**Универзитет у Београду**

**Факултет организационих наука**

**Лабораторија за софтверско инжењерство**

Семинарски рад из предмета пројектовање софтвера

**Софтверски систем за евиденцију уписаних ученика у музичку школу**

**Професор:** **Студент:**

Др Синиша Влајић Страхиња Хаџи-Тонић 73/13

**Београд, 2019.**

Садржај

[**1. Кориснички захтеви** 6](#_Toc12164117)

[1.1 Вербални опис 6](#_Toc12164118)

[1.2 Случајеви коришћења 7](#_Toc12164119)

[1.2.1. СК1: Унос новог ученика 8](#_Toc12164120)

[1.2.2. СК2: Претраживање ученика 8](#_Toc12164121)

[1.2.3. СК3: Измена ученика 9](#_Toc12164122)

[1.2.4. СК4: Брисање ученика 10](#_Toc12164123)

[1.2.5. СК5: Унос новог професора 11](#_Toc12164124)

[1.2.6. СК6: Претраживање професора 12](#_Toc12164125)

[1.2.7. СК7: Брисање професора 13](#_Toc12164126)

[1.2.8. СК8: Унос новог предавања (Сложен СК) 14](#_Toc12164127)

[1.2.9. СК9: Претраживање предавања 14](#_Toc12164128)

[1.2.10. СК10: Брисање предавања 15](#_Toc12164129)

[2. Анализа 16](#_Toc12164130)

[2.1. Понашање софтверског система – Дијаграм секвенци случаја коришћења 16](#_Toc12164131)

[ДС 1: Дијаграм секвенци случаја коришћења - Унос новог ученика 16](#_Toc12164132)

[ДС 2: Дијаграм секвенци случаја коришћења - Претраживање ученика 18](#_Toc12164133)

[ДС 3: Дијаграм секвенци случаја коришћења - Измена ученика 20](#_Toc12164134)

[ДС 4: Дијаграм секвенци случаја коришћења - Брисање ученика 22](#_Toc12164135)

[ДС 5: Дијаграм секвенци случаја коришћења - Унос новог професора 23](#_Toc12164136)

[ДС 6: Дијаграм секвенци случаја коришћења - Претраживање професора 25](#_Toc12164137)

[ДС 7: Дијаграм секвенци случаја коришћења - Брисање професора 27](#_Toc12164138)

[ДС 8: Дијаграм секвенци случаја коришћења - Унос новог предавања 28](#_Toc12164139)

[ДС 9: Дијаграм секвенци случаја коришћења - Претраживање предавања 33](#_Toc12164140)

[ДС 10: Дијаграм секвенци случаја коришћења - Брисање предавања 35](#_Toc12164141)

[2.2. Понашање софтверског система – Дефинисање уговора о системским операцијама 37](#_Toc12164142)

[1. Уговор УГ1: СачувајУченика 37](#_Toc12164143)

[2. Уговор УГ2: ЗапамтиУченика 38](#_Toc12164144)

[3. Уговор УГ3: ПронађиУченике 38](#_Toc12164145)

[4. Уговор УГ4: ПронађиУченика 38](#_Toc12164146)

[5. Уговор УГ5: ОбришиУченика 38](#_Toc12164147)

[6. Уговор УГ6: СачувајПрофесора 38](#_Toc12164148)

[7. Уговор УГ7: ПронађиПрофесоре 39](#_Toc12164149)

[8. Уговор УГ8: ПронађиПрофесора 39](#_Toc12164150)

[19. Уговор УГ19: ОбришиПрофесора 39](#_Toc12164151)

[10. Уговор УГ10: ВратиСвеУченике 39](#_Toc12164152)

[11. Уговор УГ11: ВратиСвеПрофесоре 40](#_Toc12164153)

[12. Уговор УГ12: ВратиСвеПредмете 40](#_Toc12164154)

[13. Уговор УГ13: ВратиСвеОдсеке 40](#_Toc12164155)

[14. Уговор УГ14: КреирајПредавање 40](#_Toc12164156)

[15. Уговор УГ15: ЗапамтиПредавање 40](#_Toc12164157)

[16. Уговор УГ16: ПронађиПредавања 41](#_Toc12164158)

[17. Уговор УГ17: ПронађиПредавање 41](#_Toc12164159)

[18. Уговор УГ18: ОбришиПредавање 41](#_Toc12164160)

[2.3. Структура софтверког система – Концептуални (доменски) модел 41](#_Toc12164161)

[2.4. Структура софтверског система – Релациони модел 42](#_Toc12164162)

[3. Пројектовање 46](#_Toc12164163)

[3.1.Архиктетура софтверског система 46](#_Toc12164164)

[3.2.Пројектовање софверског система 48](#_Toc12164166)

[3.3.Пројектовање екранских форми 48](#_Toc12164167)

[3.3.1. СК1: Унос новог ученика 48](#_Toc12164168)

[3.3.2. СК2: Претраживање ученика 50](#_Toc12164169)

[3.3.3. СК3: Измена ученика 53](#_Toc12164170)

[3.3.4. СК4: Брисање ученика 55](#_Toc12164171)

[3.3.5. СК5: Унос новог професора 57](#_Toc12164172)

[3.3.6. СК6: Претраживање професора 58](#_Toc12164173)

[3.3.7. СК7: Брисање професора 61](#_Toc12164174)

[3.3.8. СК8: Унос новог предавања (Сложен СК) 62](#_Toc12164175)

[3.3.9. СК9: Претраживање предавања 65](#_Toc12164176)

[3.3.10. СК10: Брисање предавања 68](#_Toc12164177)

[3.3.11. Пројектовање контролера корисничког интерфејса 69](#_Toc12164178)

[3.4. Пројектовање апликације 70](#_Toc12164180)

[3.4.1. Контролер апликационе логике 71](#_Toc12164181)

[3.4.2. Системске операције 91](#_Toc12164182)

[3.4.2.1. Уговор УГ1: СачувајУченика 92](#_Toc12164183)

[3.4.2.2. Уговор УГ2: ЗапамтиУченика 92](#_Toc12164184)

[3.4.2.3. Уговор УГ3: ПронађиУченике 92](#_Toc12164185)

[3.4.2.4. Уговор УГ4: ПронађиУченика 93](#_Toc12164186)

[3.4.2.5. Уговор УГ5: ОбришиУченика 93](#_Toc12164187)

[3.4.2.6. Уговор УГ6: СачувајПрофесора 93](#_Toc12164188)

[3.4.2.7. Уговор УГ7: ПронађиПрофесоре 94](#_Toc12164189)

[3.4.2.8. Уговор УГ8: ПронађиПрофесора 94](#_Toc12164190)

[3.4.2.9. Уговор УГ9: ОбришиПрофесора 95](#_Toc12164191)

[3.4.2.10. Уговор УГ10: ВратиСвеУченике 95](#_Toc12164192)

[3.4.2.11. Уговор УГ11: ВратиСвеПрофесоре 96](#_Toc12164193)

[3.4.2.12. Уговор УГ12: ВратиСвеПредмете 96](#_Toc12164194)

[3.4.2.13. Уговор УГ13: ВратиСвеОдсеке 96](#_Toc12164195)

[3.4.2.14. Уговор УГ14: КреирајПредавање 97](#_Toc12164196)

[3.4.2.15. Уговор УГ15: ЗапамтиПредавање 97](#_Toc12164197)

[3.4.2.16. Уговор УГ16: ПронађиПредавања 98](#_Toc12164198)

[3.4.2.17. Уговор УГ17: ПронађиПредавање 98](#_Toc12164199)

[3.4.2.18. Уговор УГ18: ОбришиПредавање 99](#_Toc12164200)

[3.4.3. Брокер базе података 99](#_Toc12164201)

[3.4.4. Пројектовање складишта података 100](#_Toc12164202)

[4. Имплементација 101](#_Toc12164203)

[5. Тестирање 103](#_Toc12164204)

[6. Литература 104](#_Toc12164205)

# **1. Кориснички захтеви**

### 1.1 Вербални опис

Потребно је направити софтверски систем који ће омогућити евиденцију уписаних ученика у музичку школу.

Систем треба да омогући запосленом лицу да бележи текуће ученике и професоре, као и све нове. Систем треба да води евиденцију о уносу нових ученика и професора, измени података о ученицима и предавањима, као и претраживању свих постојећих ученика. Постоји и могућност брисања ученика, уколико жели да се испише из школе, као и професора, уколико је дао отказ. Такође је потребно водити евиденцију о присуству ученика на предавањима које држе професори.

Коришћење апликације је ограничено на запослене који раде у администрацији.

# 1.2 Случајеви коришћења

У конкретном случају идентификовани су следећи случајеви коришћења који су приказани на слици 1:

1. Унос новог ученика
2. Претраживање ученика
3. Измена ученика
4. Брисање ученика
5. Унос новог професора
6. Измена професора
7. Унос новог предавања (сложен СК)
8. Претраживање предавања
9. Брисање предавања

### 1.2.1. СК1: Унос новог ученика

**Назив СК**

Креирање новог ученика

**Актори СК**

Запослено лице

**Учесници СК**

Запослено лице и систем (програм)

**Предуслов**: Систем је укључен и запослено лице je улоговано под својом шифром. Систем приказује форму за унос ученика. Учитана је листа одсека.

**Основни сценарио СК**

1. Запослено лице **уноси** податке у ученика. (АПУСО)
2. Запослено лице **контролише** да ли је коректно унео податке у ученика. (АНСО)
3. Запослено лице **позива** систем да сачува податке о ученику. (АПСО)
4. Систем **памти** податке о ученику. (СО)
5. Систем **приказује** запосленом лицу сачуваног ученика и поруку: “Систем је запамтио ученика“. (ИА)

Алтернативна сценарија

5.1 Уколико систем не може да запамти податке о ученику он приказује запосленом лицу поруку “Систем не може да сачува ученика”. (ИА)

### 1.2.2. СК2: Претраживање ученика

**Назив СК**

Претраживање ученика

**Актори СК**

Запослено лице

**Учесници СК**

Запослено лице и систем (програм)

**Предуслов**: Систем је укључен и запослено лице je улоговано под својом шифром. Систем приказује форму за претраживање ученика.

**Основни сценарио СК**

1. Запослено лице **уноси** вредностпо којој претражује ученике. (АПУСО)
2. Запослено лице **позива** систем да нађе ученике по задатој вредности. (АПСО)
3. Систем **тражи** ученике по задатој вредности. (СО)
4. Систем **приказује** запосленом лицу ученике (ИА)
5. Запослено лице **бира** ученика чије податке жели да види. (АПУСО)
6. Запослено лице **позива** систем да прикаже податке о одабраном ученику. (АПСО)
7. Систем **учитава** податке о одабраном ученику. (СО)
8. Систем **обавештава** запослено лице о успешном учитавању података о ученику и поруку: “Систем је успешно учитао ученика”. (ИА)

Алтернативна сценарија

4.1 Уколико систем не може да нађе ученика он приказује запосленом лицу поруку: “Систем не може да нађе ученикe по задатој вредности”. (ИА)

8.1 Уколико систем не може да нађе ученика он приказује запосленом лицу поруку “Систем не може да учита ученика”. (ИА)

### 1.2.3. СК3: Измена ученика

**Назив СК**

Промена ученика

**Актори СК**

Запослено лице

**Учесници СК**

Запослено лице и систем (програм)

**Предуслов**: Систем је укључен и Запослено лице je улоговано под својом шифром. Систем приказује форму за рад са учеником након што је претходно извршено претраживање .

**Основни сценарио СК**

1. Запослено лице **уноси** вредностпо којој претражује ученике. (АПУСО)
2. Запослено лице **позива** систем да нађе ученике по задатој вредности. (АПСО)
3. Систем **тражи** ученике по задатој вредности. (СО)
4. Систем **приказује** запосленом лицу ученике (ИА)
5. Запослено лице **бира** ученика чије податке жели да види. (АПУСО)
6. Запослено лице **позива** систем да прикаже податке о одабраном ученику. (АПСО)
7. Систем **учитава** податке о одабраном ученику. (СО)
8. Систем **обавештава** запослено лице о успешном учитавању података о ученику и поруку: “Систем је успешно учитао ученика”. (ИА)
9. Запослено лице **уноси** **(мења)** податке о ученику. (АПУСО)
10. Запослено лице **контролише** да ли је коректно унео податке о ученику. (АНСО)
11. Запослено лице **позива** систем да запамти податке о ученику. (АПСО)
12. Систем **памти** податке о ученику. (СО)
13. Систем **приказује** Запосленом лицу запамћени ученика и поруку: “Систем је запамтио ученика.” (ИА)

Алтернативна сценарија

4.1 Уколико систем не може да нађе ученика он приказује запосленом лицу поруку: “Систем не може да нађе ученикe по задатој вредности”. (ИА)

8.1 Уколико систем не може да нађе ученика он приказује запосленом лицу поруку “Систем не може да учита ученика”. (ИА)

13.1 Уколико систем не може да запамтиподатке о ученику он приказује запосленом лицу поруку: “Систем не може да запамти ученика”. (ИА)

### 1.2.4. СК4: Брисање ученика

**Назив СК**

Брисање ученика

**Актори СК**

Запослено лице

**Учесници СК**

Запослено лице и систем (програм)

**Предуслов**: Систем је укључен и запослено лице je улогован под својом шифром. Систем приказује форму за рад са учеником након што је претходно извршено претраживање .

