

Универзитет у Београду
Факултет организационих наука
Лабораторија за софтверско инжењерство



Семинарски рад
Предмет: Пројетовање софтвера
Тема: Софтверски систем за теретану

Ментор:
проф. др Синиша Влајић

Студент:
Јована Цветојевић 1013/16

Београд, 2018.

Садржај

Вербални модел система.....	1
Случајеви коришћења	2
Дијаграм случајева коришћења.....	2
СК1: Случај коришћења – Унос тренинга.....	3
СК2: Случај коришћења – Претрага тренинга	4
СК3: Случај коришћења – Измена тренинга	5
СК4: Случај коришћења – Унос корисника.....	6
СК5: Случај коришћења – Претрага корисника.....	7
СК6: Случај коришћења – Измена корисника.....	8
СК7: Случај коришћења – Унос одабраног тренинга	9
СК8: Случај коришћења – Претрага одабраног тренинга.....	10
Анализа.....	10
Системски дијаграми секвенци.....	10
ДС1: Дијаграм секвенци – Унос тренинга.....	11
ДС2: Дијаграм секвенци – Претрага тренинга	12
ДС3: Дијаграм секвенци – Измена тренинга	14
ДС4: Дијаграм секвенци – Унос корисника	17
ДС5: Дијаграм секвенци – Претрага корисника.....	18
ДС6: Дијаграм секвенци – Измена корисника.....	20
ДС7: Дијаграм секвенци – Унос одабраног тренинга	23
ДС8: Дијаграм секвенци – Претрага одабраног тренинга	25
Списак свих системских опрација	27
Дефинисње уговора о системским операцијама	27
Уговор УГ1: KreirajTrening(Trening)	27
Уговор УГ2: PronadjiTreninge(ZadataVrednost, List<Trening>).....	27
Уговор УГ3: UcitajTrening(Trening).....	27
Уговор УГ4: PromeniTrening(Trening)	28
Уговор УГ5: KreirajKorisnika(Korisnik)	28
Уговор УГ6: PronadjiKorisnike(ZadataVrednost, List<Korisnik>)	28
Уговор УГ7: UcitajKorisnika(Korisnik).....	28
Уговор УГ8: PromeniKorisnika(Korisnik)	28

Уговор УГ9: VratilistuKorisnika(List<Korisnik>).....	28
Уговор УГ10: VratilistuTreninga(List<Trening>).....	29
Уговор УГ11: KreirajOdabraniTrening(OdabraniTrening).....	29
Уговор УГ12: PronadjiOdabraniTrening(ZadataVrednost, List <OdabraniTrening>).....	29
Уговор УГ13: UcitajOdabraniTrening(OdabraniTrening)	29
Концептуални модел	30
Релациони модел	30
Пројектовање.....	34
Архитектура софтверског система.....	34
Пројектовање корисничког интерфејса	34
СК1: Случај коришћења – Унос тренинга.....	39
СК2: Случај коришћења – Претрага тренинга	41
СК3: Случај коришћења – Измена тренинга	44
СК4: Случај коришћења – Унос корисника.....	47
СК5: Случај коришћења – Претрага корисника	49
СК6: Случај коришћења – Измена корисника.....	52
СК7: Случај коришћења – Унос одабраног тренинга	55
СК8: Случај коришћења – Претрага одабраног тренинга.....	57
Пројектовање контролера корисничког интерфејса.....	60
Пројектовање апликационе логике	65
Комуникација са клијентом	65
Контролер апликационе логике	68
Пројектовање структуре софтверског система.....	71
Пројектовање системских операција	75
Уговор УГ1: KreirajTrening(Trening)	75
Уговор УГ2: PronadjiTreninge(ZadataVrednost, List<Trening>).....	75
Уговор УГ3: UcitajTrening(Trening).....	75
Уговор УГ4: PromeniTrening(Trening)	76
Уговор УГ5: KreirajKorisnika(Korisnik)	76
Уговор УГ6: PronadjiKorisnike(ZadataVrednost, List<Korisnik>)	76
Уговор УГ7: UcitajKorisnika(Korisnik).....	77
Уговор УГ8: PromeniKorisnika(Korisnik)	77
Уговор УГ9: VratilistuKorisnika(List<Korisnik>).....	77

Уговор УГ10: VratiListuTreninga(List<Trening>)	78
Уговор УГ11: KreirajOdabraniTrening(OdabraniTrening)	78
Уговор УГ12: PronadjiOdabraniTrening(ZadataVrednost, List <OdabraniTrening>)	79
Уговор УГ13: UcitajOdabraniTrening(OdabraniTrening)	79
Брокер базе података	80
Пројектовање складишта података	82
Имплементација	83
Тестирање	85
Литература	85

Вербални модел система

Потребно је имплементирати део информационог система за теретану. Системом управља радник, који мора бити пријављен како би му се уопште омогућио приступ подацима. Уколико постоји потреба за додавањем нових радника то се може постићи регистрацијом. Процес почиње када корисник пристигне у теретану. Радник претражује кориснике и уколико не постоји корисник, радник га уноси у систем. Осим тога пријављени радник увек може додати нове кориснике, претраживати базу корисника и мењати њихове податке.

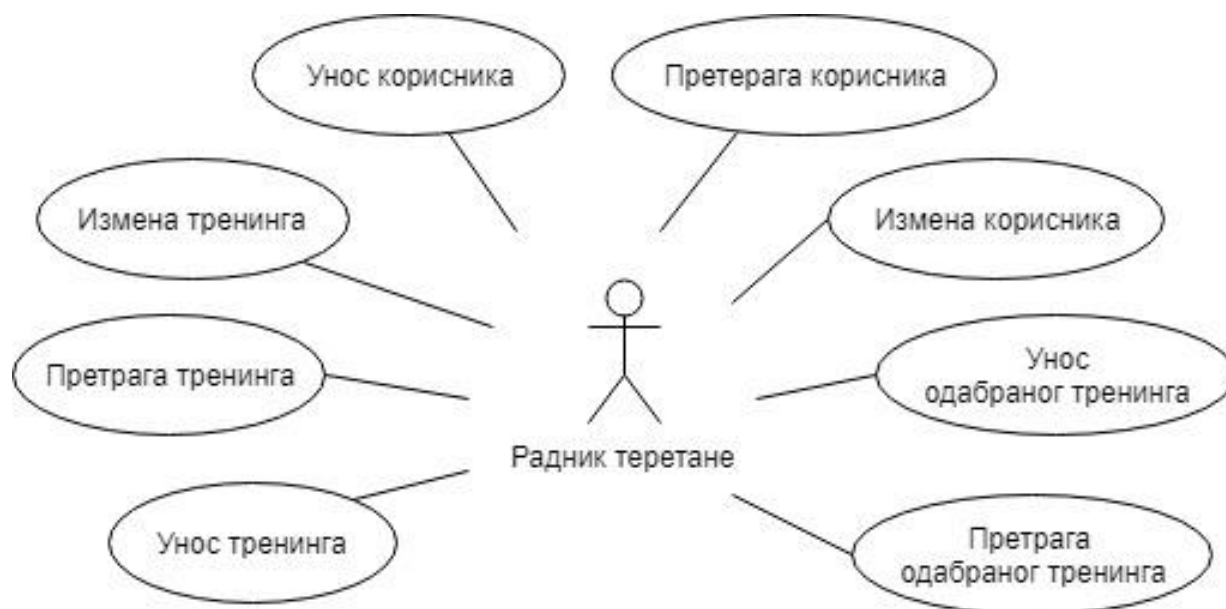
Након што радник унесе корисника у систем и он одабере жељен тренинг, врши се наплата и одабрани тренинг се уноси у систем. Потребно је водити евиденцију о датуму уплате, датуму докле корисник може да похађа одабрани тренинг, а самим тим и о статусу да ли је важећи или неважећи у зависности да ли му је истекао рок коришћења. За сваки одабрани тренинг бележе се термини, односно када се одабрани тренинзи одржавају, датум и време. Такође, када се оформи нова врста тренинга додаје се у систем.

Случајеви коришћења

У делу информационог система за теретану уочени су следећи случајеви коришћења:

1. Унос тренинга
2. Претрага тренинга
3. Измена тренинга
4. Унос корисника
5. Претрага корисника
6. Измена корисника
7. Унос одабраног тренинга (сложен случај коришћења)
8. Претрага одабраног тренинга (сложен случај коришћења)

Дијаграм случајева коришћења



Слика бр. 1. Графички приказ случајева коришћења

СК1: Случај коришћења – Унос тренинга

Назив СК

Креирање тренинга

Актори СК

Радник

Учесници СК

Радник и систем (програм)

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

Основни сценарио СК

1. Радник уноси податке у тренинг. (АПУСО)
2. Радник контролише да ли је коректно унео податке у тренинг. (АНСО)
3. Радник позива систем да креира тренинг. (АПСО)
4. Систем креира тренинг. (СО)
5. Систем приказује раднику креирани тренинг и поруку: “Систем је креирао тренинг“. (ИА)

Алтернативна сценарија

- 5.1 Уколико систем не може да креира тренинг он приказује раднику поруку “Систем не може да креира тренинг“. (ИА)

СК2: Случај коришћења – Претрага тренинга

Назив СК

Претрага тренинга

Актори СК

Радник

Учесници СК

Радник и систем (програм)

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

Основни сценарио СК

1. Радник уноси вредност по којој претражује тренинг. (АПУСО)
2. Радник позива систем да нађе тренинг по задатој вредности. (АПСО)
3. Систем тражи тренинг по задатој вредности. (СО)
4. Систем приказује раднику податке о тренинзима и поруку: “Систем је нашао тренинге по задатој вредности”. (ИА)
5. Радник бира тренинг. (АПУСО)
6. Радник позива систем да учита податке о тренингу. (АПСО)
7. Систем учитава податке о тренингу. (СО)
8. Систем приказује раднику податке о изабраном тренингу и поруку: “Систем је успешно учитао податке о тренингу”. (ИА)

Алтернативна сценарија

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

8.1 Уколико систем не може да нађе податке о изабраном тренингу он приказује поруку раднику: “Систем не може да учита податке о тренингу”. (ИА)

СК3: Случај коришћења – Измена тренинга

Назив СК

Промена тренинга

Актори СК

Радник

Учесници СК

Радник и систем (програм)

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

Основни сценарио СК

1. Радник уноси вредност по којој претражује тренинг. (АПУСО)
2. Радник позива систем да нађе тренинг по задатој вредности. (АПСО)
3. Систем тражи тренинг по задатој вредности. (СО)
4. Систем приказује раднику тренинге и поруку: “Систем је нашао тренинге по задатој вредности”. (ИА)
5. Радник бира тренинг који жели да измени. (АПУСО)
6. Радник позива систем да учита податке о изабраном тренингу. (АПСО)
7. Систем проналази податке о изабраном тренингу. (СО)
8. Систем приказује раднику податке о изабраном тренингу и поруку: “Систем је успешно пронашао податке о изабраном тренингу”. (ИА)
9. Радник уноси (мења) податке о тренингу. (АПУСО)
10. Радник контролише да ли је коректно унео податке о тренингу. (АНСО)
11. Радник позива систем да промени податке о тренингу. (АПСО)
12. Систем памти податке о тренингу. (СО)
13. Систем приказује раднику промењени тренинг и поруку: “Систем је променио тренинг.” (ИА)

Алтернативна сценарија

- 4.1 Уколико систем не може да нађе тренинге он приказује раднику поруку: “Систем не може да нађе тренинге по задатој вредности”. Прекида се извршење сценарија. (ИА)
- 8.1 Уколико систем не може да учита податке о тренингу он приказује раднику поруку “Систем не може да прикаже податке о изабраном тренингу”. Прекида се извршење сценарија. (ИА)
- 13.1 Уколико систем не може да запамти тренинг он приказује раднику поруку: “Систем не може да запамти тренинг”. (ИА)

СК4: Случај коришћења – Унос корисника

Назив СК

Креирање корисника

Актори СК

Радник

Учесници СК

Радник и систем (програм)

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

Основни сценарио СК

1. Радник уноси податке о кориснику. (АПУСО)
2. Радник контролише да ли је коректно унео податке о кориснику. (АНСО)
3. Радник позива систем да запамти податке о кориснику. (АПСО)
4. Систем креира корисника. (СО)
5. Систем приказује раднику корисника и поруку: “Систем је креирао корисника“. (ИА)

Алтернативна сценарија

- 5.1 Уколико систем не може да креира корисника он приказује раднику поруку “Систем не може да креира корисника“. (ИА)

СК5: Случај коришћења – Претрага корисника

Назив СК

Претрага корисника

Актори СК

Радник

Учесници СК

Радник и систем (програм)

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

Основни сценарио СК

1. Радник уноси вредност по којој претражује корисника. (АПУСО)
2. Радник позива систем да нађе кориснике по задатој вредности. (АПСО)
3. Систем тражи кориснике по задатој вредности. (СО)
4. Систем приказује раднику податке о корисницима и поруку: “Систем је нашао кориснике по задатој вредности”. (ИА)
5. Радник бира корисника. (АПУСО)
6. Радник позива систем да учита податке о изабраном кориснику. (АПСО)
7. Систем проналази податке о изабраном кориснику. (СО)
8. Систем приказује раднику податке о изабраном кориснику и поруку: “Систем је успешно пронашао податке о изабраном кориснику”. (ИА)

Алтернативна сценарија

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

8.1 Уколико систем не може да нађе податке о изабраном кориснику он приказује поруку раднику : “Систем не може да учита податке о кориснику”. (ИА)

СК6: Случај коришћења – Измена корисника

Назив СК

Промена корисника

Актори СК

Радник

Учесници СК

Радник и систем (програм)

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

Основни сценарио СК

1. Радник уноси вредност по којој претражује кориснике. (АПУСО)
2. Радник позива систем да нађе кориснике по задатој вредности. (АПСО)
3. Систем тражи кориснике по задатој вредности. (СО)
4. Систем приказује раднику кориснике и поруку: “Систем је нашао кориснике по задатој вредности”. (ИА)
5. Радник бира корисника ког жели да измени. (АПУСО)
6. Радник позива систем да учита податке о изабраном кориснику. (АПСО)
7. Систем проналази податке о изабраном кориснику. (СО)
8. Систем приказује раднику податке о изабраном кориснику и поруку: “Систем је успешно пронашао податке о изабраном кориснику”. (ИА)
9. Радник уноси (мења) податке о кориснику. (АПУСО)
10. Радник контролише да ли је коректно унео податке о кориснику. (АНСО)
11. Радник позива систем да промени податке о кориснику. (АПСО)
12. Систем памти податке о кориснику. (СО)
13. Систем приказује раднику промењеног корисника и поруку: “Систем је променио корисника.” (ИА)

Алтернативна сценарија

- 4.1 Уколико систем не може да нађе кориснике он приказује раднику поруку: “Систем не може да нађе кориснике по задатој вредности”. Прекида се извршење сценарија. (ИА)
- 8.1 Уколико систем не може да нађе податке о кориснику он приказује раднику поруку “Систем не може да пронађе податке о кориснику. Прекида се извршење сценарија. (ИА)
- 13.1 Уколико систем не може да промени корисника он приказује раднику поруку: “Систем не може да промени корисника”. (ИА)

СК7: Случај коришћења – Унос одабраног тренинга

Назив СК

Креирање одабраног тренинга

Актори СК

Радник

Учесници СК

Радник и систем (програм)

Предуслов: Систем је укључен и радник је пријављен под својом шифром. Систем приказује форму за рад са одабраним тренингом. Учитана је листа корисника и листа тренинга.

