем (програм)

**Предуслов**:Систем је укључен и **доктор** je улогован под својом шифром. Систем приказује форму за претраживање термина. Учитана је листа термина.



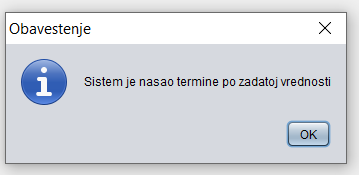
Слика Форма за претраживање темрина

**Основни сценарио СК**

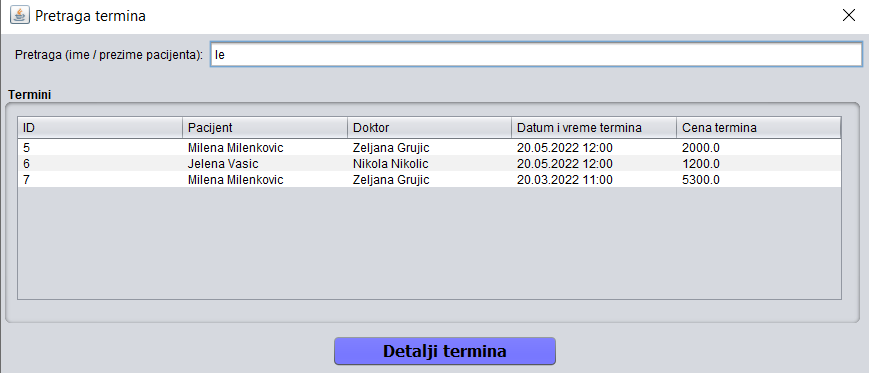
1. **Доктор** **уноси** вредност по којој претражује термине. (АПУСО)
2. **Доктор позива** систем да нађе термине по задатој вредности. (АПСО)

Опис акције: При самом отварању форме за претрагу термина, иницијално се позива системска операција **ucitajListuTermina(),** а учитани термини се приказују у табели. Доктор у поље претраге уноси критеријум по коме претражује пацијенте, при чему се позива системска операција **nadjiTermine(ZadataVrednost, List<Termin>)**

1. Систем **тражи** термине по задатој вредности. (СО)
2. Систем **приказује** **доктору** податке о терминима и поруку:„Систем је нашао термине по задатој вредности.”. (ИА)



Слика Систем обавештава доктора о успешној претрази термина

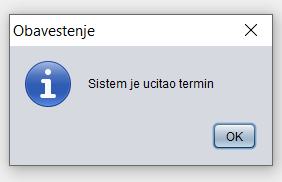


Слика Систем поналази термине по задатој вредности

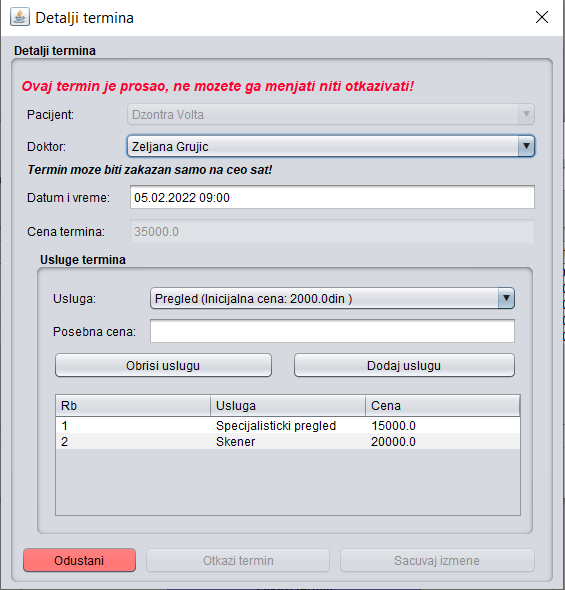
1. **Доктор** **бир** термин који жели да му систем прикаже. (АПУСО)
2. **Доктор** **позива** систем да учита термин. (АПСО)

Опис акције: Доктор кликом на дугме „Detalji termina“ позива системску операцију **ucitajTermin(Termin)**

1. Систем **учитава** термин. (СО)
2. Систем **приказује** **доктору** податке о термину и поруку:“Систем је учитао термин.“. (ИА)



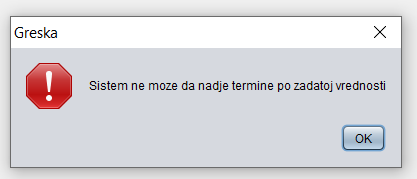
Слика Систем обавештава доктора о успешном учитавању термина



Слика Систем приказује податке о изабраном термину

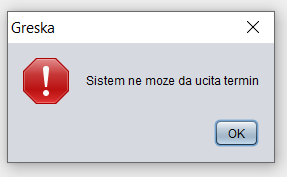
**Алтернативна сценарија:**

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



Слика Систем обавештава доктора о неуспешном учитавању свих термина

* 1. Уколико систем не може да учита термин приказује **доктор** поруку: “Систем не може да учита термин.”.(ИА)



Слика Систем обавештава доктора да не може да учита одабрани термин

### СК8: Измена термина (сложени СК)

**Назив СК**

Измена термина

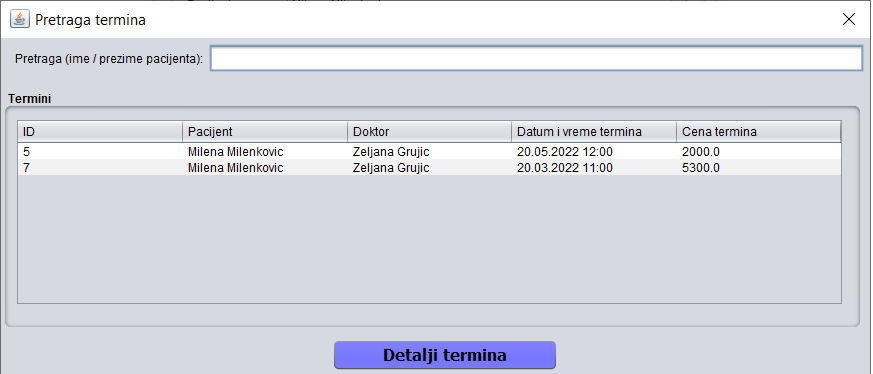
**Актори СК**

**Доктор**

**Учесници СК**

**Доктор** и систем (програм)

**Предуслов:** Систем је укључен и **доктор** je улогован под својом шифром. Систем приказује форму за рад са термина. Учитане су листе термина, пацијената и услуга.



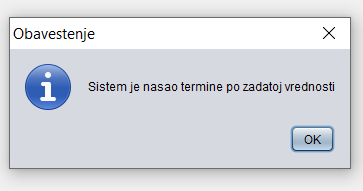
Слика Форма за рад са терминима

**Основни сценарио СК**

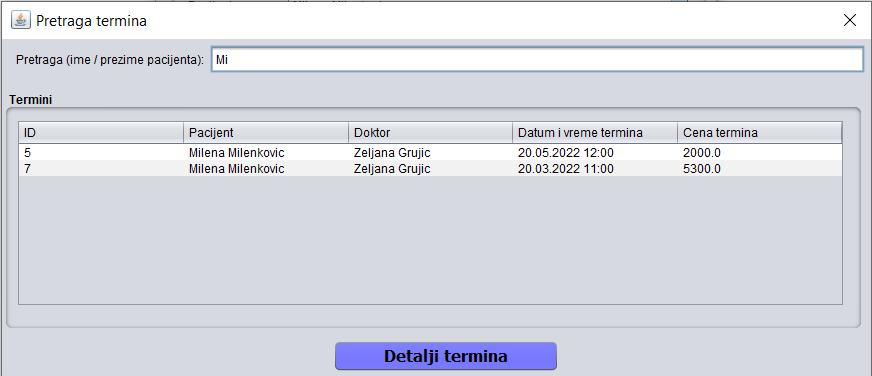
* + - * 1. **Доктор** **уноси** вредност по којој претражује термине. (АПУСО)
        2. **Доктор контролише** да ли је коректно унео податке за претрагу термина. (АНСО)
        3. **Доктор позива** систем да нађе термине по задатој вредности. (АПСО)
        4. Систем **тражи** термине по задатој вредности. (СО)

Опис акције: При самом отварању форме за претрагу термина, иницијално се позива системска операција **ucitajListuTermina(),** а учитани термини се приказују у табели. Доктор у поље претраге уноси критеријум по коме претражује пацијенте, при чему се позива системска операција **nadjiTermine(ZadataVrednost, List<Termin>)**

* + - * 1. Систем **приказује** **доктору** термине и поруку: „Систем је нашао термине по задатој вредности.“. (ИА)



Слика Систем обавештава доктора о успешном проналаску свих термина

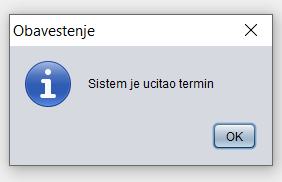


Слика Систем приказује термине по задатој вредности

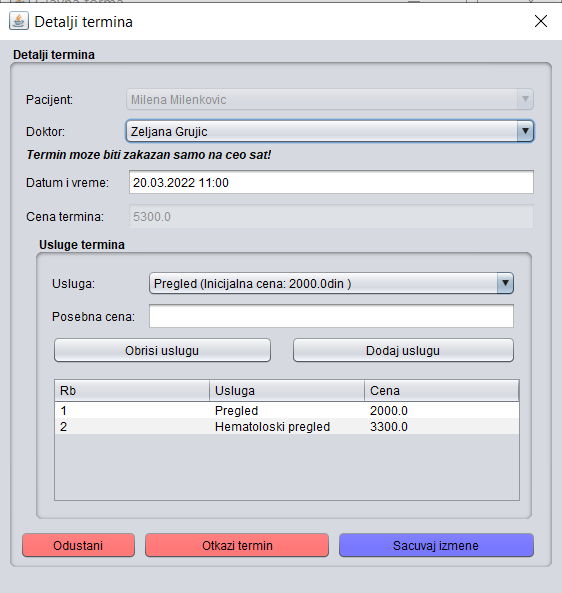
* + - * 1. **Доктор** **бира** термин који жели да измени. (АПУСО)
        2. **Доктор позива** систем да нађе изабрани термин. (АПСО)