**Основни сценарио СК**

1. Запослено лице **уноси** вредностпо којој претражује ученике. (АПУСО)
2. Запослено лице **позива** систем да нађе ученике по задатој вредности. (АПСО)
3. Систем **тражи** ученике по задатој вредности. (СО)
4. Систем **приказује** запосленом лицу ученике (ИА)
5. Запослено лице **бира** ученика чије податке жели да види. (АПУСО)
6. Запослено лице **позива** систем да прикаже податке о одабраном ученику. (АПСО)
7. Систем **учитава** податке о одабраном ученику. (СО)
8. Систем **обавештава** запослено лице о успешном учитавању података о ученику и поруку: “Систем је успешно учитао ученика”. (ИА)
9. Запослено лице **позива** систем да обрише ученика. (АПСО)
10. Систем **брише** ученика. (СО)
11. Систем **приказује** запосленом лицу поруку: “Систем је обрисао ученика.” (ИА)

Алтернативна сценарија

4.1 Уколико систем не може да нађе ученика он приказује запосленом лицу поруку: “Систем не може да нађе ученикe по задатој вредности”. (ИА)

8.1 Уколико систем не може да нађе ученика он приказује запосленом лицу поруку “Систем не може да учита ученика”. (ИА)

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

### 1.2.5. СК5: Унос новог професора

**Назив СК**

Креирање новог професора

**Актори СК**

Запослено лице

**Учесници СК**

Запослено лице и систем (програм)

**Предуслов**: Систем је укључен и запослено лице je улоговано под својом шифром. Систем приказује форму за унос професора. Учитана је листа предмета.

**Основни сценарио СК**

1. Запослено лице **уноси** податке у професора. (АПУСО)
2. Запослено лице **контролише** да ли је коректно унео податке у професора. (АНСО)
3. Запослено лице **позива** систем да сачува податке о професору. (АПСО)
4. Систем **памти** податке о професору. (СО)
5. Систем **приказује** запосленом лицу запамћеног професора и поруку: “Систем је сачувао професора“. (ИА)

Алтернативна сценарија

* 1. Уколико систем не може да запамти податке о професора он приказује заосленом лицу поруку “Систем не може да сачува професора ”. (ИА)

### 1.2.6. СК6: Претраживање професора

**Назив СК**

Претраживање професора

**Актори СК**

Запослено лице

**Учесници СК**

Запослено лице и систем (програм)

**Предуслов**: Систем је укључен и запослено лице je улоговано под својом шифром. Систем приказује форму за претраживање професора.

**Основни сценарио СК**

1. Запослено лице **уноси** вредностпо којој претражује професоре. (АПУСО)
2. Запослено лице **позива** систем да нађе професоре по задатој вредности. (АПСО)
3. Систем **тражи** професоре по задатој вредности. (СО)
4. Систем **приказује** запосленом лицу профеосре (ИА)
5. Запослено лице **бира** професора чије податке жели да види. (АПУСО)
6. Запослено лице **позива** систем да прикаже податке о одабраном професору. (АПСО)
7. Систем **учитава** податке о одабраном професору. (СО)
8. Систем **обавештава** запослено лице о успешном учитавању података о професору и поруку: “Систем је успешно учитао професора”. (ИА)

Алтернативна сценарија

4.1 Уколико систем не може да нађе професора он приказује запосленом лицу поруку: “Систем не може да нађе професоре по задатој вредности”. (ИА)

8.1 Уколико систем не може да нађе професора он приказује запосленом лицу поруку “Систем не може да учита професора”. (ИА)

### 1.2.7. СК7: Брисање професора

**Назив СК**

Брисање професора

**Актори СК**

Запослено лице

**Учесници СК**

Запослено лице и систем (програм)

**Предуслов**: Систем је укључен и запослено лице je улогован под својом шифром. Систем приказује форму за рад са професором након што је претходно извршено претраживање .

**Основни сценарио СК**

1. Запослено лице **уноси** вредностпо којој претражује професоре. (АПУСО)
2. Запослено лице **позива** систем да нађе професоре по задатој вредности. (АПСО)
3. Систем **тражи** професоре по задатој вредности. (СО)
4. Систем **приказује** запосленом лицу профеосре (ИА)
5. Запослено лице **бира** професора чије податке жели да види. (АПУСО)
6. Запослено лице **позива** систем да прикаже податке о одабраном професору. (АПСО)
7. Систем **учитава** податке о одабраном професору. (СО)
8. Систем **обавештава** запослено лице о успешном учитавању података о професору и поруку: “Систем је успешно учитао професора”. (ИА)
9. Запослено лице **позива** систем да обрише професора. (АПСО)
10. Систем **брише** професора. (СО)
11. Систем **приказује** запосленом лицу поруку: “Систем је обрисао професора.” (ИА)

Алтернативна сценарија

4.1 Уколико систем не може да нађе професора он приказује запосленом лицу поруку: “Систем не може да нађе професоре по задатој вредности”. (ИА)

8.1 Уколико систем не може да нађе професора он приказује запосленом лицу поруку “Систем не може да учита професора”. (ИА)

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

### 1.2.8. СК8: Унос новог предавања (Сложен СК)

**Назив СК**

Креирање новог предавања

**Актори СК**

Запослено лице

**Учесници СК**

Запослено лице и систем (програм)

**Предуслов**: Систем је укључен и запослено лице je улоговано под својом шифром. Систем приказује форму за унос предавања. Учитане су листе предмета, професора и ученика.

**Основни сценарио СК**

1. Запослено лице **позива** систем да креира предавање. (АПСО)
2. Систем **креира** предавање. (СО)
3. Систем **приказује** запосленом лицу предавање и поруку: “Систем је креирао предавање“. (ИА)
4. Запослено лице **уноси** податке о предавању. (АПУСО)
5. Запослено лице **контролише** да ли је коректно унео податке о предавање. (АНСО)
6. Запослено лице **позива** систем да запамти податке о предавању. (АПСО)
7. Систем **памти** податке о предавању. (СО)
8. Систем **приказује** запосленом лицу запамћену предавање и поруку: “Систем је запамтио предавање“. (ИА)

Алтернативна сценарија

3.1 Уколико систем не може да креира предавање он приказује запосленом лицу поруку: “Систем не може да креира предавање ”. Прекида се извршење сценарија. (ИА)

8.1 Уколико систем не може да запамти податке о предавању он приказује заосленом лицу поруку “Систем не може да запамти предавање ”. (ИА)

### 1.2.9. СК9: Претраживање предавања

**Назив СК**

Претраживање предавања

**Актори СК**

Запослено лице

**Учесници СК**

Запослено лице и систем (програм)

**Предуслов**: Систем је укључен и запослено лице je улоговано под својом шифром. Систем приказује форму за претраживање предавања.

**Основни сценарио СК**

1. Запослено лице **уноси** вредностпо којој претражује предавања. (АПУСО)
2. Запослено лице **позива** систем да нађе предавања по задатој вредности. (АПСО)
3. Систем **тражи** предавања по задатој вредности. (СО)
4. Систем **приказује** запосленом лицу предавања (ИА)
5. Запослено лице **бира** предавања чије податке жели да види. (АПУСО)
6. Запослено лице **позива** систем да прикаже податке о одабраном предавању. (АПСО)
7. Систем **учитава** податке о одабраном предавању. (СО)
8. Систем **обавештава** запослено лице о успешном учитавању података о предавању и поруку: “Систем је успешно учитао предавање”. (ИА)

Алтернативна сценарија

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

8.1 Уколико систем не може да нађе предавање он приказује запосленом лицу поруку “Систем не може да учита предавање”. (ИА)

### 1.2.10. СК10: Брисање предавања

**Назив СК**

Брисање предавања

**Актори СК**

Запослено лице

**Учесници СК**

Запослено лице и систем (програм)

**Предуслов**: Систем је укључен и запослено лице je улогован под својом шифром. Систем приказује форму за рад са предавањем након што је претходно извршено претраживање .

**Основни сценарио СК**

1. Запослено лице **уноси** вредностпо којој претражује предавања. (АПУСО)
2. Запослено лице **позива** систем да нађе предавања по задатој вредности. (АПСО)
3. Систем **тражи** предавања по задатој вредности. (СО)
4. Систем **приказује** запосленом лицу предавања (ИА)
5. Запослено лице **бира** предавања чије податке жели да види. (АПУСО)
6. Запослено лице **позива** систем да прикаже податке о одабраном предавању. (АПСО)
7. Систем **учитава** податке о одабраном предавању. (СО)
8. Систем **обавештава** запослено лице о успешном учитавању података о предавању и поруку: “Систем је успешно учитао предавање”. (ИА)
9. Запослено лице **позива** систем да обрише предавање. (АПСО)
10. Систем **брише** предавање. (СО)
11. Систем **приказује** запосленом лицу поруку: “Систем је обрисао предавање.” (ИА)

Алтернативна сценарија

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

8.1 Уколико систем не може да нађе предавање он приказује запосленом лицу поруку “Систем не може да учита предавање”. (ИА)

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

# 2. Анализа

## 2.1. Понашање софтверског система – Дијаграм секвенци случаја коришћења

### ДС 1: Дијаграм секвенци случаја коришћења - Унос новог ученика

**Предуслов:** Систем је укључен и запослено лице је улоговано под својом шифром. Систем приказује форму за рад са учеником.

**Основни сценарио СК:**

1. Форма позива систем да учита листу одсека. (АПСО)

2. Систем приказује листу одсека и поруку: „Систем је вратио листу одсека“. (ИА)

3. Запослено лице позива систем да запамти податке о ученику. (АПСО)

4. Систем приказује запосленом лицу запамћеног ученика и поруку: „Систем је запамтио ученика.” (ИА)



**Алтернативна сценарија:**

2.1. Уколико систем не може да врати листу одсека, он приказује запосленом лицу поруку:“Систем не може да врати листу одсека”. Прекида се извршење сценарија. (ИА)



4.1. Уколико систем не може да сачува податке о ученику, он приказује запосленом лицу поруку: “Систем не може да сачува ученика.” (ИА)



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

1. сигнал **СачувајУченика(Ученик)**
2. сигнал **ВратиСвеОдсеке(Лист<Одсек>)**

### ДС 2: Дијаграм секвенци случаја коришћења - Претраживање ученика

**Предуслов:** Систем је укључен и запослено лице је улоговано под својом шифром. Систем приказује форму за претраживање учеником.

**Основни сценарио СК:**

1. Запослено лице позива систем да нађе ученике по задатој вредности. (АПСО)

2. Систем приказује запосленом лицу ученике. (ИА)

3. Запослено лице позива систем да прикаже податке о одабраном ученику. (АПСО)

4. Систем обавештава запослено лице о успешном учитавању података о ученику поруком: „Систем је успешно учитао ученика.” и приказује податке о одабраном ученику. (ИА)



**Алтернативна сценарија:**

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



4.1. Уколико систем не може да нађе ученика, он приказује запосленом лицу поруку: “Систем не може да учита изабраног ученика.” (ИА)



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

1. сигнал **ПронађиУченике(Ученик, Лист<Ученик>)**

2. сигнал **ПронађиУченика(Ученик)**

### ДС 3: Дијаграм секвенци случаја коришћења - Измена ученика

**Предуслов:** Систем је укључен и запослено лице је улоговано под својом шифром. Систем приказује форму за рад са учеником након што је претходно извршено претраживање. Учитана је листа одсека.

**Основни сценарио СК:**

1. Запослено лице позива систем да нађе ученике по задатој вредности. (АПСО)

2. Систем приказује запосленом лицу ученике. (ИА)

3. Запослено лице позива систем да прикаже податке о одабраном ученику. (АПСО)

4. Систем обавештава запослено лице о успешном учитавању података о ученику поруком: „Систем је успешно учитао ученика.” и приказује податке о одабраном ученику. (ИА)

5. Форма позива систем да учита листу одсека. (АПСО)

6. Систем приказује листу одсека. (ИА)

7. Запослено лице позива систем да запамти податке о ученику. (АПСО)

8. Систем приказује запосленом лицу поруку: ''Систем је успешно запамтио ученика.'' (ИА)



**Алтернативна сценарија:**

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



4.1. Уколико систем не може да нађе ученика, он приказује запосленом лицу поруку: “Систем не може да учита изабраног ученика.” (ИА)



6.1. Уколико систем не може да врати листу одсека, он приказује запосленом лицу поруку:“Систем не може да врати листу одсека”. Прекида се извршење сценарија. (ИА)



8.1. Уколико систем не може да запамти податке о ученику, он приказује запосленом лицу следећу поруку: ''Систем не може да запамти ученика''(ИА)



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

1. сигнал **ЗапамтиУченика(Ученик)**
2. сигнал **ВратиСвеОдсеке(Лист<Одсек>)**

3. сигнал **ПронађиУченике(Ученик, Лист<Ученик>)**

4. сигнал **ПронађиУченика(Ученик)**

### ДС 4: Дијаграм секвенци случаја коришћења - Брисање ученика

**Предуслов:** Систем је укључен и запослено лице је улоговано под својом шифром. Систем приказује форму за рад са учеником након што је претходно извршено претраживање.

**Основни сценарио СК:**

1. Запослено лице позива систем да нађе ученике по задатој вредности. (АПСО)

2. Систем приказује запосленом лицу ученике. (ИА)

3. Запослено лице позива систем да прикаже податке о одабраном ученику. (АПСО)

4. Систем обавештава запослено лице о успешном учитавању података о ученику поруком: „Систем је успешно учитао ученика.” и приказује податке о одабраном ученику. (ИА)

5. Запослено лице позива систем да обрише ученика. (АПСО)

6. Систем приказује запосленом лицу поруку: ''Систем је успешно обрисао ученика.'' (ИА)



**Алтернативна сценарија:**

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



4.1. Уколико систем не може да нађе ученика, он приказује запосленом лицу поруку: “Систем не може да учита изабраног ученика.” (ИА)



6.1. Уколико систем не може да обрише ученика, он запосленом лицу приказује поруку: ''Систем не може да обрише ученика.''(ИА)



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

1. сигнал **ОбришиУченика(Ученик)**
2. сигнал **ПронађиУченике(Ученик, Лист<Ученик>)**
3. сигнал **ПронађиУченика(Ученик)**

### ДС 5: Дијаграм секвенци случаја коришћења - Унос новог професора

**Предуслов:** Систем је укључен и запослено лице је улоговано под својом шифром. Систем приказује форму за рад са професором. Учитана је листа предмета.

