

Универзитет у Београду  
Факултет организационих наука  
Лабараторија за софтверско инжињерство



Семинарски рад из предмета  
Пројектовање софтвера

Тема:

***Софтверски систем за праћење тренинга у  
Јава окружењу***

Ментор: Татјана Стојановић

Студент: Филип Трифуновић 2020/0437

Београд, Јул 2024.

## Садржај:

1.	Фаза прикупљања корисничких захтева .....	4
1.1.	Вербални опис .....	4
1.2.	Случајеви коришћења .....	5
	Дијаграм случајева коришћења .....	5
	СК1: Случај коришћења – Креирање корисничког налога.....	6
	СК2: Случај коришћења – Пријава на систем .....	7
	СК3: Случај коришћења – Креирање нове справе.....	8
	СК4: Случај коришћења – Креирање новог чланства .....	9
	СК5: Случај коришћења – Креирање новог тренинга (сложен случај) .....	10
	СК6: Случај коришћења – Промена тренинга (Сложен СК).....	11
	СК7: Случај коришћења – Брисање тренинга .....	12
	СК8: Случај коришћења – Претраживање тренинга .....	13
2.	Фаза анализе.....	14
2.1.	Системски дијаграм секвенци .....	14
2.1.1.	ДС1: Дијаграм секвенци СК – Креирање корисничког налога .....	14
2.1.2.	ДС2: Дијаграм секвенци СК – Пријава на систем.....	15
2.1.3.	ДС3: Дијаграм секвенци СК – Креирање нове справе.....	16
2.1.4.	ДС4: Дијаграм секвенци СК – Креирање чланства .....	17
2.1.5.	ДС5: Дијаграм секвенци СК – Креирање тренинга (сложен случај) .....	18
2.1.6.	ДС6: Дијаграм секвенци СК – Измена тренинга (сложен случај) .....	19
2.1.7.	ДС7: Дијаграм секвенци СК – Брисање тренинга .....	21
2.1.8.	ДС8: Дијаграм секвенци СК – Претраживање тренинга .....	23
2.2.	Преглед системских операција.....	24
2.3.	Понашање софтверског система – Дефинисање уговора о системским операцијама.....	25
2.4.	Структура софтверског система – Концептуални (доменски) модел.....	27
2.5.	Структура софтверског система – релациони .....	28
3.	Фаза пројектовања .....	32
3.1.	Пројектовање корисничког интерфејса .....	32
3.1.1.	Пројектовање екранских форми .....	33
	СК1: Случај коришћења – Креирање корисничког налога.....	35
	СК2: Случај коришћења – Пријава на систем .....	37
	СК3: Случај коришћења – Креирање нове справе.....	38
	СК4: Случај коришћења – Креирање чланства.....	40
	СК5: Случај коришћења – Креирање тренинга (сложен случај) .....	42
	СК6: Случај коришћења – Измена тренинга (сложен случај) .....	44
	СК7: Случај коришћења – Брисање тренинга .....	47

СК8: Случај коришћења – Претраживање тренинга .....	49
3.1.2. Пројектовање контролера корисничког интерфејса .....	52
3.2. <i>Пројектовање апликационе логике</i> .....	52
3.2.1. Контролер апликационе логике .....	52
3.2.2. Пословна логика .....	53
3.2.3. Пројектовање брокера базе података .....	61
3.3. <i>Пројектовање складишта података</i> .....	64
4. Фаза имплементација.....	66
5. Фаза тестирања .....	68
6. Литература.....	68

# 1. Фаза прикупљања корисничких захтева

## 1.1. Вербални опис

Циљ овог софтверског система био је олакшати корисницима теретана и фитнес центара праћење свог напретка, кроз сваки тренинг.

То је софтверски систем који омогућава корисницима да сами уносе податке и прате своје тренинге. Дизајниран је са циљем да омогући појединцима да на лак и ефикасан начин управљају својим фитнес рутинама и здравственим циљевима. Овај софтвер представља личног асистента за фитнес, који корисницима пружа средство за постизање жељене форме и бољег здравља.

Корисници имају могућност креирања и прилагођавања својих тренинг програма у складу са својим потребама и циљевима. Могу лако уносити информације о својој активности, вежбама које изводе и времену које посвећују тренингу. Прате свој напредак, визуализују своје резултате и добијају подстицаје за континуирано напредовање.

Уз технолошки напредак и могућност праћења тренинга у реалном времену, корисници се осећају подржаним и мотивисаним за постигнуће својих фитнес циљева. Ово обезбеђује бољу контролу над влашћу над сопственим здрављем и фитнесом и пружа једноставан и ефикасан начин за постизање бољег општег здравља и физичке спремности.

## 1.2. Случајеви коришћења

1. Креирање корисничког налога
2. Пријава на систем
3. Креирање нове вежбе
4. Претрага вежбе
5. Промена вежбе
6. Брисање вежбе
7. Креирање новог тренинга (Сложен СК)
8. Промена тренинга (Сложен СК)
9. Брисање тренинга

### *Дијаграм случајева коришћења*



Слика 1: Дијаграм случајева коришћења

## СК1: Случај коришћења – Креирање корисничког налога

### Назив СК

Креирање корисничког налога

### Актори СК

Корисник

### Учесници СК

Корисник и систем (програм)

**Предуслов:** Систем је укључен и корисник је улогован под својом шифром. Систем приказује форму за рад са корисничким налогом.

### Основни сценарио СК

1. Корисник уноси податке у кориснички налог. (АПУСО)
2. Корисник контролише да ли је коректно унео податке у кориснички налог. (АНСО)
3. Корисник позива систем да запамти податке о корисничком налогу. (АПСО)
4. Систем памти податке о корисничком налогу. (СО)
5. Систем приказује кориснику запамћени кориснички налог и поруку: “Систем је запамтио кориснички налог”. (ИА)

### Алтернативна сценарија

- 5.1. Уколико систем не може да запамти податке о корисничком налогу он приказује кориснику поруку “Систем не може да запамти кориснички налог”. (ИА)

## СК2: Случај коришћења – Пријава на систем

### Назив СК

Пријава **корисника** на систем

### Актери СК

**Корисник**

### Учесници СК

**Корисник** и **систем**

**Предуслов:** **Систем** је укључен и приказује форму за пријављивање **корисника** на систем. **Корисник** већ поседује налог.

### Основни сценарио СК

1. **Корисник** уноси своје параметре за пријављивање на **систем**. (АПУСО)
2. **Корисник** контролише своје унете податке за пријављивање. (АНСО)
3. **Корисник** позива **систем** да пронађе **корисника** са унетим параметрима. (АПСО)
4. **Систем** претражује **корисника** на основу задатих параметара како би извршио пријављивање. (СО)
5. **Систем** приказује **кориснику** поруку: „Успешно пријављивање“. (ИА)

### Алтернатива сценарија

- 5.1. Ако **систем** не може да пронађе **корисника** са задатим параметрима, **систем** избацује **кориснику** поруку: „Неуспешно пријављивање, налог не постоји“. (ИА)

## СК3: Случај коришћења – Креирање нове справе

### Назив СК

Креирање нове справе

### Актори СК

Корисник

### Учесници СК

Корисник и систем (програм)

**Предуслов:** Систем је укључен и корисник је улоган под својом шифром. Систем приказује форму за рад са новом справом.

### Основни сценарио СК

1. Корисник уноси податке у нову справу. (АПУСО)
2. Корисник контролише да ли је коректно унео податке у нову справу. (АНСО)
3. Корисник позива систем да запамти податке о новој справи. (АПСО)
4. Систем памти податке о новој справи. (СО)
5. Систем приказује кориснику запамћену нову справу и поруку: "Систем је запамтио нову справу". (ИА)

### Алтернативна сценарија

- 5.1. Уколико систем не може да запамти податке о новој справи он приказује кориснику поруку "Систем не може да запамти нова справу". (ИА)



## СК4: Случај коришћења – Креирање новог чланства

### Назив СК

Креирање **новог чланства**

### Актори СК

**Корисник**

### Учесници СК

**Корисник** и **систем** (програм)

**Предуслов:** Систем је укључен и **корисник** је улоган под својом шифром. Систем приказује форму за рад са **новим чланством**.

### Основни сценарио СК

1. **Корисник уноси** податке у **ново чланство**. (АПУСО)
2. **Корисник контролише** да ли је коректно унео податке у **ново чланство**. (АНСО)
3. **Корисник позива систем** да запамти податке о **новом чланству**. (АПСО)
4. **Систем памти** податке о **новом чланству**. (СО)
5. **Систем приказује кориснику** запамћено **ново чланство** и поруку: “**Систем** је запамтио **ново чланство**”. (ИА)

### Алтернативна сценарија

- 5.1. Уколико **систем** не може да запамти податке о **новом чланству** он приказује **кориснику** поруку “**Систем** не може да запамти **ново чланство**”. (ИА)

## СК5: Случај коришћења – Креирање новог тренинга (сложен случај)

### Назив СК

Креирање **новог тренинга**

### Актори СК

**Корисник**

### Учесници СК

**Корисник** и **систем** (програм)

**Предуслов:** Систем је укључен и **корисник** је улогован под својом шифром. Систем приказује форму за рад са **новим тренингом**.

### Основни сценарио СК

1. **Корисник уноси** податке у **нови тренинг**. (АПУСО)
2. **Корисник контролише** да ли је коректно унео податке у **нови тренинг**. (АНСО)
3. **Корисник позива систем** да запамти податке о **новом тренингу**. (АПСО)
4. **Систем памти** податке о **новом тренингу**. (СО)
5. **Систем приказује кориснику** запамћени **нови тренинг** и поруку: “**Систем** је запамтио **нови тренинг**”. (ИА)

### Алтернативна сценарија

- 5.1. Уколико **систем** не може да запамти податке о **новом тренингу** он приказује **кориснику** поруку “**Систем** не може да запамти **нови тренинг**”. (ИА)

## СК6: Случај коришћења – Промена тренинга (Сложен СК)

### Назив СК

Промена тренинга

### Актори СК

Корисник

### Учесници СК

Корисник и систем (програм)

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

### Основни сценарио СК

1. Корисник уноси вредност по којој претражује тренинге. (АПУСО)
2. Корисник позива систем да нађе тренинге по задатој вредности. (АПСО)
3. Систем тражи тренинге по задатој вредности. (СО)
4. Систем приказује кориснику тренинге и поруку: „Систем је нашао тренинге по задатој вредности“. (ИА)
5. Корисник бира жељени тренинг. (АПУСО)
6. Корисник позива систем да учита тренинг. (АПСО)
7. Систем учитава тренинг. (СО)
8. Систем приказује кориснику податке о тренингу и поруку: „Систем је учитао тренинг“. (ИА)
9. Корисник уноси (мења) податке о тренингу. (АПУСО)
10. Корисник контролише да ли је коректно унео податке о тренингу. (АНСО)
11. Корисник позива систем да запамти податке о тренингу. (АПСО)
12. Систем памти податке о тренингу. (СО)
13. Систем приказује кориснику запамћени тренинг и поруку: „Систем је запамтио тренинг.“ (ИА)

### Алтернативна сценарија

- 4.1 Уколико систем не може да нађе тренинге он приказује кориснику поруку: „Систем не може данађе тренинге по задатој вредности“. Прекида се извршење сценарија. (ИА)
- 8.1. Уколико систем не може да учита тренинг он приказује кориснику поруку „Систем не може да учита тренинг“. Прекида се извршење сценарија. (ИА)
- 13.1. Уколико систем не може да измени податке о тренингу он приказује кориснику поруку: „Систем не може да измени тренинг“. (ИА)

## СК7: Случај коришћења – Брисање тренинга

### Назив СК

Брисање тренинга

### Актори СК

Корисник

### Учесници СК

Корисник и систем (програм)

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

### Основни сценарио СК

1. Корисник уноси вредност по којој претражује тренинге. (АПУСО)
2. Корисник позива систем да нађе тренинге по задатој вредности. (АПСО)
3. Систем тражи тренинге по задатој вредности. (СО)
4. Систем приказује кориснику тренинге и поруку: “Систем је нашао тренинге по задатој вредности”. (ИА)
5. Корисник бира тренинг. (АПУСО)
6. Корисник позива систем да обрише тренинг. (АПСО)
7. Систем брише тренинг. (СО)
8. Систем приказује кориснику поруку: “Систем је обрисао тренинг.” (ИА)

### Алтернативна сценарија

- 4.1 Уколико систем не може да нађе тренинге он приказује кориснику поруку: “Систем не може данађе тренинге по задатој вредности”. Прекида се извршење сценарија. (ИА)
- 8.1 Уколико систем не може да обрише тренинг он приказује кориснику поруку “Систем не може да обрише тренинг”. (ИА)

## СК8: Случај коришћења – Претраживање тренинга

### Назив СК

Претраживање тренинга

### Актери СК

Корисник

### Учесници СК

Корисник и систем (програм)

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

### Основни сценарио СК

1. Корисник уноси вредности по којој се претражују тренинзи. (АПУСО)
2. Корисник позива систем да нађе тренинге по задатој вредности. (АПСО)
3. Систем тражи тренинге по задатој вредности. (СО)
4. Систем приказује кориснику податке о тренинзима и поруку: „Систем је нашао тренинге по задатој вредности“. (ИА)
5. Корисник бира тренинг. (АПУСО)
6. Корисник позива систем да учита тренинг. (АПСО)
7. Систем учитава тренинг. (СО)
8. Систем приказује кориснику податке о тренингу и поруку: „Систем је учитао тренинг“. (ИА)

### Алтернативна сценарија

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

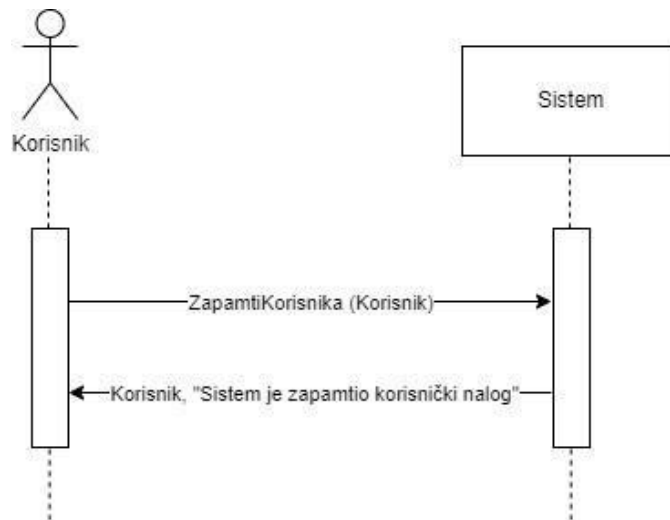
8.1. Уколико систем не може да учита тренинг, он приказује кориснику поруку: „Систем не може да учита тренинг“. (ИА)