Основни сценарио СК

1. Радник уноси податке о одабраном тренингу. (АПУСО)
2. Радник контролише да ли је коректно унео податке у одабрани тренинг. (АНСО)
3. Радник позива систем да креира одабрани тренинг. (АПСО)
4. Систем креира одабрани тренинг. (СО)
5. Систем приказује раднику одабрани тренинг и поруку: “Систем је креирао одабрани тренинг”. (ИА)

Алтернативна сценарија

- 5.1 Уколико систем не може да креира одабрани тренинг он приказује раднику поруку: “Систем не може да креира одабрани тренинг”.

СК8: Случај коришћења – Претрага одабраног тренинга

Назив СК

Претрага одабраног тренинга

Актори СК

Радник

Учесници СК

Радник и систем (програм)

Предуслов: Систем је укључен и радник је пријављен под својом шифром. Систем приказује форму за рад са одабраним тренингом.

Основни сценарио СК

1. Радник уноси вредност по којој претражује све одабране тренинге. (АПУСО)
2. Радник позива систем да нађе све одабране тренинге по задатој вредности. (АПСО)
3. Систем тражи одабрани тренинге по задатој вредности. (СО)
4. Систем приказује раднику све одабране тренинге и поруку: “Систем је нашао све одабране тренинге по задатој вредности”. (ИА)
5. Радник бира одабрани тренинг коју жели да измени. (АПУСО)
6. Радник позива систем да учита податке о одабраном тренингу. (АПСО)
7. Систем проналази податке о том одабраном тренингу. (СО)
8. Систем приказује раднику податке о одабраном тренингу и поруку: “Систем је успешно пронашао податке о одабраном тренингу”. (ИА)

Алтернативна сценарија

- 4.1 Уколико систем не може да нађе одабране тренинге он приказује раднику поруку: “Систем не може да нађе одабране тренинге по задатој вредности”. Прекида се извршење сценарија. (ИА)
- 8.1 Уколико систем не може да запамти податке о одабраном тренингу он приказује раднику поруку “Систем не може да прикаже податке о одабраном тренингу ”. Прекида се извршење сценарија. (ИА)

Анализа

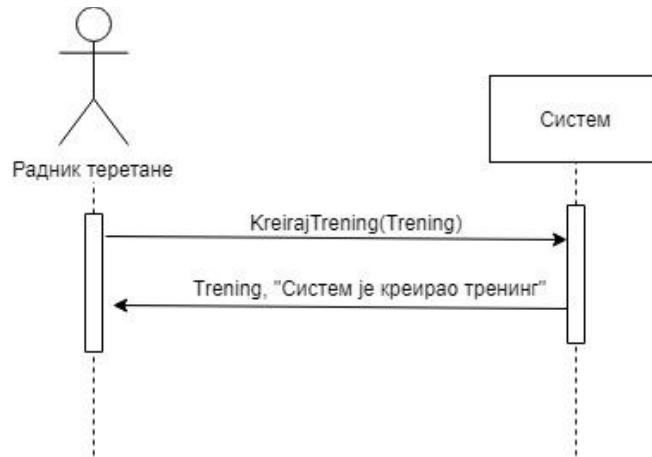
Системски дијаграми секвенци

Системски дијаграм секвенци прави се за сваки СК и он приказује догађаје у одређеном редоследу између актора, тј. радника теретане и софтверског система. Системским дијаграмима секвенци представљају се само догађаји који се праве АПСО акцијама и одговори система на те акције-ИА акције.

ДС1: Дијаграм секвенци – Унос тренинга

Основни сценарио

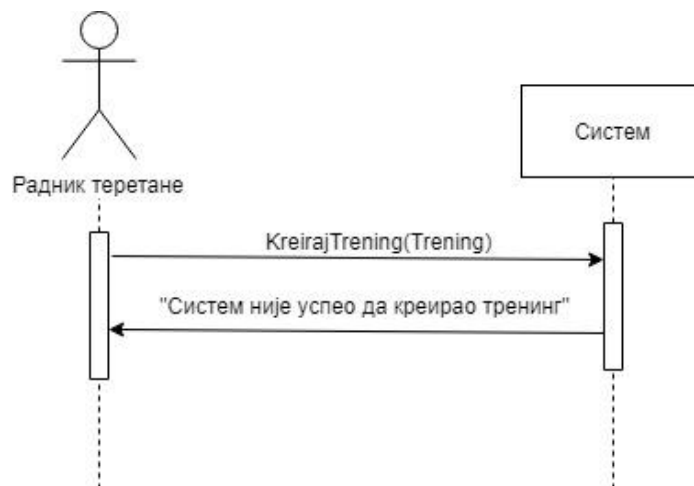
1. Радник позива систем да креира тренинг. (АПСО)
2. Систем приказује раднику креирани тренинг и поруку: “Систем је креирао тренинг“. (ИА)



Слика бр. 2. Дијаграм секвенци – унос тренинга, основни сценарио

Алтернативни сценарио бр.1.

- 2.1. Уколико систем не може да креира тренинг он приказује раднику поруку “Систем не може да креира тренинг“. (ИА)



Слика бр. 3. Дијаграм секвенци – систем не може да креира тренинг

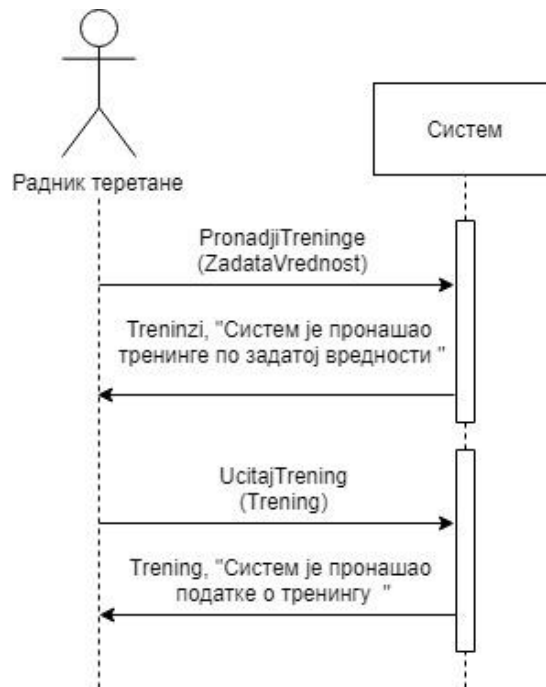
Листа системских операција:

- *signal* KreirajTrening(Trening)

ДС2: Дијаграм секвенци – Претрага тренинга

Основни сценарио

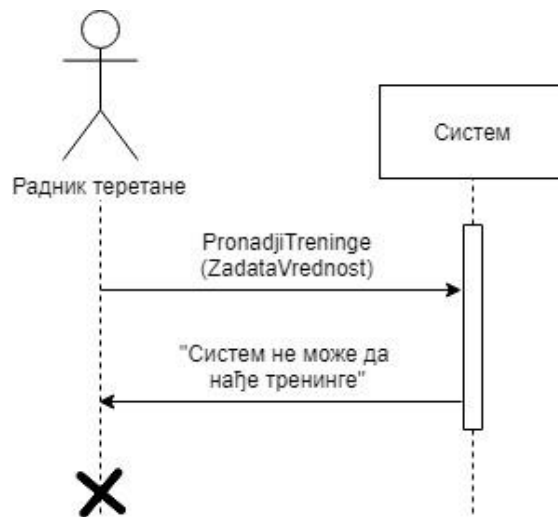
1. Радник **позива** систем да нађе тренинг по задатој вредности. (АПСО)
2. Систем приказује раднику податке о тренинзима и поруку: “Систем је нашао тренинге по задатој вредности”. (ИА)
3. Радник позива систем да учита податке о изабаном тренингу. (АПСО)
4. Систем приказује раднику податке о изабраном тренингу и поруку: “Систем је успешно прочитао податке о тренингу“. (ИА)



Слика бр. 4. Дијаграм секвенци – претрага тренинга, основни сценарио

Алтернативни сценарио бр.1.

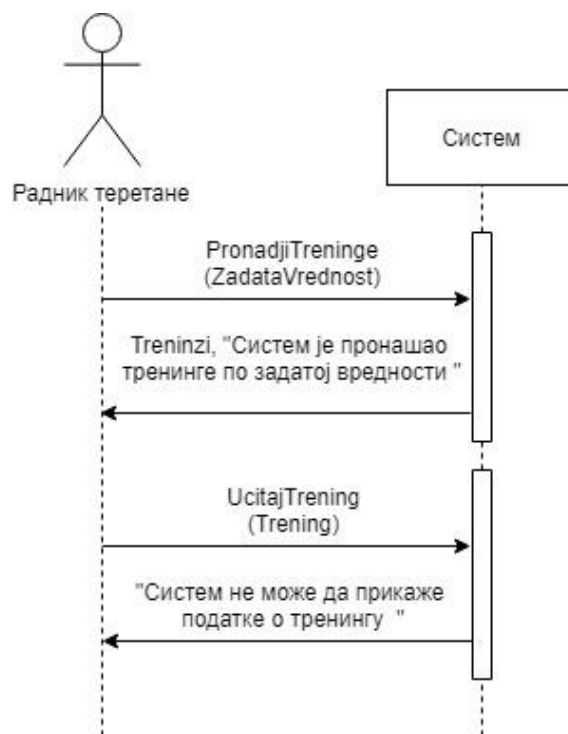
2.1 Уколико систем не може да нађе тренинге он приказује раднику поруку: “Систем не може да нађе тренинге по задатој вредности”. Прекида се извршење сценарија . (ИА)



Слика бр. 5. Дијаграм секвенци – систем не може да пронађе тренинге

Алтернативни сценарио бр.2.

4.1 Уколико систем не може да нађе податке о изабраном тренингу он приказује поруку раднику: “Систем не може да учита податке о тренингу“. (ИА)



Слика бр. 6. Дијаграм секвенци – систем не може да пронађе податке о тренингу

Листа системских операција:

- *signal* PronadjiTreninge(ZadataVrednost, List<Trening>)
- *signal* UcitajTrening(Trening)

ДС3: Дијаграм секвенци – Измена тренинга

Основни сценарио

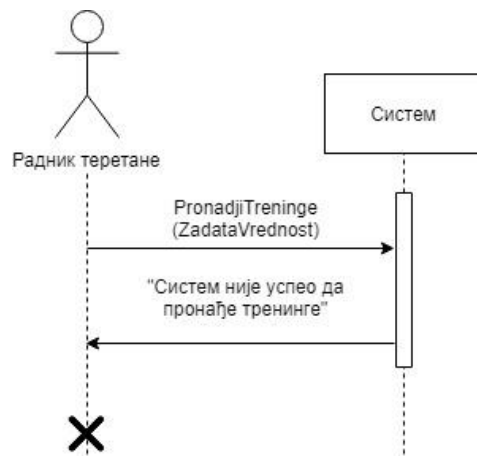
1. Радник позива систем да нађе тренинг по задатој вредности. (АПСО)
2. Систем приказује раднику тренинге и поруку: “Систем је нашао тренинге по задатој вредности”. (ИА)
3. Радник позива систем да учита податке о изабраном тренингу. (АПСО)
4. Систем приказује раднику податке о изабраном тренингу и поруку: “Систем је успешно пронашао податке о изабраном тренингу”. (ИА)
5. Радник позива систем да промени податке о тренингу. (АПСО)
6. Систем приказује раднику промењен тренинг и поруку: “Систем је променио тренинг.” (ИА)



Слика бр. 7. Дијаграм секвенци – измена тренинга, основни сценарио

Алтернативни сценарио бр.1.

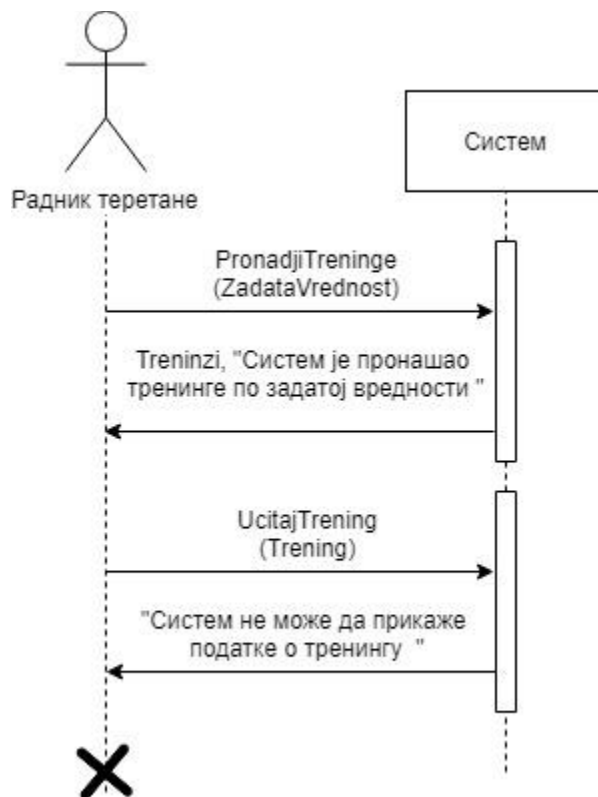
2.1 Уколико систем не може да нађе тренинг он приказује раднику поруку: “Систем не може да нађе тренинге по задатој вредности”. Прекида се извршење сценарија. (ИА)



Слика бр. 8. Дијаграм секвенци – систем не може да пронађе тренинге

Алтернативни сценарио бр.2.

4.1 Уколико систем не може да запамти податке о тренингу он приказује раднику поруку “Систем не може да прикаже податке о изабраном тренингу”. Прекида се извршење сценарија. (ИА)



Слика бр. 9. Дијаграм секвенци – систем не може да прикаже податке о изабраном тренингу

Алтернативни сценарио бр.3.