Опис акције: Доктор кликом на дугме „Detalji termina“ позива системску операцију **ucitajTermin(Termin)**

* + - * 1. Систем **проналази** изабрани термин. (СО)
        2. Систем **приказује** **доктору** термин и поруку: „Систем је учитао термин “. (ИА)

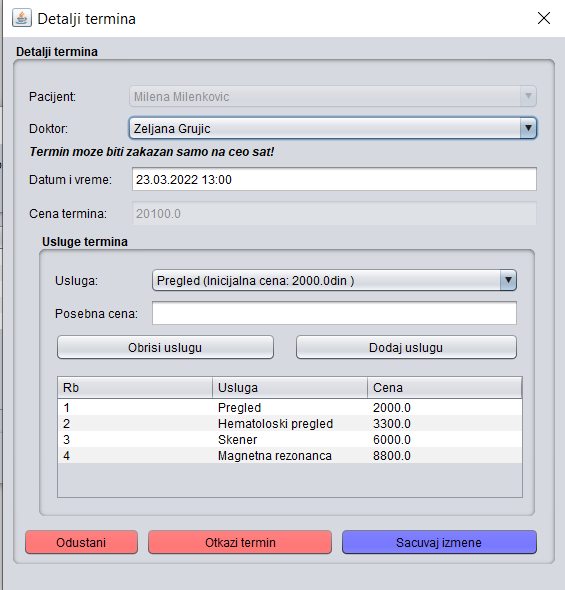


Слика Порука о успешности учитавања термина



Слика Систем приказује податке о изабраном термину

* + - * 1. **Доктор** **мења** податке о термину. (АПУСО)
        2. **Доктор контролише** да ли је коректно унео податке о термину. (АНСО)

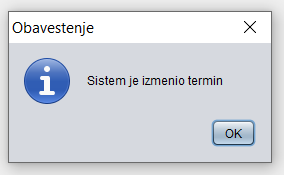


Слика Промена података изабраног термина

* + - * 1. **Доктор** **позива** систем да запамти податке о термину. (АПСО)

Опис акције: Доктор кликом на дугме „Sacuvaj izmene“ позива системску операцију **izmeniTermin(Termin)**

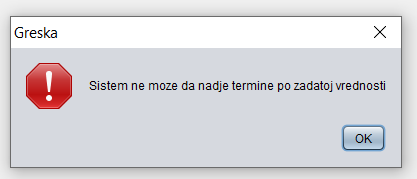
* + - * 1. Систем **памти** податке о термину. (СО)
        2. Систем **приказује доктору** поруку: „Систем је изменио термин.“. (ИА)



Слика Порука о успешности измене термина

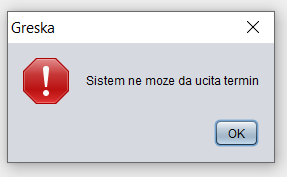
**Алтернатинвна сценарија:**

5.1 Уколико систем не може да пронађе термине он приказује **доктору** поруку: “Систем не може да пронађе термине по задатој вредности”. Прекида се извршење сценарија. (ИА)



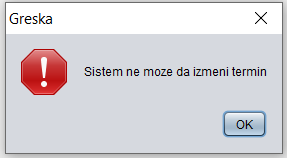
Слика Порука о неуспешности проналаска термина по задатој вредности

9.1. Уколико систем не може да прикаже податке о термину, приказује следећу поруку **доктору**: „ Систем не може учита термин.”. Прекида се извршење сценарија. (ИА)



Слика Порука о неуспешности учитавања жељеног термина

14.1 Уколико систем не може да измени податке о термину он приказује **доктору** поруку: „Систем не може да измени термин“. (ИА)



Слика Порука о неуспешности измене термина

### СК9: Отказивање термина

**Назив СК**

Брисање термина

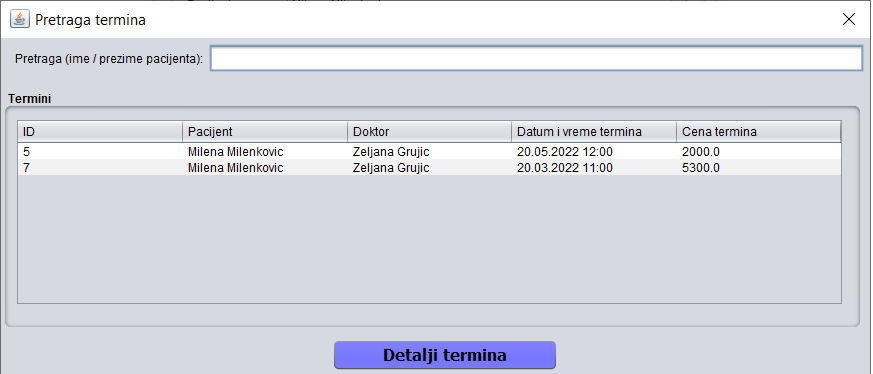
**Актори СК**

**Доктор**

**Учесници СК**

**Доктор** и систем (програм)

**Предуслов**: Систем је укључен и **доктор** je улогован под својом шифром.Систем приказује форму за рад са терминима. Учитана је листа термина.



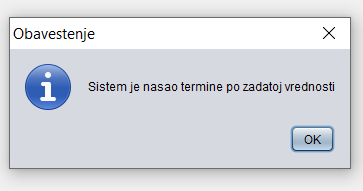
Слика Форма за рад са терминима

**Основни сценарио СК**

1. **Доктор** **уноси** вредност по којој претражује термине. (АПУСО)
2. **Доктор контролише** да ли је коректно унео податке за претрагу термина. (АНСО)
3. **Доктор позива** систем да нађе термине по задатој вредности. (АПСО)

Опис акције: При самом отварању форме за претрагу термина, иницијално се позива системска операција **ucitajListuTermina(),** а учитани термини се приказују у табели. Доктор у поље претраге уноси критеријум по коме претражује пацијенте, при чему се позива системска операција **nadjiTermine(ZadataVrednost, List<Termin>)**

1. Систем **тражи** термине по задатој вредности. (СО)
2. Систем **приказује** термине **доктору** и поруку: „Систем је нашао термине по задатој вредности“. (ИА)

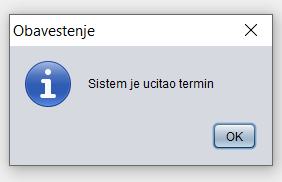


Слика Порука о успешности проналаска термина по задатој вредности

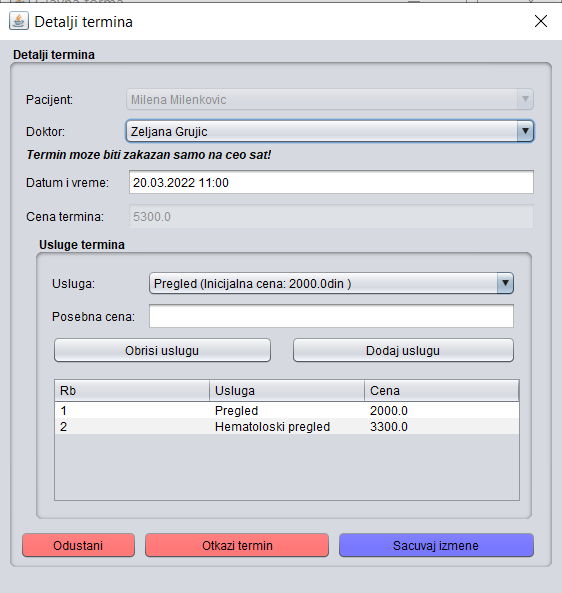
1. **Доктор бира** термин који жели да избрише. (АПУСО)
2. **Доктор позива** систем да нађе изабрани термин. (АПСО)

Опис акције: Доктор кликом на дугме „Detalji termina“ позива системску операцију **ucitajTermin(Termin)**

1. Систем **проналази** изабрани термин. (СО)
2. Систем **приказује** **доктору** термин и поруку: „Систем је учитао термин“. (ИА)



Слика Систем приказује доктору поруку о успешно учитаном термину

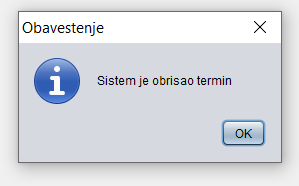


Слика Систем приказује доктору податке о учитаном термину

1. **Доктор** позива систем да обрише термин. (АПСО)

Опис акције: Доктор кликом на дугме „Otkazi“ позива системску операцију **obrisiTermin(Termin)**

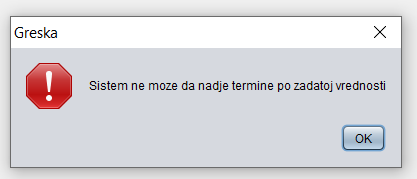
1. Систем **брише** термин. (СО)
2. Систем **приказује** **доктору** поруку: „Систем је обрисао термин“. (ИА)



Слика систем приказује поруку о успешном брисању термина

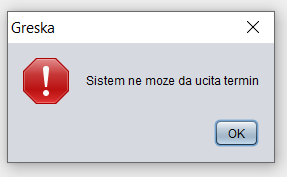
**Алтернативна сценарија:**

5.1 Уколико систем не може да нађе термине, он приказује **доктору** поруку: “Систем не може да нађе термине по задатој вредности”. Прекида се извршење сценарија. (ИА)



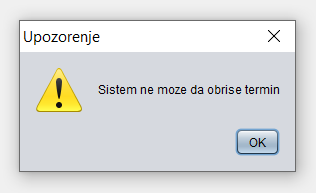
Слика Систем приказује поруку о неуспешном проналаску свих термина

9.1 Уколико систем не може да прикаже податке о термину, он приказује **доктору** поруку: “Систем не може да учита термин.”. Прекида се извршење сценарија. (ИА)



Слика Систем приказује поруку о неуспешном проналаску жељеног термина

12.1 Уколико систем не може да обрише термин он приказује **доктору** поруку: „Систем не може да обрише термин.“. (ИА)



Слика Систем приказује поруку о немогућности брисања термина

### СК10: Регистрација новог доктора

**Назив СК**

Регистрација новог доктора.