## 2. Фаза анализе

### 2.1. Системски дијаграм секвенци

#### 2.1.1. ДС1: Дијаграм секвенци СК – Креирање корисничког налога

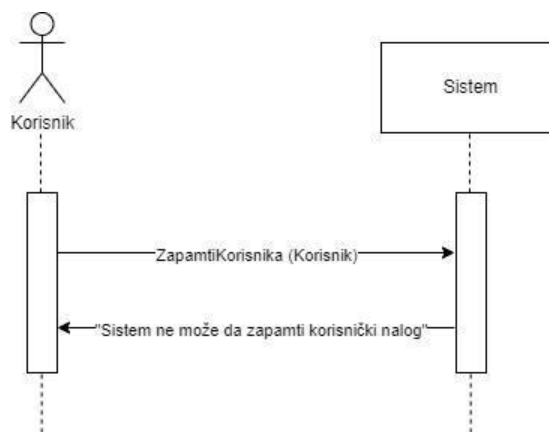
1. **Корисник** **позива** **систем** да запамти податке о **корисничком налогу**. (АПСО)
2. **Систем** **приказује** **кориснику** запамћени **кориснички налог** и поруку: “**Систем** је запамтио **кориснички налог**”. (ИА)



Слика 2: ДС1: Дијаграм секвенци СК – Креирање корисничког налога – основни сценарио

#### Алтернативна сценарија

- 2.1. Уколико **систем** не може да запамти податке о **корисничком налогу** он приказује **кориснику** поруку “**Систем** не може да запамти **кориснички налог**”. (ИА)



Слика 3: ДС1: Алтернативни сценарио

Са наведених дијаграма секвенци могу се уочити следеће системске операције:

- Сигнал **ZapamtiKorisnika (Korisnik)**

### 2.1.2. ДС2: Дијаграм секвенци СК – Пријава на систем

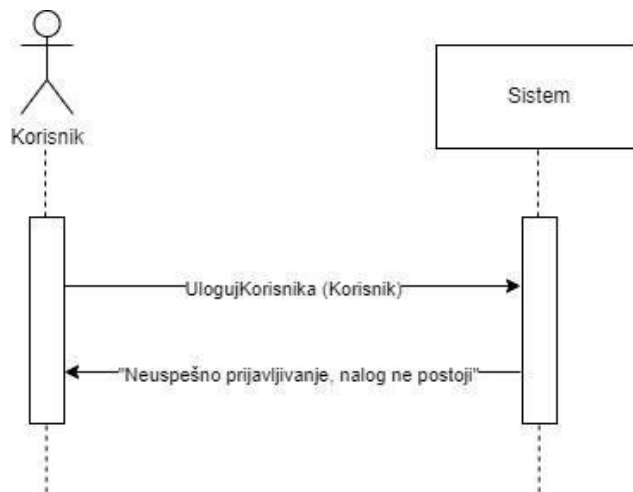
1. Корисник позива систем да пронађе корисника са унетим параметрима. (АПСО)
2. Систем приказује кориснику поруку: „Успешно пријављивање“. (ИА)



Слика 4: ДС2: Дијаграм секвенци СК – Пријава на систем – основни сценарио

#### Алтернативна сценарија

- 2.1. Ако систем не може да пронађе корисника са задатим параметрима, систем избацује кориснику поруку: „Неуспешно пријављивање, налог не постоји“. (ИА)



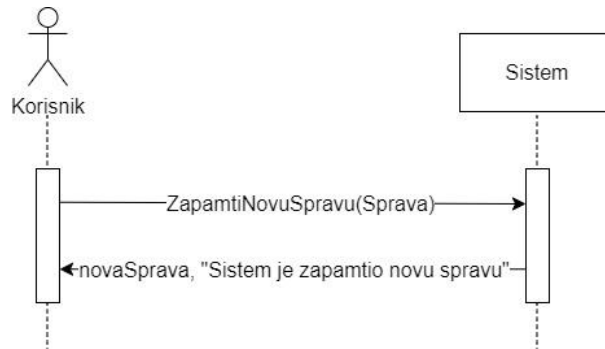
Слика 5: ДС2: Алтернативни сценарио

Са наведених дијаграма секвенци могу се уочити следеће системске операције:

- Сигнал **UlogujKorisnika (Korisnik)**

### 2.1.3. ДСЗ: Дијаграм секвенци СК – Креирање нове справе

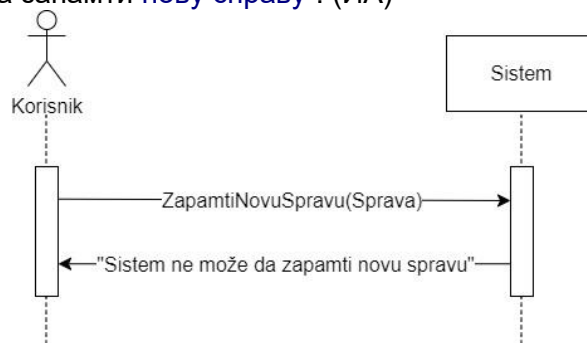
1. Корисник позива систем да запамти податке о новој справу. (АПСО)
2. Систем приказује кориснику запамћену нову справу и поруку: "Систем је запамтио нову справу". (ИА)



Слика 6: ДСЗ: Дијаграм секвенци СК – Креирање нове справе – основни сценарио

#### Алтернативна сценарија

- 2.1. Уколико систем не може да запамти податке о новој справу он приказује кориснику поруку "Систем не може да запамти нову справу". (ИА)



Слика 7: ДСЗ: Алтернативни сценарио

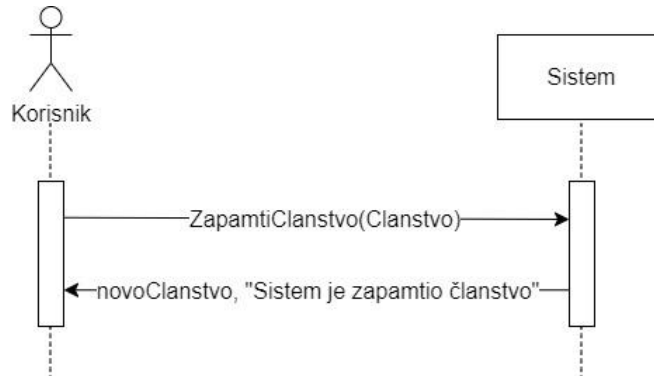
Са наведених дијаграма секвенци могу се уочити следеће системске операције:

- Сигнал **ZapamtiNovuSpravu (Sprava)**



#### 2.1.4. ДС4: Дијаграм секвенци СК – Креирање чланства

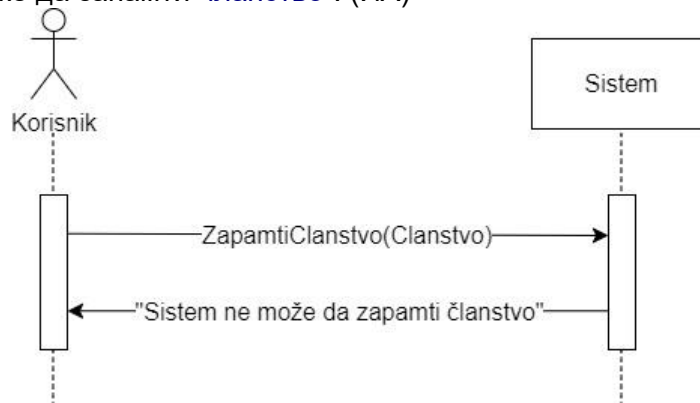
1. Корисник позива систем да запамти податке о чланству. (АПСО)
2. Систем приказује кориснику запамћено чланство и поруку: "Систем је запамтио чланство". (ИА)



Слика 8: ДС4: Дијаграм секвенци СК – Креирање чланства – основни сценарио

#### Алтернативна сценарија

- 1.1. Уколико систем не може да запамти податке о чланству он приказује кориснику поруку "Систем не може да запамти чланство". (ИА)



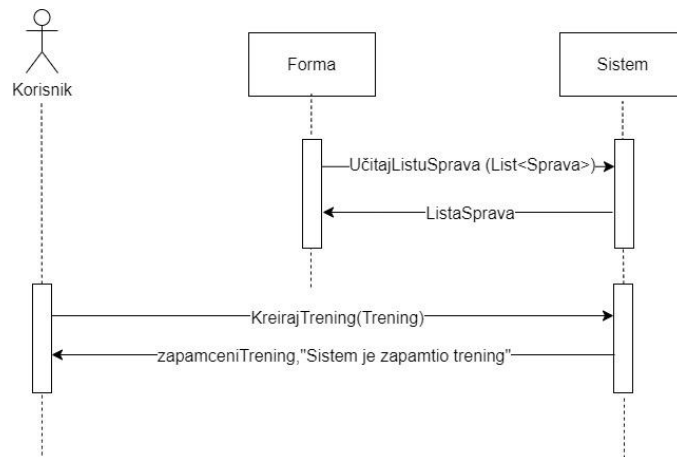
Слика 9: ДС4: Алтернативни сценарио

Са наведених дијаграма секвенци могу се уочити следеће системске операције:

- Сигнал **ZapamtiClanstvo(Clanstvo)**

### 2.1.5. ДС5: Дијаграм секвенци СК – Креирање тренинга (сложен случај)

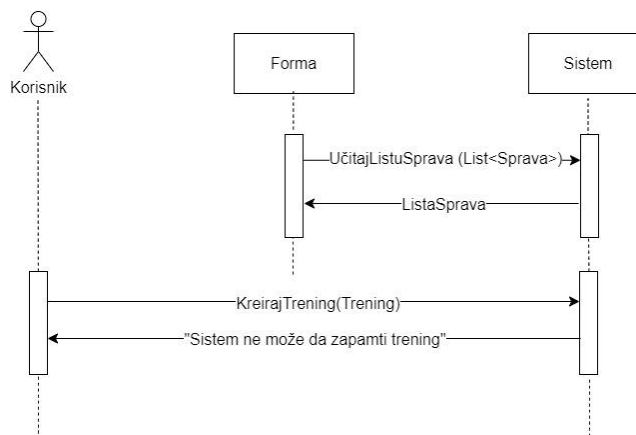
1. **Форма** **позива** **систем** да учита листу **справа**. (АПСО)
2. **Систем** **враћа** форми листу **справа**. (ИА)
3. **Корисник** **позива** **систем** да запамти податке о **тренингу**. (АПСО)
4. **Систем** **приказује** **кориснику** поруку: „**Систем** је запамтио **тренинг**“. (ИА)



Слика 10: ДС5: Дијаграм секвенци СК – Креирање тренинга – основни сценарио

#### Алтернативна сценарија

- 4.1. Уколико **систем** не може да запамти податке о **тренингу**, он приказује **кориснику** поруку: „**Систем** не може да запамти **тренинг**“. (ИА)



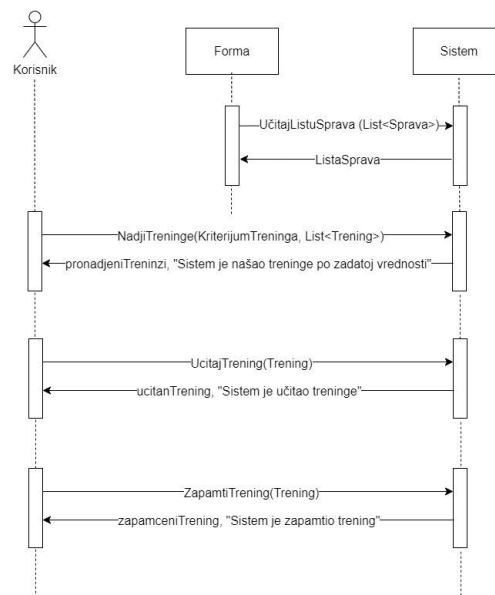
Слика 11: ДС5: Алтернативни сценарио

Са наведених дијаграма секвенци могу се уочити следеће системске операције:

- Сигнал **UčitajListuSprava (List<Sprava>)**
- Сигнал **KreirajTrening(Trening)**

## 2.1.6. ДС6: Дијаграм секвенци СК – Измена тренинга (сложен случај)

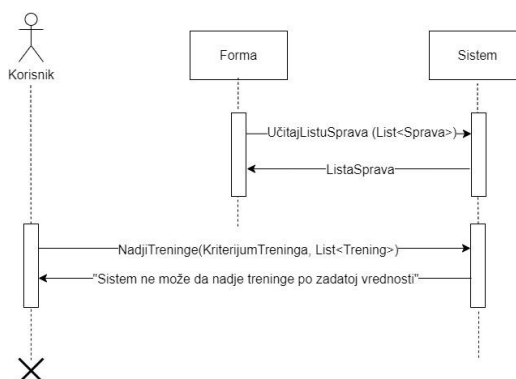
1. **Форма** **позива систем** да учита листу **справа**. (АПСО)
2. **Систем** **враћа форми** листу **справа**. (ИА)
3. **Корисник** **позива систем** да нађе **тренинге** по задатој вредности. (АПСО)
4. **Систем** **приказује кориснику** тренинге и поруку: „**Систем** је нашао тренинге по задатој вредности“. (ИА)
5. **Корисник** **позива систем** да учита **тренинге**. (АПСО)
6. **Систем** **приказује кориснику** податке о **тренингу** и поруку: „**Систем** је прочитао тренинг“. (ИА)
7. **Корисник** **позива систем** да запамти податке о **тренингу**. (АПСО)
8. **Систем** **приказује кориснику** запамћене податке о **тренингу** и поруку: „**Систем** је запамтио тренинг.“ (ИА)



Слика 12: ДС6: Дијаграм секвенци СК –Измена тренинга – основни сценарио

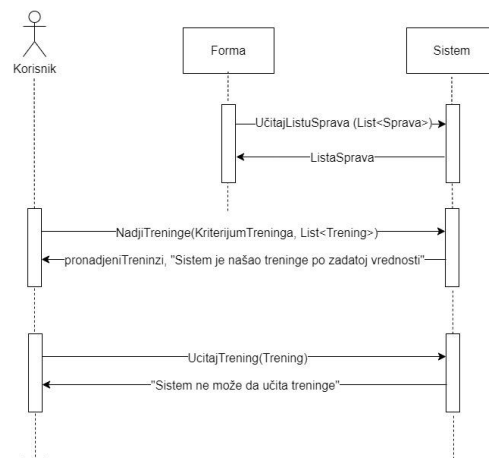
### Алтернативна сценарија

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



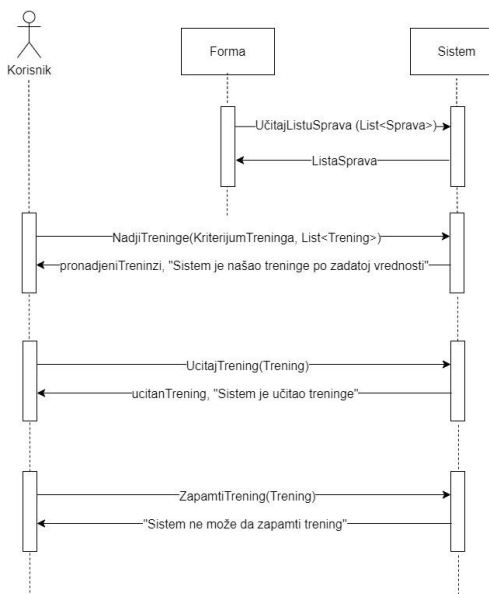
Слика 13: ДС6: Алтернативни сценарио 1

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



Слика 14: ДС6: Алтернативни сценарио 2

8.1. Уколико **систем** не може да измени податке о **тренингу**, он **кориснику** приказује поруку: „**Систем** не може да измени **тренинг**“. (ИА)