6.1 Уколико систем не може да запамти тренинг он приказује раднику поруку: “Систем не може да запамти тренинг”. (ИА)



Слика бр. 10. Дијаграм секвенци – систем не може да промени тренинг

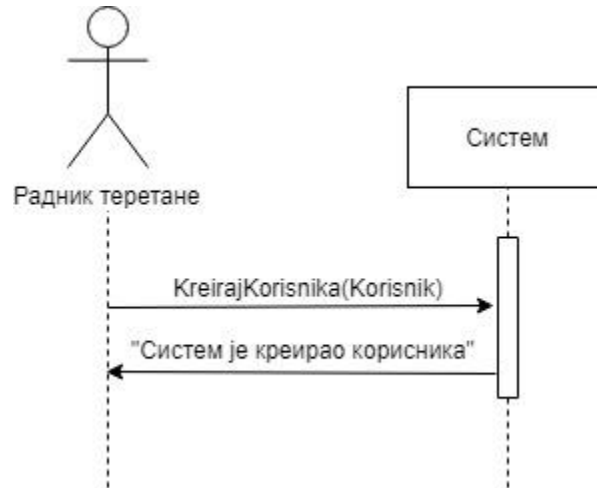
Листа системских операција:

- *signal* `PronadjiTreninge(ZadataVrednost, List<Trening>)`
- *signal* `UcitajTrening(Trening)`
- *signal* `PromeniTrening(Trening)`

ДС4: Дијаграм секвенци – Унос корисника

Основни сценарио

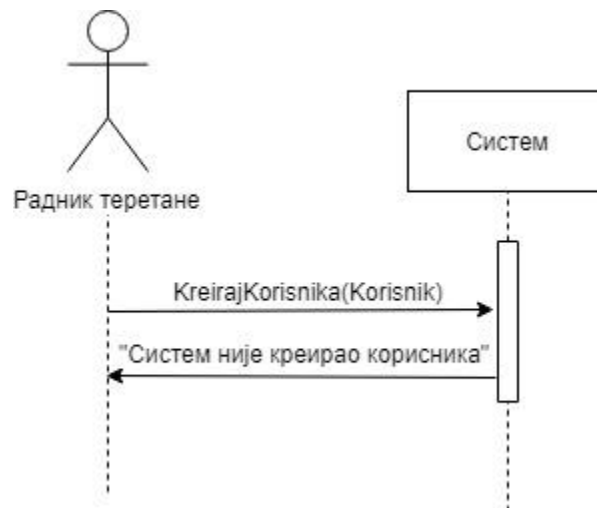
1. Радник позива систем да запамти податке о кориснику. (АПСО)
2. Систем приказује раднику корисника и поруку: “Систем је креирао корисника“. (ИА)



Слика бр. 11. Дијаграм секвенци – унос корисника, основни сценарио

Алтернативни сценарио бр.1.

- 2.1. Уколико систем не може да креира корисника он приказује раднику поруку “Систем не може да креира корисника“. (ИА)



Слика бр. 12. Дијаграм секвенци – систем не може да креира корисника

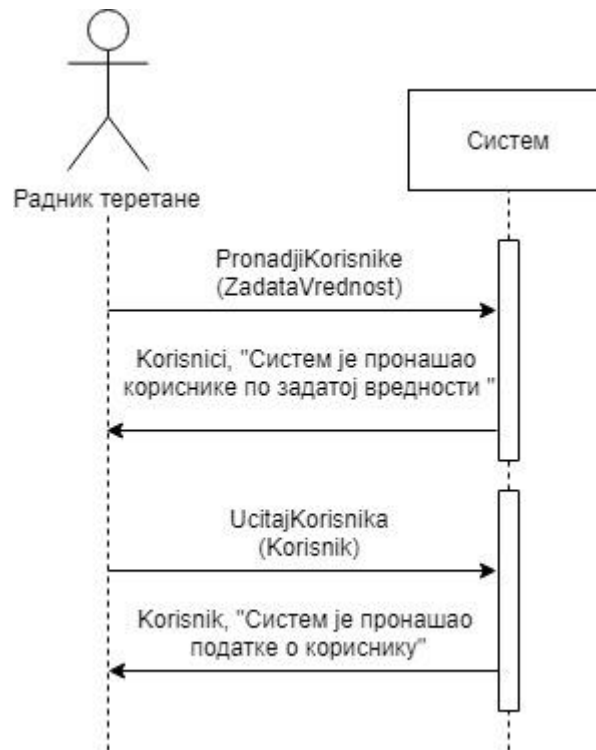
Листа системских операција:

- *signal* KreirajKorisnika(Korisnik)

ДС5: Дијаграм секвенци – Претрага корисника

Основни сценарио

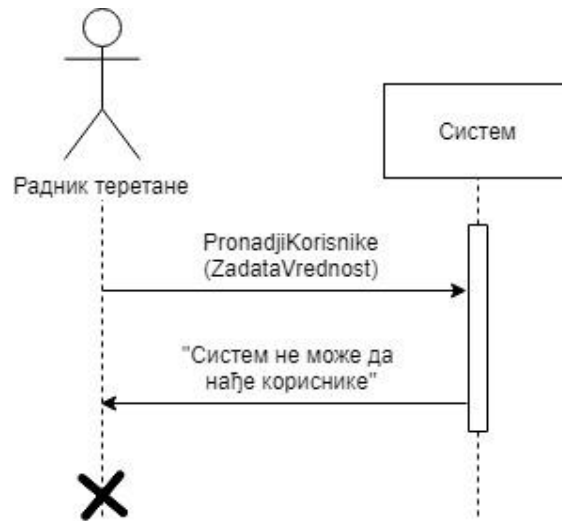
1. Радник позива систем да нађе кориснике по задатој вредности. (АПСО)
2. Систем приказује раднику податке о корисницима и поруку: “Систем је нашао кориснике по задатој вредности”. (ИА)
3. Радник позива систем да учита податке о изабраном кориснику. (АПСО)
4. Систем приказује раднику податке о изабраном кориснику и поруку: “Систем је успешно пронашао податке о изабраном кориснику”. (ИА)



Слика бр. 13. Дијаграм секвенци – претрага корисника, основни сценарио

Алтернативни сценарио бр.1.

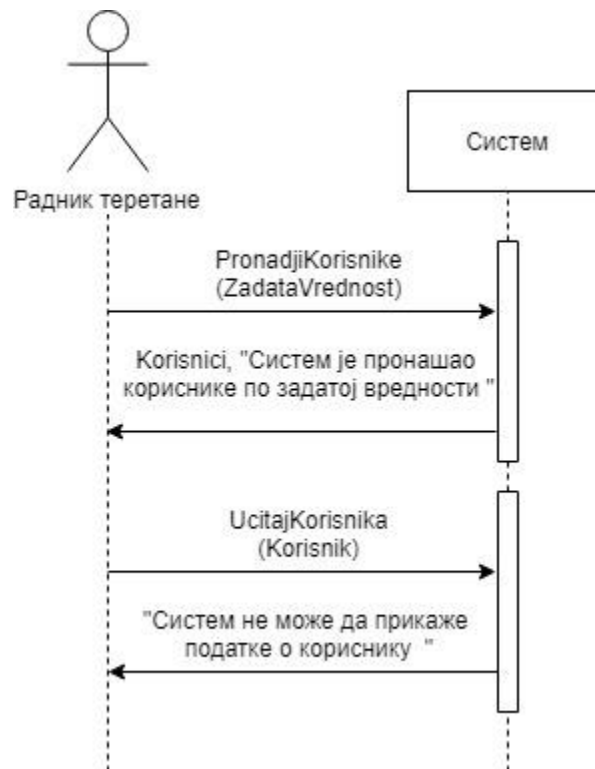
2.1 Уколико систем не може да нађе корисника он приказује раднику поруку: “Систем не може да нађе корисника по задатој вредности”. Прекида се извршење сценарија. (ИА)



Слика бр. 14. Дијаграм секвенци – систем не може да нађе кориснике

Алтернативни сценарио бр.2.

4.1 Уколико систем не може да нађе податке о изабраном кориснику он приказује поруку раднику : “Систем не може да учита податке о кориснику“. (ИА)



Слика бр. 15. Дијаграм секвенци – систем не може да нађе кориснике

Листа системских операција:

- *signal* PronadjiKorisnike(ZadataVrednost, List<Korisnik>)
- *signal* UcitajKorisnika(Korisnik)

ДС6: Дијаграм секвенци – Измена корисника

Основни сценарио

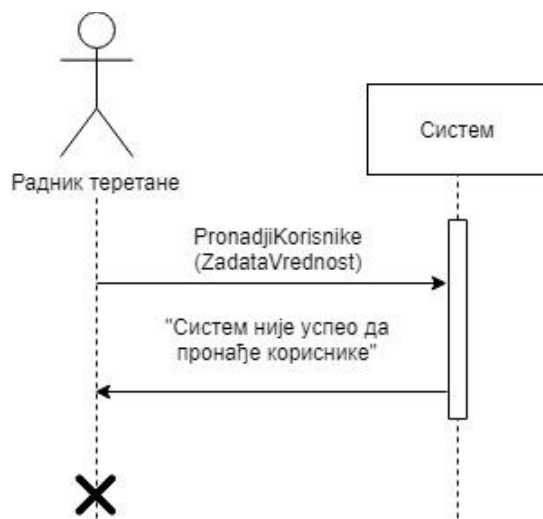
1. Радник позива систем да нађе кориснике по задатој вредности. (АПСО)
2. Систем приказује раднику кориснике и поруку: “Систем је нашао кориснике по задатој вредности”. (ИА)
3. Радник позива систем да учита податке о изабраном кориснику. (АПСО)
4. Систем приказује раднику податке о изабраном кориснику и поруку: “Систем је успешно пронашао податке о изабраном кориснику”. (ИА)
5. Радник позива систем да промени податке о кориснику. (АПСО)
6. Систем приказује раднику промењеног корисника и поруку: “Систем је променио корисника.” (ИА)



Слика бр. 16. Дијаграм секвенци – измена корисника, основни сценарио

Алтернативни сценарио бр.1.

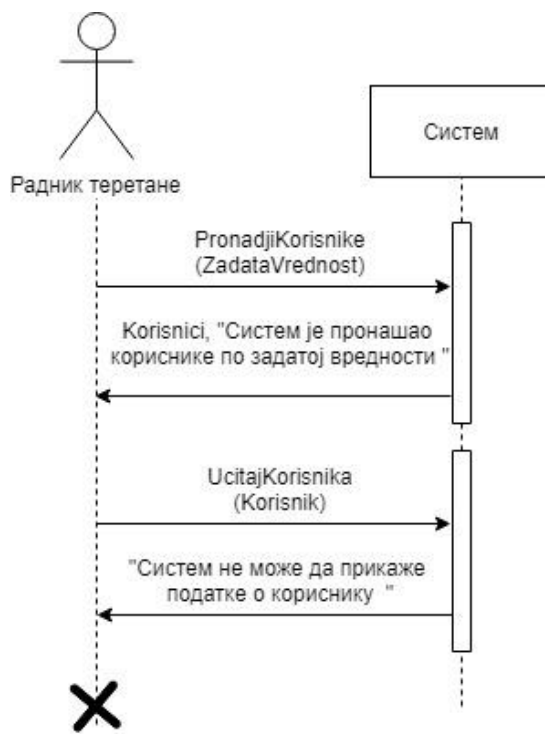
2.1 Уколико систем не може да нађе кориснике он приказује раднику поруку: “Систем не може да нађе кориснике по задатој вредности”. Прекида се извршење сценарија. (ИА)



Слика бр. 17. Дијаграм секвенци – систем не може да пронађе кориснике

Алтернативни сценарио бр.2.

4.1. Уколико систем не може да нађе податке о кориснику он приказује раднику поруку “Систем не може да пронађе податке о кориснику. Прекида се извршење сценарија. (ИА)



Слика бр. 18. Дијаграм секвенци – систем не може да пронађе податке о кориснику

Алтернативни сценарио бр.3.

6.1 Уколико систем не може да промени корисника он приказује раднику поруку: “Систем не може да промени корисника”. (ИА)



Слика бр. 19. Дијаграм секвенци – систем не може да промени податке о кориснику

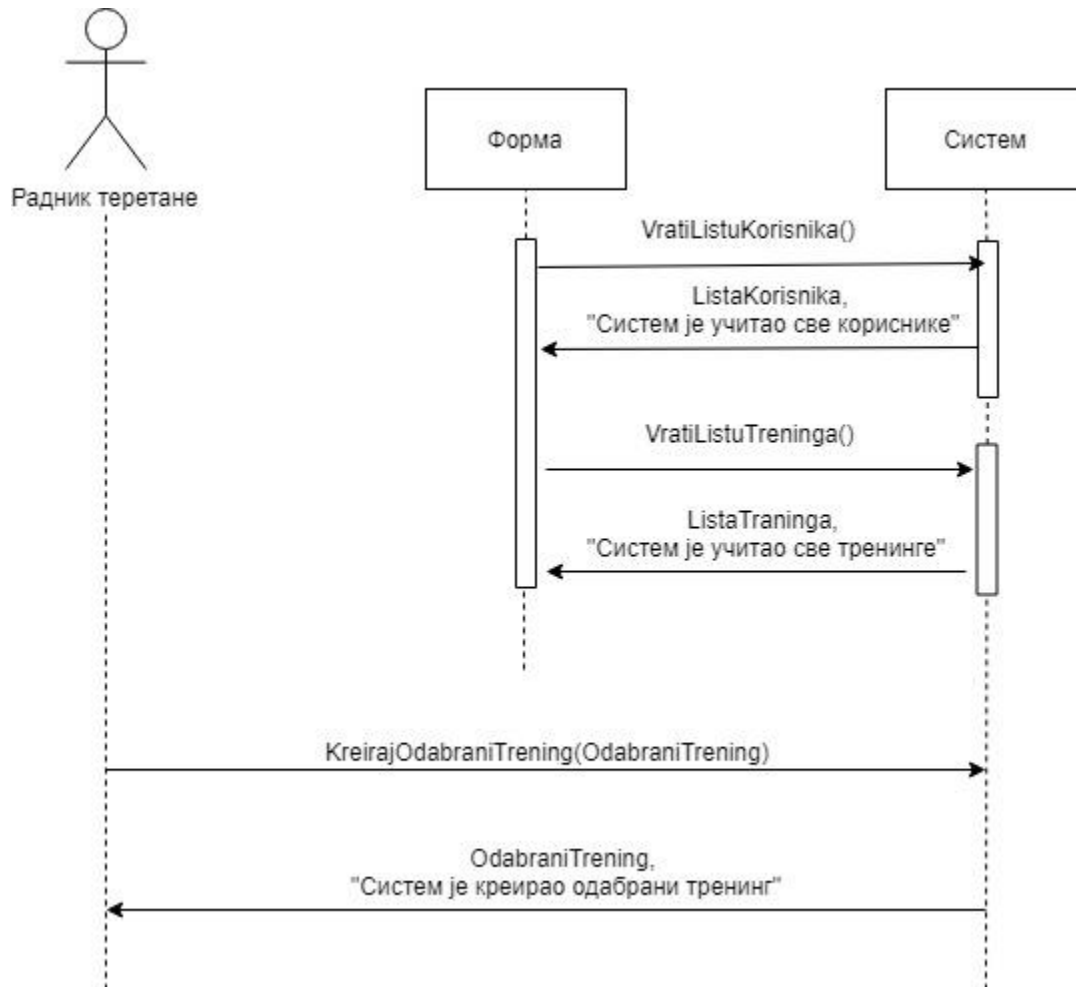
Листа системских операција:

- *signal* PronadjiKorisnike(ZadataVrednost, List<Korsinik>)
- *signal* UcitajKorisnika(Korisnik)
- *signal* PromeniKorisnika(Korisnik)