**Основни сценарио СК:**

1. Форма позива систем да учита листу предмета. (АПСО)

2. Систем приказује листу предмета. (ИА)

3. Запослено лице позива систем да запамти податке о професору. (АПСО)

4. Систем приказује запосленом лицу запамћеног професора и поруку: „Систем је запамтио професора.” (ИА)



**Алтернативна сценарија:**

2.1. Уколико систем не може да врати листу предмета, он приказује запосленом лицу поруку:“Систем не може да врати листу предмета”. Прекида се извршење сценарија. (ИА)



4.1. Уколико систем не може да сачува податке о професору, он приказује запосленом лицу поруку: “Систем не може да сачува професора.” (ИА)



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

1. сигнал **СачувајПрофесора(Професор)**
2. сигнал **ВратиСвеПредмете(Лист<Предмет>)**

### ДС 6: Дијаграм секвенци случаја коришћења - Претраживање професора

**Предуслов:** Систем је укључен и запослено лице је улоговано под својом шифром. Систем приказује форму за претраживање професора.

**Основни сценарио СК:**

1. Запослено лице позива систем да нађе професоре по задатој вредности. (АПСО)

2. Систем приказује запосленом лицу професоре. (ИА)

3. Запослено лице позива систем да прикаже податке о одабраном професору. (АПСО)

4. Систем обавештава запослено лице о успешном учитавању података о професору поруком: „Систем је успешно учитао професора.” и приказује податке о одабраном професору. (ИА)



**Алтернативна сценарија:**

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



8.1. Уколико систем не може да нађе професора, он приказује запосленом лицу поруку: “Систем не може да учита изабраног професора.” (ИА)



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

1. сигнал **ПронађиПрофесоре(Професор, Лист<Професор>)**

2. сигнал **ПронађиПрофесора(Професор)**

### ДС 7: Дијаграм секвенци случаја коришћења - Брисање професора

**Предуслов:** Систем је укључен и запослено лице је улоговано под својом шифром. Систем приказује форму за рад са професором након што је претходно извршено претраживање.

**Основни сценарио СК:**

1. Запослено лице позива систем да нађе професоре по задатој вредности. (АПСО)

2. Систем приказује запосленом лицу професоре. (ИА)

3. Запослено лице позива систем да прикаже податке о одабраном професору. (АПСО)

4. Систем обавештава запослено лице о успешном учитавању података о професорупоруком: „Систем је успешно учитао професора.” и приказује податке о одабраном професору. (ИА)

5. Запослено лице позива систем да обрише професора. (АПСО)

6. Систем приказује запосленом лицу поруку: ''Систем је успешно обрисао професора.'' (ИА)



**Алтернативна сценарија:**

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



4.1. Уколико систем не може да нађе професора, он приказује запосленом лицу поруку: “Систем не може да учита изабраног професора.” (ИА)



6.1. Уколико систем не може да обрише професора, он запосленом лицу приказује поруку: ''Систем не може да обрише професор.''(ИА)



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

1. сигнал **ОбришиПрофесора(Професор)**
2. сигнал **ПронађиПрофесоре(Професор, Лист< Професор >)**
3. сигнал **ПронађиПрофесора(Професор)**

### ДС 8: Дијаграм секвенци случаја коришћења - Унос новог предавања

**Предуслов:** Систем је укључен и запослено лице је улоговано под својом шифром. Систем приказује форму за унос предавања. Учитане су листе предмета, професора и ученика.

**Основни сценарио СК:**

1. Форма позива систем да врати листу предмета. (АПСО)

2. Систем приказује листу предмета на форми. (ИА)

3. Форма позива систем да врати листу професора. (АПСО)

4. Систем приказује листу професора на форми. (ИА)

5. Форма позива систем да врати листу ученика. (АПСО)

6. Систем приказује листу ученика на форми. (ИА)

7. Запослено лице позива систем да креира ново предавање. (АПСО)

8. Систем приказује запосленом лицу ново предавање и поруку: “Систем је успешно креирао ново предавање”. (ИА)

9. Запослено лице позива систем да запамти ново предавање. (АПСО)

10. Систем приказује запосленом лицу запамћено предавање и поруку: “Систем је успешно запамтио ново предавање”. (ИА)



**Алтернативна сценарија:**

2.1. Уколико систем не може да врати листу предмета он приказује запосленом лицу поруку: „Систем не може да врати листу предмета“. Прекида се извршење сценарија. (ИА)



4.1. Уколико систем не може да врати листу професора он приказује запосленом лицу поруку: „Систем не може да врати листу професора“. Прекида се извршење сценарија. (ИА)



6.1. Уколико систем не може да врати листу ученика он приказује запосленом лицу поруку: „Систем не може да врати листу ученика“. Прекида се извршење сценарија. (ИА)



8.1. Уколико систем не може да креира ново предавање, он приказује запосленом лицу поруку: “Систем не може да креира ново предавање.”. Прекида се извршење сценарија. (ИА)



10.1. Уколико систем не може да запамти ново предавање, он приказује запосленом лицу поруку: “Систем не може да запамти ново предавање.”. Прекида се извршење сценарија. (ИА)



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

1. сигнал **ВратиСвеПредмете( Лист<Предмет>)**

2. сигнал **ВратиСвеПрофесоре(Лист<Професор>)**

3. сигнал **ВратиСвеУченике(Лист<Ученик>)**

4. сигнал **КреирајПредавање(Предавање)**

5. сигнал **ЗапамтиПредавање(Предавање)**

### ДС 9: Дијаграм секвенци случаја коришћења - Претраживање предавања

**Предуслов:** Систем је укључен и запослено лице је улоговано под својом шифром. Систем приказује форму за претраживање предавања.

**Основни сценарио СК:**

1. Запослено лице позива систем да нађе предавања по задатој вредности. (АПСО)

2. Систем приказује запосленом лицу предавања. (ИА)

3. Запослено лице позива систем да прикаже податке о одабраном предавању. (АПСО)

4. Систем обавештава запослено лице о успешном учитавању података о предавању поруком: „Систем је успешно учитао предавање.” и приказује податке о одабраном професору. (ИА)



**Алтернативна сценарија:**

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



4.1. Уколико систем не може да учита предавање, он приказује запосленом лицу поруку: “Систем не може да учита изабрано предавање.” (ИА)



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

1. сигнал **ПронађиПредавања(Предавање, Лист<Предавање>)**

2. сигнал **ПронађиПредавање (Предавање)**

### ДС 10: Дијаграм секвенци случаја коришћења - Брисање предавања

**Предуслов:** Систем је укључен и запослено лице је улоговано под својом шифром. Систем приказује форму за рад са предавањем након што је претходно извршено претраживање .

**Основни сценарио СК:**

1. Запослено лице позива систем да нађе предавања по задатој вредности. (АПСО)

2. Систем приказује запосленом лицу предавања. (ИА)

3. Запослено лице позива систем да прикаже податке о одабраном предавању. (АПСО)

4. Систем обавештава запослено лице о успешном учитавању података о предавању поруком: „Систем је успешно учитао предавање.” и приказује податке о одабраном професору. (ИА)

5. Запослено лице позива систем да обрише предавање. (АПСО)

6. Систем приказује запосленом лицу поруку: ''Систем је успешно обрисао предавање.'' (ИА)



**Алтернативна сценарија:**

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



4.1. Уколико систем не може да учита предавање, он приказује запосленом лицу поруку: “Систем не може да учита изабрано предавање.” (ИА)



6.1. Уколико систем не може да обрише предавање, он запосленом лицу приказује поруку: ''Систем не може да обрише предавање.''(ИА)



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

1. сигнал **ПронађиПредавања(Предавање, Лист<Предавање>)**

2. сигнал **ПронађиПредавање (Предавање)**

3. сигнал **ОбришиПредавање(Предавање)**

Као резултат анализе сценарија добијене су следеће системске операције које треба пројектовати:

1. сигнал **СачувајУченика(Ученик)**

2. сигнал **ЗапамтиУченика(Ученик)**

3. сигнал **ПронађиУченике(Ученик, Лист<Ученик>)**

4. сигнал **ПронађиУченика(Ученик)**

5. сигнал **ОбришиУченика(Ученик)**

6. сигнал **СачувајПрофесора(Професор)**

7. сигнал **ЗапамтиПрофесора(Професор)**

8. сигнал **ПронађиПрофесоре(Професор, Лист<Професор>)**

9. сигнал **ПронађиПрофесора(Професор)**

10. сигнал **ОбришиПрофесора(Професор)**

11. сигнал **ВратиСвеУченике( Лист<Ученик>)**

12. сигнал **ВратиСвеПрофесоре(Лист<Професор>)**

13. сигнал **ВратиСвеПредмете(Лист<Предмет>)**

14. сигнал **ВратиСвеОдсеке(Лист<Одсек>)**

15. сигнал **КреирајПредавање(Предавање)**

16. сигнал **ЗапамтиПредавање(Предавање)**

17. сигнал **ПронађиПредавања(Предавање, Лист<Предавање>)**

18. сигнал **ПронађиПредавање(Предавање)**

19. сигнал **ОбришиПредавање(Предавање)**

## 2.2. Понашање софтверског система – Дефинисање уговора о системским операцијама

### 1. Уговор УГ1: СачувајУченика

Операција: СачувајУченика(Ученик):сигнал;

Веза са СК: СК1

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

Постуслови: Сачуван је нови ученик.

### 2. Уговор УГ2: ЗапамтиУченика

Операција: ЗапамтиУченика(Ученик):сигнал;

Веза са СК: СК1, СК3

Предуслови: Просто вредносно ограничење над објектом Ученик мора бити задовољено.

Структурно ограничење над објектом Ученик мора бити задовољено.

Постуслови: Подаци о ученику су запамћени.

### 3. Уговор УГ3: ПронађиУченике

Операција: ПронађиУченике(Ученик, Лист<Ученик>):сигнал;

Веза са СК: СК2, СК3, СК4

Предуслови: -

Постуслови: -

### 4. Уговор УГ4: ПронађиУченика

Операција: ПронађиУченика(Ученик):сигнал;

Веза са СК: СК2, СК3, СК4

Предуслови: -

Постуслови: -

### 5. Уговор УГ5: ОбришиУченика

Операција: ОбришиУченика(Ученик):сигнал;

Веза са СК: СК4

Предуслови: Структурно ограничење над објектом Ученик мора бити задовољено.

Постуслови: Ученик је обрисан.

### 6. Уговор УГ6: СачувајПрофесора

Операција: СачувајПрофесора(Професор):сигнал;

Веза са СК: СК5

Предуслови: Вредносна и структурна ограничења над објектом Професорморају бити задовољена.

Постуслови: Сачуван је нови професор.

### 7. Уговор УГ7: ПронађиПрофесоре

Операција: ПронађиПрофесоре(Професор, Лист<Професор>):сигнал;

Веза са СК: СК6, СК7

Предуслови: -

Постуслови: -

### 8. Уговор УГ8: ПронађиПрофесора

Операција: ПронађиПрофесора(Професор):сигнал;

Веза са СК: СК6, СК7

Предуслови: -

Постуслови: -

### 19. Уговор УГ19: ОбришиПрофесора

Операција: ОбришиПрофесора(Професор):сигнал;

Веза са СК: СК7

Предуслови: Структурно ограничење над објектом Професор мора бити задовољено.

Постуслови: Професор је обрисан.

### 10. Уговор УГ10: ВратиСвеУченике

Операција: ВратиСвеУченике(Лист<Ученик>):сигнал;

Веза са СК: СК8

Предуслови: -

Постуслови: -

### 11. Уговор УГ11: ВратиСвеПрофесоре

Операција: ВратиСвеПрофесоре(Лист<Професор>):сигнал;

Веза са СК: СК8

Предуслови: -

Постуслови: -

### 12. Уговор УГ12: ВратиСвеПредмете

Операција: ВратиСвеПредмете(Лист<Предмет>):сигнал;

Веза са СК: СК5, СК8

Предуслови: -

Постуслови: -

### 13. Уговор УГ13: ВратиСвеОдсеке

Операција: ВратиСвеПредмете(Лист<Одсек>):сигнал;

Веза са СК: СК1, СК3

Предуслови: -

Постуслови: -

### 14. Уговор УГ14: КреирајПредавање

Операција: КреирајПредавање(Предавање):сигнал;

Веза са СК: СК8

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

Постуслови: Креирано је ново предавање.

### 15. Уговор УГ15: ЗапамтиПредавање

Операција: Запамти Предавање(Предавање):сигнал;

Веза са СК: СК8

Предуслови: Просто вредносно ограничење над објектом Предавање мора бити задовољено.

Структурно ограничење над објектом Предавање мора бити задовољено.

Постуслови: Подаци о предавању су запамћени.

### 16. Уговор УГ16: ПронађиПредавања

Операција: ПронађиПредавања(Предавање, Лист<Предавање>):сигнал;

Веза са СК: СК6, СК7

Предуслови: -

Постуслови: -

### 17. Уговор УГ17: ПронађиПредавање

Операција: ПронађиПредавање(Предавање):сигнал;

Веза са СК: СК6, СК7

Предуслови: -

Постуслови: -

### 18. Уговор УГ18: ОбришиПредавање

Операција: ОбришиПредавање(Предавање):сигнал;

Веза са СК: СК7

Предуслови: Структурно ограничење над објектом Предавање мора бити задовољено.

Постуслови: Предавање је обрисано.