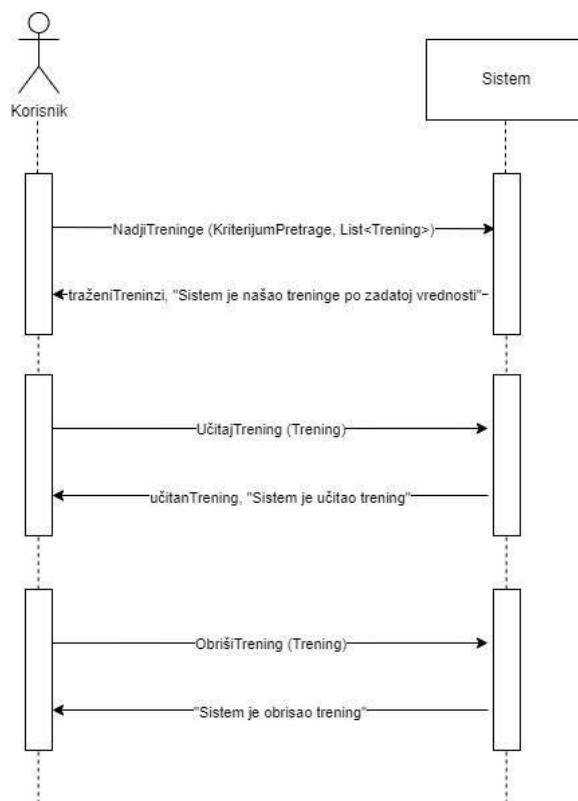
Слика 15: ДС6: Алтернативни сценарио 3

Са наведених дијаграма секвенци могу се уочити следеће системске операције:

- Сигнал **NadjiTreninge(KriterijumPretrage, List<Trening>)**
- Сигнал **UčitajTrening(Trening)**
- Сигнал **IzmeniTrening(Trening)**
- Сигнал **UcitajListuSprava(List<Sprava>)**

### 2.1.7. ДС7: Дијаграм секвенци СК – Брисање тренинга

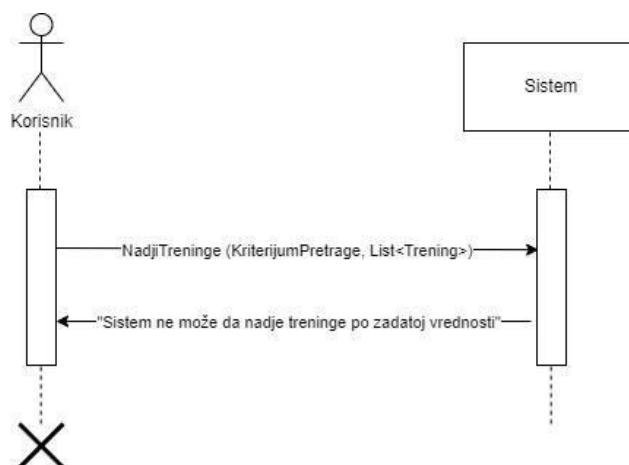
1. Корисник позива систем да нађе тренинге по задатој вредности. (АПСО)
2. Систем приказује кориснику тренинге и поруку: „Систем је нашао тренинге по задатој вредности“. (ИА)
3. Корисник позива систем да учита тренинг. (АПСО)
4. Систем приказује кориснику тренинг и поруку: „Систем је учитао тренинг“. (ИА)
5. Корисник позива систем да обрише тренинг. (АПСО)
6. Систем приказује кориснику поруку: „Систем је обрисао тренинг.“ (ИА)



Слика 16: ДС7: Дијаграм секвенци СК –Брисање тренинга –основни сценарио

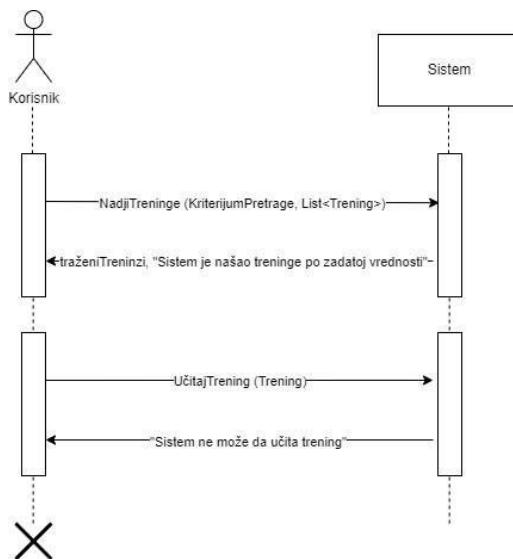
### Алтернативна сценарија

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



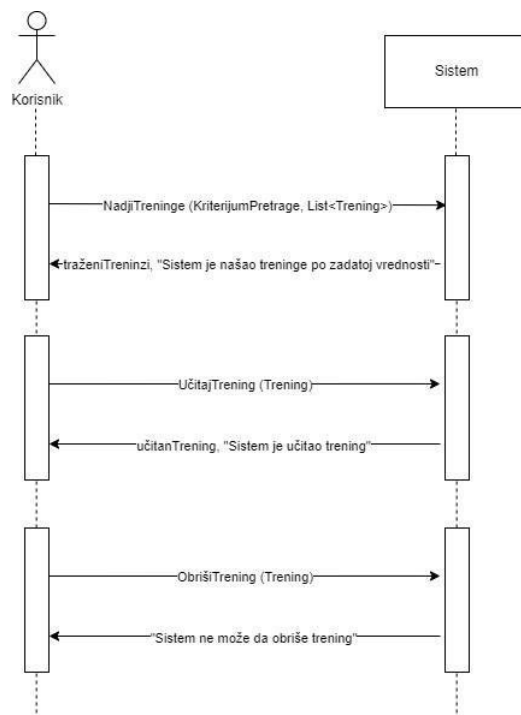
Слика 17: ДС7: Алтернативни сценарио 1

4.1. Уколико **систем** не може да учита **тренинг** он приказује **кориснику** поруку: „**Систем** не може даучита **тренинг**“. Прекида се извршавање сценарија (ИА)



Слика 18: ДС7: Алтернативни сценарио 2

6.1. Уколико **систем** не може да обрише **тренинг** он приказује **кориснику** поруку “**Систем** не може даобрише **тренинг**“. (ИА)



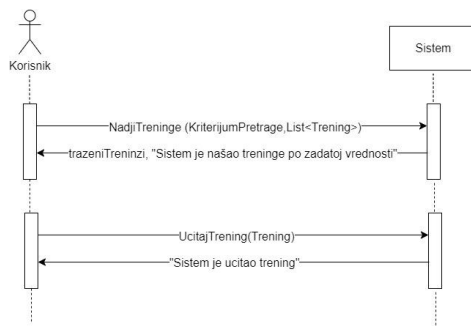
Слика 19: ДС7: Алтернативни сценарио 3

Са наведених дијаграма секвенци могу се уочити следеће системске операције:

- Сигнал **NadjiTreninge(KriterijumPretrage, List<Trening>)**
- Сигнал **UčitajTrening (Trening)**
- Сигнал **ObrišiTrening (Trening)**

### 2.1.8. ДС8: Дијаграм секвенци СК – Претраживање тренинга

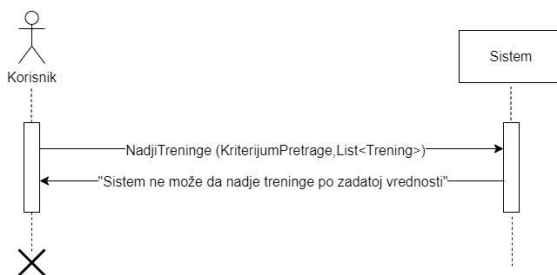
1. Корисник позива систем да нађе тренинге по задатој вредности. (АПСО)
2. Систем приказује кориснику податке о тренинзима и поруку: „Систем је нашао тренинге по задатој вредности“. (ИА)
3. Корисник позива систем да да учита тренинг. (АПСО)
4. Систем приказује кориснику поруку: „Систем је учитао тренинг“. (ИА)



Слика 20: Дијаграм секвенци – Претраживање тренинга – основни сценарио

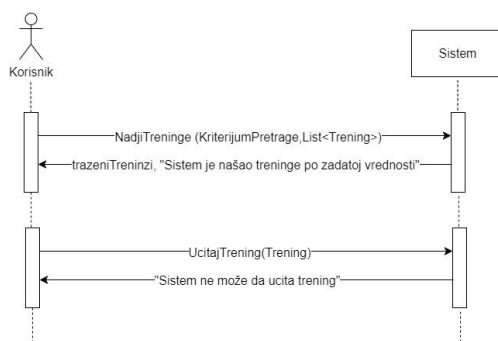
#### Алтернативна сценарија

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



Слика 21: Алтернативни сценарио 1

- 4.1. Уколико систем не мође да учита тренинг, он приказује кориснику поруку: „Систем не може да учита тренинг“. (ИА)



Слика 22: Алтернативни сценарио 2

Са наведених дијаграма секвенци могу се уочити следеће системске операције:

- Сигнал **NadjiTreninge(KriterijumPretrage, List<Trening>)**
- Сигнал **UcitajTrening (Trening)**

## 2.2. Преглед системских операција

Као резултат анализе сценарија добијено је 10 системских операција које је потребно пројектовати:

- Сигнал **ZapamtiKorisnika (Korisnik)**
- Сигнал **UlogujKorisnika (Korisnik)**
- Сигнал **ZapamtiNovuSpravu (Sprava)**
- Сигнал **ZapamtiClanstvo(Clanstvo)**
- Сигнал **UčitajListuSprava(List<Sprava>)**
- Сигнал **KreirajTrening (Trening)**
- Сигнал **NadjiTreninge(KriterijumPretrage, List<Trening>)**
- Сигнал **UčitajTrening (Trening)**
- Сигнал **IzmeniTrening(Trening)**
- Сигнал **ObrišiTrening (Trening)**



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

Уговор УГ1: ZapamtiKorisnika

**Операција:** ZapamtiKorisnika (Korisnik) сигнал

**Веза са СК:** СК1

**Предуслови:** Структурна и вредносна ограничења над објектом *Korisnik* треба да буду задовољена

**Постуслови:** Нови корисник је регистрован

Уговор УГ2: UlogujKorisnika

**Операција:** UlogujKorisnika (Korisnik) сигнал

**Веза са СК:** СК2

**Предуслови:** /

**Постуслови:** Корисник је пријављен на систем

Уговор УГ3: ZapamtiNovuSpravu

**Операција:** ZapamtiNovuSpravu (Sprava) сигнал

**Веза са СК:** СК3

**Предуслови:** Структурна и вредносна ограничења над објектом *Sprava* треба да буду задовољена

**Постуслови:** Справа је запамћена

Уговор УГ4: ZapamtiClanstvo

**Операција:** ZapamtiClanstvo (Clanstvo) сигнал

**Веза са СК:** СК4

**Предуслови:** Структурна и вредносна ограничења над објектом *Sprava* треба да буду задовољена

**Постуслови:** Чланство је запамћено

Уговор УГ5: UcitajListuSprava

**Операција:** UcitajListuSprava (List<Sprava>) сигнал

**Веза са СК:** СК5, СК 6

**Предуслови:** /

**Постуслови:** /

Уговор УГ6: KreirajTrening

**Операција:** KreirajTrening (Trening) сигнал

**Веза са СК:** СК5

**Предуслови:** Структурна и вредносна ограничења над објектом *Trening* треба да буду задовољена

**Постуслови:** Тренинг је запамћен

Уговор УГ7: NadjiTreninge

**Операција:** NadjiTreninge (KriterijumPretrage, List<Trening>) сигнал

**Веза са СК:** СК6, СК7, СК 8

**Предуслови:** /

**Постуслови:** /

Уговор УГ8: UčitajTrening

**Операција:** UčitajTrening (Trening) сигнал

**Веза са СК:** СК6, СК 7, СК8

**Предуслови:** /

**Постуслови:** /

Уговор УГ9: IzmeniTrening(Trening)