ДС7: Дијаграм секвенци – Унос одабраног тренинга

Основни сценарио

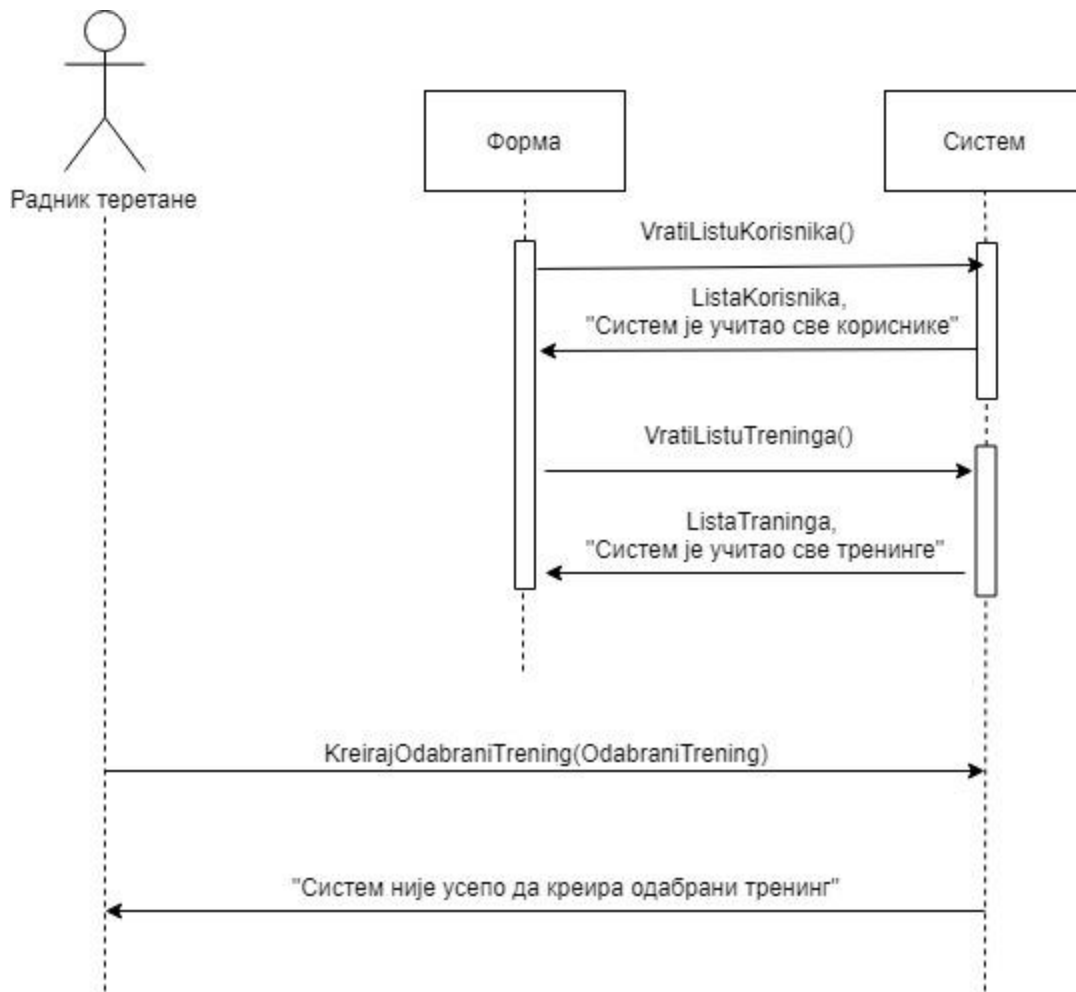
1. Форма позива систем да прикаже листу корисника. (АПСО)
2. Систем приказује форми листу корисника. (ИА)
3. Форма позива систем да прикаже листу тренинга. (АПСО)
4. Систем приказује форми листу тренинга. (ИА)
5. Радник позива систем да креира одабрани тренинг. (АПСО)
6. Систем приказује раднику одабрани тренинг и поруку: “Систем је креирао одабрани тренинг“. (ИА)



Слика бр. 20. Дијаграм секвенци – унос одабраног тренинга, основни сценарио

Алтернативни сценарио бр.1.

- 6.1 Уколико систем не може да креира одабрани тренинг он приказује раднику поруку: “Систем не може да креира одабрани тренинг”.



Слика бр. 21. Дијаграм секвенци – систем није успео да креира одабрани тренинг

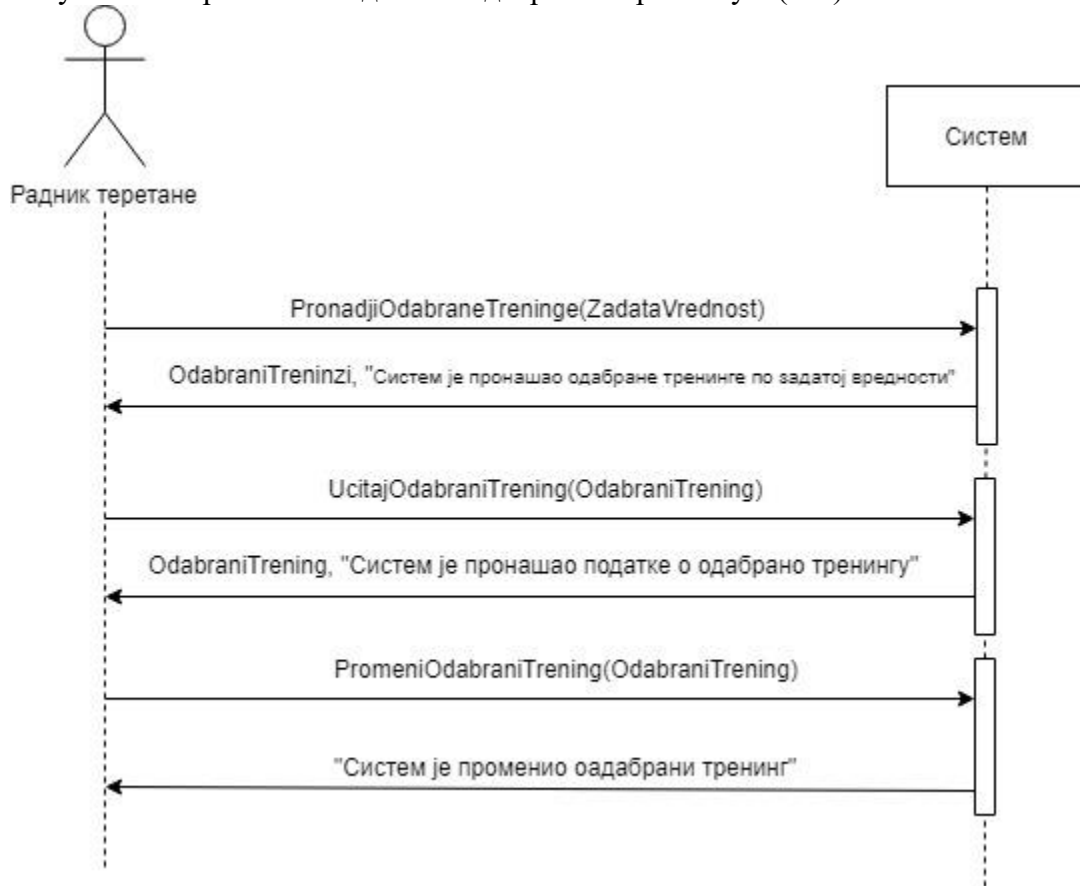
Листа системских операција:

- *signal* VratiListuKorisnika(List<Korisnika>)
- *signal* VratiListuTreninga(List<Trening>)
- *signal* KreirajOdabraniTrening(OdabraniTrening)

ДС8: Дијаграм секвенци – Претрага одабраног тренинга

Основни сценарио

1. Радник позива систем да нађе одабрани тренинг по задатој вредности. (АПСО)
2. Систем приказује раднику одабране тренинге и поруку: “Систем је нашао одабране тренинге по задатој вредности”. (ИА)
3. Радник позива систем да учита податке о одабраном тренингу. (АПСО)
4. Систем приказује раднику податке о одабраном тренингу и поруку: “Систем је успешно пронашао податке о одабраном тренингу“. (ИА)



Слика бр. 22. Дијаграм секвенци – измена одабраног тренинга, основни сценарио

Алтернативни сценарио бр.1.

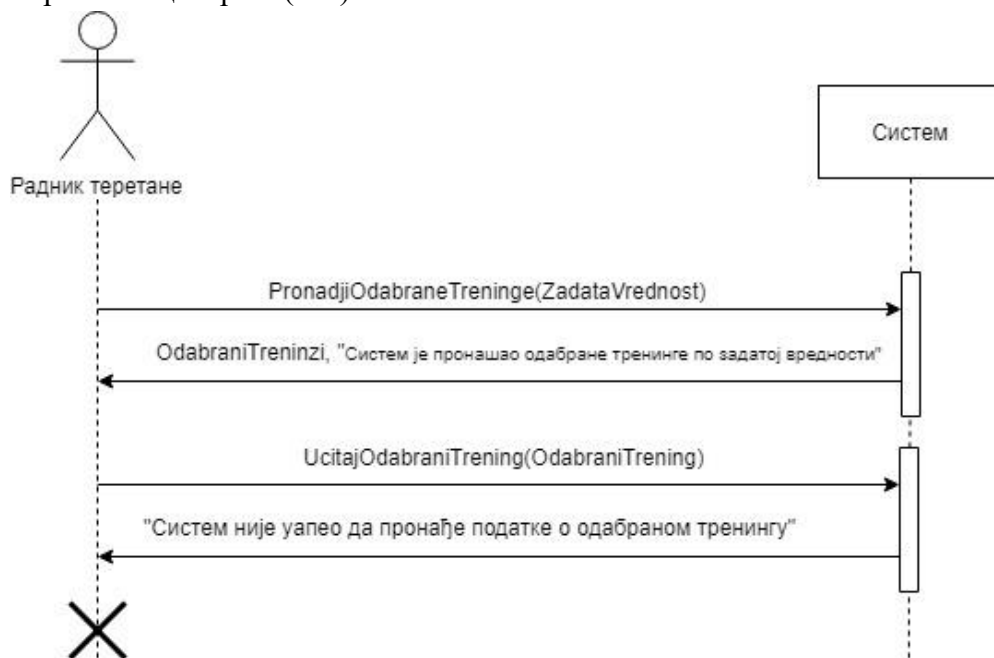
2.1 Уколико систем не може да нађе одабране тренинге он приказује раднику поруку: “Систем не може да нађе одабране тренинге по задатој вредности”. Прекида се извршење сценарија. (ИА)



Слика бр. 23. Дијаграм секвенци – систем није успео да пронађе одабране тренинге

Алтернативни сценарио бр.2.

4.1 Уколико систем не може да запамти податке о одабраном тренингу он приказује раднику поруку “Систем не може да прикаже податке о одабраном тренингу”. Прекида се извршење сценарија. (ИА)



Слика бр. 24. Дијаграм секвенци – систем није успео да прикаже податке о одабраном тренингу

Листа системских операција:

- *signal* PronadjiOdabranitrening(ZadataVrednost, List<OdabraniTrening>)
- *signal* UcitajOdabraniTrening(OdabraniTrening)

Списак свих системских операција

1. KreirajTrening(Trening)
2. PronadjiTreninge(ZadataVrednost, List<Trening>)
3. UcitajTrening(Trening)
4. PromeniTrening(Trening)
5. KreirajKorisnika(Korisnik)
6. PronadjiKorisnike(ZadataVrednost, List<Korisnik>)
7. UcitajKorisnika(Korisnik)
8. PromeniKorisnika(Korisnik)
9. VратиListuKorisnika(List<Korisnik>)
10. VратиListuTreninga(List<Trening>)
11. KreirajOdabraniTrening(OdabraniTrening)
12. PronadjiOdabraniTrening(ZadataVrednost, List<OdabraniTrening>)
13. UcitajOdabraniTrening(OdabraniTrening)

Дефинисње уговора о системским операцијама

Системска операција представља понашање софтверског система. За сваку СО праве се уговори, који описују понашање операције, тј. оно што операција треба да уради, али не и начин на који ће се то одрадiti. Један уговор везује се за једну системску операцију.

Уговор УГ1: KreirajTrening(Trening)

Операција: KreirajTrening(Trening): signal;

Веза са СК: СК1

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

Постуслови: Креиран је нови тренинг.

Уговор УГ2: PronadjiTreninge(ZadataVrednost, List<Trening>)

Операција: PronadjiTreninge(ZadataVrednost, List<Trening>): signal;

Веза са СК: СК2, СК3

Предуслови: /

Постуслови: /

Уговор УГ3: UcitajTrening(Trening)

Операција: UcitajTrening(Trening): signal;

Веза са СК: СК2, СК3

Предуслови: /

Постуслови: /

Уговор УГ4: PromeniTrening(Trening)

Операција: PromeniTreninge(Trening): signal;

Веза са СК: СК3

Предуслови: Тренинг мора постојати у систему да би био измењен. Вредносна и структурна ограничења над објектом *Trening* морају бити задовољена

Постуслови: Измењени су подаци о тренингу.

Уговор УГ5: KreirajKorisnika(Korisnik)

Операција: KreirajKorisnika(Korisnik): signal;

Веза са СК: СК4

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

Постуслови: Креиран је нови корисник.

Уговор УГ6: PronadjiKorisnike(ZadataVrednost, List<Korisnik>)

Операција: PronadjiKorisnike(ZadataVrednost): signal;

Веза са СК: СК4, СК5, СК6

Предуслови: /

Постуслови: /

Уговор УГ7: UcitajKorisnika(Korisnik)

Операција: UcitajKorisnika(Korisnik): signal;

Веза са СК: СК4, СК5, СК6

Предуслови: /

Постуслови: /

Уговор УГ8: PromeniKorisnika(Korisnik)

Операција: PromeniKorisnika(Korisnik): signal;

Веза са СК: СК6

Предуслови: Корисник мора постојати у систему да би био измењен. Вредносна и структурна ограничења над објектом *Korisnik* морају бити задовољена

Постуслови: Измењени су подаци о кориснику.

Уговор УГ9: VratiListuKorisnika(List<Korisnik>)

Операција: VratiListuKorisnika(List<Korisnik>): signal;

Веза са СК: СК7

Предуслови: /

Постуслови: /

Уговор УГ10: VратиListuTreninga(List<Trening>)

Операција: VратиListuTreninga(List<Trening>): signal;

Веза са СК: СК7

Предуслови: /

Постуслови: /

Уговор УГ11: KreirajOdabraniTrening(OdabraniTrening)

Операција: KreirajOdabraniTrening (OdabraniTrening): signal;

Веза са СК: СК7

Предуслови: Вредносна и структурна ограничења над објектом *OdabraniTrening* морају бити задовољена. У систему морају постојати клијенти и тренинзи.

Постуслови: Креиран је нови одабрани тренинг.

