Универзитет у Београду

Факултет оpганизационих наука

**Семинарски рад**

тема:

Софтверски систем за плесни клуб у Јава развојном окружењу

Професор: Студент:

Др Синиша Влајић Зоја Миљанић 275/16

Садржај

[1. Прикупљање захтјева од корисника 3](#_Toc75088100)

[1.2 Случајеви коришћења 4](#_Toc75088101)

[1.2.1 СК01: Пријава администратора 5](#_Toc75088102)

[1.2.2 СК02: Унос плесача 6](#_Toc75088103)

[1.2.3 СК03: Претрага плесача 7](#_Toc75088104)

[1.2.4 СК04: Измјена података о плесачу 8](#_Toc75088105)

[1.2.5 СК05: Брисање плесача 9](#_Toc75088106)

[1.2.6 СК06: Унос тренера 10](#_Toc75088107)

[1.2.7 СК07: Измјена тренера 11](#_Toc75088108)

[1.2.8 СК08: Брисање тренера 12](#_Toc75088109)

[1.2.9 СК09: Претрага тренера 13](#_Toc75088110)

[1.2.10 СК10: Додај дневни распоред 14](#_Toc75088111)

[1.2.11 СК11: Брисање дневног распореда 15](#_Toc75088112)

[2. Анализа 16](#_Toc75088113)

[2.1 Системски дијаграми секвенци 16](#_Toc75088114)

[2.1.1 ДС1: Дијаграм секвенци случаја коришћења - Пријава администратора 16](#_Toc75088115)

[2.1.2 ДС2: Дијаграм секвенци случаја коришћења - Унос плесача 18](#_Toc75088116)

[2.1.3 ДС3 Дијаграм секвенци случаја коришћења - Претрага плесача 19](#_Toc75088117)

[2.1.4 ДС4 Дијаграм секвенци случаја коришћења - Измјена података о плесачу 22](#_Toc75088118)

[2.1.5 ДС5: Дијаграм секвенци случаја коришћења - Брисање плесача 25](#_Toc75088119)

[2.1.6 ДС6: Дијаграм секвенци случаја коришћења - Унос тренера 28](#_Toc75088120)

[2.1.7 ДС7: Дијаграм секвенци случаја коришћења - Претрага тренера 30](#_Toc75088121)

[2.1.8 ДС8: Дијаграм секвенци случаја коришћења - Измјена тренера 31](#_Toc75088122)

[2.1.9 ДС9: Дијаграм секвенци случаја коришћења - Брисање тренера 35](#_Toc75088123)

[2.1.10 ДС10: Дијаграм секвенци случаја коришћења - Унос дневног распореда 38](#_Toc75088124)

[2.1.11 ДС11: Дијаграм секвенци случаја коришћења - Брисање дневног распореда 40](#_Toc75088125)

[2.2 Понашање софтверског система-Дефинисање уговора о системским операцијама 43](#_Toc75088126)

[2.3 Структура софтверског система - Концептуални (доменски) модел 46](#_Toc75088127)

[2.4 Структура софтверског система - Релациони модел 49](#_Toc75088128)

[3.Фаза пројектовања 52](#_Toc75088129)

[3.1 Архитектура софтверског система 52](#_Toc75088130)

[3.1.1 Пројектовање кросничког интерфејса 53](#_Toc75088131)

[3.2 Пројектовање екранске форме 53](#_Toc75088132)

[3.2.1 СК1: Пријављивање администратора 54](#_Toc75088133)

[3.2.2 СК2: УносПлесача 56](#_Toc75088134)

[3.2.3 СК3: Претрага плесача 58](#_Toc75088135)

[3.2.4 СК4: Измјена података о плесачу 60](#_Toc75088136)

[3.2.5 СК5: Брисање плесача 63](#_Toc75088137)

[3.2.6 СК6: Унос тренера 65](#_Toc75088138)

[3.2.7 СК7: Измјена тренера 67](#_Toc75088139)

[3.2.8 СК8: Брисање тренера 69](#_Toc75088140)

[3.2.9 СК9: Претрага тренера 72](#_Toc75088141)

[3.2.10 СК10: Додај дневни распоред 73](#_Toc75088142)

[3.3 Пројектовање контролера корисничког интерфејса 76](#_Toc75088143)

[3.4 Пројектовање апликационе логике 76](#_Toc75088144)

[3.4.1 Контролер апликационе логике 77](#_Toc75088145)

[3.5 Пословна логика 77](#_Toc75088146)

[3.5.1 Пројектовање понашања (системске операције) 77](#_Toc75088147)

[3.6 Пројектовање структуре софтверског система (доменске класе) 90](#_Toc75088148)

[3.6.1 Пројектовање складишта података 91](#_Toc75088149)

[4. Имплементација 95](#_Toc75088150)

[5.Закључак 99](#_Toc75088151)

[6. Литература 100](#_Toc75088152)

# 1. Прикупљање захтјева од корисника

Потребно је направити софтверски систем који ће се користити за потребе плесног клуба, путем кога ће се водити евиденција о плесачима и тренерима.

Апликација треба да обезбиједи администратору лако управљање софтвером тако што се администратор прво пријављује и врши операције које су му потребне. Систем омогућава евиденцију плесача и термина тренинга за сваког члана.

Приликом прављења распореда потребно је сврстати члана у одговарајућу групу, по терминима и назначити који тренер ће водити ту групу. Систем дозвољава унос чланова, брисање постојећег члана (у случају да он жели да се испише из клуба) и измјену података о постојећем плесачу. Такође је могуће уношење нове врсте тренинга, измјена података о већ постојећем тренингу, брисање тренинга и претрага по тренингу.

## 1.2 Случајеви коришћења

У овом систему постоје следећи случајеви коришћења (слика):

1. Пријава администратора
2. Унос плесача
3. Претрага плесача
4. Измјена података о плесачу
5. Брисање плесача
6. Унос тренера
7. Претрага тренера
8. Измјена тренера
9. Брисање тренера
10. Унос дневног распореда (сложен случај коришћења)
11. Брисање дневног распореда (сложен случај коришћења)



*Слика 1 - Дијаграм случајева коришћења*

### 1.2.1 СК01: Пријава администратора

НАЗИВ СК**:**

Пријава администратора

АКТОРИ СК**:**Администратор

УЧЕСНИЦИ СК**:**

Администратор и систем (програм)

ПРЕДУСЛОВИ**:**Систем је укључен. Систем приказује форму за пријаву администратора.

**ОСНОВНИ СЦЕНАРИО СЛУЧАЈА КОРИШЋЕЊА:**

1. Администратор уноси корисничко име и лозинку. (АПУСО)
2. Администратор контролише да ли је правилно унио неопходне податке. (АНСО)
3. Администратор позива систем да га пријави. (АПСО)
4. Систем претражује да ли постоји администратор међу постојећима. (СО)
5. Систем приказује администратору поруку „Успјешно сте се пријавили на систем”. (ИА)

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

5.1 Уколико систем препозна да унесени подаци нису у складу са постојећим, приказује се порука: „Креденцијали нису усправни ”. (ИА)

### 1.2.2 СК02: Унос плесача

НАЗИВ СК**:**

Унос плесача

АКТОРИ СК**:**Администратор

УЧЕСНИЦИ СК**:**

Администратор и систем (програм)

ПРЕДУСЛОВИ**:**Систем је укључен и администратор је улогован под својом шифром. Систем приказује форму за рад са плесачима.

**ОСНОВНИ СЦЕНАРИО СЛУЧАЈА КОРИШЋЕЊА:**

1. Администратор уноси податке о плесачу. (АПУСО)
2. Администратор контролише да ли је правилно унио податке о плесачу. (АНСО)
3. Администратор позива систем да креира плесача. (АПСО)
4. Систем креира плесача. (СО)
5. Систем приказује администратору плесача и поруку „Систем је креирао плесача“. (ИА)

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

5.1 Ако систем не може да креира плесача, приказује поруку: „Систем не може да креира плесача“. Прекида се извршење сценарија. (ИА)

### 1.2.3 СК03: Претрага плесача

НАЗИВ СК**:**

Претрага плесача

АКТОРИ СК**:**Администратор

УЧЕСНИЦИ СК**:**

Администратор и систем (програм)

ПРЕДУСЛОВИ**:**Систем је укључен. Администратор је улогован под својом шифром. Систем приказује форму за рад са плесачима.

**ОСНОВНИ СЦЕНАРИО СЛУЧАЈА КОРИШЋЕЊА:**

1. Администратор уноси вриједност по којој претражује плесаче. (АПУСО)
2. Администратор позива систем да пронађе плесаче по задатој вриједности. (АПСО)
3. Систем претражује плесаче по задатој вриједности. (СО)
4. Систем приказује администратору поруку „Систем је пронашао клијнте по задатој вриједности”. (ИА)

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

4.1 Ако систем не може да пронађе плесаче по задатој вриједности, приказује поруку: „Систем не може да пронађе плесаче по задатој вриједности”. Прекида се извршење сценарија. (ИА)

### 1.2.4 СК04: Измјена података о плесачу

НАЗИВ СК**:**

Измјена података о плесачу

АКТОРИ СК**:**Администратор

УЧЕСНИЦИ СК**:**

Администратор и систем (програм)

ПРЕДУСЛОВИ**:**Систем је укључен. Администратор је пријављен на систем и систем приказује налог на који је корисник пријављен. Систем приказује форму за рад са плесачима.

**ОСНОВНИ СЦЕНАРИО СЛУЧАЈА КОРИШЋЕЊА:**

1. Администратор бира плесача којег жели да измјени. (АПУСО)
2. Администратор позива систем да учита податке о изабраном плесачу. (АПСО)
3. Систем проналази податке о изабраном плесачу. (СО)
4. Систем приказује администратору жељеног плесача и поруку: „Систем је пронашао податке о изабраном плесачу“. (ИА)
5. Администратор мијења податке о плесачу. (АПУСО)
6. Администратор контролише да ли је правилно унио податке о плесачу. (АНСО)
7. Администратор позива систем да запамти податке о плесачу. (АПСО)
8. Систем памти податке о плесачу. (СО)
9. Систем приказује администратору поруку: „Успјешно сте промијенили податке о плесачу”. (ИА)

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

4.1 Aко систем не може да нађе податке о изабраном плесачу приказује поруку: „Систем не може да нађе податке о изабраном плесачу”. Прекида се извршење сценарија. (ИА)

9.1 Ако систем не може да измијени податке, приказује кориснику поруку: „Систем не може да изврши измјену података о плесачу”. (ИА)

### 1.2.5 СК05: Брисање плесача

НАЗИВ СК**:**

Брисање плесача

АКТОРИ СК**:**Администратор

УЧЕСНИЦИ СК**:**

Администратор и систем (програм)

ПРЕДУСЛОВИ**:**Систем је укључен. Администратор је пријављен на систем и систем приказује форму за рад са плесачима. Учитана је листа плесача.

**ОСНОВНИ СЦЕНАРИО СЛУЧАЈА КОРИШЋЕЊА:**

1. Администратор уноси вриједност по којој претражује плесаче. (АПУСО)
2. Администратор позива систем да нађе плесача по задатим критеријумима. (АПСО)
3. Систем тражи плесаче по задатим критеријумима. (СО)
4. Систем приказује администратору податке о пронађеним плесачима и поруку: „Систем је пронашао плесаче по задатом критеријуму”.(ИА)
5. Администратор позива систем да обрише изабраног плесача. (АПСО)
6. Систем брише плесача. (СО)
7. Систем приказује администратору поруку: „Успјешно сте обрисали изабраног плесача”. (ИА)

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

4.1 Ако систем не може да нађе плесаче са задатим вриједностима, приказује се порука: „Систем не може да пронађе плесаче по задатом критеријуму”.(ИА)

7.1 Ако систем не може да нађе податке о изабраном плесачу приказује се порукa: „Систем не може да пронађе податке о плесачу“. Прекида се извршење сценарија. (ИА)

### 1.2.6 СК06: Унос тренера

НАЗИВ СК**:**

Унос тренера

АКТОРИ СК**:**Администратор

УЧЕСНИЦИ СК**:**

Администратор и систем (програм)

ПРЕДУСЛОВИ**:**Систем је укључен. Администратор је пријављен на систем и систем приказује налог на који је корисник пријављен. Систем приказује форму за унос новог тренера.

**ОСНОВНИ СЦЕНАРИО СЛУЧАЈА КОРИШЋЕЊА:**

1. Администратор уноси податке о новом тренеру. (АПУСО)
2. Администратор контролише да ли је правилно унио податке о тренеру. (АНСО)
3. Администратор позива систем да креира новог тренера. (АПСО)
4. Систем креира новог тренера. (СО)
5. Систем приказује администратору новог тренера и поруку „Систем је креирао тренера“. (ИА)

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

5.1 Ако систем не може да унесе новог тренера, приказује поруку: „Систем није успјешно унио новог тренера”. (ИА)

### 1.2.7 СК07: Измјена тренера

НАЗИВ СК**:**

Измјена тренера

АКТОРИ СК**:**Администратор

УЧЕСНИЦИ СК**:**

Администратор и систем (програм)

ПРЕДУСЛОВИ**:**Систем је укључен. Администратор је пријављен на систем и систем приказује налог на који је корисник пријављен. Учитана је листа тренера.

**ОСНОВНИ СЦЕНАРИО СЛУЧАЈА КОРИШЋЕЊА:**

1. Администратор бира тренера којег жели да измјени. (АПУСО)
2. Администратор позива систем да учита податке о изабраном тренеру. (АПСО)
3. Систем проналази податке о изабраном тренеру. (СО)
4. Систем приказује администратору податке о изабраном тренеру и поруку: „Систем је успјешно пронашао податке о изабраном тренеру“. (ИА)
5. Администратор уноси измијењене податке о тренеру.(АПУСО)
6. Администратор контролише да ли је коректно унио податке о тренеру. (АНСО)
7. Администратор позива систем да запамти податке о тренеру. (АПСО)
8. Систем памти податке. (СО)
9. Систем приказује администратору поруку: „Систем је успјешно измијенио податке о тренеру”. (ИА)

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

4.1 Ако систем не може да нађе податке о задатом тренингу, приказује се порука: „Систем није пронашао податке о тренеру“. (ИА)

9.1 Ако систем не може да измијени податке, приказује поруку: „Систем није измијенио податке о тренеру”. (ИА)

### 1.2.8 СК08: Брисање тренера

НАЗИВ СК**:**

Брисање тренера

АКТОРИ СК**:**Администратор

УЧЕСНИЦИ СК**:**

Администратор и систем (програм)

ПРЕДУСЛОВИ**:**Систем је укључен. Администратор је пријављен на систем и учитана је листа тренера.