**Операција:** IzmeniTrening (Trening) сигнал

**Веза са СК:** СК6

**Предуслови:** Структурна и вредносна ограничења над објектом *Trening* треба да буду задовољена

**Постуслови:** Подаци о вежби су измењен

Уговор УГ10: ObrišiTrening

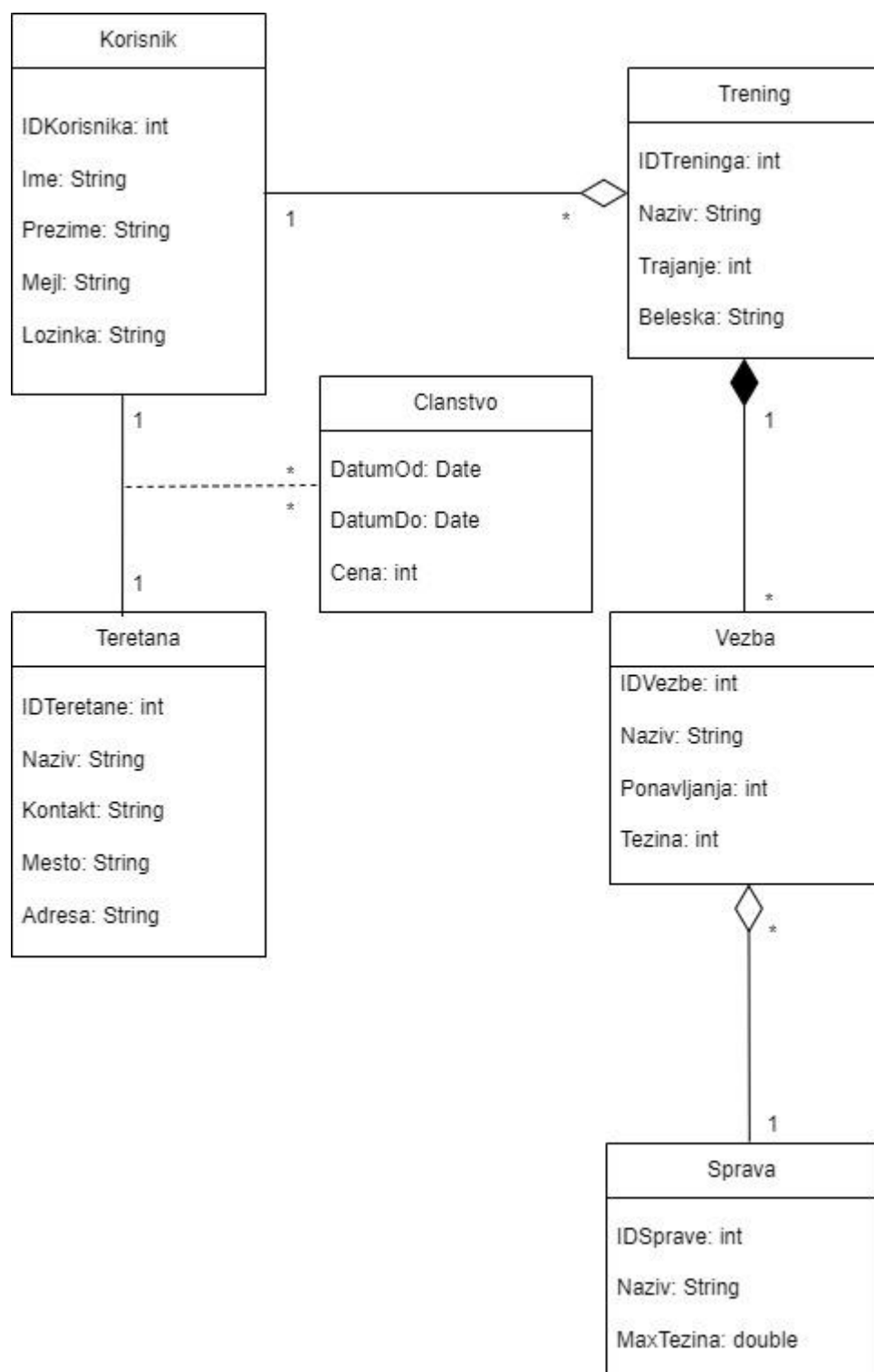
**Операција:** ObrišiTrening (Trening) сигнал

**Веза са СК:** СК7

**Предуслови:** Структурна ограничења над објектом *Trening* треба да буду задовољена

**Постуслови:** Тренинг је обрисан

## 2.4. Структура софтверског система – Концептуални (доменски) модел



Слика 23: Концептуални модел

## 2.5. Структура софтверског система – релациони

Korisnik (IDKorisnika, Ime, Prezime, Mejl, Lozinka)

Vezba (IDVezbe, IDTreninga, Naziv, Ponavljanja, Tezina, IDSprave)

Trening (IDTreninga, Naziv, Trajanje, Beleska, IDKorisnika)

Teretana (IDTeretane, Naziv, Kontakt, Mesto, Adresa)

Clanstvo (IDTeretane, IDKorisnika, DatumOd, DatumDo, Cena)

Sprava (IDSprave, Naziv, MaxTezina)

Табела 1: Корисник

Tabela Korisnik		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Атрибути	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	<b>INSERT:</b> /  <b>UPDATE:</b> CASCADES Trening, Clanstvo  <b>DELETE:</b> RESTRICTED Trening, Clanstvo
	IDKorisnika	Integer	Not null and >0			
	Ime	String	Not null			
	Prezime	String	Not null			
	Mejl	String	Not null			
	Lozinka	String	Not null			

Табела 2: Вежба

Tabela Vezba		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Атрибути	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	<b>INSERT:</b> CASCADES Sprava, Trening <b>UPDATE:</b> CASCADES Sprava, Trening <b>DELETE:</b> /
	IDVezbe	Integer	Not null and >0			
	Naziv	String	Not null			
	Ponavljanja	Integer	Not null and >0			
	Tezina	Integer	Not null and >0			
	IDTreninga	Integer	Not null and >0			
	IDSprave	Integer	Not null and >0			

Табела 3: Тренинг

Tabela Trening		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Атрибути	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	<b>INSERT:</b> RESTRICTED Korisnik
	IDTreninga	Integer	Not null and >0			<b>UPDATE:</b> RESTRICTED Korisnik
	Naziv	String	Not null			
	Trajanje	Integer	Not null and >0			<b>UPDATE:</b> CASCADES Vezba
	Beleska	String	Not null			
	IDKorisnika	Integer	Not null and >0			<b>DELETE:</b> RESTRICTED Vezba

Табела 4: Теретана

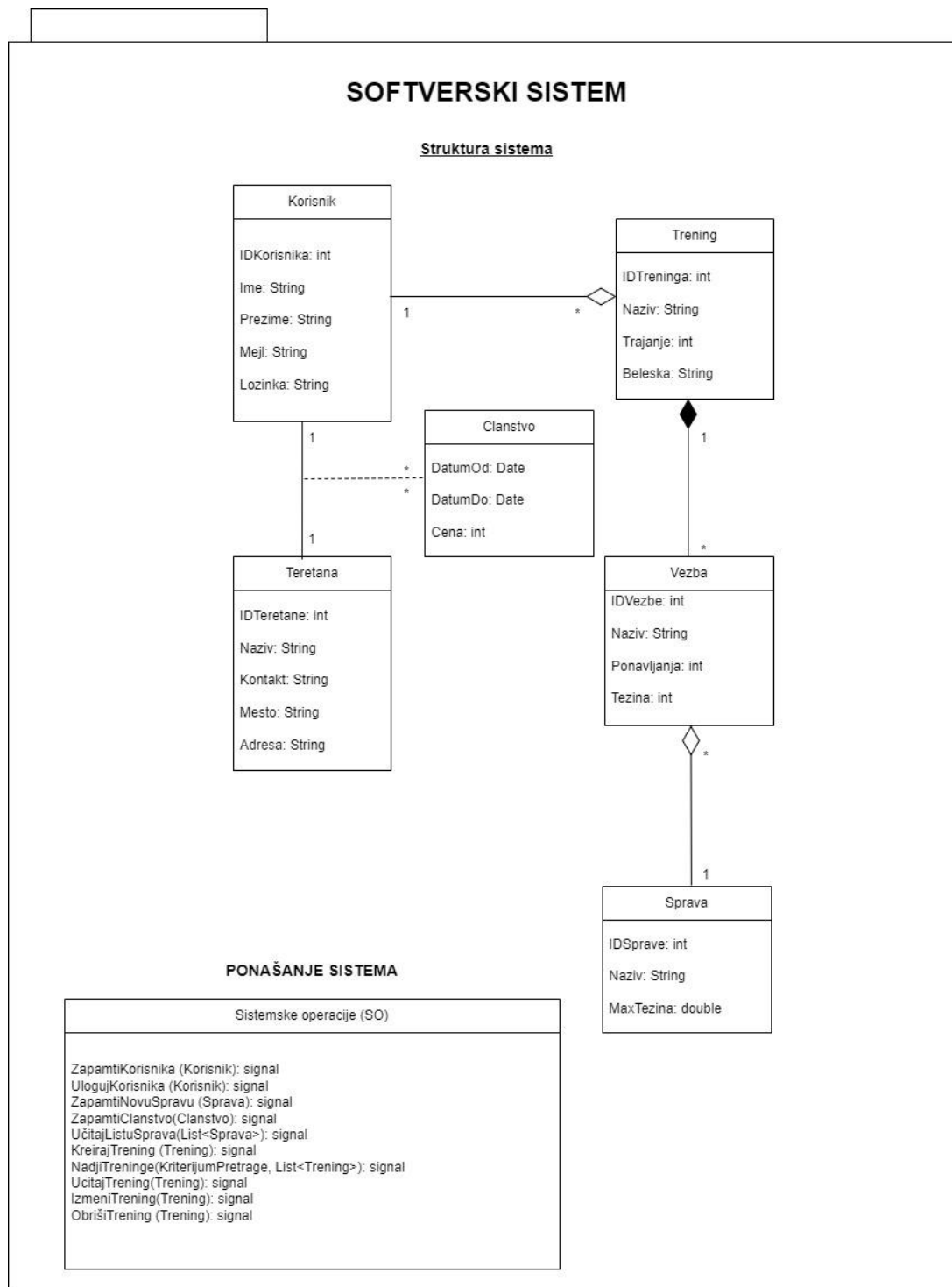
Tabela Teretana		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Атрибути	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	<b>INSERT:</b> /
	IDTeretane	Integer	Not null and >0			<b>UPDATE:</b> CASCADES Clanstvo
	Naziv	String	Not null			
	Kontakt	String	Not null			<b>DELETE:</b> RESTRICTED Clanstvo
	Mesto	String	Not null			
	Adresa	String	Not null			

Табела 5: Чланство

Tabela Clanstvo		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Атрибути	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	<b>INSERT:</b> RESTRICTED Korisnik, Teretana  <b>UPDATE:</b> RESTRICTED Korisnik, Teretana  <b>DELETE:</b> /
	IDTeretane	Integer	Not null and >0			
	IDKorisnika	Integer	Not null and >0			
	DatumOd	Date	Not null			
	DatumDo	Date	Not null			
	Cena	Integer	Not null and >0			

Табела 6: Справа

Tabela Sprava		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Атрибути	Ime	Tip atributa	Vrednost atributa	Međuzavisnost atributa jedne tabele	Međuzavisnost atributa više tabela	<b>INSERT:</b> /  <b>UPDATE:</b> CASCADES Vezba  <b>DELETE:</b> RESTRICTED Vezba
	IDSprave	Integer	Not null and >0			
	Naziv	String	Not null			
	MaxTezina	Double	Not null and >0			



Слика 24: Софтверски систем

### 3. Фаза пројектовања

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

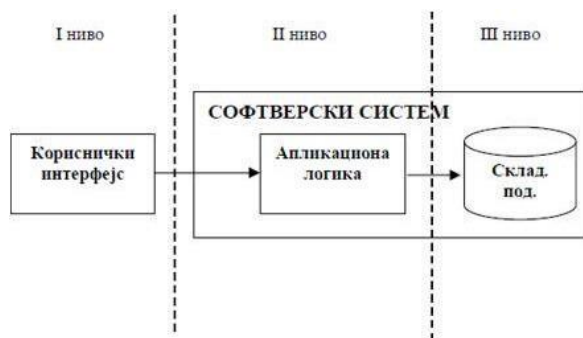
Пројектовање архитектуре софтверског система обухвата пројектовање корисничког интерфејса (пројектовање контролера корисничког интерфејса и екранских форми), апликационе логике (пројектовање контролера апликационе логике и пословне логике) и складишта података (брокер базе података).

#### Архитектура софтверског система

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

- Кориснички интерфејс
- Апликациона логика
- Складиште података

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

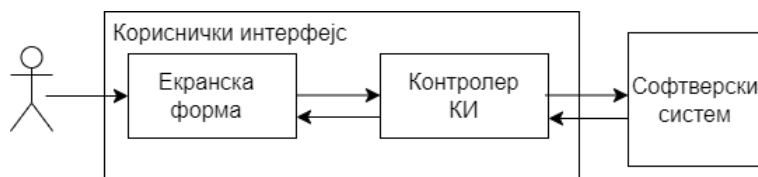


Слика 25: Тронивојска архитектура

#### 3.1. Пројектовање корисничког интерфејса

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

- Екранске форме
- Контролера корисничког интерфејса



Слика 26: Структура корисничког интерфејса

Корисник комуницира са екранском формом уз помоћ графичких компоненти. Екранска форма прихвата унете податке као и догађаје које корисник иницира, позива контролера корисничког интерфејса и прослеђује му те податке, али и прима податке од корисничког контролера које потом приказује кориснику. Контролер има улогу да конвертује податке добијене из графичких компоненти у одговарајуће доменске класе и да пошаље захтев за извршење системске операције. Такође, он прима од софтверског система податке које конвертује у елементе екранске форме.



### 3.1.1. Пројектовање екранских форми

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

Постоје два аспекта пројектовања екранске форме:

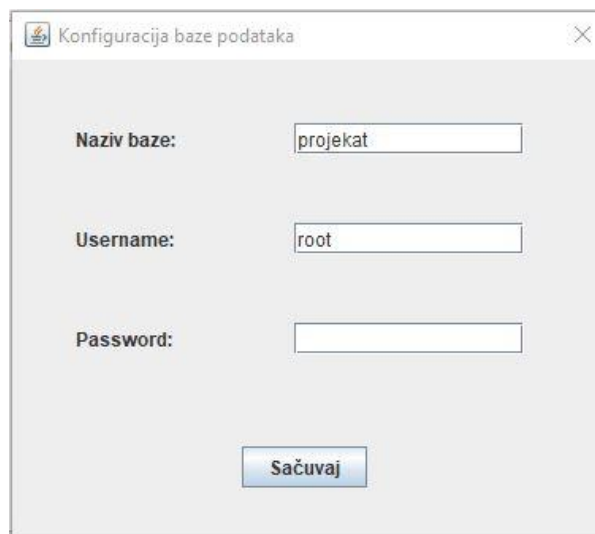
1. Пројектовање сценарија СК који се изводе преко екранске форме
2. Пројектовање метода екранске форме

На серверској страни програма налази се форма која регулише стање сервера (да ли је покренут или не). Приликом покретања она изгледа овако:



Слика 27: Серверска форма – заустављен сервер

Кликом на дугме „**Konfiguracija**“ уносимо податке који су везани конкретно за базу података на којој ће овај пројекат радити.



Слика 28: Конфигурација базе података

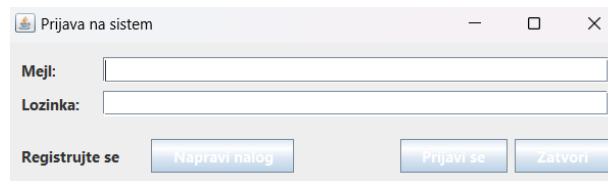
Затим, кликом на дугме „**Pokreni server**“, покрећемо сервер и тада серверски сокет ослушкује пристизање клијената. Тада форма сервера изгледа овако:



Слика 29: Серверска форма – покренут сервер

Кликом на дугме „**Zaustavi server**“, сервер се поново зауставља и престаје да обрађује клијентске захтеве.

На клијентској страни потребно је уловати се како би се апликација користила. Форма за логовање изгледа овако:



Слика 30: Форма за пријаву корисника на систем

Након логовања на систем, приказује се главна екранска форма из које се може доћи до осталих екранских форми.