Уговор УГ12: PronadjiOdabraniTrening(ZadataVrednost, List<OdabraniTrening>)

Операција: PronadjiOdabraniTrening(ZadataVrednost, List<Trening>): signal;

Веза са СК: СК8

Предуслови: /

Постуслови: /

Уговор УГ13: UcitajOdabraniTrening(OdabraniTrening)

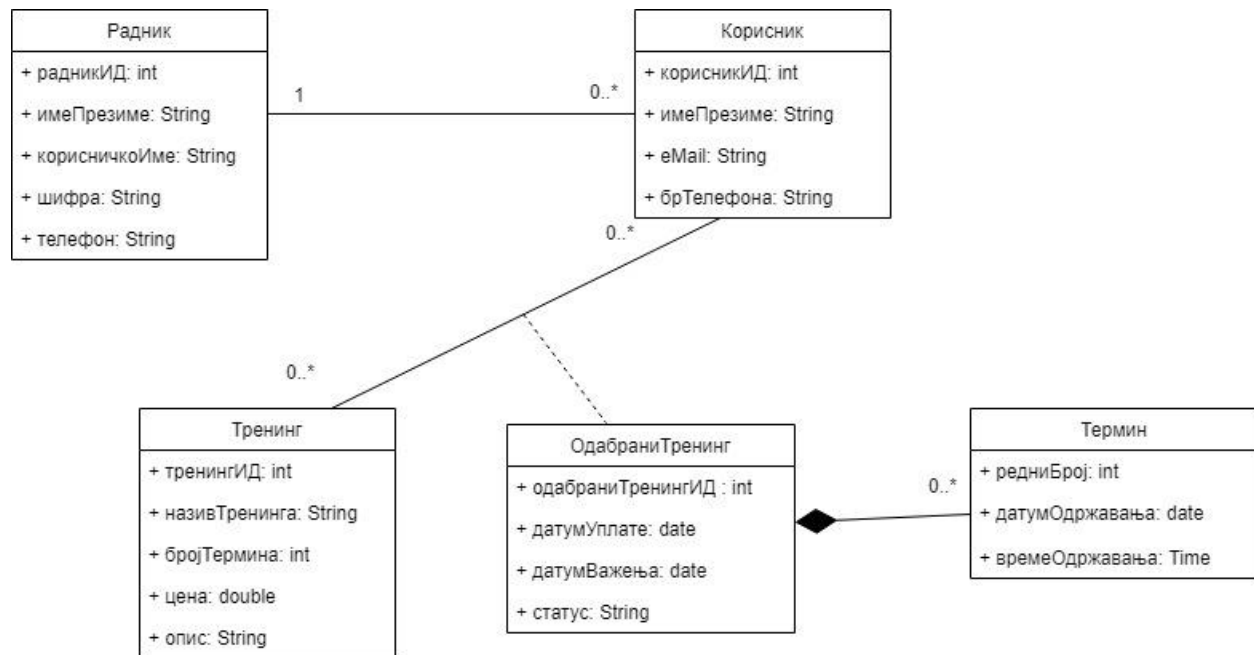
Операција: UcitajOdabraniTrening(OdabraniTrening): signal;

Веза са СК: СК8

Предуслови: /

Постуслови: /

Концептуални модел



Слика бр. 25. Концептулани модел

Релациони модел

Радник (радни́кИД, корисничкоИме, шифра, имеПрезиме, телефон)

Корисник (корисникИД, имеПрезиме, eMail, бројТелефона, *радни́кИД*)

Тренинг (тренингИД, називТренинга, опис, бројТермина, цена)

ОдабраниТренинг (одабраниТренингИД, датумУплате, датумВажења, статус, *корисникИД*, *тренингИД*)

Термин (одабраниТренингИД, редниБрој, датумОдржавања, времеОдржавања)

Табела Радник		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип атрибута	Вредност атрибута	Међузав. атрибута једне табеле	Међузав. атрибута више табеле	INSERT /
	радникИД	int	not null, >0			UPDATE CASCADES Корисник DELETE RESTRICTED Корисник
	имеПрезиме	String				
	корисничкиИме	String	not null			
	шифра	String	not null			
	телефон	String				

Табела бр.1 табела Радник

Табела Корисник		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип атрибута	Вредност атрибута	Међузав. атрибута једне табеле	Међузав. атрибута више табеле	INSERT RESTRICTED Радник UPDATE CASCADES ОдабраниТренинг RESTRICTED Радник DELETE CASCADES ОдабраниТренинг
	корисникИД	int	not null, >0			
	имеПрезиме	String	not null			
	eMail	String				
	бројТелефона	String				
	радникИД	int				

Табела бр.2 табела Корисник

Табела Тренинг		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип атрибута	Вредност атрибута	Међузав. атрибута једне табеле	Међузав. атрибута више табеле	INSERT / UPDATE CASCADES ОдабраниТренинг DELETE RESTRICTED ОдабраниТренинг
	тренингИД	int	not null, >0			
	називТренинга	String				
	брТермина	int	>0			
	цена	double	>0			
	опис	String				

Табела бр.3 табела *Тренинг*

Табела ОдабраниТренинг		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип атрибута	Вредност атрибута	Међузав. атрибута једне табеле	Међузав. атрибута више табеле	INSERT RESTRICTED Корисник, Тренинг UPDATE RESTRICTED Корисник, Тренинг CASCADE Термин DELETE CASCADES Термин
	ОдабраниТренингИД	int	not null, >0			
	статус	String				
	датумУплате	Date				
	датумВажења	Date				
	корисникИД	int				
	радникИД	int				

Табела бр.4 табела *ОдабраниТренинг*

Табела Термин		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип атрибута	Вредност атрибута	Међузав. атрибута једне табеле	Међузав. атрибута више табеле	INSERT RESTRICTED Одабрани Тренинг UPDATE RESTRICTED Одабрани Тренинг DELETE /
	редниБрој	int	not null, >0			
	одабрани ТренингИД	int	not null, >0			
	датум Одржавања	Date			>ОдабраниТренинг .датумУплате, <ОдабраниТренинг .датумВажење	
	време Одржавања	Time				

Табела бр.5 табела *Термин*

Пројектовање

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

Архитектура софтверског система

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

- Кориснички интерфејс
- Апликациона логика
- Складиште података

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



Слика бр. 26. Тронивојска архитектура

Пројектовање корисничког интерфејса

Кориснички интерфејс је дефинисан преко скупа екранских форми. Сценарија коришћења екранских форми су директно повезани са сценаријима случајева коришћења. Екранска форма има улогу да прихвати податке које уноси актор, прихвата догађаје које прави актор, позива контролера корисничког интерфејса како би му проследио те податке и приказује податке добијене од контролера корисничког интерфејса.

Изглед серверске форме је приказан на слици. Треба омогућити да корисник може сам да унесе порт на ком ће бити покренут сервер, као и могућност да заустави рад сервера у било ком тренутку. Поред тога, корисник треба да може да подешава конфигурацију базе података и да има преглед свих пријављених корисника.

[TERETANA] Server

Status server programa

Server program nije pokrenut.

Pokreni server **Zaustavi server**

Konfiguracija baze

Url:

Username:

Password:

Podesi

Ulogovani radnici

Ime i prezime	Korisnicko ime	Telefon

Konfiguracija konekcije

IP adresa:

Port:

Podesi

Слика бр. 27. Изглед серверске форме пре покретања сервера

[TERETANA] Server

Status server programa

Server program je pokrenut na portu: 9004.

Pokreni server **Zaustavi server**

Konfiguracija baze

Url:

Username:

Password:

Podesi

Ulogovani radnici

Ime i prezime	Korisnicko ime	Telefon

Konfiguracija konekcije

IP adresa:

Port:

Podesi

Слика бр. 28. Изглед серверске форме након покретања сервера

[TERETANA] Server

Status server programa

Server program je pokrenut na portu: 9004.

Pokreni server Zaustavi server

Konfiguracija baze

Url: jdbc:mysql://localhost:3306/softveri

Username: root

Password: root

Podesi

Ulogovani radnici

Ime i prezime	Korisnicko ime	Telefon
Ana Cvetojevic	ana	064 931 3038
Jovana Cvetojevic	jovana	064 911 3457

Konfiguracija konekcije

IP adresa: localhost

Port: 9004

Podesi

Слика бр. 29. Изглед серверске форме након што се корисници пријаве на систем

[TERETANA] Server

Status server programa

Server program nije pokrenut.

Pokreni server Zaustavi server

Konfiguracija baze

Url: jdbc:mysql://localhost:3306/softveri

Username: root

Password: root

Podesi

Ulogovani radnici

Ime i prezime	Korisnicko ime	Telefon

Konfiguracija konekcije

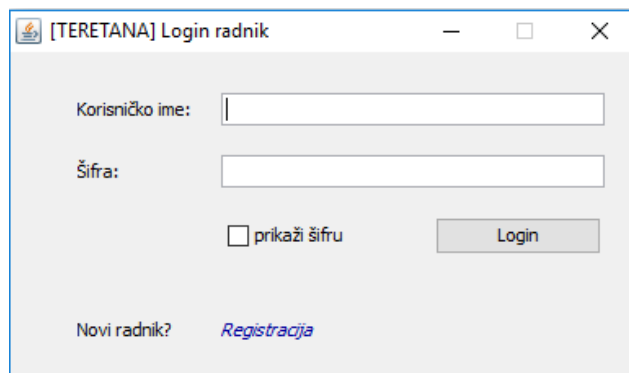
IP adresa: localhost

Port: 9004

Podesi

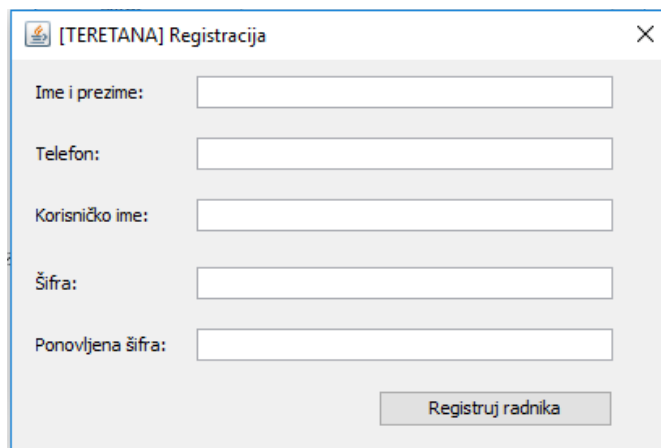
Слика бр. 30. Изглед серверске форме након заустављања сервера

Након покретања клијентске форме појављује се екранска форма за пријаву радника теретане. На форми постоји могућност регистрације нових корисника.



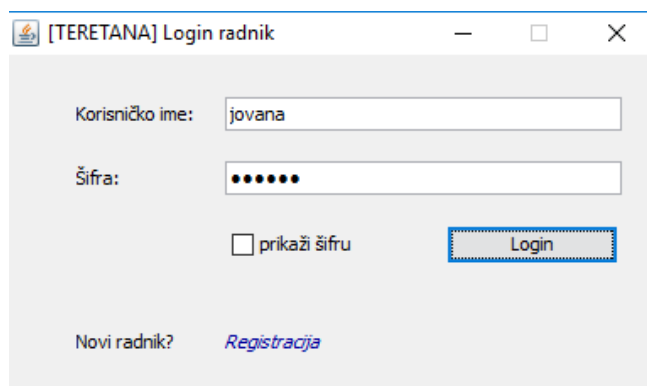
The screenshot shows a window titled "[TERETANA] Login radnik". It contains two text input fields: "Korisničko ime:" and "Šifra:". Below the "Šifra:" field is a checkbox labeled "prikaži šifru". To the right of the checkbox is a "Login" button. At the bottom left, it says "Novi radnik?" followed by a blue link "Registracija".

Слика бр. 31. Изглед након покретања клијентске апликације



The screenshot shows a window titled "[TERETANA] Registracija". It contains five text input fields: "Ime i prezime:", "Telefon:", "Korisničko ime:", "Šifra:", and "Ponovljena šifra:". At the bottom right is a "Registruj radnika" button.

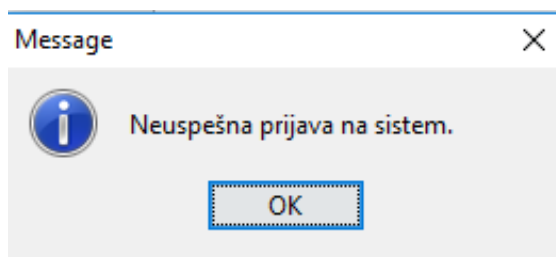
Слика бр. 32. Изглед форме за регистрацију радника



The screenshot shows the same window as in Figure 31, but with data entered. The "Korisničko ime:" field contains "jovana" and the "Šifra:" field contains seven dots. The "Login" button is highlighted with a blue dashed border. The "prikaži šifru" checkbox is still unchecked. The "Novi radnik?" link "Registracija" is still present at the bottom.

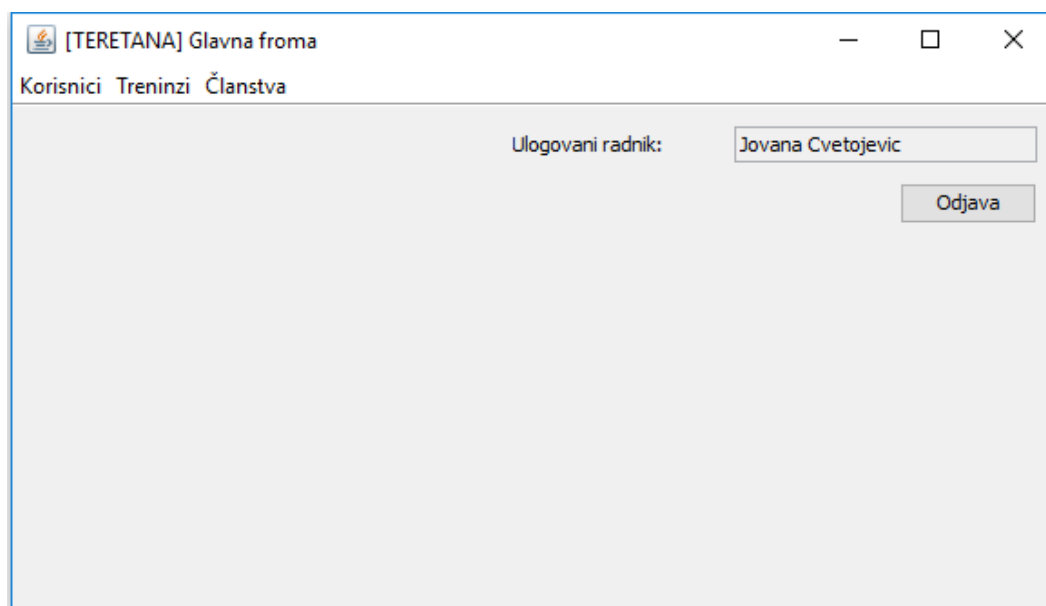
Слика бр. 33. Изглед форме пријаве на систем

Уколико подаци за пријаву нису тачни систем приказује поруку о томе.



Слика бр. 34. Изглед након неуспшне пријаве на систем

Након успешне пријаве раднику сте отвара главна форма.



Слика бр. 35. Изглед главне форме

Кроз случајеве коришћења приказане су и друге екранске форме дате апликације. Оне се позивају из менија главне екранске форме клијентског дела апликације.