**ОСНОВНИ СЦЕНАРИО СЛУЧАЈА КОРИШЋЕЊА:**

1. Администратор уноси вриједност по којој претражује тренере. (АПУСО)
2. Администратор позива систем да нађе тренера по задатом критеријуму. (АПСО)
3. Систем тражи тренера са задатим критеријумом. (СО)
4. Систем приказује администратору податке о пронађеним тренерима и поруку „Систем је пронашао тренере по задатом критеријуму”.(ИА)
5. Администратор позива систем да обрише тренера. (АПСО)
6. Систем брише изабраног тренера. (СО)
7. Систем приказује администратору поруку: „Систем је успјешно обрисао тренера”. (ИА)

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

4.1 Ако систем не може да нађе тренера који се поклапа са критеријумима, систем приказује поруку: „Систем није пронашао тренере по задатом критеријуму”. (ИА)

7.1 Ако систем не може да обрише изабраног тренера, приказује поруку: „Систем није успјешно обрисао тренера”. (ИА)

### 1.2.9 СК09: Претрага тренера

НАЗИВ СК**:**

Претрага тренера

АКТОРИ СК**:**Администратор

УЧЕСНИЦИ СК**:**

Администратор и систем (програм)

ПРЕДУСЛОВИ**:**Систем је укључен. Администратор је пријављен на систем и систем приказује форму за претрагу тренинга. Учитана је листа тренера.

**ОСНОВНИ СЦЕНАРИО СЛУЧАЈА КОРИШЋЕЊА:**

1. Администратор уноси вриједност по којој жели да претражи тренере. (АПУСО)
2. Администратор позива систем да пронађе тренере по задатом критеријуму. (АПСО)
3. Систем претражује тренере. (СО)
4. Систем приказује администратору податке о пронађеним тренерима и поруку: поруку „Систем је пронашао тренере по задатом критеријуму”.(ИА)

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

4.1 Ако систем не може да пронађе ниједног тренера који задовољава задате критеријуме, приказује поруку: „Систем није пронашао тренере по задатом критеријуму”. (ИА)

### 1.2.10 СК10: Додај дневни распоред

НАЗИВ СК**:**

Додај дневни распоред

АКТОРИ СК**:**Администратор

УЧЕСНИЦИ СК**:**

Администратор и систем (програм)

ПРЕДУСЛОВИ**:**Систем је укључен. Администратор је пријављен на систем и систем приказује форму за евиденцију дневног распореда. Учитана је листа типова плеса и тренера.

**ОСНОВНИ СЦЕНАРИО СЛУЧАЈА КОРИШЋЕЊА:**

1. Администратор уноси податке о дневном распореду. (АПУСО)
2. Администратор контролише да ли је унио коректно све податке. (АНСО)
3. Администратор позива систем да запамти податке о дневном распореду. (АПСО)
4. Систем памти податке о дневом распореду. (СО)
5. Систем приказује администратору запамћен дневни распоред и поруку: “Систем је успјешно евидентирао дневни распоред”. (ИА)

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

5.1 Ако систем не може да запамти дневни распоред, приказује поруку: „Систем не може успјешно да евидентира дневни распоред“. (ИА)

### 1.2.11 СК11: Брисање дневног распореда

НАЗИВ СК**:**

Брисање дневног распореда

АКТОРИ СК**:**Администратор

УЧЕСНИЦИ СК**:**

Администратор и систем (програм)

ПРЕДУСЛОВИ**:**Систем је укључен. Администратор је пријављен на систем и систем приказује форму за рад са дневним распоредима и тренинзима. Учитана је листа дневних распореда.

**ОСНОВНИ СЦЕНАРИО СЛУЧАЈА КОРИШЋЕЊА:**

1. Администратор уноси вриједност по којој претражује дневни распоред. (АПУСО)
2. Администратор позива систем да нађе дневни распоред по задатом критеријуму. (АПСО)
3. Систем тражи дневне распореде по задатом критеријуму. (СО)
4. Систем приказује администратору податке о пронађеним дневним распоредима и поруку: „Систем је пронашао дневне распореде по задатом критеријуму”.(ИА)
5. Администратор позива систем да обрише изабрани дневни распоред. (АПСО)
6. Систем брише дневни распоред. (СО)
7. Систем приказује администратору поруку: „Успјешно сте обрисали изабрани дневни распоред”. (ИА)

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

4.1 Ако систем не може да нађе дневне распореде са задатим вриједностима, приказује се порука: „Систем не може да пронађе дневни распоред по задатом критеријуму”.(ИА)

7.1 Ако систем не може да нађе податке о изабраном дневном распореду приказује се порукa: „Систем не може да пронађе податке о дневном распореду“. Прекида се извршење сценарија. (ИА)

# 2. Анализа

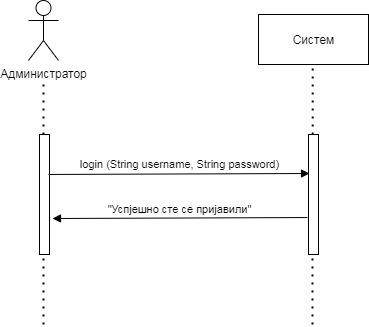
## 2.1 Системски дијаграми секвенци

Путем системских дијаграма секвенци и системских операција ћемо дати опис понашања система, након чега ћемо, уз помоћ концептуалних и релационих модела, представити структуру система.

За сваки случај коришћења, који је уочен у фази прикупљања захтјева, приказаћемо системске дијаграме секвенци, којима моделујемо интеракцију између актора и система.

### 2.1.1 ДС1: Дијаграм секвенци случаја коришћења - Пријава администратора

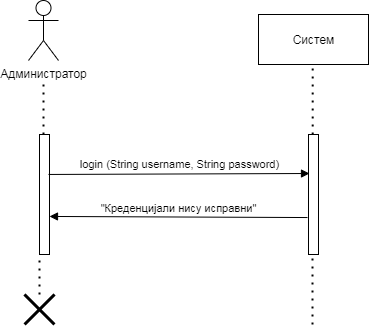
1. Администратор позива систем да пронађе у бази администратора са задатим подацима. (АПСО)
2. Систем приказује администратору поруку: „Успјешно сте се пријавили“. (ИА)



*Слика 2: ДС Пријава администратора (основни сценарио)*

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

5.1 Ако систем није успио да пронађе у бази администратора са тим креденцијалима, приказује поруку: „Креденцијали нису исправни”. (ИА)



*Слика 3: ДС Пријава администратора (алтернативни сценарио)*

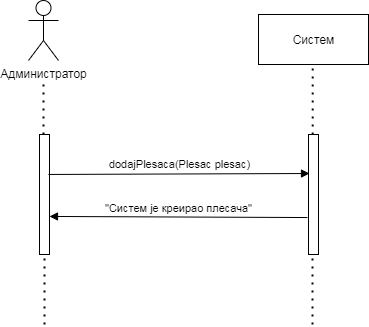
Са наведених дијаграма секвенци се уочава једна системска операција коју треба

пројектовати:

1. *signal* **login(String username, String password)**

### 2.1.2 ДС2: Дијаграм секвенци случаја коришћења - Унос плесача

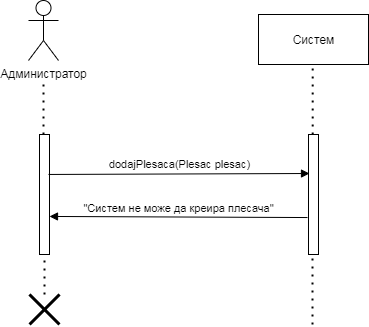
1. Администратор позива систем да креира плесача. (АПСО)
2. Систем приказује администратору плесача и поруку „Систем је креирао плесача“. (ИА)



*Слика 4: ДС Унос плесача (основни сценарио)*

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

2.1 Ако систем не може да креира плесача, приказује поруку: „Систем не може да креира плесача“. (ИА)



*Слика 5: ДС Унос плесача (алтернативни сценарио)*

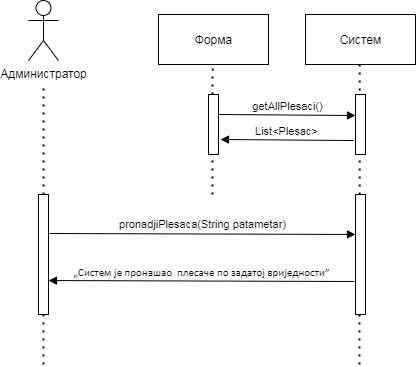
Са наведених дијаграма секвенци се уочава једна системска операција коју треба

пројектовати:

1. *signal* **dodajPlesaca(Plesac plesac)**

### 2.1.3 ДС3 Дијаграм секвенци случаја коришћења - Претрага плесача

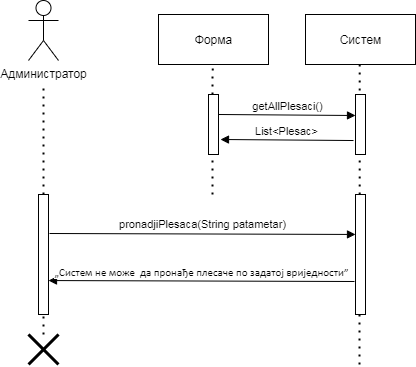
1. Администратор позива систем да пронађе плесаче по задатој вриједности. (АПСО)
2. Систем приказује администратору поруку „Систем је пронашао плесаче по задатој вриједности”. (ИА)



*Слика 6: Претрага плесача (основни сценарио)*

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

2.1 Ако систем не може да пронађе плесаче по задатој вриједности, приказује поруку: „Систем не може да пронађе плесаче по задатој вриједности”. (ИА)



*Слика 7: ДС Претрага плесача (алтернативни сценарио)*

Са наведених дијаграма секвенци се уочава једна системска операција коју треба

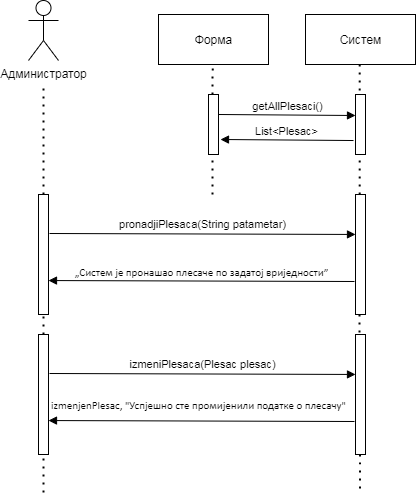
пројектовати:

1. *signal* **getAllPlesaci()**

2. *signal* **pronadjiPlesaca(String parametar)**

### 2.1.4 ДС4 Дијаграм секвенци случаја коришћења - Измјена података о плесачу

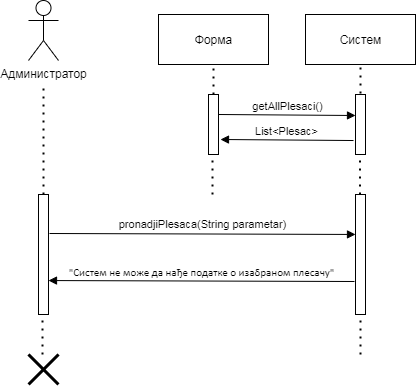
1. Администратор позива систем да учита податке о изабраном плесачу. (АПСО)
2. Систем приказује администратору жељеног плесача и поруку: „Систем је пронашао податке о изабраном плесачу“. (ИА)
3. Администратор позива систем да запамти податке о плесачу. (АПСО)
4. Систем приказује администратору поруку: „Успјешно сте промијенили податке о плесачу”. (ИА)



*Слика 8: ДС Измјена плесача (основни сценарио)*

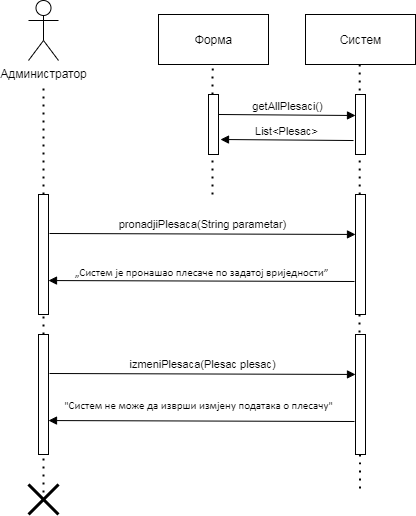
**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

2.1 Aко систем не може да нађе податке о изабраном плесачу приказује поруку: „Систем не може да нађе податке о изабраном плесачу”. (ИА)



*Слика 9: ДС Измјена плесача (алтернативни сценарио 1)*

4.1 Ако систем не може да измијени податке, приказује кориснику поруку: „Систем не може да изврши измјену података о плесачу”. (ИА)



*Слика 10: ДС Измјена плесача (алтернаривни сценарио 2)*

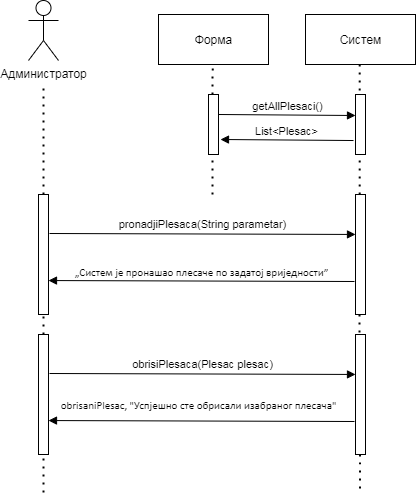
Са наведених дијаграма секвенци се уочавају системске операције које треба

пројектовати:

1. *signal* **getAllPlesaci()**
2. *signal* **pronadjiPlesaca(String parametar)**
3. *signal* **izmijeniPlesaca(Plesac plesac)**

### 2.1.5 ДС5: Дијаграм секвенци случаја коришћења - Брисање плесача

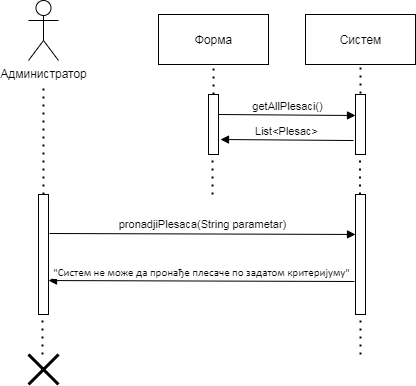
1. Администратор позива систем да нађе плесача по задатим критеријумима. (АПСО)
2. Систем приказује администратору податке о пронађеним плесачима и поруку: „Систем је пронашао плесаче по задатом критеријуму”.(ИА)
3. Администратор позива систем да обрише изабраног плесача. (АПСО)
4. Систем приказује администратору поруку: „Успјешно сте обрисали изабраног плесача”. (ИА)