## 2.3. Структура софтверког система – Концептуални (доменски) модел



## 2.4. Структура софтверског система – Релациони модел

**-Ученик(**ЈМБГУченика, ИмеУченика, ПрезимеУченика, ТелефонУченика, Година, ОдсекИД, Година**)**

**-Професор(**ЈМБГПрофесора, ИмеПрофесора, ПрезимеПрофесора, ТелефонПрофесора, Предмет**)**

**-Одсек(**ОдсекИД, НазивОдсека, Трајање**)**

**-Предмет(**ПредметИД, НазивПредмета**)**

**-Евиденција(**ЕвиденцијаИД, ЈМБГУченика, ЈМБГПрофесора, ДатумЕвиденције**)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Taбела Ученик | | Просто вредносно ограничење | | Сложено вредносно ограничење | | Структурно ограничење |
| Атрибути | Име | Тип aтрибута | Вредност атрибута | Међузависнот атрибута једне табеле | Међузависнот атрибута више табеле | INSERT  RESTRICTED  Одсек  UPDATE RESTRICTED  Одсек  CASCADES  Евиденција  DELETE RESTRICTED  Евиденција |
|  | ЈМБГУченика | String | not null |  |  |
| ИмеУченика | String | not null |  |  |
| ПрезимеУченика | String | not null |  |  |
|  | ТелефонУченика | String | not null |  |  |
|  | ОдсекИД | Int | not null and >0 |  |  |
|  | Година | Int | not null and >0 |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Taбела Професор | | Просто вредносно ограничење | | Сложено вредносно ограничење | | Структурно ограничење |
| Атрибути | Име | Тип aтрибута | Вредност атрибута | Међузависнот атрибута једне табеле | Међузависнот атрибута више табеле | CASCADES  Евиденција  DELETE RESTRICTED  Евиденција |
|  | ЈМБГПрофесора | String | not null |  |  |
| ИмеПрофесора | String | not null |  |  |
| ПрезимеПрофесора | String | not null |  |  |
|  | ТелефонПрофесора | String | not null |  |  |
|  | Предмет | String | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Taбела Одсек | | Просто вредносно ограничење | | Сложено вредносно ограничење | | Структурно ограничење |
| Атрибути | Име | Тип aтрибута | Вредност атрибута | Међузависнот атрибута једне табеле | Међузависнот атрибута више табеле | INSERT/ UPDATE  CASCADES  Ученик  DELETE  RESTRICTED  Ученик |
|  | ОдсекИД | Int | not null and >0 |  |  |
| НазивОдсека | String | not null |  |  |
|  | Трајање | Int | Not null and >0 |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Taбела Евиденција | | Просто вредносно ограничење | | Сложено вредносно ограничење | | Структурно ограничење |
| Атрибути | Име | Тип aтрибута | Вреднот атрибута | Међузависнот атрибута једне табеле | Међузависнот атрибута више табеле | INSERT  RESTRICTED  Ученик, Професор  UPDATE  RESTRICTED  Ученик, Професор |
|  | ЕвиденцијаИД | Integer | not null |  |  |
| ЈМБГУченика | String | not null |  |  |
| ЈМБГПрофесора | String | not null |  |  |
|  | ДатумЕвиденције | Date | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Taбела Предмет | | Просто вредносно ограничење | | Сложено вредносно ограничење | | Структурно ограничење |
| Атрибути | Име | Тип aтрибута | Вредност атрибута | Међузависнот атрибута једне табеле | Међузависнот атрибута више табеле | INSERT/ UPDATE  CASCADES  Професор  DELETE  RESTRICTED  Професор |
|  | ПредметИД | Int | not null and >0 |  |  |
| НазивПредмета | String | not null |  |  |

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





# Пројектовање

Фаза пројектовања описује физичку структуру и понашање софтверског система архитектуру софтверског система). Пројектовање архитектуре софтверског система обухвата пројектовање корисничког интерфејса, апликационе логике и складишта података. Пројектовање корисничког интерфејса обухвата пројектовање екранских форми и контролера корисничког интерфејса. У оквиру апликационе логике се пројектују контролер апликационе логике, пословна логика И брокер базе података. Пројектовање пословне логика обухвата пројектовање логичке струцтуре и понашања софтверског система.

## 3.1.Архиктетура софтверског система

У оквиру фазе пројектовања описује се физичка структура и понашање софтверског система тј. архитектура софтверског система. У овом семинарском раду је коришћена класична трослојна архитектура, која се састоји од:

* Корисничког интерфејса
* Апликационе логике
* Складишта података

Korisnički interfejs

Aplikaciona logika

Skladište podataka

**I sloj**

**II sloj**

**III sloj**

**Softverski sistem**

Слика: Трослојна архитектура

## 

Skladište podataka

**Korisnički interfejs**

Ekranska forma

KontrolerKI

**Aplikaciona logika**

KontrolerAL

Poslovna logika

Database broker

Слика: Трослојна архитектура – детаљнији приказ

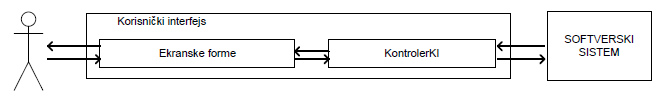
## 

## 3.2.Пројектовање софверског система

Контролер корисничког интерфејса и екранске форме не представљају софтверски систем, већ реализацију улаза и излаза из софтверског система. Контролер корисничког интерфејса има улогу да конвертује податке у графичке елементе екранских форми и да податке или пошаље у софтверски систем или да прими податке из софтверског система и исте конвертује у елементе екранске форме.

Кориснички интерфејс се састоји из:

* Екранске форме,
* Контролера корисничког интерфејса.



## 3.3.Пројектовање екранских форми

### 3.3.1. СК1: Унос новог ученика

**Назив СК**

Креирање новог ученика

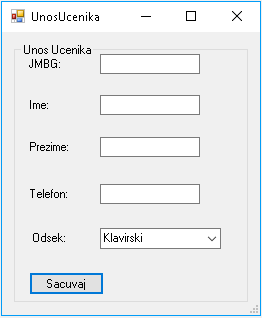
**Актори СК**

Запослено лице

**Учесници СК**

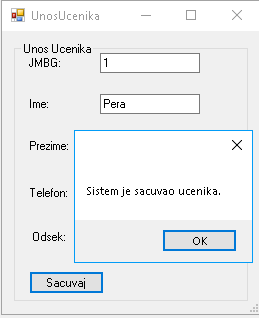
Запослено лице и систем (програм)

**Предуслов**: Систем је укључен и запослено лице je улоговано под својом шифром. Систем приказује форму за унос ученика. Учитана је листа одсека.



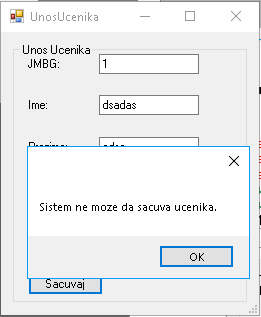
**Основни сценарио СК**

1. Запослено лице **уноси** податке у ученика. (АПУСО)
2. Запослено лице **контролише** да ли је коректно унео податке у ученика. (АНСО)
3. Запослено лице **позива** систем да сачува податке о ученику. (АПСО)
4. Систем **памти** податке о ученику. (СО)
5. Систем **приказује** запосленом лицу сачуваног ученика и поруку: “Систем је запамтио ученика“. (ИА)



Алтернативна сценарија

5.1 Уколико систем не може да запамти податке о ученику он приказује запосленом лицу поруку “Систем не може да сачува ученика”. (ИА)



### 3.3.2. СК2: Претраживање ученика

**Назив СК**

Претраживање ученика

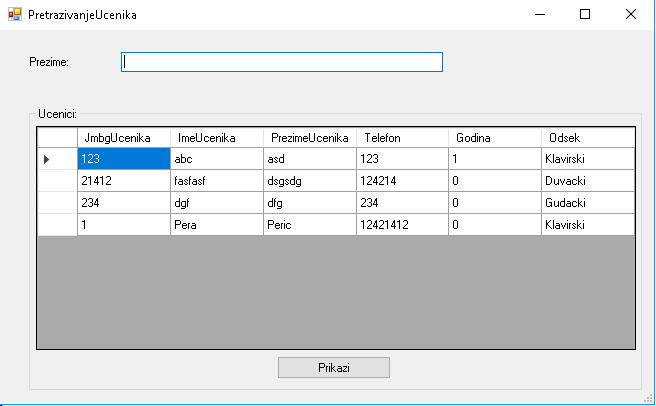
**Актори СК**

Запослено лице

**Учесници СК**

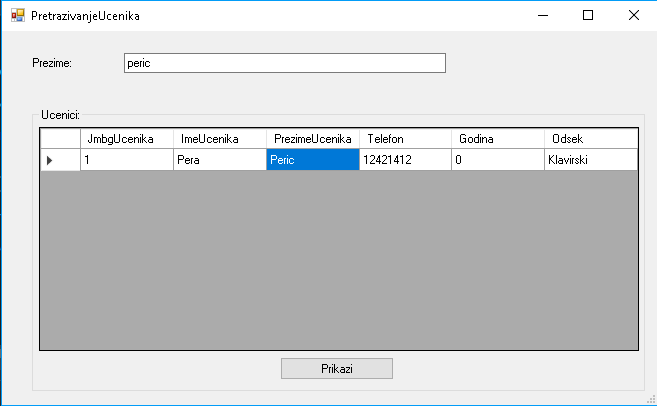
Запослено лице и систем (програм)

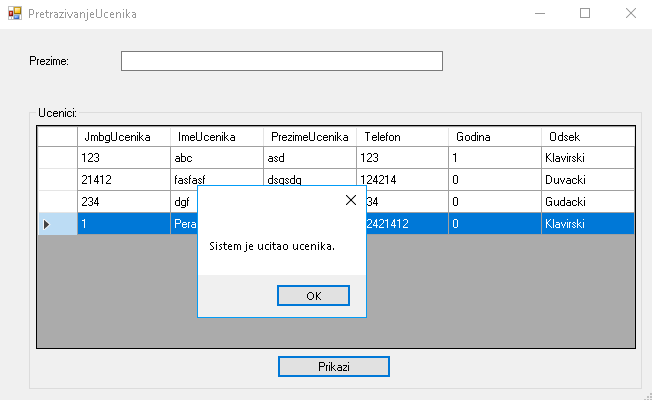
**Предуслов**: Систем је укључен и запослено лице je улоговано под својом шифром. Систем приказује форму за претраживање ученика.



**Основни сценарио СК**

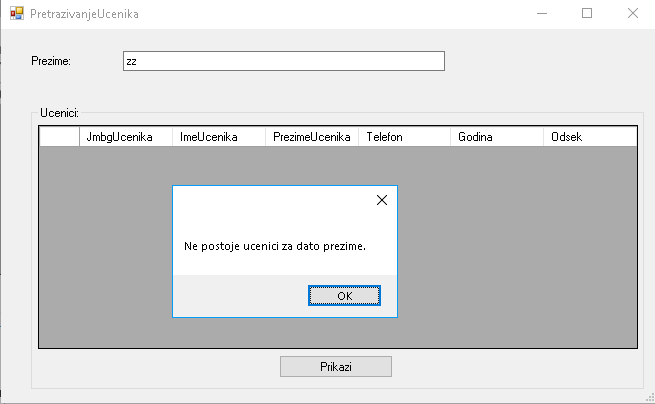
1. Запослено лице **уноси** вредностпо којој претражује ученике. (АПУСО)
2. Запослено лице **позива** систем да нађе ученике по задатој вредности. (АПСО)
3. Систем **тражи** ученике по задатој вредности. (СО)
4. Систем **приказује** запосленом лицу ученике (ИА)
5. Запослено лице **бира** ученика чије податке жели да види. (АПУСО)
6. Запослено лице **позива** систем да прикаже податке о одабраном ученику. (АПСО)
7. Систем **учитава** податке о одабраном ученику. (СО)
8. Систем **обавештава** запослено лице о успешном учитавању података о ученику и поруку: “Систем је успешно учитао ученика”. (ИА)





Алтернативна сценарија

4.1 Уколико систем не може да нађе ученика он приказује запосленом лицу поруку: “Систем не може да нађе ученикe по задатој вредности”. (ИА)



8.1 Уколико систем не може да нађе ученика он приказује запосленом лицу поруку “Систем не може да учита ученика”. (ИА)

### 3.3.3. СК3: Измена ученика

**Назив СК**

Промена ученика

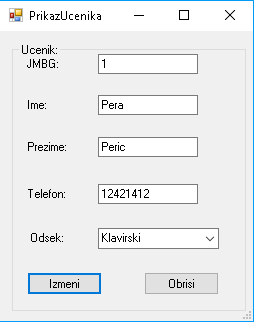
**Актори СК**

Запослено лице

**Учесници СК**

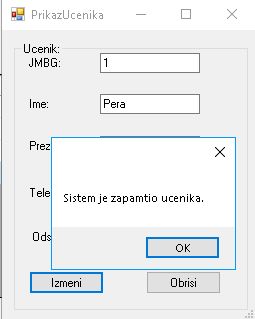
Запослено лице и систем (програм)

**Предуслов**: Систем је укључен и Запослено лице je улоговано под својом шифром. Систем приказује форму за рад са учеником након што је претходно извршено претраживање .