Слика 31: Главна клијентска форма

## СК1: Случај коришћења – Креирање корисничког налога

### Назив СК

Креирање корисничког налога

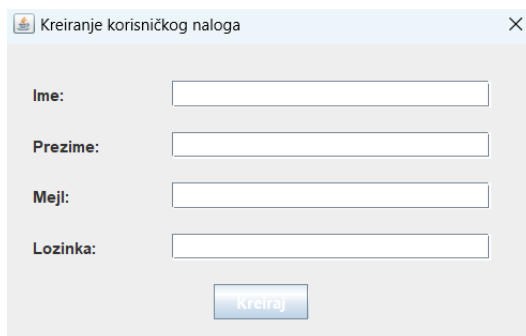
### Актори СК

Корисник

### Учесници СК

Корисник и систем (програм)

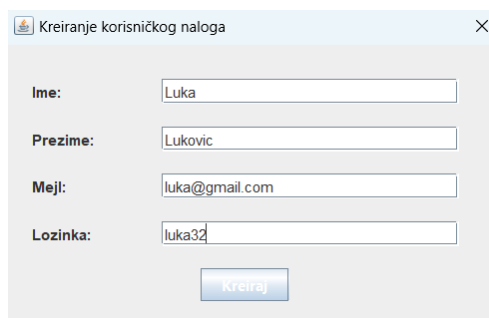
**Предуслов:** Систем је укључен и корисник је улогован под својом шифром. Систем приказује форму за рад са корисничким налогом.



Слика 32: Форма за креирање корисничког налога

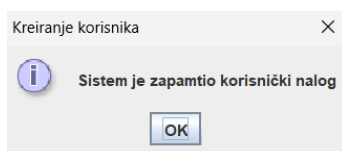
### Основни сценарио СК

1. Корисник уноси податке у кориснички налог. (АПУСО)
2. Корисник контролише да ли је коректно унео податке у кориснички налог. (АНСО)



Слика 33: попуњени подаци за креирање корисничког налога

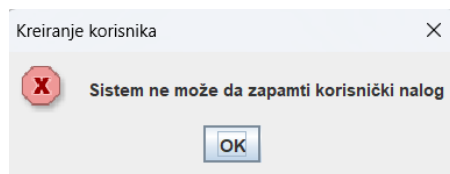
3. Корисник позива систем да запамти податке о корисничком налогу. (АПСО)
4. Систем памти податке о корисничком налогу. (СО)
5. Систем приказује кориснику запамћени кориснички налог и поруку: “Систем је запамтио кориснички налог”. (ИА)



Слика 34: Успешно креирање корисничког налога

### Алтернативна сценариа

- 5.1. Уколико **систем** не може да запамти податке о **корисничком налогу** он приказује **кориснику** поруку “**Систем** не може да запамти **кориснички налог**”. (ИА)



Слика 35: Неуспешно креирање корисничког налога

## СК2: Случај коришћења – Пријава на систем

### Назив СК

Пријава **корисника** на систем

### Актори СК

**Корисник**

### Учесници СК

**Корисник** и **систем**

**Предуслов:** **Систем** је укључен и приказује форму за пријављивање **корисника** на систем. **Корисник** већ поседује налог.

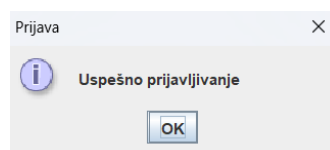
Слика 36: Форма за пријављивање на систем

### Основни сценарио СК

1. **Корисник** уноси своје параметре за пријављивање на **систем**. (АПУСО)

Слика 37: Пријављивање корисника на систем

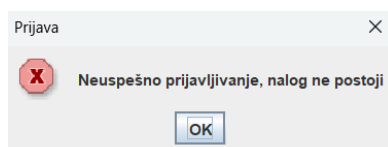
2. **Корисник** контролише своје унете податке за пријављивање. (АНСО)
3. **Корисник** позива **систем** да пронађе **корисника** са унетим параметрима. (АПСО)
4. **Систем** претражује **корисника** на основу задатих параметара како би извршио пријављивање. (СО)
5. **Систем** приказује **кориснику** поруку: „Успешно пријављивање“. (ИА)



Слика 38: Успешно пријављивање корисника

### Алтернатива сценарија

- 5.1. Ако **систем** не може да пронађе **корисника** са задатим параметрима, **систем** избацује **кориснику** поруку: „Неуспешно пријављивање, налог не постоји“. (ИА)



Слика 39: Неуспешно пријављивање корисника на систем

### СКЗ: Случај коришћења – Креирање нове справе

#### Назив СК

Креирање нове справе

#### Актори СК

Корисник

#### Учесници СК

Корисник и систем (програм)

**Предуслов:** Систем је укључен и корисник је улоган под својом шифром. Систем приказује форму за рад са новом справом.

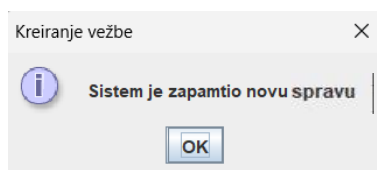
Слика 40: Форма за креирање нове справе

#### Основни сценарио СК

1. Корисник уноси податке у нову справу. (АПУСО)

Слика 41: Попуњена форма за креирање нове справе

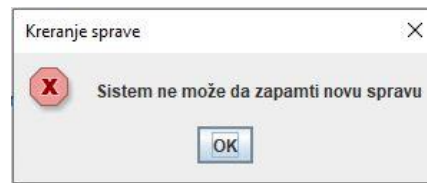
2. Корисник контролише да ли је коректно унео податке у нову справу. (АНСО)
3. Корисник позива систем да запамти податке о новој справи. (АПСО)
4. Систем памти податке о новој справи. (СО)
5. Систем приказује кориснику запамћену нову справу и поруку: "Систем је запамтио нову справу". (ИА)



Слика 42: Успешно креирање нове справе

### Алтернативна сценарија

5.1. Уколико **систем** не може да запамти податке о **новој вежби** он приказује **кориснику** поруку “**Систем** не може да запамти **нову вежбу**”. (ИА)



Слика 43: Неуспешно креирање нове справе

## СК4: Случај коришћења – Креирање чланства

### Назив СК

Креирање чланства

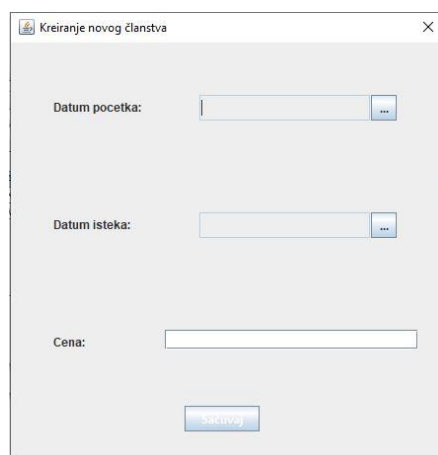
### Актори СК

Корисник

### Учесници СК

Корисник и систем (програм)

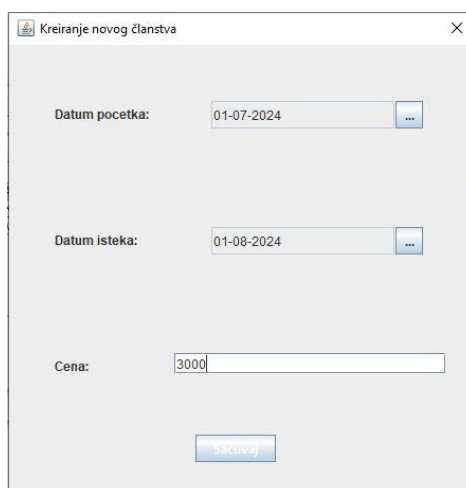
**Предуслов:** Систем је укључен и корисник је улогован под својом шифром. Систем приказује форму за рад са чланством.



Слика 44: Форма за рад са чланством

### Основни сценарио СК

1. Корисник уноси податке у чланство. (АПУСО)

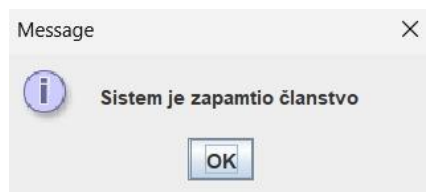


Слика 45: Попуњени подаци о чланству

2. Корисник контролише да ли је коректно унео податке у чланство. (АНСО)
3. Корисник позива систем да запамти податке о чланству. (АПСО)
4. Систем памти податке о чланству. (СО)



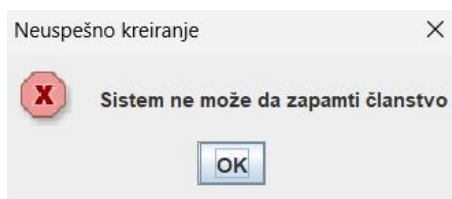
5. Систем приказује кориснику запамћено чланство и поруку: “Систем је запамтио чланство”. (ИА)



Слика 46: Порука система о успешном креирању чланства

#### Алтернативна сцена

- 5.1. Уколико систем не може да запамти податке о чланству он приказује кориснику поруку “Систем не може да запамти чланство”. (ИА)



Слика 47: Порука система о неуспешном креирању чланства

## СК5: Случај коришћења – Креирање тренинга (сложен случај)

### Назив СК

Креирање тренинга

### Актори СК

Корисник

### Учесници СК

Корисник и систем (програм)

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

Naziv	Ponavljanja	Tezina	Sprava
-------	-------------	--------	--------

Слика 48: Форма за рад са тренингом

### Основни сценарио СК

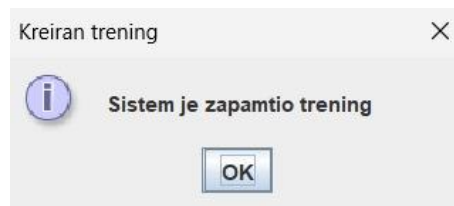
1. Корисник уноси податке о тренингу. (АПУСО)

Naziv	Ponavljanja	Tezina	Sprava
Leg press	15	120	leg press

Слика 49: Попуњени подаци о тренингу

2. Корисник контролише да ли је исправно унео податке о тренингу (АНСО)
3. Корисник позива систем да запамти податке о тренингу. (АПСО)
4. Систем памти и податке о тренингу (СО)

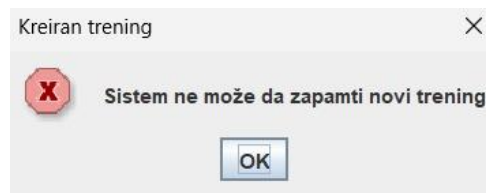
5. Систем приказује кориснику поруку: „Систем је запамтио тренинг“(ИА)



Слика 50: Порука система о успешном креирању тренинга

#### Алтернативна сценарија:

5.1. Уколико систем не може да запамти податке о тренингу, он приказује кориснику поруку: „Систем не може да запамти тренинг“ (ИА)



Слика 51: Порука система о неуспешном креирању тренинга

## СК6: Случај коришћења – Измена тренинга (сложен случај)

### Назив СК

Измена **тренинга**

### Актори СК

**Корисник**

### Учесници СК

**Корисник** и **систем** (програм)

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

The screenshot shows a window titled "Pretraga treninga". At the top, there is a text input field labeled "Naziv treninga:" followed by "Pretrazi" and "Osveži" buttons. Below this is a table with four columns: "naziv", "trajanje", "beleska", and "korisnik". The table contains three rows of data: "Pull" (65, ledja i biceps, Filip), "Legs" (155, Donji deo tela, Filip), and "Push" (70, gudi i triceps, Marko). Below the table are three buttons: "Izmeni", "Dodaj", and "Obriši".

naziv	trajanje	beleska	korisnik
Pull	65	ledja i biceps	Filip
Legs	155	Donji deo tela	Filip
Push	70	gudi i triceps	Marko

Слика 52: Форма за приказ свих тренинга

### Основни сценарио СК

1. **Корисник** уноси вредност по којој претражује **тренинге**. (АПУСО)

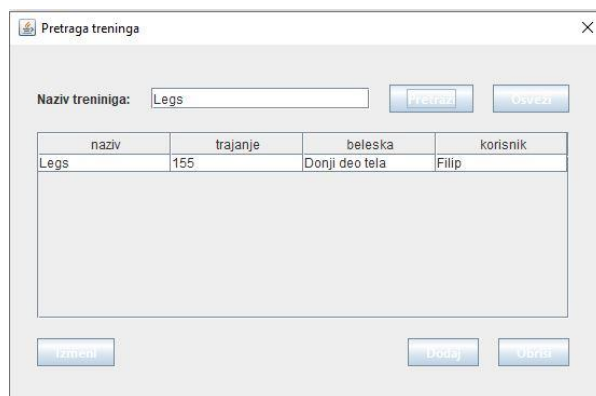
This screenshot is identical to the previous one, but the "Naziv treninga:" input field now contains the text "Legs".

Слика 53: Унета вредност по којој се претражују тренинзи

2. **Корисник** позива **систем** да нађе **тренинге** по задатој вредности. (АПСО)
3. **Систем** **тражи** **тренинге** по задатој вредности. (СО)
4. **Систем** приказује **кориснику** **тренинге** и поруку: "Систем је нашао **тренинге** по задатој вредности". (ИА)

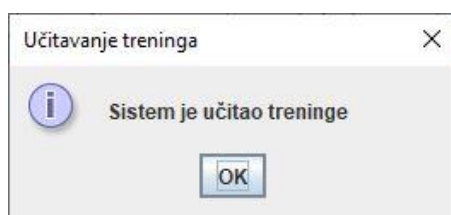
The screenshot shows a small dialog box titled "Pretraga treninga". It contains an information icon (i) and the text "Sistem je našao treninge po zadatoj vrednosti". There is an "OK" button at the bottom.