*Слика 11: ДС Брисање плесача (основни сценарио)*

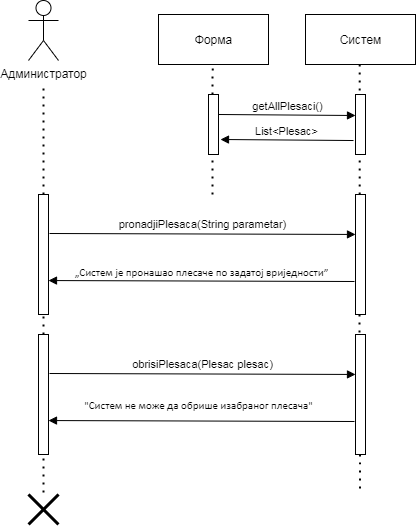
**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

2.1 Ако систем не може да нађе плесаче са задатим вриједностима, приказује се порука: „Систем не може да пронађе плесаче по задатом критеријуму”.(ИА)



*Слика 12: ДС Брисање плесача (алтернативни сценарио 1)*

4.1 Ако систем не може да обрише плесача, приказује поруку: „Систем не може да обрише изабраног плесача”. (ИА)



*Слика 13: ДС Брисање плесача (алтернативни сценарио 2)*

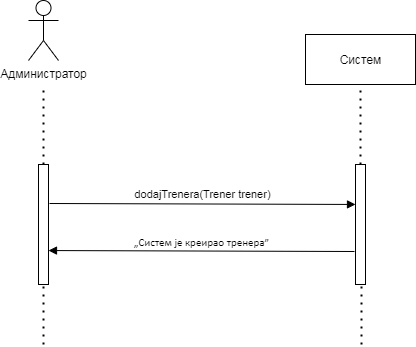
Са наведених дијаграма секвенци се уочавају системске операције које треба

пројектовати:

1. *signal* **getAllPlesaci()**
2. *signal* **pronadjiPlesaca(String parametar)**
3. *signal* **obrisiPlesaca(Plesac plesac)**

### 2.1.6 ДС6: Дијаграм секвенци случаја коришћења - Унос тренера

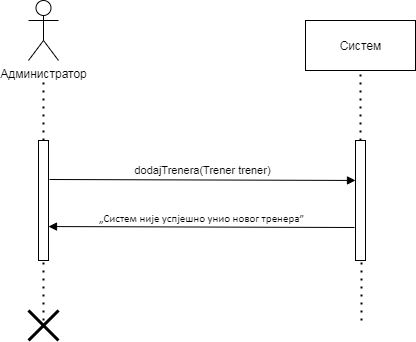
1. Администратор позива систем да креира новог тренера. (АПСО)
2. Систем приказује администратору новог тренера и поруку: „Систем је креирао тренера“. (ИА)



*Слика 14: ДС Унос тренера (основни сценарио)*

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

2.1 Ако систем не може да унесе новог тренера, приказује поруку: „Систем није успјешно унио новог тренера”. (ИА)



*Слика 15: ДС Унос тренера (алтернативни сценарио)*

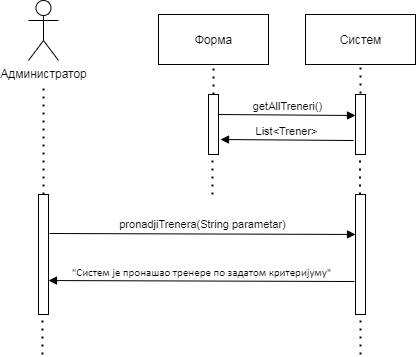
Са наведених дијаграма секвенци се уочава системска операција коју треба

пројектовати:

1. *signal* **dodajTrenera(Trener trener)**

### 2.1.7 ДС7: Дијаграм секвенци случаја коришћења - Претрага тренера

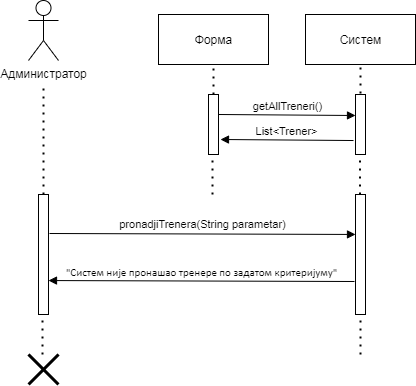
1. Администратор позива систем да пронађе тренере по задатом критеријуму. (АПСО)
2. Систем приказује администратору податке о пронађеним тренерима и поруку: поруку „Систем је пронашао тренере по задатом критеријуму”.(ИА)



*Слика 16: ДС Претрага тренера (основни сценарио)*

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

2.1 Ако систем не може да пронађе ниједног тренера који задовољава задате критеријуме, приказује поруку: „Систем није пронашао тренере по задатом критеријуму”. (ИА)



*Слика 17: ДС Претрага тренера (алтернативни сценарио)*

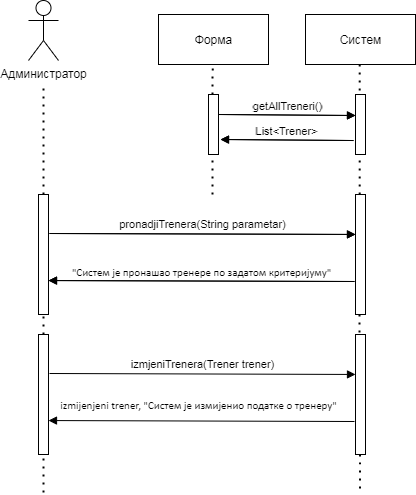
Са наведених дијаграма секвенци се уочавају системске операције које треба

пројектовати:

1. *signal* **getAllTreneri**(List <Trener>)
2. *signal* **pronadjiTrenera**(String parametar)

### 2.1.8 ДС8: Дијаграм секвенци случаја коришћења - Измјена тренера

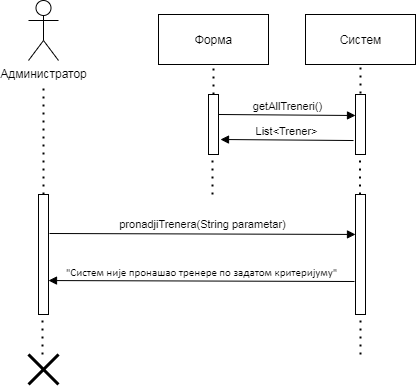
1. Администратор позива систем да учита податке о изабраном тренеру. (АПСО)
2. Систем приказује администратору податке о изабраном тренеру и поруку: „Систем је успјешно пронашао податке о изабраном тренеру“. (ИА)
3. Администратор позива систем да запамти податке о тренеру. (АПСО)
4. Систем приказује администратору поруку: „Систем је успјешно измијенио податке о тренеру”. (ИА)



*Слика 18: ДС Измјена тренера (основни сценарио)*

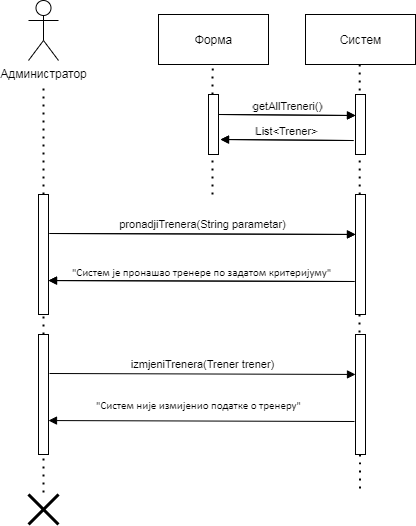
**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

2.1 Ако систем не може да нађе податке о задатом тренингу, приказује порукy: „Систем није пронашао податке о тренеру“. (ИА)



*Слика 19: ДС Измјена тренера (алтернативни сценарио 1)*

4.1 Ако систем не може да измијени податке, приказује поруку: „Систем није измијенио податке о тренеру”. (ИА)



*Слика 20: ДС Измјена тренера (алтернативни сценарио 2)*

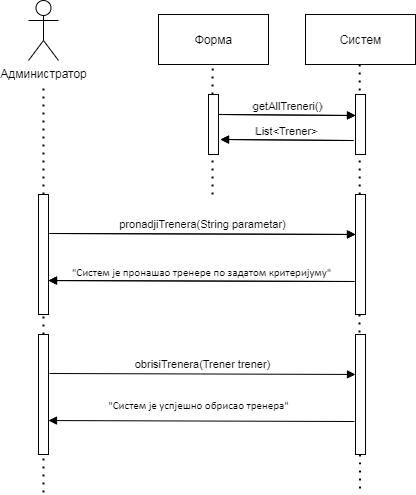
Са наведених дијаграма секвенци се уочавају системске операције које треба

пројектовати:

1. *signal* **getAllTreneri(List<Trener>)**
2. *signal* **pronadjiTrenera(String parametar)**
3. *signal* **izmjeniTrenera(Trener trener)**

### 2.1.9 ДС9: Дијаграм секвенци случаја коришћења - Брисање тренера

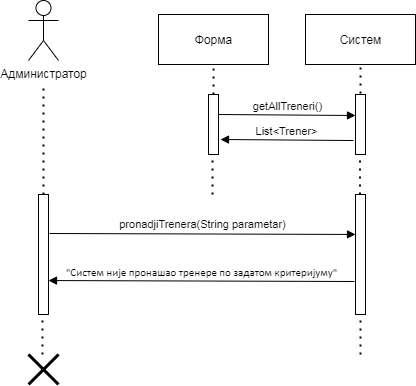
1. Администратор позива систем да нађе тренера по задатом критеријуму. (АПСО)
2. Систем приказује администратору податке о пронађеним тренерима и поруку „Систем је пронашао тренере по задатом критеријуму”.(ИА)
3. Администратор позива систем да обрише тренера. (АПСО)
4. Систем приказује администратору поруку: „Систем је успјешно обрисао тренера”. (ИА)



*Слика 21: ДС Брисање тренера (основни сценарио)*

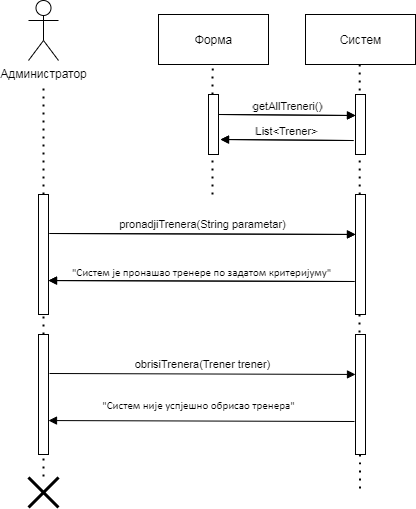
**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

2.1 Ако систем не може да нађе тренера који се поклапа са критеријумима, систем приказује поруку: „Систем није пронашао тренере по задатом критеријуму”. (ИА)



*Дијаграм 22: ДС Брисање тренера (алтернативни сценарио 1)*

4.1 Ако систем не може да обрише изабраног тренера, приказује поруку: „Систем није успјешно обрисао тренера”. (ИА)



*Слика 23: ДС Брисање тренера (алтернативни сценарио 2)*

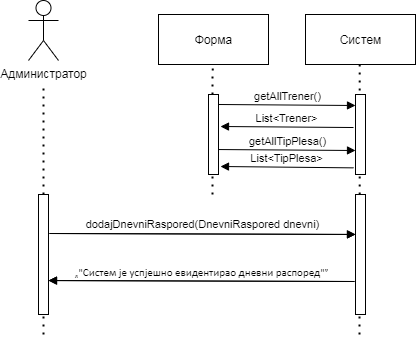
Са наведених дијаграма секвенци се уочавају системске операције које треба

пројектовати:

1. *signal* **getAllTreneri(List<Trener>)**
2. *signal* **pronadjiTrenera(String parametar)**
3. *signal* **obrisiTrenera(Trener trener)**

### 2.1.10 ДС10: Дијаграм секвенци случаја коришћења - Унос дневног распореда

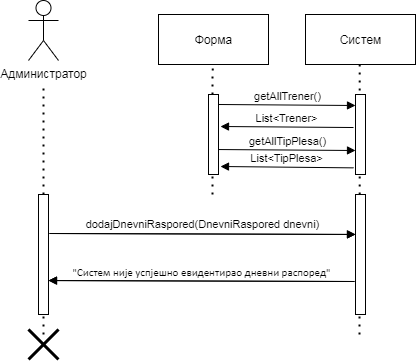
1. Администратор позива систем да запамти податке о дневном распореду. (АПСО)
2. Систем приказује администратору запамћен дневни распоред и поруку: “Систем је успјешно евидентирао дневни распоред”. (ИА)



*Слика 24: ДС Унос дневног распореда (основни сценарио)*

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

2.1 Ако систем не може да запамти дневни распоред, приказује поруку: „Систем не може успјешно да евидентира дневни распоред“. (ИА)



*Слика 25: Унос дневног распореда (алтернативни сценарио)*

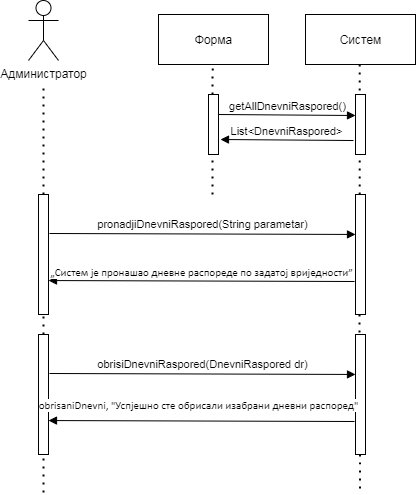
Са наведених дијаграма секвенци се уочавају системске операције које треба

пројектовати:

1. *signal* **getAllTrener(List<Trener>)**
2. *signal* **getAllTipPlesa(List<TipPlesa>)**
3. *signal* **dodajDnevniRaspored(DnevniRaspored dnevniRaspored)**