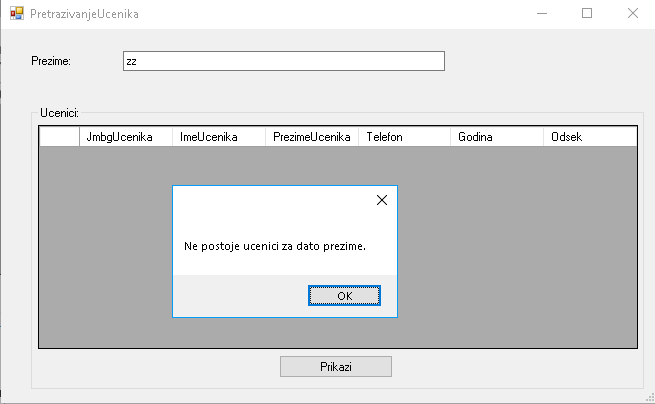
**Основни сценарио СК**

1. Запослено лице **уноси** вредностпо којој претражује ученике. (АПУСО)
2. Запослено лице **позива** систем да нађе ученике по задатој вредности. (АПСО)
3. Систем **тражи** ученике по задатој вредности. (СО)
4. Систем **приказује** запосленом лицу ученике (ИА)
5. Запослено лице **бира** ученика чије податке жели да види. (АПУСО)
6. Запослено лице **позива** систем да прикаже податке о одабраном ученику. (АПСО)
7. Систем **учитава** податке о одабраном ученику. (СО)
8. Систем **обавештава** запослено лице о успешном учитавању података о ученику и поруку: “Систем је успешно учитао ученика”. (ИА)
9. Запослено лице **уноси** **(мења)** податке о ученику. (АПУСО)
10. Запослено лице **контролише** да ли је коректно унео податке о ученику. (АНСО)
11. Запослено лице **позива** систем да запамти податке о ученику. (АПСО)
12. Систем **памти** податке о ученику. (СО)
13. Систем **приказује** Запосленом лицу запамћени ученика и поруку: “Систем је запамтио ученика.” (ИА)



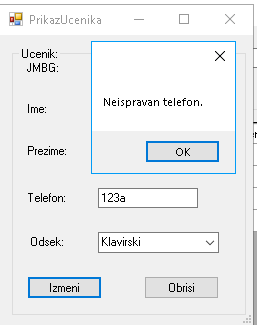
Алтернативна сценарија

4.1 Уколико систем не може да нађе ученика он приказује запосленом лицу поруку: “Систем не може да нађе ученикe по задатој вредности”. (ИА)



8.1 Уколико систем не може да нађе ученика он приказује запосленом лицу поруку “Систем не може да учита ученика”. (ИА)

13.1 Уколико систем не може да запамтиподатке о ученику он приказује запосленом лицу поруку: “Систем не може да запамти ученика”. (ИА)



### 3.3.4. СК4: Брисање ученика

**Назив СК**

Брисање ученика

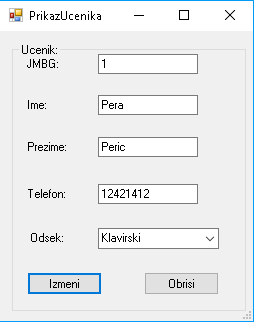
**Актори СК**

Запослено лице

**Учесници СК**

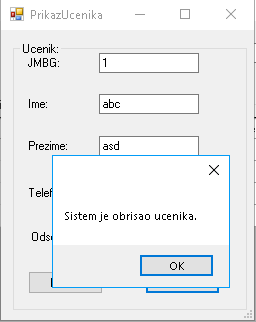
Запослено лице и систем (програм)

**Предуслов**: Систем је укључен и запослено лице je улогован под својом шифром. Систем приказује форму за рад са учеником након што је претходно извршено претраживање .



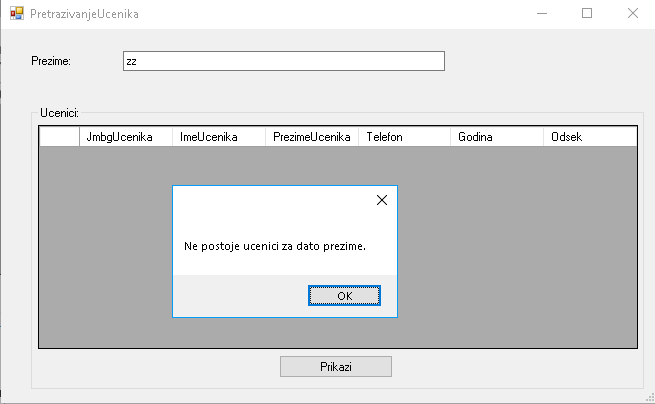
**Основни сценарио СК**

1. Запослено лице **уноси** вредностпо којој претражује ученике. (АПУСО)
2. Запослено лице **позива** систем да нађе ученике по задатој вредности. (АПСО)
3. Систем **тражи** ученике по задатој вредности. (СО)
4. Систем **приказује** запосленом лицу ученике (ИА)
5. Запослено лице **бира** ученика чије податке жели да види. (АПУСО)
6. Запослено лице **позива** систем да прикаже податке о одабраном ученику. (АПСО)
7. Систем **учитава** податке о одабраном ученику. (СО)
8. Систем **обавештава** запослено лице о успешном учитавању података о ученику и поруку: “Систем је успешно учитао ученика”. (ИА)
9. Запослено лице **позива** систем да обрише ученика. (АПСО)
10. Систем **брише** ученика. (СО)
11. Систем **приказује** запосленом лицу поруку: “Систем је обрисао ученика.” (ИА)



Алтернативна сценарија

4.1 Уколико систем не може да нађе ученика он приказује запосленом лицу поруку: “Систем не може да нађе ученикe по задатој вредности”. (ИА)



8.1 Уколико систем не може да нађе ученика он приказује запосленом лицу поруку “Систем не може да учита ученика”. (ИА)

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

### 3.3.5. СК5: Унос новог професора

**Назив СК**

Креирање новог професора

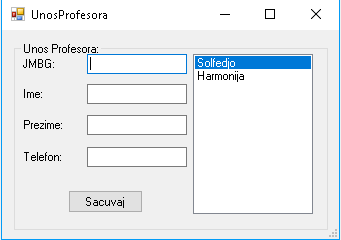
**Актори СК**

Запослено лице

**Учесници СК**

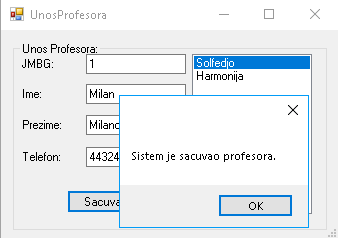
Запослено лице и систем (програм)

**Предуслов**: Систем је укључен и запослено лице je улоговано под својом шифром. Систем приказује форму за унос професора. Учитана је листа предмета.



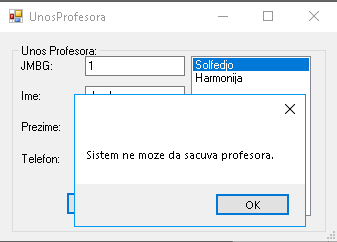
**Основни сценарио СК**

1. Запослено лице **уноси** податке у професора. (АПУСО)
2. Запослено лице **контролише** да ли је коректно унео податке у професора. (АНСО)
3. Запослено лице **позива** систем да сачува податке о професору. (АПСО)
4. Систем **памти** податке о професору. (СО)
5. Систем **приказује** запосленом лицу запамћеног професора и поруку: “Систем је сачувао професора“. (ИА)



Алтернативна сценарија

* 1. Уколико систем не може да запамти податке о професора он приказује заосленом лицу поруку “Систем не може да сачува професора ”. (ИА)



### 3.3.6. СК6: Претраживање професора

**Назив СК**

Претраживање професора

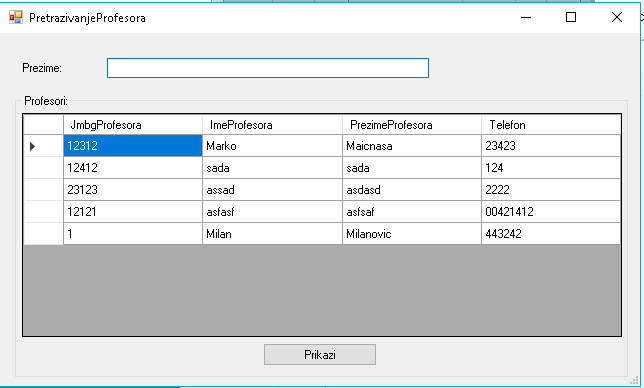
**Актори СК**

Запослено лице

**Учесници СК**

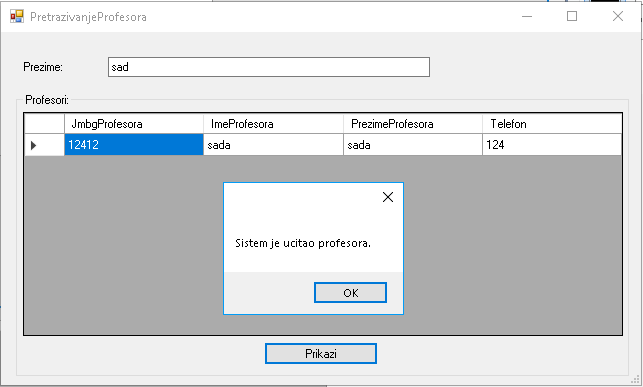
Запослено лице и систем (програм)

**Предуслов**: Систем је укључен и запослено лице je улоговано под својом шифром. Систем приказује форму за претраживање професора.



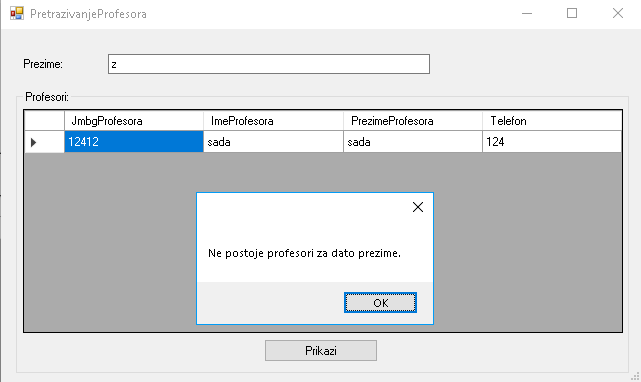
**Основни сценарио СК**

1. Запослено лице **уноси** вредностпо којој претражује професоре. (АПУСО)
2. Запослено лице **позива** систем да нађе професоре по задатој вредности. (АПСО)
3. Систем **тражи** професоре по задатој вредности. (СО)
4. Систем **приказује** запосленом лицу профеосре (ИА)
5. Запослено лице **бира** професора чије податке жели да види. (АПУСО)
6. Запослено лице **позива** систем да прикаже податке о одабраном професору. (АПСО)
7. Систем **учитава** податке о одабраном професору. (СО)
8. Систем **обавештава** запослено лице о успешном учитавању података о професору и поруку: “Систем је успешно учитао професора”. (ИА)



Алтернативна сценарија

4.1 Уколико систем не може да нађе професора он приказује запосленом лицу поруку: “Систем не може да нађе професоре по задатој вредности”. (ИА)



8.1 Уколико систем не може да нађе професора он приказује запосленом лицу поруку “Систем не може да учита професора”. (ИА)

### 3.3.7. СК7: Брисање професора

**Назив СК**

Брисање професора

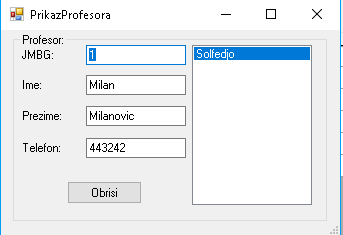
**Актори СК**

Запослено лице

**Учесници СК**

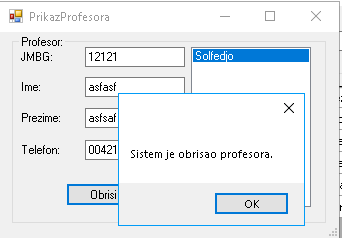
Запослено лице и систем (програм)

**Предуслов**: Систем је укључен и запослено лице je улогован под својом шифром. Систем приказује форму за рад са професором након што је претходно извршено претраживање .



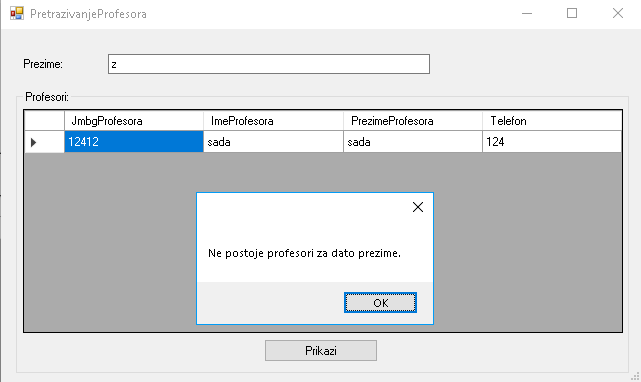
**Основни сценарио СК**

1. Запослено лице **уноси** вредностпо којој претражује професоре. (АПУСО)
2. Запослено лице **позива** систем да нађе професоре по задатој вредности. (АПСО)
3. Систем **тражи** професоре по задатој вредности. (СО)
4. Систем **приказује** запосленом лицу профеосре (ИА)
5. Запослено лице **бира** професора чије податке жели да види. (АПУСО)
6. Запослено лице **позива** систем да прикаже податке о одабраном професору. (АПСО)
7. Систем **учитава** податке о одабраном професору. (СО)
8. Систем **обавештава** запослено лице о успешном учитавању података о професору и поруку: “Систем је успешно учитао професора”. (ИА)
9. Запослено лице **позива** систем да обрише професора. (АПСО)
10. Систем **брише** професора. (СО)
11. Систем **приказује** запосленом лицу поруку: “Систем је обрисао професора.” (ИА)



Алтернативна сценарија

4.1 Уколико систем не може да нађе професора он приказује запосленом лицу поруку: “Систем не може да нађе професоре по задатој вредности”. (ИА)



8.1 Уколико систем не може да нађе професора он приказује запосленом лицу поруку “Систем не може да учита професора”. (ИА)

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

### 3.3.8. СК8: Унос новог предавања (Сложен СК)

**Назив СК**

Креирање новог предавања

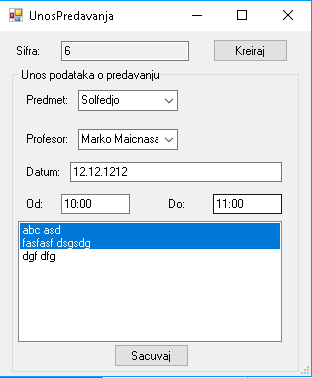
**Актори СК**

Запослено лице

**Учесници СК**

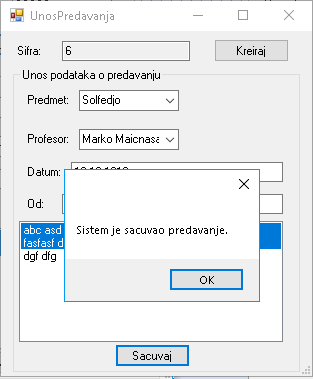
Запослено лице и систем (програм)

**Предуслов**: Систем је укључен и запослено лице je улоговано под својом шифром. Систем приказује форму за унос предавања. Учитане су листе предмета, професора и ученика.