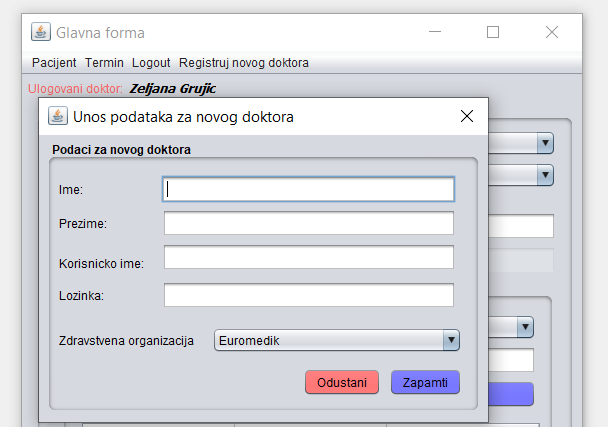
**Актори СК**

**Доктор**

**Учесници СК**

**Доктор и** систем (програм)

**Предуслов**: Систем је укључен и **доктор** је улогован под својом шифром. **Доктор** поседује потребне информације о новом доктору. Систем приказује форму за рад са докторима.



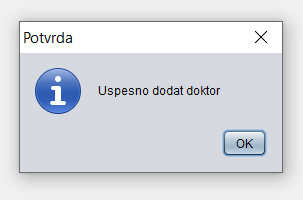
Слика Форма за унос новог доктора

**Основни сценарио СК**

1. **Доктор уноси** податке о новом доктору. (АПУСО)
2. **Доктор контролише** да ли је коректно унео податке о доктору. (АНСО)
3. **Доктор позива** систем да креира новог доктора.(АПСО)

Опис акције: Доктор кликом на дугме „Zapamti“ позива системску операцију **kreirajDoktora(Doktor)**

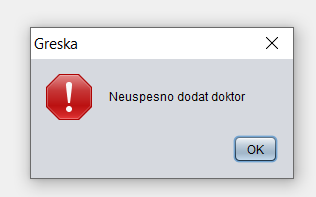
1. Систем **креира** новог доктора. (СО)
2. Систем **приказује** поруку „Систем је успешно креирао новог доктора“. (ИА)



Слика Порука о успешности креирања новог доктора

**Алтернативна сценарија**

5.1 Уколико систем не може да креира новог доктора, он приказује **доктору** поруку “Систем не може да креира новог доктора”.(ИА)

****

Слика Порука о неуспешности додавања доктора

### 3.1.2 Пројектовање контролера корисничког интерфејса

Kонтролер корисничког интерфејса је одговоран да:

1. прихвати податке које шаље екранска форма
2. конвертује податке (који се налазе у графичким елементима) у објекат који представља улазни аргумент СО која ће бити позвана
3. шаље захтев за извршење системске операције до апликационог сервера (софтверског система)
4. прихвата објекат (излаз) софтверског система настаo као резултат извршења системске операције
5. конвертује објекат у податке графичких елемената

## 3.2 Пројектовање апликационе логике

Апликациона логика описује структуру и понашање система. Апликациони сервер се састоји из:

1. **Контролера апликационе логике** – треба да подигне серверски сокет који ће да ослушкује мрежу. Служи за комуникацију са клијентом и одговоран је да прихвати захтев за извршење системске операције од клијента и проследи га до пословне логике која је одговорна за извршење СО
2. **Пословна логика** – описана је структуром (доменске класе) и понашањем (системске операције)
3. **Брокер базе података** – служи за комуникацију између пословне логике и базе података

### 3.2.1 Конторлер апликационе логике

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.

Софтверски систем реализован је као клијент-сервер апликација. На серверској страни је нит *ThreadServer* која садржи објекат класе *ServerSocket*. Нит константно позива методу *accept* која чека да се покрене клијентска апликација која, кад се то деси, ће покушати да се повеже на сервер. Слање и примање података од клијента се остварује преко сокета, метода *accept* креира објекат класе *Socket*. Клијент шаље захтев за извршење неке од СО до одговарајуће нити (коју смо назвали *ThreadClient*), која је повезана са тим клијентом. *ThreadClient* прима захтев и даље га преусмерава до класа које су одговорне за извршење СО. Након извршења СО резултат се враћа до апликационе логике, односно до класе *ThreadClient* на серверској страни која тај резултат шаље назад до клијента путем сокета

### 3.2.2 Пословна логика

#### 3.2.2.1 Пројектовање понашања софтверског система (системске операције)

За сваки од претходно дефинисаних уговора правимо системску операцију, што заправо представља пројектовање понашања. Класа *AbstractSO* која представља апстрактну класу која садржи методу *templateExecute*, која представља шаблон извршавања сваке операције над базом података, а као параметар прима објекат класе *AbstractDomainObject*. У тој методи се позивају методе *validate* и *execute*, које су апстрактне и које ће свака класа системске операције имплементирати. Након тога се позива метода *commit* која узима објекат *Connection*, класе *Кonekcija* и позива њену методу *commit*. Провера предуслова се извршава на клијентској страни уколико постоји, а постуслови се очитавају у оквиру *Response* објекта који шаље сервер клијенту и на основу кога клијент закључује да ли је операција успешно извршена на серверској страни или је дошло до грешке.

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

За сваки уговор пројектује се концептуално решење.

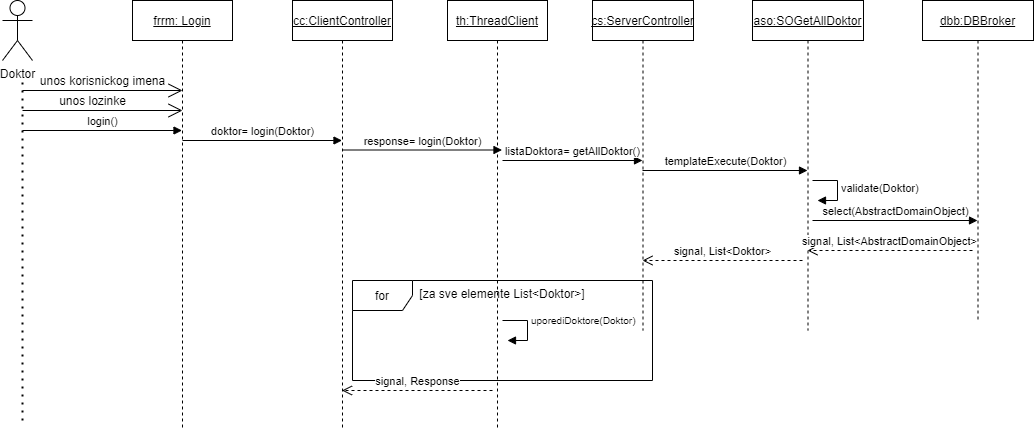
#### Уговор UG1: Login

**Операција: login(Doktor):** signal;

**Веза са СК:** СК1

**Предуслови:** */*

**Постуслови:** */*



Слика Дијаграм секвенци за уговор login

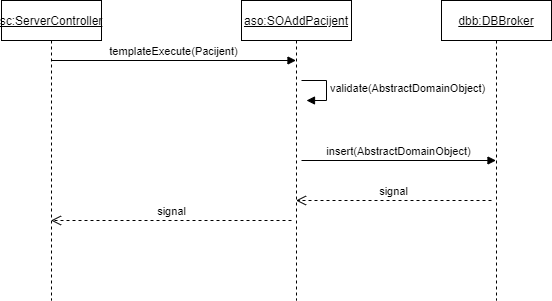
#### Уговор UG2: kreirajPacijenta

**Операција: kreirajPacijenta(Pacijent):**signal;

**Веза са СК:** СК2

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

**Постуслови:** *Kreiran je novi pacijent.*



Слика Дијаграм секвенци за уговор kreirajPacijenta

#### Уговор UG3: ucitajListuPacijenta

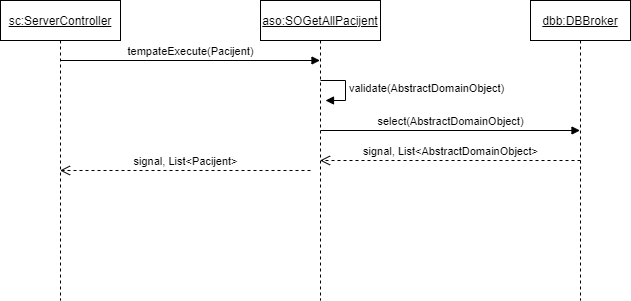
**Операција: getAllPacijent(List<Pacijent>):**signal;