### 2.1.11 ДС11: Дијаграм секвенци случаја коришћења - Брисање дневног распореда

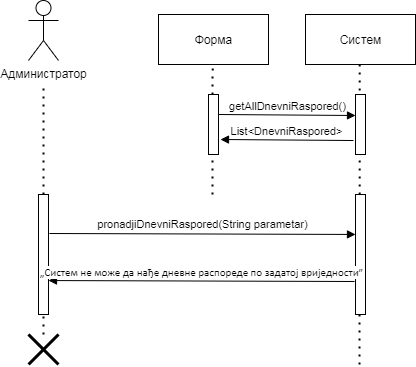
1. Администратор позива систем да нађе дневни распоред по задатим критеријумима. (АПСО)
2. Систем приказује администратору податке о пронађеним дневним распоредима и поруку: „Систем је пронашао дневне распореде по задатом критеријуму”.(ИА)
3. Администратор позива систем да обрише изабрани дневни распоред. (АПСО)
4. Систем приказује администратору поруку: „Успјешно сте обрисали изабрани дневни распоред”. (ИА)



*Слика 26: ДС Брисање дневног распореда (основни сценарио)*

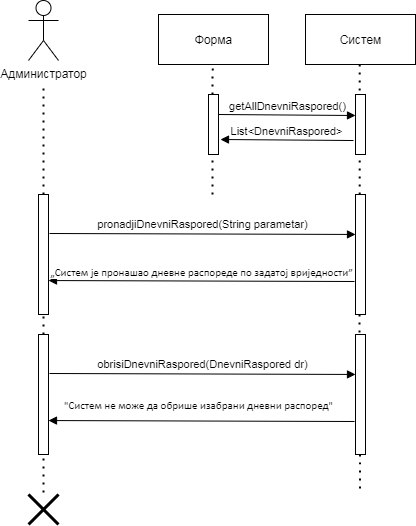
**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

2.1 Ако систем не може да нађе дневни распоред са задатим вриједностима, приказује се порука: „Систем не може да пронађе дневни распоред по задатом критеријуму”.(ИА)



*Слика 27: ДС Брисање дневног распореда (алтернативни сценарио 1)*

4.1 Ако систем не може да обрише дневни распоред, приказује поруку: „Систем не може да обрише изабрани дневни распоред”. (ИА)



*Слика 28: ДС Брисање дневног распореда (алтернативни сценарио 2)*

Са наведених дијаграма секвенци се уочавају системске операције које треба

пројектовати:

1. *signal* **getAllDnevniRaspored ()**
2. *signal* **pronadjiDnevniRaspored(String parametar)**
3. *signal* **obrisiDnevniRaspored(DnevniRaspored dnevniRaspored)**

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

1. *signal* **login(String username, String password)**
2. *signal* **getAllPlesaci(List<Plesac>)**
3. *signal* **dodajPlesaca(Plesac plesac)**
4. *signal* **pronadjiPlesaca(String parametar)**
5. *signal* **izmijeniPlesaca(Plesac plesac)**
6. *signal* **obrisiPlesaca(Plesac plesac)**
7. *signal* **getAllTrener(List<Trener>)**
8. *signal* **dodajTrenera(Trener trener)**
9. *signal* **pronadjiTrenera(String parametar)**
10. *signal* **izmijeniTrenera(Trener trener)**
11. *signal* **obrisiTrenera(Trener trener)**
12. *signal* **getAllTipPlesa(List<TipPlesa>)**
13. *signal* **dodajDnevniRaspored(DnevniRaspored dnevniRaspored)**
14. *signal* **getAllDnevniRaspored ()**
15. *signal* **pronadjiDnevniRaspored(String parametar)**
16. *signal* **obrisiDnevniRaspored(DnevniRaspored dnevniRaspored)**

## 2.2 Понашање софтверског система-Дефинисање уговора о системским операцијама

Уговори о системским операцијама описују шта операција треба да ради, без детаља о имплементацији. Један уговор се односи на једну системску операцију. Уговор се састоји од следећих елемената:

*Операција*– назив операције са улазним елементима;

*Веза са СК* – називи случајева коришћења у којима се системска операција позива; *Предуслов*– стање у ком систем мора бити пре извршења системске операције; *Постуслови*– стање у ком систем мора бити након извршења системске операције, иначе се поништава операција.

**Уговор У1: Login**

Операција: Login(String username, String password): signal;

Веза са СК: СК1

Предуслов: */*

Постуслов: */*

**Уговор У2: DodajPlesaca**

Операција: **dodajPlesaca(Plesac plesac)**: signal;

Веза са СК: СК2

Предуслов: *Вриједносна и структурна ограничења над објектом Плесач морају бити задовољена.*

Постуслов: *Подаци о плесачу су запамћени.*

**Уговор У3: GetAllPlesaci**

Операција: **getAllPlesaci(List<Plesac>):** signal;

Веза са СК: СК3, СК4, СК5

Предуслов: /

Постуслов: /

**Уговор У4: PronadjiPlesaca**

Операција: **pronadjiPlesaca(String parametar)**: signal;

Веза са СК: СК3,СK4, СК5

Предуслов: /

Постуслов: /

**Уговор У5: IzmijeniPlesaca**

Операција: **izmijeniPlesaca(Plesac plesac)**: signal;

Веза са СК: СК4

Предуслов: *Вриједносна и структурна ограничења над објектом Плесач морају бити задовољена.*

Постуслов: *Подаци о плесачу су измењени.*

**Уговор У6: ObrisiPlesaca**

Операција: **obrisiPlesaca(Plesac plesac):** signal;

Веза са СК: СК5

Предуслов: *Вриједносна и структурна ограничења над објектом Плесач морају бити задовољена.*

Постуслов: *Плесач је обрисан.*

**Уговор У7: GetAllTreneri**

Операција: **getAllTreneri(List<Trener>)**: signal;

Веза са СК: СК7, СК8, СК9, СК10

Предуслов: /

Постуслов: /

**Уговор У8: DodajTrenera**

Операција: **dodajTrenera(Trener trener)**: signal;

Веза са СК: СК6

Предуслов: *Вриједносна и структурна ограничења над објектом Тренер морају бити задовољена.*

Постуслов: *Подаци о тренеру су запамћени.*

**Уговор У9: PronadjiTrenera**

Операција: **pronadjiTrenera(String parametar)**: signal;

Веза са СК: СК7, СК8, СК9

Предуслов: */*

Постуслов: */*

**Уговор У10: IzmijeniTrenera**

Операција: **izmijeniTrenera(Trener trener):** signal;

Веза са СК: СК8

Предуслов: *Вриједносна и структурна ограничења над објектом Тренер морају бити задовољена.*

Постуслов: *Подаци о тренеру су измијењени.*

**Уговор У11: ObrisiTrenera**

Операција: **obrisiTrenera(Trener trener)**: сигнал;

Веза са СК: СК9

Предуслов: *Вриједносна и структурна ограничења над објектом Tренер морају бити задовољена.*

Постуслов: *Тренер је обрисан.*

**Уговор У12: GetAllTipPlesa**

Операција: **getAllTrening(List<TipPlesa>)**: signal;

Веза са СК: СК10

Предуслов: /

Постуслов: /

**Уговор У13: DodajDnevniRaspored**

Операција: **dodajDnevniRaspored(DnevniRaspored dnevniRaspored)**: signal;

Веза са СК: СК10

Предуслов: *Вриједносна и структурна ограничења над објектом ДневниРаспоред морају бити задовољена.*

Постуслов: *Подаци о дневном радпореду су запамћени.*

**Уговор У14: GetAllDnevniRaspored**

Операција: **getAllDnevniRaspored(List<DnevniRaspored>)**: signal;

Веза са СК: СК11

Предуслов: /

Постуслов: /

**Уговор У15: PronadjiDnevniRaspored**

Операција: **pronadjiDnevniRaspored(String parametar)**: signal;

Веза са СК: С11

Предуслов: */*

Постуслов: */*

**Уговор У16: ObrisiDnevniRaspored**

Операција: **obrisiDnevniRaspored(DnevniRaspored dnevniRaspored)**: сигнал;

Веза са СК: СК11

Предуслов: *Вриједносна и структурна ограничења над објектом Дневни распоред морају бити задовољена.*

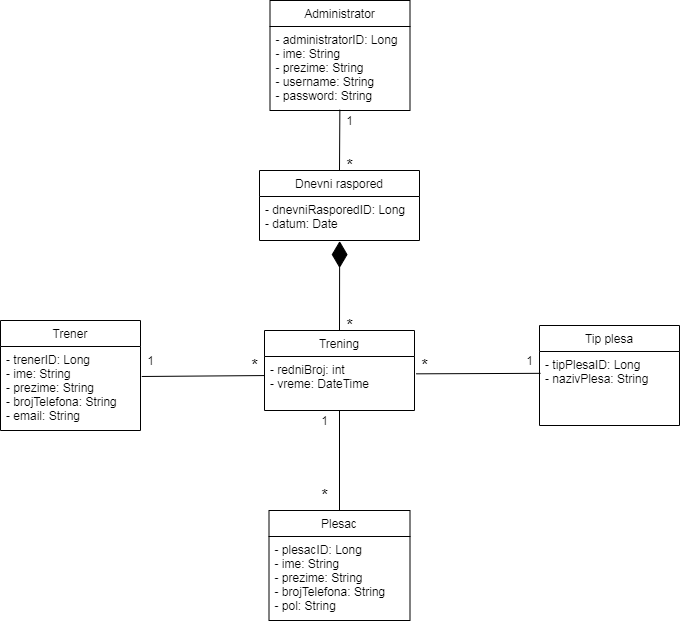
Постуслов: *Дневни распоред је обрисан.*

## 2.3 Структура софтверског система - Концептуални (доменски) модел

Структура софтверског система се описује уз помоћу *концептуалног модела*.

Концептуални модел садржи концептуалне класе(доменске објекте) и асоцијације између концептуалних класа. Често се још и називају *доменским моделима* или *моделима објектне анализе.*

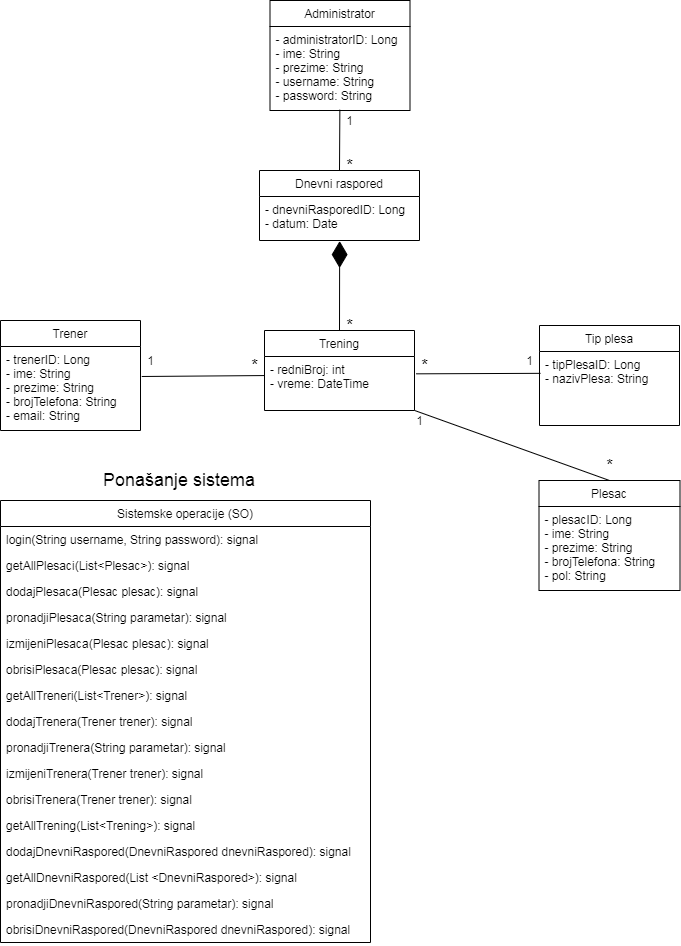
Резултат фазе анализе сценарија случајева коришћења и прављења концептуалног модела је логичка структура и понашање софтверског система који је приказан је на слици испод.



*Слика 29: Дијаграм класа*

**Софтверски систем**

Структура система



*Слика 30: Софтверски систем-Структура и понашање*

## 2.4 Структура софтверског система - Релациони модел

На основу концептуалног модела прави се *релациони модел*, који нам служи као основа за пројектовање релационе базе података.

Идентификовали смо следеће класе у концептуалном моделу: **Плесач, Администратор, Тренер, Тренинг, ДневниРаспоред, ТипПлеса**. Њих ћемо потом представити као табеле у релацоном моделу.