**Основни сценарио СК**

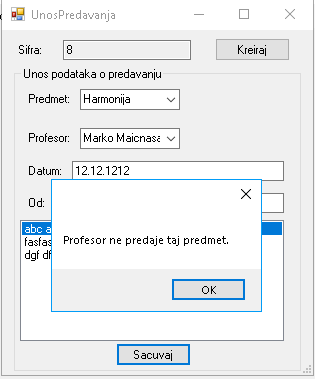
1. Запослено лице **позива** систем да креира предавање. (АПСО)
2. Систем **креира** предавање. (СО)
3. Систем **приказује** запосленом лицу предавање и поруку: “Систем је креирао предавање“. (ИА)
4. Запослено лице **уноси** податке о предавању. (АПУСО)
5. Запослено лице **контролише** да ли је коректно унео податке о предавање. (АНСО)
6. Запослено лице **позива** систем да запамти податке о предавању. (АПСО)
7. Систем **памти** податке о предавању. (СО)
8. Систем **приказује** запосленом лицу запамћену предавање и поруку: “Систем је запамтио предавање“. (ИА)



Алтернативна сценарија

3.1 Уколико систем не може да креира предавање он приказује запосленом лицу поруку: “Систем не може да креира предавање ”. Прекида се извршење сценарија. (ИА)

8.1 Уколико систем не може да запамти податке о предавању он приказује заосленом лицу поруку “Систем не може да запамти предавање ”. (ИА)



### 3.3.9. СК9: Претраживање предавања

**Назив СК**

Претраживање предавања

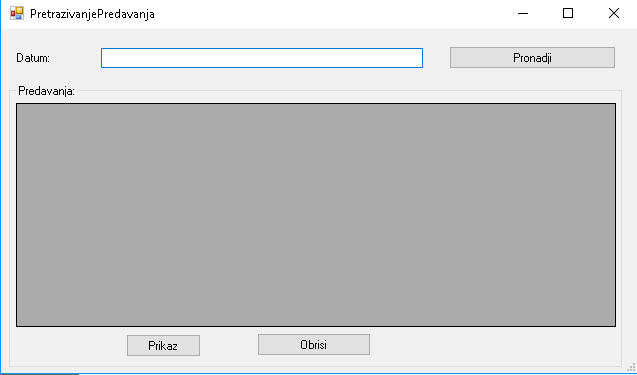
**Актори СК**

Запослено лице

**Учесници СК**

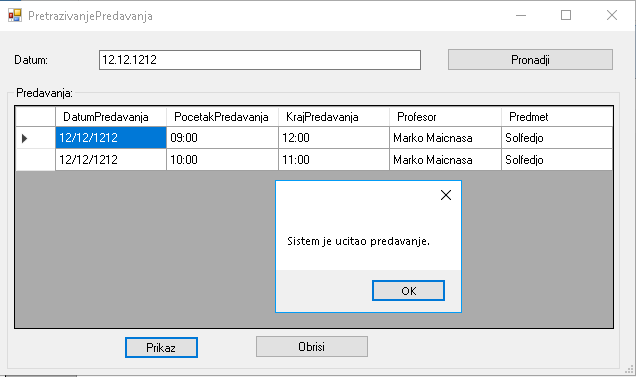
Запослено лице и систем (програм)

**Предуслов**: Систем је укључен и запослено лице je улоговано под својом шифром. Систем приказује форму за претраживање предавања.



**Основни сценарио СК**

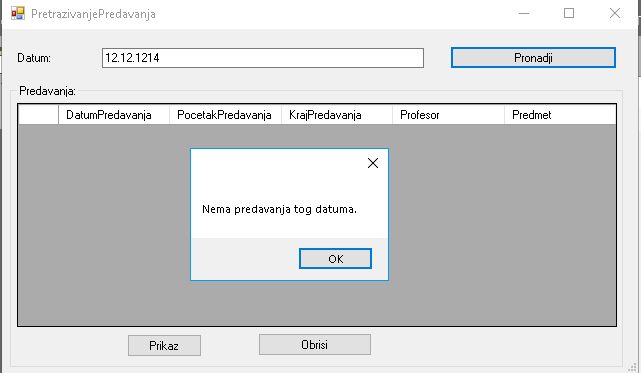
1. Запослено лице **уноси** вредностпо којој претражује предавања. (АПУСО)
2. Запослено лице **позива** систем да нађе предавања по задатој вредности. (АПСО)
3. Систем **тражи** предавања по задатој вредности. (СО)
4. Систем **приказује** запосленом лицу предавања (ИА)
5. Запослено лице **бира** предавања чије податке жели да види. (АПУСО)
6. Запослено лице **позива** систем да прикаже податке о одабраном предавању. (АПСО)
7. Систем **учитава** податке о одабраном предавању. (СО)
8. Систем **обавештава** запослено лице о успешном учитавању података о предавању и поруку: “Систем је успешно учитао предавање”. (ИА)



Алтернативна сценарија

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

8.1 Уколико систем не може да нађе предавање он приказује запосленом лицу поруку “Систем не може да учита предавање”. (ИА)



### 3.3.10. СК10: Брисање предавања

**Назив СК**

Брисање предавања

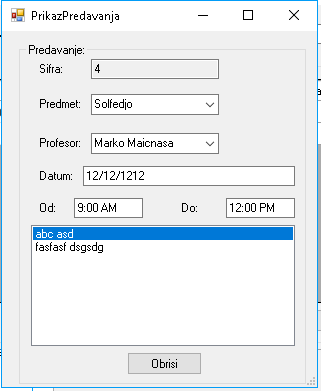
**Актори СК**

Запослено лице

**Учесници СК**

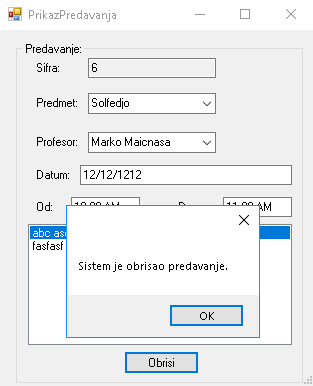
Запослено лице и систем (програм)

**Предуслов**: Систем је укључен и запослено лице je улогован под својом шифром. Систем приказује форму за рад са предавањем након што је претходно извршено претраживање .



**Основни сценарио СК**

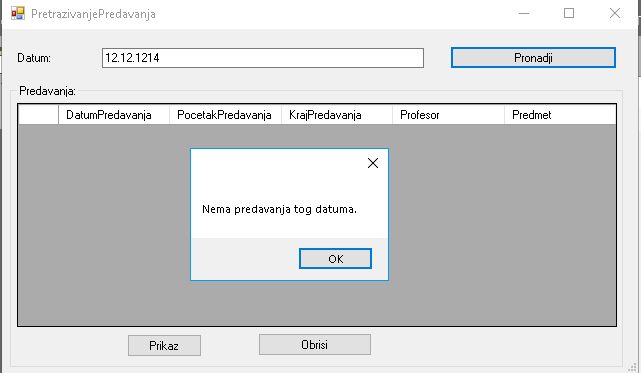
1. Запослено лице **уноси** вредностпо којој претражује предавања. (АПУСО)
2. Запослено лице **позива** систем да нађе предавања по задатој вредности. (АПСО)
3. Систем **тражи** предавања по задатој вредности. (СО)
4. Систем **приказује** запосленом лицу предавања (ИА)
5. Запослено лице **бира** предавања чије податке жели да види. (АПУСО)
6. Запослено лице **позива** систем да прикаже податке о одабраном предавању. (АПСО)
7. Систем **учитава** податке о одабраном предавању. (СО)
8. Систем **обавештава** запослено лице о успешном учитавању података о предавању и поруку: “Систем је успешно учитао предавање”. (ИА)
9. Запослено лице **позива** систем да обрише предавање. (АПСО)
10. Систем **брише** предавање. (СО)
11. Систем **приказује** запосленом лицу поруку: “Систем је обрисао предавање.” (ИА)



Алтернативна сценарија

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

8.1 Уколико систем не може да нађе предавање он приказује запосленом лицу поруку “Систем не може да учита предавање”. (ИА)



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

### 3.3.11. Пројектовање контролера корисничког интерфејса

Апликациона логика служи за описивање структуре и понашања софтверског система и пројектује се независно од корисничког интерфејса и обрнуто. Другим речима, апликациона логика (која представља Модел у MVC патерну) нема знања о томе где се налази кориснички интерфејс (која представља View у MVC патерну).

Контролер је одговоран да прихвати захтев за извршење системске операције од клијента и да га проследи до пословне логике која је одговорна за извршење системске операције.

### 

## Пројектовање апликације

### 3.4.1. Контролер апликационе логике

Контролер апликационе логике треба да подигне серверски сокет који ће да ослушкује мрежу. Када клијент (клијентски сокет) успостави конекцију са контролером (серверским сокетом), тада контролер треба да генерише нит која ће успоставити двосмерну везу са клијентом (улазну и излазну). Слање и примање података од клијента се остварује преко сокета.

Клијент шаље захтев за извршење неке од СО до одговарајуће нити (коју смо назвали “Обрада”), која је повезана са тим клијентом. “Обрада” 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 “Обраде”, која тај резултат шаље назад до клијента.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Komunikacija;

using Biblioteka;

using System.Windows.Forms;

using System.ComponentModel;

namespace KontrolerKorisnickogInterfejsa

{

public class KontrolerKI

{

public static Komunikacija.Komunikacija komunikacija;

public static Ucenik ucenik;

public static Profesor profesor;

public static Predavanje predavanje;

public static Radnik radnik;

public static bool poveziSeNaServer()

{

komunikacija = new Komunikacija.Komunikacija();

return komunikacija.poveziSeNaServer();

}

public void popuniPoljaPredavanje(TextBox txtSifra, ComboBox cmbPredmet, ComboBox cmbProfesor, TextBox txtDatum, TextBox txtOd, TextBox txtDo, ListBox listUcenici)

{

txtSifra.Text = predavanje.PredavanjeID.ToString();

cmbPredmet.DataSource = vratiSvePredmete();

cmbPredmet.SelectedItem = predavanje.Predmet as Predmet;

cmbProfesor.DataSource = vratiSveProfesore();

cmbProfesor.SelectedItem = predavanje.Profesor as Profesor;

txtDatum.Text = predavanje.DatumPredavanja.ToShortDateString();

txtOd.Text = predavanje.PocetakPredavanja.ToShortTimeString();

txtDo.Text = predavanje.KrajPredavanja.ToShortTimeString();

List<Ucenik> lista = new List<Ucenik>();

foreach (Prisustvuje p in predavanje.ListaUcenika)

{

Ucenik ucenik = new Ucenik();

ucenik = p.Ucenik;

foreach (Ucenik uc in vratiSveUcenike())

{

if (ucenik.JmbgUcenika == uc.JmbgUcenika)

{

ucenik = uc;

}

}

lista.Add(ucenik);

}

listUcenici.DataSource = lista;

}

public void popuniPoljaProfesor(TextBox txtJMBG, TextBox txtIme, TextBox txtPrezime, TextBox txtTelefon, ListBox listPredmeti)

{

txtJMBG.Text = profesor.JmbgProfesora;

txtIme.Text = profesor.ImeProfesora;

txtPrezime.Text = profesor.PrezimeProfesora;

txtTelefon.Text = profesor.Telefon;

// List<Predmet> lp = komunikacija.vratiSvePredmete();

List<Predmet> lista = new List<Predmet>();

foreach (Predaje p in profesor.ListaPredmeta)

{

Predmet predmet = new Predmet();

predmet = p.Predmet;

foreach(Predmet pred in vratiSvePredmete())

{

if (predmet.PredmetID == pred.PredmetID)

{

predmet.NazivPredmeta = pred.NazivPredmeta;

}

}

lista.Add(predmet);

}

listPredmeti.DataSource = lista;

}

public bool pronadjiPredavanje(DataGridView dataGridView1)

{

try

{

predavanje = dataGridView1.CurrentRow.DataBoundItem as Predavanje;

predavanje = komunikacija.pronadjiPredavanje(predavanje);

if (predavanje == null)

{

MessageBox.Show("Sistem ne moze da ucita predavanje.");

return false;

}

else

{

MessageBox.Show("Sistem je ucitao predavanje.");

return true;

}

}

catch (Exception)

{

MessageBox.Show("Nije odabrano predavanje.");

return false;

}

}

public bool pronadjiProfesora(DataGridView dataGridView1)

{

try

{

profesor = dataGridView1.CurrentRow.DataBoundItem as Profesor;

profesor = komunikacija.pronadjiProfesora(profesor);

if (profesor == null)

{

MessageBox.Show("Sistem ne moze da ucita profesora.");

return false;

}

else

{

MessageBox.Show("Sistem je ucitao profesora.");

return true;

}

}

catch (Exception)

{

MessageBox.Show("Nije odabran profesor.");

return false;

}

}

public string dobrodoslica()

{

return "Dobrodosao/la, " + radnik.ToString();

}

public bool pronadjiRadnika(TextBox txtIme, TextBox txtLozinka)

{

radnik = new Radnik();

radnik.KorisnickoIme = txtIme.Text;

radnik.Lozinka = txtLozinka.Text;

radnik = komunikacija.pronadjiRadnika(radnik);

if(radnik==null)

{

MessageBox.Show("Pogresno ime ili lozinka.");

return false;

}

else

{

MessageBox.Show("Uspesno prijavljivanje.");

return true;

}

}

public void pronadjiPredavanja(TextBox txtDatum, DataGridView dataGridView1)

{

predavanje = new Predavanje();

predavanje.DatumPredavanja = DateTime.ParseExact(txtDatum.Text, "dd.MM.yyyy", null);

List<Predavanje> lista = komunikacija.pronadjiPredavanja(predavanje);

if (lista == null || lista.Count == 0)

{

MessageBox.Show("Nema predavanja tog datuma.");

}

else

{

// MessageBox.Show("Sistem je pronasao predavanja.");

}

foreach(Predavanje p in lista)

{

foreach(Profesor pr in komunikacija.vratiSveProfesore())

{

if (p.Profesor.JmbgProfesora == pr.JmbgProfesora) p.Profesor = pr;

}

foreach (Predmet pr in komunikacija.vratiSvePredmete())

{

if (p.Predmet.PredmetID == pr.PredmetID) p.Predmet = pr;

}

}

dataGridView1.DataSource = lista;

dataGridView1.Columns["PocetakPredavanja"].DefaultCellStyle.Format = "HH:mm";

dataGridView1.Columns["KrajPredavanja"].DefaultCellStyle.Format = "HH:mm";

}

public bool obrisiPredavanje()

{

try

{

object o = komunikacija.obrisiPredavanje(predavanje);

if (o == null)

{

MessageBox.Show("Sistem ne moze da obrise predavanje.");

return false;

}

else

{

MessageBox.Show("Sistem je obrisao predavanje.");

return true;

}

}

catch (Exception)

{

MessageBox.Show("Nije odabrano predavanje.");

return false;

}

}

public void popuniPoljaUcenik(TextBox txtJMBG, TextBox txtIme, TextBox txtPrezime, TextBox txtTelefon, ComboBox cmbOdsek)

{

txtJMBG.Text = ucenik.JmbgUcenika;

txtIme.Text = ucenik.ImeUcenika;

txtPrezime.Text = ucenik.PrezimeUcenika;

txtTelefon.Text = ucenik.Telefon;

cmbOdsek.DataSource = vratiSveOdseke();

cmbOdsek.SelectedItem= ucenik.Odsek as Odsek; //!!!!!!!!!!!!!!!!!!!!!!!!!!!!