СК1: Случај коришћења – Унос тренинга

Назив СК

Креирање тренинга

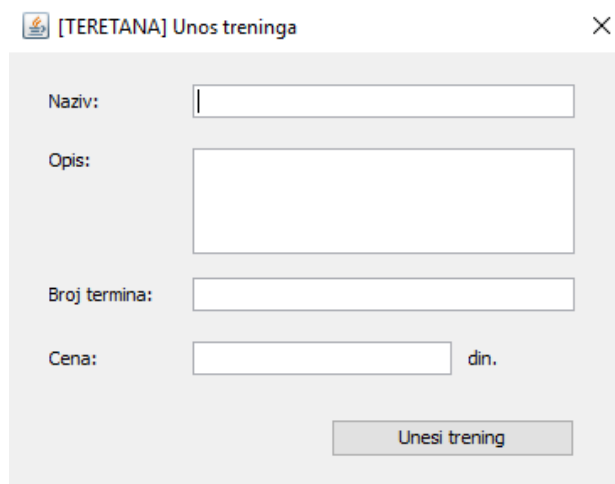
Актори СК

Радник

Учесници СК

Радник и систем (програм)

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



The screenshot shows a window titled "[TERETANA] Unos treninga" with a close button (X) in the top right corner. The form contains the following fields:

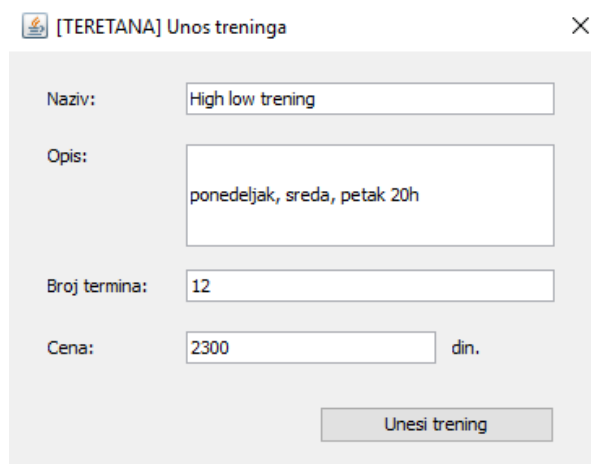
- Naziv:** A text input field.
- Opis:** A larger text input field.
- Broj termina:** A text input field.
- Cena:** A text input field followed by the label "din.".

At the bottom right of the form is a button labeled "Unesi trening".

Слика бр. 36. Изглед форме за унос тренинга

Основни сценарио СК

1. Радник уноси податке у тренинг. (АПУСО)

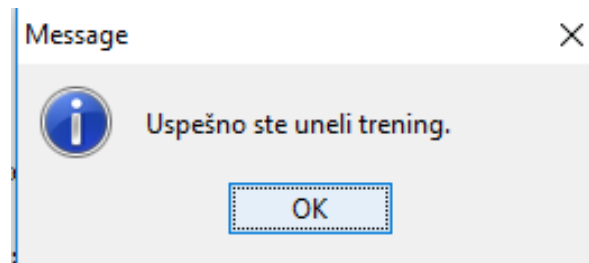


This screenshot shows the same "[TERETANA] Unos treninga" window, but with data entered into the fields:

- Naziv:** "High low trening"
- Opis:** "ponedeljak, sreda, petak 20h"
- Broj termina:** "12"
- Cena:** "2300" followed by "din."

The "Unesi trening" button is still visible at the bottom right.

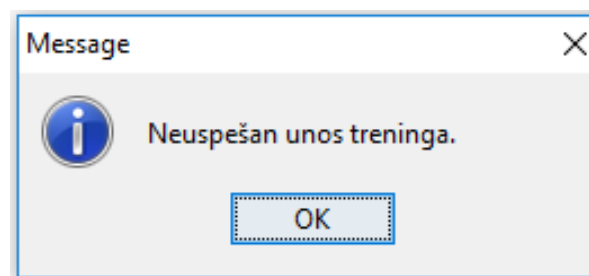
2. Радник контролише да ли је коректно унео податке у тренинг. (АНСО)
3. Радник позива систем да креира тренинг. (АПСО)
4. Систем креира тренинг. (СО)
5. Систем приказује раднику креирани тренинг и поруку: “Систем је креирао тренинг“. (ИА)



Слика бр. 37. Изглед након успешног уноса

Алтернативна сценарија

- 5.1 Уколико систем не може да креира тренинг он приказује раднику поруку “Систем не може да креира тренинг”. (ИА)



Слика бр. 38. Изглед након неуспешног уноса

СК2: Случај коришћења – Претрага тренинга

Назив СК

Претрага тренинга

Актори СК

Радник

Учесници СК

Радник и систем (програм)

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

The screenshot shows a window titled "[TERETANA] Treninzi". It contains a search form with the following elements:

- Input fields: ID, Naziv, Opis, Cena.
- Buttons: Pretraži (search), Unesi (add), Prikaži (show), Izmeni (edit).
- Table structure with headers: ID, Naziv, Opis, Broj termina, Cena.

Слика бр. 39. Изглед форме за претрагу тренинга

Основни сценарио СК

1. Радник уноси вредност по којој претражује тренинг. (АПУСО)

This screenshot shows the same search form as the previous one, but with sample data entered:

- Naziv: Aerobik
- Opis: trajanje
- The Pretraži button is highlighted, indicating the search action.

2. Радник позива систем да нађе тренинг по задатој вредности. (АПСО)

3. Систем тражи тренинг по задатој вредности. (СО)

4. Систем приказује раднику податке о тренинзима и поруку: “Систем је нашао тренинге по задатој вредности”. (ИА)

The screenshot shows a window titled "[TERETANA] Treninzi". It contains search fields for ID, Naziv, Opis (with the value "trajanje"), and Cena. A "Pretraži" button is highlighted. Below the search fields is a table with 5 columns: ID, Naziv, Opis, Broj termina, and Cena. The table contains 5 rows of data. At the bottom, there are three buttons: "Unesi", "Prikaži", and "Izmeni".

ID	Naziv	Opis	Broj termina	Cena
1	Aerobik za zene	1h trajanje treninga, p...	12	2500.0
2	Decija skolica sporta	Rekreacija za decu, tra...	12	1890.0
4	Kik box za muskarce	3 puta nedeljno, 2 h tr...	12	2900.0
3	Kik box za zene	3 puta nedeljno, 2 h tr...	12	2700.0

Слика бр. 40. Изглед након успешне претраге

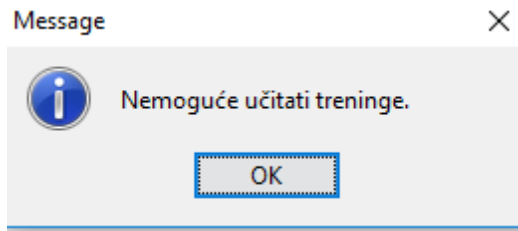
5. Радник бира тренинг. (АПУСО)
6. Радник позива систем да учита податке о тренингу. (АПСО)
7. Систем учитава податке о тренингу. (СО)
8. Систем приказује раднику податке о изабраном тренингу и поруку:“Систем је успешно учитао податке о тренингу“. (ИА)

The screenshot shows a window titled "[TERETANA] Prikaz treninga". It displays the details of a selected training session. The fields are: ID (4), Naziv (Kik box za muskarce), Opis (3 puta nedeljno, 2 h trajanje treninga), Broj termina (12), and Cena (2900.0 din.).

Слика бр. 41. Изглед након одабраног тренинга

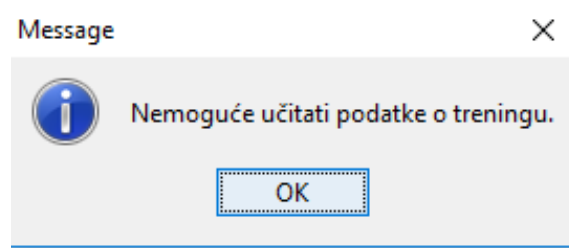
Алтернативна сценарија

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



Слика бр. 42. Изглед након неуспешне претраге тренинга

8.1 Уколико систем не може да нађе податке о изабраном тренингу он приказује поруку раднику: “Систем не може да учита податке о тренингу“. (ИА)



Слика бр. 43. Изглед након неуспешног учитавања података о тренингу

СКЗ: Случај коришћења – Измена тренинга

Назив СК

Промена тренинга

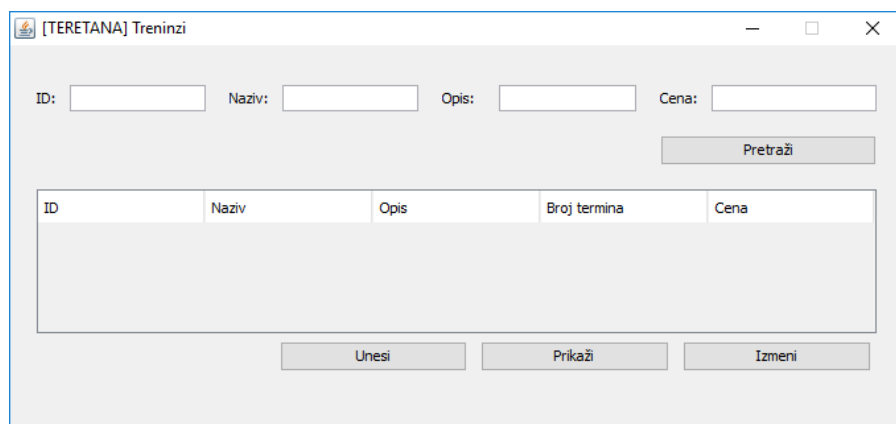
Актори СК

Радник

Учесници СК

Радник и систем (програм)

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

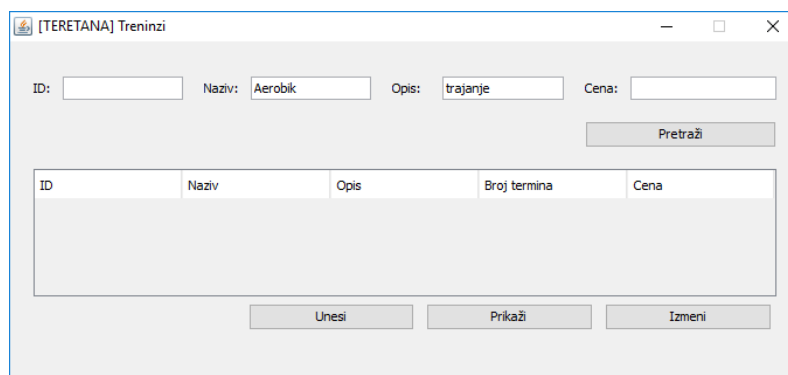


ID	Naziv	Opis	Broj termina	Cena
----	-------	------	--------------	------

Слика бр. 44. Изглед форме за претрагу тренинга

Основни сценарио СК

1. Радник уноси вредност по којој претражује тренинг. (АПУСО)



ID	Naziv	Opis	Broj termina	Cena
----	-------	------	--------------	------

2. Радник позива систем да нађе тренинг по задатој вредности. (АПСО)
3. Систем тражи тренинг по задатој вредности. (СО)
4. Систем приказује раднику тренинге и поруку: “Систем је нашао тренинге по задатој вредности”. (ИА)

[TERETANA] Treninzi

ID: Naziv: Opis: Cena:

ID	Naziv	Opis	Broj termina	Cena
1	Aerobik za zene	1h trajanje treninga, p...	12	2500.0
2	Decija skolica sporta	Rekreacija za decu, tra...	12	1890.0
4	Kik box za muskarce	3 puta nedeljno, 2 h tr...	12	2900.0
3	Kik box za zene	3 puta nedeljno, 2 h tr...	12	2700.0

Слика бр. 45. Изглед форме успешне претраге тренинга

5. Радник бира тренинг који жели да измени. (АПУСО)
6. Радник позива систем да учита податке о изабраном тренингу. (АПСО)
7. Систем проналази податке о изабраном тренингу. (СО)
8. Систем приказује раднику податке о изабраном тренингу и поруку: "Систем је успешно пронашао податке о изабраном тренингу". (ИА)

[TERETANA] Izmena treninga

ID:

Naziv:

Opis:

Broj termina:

Cena: din.

Слика бр. 46. Изглед форме за измену тренинга

9. Радник уноси (мења) податке о тренингу. (АПУСО)

[TERETANA] Izmena treninga

ID:

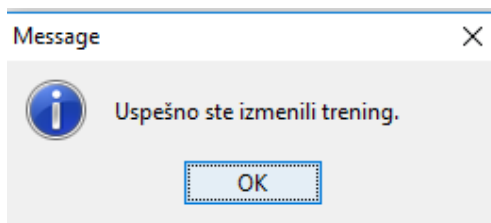
Naziv:

Opis:

Broj termina:

Cena: din.

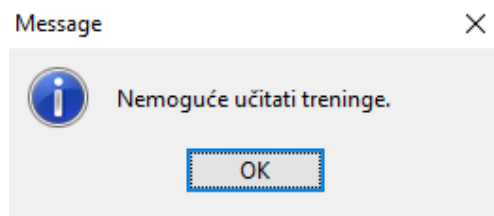
10. Радник контролише да ли је коректно унео податке о тренингу. (АНСО)
11. Радник позива систем да промени податке о тренингу. (АПСО)
12. Систем памти податке о тренингу. (СО)
13. Систем приказује раднику промењени тренинг и поруку: “Систем је променио тренинг.” (ИА)



Слика бр. 47. Изглед након успешне измене тренинга

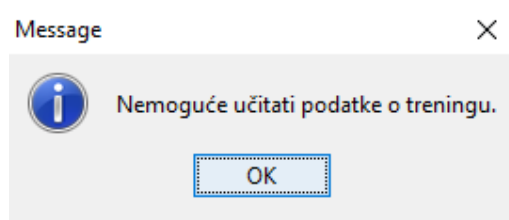
Алтернативна сценарија

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



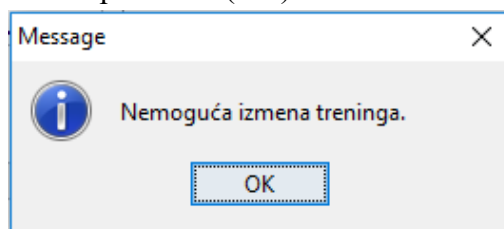
Слика бр. 48. Изглед након успешне претраге тренинга

8.1 Уколико систем не може да учита податке о тренингу он приказује раднику поруку: “Систем не може да прикаже податке о изабраном тренингу”. Прекида се извршење сценарија. (ИА)



Слика бр. 49. Изглед након успешне учитавања тренинга

13.1 Уколико систем не може да запамти тренинг он приказује раднику поруку: “Систем не може да запамти тренинг”. (ИА)



Слика бр. 50. Изглед након неуспешне измене тренинга

СК4: Случај коришћења – Унос корисника

Назив СК

Креирање корисника

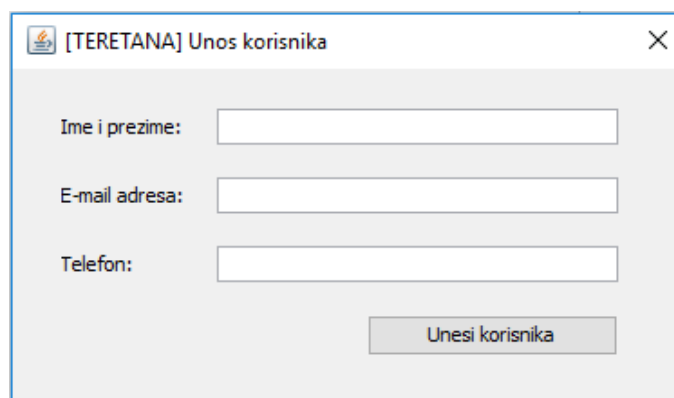
Актори СК

Радник

Учесници СК

Радник и систем (програм)

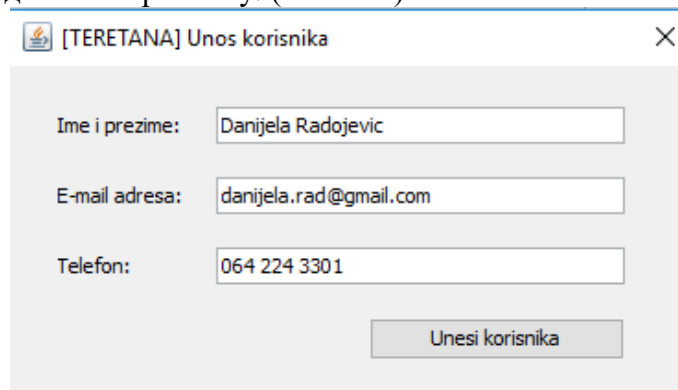
Предуслов: Систем је укључен и радник је пријављен под својом шифром. Систем приказује форму за рад са корисником.