Administrator(AdministratorID, Ime, Prezime, Username, Password)

Plesac(PlesacID, Ime, Prezime, BrojTelefona, Pol)

Trener(TrenerID, Ime, Prezime, BrojTelefona, Email)

TipPlesa(TipPlesaID, NazivPlesa)

DnevniRaspored(DnevniRasporedID, Datum)

Trening(DnevniRasporedID, RedniBroj, Vreme, *TrenerID, TipPlesaID*)

**Табеле ограничења**

За сваку од идентификованих табела треба дефинисати структурна и вредносна ограничења, што наводимо у следећим табелама ограничења.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Administrator** | | **Prosto vrijednosno ograničenje** | | **Složeno vrijednosno ograničenje** | | **Strukturno**  **ograničenje** |
| **Atributi** | **Ime** | **Tip atributa** | **Vrijednost atributa** | **Međuzav.**  **atributa jedne tabele** | **Međuzav. atributa više tabela** | INSERT/UPDATE  CASCADES  DNEVNI RASPORED DELETE RESTRICTED DNEVNI RASPORED |
|  | AdministratorID | Long | Not null and >0 |  |  |
|  | Ime | String | Not null |  |  |
|  | Prezime | String | Not null |  |  |
|  | Username | String | Not null |  |  |
|  | Password | String | Not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Trener** | | **Prosto vrijednosno ograničenje** | | **Složeno vrijednosno ograničenje** | | **Strukturno ograničenje** |
| **Atributi** | **Ime** | **Tip atributa** | **Vrijednost atributa** | **Međuzav. atributa jedne tabele** | **Međuzav. atributa više**  **tabela** | INSERT / UPDATE CASCADES TRENING UPDATE CASCADES TRENING |
| PlesacID | Long | Not null and >0 |  |  |
| Ime | String |  |  |  |
| Prezime | String |  |  |  |
| BrojTelefona | String |  |  |  |
|  | Email | String |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Plesač** | | **Prosto vrijednosno ograničenje** | | **Složeno vrijednosno ograničenje** | | **Strukturno ograničenje** |
| **Atributi** | **Ime** | **Tip atributa** | **Vrijednost atributa** | **Međuzav. atributa jedne tabele** | **Međuzav. atributa više**  **tabela** | INSERT /  UPDATE TRENING |
| PlesačID | Integer | Not null and >0 |  |  |
| Ime | String | Not null |  |  |
| Prezime | String | Not null |  |  |
| Pol | Char | Not null |  |  |
| BrojTelefona | String | Not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela TipPlesa** | | **Prosto vrijednosno ograničenje** | | **Složeno vrijednosno ograničenje** | | **Strukturno**  **ograničenje** |
| **Atributi** | **Ime** | **Tip atributa** | **Vrijednost atributa** | **Međuzav.**  **atributa jedne tabele** | **Međuzav. atributa više tabela** | INSERT /  UPDATE /  DELETE / |
|  | TipPlesaID | Long | Not null and >0 |  |  |
|  | NazivPlesa | String | Not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela DnevniRaspored** | | **Prosto vrijednosno ograničenje** | | **Složeno vrijednosno ograničenje** | | **Strukturno**  **ograničenje** |
| **Atributi** | **Ime** | **Tip atributa** | **Vrijednost atributa** | **Međuzav.**  **atributa jedne tabele** | **Međuzav. atributa više tabela** | UPDATE /  INSERT RESTRECRED TRENING  DELETE TRENING |
|  | DnevniRasporedID | Long | Not null and >0 |  |  |
|  | Datum | Date | Not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela Trening** | | **Prosto vrijednosno ograničenje** | | **Složeno vrijednosno ograničenje** | | **Strukturno**  **ograničenje** |
| **Atributi** | **Ime** | **Tip atributa** | **Vrijednost atributa** | **Međuzav.**  **atributa jedne tabele** | **Međuzav. atributa više tabela** | INSERT TRENER  UPDATE TRENER DELETE / |
|  | RedniBroj | Long | Not null and >0 |  |  |
|  | Vreme | String | Not null |  |  |
|  | DnevniRasporedID | Long | Not null and >0 |  |  |
|  | TrenerID | Long | Not null and >0 |  |  |
|  | TipPlesaID | Long | Not null and >0 |  |  |

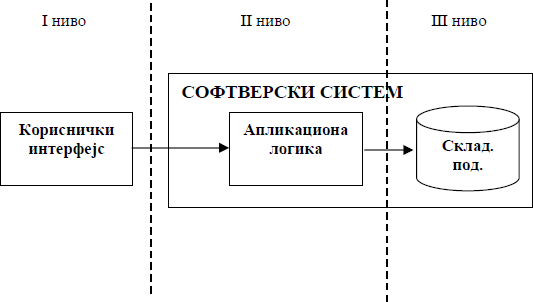
# 3.Фаза пројектовања

## 3.1 Архитектура софтверског система

Архитектура система је тронивојска и састоји се од следећих нивоа:

* кориснички интерфејс
* апликациона логика
* складиште података

Ниво корисничког интерфејса ја на страни клијента, а апликациона логика и складиште се налазе на страни сервера.

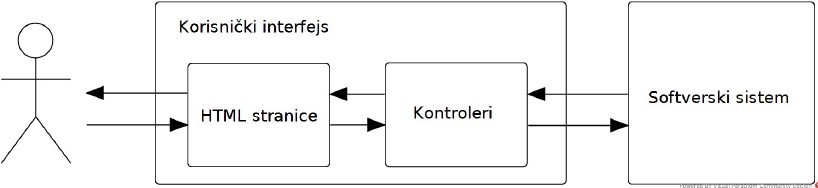


*Слика 31: Архитектура софтверског система*

### 3.1.1 Пројектовање кросничког интерфејса

Кориснички интерфејс представља улазно-излазну реализацију софтверског система. Систем се састоји од

* Екранске форме
* Контролера корисничког интерфејса



*Слика 32: Кориснички интерфејс*

## 3.2 Пројектовање екранске форме

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

### 3.2.1 СК1: Пријављивање администратора

НАЗИВ СК**:**

Пријављивање администратора

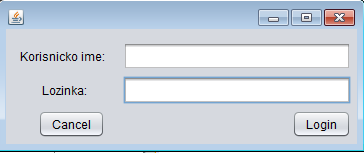
АКТОРИ СК**:**Администратор

УЧЕСНИЦИ СК**:**

Администратор и систем (програм)

ПРЕДУСЛОВИ:

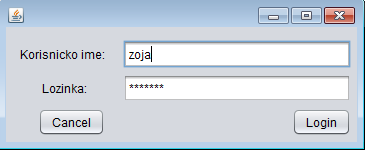
Систем је укључен и приказује форму за пријављивање администратора



*Слика 33: Приказ форме за пријављивање администратора на систем*

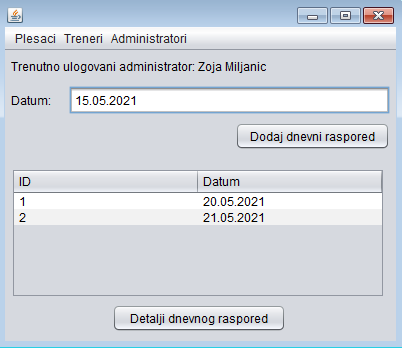
**ОСНОВНИ СЦЕНАРИО СЛУЧАЈА КОРИШЋЕЊА:**

1. Администратор уноси корисничко име и лозинку. (АПУСО)
2. Администратор контролише да ли је правилно унио неопходне податке. (АНСО)



*Слика 34: Пријављивање администратора на систем*

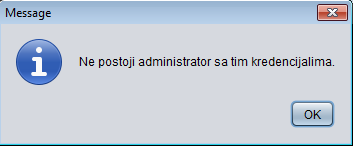
1. Администратор позива систем да га пријави. (АПСО)
2. Систем претражује да ли постоји администратор међу постојећима. (СО)
3. Систем приказује администратору поруку „Успјешно сте се пријавили на систем”. (ИА)



*Слика 35: Главна форма система*

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

5.1 Уколико систем препозна да унесени подаци нису у складу са постојећим, приказује се порука: „Креденцијали нису усправни ”. (ИА)



*Слика 36: Неуспешно пријављивање администратора*

### 3.2.2 СК2: УносПлесача

НАЗИВ СК**:**

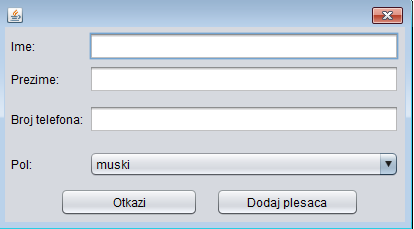
Унос плесача

АКТОРИ СК**:**Администратор

УЧЕСНИЦИ СК**:**

Администратор и систем (програм)

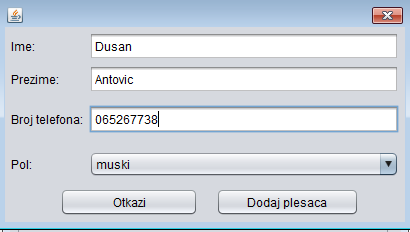
ПРЕДУСЛОВИ**:**Систем је укључен и администратор је улогован под својом шифром. Систем приказује форму за рад са плесачима.



*Слика 37: Форма за унос плесача*

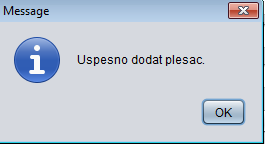
**ОСНОВНИ СЦЕНАРИО СЛУЧАЈА КОРИШЋЕЊА:**

1. Администратор уноси податке о плесачу. (АПУСО)
2. Администратор контролише да ли је правилно унио податке о плесачу. (АНСО)



Слика 38: Форма за унос плесача

1. Администратор позива систем да запамти податке о плесачу. (АПСО)
2. Систем памти податке о плесачу. (СО)
3. Систем приказује администратору запамћеног плесача и поруку „Систем је запамтио плесача“. (ИА)



*Слика 39: Успјешно додај плесач*

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

3.1 Ако систем не може да креира плесача, приказује поруку: „Систем не може да креира плесача“. Прекида се извршење сценарија. (ИА)

5.1 Ако систем не може да запамти податке о плесачу, приказује поруку: “Систем не може да запамти плесача”. (ИА)

### 3.2.3 СК3: Претрага плесача

НАЗИВ СК**:**

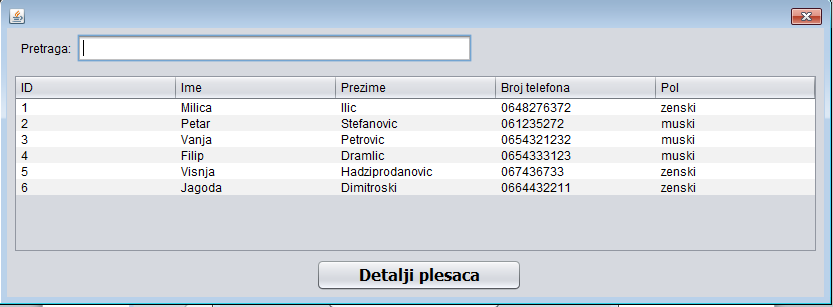
Претрага плесача

АКТОРИ СК**:**Администратор

УЧЕСНИЦИ СК**:**

Администратор и систем (програм)

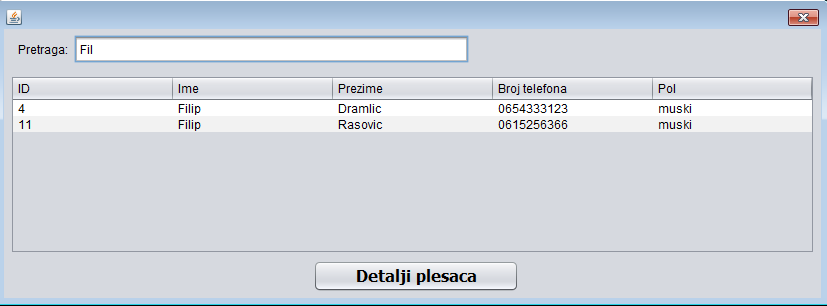
ПРЕДУСЛОВИ**:**Систем је укључен. Администратор је улогован под својом шифром. Систем приказује форму за рад са плесачима.



*Слика 40: Форма за претрагу плесача*

**ОСНОВНИ СЦЕНАРИО СЛУЧАЈА КОРИШЋЕЊА:**

1. Администратор уноси вриједност по којој претражује плесаче. (АПУСО)
2. Администратор позива систем да пронађе плесаче по задатој вриједности. (АПСО)
3. Систем претражује плесаче по задатој вриједности. (СО)
4. Систем приказује администратору поруку „Систем је пронашао клијнте по задатој вриједности”. (ИА)



*Слика 41: Претрага по задатом критеријуму*

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

4.1 Ако систем не може да пронађе плесаче по задатој вриједности, приказује поруку: „Систем не може да пронађе плесаче по задатој вриједности”. Прекида се извршење сценарија. (ИА)

### 3.2.4 СК4: Измјена података о плесачу

НАЗИВ СК**:**

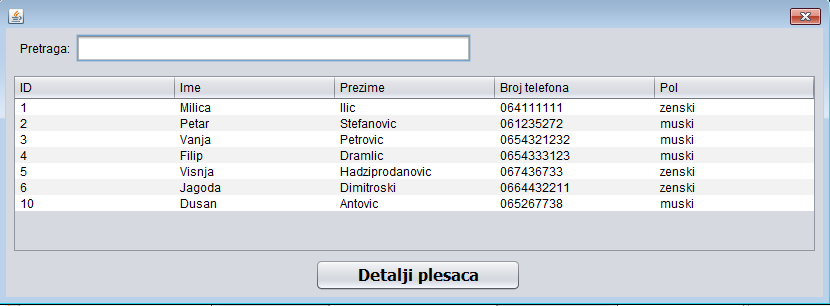
Измјена података о плесачу

АКТОРИ СК**:**Администратор

УЧЕСНИЦИ СК**:**

Администратор и систем (програм)

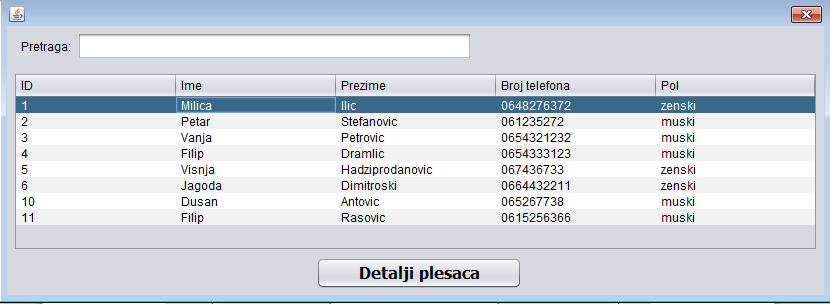
ПРЕДУСЛОВИ**:**Систем је укључен. Администратор је пријављен на систем и систем приказује налог на који је корисник пријављен. Систем приказује форму за рад са плесачима.