}

public bool zapamtiPredavanje(TextBox txtDatum, TextBox txtOd, TextBox txtDo, ComboBox cmbPredmet, ComboBox cmbProfesor, ListBox listUcenici)

{

try

{

predavanje.DatumPredavanja = DateTime.ParseExact(txtDatum.Text, "dd.MM.yyyy", null);

}

catch (Exception)

{

MessageBox.Show("Neispravan datum.");

txtDatum.Focus();

return false;

}

try

{

predavanje.PocetakPredavanja = DateTime.ParseExact(txtOd.Text, "HH:mm", null);

predavanje.KrajPredavanja = DateTime.ParseExact(txtDo.Text, "H:mm", null);

}

catch (Exception)

{

MessageBox.Show("Neispravno vreme.");

txtOd.Focus();

txtDo.Focus();

return false;

}

if (predavanje.PocetakPredavanja == null || predavanje.KrajPredavanja == null)

{

MessageBox.Show("Niste dobro uneli vreme predavanja.");

return false;

}

predavanje.Predmet = cmbPredmet.SelectedItem as Predmet;

predavanje.Profesor = cmbProfesor.SelectedItem as Profesor;

foreach (Predaje pred in predavanje.Profesor.ListaPredmeta)

{

if (pred.Predmet.PredmetID == predavanje.Predmet.PredmetID)

{

break;

}

else

{

MessageBox.Show("Profesor ne predaje taj predmet.");

return false;

}

}

foreach (Ucenik u in listUcenici.SelectedItems)

{

Prisustvuje p = new Prisustvuje();

p.Ucenik = u;

p.Predavanje = predavanje;

predavanje.ListaUcenika.Add(p);

}

Object o = komunikacija.zapamtiPredavanje(predavanje);

if (o == null)

{

MessageBox.Show("Sistem ne moze da sacuva predavanje.");

return false;

}

else

{

MessageBox.Show("Sistem je sacuvao predavanje.");

return true;

}

}

public void kreirajPredavanje(TextBox txtSifra, GroupBox groupBox1)

{

predavanje = komunikacija.kreirajPredavanje();

if (predavanje==null)

{

MessageBox.Show("Sistem ne moze da kreira predavanje.");

}

else

{

MessageBox.Show("Sistem je kreirao predavanje.");

txtSifra.Text = predavanje.PredavanjeID.ToString();

groupBox1.Enabled = true;

}

}

public void pronadjiProfesore(TextBox txtPrezime, DataGridView dataGridView1)

{

profesor = new Profesor();

profesor.PrezimeProfesora = txtPrezime.Text;

List<Profesor> lista = komunikacija.pronadjiProfesore(profesor);

if (lista == null || lista.Count == 0)

{

MessageBox.Show("Ne postoje profesori za dato prezime.");

}

else

{

// MessageBox.Show("Sistem je pronasao profesore.");

}

dataGridView1.DataSource = lista;

}

public bool sacuvajProfesora(TextBox txtJMBG, TextBox txtIme, TextBox txtPrezime, TextBox txtTelefon, ListBox listPredmeti)

{

profesor = new Profesor();

try

{

if(txtJMBG.Text.All(char.IsDigit))

profesor.JmbgProfesora = txtJMBG.Text;

else

{

MessageBox.Show("Neispravan JMBG.");

txtJMBG.Focus();

return false;

}

}

catch (Exception)

{

MessageBox.Show("Neispravan JMBG.");

txtJMBG.Focus();

return false;

}

profesor.ImeProfesora = txtIme.Text;

profesor.PrezimeProfesora = txtPrezime.Text;

if (profesor.ImeProfesora == "" || profesor.PrezimeProfesora == "")

{

MessageBox.Show("Niste uneli ime i prezime.");

return false;

}

if(txtTelefon.Text.All(char.IsDigit))

profesor.Telefon = txtTelefon.Text;

else

{

MessageBox.Show("Neispravan telefon.");

txtTelefon.Focus();

return false;

}

foreach(Predmet p in listPredmeti.SelectedItems)

{

Predaje pre = new Predaje();

pre.Predmet = p;

pre.Profesor = profesor;

profesor.ListaPredmeta.Add(pre);

}

Object o = komunikacija.sacuvajProfesora(profesor);

if (o == null)

{

MessageBox.Show("Sistem ne moze da sacuva profesora.");

return false;

}

else

{

MessageBox.Show("Sistem je sacuvao profesora.");

return true;

}

}

public bool obrisiUcenika()

{

Object o = komunikacija.obrisiUcenika(ucenik);

if(o==null)

{

MessageBox.Show("Sistem ne moze da obrise ucenika.");

return false;

}

else

{

MessageBox.Show("Sistem je obrisao ucenika.");

return true;

}

}

public bool pronadjiUcenika(DataGridView dataGridView1)

{

try

{

ucenik = dataGridView1.CurrentRow.DataBoundItem as Ucenik;

ucenik = komunikacija.pronadjiUcenika(ucenik);

if(ucenik==null)

{

MessageBox.Show("Sistem ne moze da ucita ucenika.");

return false;

}

else

{

MessageBox.Show("Sistem je ucitao ucenika.");

return true;

}

}

catch (Exception)

{

MessageBox.Show("Nije odabran ucenik.");

return false;

}

}

public void pronadjiUcenike(TextBox txtPrezime, DataGridView dataGridView1)

{

ucenik = new Ucenik();

ucenik.PrezimeUcenika = txtPrezime.Text;

List<Ucenik> lista = komunikacija.pronadjiUcenike(ucenik);

if(lista==null || lista.Count==0)

{

MessageBox.Show("Ne postoje ucenici za dato prezime.");

}

else

{

// MessageBox.Show("Sistem je pronasao clanove.");

}

dataGridView1.DataSource = lista;

}

public void kraj()

{

komunikacija.kraj();

}

public List<Ucenik> vratiSveUcenike()

{

return komunikacija.vratiSveUcenike();

}

public List<Predmet> vratiSvePredmete()

{

return komunikacija.vratiSvePredmete();

}

// !!!

public bool sacuvajUcenika(TextBox txtJMBG, TextBox txtIme, TextBox txtPrezime, TextBox txtTelefon, ComboBox cmbOdsek)

{

ucenik = new Ucenik();

try

{

if(txtJMBG.Text.All(char.IsDigit))

ucenik.JmbgUcenika = txtJMBG.Text;

else

{

MessageBox.Show("Neispravan JMBG.");

txtJMBG.Focus();

return false;

}

}

catch (Exception)

{

MessageBox.Show("Neispravan JMBG.");

txtJMBG.Focus();

return false;

}

ucenik.ImeUcenika = txtIme.Text;

ucenik.PrezimeUcenika = txtPrezime.Text;

if(ucenik.ImeUcenika=="" || ucenik.PrezimeUcenika=="")

{

MessageBox.Show("Niste uneli ime i prezime.");

return false;

}

if(txtTelefon.Text.All(char.IsDigit))

ucenik.Telefon = txtTelefon.Text;

else

{

MessageBox.Show("Neispravan telefon.");

txtTelefon.Focus();

return false;

}

ucenik.Odsek = cmbOdsek.SelectedItem as Odsek;

Object o = komunikacija.sacuvajUcenika(ucenik);

if(o==null)

{

MessageBox.Show("Sistem ne moze da sacuva ucenika.");

return false;

}

else

{

MessageBox.Show("Sistem je sacuvao ucenika.");

return true;

}

}

public bool zapamtiUcenika(TextBox txtJMBG, TextBox txtIme, TextBox txtPrezime, TextBox txtTelefon, ComboBox cmbOdsek)

{

try

{

if (txtJMBG.Text.All(char.IsDigit))

ucenik.JmbgUcenika = txtJMBG.Text;

else

{

MessageBox.Show("Neispravan JMBG.");

txtJMBG.Focus();

return false;

}

}

catch (Exception)

{

MessageBox.Show("Neispravan JMBG.");

txtJMBG.Focus();

return false;

}

ucenik.ImeUcenika = txtIme.Text;

ucenik.PrezimeUcenika = txtPrezime.Text;

if (ucenik.ImeUcenika == "" || ucenik.PrezimeUcenika == "")

{

MessageBox.Show("Niste uneli ime i prezime.");

return false;

}

if (txtTelefon.Text.All(char.IsDigit))

ucenik.Telefon = txtTelefon.Text;

else

{

MessageBox.Show("Neispravan telefon.");

txtTelefon.Focus();

return false;

}

ucenik.Odsek = cmbOdsek.SelectedItem as Odsek;

Object o = komunikacija.zapamtiUcenika(ucenik);

if (o == null)

{

MessageBox.Show("Sistem ne moze da zapamti ucenika.");

return false;

}

else

{

MessageBox.Show("Sistem je zapamtio ucenika.");

return true;

}

}

public List<Odsek> vratiSveOdseke()

{

return komunikacija.vratiSveOdseke();

}

public bool obrisiProfesora()

{

try

{

object o = komunikacija.obrisiProfesora(profesor);

if (o == null)

{

MessageBox.Show("Sistem ne moze da obrise profesora.");

return false;

}

else

{

MessageBox.Show("Sistem je obrisao profesora.");

return true;

}

}

catch (Exception)

{

MessageBox.Show("Nije odabran profesor.");

return false;

}

}

public List<Profesor> vratiSveProfesore()

{

return komunikacija.vratiSveProfesore();

}

public List<Predavanje> vratiSvaPredavanja(DataGridView dataGridView1)

{

List<Predavanje> lista= komunikacija.vratiSvaPredavanja();

dataGridView1.DataSource = lista;

return lista;

}

}

}

### 3.4.2. Системске операције

У претходној фази, фази анализе, одредили смо уговоре о системским операцијама, при чему смо рекли да један уговор описује понашање једне системске операције, тако што описује ШТА операција треба да ради, али не и како. Сада се у фази пројектовања за сваки од уговора пројектује концептуално решење (реализација) СО. То значи да ћемо за сваку класу одговорну за извршење СО дефинисати КАКО ће се системска операција извршити.

На самом почетку пројектовања СО избећи ћемо аспекте реализације који су везани за конекцију са базом, перзистентност и трансакције.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Biblioteka;

using Sesija;

namespace SistemskeOperacije

{

public abstract class OpstaSO

{

public Object izvrsiSO(OpstiDomenskiObjekat odo)

{

Object rezultat = null;

Broker.dajSesiju().otvoriKonekciju();

Broker.dajSesiju().zapocniTransakciju();

try

{

rezultat = Izvrsi(odo);

Broker.dajSesiju().potvrdiTransakciju();

}

catch (Exception)

{

Broker.dajSesiju().ponistiTransakciju();

}

finally

{

Broker.dajSesiju().zatvoriKonekciju();

}

return rezultat;

}

public abstract object Izvrsi(OpstiDomenskiObjekat odo);

}

}

### 3.4.2.1. Уговор УГ1: СачувајУченика

Операција: СачувајУченика(Ученик):сигнал;

Веза са СК: СК1

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

Постуслови: Сачуван је нови ученик.

public class SacuvajUcenika : OpstaSO

{

public override object Izvrsi(OpstiDomenskiObjekat odo)

{

return Sesija.Broker.dajSesiju().sacuvaj(odo);

}

}

### 3.4.2.2. Уговор УГ2: ЗапамтиУченика

Операција: ЗапамтиУченика(Ученик):сигнал;

Веза са СК: СК1, СК3

Предуслови: Просто вредносно ограничење над објектом Ученик мора бити задовољено.

Структурно ограничење над објектом Ученик мора бити задовољено.

Постуслови: Подаци о ученику су запамћени.

public class ZapamtiUcenika : OpstaSO

{

public override object Izvrsi(OpstiDomenskiObjekat odo)

{

return Sesija.Broker.dajSesiju().izmeni(odo);

}

}

### 3.4.2.3. Уговор УГ3: ПронађиУченике

Операција: ПронађиУченике(Ученик, Лист<Ученик>):сигнал;

Веза са СК: СК2, СК3, СК4

Предуслови: -

Постуслови: -

public class PronadjiUcenike : OpstaSO

{

public override object Izvrsi(OpstiDomenskiObjekat odo)

{

List<Ucenik> lista = Sesija.Broker.dajSesiju().vratiSveZaUslovVise(odo).OfType<Ucenik>().ToList<Ucenik>();

foreach (Ucenik u in lista)

{

Odsek o = new Odsek();

o.OdsekID = u.Odsek.OdsekID;

u.Odsek = Sesija.Broker.dajSesiju().vratiZaUslovJedan(o) as Odsek;

}

return lista;

}

}

### 3.4.2.4. Уговор УГ4: ПронађиУченика

Операција: ПронађиУченика(Ученик):сигнал;

Веза са СК: СК2, СК3, СК4

Предуслови: -

Постуслови: -

public class PronadjiUcenika : OpstaSO

{

public override object Izvrsi(OpstiDomenskiObjekat odo)

{

return Sesija.Broker.dajSesiju().vratiZaUslovJedan(odo) as Ucenik;

}

}

### 3.4.2.5. Уговор УГ5: ОбришиУченика

Операција: ОбришиУченика(Ученик):сигнал;

Веза са СК: СК4

Предуслови: Структурно ограничење над објектом Ученик мора бити задовољено.