Слика 54: Порука система о успешном проналаску тренинга по задатој вредности



Слика 55: Приказ тренинга по задатој вредности

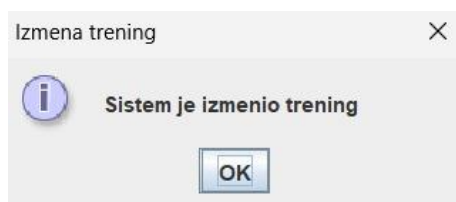
5. Корисник бира тренинг. (АПУСО)
6. Корисник позива систем да учита тренинге. (АПСО)
7. Систем учитава тренинге. (СО)
8. Систем приказује кориснику поруку: „Систем је прочитао тренинге“. (ИА)



Слика 56: Успешно учитани тренинзи

Слика 57: Приказ учитаних тренинга

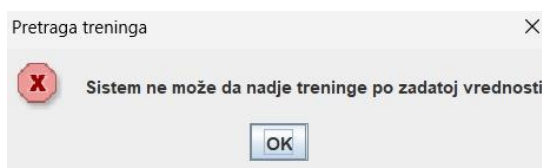
9. Корисник уноси (мења) податке о тренингу. (АПУСО)
10. Корисник контролише да ли је исправно унео податке о тренингу. (АНСО)
11. Корисник позива систем да запамти податке о тренингу. (АПСО)
12. Систем памти податке о тренингу. (СО)
13. Систем приказује кориснику запамћени тренинг и поруку: „Систем је изменио тренинг“. (ИА)



Слика 58: Порука система о успешној измени тренинга

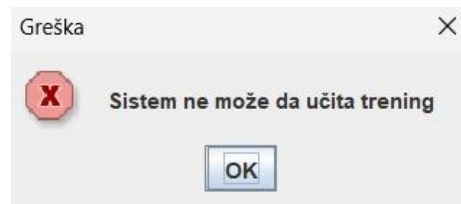
## Алтернативна сценарија

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



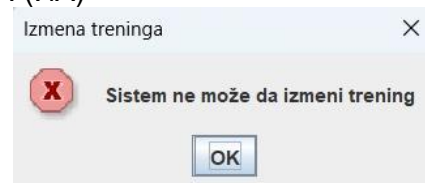
Слика 59: Порука система о неуспешно пронађеним тренинзима по задатој вредности

- 8.1. Уколико **систем** не може да учита **тренинг** он приказује **кориснику** поруку “**Систем** не може да учита **тренинг**”. (ИА)



Слика 60: Порука система о неуспешном учитавању тренинга

- 13.1. Уколико **систем** не може да измени **тренинг** он приказује **кориснику** поруку: „**Систем** не може да измени **тренинг**”. (ИА)



Слика 61: Порука система о неуспешној измени тренинга

## СК7: Случај коришћења – Брисање тренинга

### Назив СК

Брисање **тренинга**

### Актори СК

**Корисник**

### Учесници СК

**Корисник** и **систем** (програм)

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

The screenshot shows a window titled 'Pretraga treninga'. At the top, there is a text input field labeled 'Naziv treninga:' followed by 'Pretraži' and 'Brisa' buttons. Below this is a table with four columns: 'naziv', 'trajanje', 'beleska', and 'korisnik'. The table contains three rows of data: 'Pull' (65, ledja i biceps, Filip), 'Legs' (155, Donji deo tela, Filip), and 'Push' (70, gudi i triceps, Marko). At the bottom of the window are 'Izmeni', 'Dodaj', and 'Brisa' buttons.

naziv	trajanje	beleska	korisnik
Pull	65	ledja i biceps	Filip
Legs	155	Donji deo tela	Filip
Push	70	gudi i triceps	Marko

Слика 62: Форма за креирање тренинга

### Основни сценарио СК

1. **Корисник** уноси вредности по којој претражује **тренинге**. (АПУСО)

This screenshot is identical to the previous one, but the 'Naziv treninga:' input field now contains the text 'Legs'.

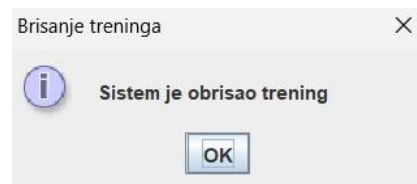
Слика 63: Попуњени подаци о тренингу

2. **Корисник** позива **систем** да нађе **тренинге** по задатој вредности (АПСО)
3. **Систем** тражи **тренинге** по задатој вредности. (СО)
4. **Систем** приказује **кориснику** поруку: „Систем је нашао **тренинге** по задатој вредности“. (ИА)

The screenshot shows a small message box titled 'Pretraga treninga'. It contains an information icon (i) and the text 'Sistem je našao treninge po zadatoj vrednosti'. There is an 'OK' button at the bottom.

Слика 64: Порука система о успешном проналажењу тренинга по задатој вредности

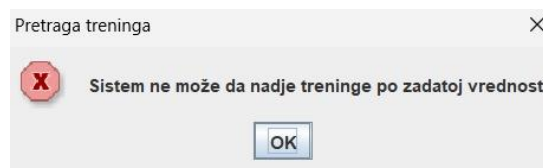
5. Корисник бира тренинг. (АПУСО)
6. Корисник позива систем да обрише податке о тренингу. (АПСО)
7. Систем брише податке о тренингу. (СО)
8. Систем приказује кориснику поруку: „Систем је обрисао тренинг“. (ИА)



Слика 65: Порука система о успешном брисању тренинга

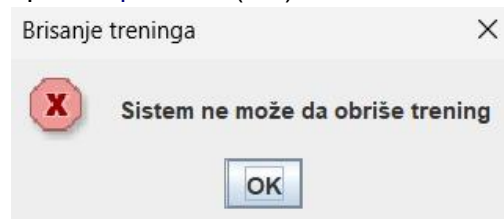
## Алтернативна сценарија

- 4.1. Уколико систем не може да нађе податке о тренингу, он приказује кориснику поруку: „Систем не може да нађе тренинге по задатој вредности“. (ИА)



Слика 66: Порука система о неуспешном проналажењу тренинга по задатој вредности

- 8.1. Уколико систем не може да обрише податке о тренингу, он приказује кориснику поруку: „Систем не може да обрише тренинг“. (ИА)



Слика 67: Порука система о неуспешном брисању тренинга



## СК8: Случај коришћења – Претраживање тренинга

### Назив СК

Претраживање тренинга

### Актори СК

Корисник

### Учесници СК

Корисник и систем (програм)

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

The screenshot shows a window titled 'Pretraga treninga'. At the top, there is a search bar labeled 'Naziv treninga:' with a text input field and two buttons: 'Pretrazi' and 'Osvesi'. Below this is a table with four columns: 'naziv', 'trajanje', 'beleska', and 'korisnik'. The table contains the following data:

naziv	trajanje	beleska	korisnik
Push	65	Grudi, triceps i ramena	Filip
Pull	57	Ledja i biceps	Filip
Legs	131	Noge	Filip
Abs	33	Trbusnjaci	Petar
Cardio	15	Trcanje i stepenice	Marko
Upper	45	Gornji deo tela	Marko
Lower	55	Donji deo tela	Marko
Full body	150	Celo telo	Petar

At the bottom of the window, there are three buttons: 'Izmeni', 'Dodaj', and 'Obriši'.

Слика 68: Форма за приказ свих тренинга

## Основни сценарио СК

1. Корисник уноси вредност по којој претражује тренинге. (АПУСО)

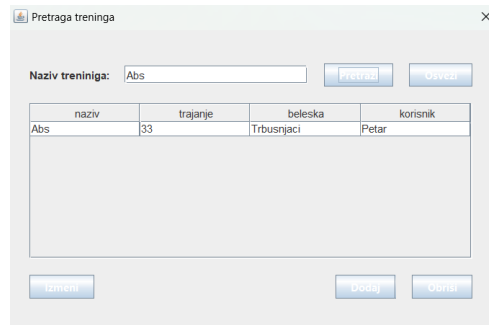
This screenshot is similar to the previous one, but the text 'Abs' has been entered into the 'Naziv treninga:' search bar. The table and buttons remain the same.

Слика 69. Попуњена вредност по којој се претражују тренинзи

2. Корисник позива систем да нађе тренинге по задатој вредности. (АПСО)
3. Систем тражи тренинге по задатој вредности. (СО)
4. Систем приказује кориснику тренинге и поруку: "Систем је нашао тренинге по задатој вредности". (ИА)

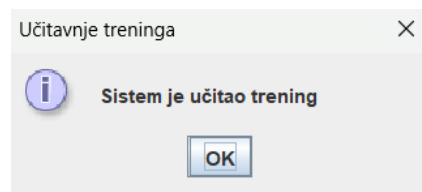
The screenshot shows a small dialog box titled 'Pretraga treninga'. It contains an information icon (i) and the text 'Sistem je našao treninge po zadatoj vrednosti'. At the bottom is an 'OK' button.

Слика 70: Порука система о успешном проналазку тренинга по задатој вредности

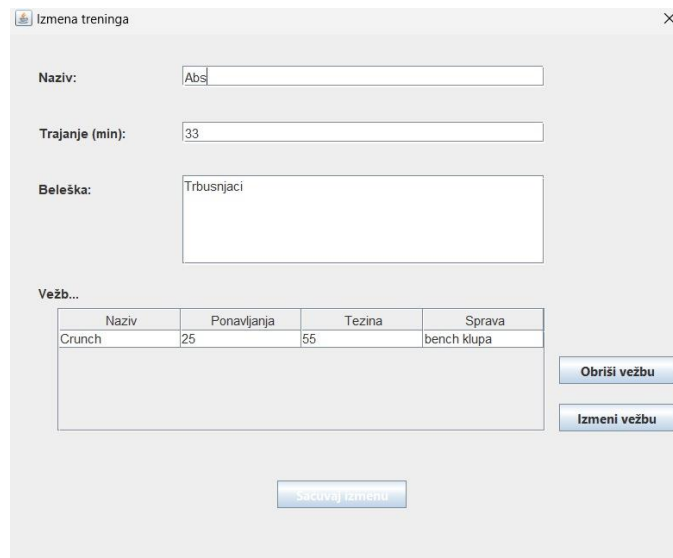


Слика 71: Приказ тренинга по задатој вредности

5. Корисник бира жељени тренинг. (АПУСО)
6. Корисник позива систем да учита тренинг. (АПСО)
7. Систем учитава тренинг. (СО)
8. Систем приказује кориснику податке о тренингу и поруку: „Систем је прочитао тренинг “. (ИА)



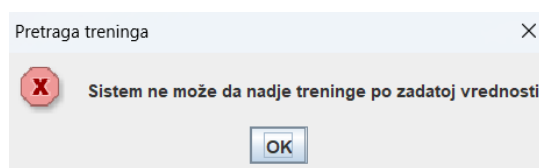
Слика 72: Порука система о успешном учитавању тренинга



Слика 73: Учитани тренинге

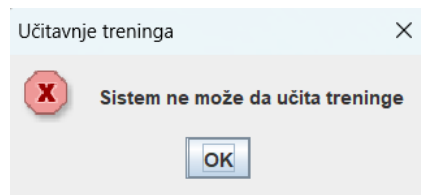
## Алтернативна сценарија

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



Слика 74: Неуспешно пронађени тренинзи по задатој вредности

8.1. Уколико **систем** не може да учита **тренинг** он приказује **кориснику** поруку “**Систем** не може да учита **тренинг**”. (ИА)



Слика 75: Неупешно учитани тренинзи

### 3.1.2. Пројектовање контролера корисничког интерфејса

Контролер корисничког интерфејса има улогу да:

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

### 3.2. Пројектовање апликационе логике

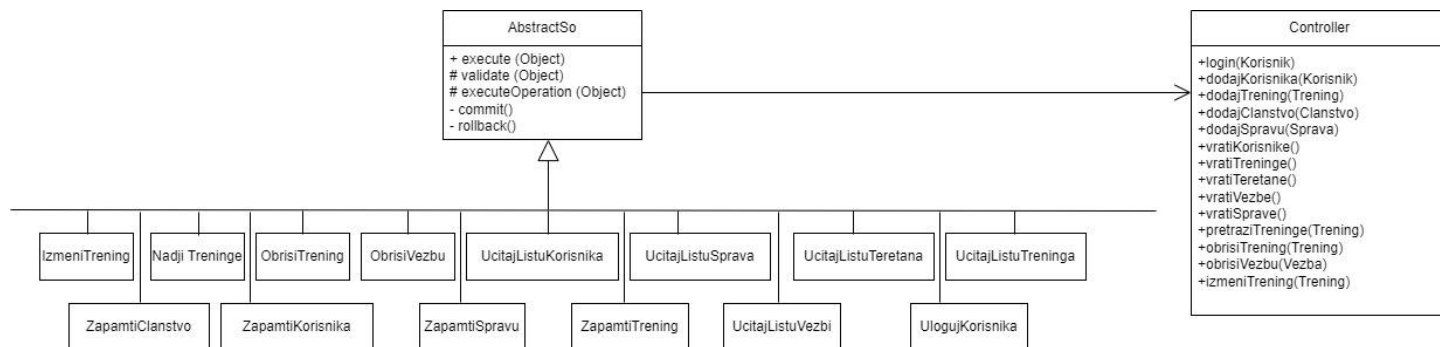
Апликациона логика описује структуру и понашање система. Апликациони сервер се састоји из:

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

#### 3.2.1. Контролер апликационе логике

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

Свака системска операција имплементирана је као засебна класа.



Слика 76: Пројектовање контролера апликационе логике

### 3.2.2. Пословна логика

Пословна логика је описана структуром (доменским класама) и понашањем (системским операцијама).

#### 3.2.2.1. Пројектовање понашања софтверског система – системске операције

За сваку системску операцију потребно је направити концептуална решења која су директно повезана са логиком проблема. За сваки уговор пројектује се концептуално решење.