*Слика 42: Форма за претрагу плесача*

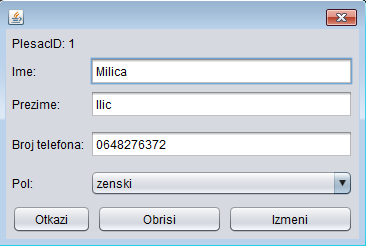
**ОСНОВНИ СЦЕНАРИО СЛУЧАЈА КОРИШЋЕЊА:**

1. Администратор бира плесача којег жели да измјени. (АПУСО)



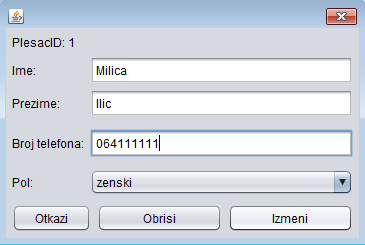
*Слика 43: Форма за претрагу-Изабрани плесач*

1. Администратор позива систем да учита податке о изабраном плесачу. (АПСО)
2. Систем проналази податке о изабраном плесачу. (СО)
3. Систем приказује администратору жељеног плесача и поруку: „Систем је пронашао податке о изабраном плесачу“. (ИА)



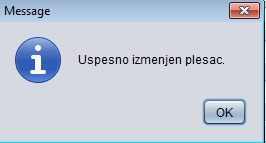
*Слика 44: Форма за измјену плесача*

1. Администратор мијења податке о плесачу. (АПУСО)
2. Администратор контролише да ли је правилно унио податке о плесачу. (АНСО)



*Слика 45: Форма за измјену плесача са измјењеним подацима*

1. Администратор позива систем да запамти податке о плесачу. (АПСО)
2. Систем памти податке о плесачу. (СО)
3. Систем приказује администратору поруку: „Успјешно сте промијенили податке о плесачу”. (ИА)



*Слика 46: Успјешно измјењен плесач*

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

4.1 Aко систем не може да нађе податке о изабраном плесачу приказује поруку: „Систем не може да нађе податке о изабраном плесачу”. Прекида се извршење сценарија. (ИА)

9.1 Ако систем не може да измијени податке, приказује кориснику поруку: „Систем не може да изврши измјену података о плесачу”. (ИА)

### 3.2.5 СК5: Брисање плесача

НАЗИВ СК**:**

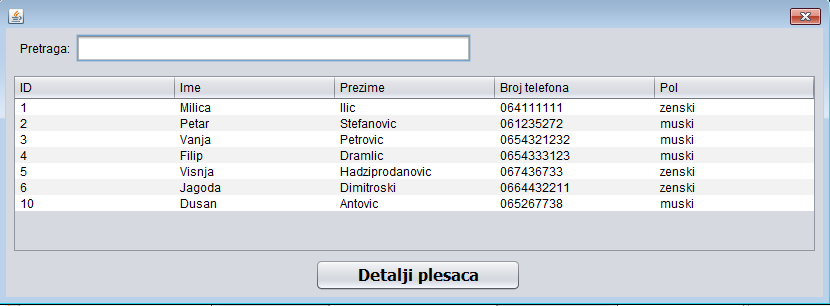
Брисање плесача

АКТОРИ СК**:**Администратор

УЧЕСНИЦИ СК**:**

Администратор и систем (програм)

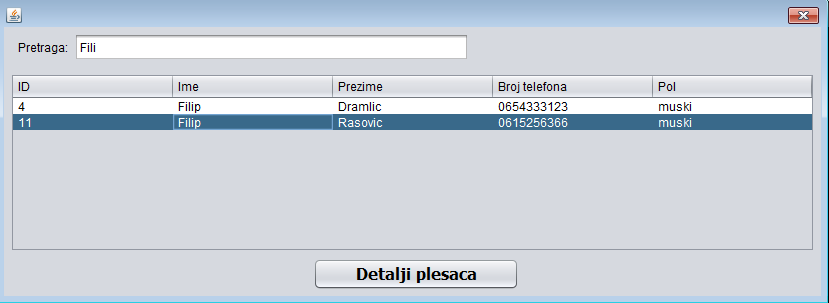
ПРЕДУСЛОВИ**:**Систем је укључен. Администратор је пријављен на систем и систем приказује форму за рад са плесачима. Учитана је листа плесача.



*Слика 47: Форма за претрагу плесача*

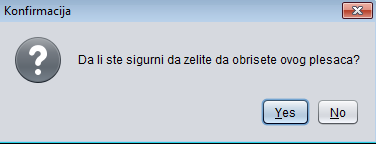
**ОСНОВНИ СЦЕНАРИО СЛУЧАЈА КОРИШЋЕЊА:**

1. Администратор уноси вриједност по којој претражује плесаче. (АПУСО)
2. Администратор позива систем да нађе плесача по задатим критеријумима. (АПСО)
3. Систем тражи плесаче по задатим критеријумима. (СО)
4. Систем приказује администратору податке о пронађеним плесачима и поруку: „Систем је пронашао плесаче по задатом критеријуму”.(ИА)

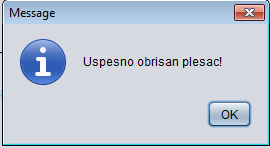


*Слика 48: Форма за претрагу плесача-Изабран плесач*

1. Администратор позива систем да обрише изабраног плесача. (АПСО)
2. Систем брише плесача. (СО)
3. Систем приказује администратору поруку: „Успјешно сте обрисали изабраног плесача”. (ИА)



*Слика 49: Конфирмација брисања изабраног плесача*



*Слика 50: Успјешно обрисан плесач*

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

4.1 Ако систем не може да нађе плесаче са задатим вриједностима, приказује се порука: „Систем не може да пронађе плесаче по задатом критеријуму”.(ИА)

7.1 Ако систем не може да нађе податке о изабраном плесачу приказује се порукa: „Систем не може да пронађе податке о плесачу“. Прекида се извршење сценарија. (ИА)

### 3.2.6 СК6: Унос тренера

НАЗИВ СК**:**

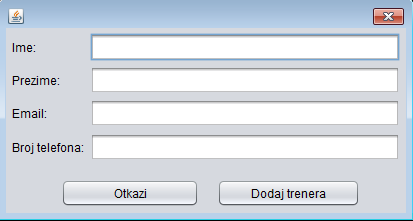
Унос тренера

АКТОРИ СК**:**Администратор

УЧЕСНИЦИ СК**:**

Администратор и систем (програм)

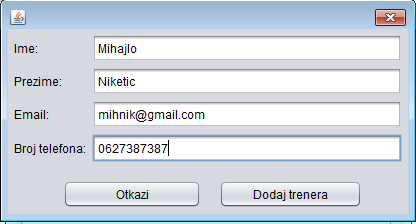
ПРЕДУСЛОВИ**:**Систем је укључен. Администратор је пријављен на систем и систем приказује налог на који је корисник пријављен. Систем приказује форму за унос новог тренера.



*Слика 51: Форма за унос тренера*

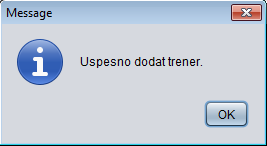
**ОСНОВНИ СЦЕНАРИО СЛУЧАЈА КОРИШЋЕЊА:**

1. Администратор уноси податке о новом тренеру. (АПУСО)



*Слика 52: Попуњена форма за унос тренера*

1. Администратор контролише да ли је правилно унио податке о тренеру. (АНСО)
2. Администратор позива систем да креира новог тренера. (АПСО)
3. Систем креира новог тренера. (СО)
4. Систем приказује администратору новог тренера и поруку „Систем је креирао тренера“. (ИА)



*Слика 53: Успјешно унесен тренер*

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

5.1 Ако систем не може да унесе новог тренера, приказује поруку: „Систем није успјешно унио новог тренера”. (ИА)

### 3.2.7 СК7: Измјена тренера

НАЗИВ СК**:**

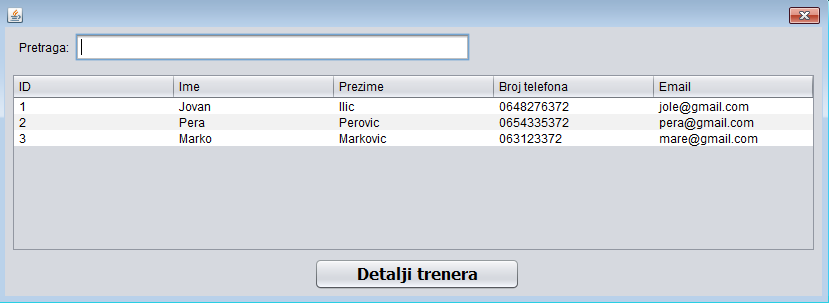
Измјена тренера

АКТОРИ СК**:**Администратор

УЧЕСНИЦИ СК**:**

Администратор и систем (програм)

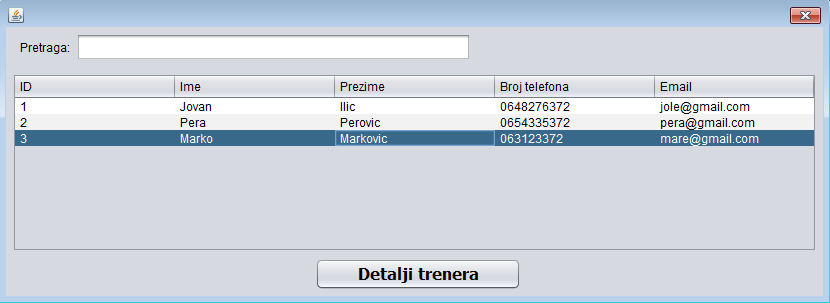
ПРЕДУСЛОВИ**:**Систем је укључен. Администратор је пријављен на систем и систем приказује налог на који је корисник пријављен. Учитана је листа тренера.



*Слика 54: Форма за претрагу тренера*

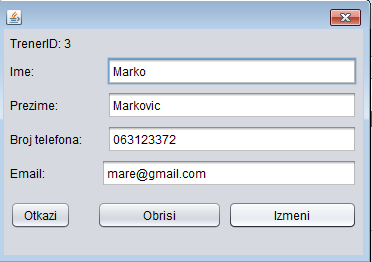
**ОСНОВНИ СЦЕНАРИО СЛУЧАЈА КОРИШЋЕЊА:**

1. Администратор бира тренера којег жели да измјени. (АПУСО)



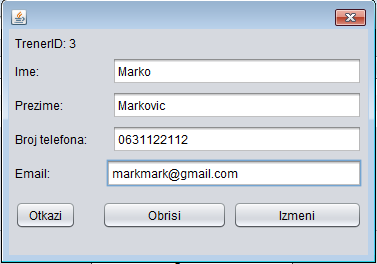
*Слика 55: Изабран тренер*

1. Администратор позива систем да учита податке о изабраном тренеру. (АПСО)
2. Систем проналази податке о изабраном тренеру. (СО)
3. Систем приказује администратору податке о изабраном тренеру и поруку: „Систем је успјешно пронашао податке о изабраном тренеру“. (ИА)



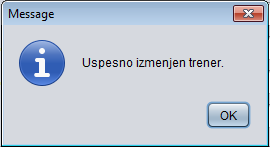
*Слика 56: Форма за измјену тренера*

1. Администратор уноси измијењене податке о тренеру.(АПУСО)
2. Администратор контролише да ли је коректно унио податке о тренеру. (АНСО)



*Слика 57: Форма попуњена измјењеним подацима о тренеру*

1. Администратор позива систем да запамти податке о тренеру. (АПСО)
2. Систем памти податке. (СО)
3. Систем приказује администратору поруку: „Систем је успјешно измијенио податке о тренеру”. (ИА)



*Слика 58: Измјењен тренер*

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

4.1 Ако систем не може да нађе податке о задатом тренингу, приказује се порука: „Систем није пронашао податке о тренеру“. (ИА)

9.1 Ако систем не може да измијени податке, приказује поруку: „Систем није измијенио податке о тренеру”. (ИА)

### 3.2.8 СК8: Брисање тренера

НАЗИВ СК**:**

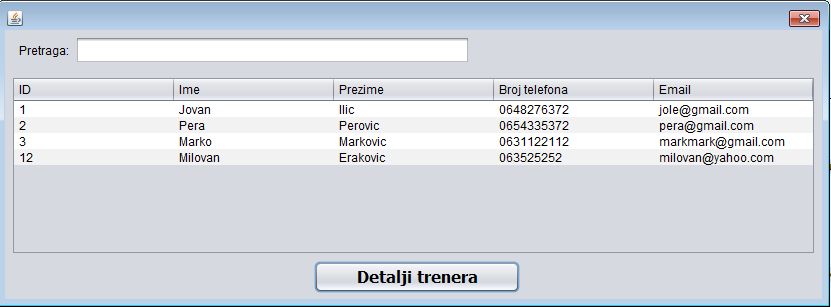
Брисање тренера

АКТОРИ СК**:**Администратор

УЧЕСНИЦИ СК**:**

Администратор и систем (програм)

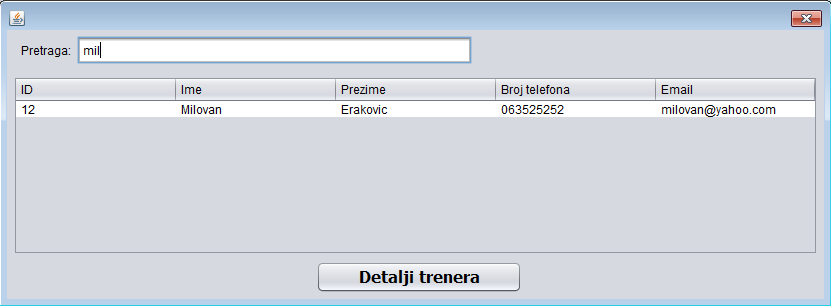
ПРЕДУСЛОВИ**:**Систем је укључен. Администратор је пријављен на систем и учитана је листа тренера.



*Слика 59: Форма за претрагу тренера*

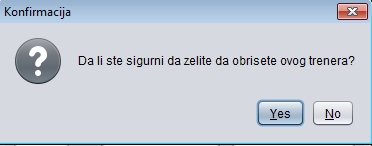
**ОСНОВНИ СЦЕНАРИО СЛУЧАЈА КОРИШЋЕЊА:**

1. Администратор уноси вриједност по којој претражује тренере. (АПУСО)
2. Администратор позива систем да нађе тренера по задатом критеријуму. (АПСО)
3. Систем тражи тренера са задатим критеријумом. (СО)
4. Систем приказује администратору податке о пронађеним тренерима и поруку „Систем је пронашао тренере по задатом критеријуму”.(ИА)

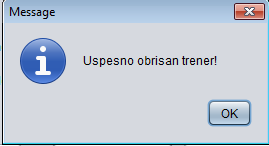


*Слика 60: Претрага по задатом критеријуму за тренере*

1. Администратор позива систем да обрише тренера. (АПСО)
2. Систем брише изабраног тренера. (СО)
3. Систем приказује администратору поруку: „Систем је успјешно обрисао тренера”. (ИА)



*Слика 61: Конфирмација брисања тренера*



*Слика 62: Успјешно обрисан тренер*

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

4.1 Ако систем не може да нађе тренера који се поклапа са критеријумима, систем приказује поруку: „Систем није пронашао тренере по задатом критеријуму”. (ИА)

7.1 Ако систем не може да обрише изабраног тренера, приказује поруку: „Систем није успјешно обрисао тренера”. (ИА)

### 3.2.9 СК9: Претрага тренера

НАЗИВ СК**:**

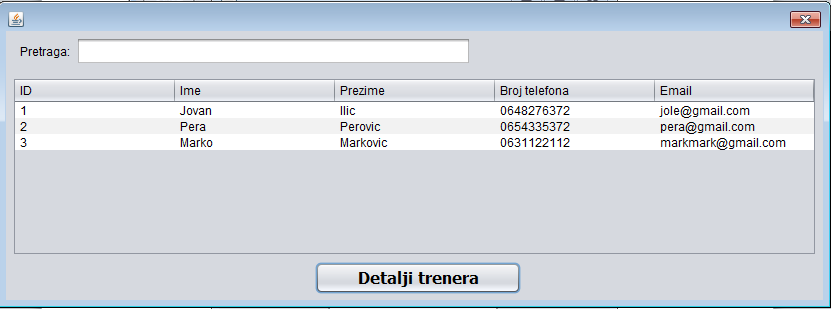
Претрага тренера

АКТОРИ СК**:**Администратор

УЧЕСНИЦИ СК**:**

Администратор и систем (програм)

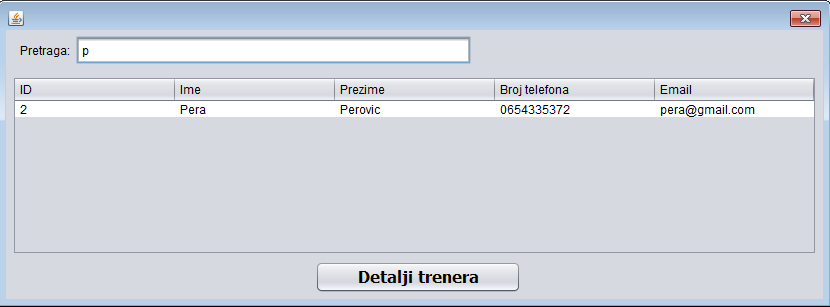
ПРЕДУСЛОВИ**:**Систем је укључен. Администратор је пријављен на систем и систем приказује форму за претрагу тренинга. Учитана је листа тренера.