**Веза са СК:** СК3, СК4, CK5, СК6

**Предуслови:** */*

**Постуслови:** */*

\*ne vrsi se nikakvo azuriranje pa mi pred I postsulovi ne trebaju



Слика Дијаграм секвенци за уговор ucitajListuPacijenata

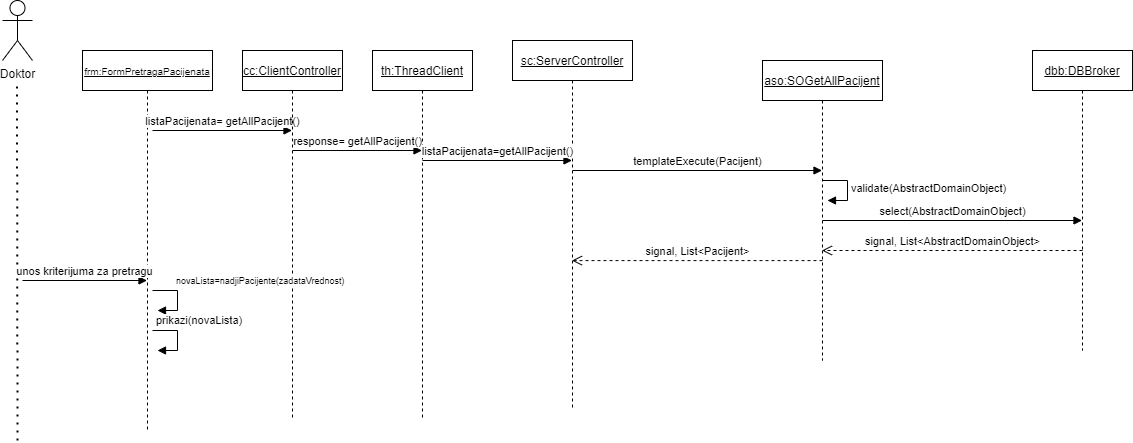
#### Уговор UG4: nadjiPacijente

**Операција: nadjiPacijente(ZadataVrednost, List<Pacijent>):**signal;

**Веза са СК:** СК3, СК4, СК5

**Предуслови:** */*

**Постуслови:** */*



Слика Дијаграм секвенци за уговор nadjiPacijente

Ова операција позива операцију која враћа све пацијенте (getAllPacijent) и затим на клијентској страни врши филтрирање листе пацијената по критеријуму.

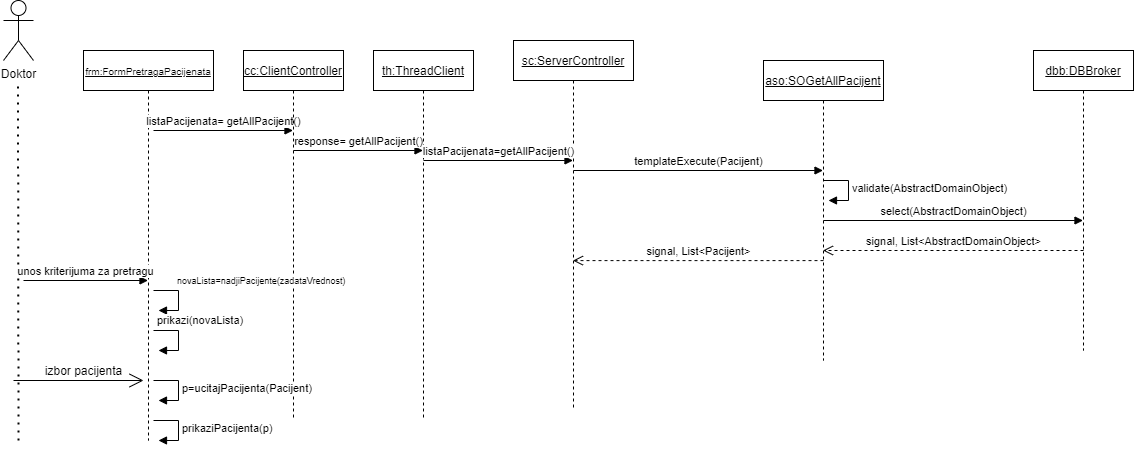
#### Уговор UG5: ucitajPacijenta

**Операција: ucitajPacijenta(Pacijent):**signal;

**Веза са СК:** СК3, СК4, СК5

**Предуслови:** */*

**Постуслови:** */*



Слика Дијаграм секвенци за уговор ucitajPacijenta.

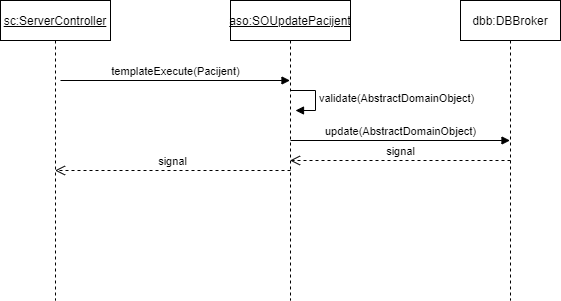
#### Уговор UG6: izmeniPacijenta

**Операција: izmeniPacijenta(Pacijent)**:signal;

**Веза са СК:** СК4

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

**Постуслови:** *Подаци о пацијенту су измењени.*



Слика Дијаграм секвенци за уговор izmeniPacijenta

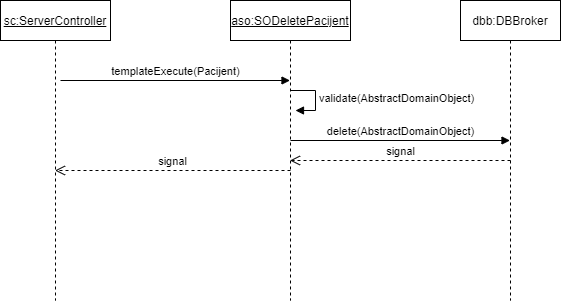
#### Уговор UG7: obrisiPacijenta

**Операција: obrisiPacijenta(Pacijent):**signal;

**Веза са СК:** СК5

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

**Постуслови:** *Пацијент је обрисана.*



Слика Дијаграм секвенци за уговор obrisiPacijenta

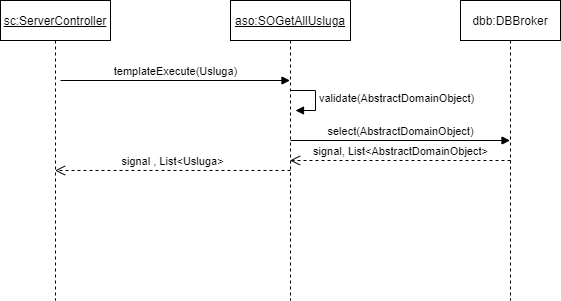
#### Уговор UG8: ucitajListuUsluga

**Операција: getAllUsluga(List<Usluga>):**signal;

**Веза са СК:** СК6

**Предуслови:** */*

**Постуслови:** */*



Слика Дијаграм секвенци за уговор ucitajListuUsluga

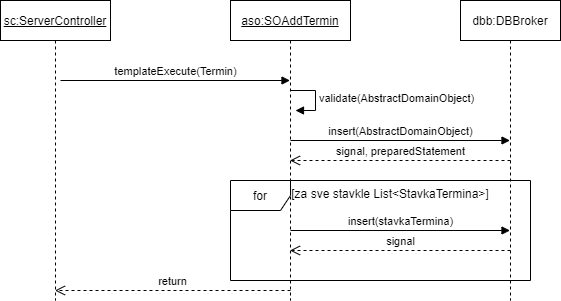
#### Уговор UG9: kreirajTermin

**Операција: kreirajTermin(Termin):**signal;

**Веза са СК:** СК6

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

**Постуслови:** *Унети подаци о заказаном термину прегледа су запамћени.*



Слика Дијаграм секвенци за уговор kreirajTermin

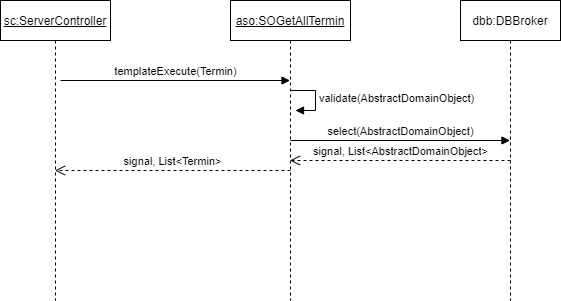
#### Уговор UG10: ucitajListuTermina

**ОперацијаgetAllTermin(List<Termin>):**signal;

**Веза са СК:** СК7, СК8, СК9

**Предуслови:** */*

**Постуслови:** */*



Слика Дијаграм секвенци за уговор ucitajListuTermina

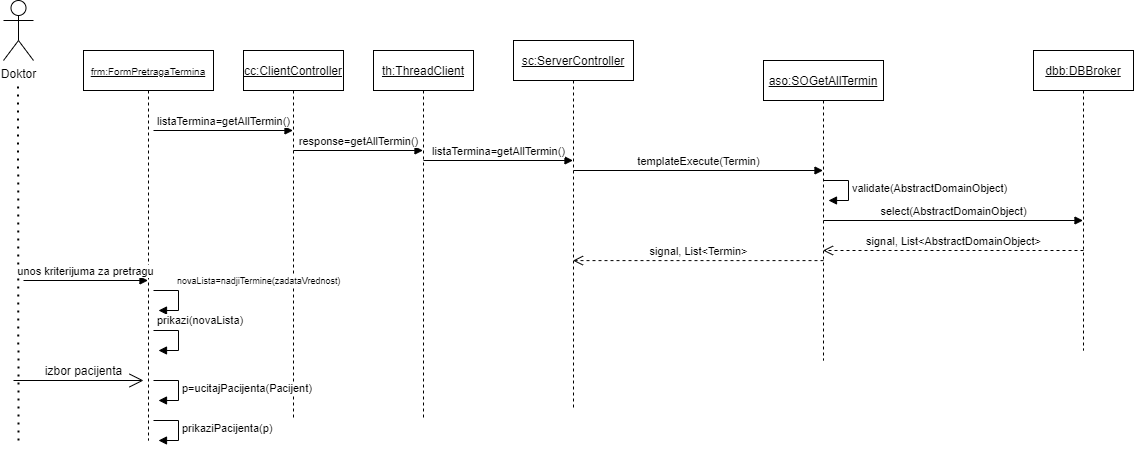
#### Уговор UG11: nadjiTermine

**Операција: nadjiTermine(ZadataVrednost, List<Termin>)**signal;

**Веза са СК:** СК7, СК8, СК9

**Предуслови:** */*

**Постуслови:** */*

**

Слика Дијаграм секвенци за уговор nadjiTermine

Ова операција позива операцију која враћа све термине (getAllTermin) и затим на клијентској страни врши филтрирање листе термина по критеријуму.