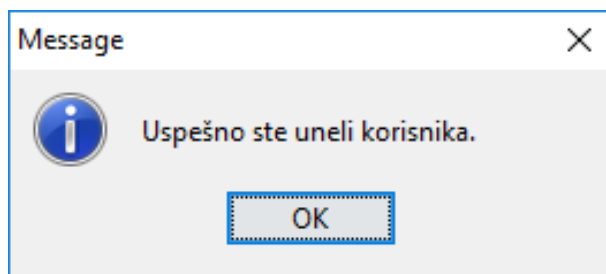
Слика бр. 51. Изглед форме за унос корисника

Основни сценарио СК

1. Радник уноси податке о кориснику. (АПУСО)



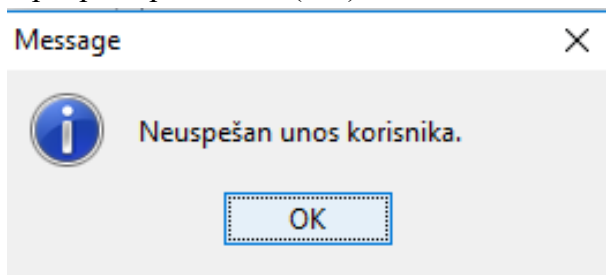
2. Радник контролише да ли је коректно унео податке о кориснику. (АНСО)
3. Радник позива систем да запамти податке о кориснику. (АПСО)
4. Систем креира корисника. (СО)
5. Систем приказује раднику корисника и поруку: “Систем је креирао корисника“. (ИА)



Слика бр. 52. Изглед након успешног уноса корисника

Алтернативна сценарија

5.1 Уколико систем не може да креира корисника он приказује раднику поруку “Систем не може да креира корисника”. (ИА)



Слика бр. 53. Изглед након неуспешног уноса корисника

СК5: Случај коришћења – Претрага корисника

Назив СК

Претрага корисника

Актори СК

Радник

Учесници СК

Радник и систем (програм)

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

The screenshot shows a window titled "[TERETANA] Korisnici". It contains three input fields: "ID:", "Ime i prezime:", and "Radnik:". Below these fields is a "Pretraži" button. Underneath is a table with five columns: "ID", "Ime i prezime", "E-Mail", "Telefon", and "Radnik". The table body is empty. At the bottom of the window are three buttons: "Unesi novi", "Prikaži", and "Izmeni".

Слика бр. 54. Изглед форме за претрагу корисника

Основни сценарио СК

1. Радник уноси вредност по којој претражује корисника. (АПУСО)

This screenshot is identical to the previous one, but the "Radnik:" input field now contains the text "Jovana".

2. Радник позива систем да нађе кориснике по задатој вредности. (АПСО)
3. Систем тражи кориснике по задатој вредности. (СО)
4. Систем приказује раднику податке о корисницима и поруку: “Систем је нашао кориснике по задатој вредности”. (ИА)

[TERETANA] Korisnici

ID: Ime i prezime: Radnik:

ID	Ime i prezime	E-Mail	Telefon	Radnik
13	Danijela Radojevic	danijela.rad@gmail...	064 224 3301	Jovana Cvetojevic
1	Jolaaa	Jolo menjao	9	Jovana Cvetojevic
2	Jovan Jovovic	jovica.jovica@gmail...	063 333 4556	Jovana Cvetojevic

Слика бр. 55. Изглед након успешне претраге корисника

5. Радник бира корисника. (АПУСО)
6. Радник позива систем да учита податке о изабраном кориснику. (АПСО)
7. Систем проналази податке о изабраном кориснику. (СО)
8. Систем приказује раднику податке о изабраном кориснику и поруку: “Систем је успешно пронашао податке о изабраном кориснику“. (ИА)

[TERETANA] Prikaz korisnika

ID:

Ime i prezime:

E-mail adresa:

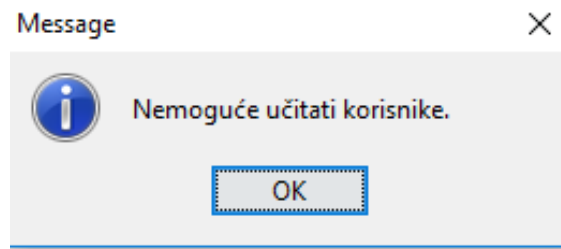
Telefon:

Uneo:

Слика бр. 56. Изглед форме за приказ података корисника

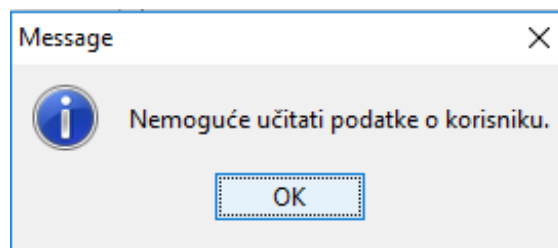
Алтернативна сценарија

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



Слика бр. 57. Изглед након неуспешне претраге корисника

8.1 Уколико систем не може да нађе податке о изабраном кориснику он приказује поруку раднику : “Систем не може да учита податке о кориснику“. (ИА)



Слика бр. 58. Изглед након неуспелог учитавања корисника

СК6: Случај коришћења – Измена корисника

Назив СК

Промена корисника

Актори СК

Радник

Учесници СК

Радник и систем (програм)

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

The screenshot shows a window titled "[TERETANA] Korisnici". It contains three input fields: "ID:", "Ime i prezime:", and "Radnik:". Below these is a "Pretraži" button. Underneath is a table with five columns: "ID", "Ime i prezime", "E-Mail", "Telefon", and "Radnik". The table body is empty. At the bottom are three buttons: "Unesi novi", "Prikaži", and "Izmeni".

Слика бр. 59. Изглед форме за претрагу корисника

Основни сценарио СК

1. Радник уноси вредност по којој претражује кориснике. (АПУСО)

This screenshot is identical to the previous one, but the "Radnik:" input field now contains the text "Jovana".

2. Радник позива систем да нађе кориснике по задатој вредности. (АПСО)
3. Систем тражи кориснике по задатој вредности. (СО)
4. Систем приказује раднику кориснике и поруку: “Систем је нашао кориснике по задатој вредности”. (ИА)

The screenshot shows a window titled "[TERETANA] Korisnici". It has input fields for "ID:", "Ime i prezime:", and "Radnik:" with the value "Jovana" entered in the last field. A "Pretraži" button is to the right. Below is a table with search results:

ID	Ime i prezime	E-Mail	Telefon	Radnik
13	Danijela Radojevic	danijela.rad@gmail...	064 224 3301	Jovana Cvetojevic
1	Jolaaa	Jolo menjao	9	Jovana Cvetojevic
2	Jovan Jovovic	jovica.jovica@gmai...	063 333 4556	Jovana Cvetojevic

At the bottom are buttons: "Unesi novi", "Prikaži", and "Izmeni".

Слика бр. 60. Изглед након успешне претраге корисника

5. Радник бира корисника ког жели да измени. (АПУСО)
6. Радник позива систем да учита податке о изабраном кориснику. (АПСО)
7. Систем проналази податке о изабраном кориснику. (СО)
8. Систем приказује раднику податке о изабраном кориснику и поруку: "Систем је успешно пронашао податке о изабраном коринику". (ИА)

The screenshot shows a window titled "[TERETANA] Izmena korisnika". It contains the following fields:

- ID: 13
- Ime i prezime: Danijela Radojevic
- E-mail adresa: danijela.rad@gmail.com
- Telefon: 064 224 3301
- Uneo: Jovana Cvetojevic

An "Izmeni korisnika" button is at the bottom right.

Слика бр. 61. Изглед форме за измену корисника

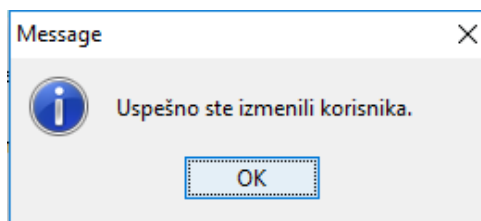
9. Радник уноси (мења) податке о кориснику. (АПУСО)

This screenshot is similar to the previous one but shows updated data:

- ID: 13
- Ime i prezime: Danijela Radojevic
- E-mail adresa: danijela.rad@hotmail.com
- Telefon: 064 224 3305
- Uneo: Jovana Cvetojevic

The "Izmeni korisnika" button remains at the bottom right.

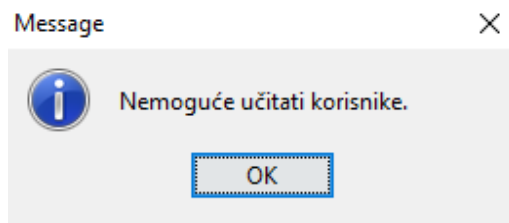
10. Радник контролише да ли је коректно унео податке о кориснику. (АНСО)
11. Радник позива систем да промени податке о кориснику. (АПСО)
12. Систем памти податке о кориснику. (СО)
13. Систем приказује раднику промењеног корисника и поруку: “Систем је променио корисника.” (ИА)



Слика бр. 62. Изглед након успешне измене корисника

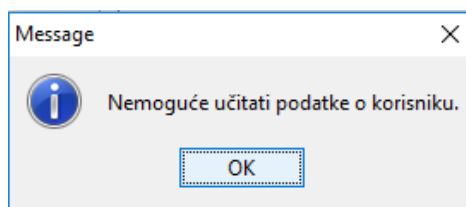
Алтернативна сценарија

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



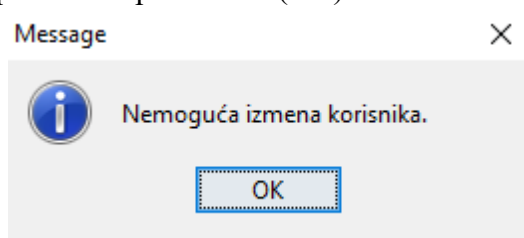
Слика бр. 62. Изглед након неуспешне претраге корисника

8.1 Уколико систем не може да нађе податке о кориснику он приказује раднику поруку “Систем не може да пронађе податке о кориснику. Прекида се извршење сценарија. (ИА)



Слика бр. 62. Изглед након немогућег учитавања корисника

13.1 Уколико систем не може да промени корисника он приказује раднику поруку: “Систем не може да промени корисника”. (ИА)



Слика бр. 65. Изглед након неуспешне измене корисника

СК7: Случај коришћења – Унос одабраног тренинга

Назив СК

Креирање одабраног тренинга

Актори СК

Радник

Учесници СК

Радник и систем (програм)

Предуслов: Систем је укључен и радник је пријављен под својом шифром. Систем приказује форму за рад са одабраним тренингом. Учитана је листа корисника и листа тренинга.

[TERETANA] Unos odabranog treninga

Korisnik: Ana Zdravkovic

Trening: Aerobik za zene

Datum uplate: 27.1.19.

Datum vazenja: 27.1.19.

Status: Važeći

Unos termina

Dan: 27.1.19. Vreme: 16:00 Dodaj

Redni broj	Datum	Vreme
------------	-------	-------

Obriši

Unesi novi

Слика бр. 66. Изглед форме за унос одабраног тренинга

Основни сценарио СК

6. Радник уноси податке о одабраном тренингу. (АПУСО)

[TERETANA] Unos odabranog treninga

Korisnik: Ana Zdravkovic

Trening: Aerobik za zene

Datum uplate: 10. 1. 19.

Datum vazenja: 10. 2. 19.

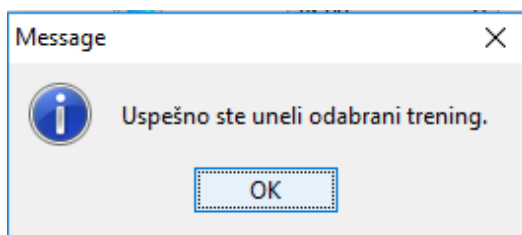
Status: Važeći

Unos termina

Dan: 21. 1. 19. Vreme: 18:00

Redni broj	Datum	Vreme
1	11. 01. 2019.	18:00
2	13. 01. 2019.	18:00
3	15. 01. 2019.	20:00
4	17. 01. 2019.	20:00
5	19. 01. 2019.	18:00
6	21. 01. 2019.	18:00

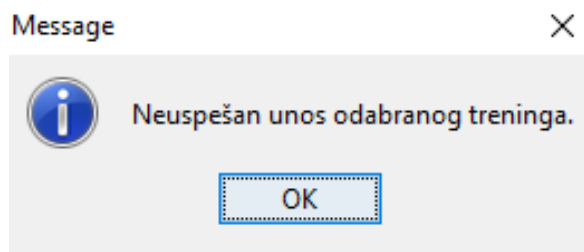
7. Радник контролише да ли је коректно унео податке у одабрани тренинг. (АНСО)
8. Радник позива систем да креира одабрани тренинг. (АПСО)
9. Систем креира одабрани тренинг. (СО)
10. Систем приказује раднику одабрани тренинг и поруку: “Систем је креирао одабрани тренинг”. (ИА)



Слика бр. 67. Изглед након успешног уноса одабраног тренинга

Алтернативна сценарија

- 10.1 Уколико систем не може да креира одабрани тренинг он приказује раднику поруку: “Систем не може да креира одабрани тренинг”.