*Слика 63: Претрага тренера*

**ОСНОВНИ СЦЕНАРИО СЛУЧАЈА КОРИШЋЕЊА:**

1. Администратор уноси вриједност по којој жели да претражи тренере. (АПУСО)
2. Администратор позива систем да пронађе тренере по задатом критеријуму. (АПСО)
3. Систем претражује тренере. (СО)
4. Систем приказује администратору податке о пронађеним тренерима и поруку: поруку „Систем је пронашао тренере по задатом критеријуму”.(ИА)



*Слика 64: Претрага тренера са задатим критеријумом*

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

4.1 Ако систем не може да пронађе ниједног тренера који задовољава задате критеријуме, приказује поруку: „Систем није пронашао тренере по задатом критеријуму”. (ИА)

### 3.2.10 СК10: Додај дневни распоред

НАЗИВ СК**:**

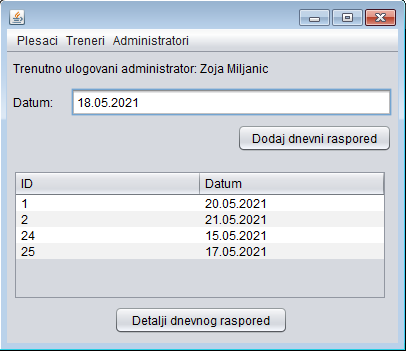
Додај дневни распоред

АКТОРИ СК**:**Администратор

УЧЕСНИЦИ СК**:**

Администратор и систем (програм)

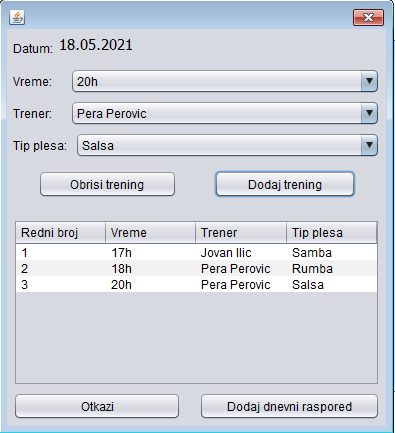
ПРЕДУСЛОВИ**:**Систем је укључен. Администратор је пријављен на систем и систем приказује форму за евиденцију дневног распореда. Учитана је листа плесача.



*Слика 65: Форма за рад са дневним распоредом*

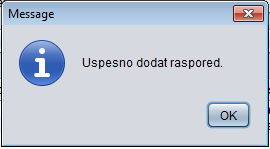
**ОСНОВНИ СЦЕНАРИО СЛУЧАЈА КОРИШЋЕЊА:**

1. Администратор уноси податке о дневном распореду. (АПУСО)
2. Администратор контролише да ли је унио коректно све податке. (АНСО)



*Слика 66: Форма за додавање новог распореда*

1. Администратор позива систем да запамти податке о дневном распореду. (АПСО)
2. Систем памти податке о дневом распореду. (СО)
3. Систем приказује администратору запамћен дневни распоред и поруку: “Систем је успјешно евидентирао дневни распоред”. (ИА)



*Слика 67: Успјешно додај дневни распоред*

**АЛТЕРНАТИВНА СЦЕНАРИЈА:**

5.1 Ако систем не може да запамти дневни распоред, приказује поруку: „Систем не може успјешно да евидентира дневни распоред“. (ИА)

## 3.3 Пројектовање контролера корисничког интерфејса

Kонтролер корисничког интерфејса је одговоран да:

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

## 3.4 Пројектовање апликационе логике

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

1. Контролера апликационе логике – служи за комуникацију са клијентом и одговоран је да прихвати захтев за извршење системске операције од клијента и прослиједи 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

### 3.4.1 Контролер апликационе логике

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 ће, кад се то деси, 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.

## 3.5 Пословна логика

### 3.5.1 Пројектовање понашања (системске операције)

За сваки од претходно дефинисаних уговора правимо системску операцију, што заправо представља пројектовање понашања. Класа *AbstractSO*, 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*, оне су апстрактне и њих ће свака класа системске операције имплементирати.

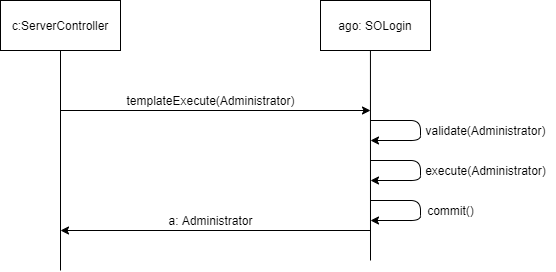
Након тога се позива метода *commit* која узима објекат *Connection*, класе *Кonekcija* и позива њену методу *commit*. Провера предуслова се извршава на клијентској страни уколико постоји, а постуслови се очитавају у оквиру *Response* објекта који шаље сервер клијенту и на основу кога клијент закључује да ли је операција успешно извршена на серверској страни или је дошло до грешке.

#### 3.5.1.1 Операција: Login(String username, String password): signal;

Веза са СК: СК1

Предуслов: */*

Постуслов: */*



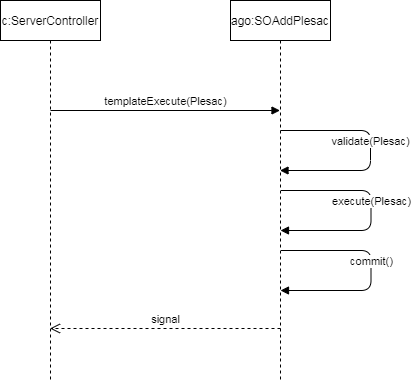
*Слика 68: Дијаграм секвенци за уговор login*

#### 3.5.1.2 Операција: dodajPlesaca(Plesac plesac): signal;

Веза са СК: СК2

Предуслов: *Вриједносна и структурна ограничења над објектом Плесач морају бити задовољена.*

Постуслов: *Подаци о плесачу су запамћени.*



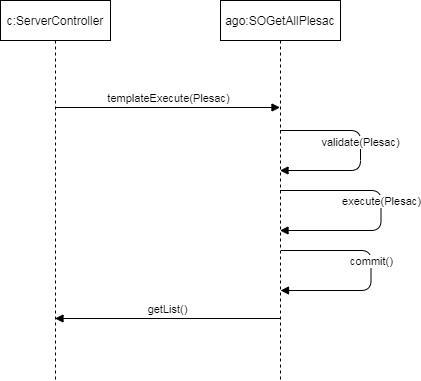
*Слика 69: Дијаграм секвенци за уговор dodajPlesaca*

#### 3.5.1.3 Операција: getAllPlesaci(List<Plesac>): signal;

Веза са СК: СК3, СК4, СК5

Предуслов: /

Постуслов: /



*Слика 70: Дијаграм секвенци за уговор getAllPlesaci*

#### 3.5.1.4 Операција: pronadjiPlesaca(String parametar): signal;

Веза са СК: СК3,СK4, СК5

Предуслов: /

Постуслов: /

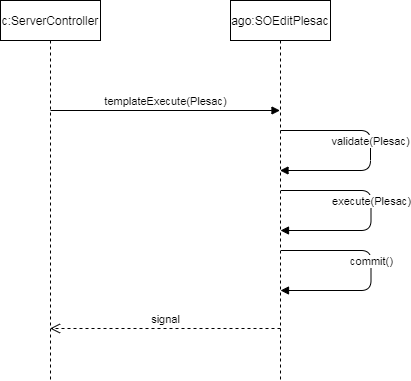
Ова операција позива операцију која враћа све клијенте и затим на клијентској страни врши филтрирање по критеријуму.

#### 3.5.1.5 Операција: IzmijeniPlesaca(Plesac plesac): signal;

Веза са СК: СК4

Предуслов: *Вриједносна и структурна ограничења над објектом Плесач морају бити задовољена.*

Постуслов: *Подаци о плесачу су измењени.*

**

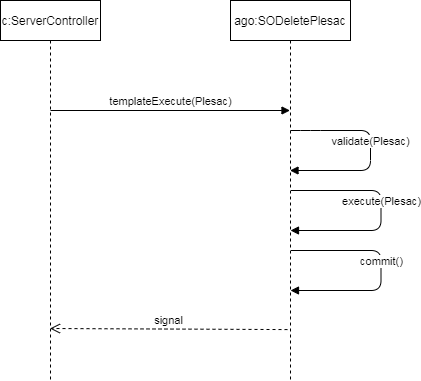
*Слика 71: Дијаграм секвенци за уговор izmijeniPlesaca*

#### 3.5.1.6 Операција: obrisiPlesaca(Plesac plesac): signal;

Веза са СК: СК5

Предуслов: *Вриједносна и структурна ограничења над објектом Плесач морају бити задовољена.*

Постуслов: *Плесач је обрисан.*



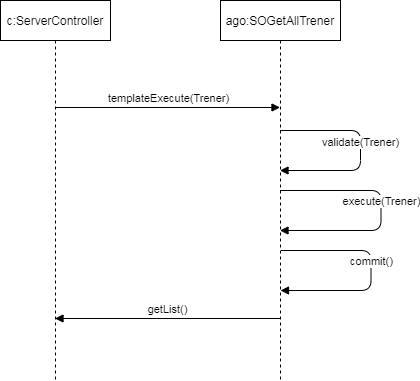
*Слика 72: Дијаграм секвенци за уговор obrisiPlesaca*

#### 3.5.1.7 Операција: getAllTreneri(List<Trener>): signal;

Веза са СК: СК7, СК8, СК9

Предуслов: /

Постуслов: /



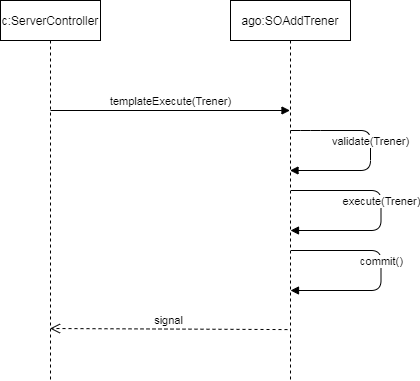
*Слика 73: Дијаграм секвенци за уговор getAllTreneri*

#### 3.5.1.8 Операција: dodajTrenera(Trener trener): signal;

Веза са СК: СК6

Предуслов: *Вриједносна и структурна ограничења над објектом Тренер морају бити задовољена.*

Постуслов: *Подаци о тренеру су запамћени.*

**

*Слика 74: Дијаграм секвенци за уговор dodajTrenera*

#### 3.5.1.9 Операција: pronadjiTrenera(String parametar): signal;

Веза са СК: СК7, СК8, СК9

Предуслов: */*

Постуслов: */*

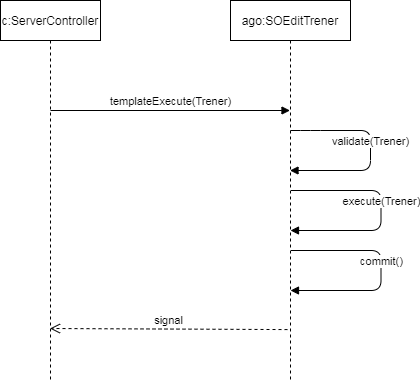
Ова операција позива операцију која враћа све клијенте и затим на клијентској страни врши филтрирање по критеријуму.