Уговор УГ1: ZapamtiKorisnika

**Операција:** ZapamtiKorisnika (Korisnik) сигнал

**Веза са СК:** СК1

**Предуслови:** Структурна и вредносна ограничења над објектом *Korisnik* треба да буду задовољена

**Постуслови:** Нови корисник је регистрован



Слика 77: Дијаграм секвенци за уговор ZapamtiKorisnika

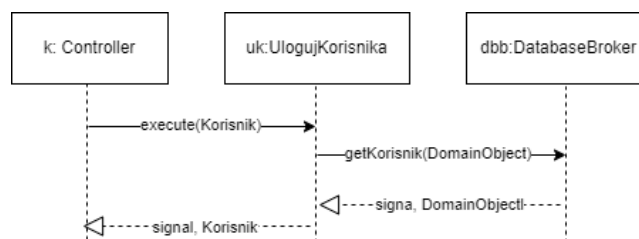
Уговор УГ2: UlogujKorisnika

**Операција:** UlogujKorisnika (Korisnik) сигнал

**Веза са СК:** СК2

**Предуслови:** /

**Постуслови:** Корисник је пријављен на систем



Слика 78: Дијаграм секвенци за уговор UlogujKorisnika

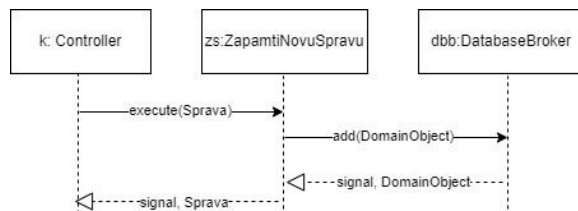
#### Уговор УГ3: ZapamtiNovuSpravu

**Операција:** ZapamtiNovuSpravu (Sprava) сигнал

**Веза са СК:** СК3

**Предуслови:** Структурна и вредносна ограничења над објектом Sprava треба да буду задовољена

**Постуслови:** Sprava је запамћена



Слика 79: Дијаграм секвенци за уговор ZapamtiNovuSpravu

#### Уговор УГ4: ZapamtiClanstvo

**Операција:** ZapamtiClanstvo (Clanstvo) сигнал

**Веза са СК:** СК4

**Предуслови:** Структурна и вредносна ограничења над објектом Clanstvo треба да буду задовољена

**Постуслови:** Clanstvo је запамћено



Слика 80: Дијаграм секвенци за уговор ZapamtiClanstvo

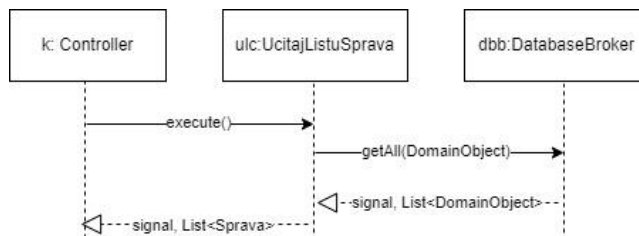
#### Уговор УГ5: UcitajListuSprava

**Операција:** UcitajListuSprava (List<Sprava>) сигнал

**Веза са СК:** СК5, СК6

**Предуслови:** /

**Постуслови:** /



Слика 81: Дијаграм секвенци за уговор UcitajListuSprava

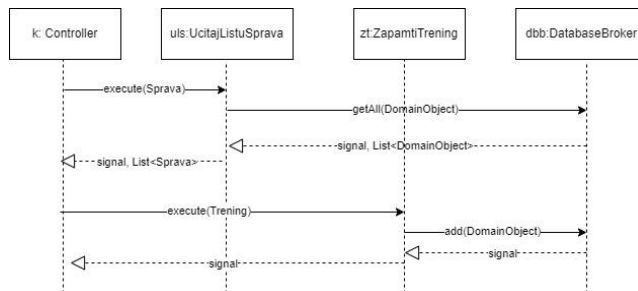
### Уговор УГ6: KreirajTrening

**Операција:** KreirajTrening(Trening) сигнал

**Веза са СК:** СК5

**Предуслови:** Структурна и вредносна ограничења над објектом *Trening* треба да буду задовољена

**Постуслови:** Trening је запамћен



Слика 82: Дијаграм секвенци за уговор *KreirajTrening*

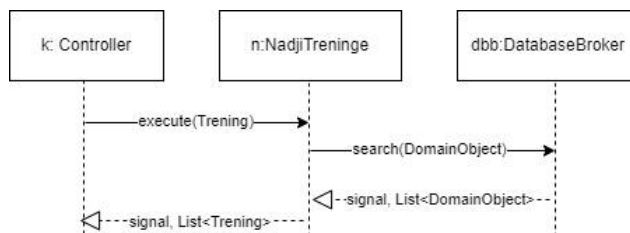
### Уговор УГ7: NadjiTreninge

**Операција:** NadjiTreninge(KriterijumPretrage, List<Trening>) сигнал

**Веза са СК:** СК6, СК7, СК8

**Предуслови:** /

**Постуслови:** /



Слика 83: Дијаграм секвенци за уговор *NadjiTreninge*

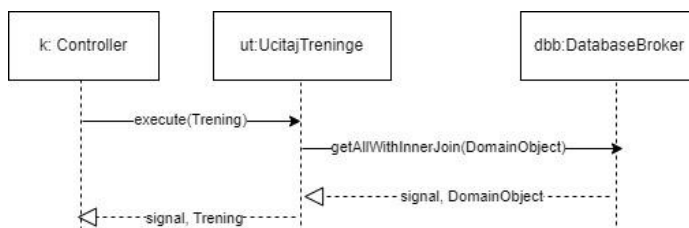
### Уговор УГ8: UcitajTrening

**Операција:** UcitajTrening(Trening) сигнал

**Веза са СК:** СК6, СК7, СК8

**Предуслови:** /

**Постуслови:** /



Слика 84: Дијаграм секвенци за уговор *UcitajTrening*

#### Уговор УГ9: IzmeniTrening

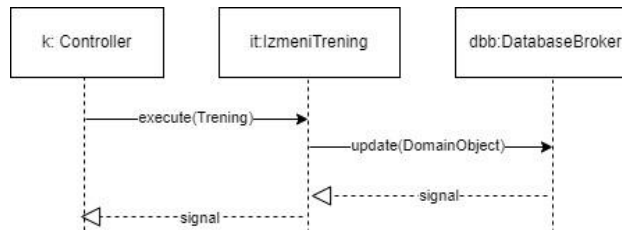
**Операција:** IzmeniTrening(Trening сигнал

**Веза са СК:** СК6

**Предуслови:** Структурна и вредносна ограничења над објектом Trening

треба да буду задовољена

**Постуслови:** Подаци о Treningу су измењени



Слика 85: Дијаграм секвенци за уговор IzmeniTrening

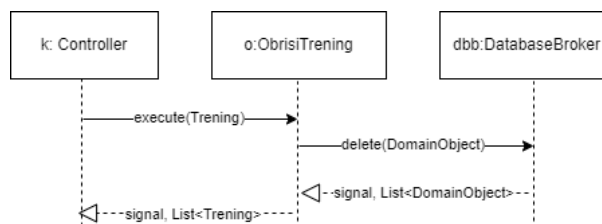
#### Уговор УГ10: ObrisiTrening

**Операција:** ObrisiTrening (Trening) сигнал

**Веза са СК:** СК7

**Предуслови:** Структурна ограничења над објектом Trening треба да буду задовољена

**Постуслови:** Trening је обрисан



Слика 86: Дијаграм секвенци за уговор ObrisiTrening

Класе које су одговорне за извршење системских операција наслеђују класу *AbstractSO* како би могле да се повежу са базом података. *AbstractSO* представља апстрактну класу чија главна метода (*execute()*) у себи садржи валидацију, извршавање операције, потврду у бази уколико је извршење операције успешно, као и поништавање операције уколико извршење операције није било успешно. Свака од системских операција даје своју имплементацију метода за валидацију и извршење конкретне системске операције.



### 3.2.2.2. Пројектовање структуре софтверског система - доменске класе

Софтверске класе структуре:

```
public class Korisnik extends DomainObject implements Serializable{
    private int IDKorisnika;
    private String mejl;
    private String lozinka;
    private String ime;
    private String prezime;

    public Korisnik() {
    }

    public Korisnik(int IDKorisnika, String mejl, String lozinka, String ime, String prezime) {
        this.IDKorisnika = IDKorisnika;
        this.mejl = mejl;
        this.lozinka = lozinka;
        this.ime = ime;
        this.prezime = prezime;
    }
}
```

Слика 87: Доменска класа Корисник

```
public class Vezba extends DomainObject implements Serializable{

    private int IDVezbe;
    private String naziv;
    private int ponavljanja;
    private int tezina;
    private Sprava IDSprave;
    private Trening IDTreninga;

    public Vezba() {
    }

    public Vezba(int IDVezbe, String naziv, int ponavljanja, int tezina, Sprava IDSprave, Trening IDTreninga) {
        this.IDVezbe = IDVezbe;
        this.naziv = naziv;
        this.ponavljanja = ponavljanja;
        this.tezina = tezina;
        this.IDSprave = IDSprave;
        this.IDTreninga = IDTreninga;
    }
}
```

Слика 88: Доменска класа Вежба

```

public class Trening extends DomainObject implements Serializable{

    private int IDTreninga;
    private String naziv;
    private int trajanje;
    private String beleska;
    private Korisnik IDKorisnika;
    private List<Vezba> vezbe = new ArrayList<>();

    public Trening() {
    }

    public Trening(int IDTreninga, String naziv, int trajanje, String beleska, Korisnik IDKorisnika) {
        this.IDTreninga = IDTreninga;
        this.naziv = naziv;
        this.trajanje = trajanje;
        this.beleska = beleska;
        this.IDKorisnika = IDKorisnika;
    }
}

```

Слика 89: Доменска класа Тренинг

```

public class Sprava extends DomainObject implements Serializable{

    private int IDSprave;
    private String naziv;
    private double maxTezina;

    public Sprava() {
    }

    public Sprava(int IDSprave, String naziv, double maxTezina) {
        this.IDSprave = IDSprave;
        this.naziv = naziv;
        this.maxTezina = maxTezina;
    }
}

```

Слика 90: Доменска класа Справа

```

public class Teretana extends DomainObject implements Serializable{

    private int IDTeretane;
    private String naziv;
    private String kontakt;
    private String mesto;
    private String adresa;

    public Teretana() {
    }

    public Teretana(int IDTeretane, String naziv, String kontakt, String mesto, String adresa) {
        this.IDTeretane = IDTeretane;
        this.naziv = naziv;
        this.kontakt = kontakt;
        this.mesto = mesto;
        this.adresa = adresa;
    }
}

```

Слика 91: Доменска класа Теретана

```

public class Clanstvo extends DomainObject implements Serializable{

    private Teretana IDTeretane;
    private Korisnik IDKorinsika;
    private LocalDate datumOd;
    private LocalDate datumDo;
    private int cena;

    public Clanstvo() {
    }

    public Clanstvo(Teretana IDTeretane, Korisnik IDKorinsika, LocalDate datumOd, LocalDate datumDo, int cena) {
        this.IDTeretane = IDTeretane;
        this.IDKorinsika = IDKorinsika;
        this.datumOd = datumOd;
        this.datumDo = datumDo;
        this.cena = cena;
    }
}

```

Слика 92: Доменска класа Чланство

Поред ових додате су и следеће класе:

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

```

public class Request implements Serializable{
    private Operation operation;
    private Object argument;

    public Request(Operation operation, Object argument) {
        this.operation = operation;
        this.argument = argument;
    }

    public Request() {
    }
}

```

Слика 93: Класа Request

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

```
public class Response implements Serializable{
    private Object result;
    private Exception exception;

    public Response() {
    }

    public Response(Object result, Exception exception) {
        this.result = result;
        this.exception = exception;
    }
}
```

Слика 94: Класа Response

**Receiver** – служи за комуникацију између клијентске и серверске стране. Служи за примање порука.

```
public class Receiver {
    private Socket socket;

    public Receiver(Socket socket) {
        this.socket = socket;
    }

    public Object receive() throws Exception{
        try {
            ObjectInputStream in=new ObjectInputStream(in:socket.getInputStream());
            return in.readObject();
        } catch (IOException ex) {
            ex.printStackTrace();
            throw new Exception("Error receiving object!\n"+ex.getMessage());
        }
    }
}
```

Слика 95: Класа Receiver

**Sender** - служи за комуникацију између клијентске и серверске стране. Служи за слање порука.

```
public class Sender {
    private Socket socket;

    public Sender(Socket socket) {
        this.socket = socket;
    }

    public void send(Object object) throws Exception{
        try {
            ObjectOutputStream out=new ObjectOutputStream(out:socket.getOutputStream());
            out.writeObject(obj:object);
            out.flush();
        } catch (IOException ex) {
            ex.printStackTrace();
            throw new Exception("Error sending object: \n"+ex.getMessage());
        }
    }
}
```

Слика 96: Класа Sender

**Operation** – класа која садржи све операције које се шаљу од клијента ка серверу

```
public enum Operation {  
    LOGIN,  
    LOGOUT,  
    DODAJ_KORISNIKA,  
    DODAJ_TRENING,  
    DODAJ_CLANSTVO,  
    DODAJ_SPRAVU,  
    VRATI_SVE_TERETANE,  
    VRATI_SVE_VEZBE,  
    VRATI_SVE_KORISNIKE,  
    VRATI_SVE_TRENINGE,  
    VRATI_SVE_SPRAVE,  
    OBRISI_TRENING,  
    OBRISI_VEZBU,  
    IZMENI_TRENING,  
    PRETRAZI_TRENING,  
}
```

Слика 97: Класа Operation

### 3.2.3. Пројектовање брокера базе података

Класа DatabaseBroker је класа која посредује у свим операцијама над базом података и реализује следеће методе:

```
public class DatabaseBroker {  
    private final Connection connection;  
  
    public DatabaseBroker(Connection connection) {  
        this.connection = connection;  
    }  
  
    public Connection getConnection() {  
        return connection;  
    }  
  
    public List<DomainObject> getAll(DomainObject domainObject) throws Exception {...21 lines }  
    public List<DomainObject> getAllWithInnerJoin(DomainObject domainObject) throws Exception {...21 lines }  
    public boolean update(DomainObject domainObject) throws SQLException {...33 lines }  
    public void delete(DomainObject domainObject) throws SQLException {...20 lines }  
    public DomainObject getKorisnik(DomainObject domainObject) throws Exception {...23 lines }  
    public boolean add(DomainObject domainObject) throws SQLException {...31 lines }  
    public List<DomainObject> search(DomainObject domainObject) throws Exception {...23 lines }  
}
```