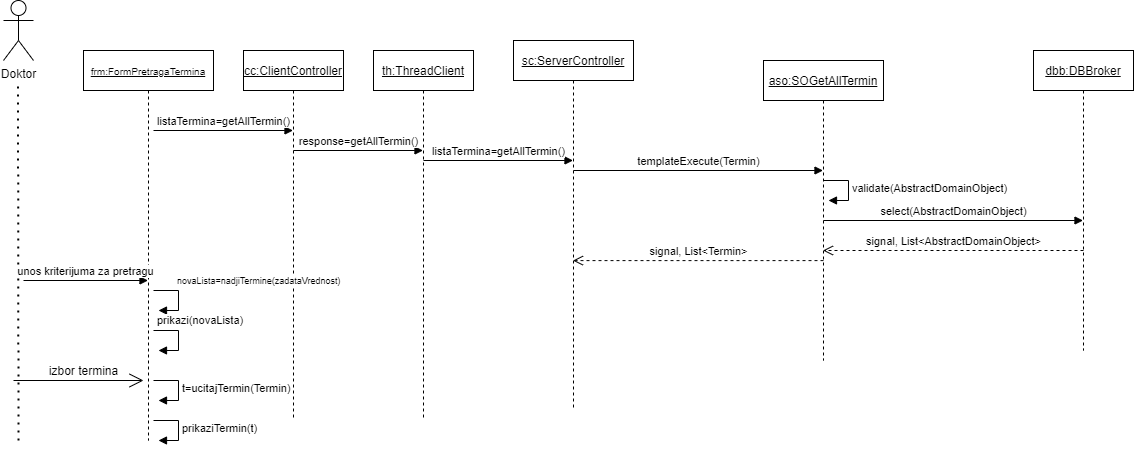
#### Уговор UG12: ucitajTermin

**Операција: ucitajTermin(Termin) :**signal;

**Веза са СК:** СК7, СК8, СК9

**Предуслови:** */*

**Постуслови:** */*

**

Ова операција позива операцију која враћа све термине (getAllTermin) и затим на клијентској страни врши филтрирање листе термина и проналази одабрани термин.

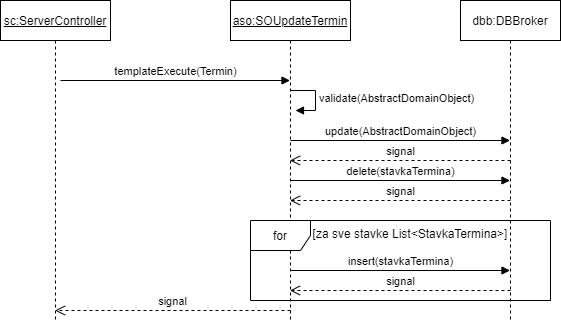
#### Уговор UG13: izmeniTermin

**Операција: izmeniTermin(Termin):**signal;

**Веза са СК:** СК8

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

**Постуслови:** *Заказани термин прегледа је измењен.*



Слика Дијаграм секвенци за уговор izmeniTermin

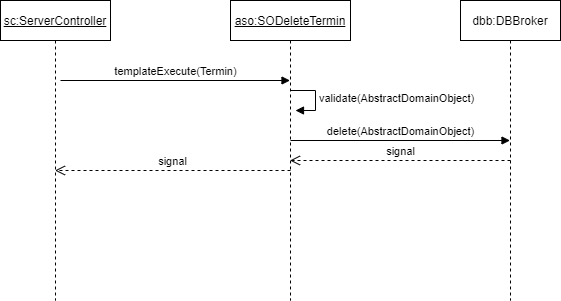
#### Уговор UG14: obrisiTermin

**Операција: obrisiTermin(Termin):**signal;

**Веза са СК:** СК9

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

**Постуслови:** *Заказани термин прегледа је обрисана.*



Слика Дијаграм секвенци за уговор obrisiTermin

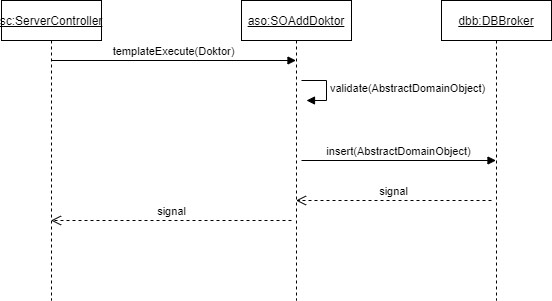
### Уговор UG15: kreirajDoktora

**Операција: kreirajDoktora(Doktor):**signal;

**Веза са СК:** СК10

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

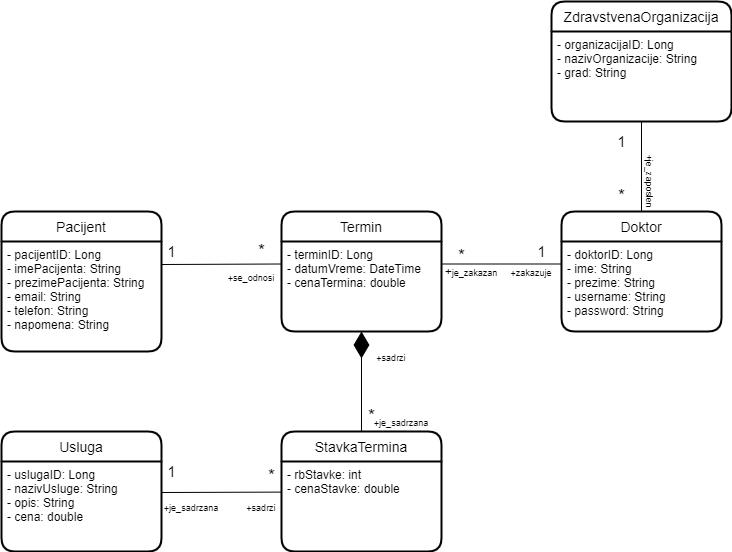
**Постуслови:** *Kreiran je novi доктор*



Слика Дијаграм секвенци за уговор kreirajDoktora

#### 3.2.2.2 Пројектовање структуре софтверског система (доменске класе)

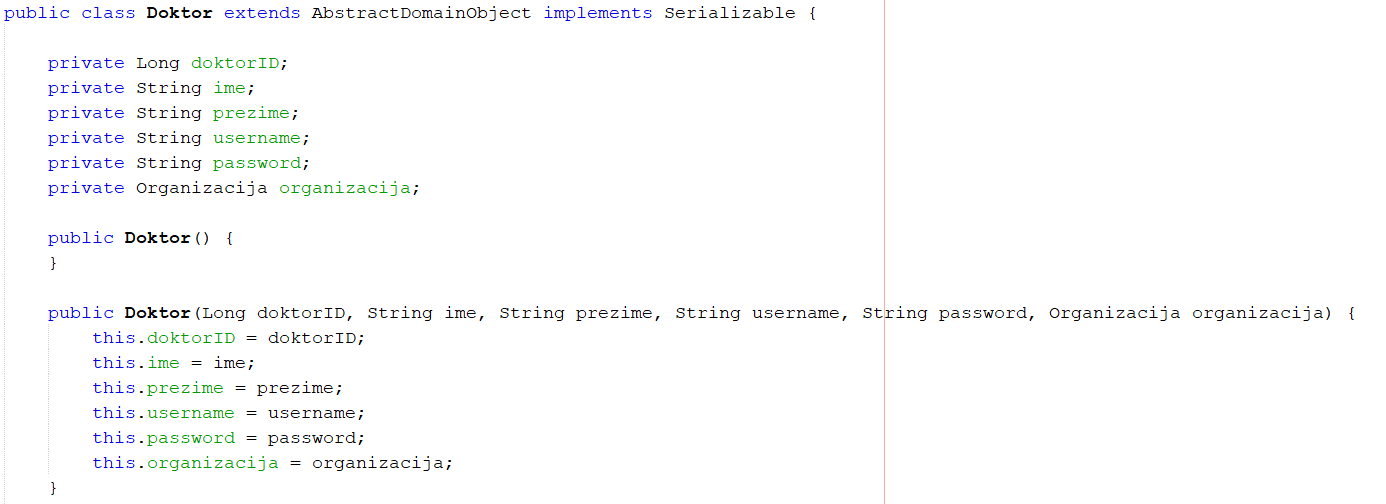
На основу концептуалних класа креирају се софтверске класе структуре.