#### 3.5.1.10 Операција: izmijeniTrenera(Trener trener): signal;

Веза са СК: СК8

Предуслов: *Вриједносна и структурна ограничења над објектом Тренер морају бити задовољена.*

Постуслов: *Подаци о тренеру су измијењени.*

**

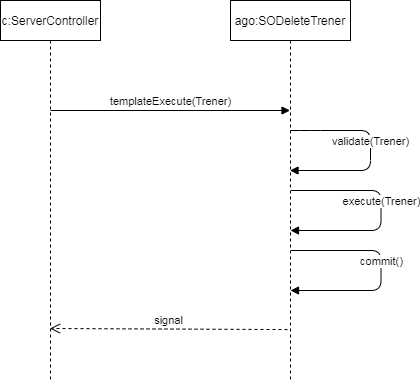
*Слика 75: Дијаграм секвенци за уговор izmijeniTrenera*

#### 3.5.1.11 Операција: obrisiTrenera(Trener trener): signal;

Веза са СК: СК9

Предуслов: *Вриједносна и структурна ограничења над објектом Tренер морају бити задовољена.*

Постуслов: *Тренер је обрисан.*

**

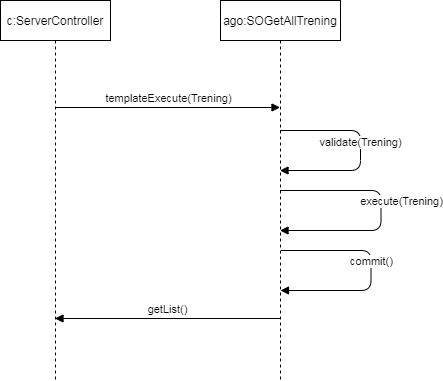
*Слика 76: Дијаграм секвенци за уговор obrisiTrenera*

#### 3.5.1.12 Операција: getAllTrening(List<Trening>): signal;

Веза са СК: СК10

Предуслов: /

Постуслов: /



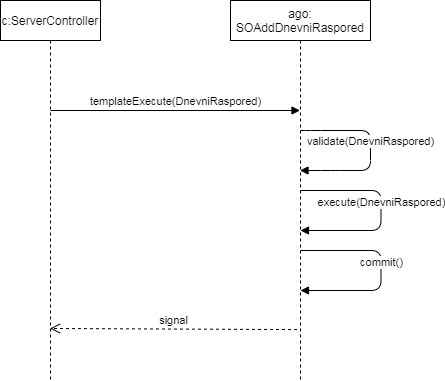
*Слика 77: Дијаграм секвенци за уговор getAllTreneri*

#### 3.5.1.13 Операција: dodajDnevniRaspored(DnevniRaspored dnevniRaspored): signal;

Веза са СК: СК10

Предуслов: *Вриједносна и структурна ограничења над објектом ДневниРаспоред морају бити задовољена.*

Постуслов: *Подаци о дневном радпореду су запамћени.*



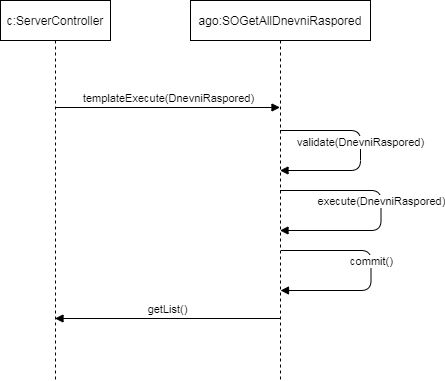
*Слика 78: Дијаграм секвенци за уговор dodajDnevniRaspored*

#### 3.5.1.14 Операција: getAllDnevniRaspored (List<DnevniRaspored>): signal;

Веза са СК: СК11

Предуслов: /

Постуслов: /



*Слика 79: Дијаграм секвенци за уговор getAllDnevniRaspored*

#### 3.5.1.15 Операција: pronadjiDnevniRaspored(String parametar): signal;

Веза са СК: СК11

Предуслов: */*

Постуслов: */*

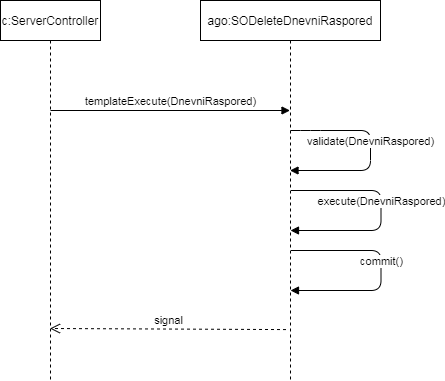
Ова операција позива операцију која враћа све клијенте и затим на клијентској страни врши филтрирање по критеријуму.

#### 3.5.1.16 Операција: obrisiDnevniRaspored(DnevniRaspored dnevniRaspored): signal;

Веза са СК: СК11

Предуслов: *Вриједносна и структурна ограничења над објектом Дневни распоред морају бити задовољена.*

Постуслов: *Дневни распоред је обрисан.*



*Слика 80: Дијаграм секвенци за уговор deleteDnevniRaspored*

## 3.6 Пројектовање структуре софтверског система (доменске класе)

На основу концептуалних класа креирају се софтверске класе структуре. Идентификоване су следедће класе:

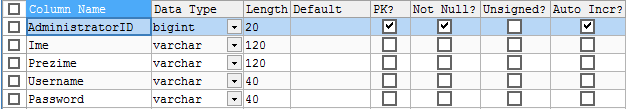
* Администратор
* Плесач
* Тренер
* Тренинг
* Тип плеса
* Дневни распоред

Поред њих додате су и следеће класе:

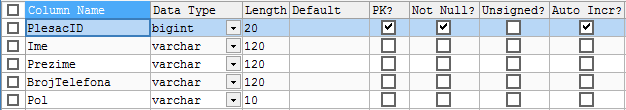
* AbstractDomainObject - апстрактна класа које све доменске класе наслеђују.
* Operation – интерфејс који садржи све операције које се шаљу од клијента серверу.
* Request – служи за слање објекта од клијента ка серверу. Садржи један *Object* атрибут који представља објекат над којим треба извршити захтјевану операцију и један *int* атрибут који представља операцију која треба да се изврши.
* 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 догодио.

### 3.6.1 Пројектовање складишта података

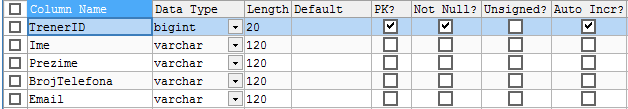
На основу релационог модела и ограничења пројектоване су табеле базе података које користи наш софтверски систем:

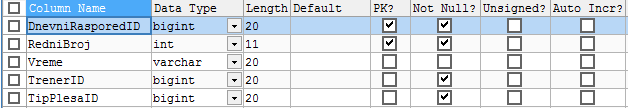






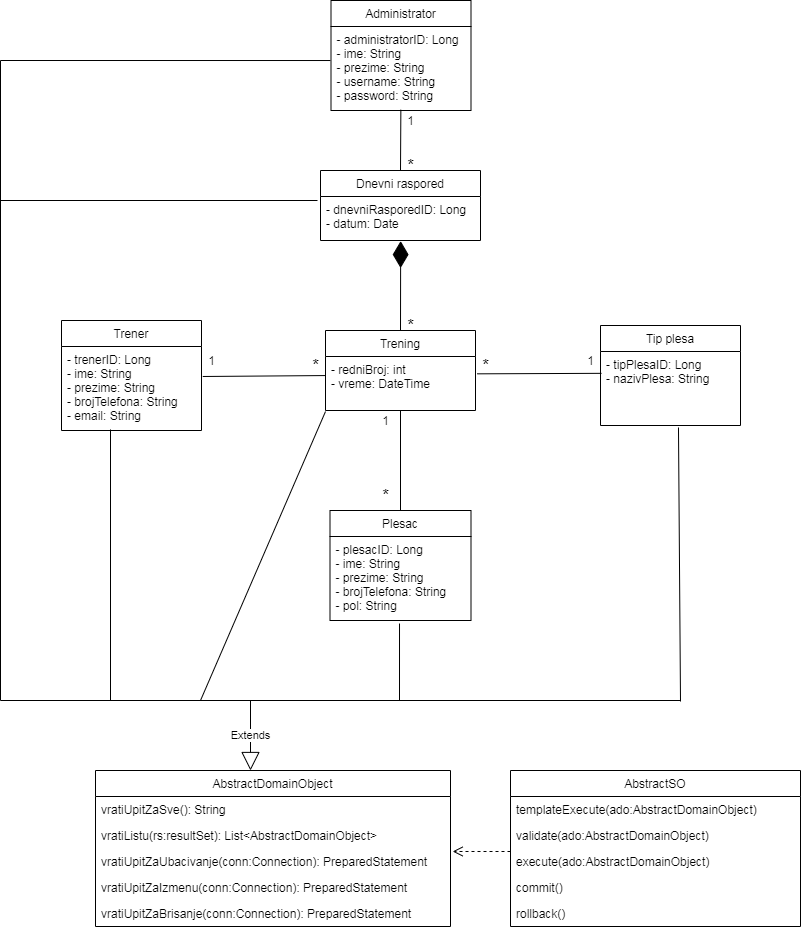




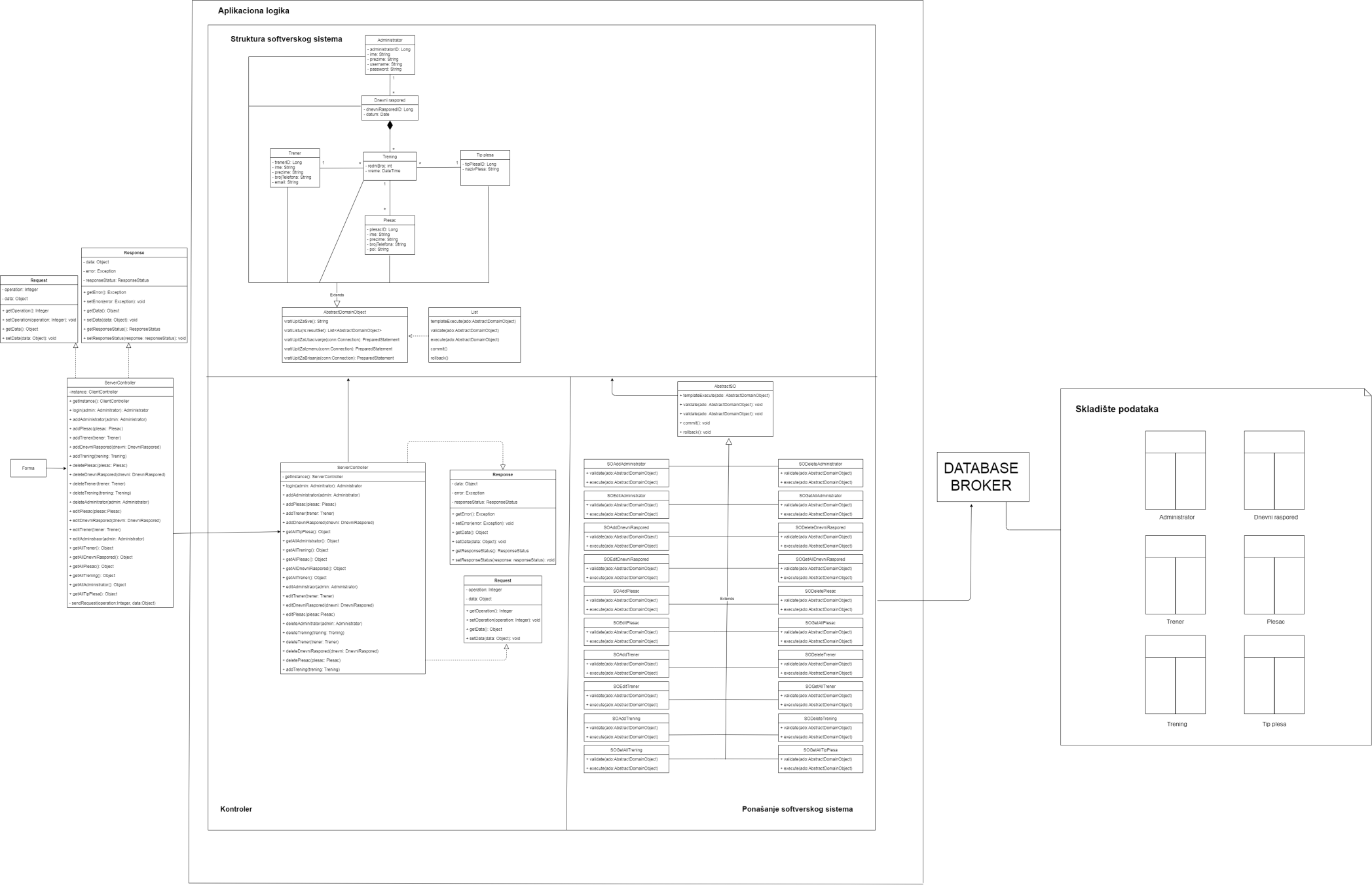


За комуникацију са базом података правимо генеричку класу *AbstractSO* која има методе за валидацију и извршавања трансакције, које су апстрактне и које ће имплементирати свака класа системске операције која ће се извршавати, као и методе за потврђивање и поништавање трансакције. Она се служи класом *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 81: Дијаграм класа добијен након ппројектовања доменских класа и AbstractSO класе*



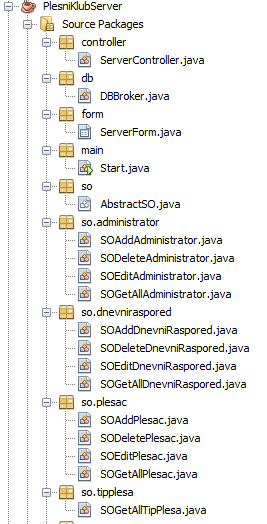
*Слика 82: Коначна архитектура софтверског система*

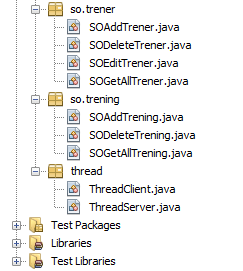
# 4. Имплементација

Имплементација софтверско система „Плесни клуб“ је рађена у Јава програмском језику. Организација је приказана следећом сликом:



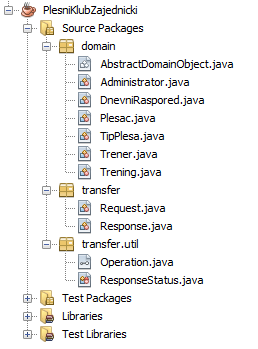
*Слика 83: Организација пројекта*





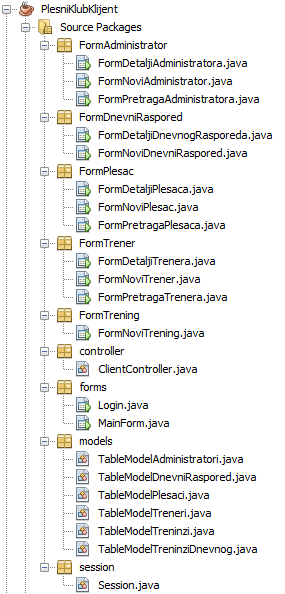
*Слика 84: Пројекат Плесни клуб сервер*

Пројекат *PlesniKlubServer* 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 85: Пројекат Плесни клуб заједнички*

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





*Слика 86: Пројекат Плесни клуб клијент*

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

# 5.Закључак

У овом раду смо видели општи начин за развој софтверског система, у нашем случају софтверског система плесног клуба. Иако је овај систем применљив, постоји још много простора за дораду и преправку како би он у потпуности задовољио потребе његових корисника.

# 6. Литература

”Projektovanje softvera - skripta”, Siniša Vlajić, 2015 (преузето са: <https://www.researchgate.net/publication/303858135_Projektovanje_softvera_-_skripta?enrichId=rgreq-43bc76d9065bff75557b5ab9b01ee2d2-XXX&enrichSource=Y292ZXJQYWdlOzMwMzg1ODEzNTtBUzozNzA4NDg4NzI1ODMxNjhAMTQ2NTQyODY2OTAyNQ%3D%3D&el=1_x_2&_esc=publicationCoverPdf>