Постуслови: Ученик је обрисан.

public class ObrisiUcenika : OpstaSO

{

public override object Izvrsi(OpstiDomenskiObjekat odo)

{

return Sesija.Broker.dajSesiju().obrisi(odo);

}

}

### 3.4.2.6. Уговор УГ6: СачувајПрофесора

Операција: СачувајПрофесора(Професор):сигнал;

Веза са СК: СК5

Предуслови: Вредносна и структурна ограничења над објектом Професорморају бити задовољена.

Постуслови: Сачуван је нови професор.

public class SacuvajProfesora : OpstaSO

{

public override object Izvrsi(OpstiDomenskiObjekat odo)

{

Sesija.Broker.dajSesiju().sacuvaj(odo);

Profesor p = odo as Profesor;

Predaje pr = new Predaje();

pr.Profesor = p;

Sesija.Broker.dajSesiju().obrisi(pr);

foreach(Predaje pre in p.ListaPredmeta)

{

Sesija.Broker.dajSesiju().sacuvaj(pre);

}

return 1;

}

}

### 3.4.2.7. Уговор УГ7: ПронађиПрофесоре

Операција: ПронађиПрофесоре(Професор, Лист<Професор>):сигнал;

Веза са СК: СК6, СК7

Предуслови: -

Постуслови: -

public class PronadjiProfesore : OpstaSO

{

public override object Izvrsi(OpstiDomenskiObjekat odo)

{

List<Profesor> lista= Sesija.Broker.dajSesiju().vratiSveZaUslovVise(odo).OfType<Profesor>().ToList<Profesor>();

foreach (Profesor p in lista)

{

Predaje pr = new Predaje();

pr.Profesor = p;

p.ListaPredmeta = Sesija.Broker.dajSesiju().vratiSveZaUslovJedan(pr).OfType<Predaje>().ToList<Predaje>();

}

return lista;

}

### 3.4.2.8. Уговор УГ8: ПронађиПрофесора

Операција: ПронађиПрофесора(Професор):сигнал;

Веза са СК: СК6, СК7

Предуслови: -

Постуслови: -

public class PronadjiProfesora : OpstaSO

{

public override object Izvrsi(OpstiDomenskiObjekat odo)

{

Profesor p = Sesija.Broker.dajSesiju().vratiZaUslovJedan(odo) as Profesor;

Predaje pr = new Predaje();

pr.Profesor = p;

p.ListaPredmeta= Sesija.Broker.dajSesiju().vratiSveZaUslovJedan(pr).OfType<Predaje>().ToList<Predaje>();

return p;

}

}

### 3.4.2.9. Уговор УГ9: ОбришиПрофесора

Операција: ОбришиПрофесора(Професор):сигнал;

Веза са СК: СК7

Предуслови: Структурно ограничење над објектом Професор мора бити задовољено.

Постуслови: Професор је обрисан.

public class ObrisiProfesora : OpstaSO

{

public override object Izvrsi(OpstiDomenskiObjekat odo)

{

return Sesija.Broker.dajSesiju().obrisi(odo);

}

}

### 3.4.2.10. Уговор УГ10: ВратиСвеУченике

Операција: ВратиСвеУченике(Лист<Ученик>):сигнал;

Веза са СК: СК8

Предуслови: -

Постуслови: -

public class VratiSveUcenike : OpstaSO

{

public override object Izvrsi(OpstiDomenskiObjekat odo)

{

return Sesija.Broker.dajSesiju().vratiSve(odo).OfType<Ucenik>().ToList<Ucenik>();

}

}

### 3.4.2.11. Уговор УГ11: ВратиСвеПрофесоре

Операција: ВратиСвеПрофесоре(Лист<Професор>):сигнал;

Веза са СК: СК8

Предуслови: -

Постуслови: -

public class VratiSveProfesore : OpstaSO

{

public override object Izvrsi(OpstiDomenskiObjekat odo)

{

List<Profesor> lista=Sesija.Broker.dajSesiju().vratiSve(odo).OfType<Profesor>().ToList<Profesor>();

foreach(Profesor p in lista)

{

Predaje pr = new Predaje();

pr.Profesor = p;

p.ListaPredmeta = Sesija.Broker.dajSesiju().vratiSveZaUslovJedan(pr).OfType<Predaje>().ToList<Predaje>();

}

return lista;

}

}

### 3.4.2.12. Уговор УГ12: ВратиСвеПредмете

Операција: ВратиСвеПредмете(Лист<Предмет>):сигнал;

Веза са СК: СК5, СК8

Предуслови: -

Постуслови: -

public class VratiSvePredmete : OpstaSO

{

public override object Izvrsi(OpstiDomenskiObjekat odo)

{

return Sesija.Broker.dajSesiju().vratiSve(odo).OfType<Predmet>().ToList<Predmet>();

}

}

### 3.4.2.13. Уговор УГ13: ВратиСвеОдсеке

Операција: ВратиСвеПредмете(Лист<Одсек>):сигнал;

Веза са СК: СК1, СК3

Предуслови: -

Постуслови: -

public class VratiSveOdseke : OpstaSO

{

public override object Izvrsi(OpstiDomenskiObjekat odo)

{

return Sesija.Broker.dajSesiju().vratiSve(odo).OfType<Odsek>().ToList<Odsek>();

}

}

### 3.4.2.14. Уговор УГ14: КреирајПредавање

Операција: КреирајПредавање(Предавање):сигнал;

Веза са СК: СК8

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

Постуслови: Креирано је ново предавање.

public class KreirajPredavanje : OpstaSO

{

public override object Izvrsi(OpstiDomenskiObjekat odo)

{

Predavanje p = odo as Predavanje;

p.PredavanjeID = Sesija.Broker.dajSesiju().vratiSifru(p);

Sesija.Broker.dajSesiju().sacuvaj(p);

return p;

}

}

### 3.4.2.15. Уговор УГ15: ЗапамтиПредавање

Операција: Запамти Предавање(Предавање):сигнал;

Веза са СК: СК8

Предуслови: Просто вредносно ограничење над објектом Предавање мора бити задовољено.

Структурно ограничење над објектом Предавање мора бити задовољено.

Постуслови: Подаци о предавању су запамћени.

public class ZapamtiPredavanje : OpstaSO

{

public override object Izvrsi(OpstiDomenskiObjekat odo)

{

Sesija.Broker.dajSesiju().izmeni(odo);

Predavanje p = odo as Predavanje;

Prisustvuje pr = new Prisustvuje();

pr.Predavanje = p;

Sesija.Broker.dajSesiju().obrisiZaUslovVise(pr);

foreach(Prisustvuje pri in p.ListaUcenika)

{

Sesija.Broker.dajSesiju().sacuvaj(pri);

}

return 1;

}

}

### 3.4.2.16. Уговор УГ16: ПронађиПредавања

Операција: ПронађиПредавања(Предавање, Лист<Предавање>):сигнал;

Веза са СК: СК6, СК7

Предуслови: -

Постуслови: -

public class PronadjiPredavanja : OpstaSO

{

public override object Izvrsi(OpstiDomenskiObjekat odo)

{

//return Sesija.Broker.dajSesiju().vratiSveZaUslovVise(odo).OfType<Predavanje>().ToList<Predavanje>();

List<Predavanje> lista = Sesija.Broker.dajSesiju().vratiSveZaUslovVise(odo).OfType<Predavanje>().ToList<Predavanje>();

foreach (Predavanje p in lista)

{

Prisustvuje pr = new Prisustvuje();

pr.Predavanje = p;

p.ListaUcenika = Sesija.Broker.dajSesiju().vratiSveZaUslovVise(pr).OfType<Prisustvuje>().ToList<Prisustvuje>();

}

return lista;

}

}

### 3.4.2.17. Уговор УГ17: ПронађиПредавање

Операција: ПронађиПредавање(Предавање):сигнал;

Веза са СК: СК6, СК7

Предуслови: -

Постуслови: -

public class PronadjiPredavanje : OpstaSO

{

public override object Izvrsi(OpstiDomenskiObjekat odo)

{

Predavanje p = Sesija.Broker.dajSesiju().vratiZaUslovJedan(odo) as Predavanje;

Prisustvuje pr = new Prisustvuje();

pr.Predavanje = p;

p.ListaUcenika = Sesija.Broker.dajSesiju().vratiSveZaUslovVise(pr).OfType<Prisustvuje>().ToList<Prisustvuje>();

return p;

}

}

### 3.4.2.18. Уговор УГ18: ОбришиПредавање

Операција: ОбришиПредавање(Предавање):сигнал;

Веза са СК: СК7

Предуслови: Структурно ограничење над објектом Предавање мора бити задовољено.

Постуслови: Предавање је обрисано.

public class ObrisiPredavanje : OpstaSO

{

public override object Izvrsi(OpstiDomenskiObjekat odo)

{

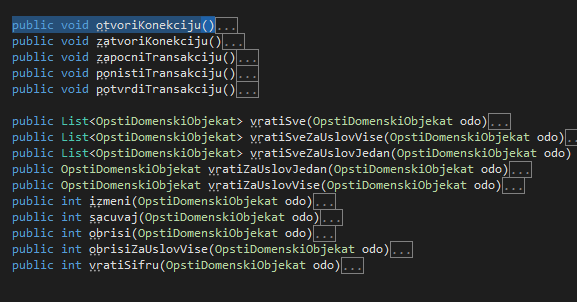
return Sesija.Broker.dajSesiju().obrisi(odo);

}

}

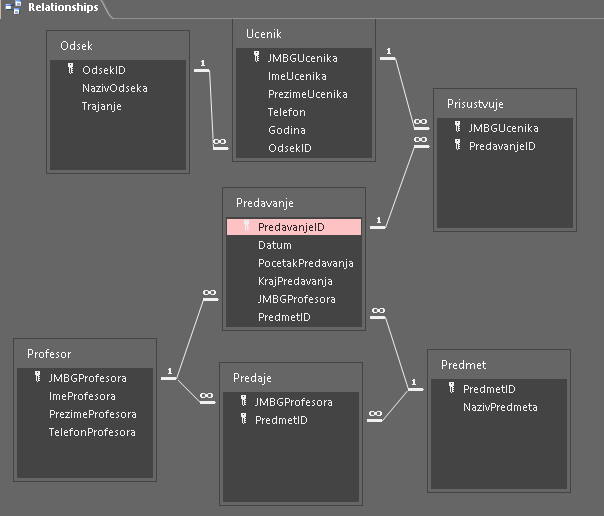
### 3.4.3. Брокер базе података

Брокер базе података је софтверска класа одговорна за комуникацју између пословне логике и складишта података. Другим речима, пројектује се како би обезбедио перзистентни сервис објектима доменских класа који се чувају у бази података. Класа Брокер представља перзистентни оквир који посредује у свим операцијама над базом података и реализује следеће методе:



### 3.4.4. Пројектовање складишта података

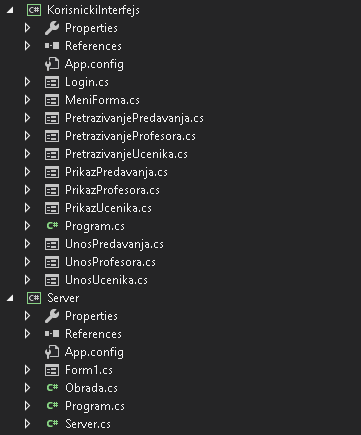
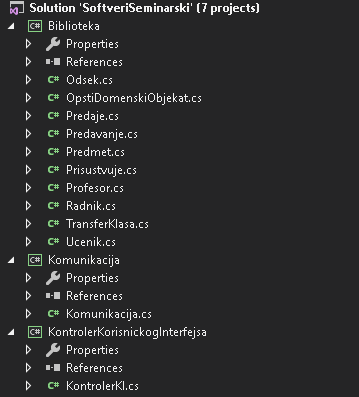
На основу структуре софтверских класа пројектоване су табеле (складишта података) релационог система за управљање базом података. У овом раду је коришћен MS Access.

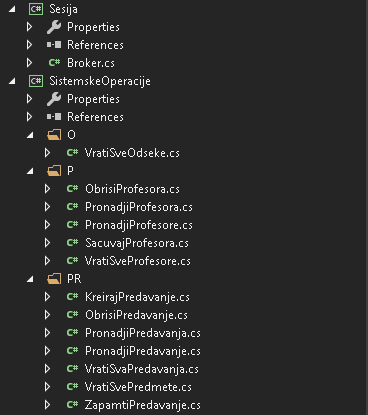


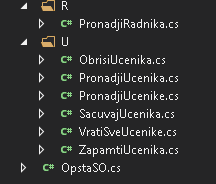
# 4. Имплементација

Софтверски систем, резултат овог рада, развијен је у програмском језику C#, а изворни код је приложен на пратећем ЦД-у. Систем је пројектован као клијент-сервер. Као развојно окружење коришћен је Microsoft Visual Studio 2017. Као систем за управљањем базом података коришћен је MS Access.

На основу архитектуре софтверског система добијене су следеће софтверске класе:







# 5. Тестирање

Сваки од имплементираних случајева коришћења је тестиран. Приликом тестирања сваког случаја коришћења поред унетих правилних података уношени су и неправилни подаци да би се утврдило какав ће резултат извршења бити. На основу извршених тестирања отклоњени су уочени недостаци.

# 6. Литература

Пројектовање софтвера – Скрипта, др Синиша Влајић, Београд - 2015.