Слика Концептулани дијаграм класа

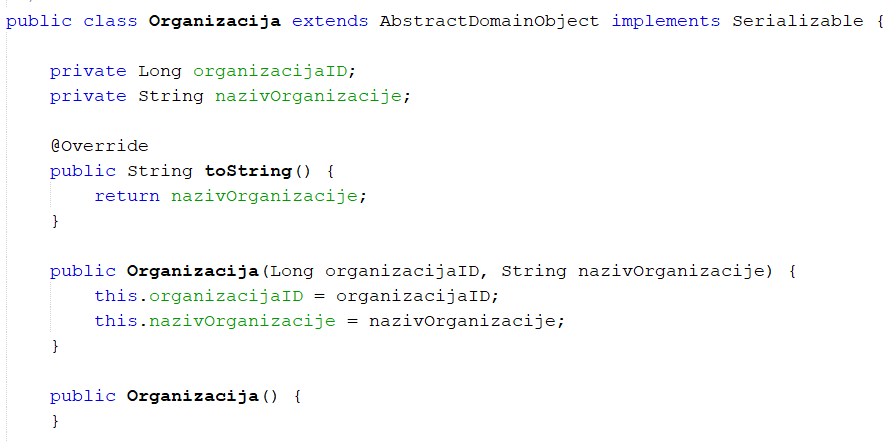
Идентификоване су следедће класе:

* Доктор



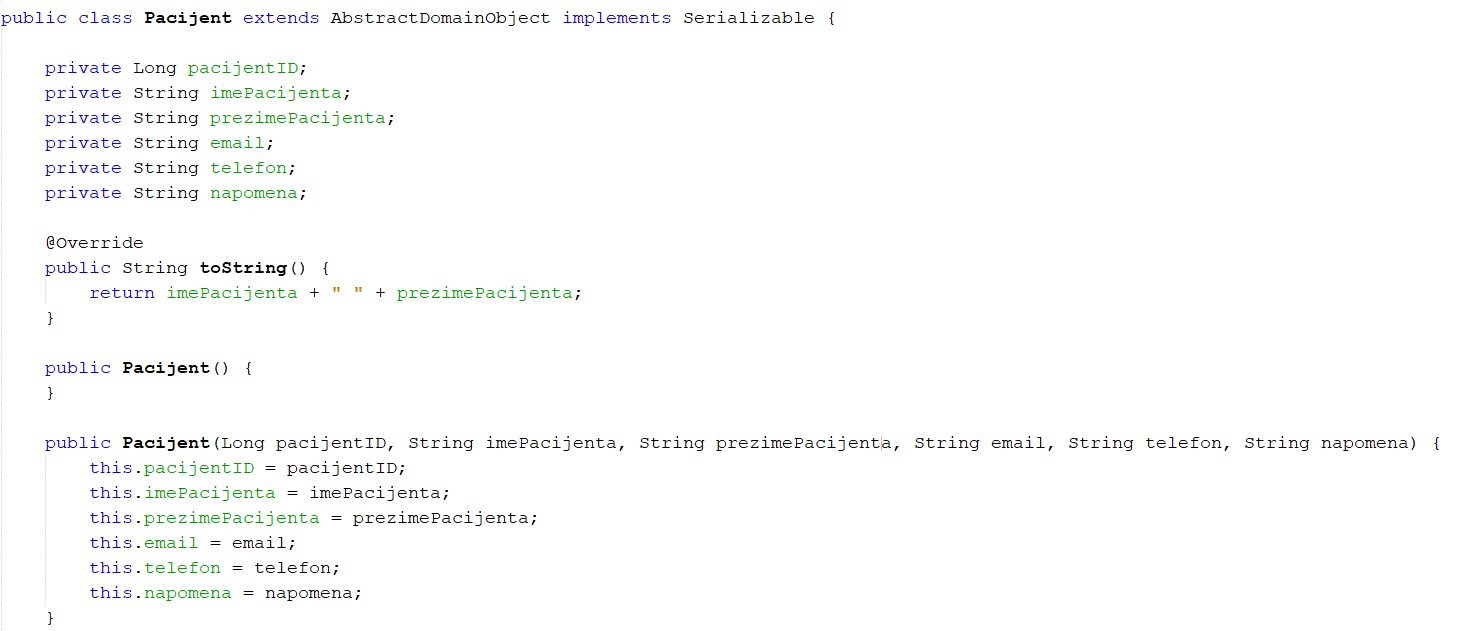
Слика Класа Доктор

* Здравствена организација



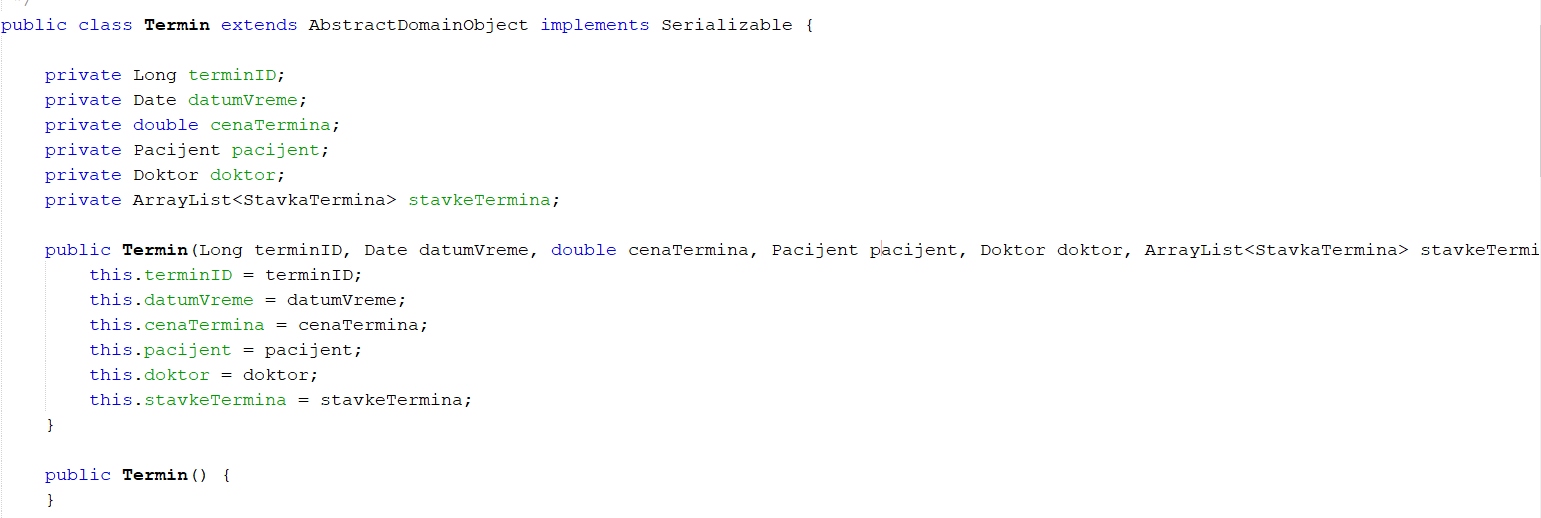
Слика Класа Организација

* Пацијент



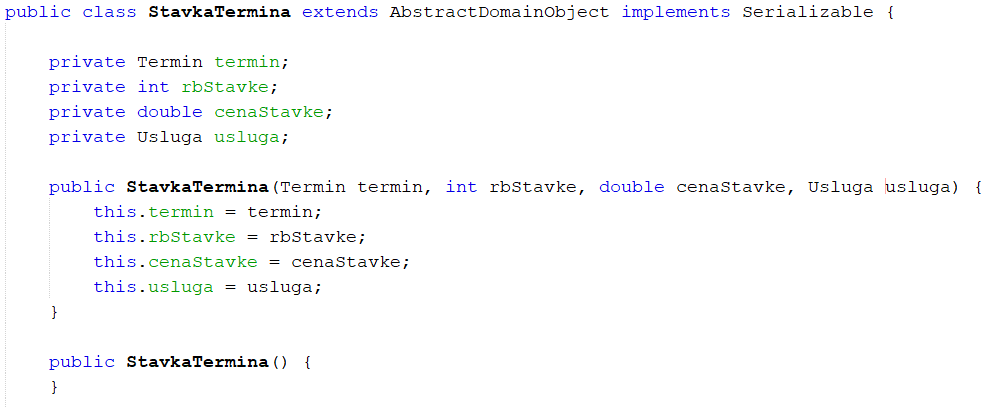
Слика Класа Пацијент

* Термин



Слика Класа Термин

* Ставка термина



Слика Класа СтавкаТермина

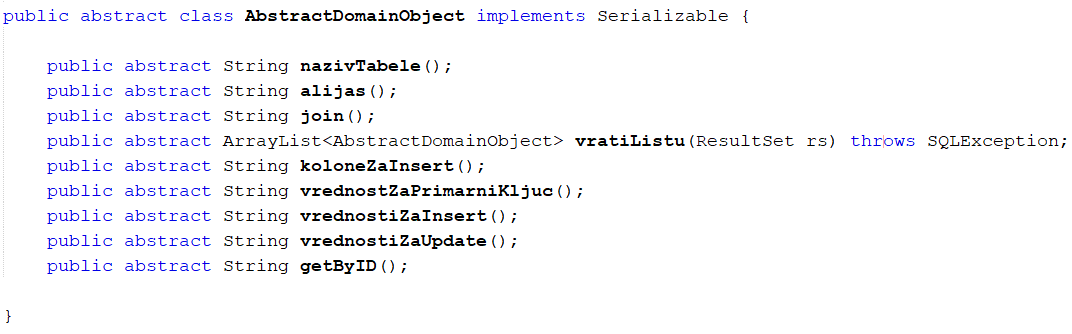
* Услуга



Слика Класа Услуга

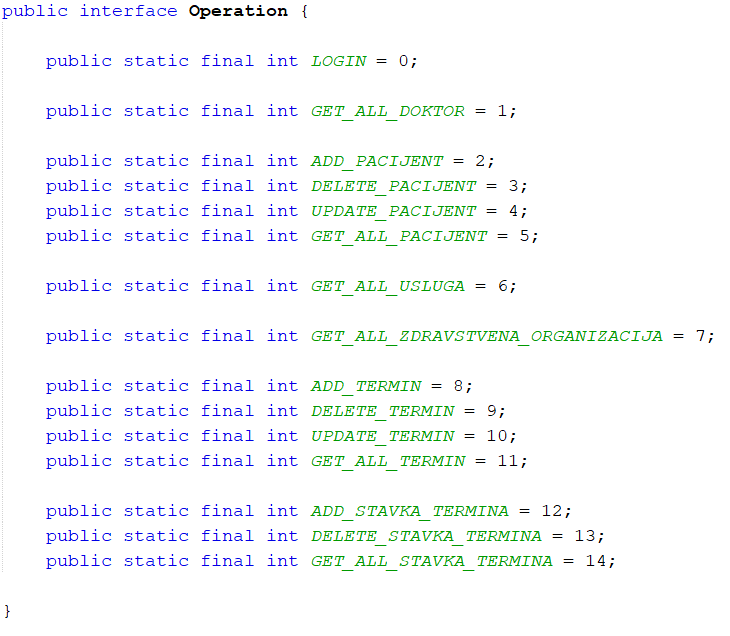
Поред њих додате су и следеће класе:

* AbstractDomainObject - апстрактна класа које све доменске класе наслеђују.



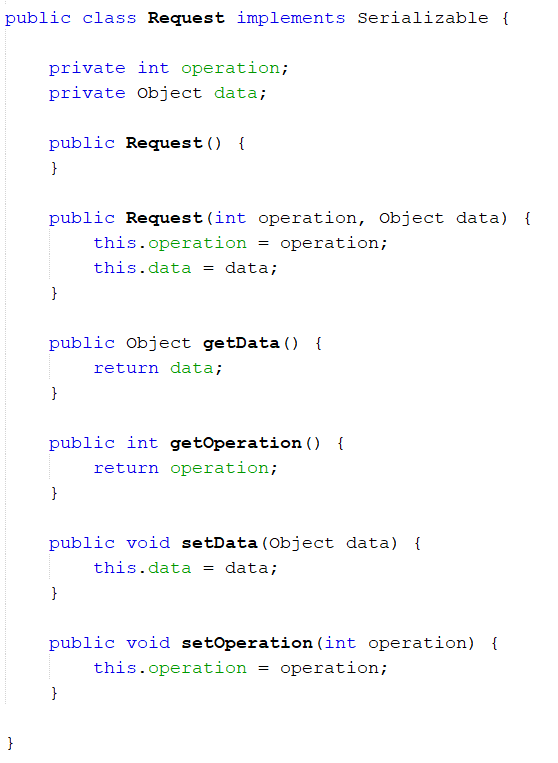
Слика Класа AbstractDomainObject

* Operation – интерфејс који садржи све операције које се шаљу од клијента серверу



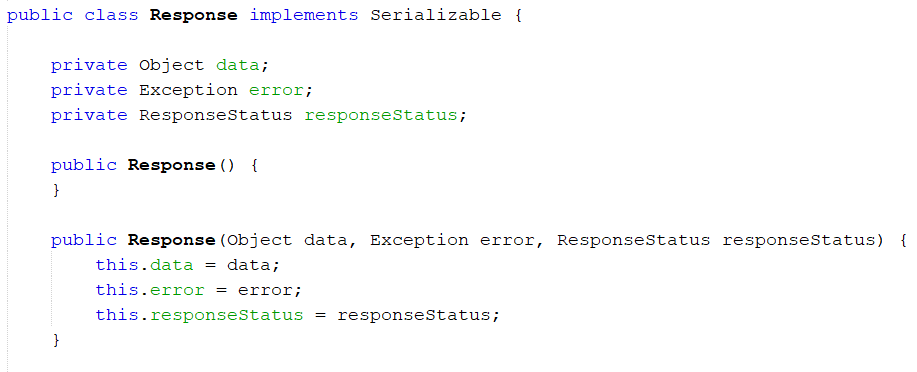
Слика Interface Operations

* Request – служи за слање објекта од клијента ка серверу. Садржи један *Object* атрибут који представља објекат над којим треба извршити захтевану операцију и један *int* атрибут који представља операцију која треба да се изврши.



Слика Класа Request

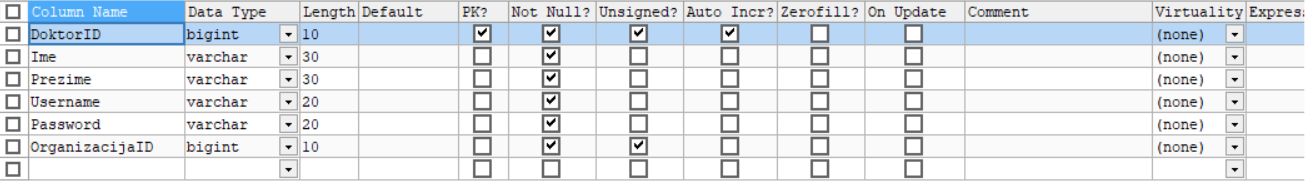
* Response – служи за слање објекта од сервера ка клијенту. Садржи један *Object* атрибут који представља резултат извршене операције, један *Exception* атрибут који представља изузетак који се можда десио.



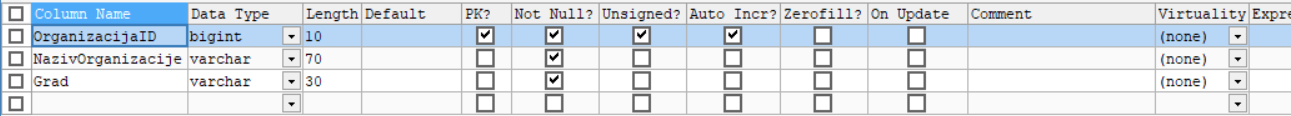
Слика Класа Response

### 3.2.3 Пројектовање складишта података

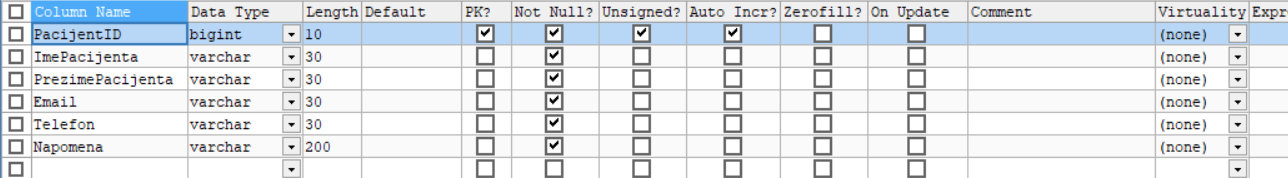
На основу релационог модела и ограничења пројектоване су табеле базе података које користи наш софтверски систем:



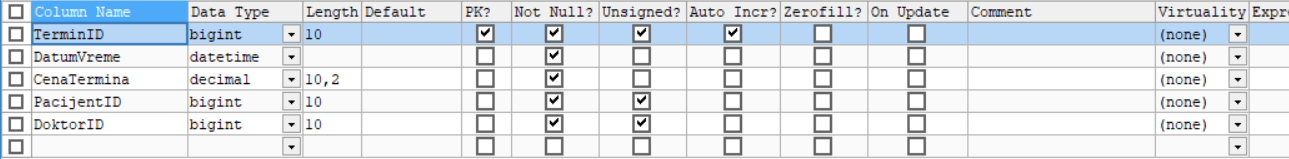
Слика Табела Доктор



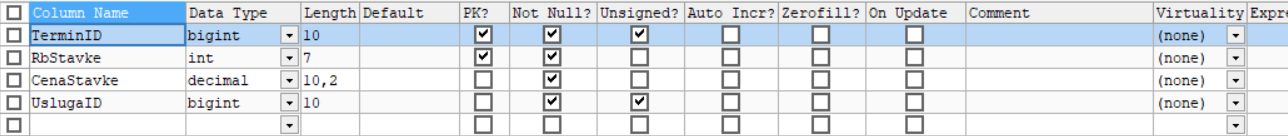
Слика Табела ЗдравственаОрганизација



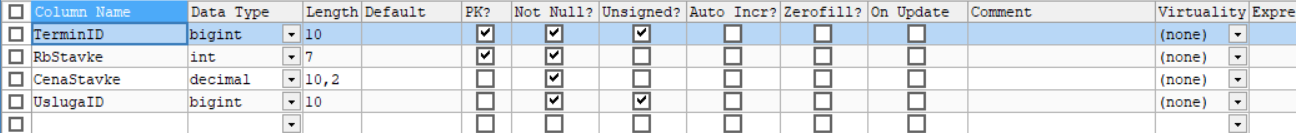
Слика Табела Пацијент



Слика Табела Термин



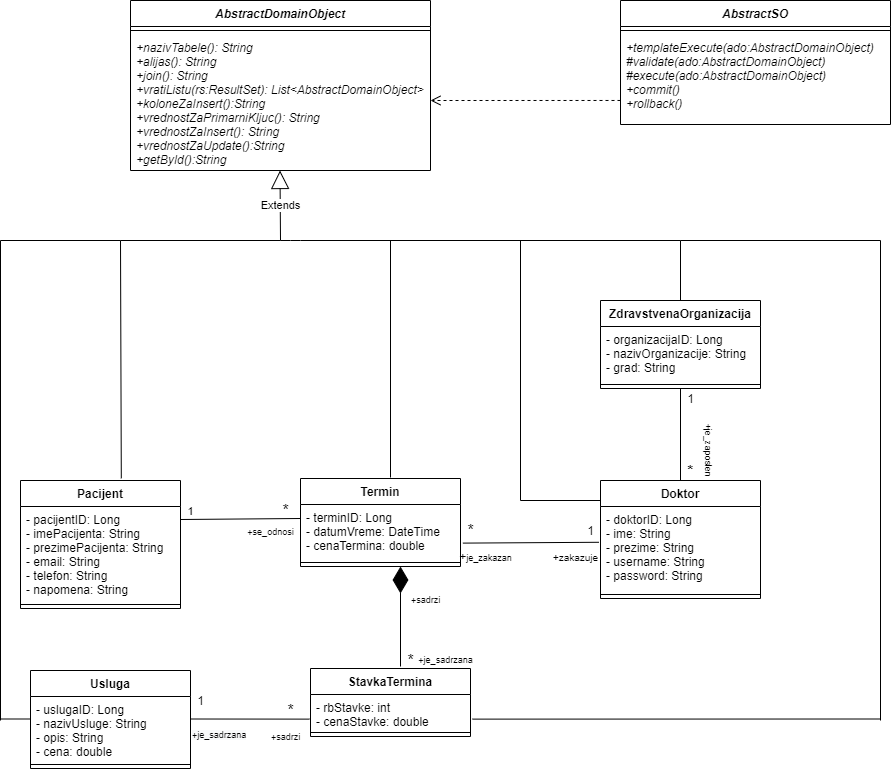
Слика Табела СтавкаТермина



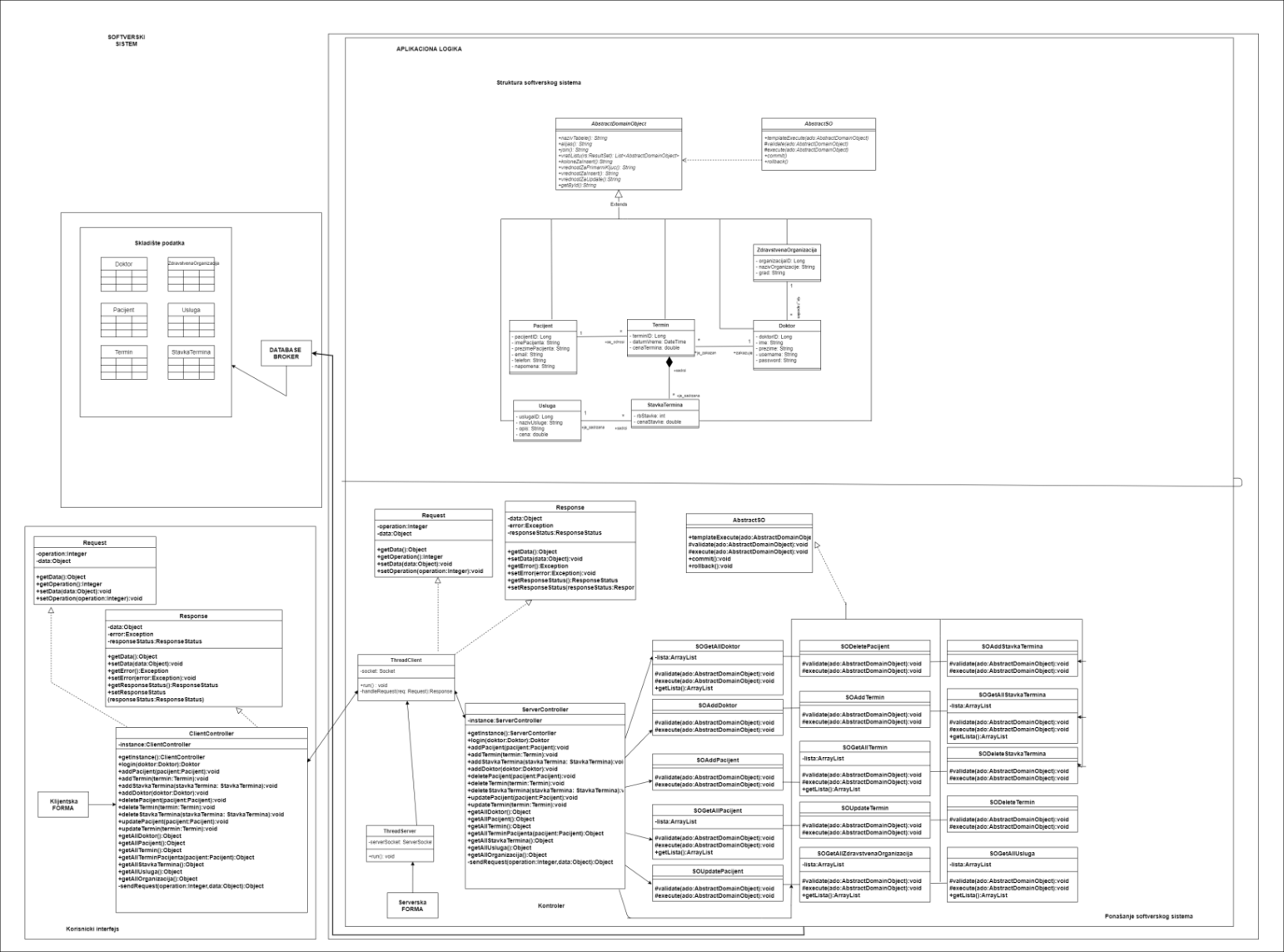
Слика Табела Услуга

За комуникацију са базом података правимо генеричку класу *AbstractSO*која има методе за валидацију и извршавања трансакције, које су апстрактне и које ће имплементирати свака класа системске операције која ће се извршавати, као и методе за потврђивање и поништавање трансакције. Она се служи класом *Konekcija* која је имплементирана помоћу *Singleton* патерна и која параметре за повезивање на базу података чита из текстуалног фајла који садржи све потребне параметре и помоћу ње наша генеричка класа добија конекцију на базу података.

Као резултат пројектовања класе *AbstractSO, DatabaseConnection* и доменских објеката добијамо следећи дијаграм класа:



Слика Дијаграм класа добијен након пројектовања доменских класа и AbstractSOклаce

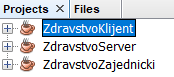


Слика Коначна архитектура софтверског система

# 4. Фаза имплементације

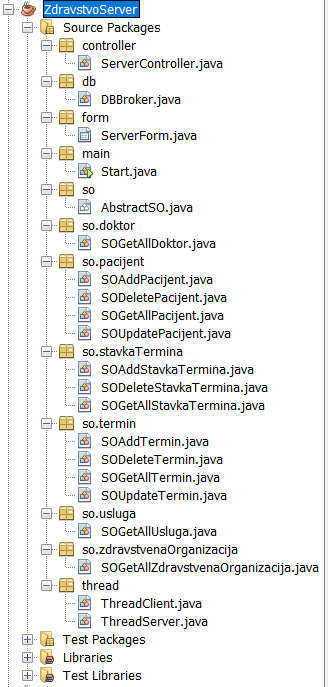
Софтверски систем је развијан у програмском језику “Java”, развојно окружење NetBeans. Као систем за управљање базом података коришћен je MySQL.

Организација пројеката је приказана на следећој слици.



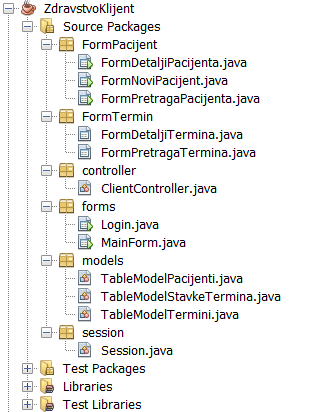
Слика Организација пројекта

Пројекат *ZdravstvoServer* садржи нити за покретање комуникације са клијентом, контролера, генеричку класу за базу података, класу за генерисање конекције на базу података и серверске форме неопходне за покретање сервера и приказивање тренутно повезаних корисника на сервер



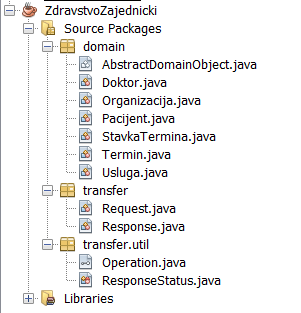
Слика Серверски део

Пројекат *ZdravstvoKlijent* садржи форме на којима запослени ради и сокет који служи за комуникацију са сервером.



Слика Клијентски део

Пројекат *ZdravstvoZajednicki* садржи заједничке класе које користе клијент и сервер пројекти. То су доменске класе, класе *Request* и *Response* и интерфејс *Operation* и енум *ResponseStatus*



Слика Заједнички део

# 5. Тестирање

У фази тестирања, тестиран је сваки од имплементираних случајева коришћења. Приликом тестирања сваког случаја коришћења, поред унетих правилних података, уношени су и неправилни подаци да би се утврдио резултат извршења. Након фазе тестирања, софтвер је спреман за коришћење од стране крајњег корисника

# 6. Закључак

За развој софтверског система за Здравствени систем, коришћена је поједностављена Ларманова метода за развој софтвера. Тренутно развијено софтверско решење јесте применљиво, али такође оставља пуно простора, да се коришћењем савремених технологија, побољшају и отклоне потенцијални недостаци, као и простора да се надограде нове функционалности које би задовољиле потребе корисника, пруживши му већу употребну вредност и доживљај.

# Коришћена литература

”ПРОЈЕКТОВАЊЕ СОФТВЕРА СКРИПТА- радни материјал, вер. 1.3”, др Синиша Влајић, Београд 2020