Слика 98: Класа DatabaseBroker

Све методе класе **DatabaseBroker** су пројектоване као генеричке, што значи да могу да прихвате различите доменске објекте преко параметара. Ово је остварено дефинисањем апстрактне методе **DomainObject** коју имплементирају све доменске класе.

```
public abstract class DomainObject {

    public abstract String getTableName();

    public abstract String getColumnsForInsert();

    public abstract String getParamsForInsert();

    public abstract String getParamsForUpdate();

    public abstract void setParamsForInsert(PreparedStatement statement, DomainObject domainObject) throws SQLException;

    public abstract void setParamsForUpdate(PreparedStatement statement, DomainObject domainObject) throws SQLException;

    public abstract void setParamsForDelete(PreparedStatement statement, DomainObject domainObject) throws SQLException;

    public abstract void setAutoIncrementPrimaryKey(int primaryKey);

    public abstract String getParamsForDelete();

    public abstract String getColumnsForSelect() throws SQLException;

    public abstract List<DomainObject> getParamsForSelect(ResultSet rs) throws SQLException;

    public abstract String getInnerJoin() throws SQLException;

    public abstract String getTableName2();

    public abstract String getParamsForSelect1();

    public abstract void setParamsForSelect(PreparedStatement statement, DomainObject domainObject) throws SQLException;

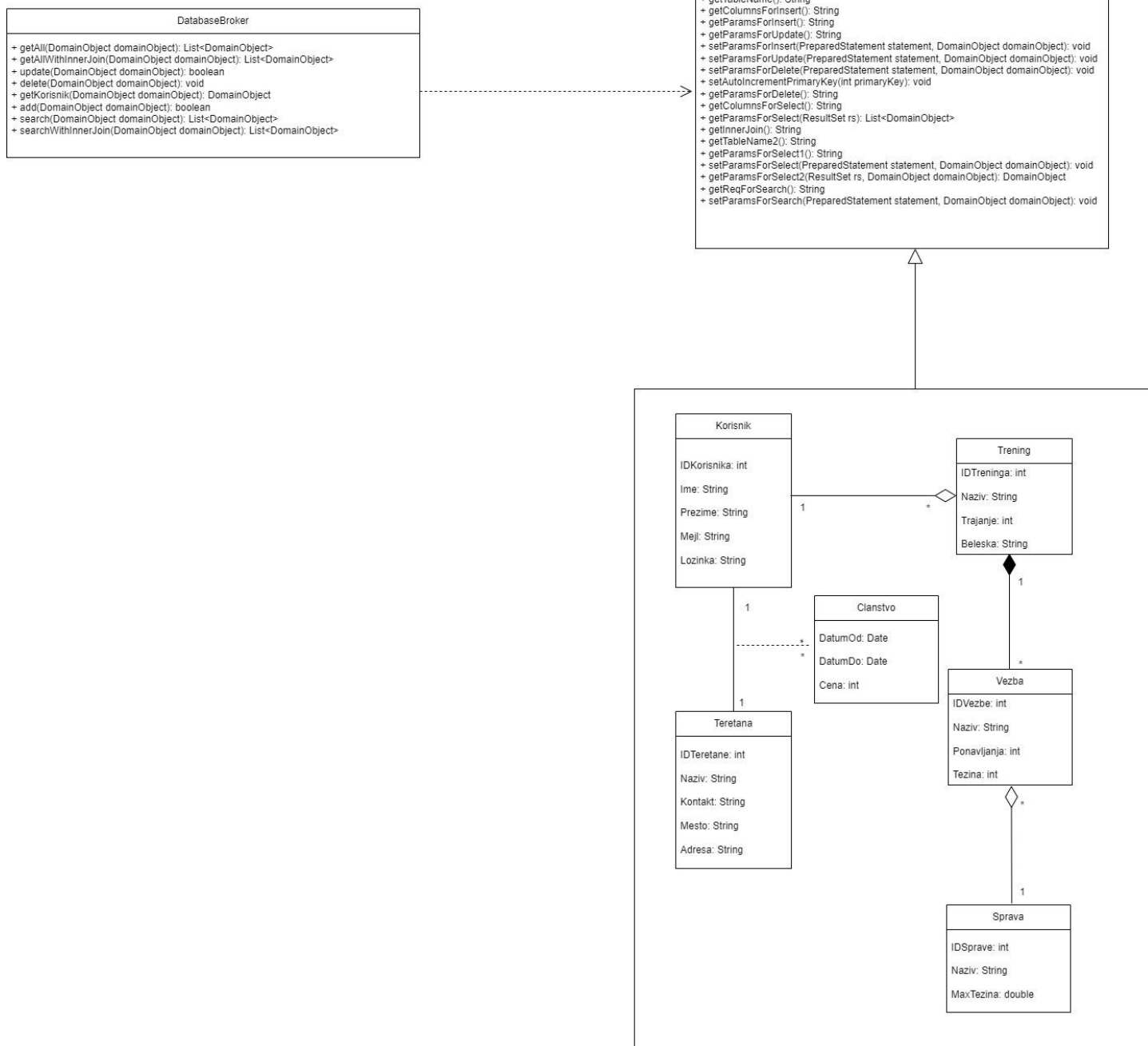
    public abstract DomainObject getParamsForSelect2(ResultSet rs, DomainObject domainObject) throws SQLException;

    public abstract String getReqForSearch();

    public abstract void setParamsForSearch(PreparedStatement statement, DomainObject domainObject) throws SQLException;
```

Слика 99: Апстрактна класа *DomainObject*





Слика 100: Веза између брокера и општег доменског објекта *DomainObject*

### 3.3. Пројектовање складишта података

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

Табела Корисник:

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	ZeroFill?	On Update	Comment	Virtuality	Expression	Check Constraint
<input type="checkbox"/>	IDKorisnika	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/>	ime	varchar	30		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/>	prezime	varchar	30		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/>	mej1	varchar	40		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/>	lozinka	varchar	30		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				

Слика 101: Табела Корисник у бази података

Табела Вежба:

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	ZeroFill?	On Update	Comment	Virtuality	Expression	Check Constraint
<input type="checkbox"/>	IDVezbe	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/>	naziv	varchar	30		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/>	ponavljanja	int	11		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/>	tezina	int	11		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/>	IDSprave	int	11		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/>	IDTreninga	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		

Слика 102: Табела Вежба у бази података

Табела Тренинг:

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	ZeroFill?	On Update	Comment	Virtuality	Expression	Check Constraint
<input type="checkbox"/>	IDTreninga	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/>	naziv	varchar	30		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/>	trajanje	int	11		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/>	beleska	varchar	150		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/>	IDKorisnika	int	11		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				

Слика 103: Табела Тренинг у бази података

Табела Справа:

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	ZeroFill?	On Update	Comment	Virtuality	Expression	Check Constraint
<input type="checkbox"/>	IDSprave	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/>	naziv	varchar	30		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/>	maxTezina	double			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		

Слика 104: Табела Справа у бази података

Табела Теретана:

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	ZeroFill?	On Update	Comment	Virtuality	Expression	Check Constraint
<input type="checkbox"/>	IDTeretane	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/>	naziv	varchar	30		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/>	kontakt	varchar	30		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/>	mesto	varchar	30		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		
<input type="checkbox"/>	adresa	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		(none)		

Слика 105: Табела Теретана у бази података

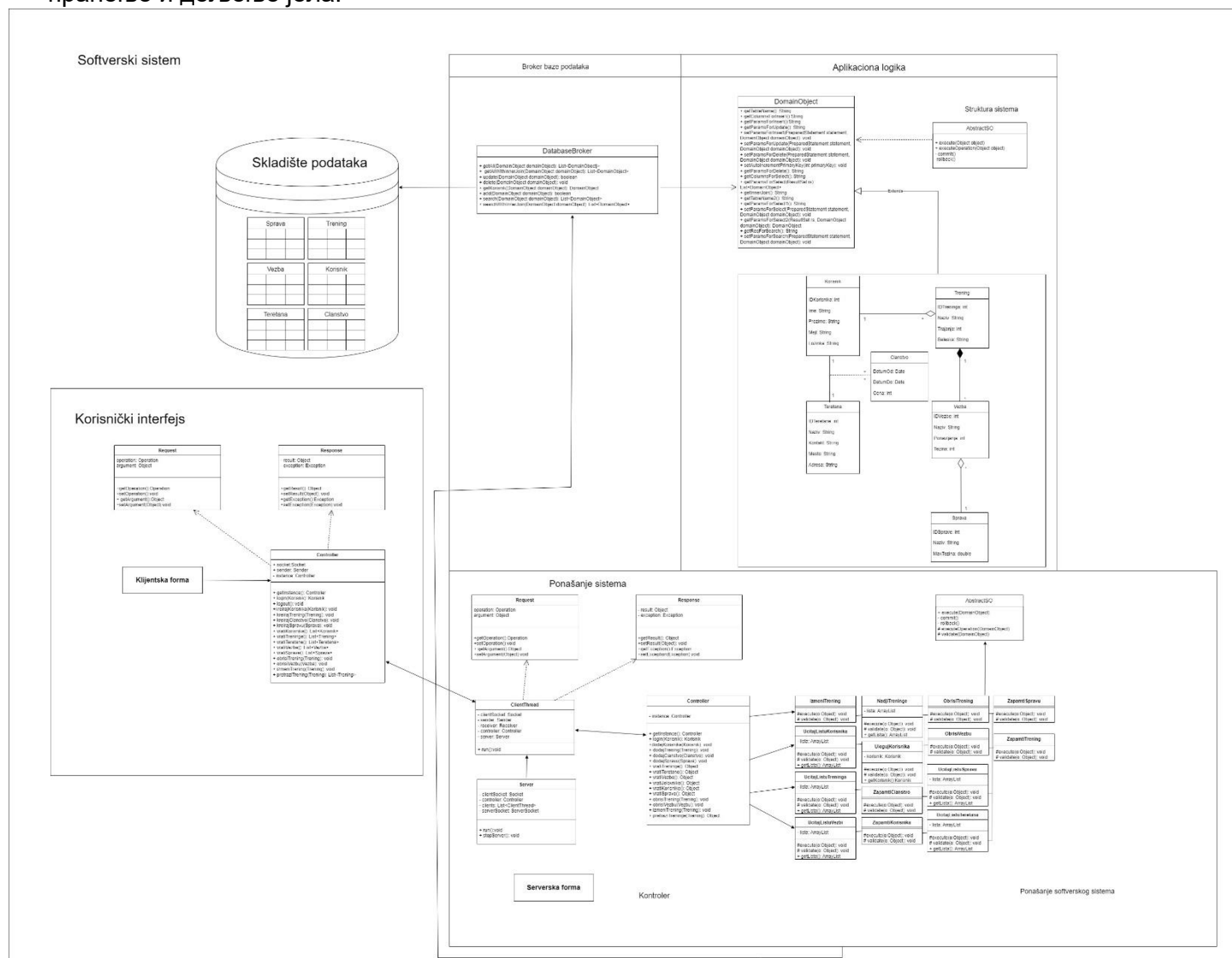
Табела Чланство:

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

Слика 106: Табела Чланство у бази података



На основу претходних целина, може се саставити цела архитектура софтверског система за праћење и дељење јела.

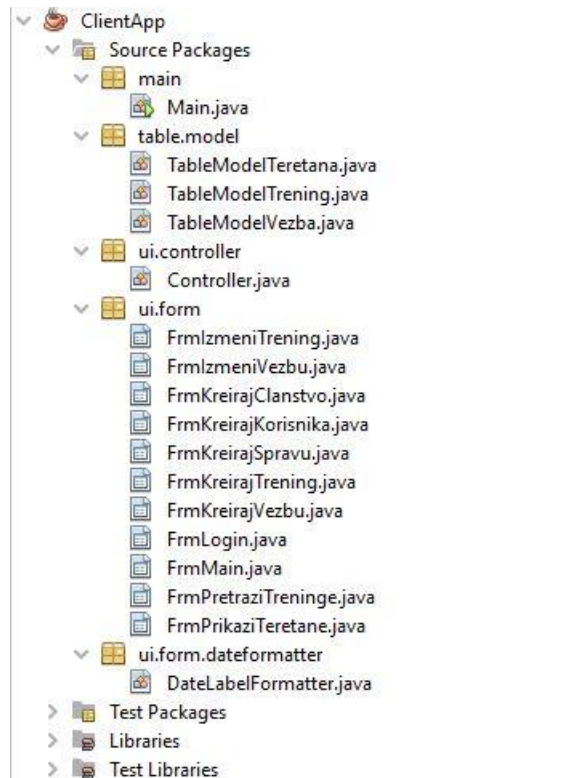


Слика 107: Комплетна архитектура софтверског система

## 4. Фаза имплементација

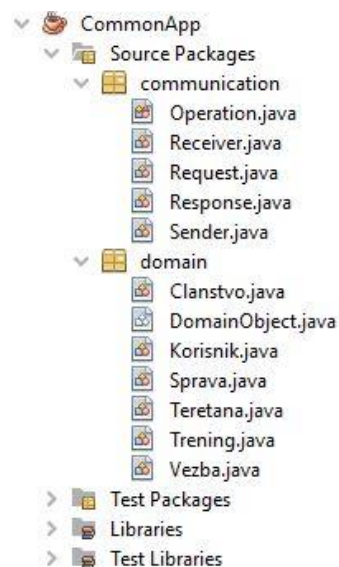
Софтверски систем је развијен у програмском језику „Java“ . Систем је пројектован као клијент-серверапликација. Као систем за управљање базом података коришћен је MySQL, док је развојно окружење „NetBeans IDE 15“. На основу архитектуре софтверског система добијене су следеће софтверске класе:

ClientApp:



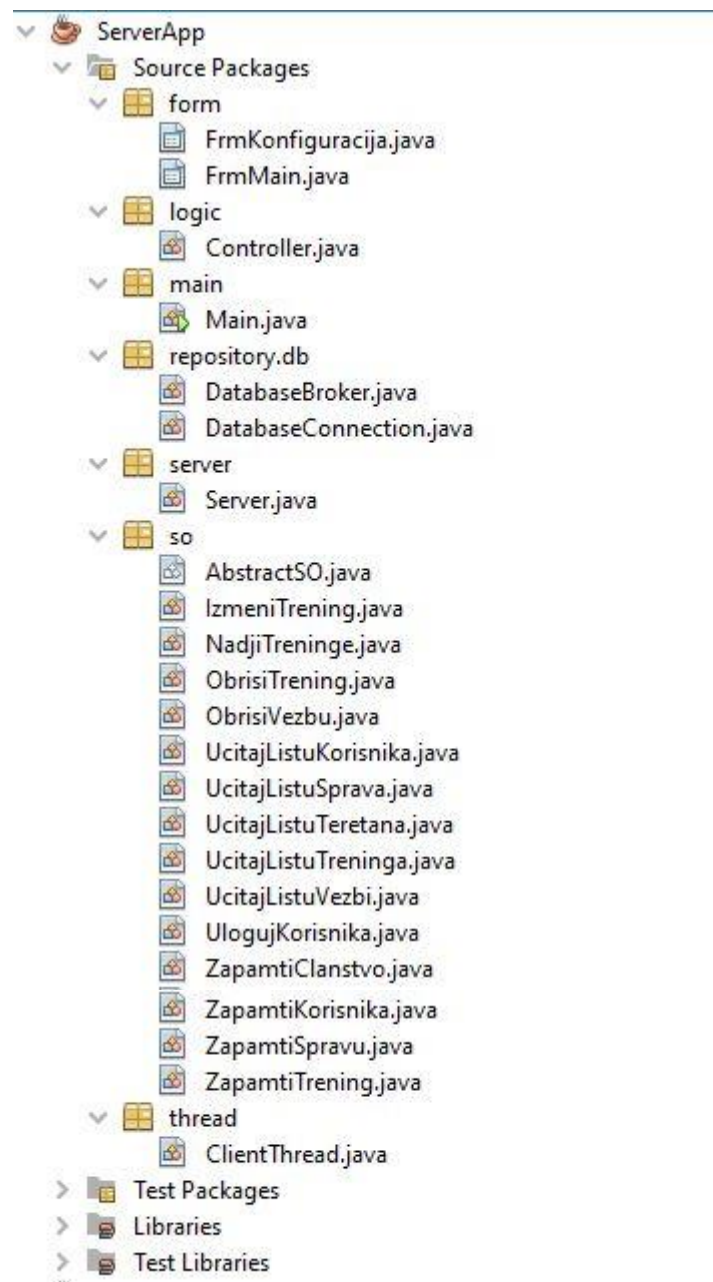
Слика 108: Приказ ClientApp дела апликације

CommonApp:



Слика 109: Приказ CommonApp дела апликације

ServerApp:



Слика 110: Приказ ServerApp дела апликације

## 5. Фаза тестирања

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

## 6. Литература

- Влајић, С. (2020). Пројектовање софтвера (скрипта). Београд, Србија: Факултет организационих наука