Слика бр. 67. Изглед након неуспешног уноса одабраног тренинга

СК8: Случај коришћења – Претрага одабраног тренинга

Назив СК

Претрага одабраног тренинга

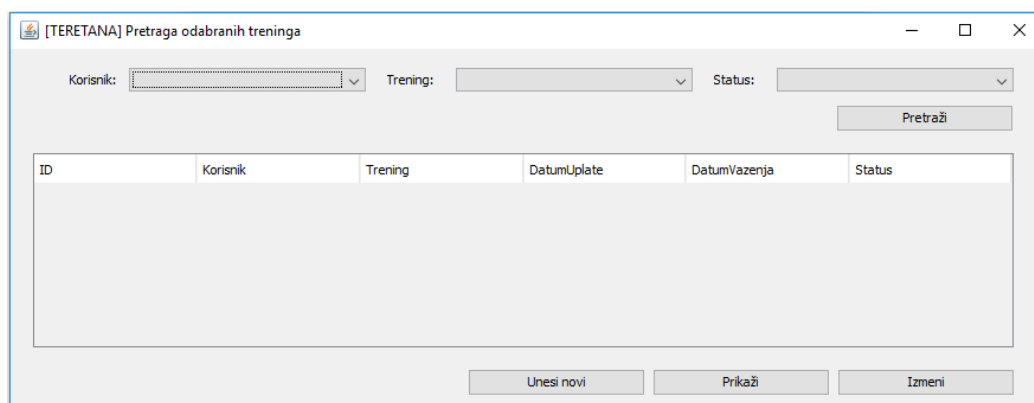
Актори СК

Радник

Учесници СК

Радник и систем (програм)

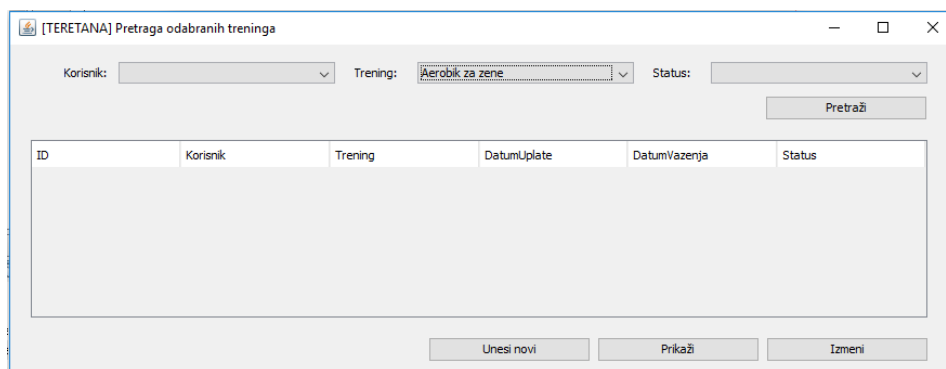
Предуслов: Систем је укључен и радник је пријављен под својом шифром. Систем приказује форму за рад са одабраним тренингом.



Слика бр. 68. Изглед форме за претрагу одабраних тренинга

Основни сценарио СК

1. Радник уноси вредност по којој претражује све одабране тренинге. (АПУСО)



2. Радник позива систем да нађе све одабране тренинге по задатој вредности. (АПСО)
3. Систем тражи одабрани тренинге по задатој вредности. (СО)
4. Систем приказује раднику све одабране тренинге и поруку: “Систем је нашао све одабране тренинге по задатој вредности”. (ИА)

ID	Korisnik	Trening	DatumUpdate	DatumVazenja	Status
5	Ana Zdravkovic	Aerobik za zene	10.01.2019.	24:00	Važeći
1	Ivana Popovic	Aerobik za zene	27.01.2019.	24:00	Važeći
2	Jovan Jovovic	Aerobik za zene	31.01.2019.	24:00	Važeći

Слика бр. 67. Изглед након успешне претраге одабраних тренинга

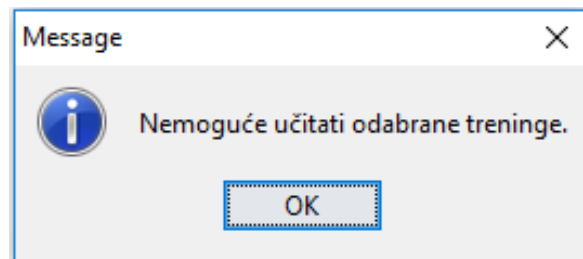
5. Радник бира одабрани тренинг коју жели да измени. (АПУСО)
6. Радник позива систем да учита податке о одабраном тренингу. (АПСО)
7. Систем проналази податке о том одабраном тренингу. (СО)
8. Систем приказује раднику податке о одабраном тренингу и поруку: “Систем је успешно пронашао податке о одабраном тренингу“. (ИА)

Redni broj	Datum	Vreme
1	11.01.2019.	18:00
2	13.01.2019.	18:00
3	15.01.2019.	20:00
4	17.01.2019.	20:00
5	19.01.2019.	18:00
6	21.01.2019.	18:00

Слика бр. 68. Изглед форме за приказ одабраних тренинга

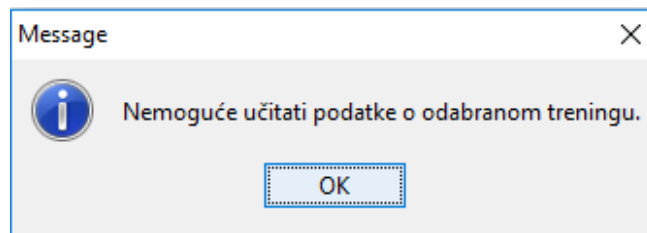
Алтернативна сценарија

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



Слика бр. 69. Изглед након неуспешне претраге одабраних тренинга

- 8.1 Уколико систем не може да учита податке о одабраном тренингу он приказује раднику поруку “Систем не може да прикаже податке о одабраном тренингу ”. Прекида се извршење сценарија. (ИА)



Слика бр. 70. Изглед након неуспешног учитавања одабраног тренинга

Пројектовање контролера корисничког интерфејса

Контролер корисничког интерфејса је одговоран за:

- прихватање графичких објеката од екранске форме,
- конвертовање података који се налазе у графичким објектима у доменске објекте који ће бити прослеђени преко мреже до апликационог сервера
- конвертовање доменских објеката у графичке објекте и прослеђивање до екранске форме.

У овом раду коришћен је један контролер којег користе све екранске форме.

```
public class Kontroler {
    private static Kontroler instance;
    private Kontroler() {
    }

    public static Kontroler getInstance() {
        if (instance == null) {
            instance = new Kontroler();
        }
        return instance;
    }

    public Object prijavaNaSistem(String username, String password) throws Exception {
        Radnik r = new Radnik();
        r.setKorisnickoIme(username);
        r.setSifra(password);
        ZahtevKlijenta zk = new ZahtevKlijenta(operacija.Operacija.LOGIN, r);
        komunikacija.Komunikacija.getInstance().posalji(zk);
        OdgovorServera odg = komunikacija.Komunikacija.getInstance().primi();
        System.out.println(odg.getGreska() + " " + odg.getStatus() + " " + odg.getOdgovor());
        if (odg.getStatus() == StatusOperacije.OK) {
            sesija.Sesija.getInstance().setRadnik((Radnik) odg.getOdgovor());
            return (Radnik) odg.getOdgovor();
        } else {
            System.out.println((String) odg.getGreska());
            throw new Exception((String) odg.getOdgovor());
        }
    }

    public Object unosKorisnika(String imePrezime, String email, String telefon, Radnik radnik) throws
    Exception {
        radnik = sesija.Sesija.getInstance().getRadnik();
        Korisnik korisnik = new Korisnik(imePrezime, email, telefon, radnik);
        ZahtevKlijenta zk = new ZahtevKlijenta(operacija.Operacija.UNOS_KORISNIKA, korisnik);
        komunikacija.Komunikacija.getInstance().posalji(zk);
        OdgovorServera odg = komunikacija.Komunikacija.getInstance().primi();
        if (odg.getStatus() == StatusOperacije.OK) {
```



```

        return odg.getOdgovor();
    } else {
        System.out.println((String) odg.getOdgovor());
        throw new Exception((String) odg.getGreska());
    }
}

public Object registrujRadnika(String imePrezime, String korisnickoIme, String sifra, String telefon)
throws Exception {
    Radnik rad = new Radnik(imePrezime, korisnickoIme, sifra, telefon);
    ZahtevKlijenta zk = new ZahtevKlijenta(operacija.Operacija.REGISTRACIJA, rad);
    komunikacija.Komunikacija.getInstance().posalji(zk);
    OdgovorServera odg = komunikacija.Komunikacija.getInstance().primi();
    if (odg.getStatus() == StatusOperacije.OK) {
        return odg.getOdgovor();
    } else {
        System.out.println((String) odg.getGreska());
        throw new Exception((String) odg.getOdgovor());
    }
}

public Korisnik izmeniKorisnika(Korisnik k) throws Exception {
    ZahtevKlijenta zk = new ZahtevKlijenta(operacija.Operacija.IZMENA_KORISNIKA, k);
    komunikacija.Komunikacija.getInstance().posalji(zk);
    OdgovorServera odg = komunikacija.Komunikacija.getInstance().primi();
    if (odg.getStatus() == StatusOperacije.OK) {
        return (Korisnik) odg.getOdgovor();
    } else {
        System.out.println((String) odg.getGreska());
        throw new Exception((String) odg.getOdgovor());
    }
}

public ArrayList<Korisnik> pretraziKorisnike(String id, String imePrezime, String radnikUneo) throws
Exception {
    Korisnik k = new Korisnik();
    if (!id.trim().isEmpty()) {
        try {
            int idINT = Integer.parseInt(id);
            k.setKorisnikID(idINT);
        } catch (Exception ex) {
            throw new Exception("ID mora biti broj!");
        }
    }
    if (!imePrezime.trim().isEmpty()) {
        k.setImePrezime(imePrezime);
    }
}

```

```

    }
    if (!radnikUneo.trim().isEmpty()) {
        k.setRadnik(new Radnik(radnikUneo, null, null, null));
    }
    ZahtevKlijenta zk = new ZahtevKlijenta(operacija.Operacija.PRETRAGA_KORISNIKA, k);
    komunikacija.Komunikacija.getInstance().posalji(zk);
    OdgovorServera odg = komunikacija.Komunikacija.getInstance().primi();
    if (odg.getStatus() == StatusOperacije.OK) {
        return (ArrayList<Korisnik>) odg.getOdgovor();
    } else {
        System.out.println((String) odg.getGreska());
        throw new Exception((String) odg.getOdgovor());
    }
}

```

```

public Object unesiTrening(String naziv, String opis, int broj, double cenaD) throws Exception {
    Trening trening = new Trening(naziv, opis, broj, cenaD);
    ZahtevKlijenta zk = new ZahtevKlijenta(operacija.Operacija.UNOS_TRENINGA, trening);
    komunikacija.Komunikacija.getInstance().posalji(zk);
    OdgovorServera odg = komunikacija.Komunikacija.getInstance().primi();
    if (odg.getStatus() == StatusOperacije.OK) {
        return odg.getOdgovor();
    } else {
        System.out.println((String) odg.getGreska());
        throw new Exception((String) odg.getOdgovor());
    }
}

```

```

public ArrayList<Trening> pretraziTreninge(String id, String naziv, String opis, String cena) throws
Exception {
    Trening trening = new Trening();
    if (!id.trim().isEmpty()) {
        try {
            int idINT = Integer.parseInt(id);
            trening.setTreningID(idINT);
        } catch (Exception ex) {
            throw new Exception("ID mora biti broj!");
        }
    }
    if (!cena.trim().isEmpty()) {
        try {
            double cenaN = Double.parseDouble(cena);
            trening.setCena(cenaN);
        } catch (Exception e) {
            throw new Exception("Cena mora biti broj");
        }
    }
}

```

```

        if (!naziv.trim().isEmpty()) {
            trening.setNaziv(naziv);
        }
        if (!opis.trim().isEmpty()) {
            trening.setOpis(opis);
        }
        ZahtevKlijenta zk = new ZahtevKlijenta(operacija.Operacija.PRETRAGA_TRENINGA, trening);
        komunikacija.Komunikacija.getInstance().posalji(zk);
        OdgovorServera odg = komunikacija.Komunikacija.getInstance().primi();
        if (odg.getStatus() == StatusOperacije.OK) {
            return (ArrayList<Trening>) odg.getOdgovor();
        } else {
            System.out.println((String) odg.getGreska());
            throw new Exception((String) odg.getOdgovor());
        }
    }
}

public Trening izmeniTrening(Trening trening) throws Exception {
    ZahtevKlijenta zk = new ZahtevKlijenta(operacija.Operacija.IZMENA_TRENINGA, trening);
    komunikacija.Komunikacija.getInstance().posalji(zk);
    OdgovorServera odg = komunikacija.Komunikacija.getInstance().primi();
    if (odg.getStatus() == StatusOperacije.OK) {
        return (Trening) odg.getOdgovor();
    } else {
        System.out.println((String) odg.getGreska());
        throw new Exception((String) odg.getOdgovor());
    }
}

public ArrayList<Trening> vratiSveTreninge() throws Exception {
    ZahtevKlijenta zk = new ZahtevKlijenta(operacija.Operacija.VRATI_SVE_TRENINGE, null);
    komunikacija.Komunikacija.getInstance().posalji(zk);
    OdgovorServera odg = komunikacija.Komunikacija.getInstance().primi();
    if (odg.getStatus() == StatusOperacije.OK) {
        return (ArrayList<Trening>) odg.getOdgovor();
    } else {
        System.out.println((String) odg.getGreska());
        throw new Exception((String) odg.getOdgovor());
    }
}

public ArrayList<Korisnik> vratiSveKorisnike() throws Exception {
    ZahtevKlijenta zk = new ZahtevKlijenta(operacija.Operacija.VRATI_SVE_KORISNIKE, null);
    komunikacija.Komunikacija.getInstance().posalji(zk);
    OdgovorServera odg = komunikacija.Komunikacija.getInstance().primi();
    if (odg.getStatus() == StatusOperacije.OK) {
        return (ArrayList<Korisnik>) odg.getOdgovor();
    }
}

```

```

    } else {
        System.out.println((String) odg.getGreska());
        throw new Exception((String) odg.getOdgovor());
    }
}

public Object unesiOdabraniTrening(Date datumUplate, Date datumVaz, String status, Korisnik kor,
Trening tren, ArrayList<Termin> listaTermina) throws Exception {
    OdabraniTrening ot = new OdabraniTrening(0, datumUplate, datumVaz, status, tren, kor,
listaTermina);
    ZahtevKlijenta zk = new ZahtevKlijenta(operacija.Operacija.UNOS_ODABRANOG_TRENINGA,
ot);
    komunikacija.Komunikacija.getInstance().posalji(zk);
    OdgovorServera odg = komunikacija.Komunikacija.getInstance().primi();
    if (odg.getStatus() == StatusOperacije.OK) {
        return odg.getOdgovor();
    } else {
        System.out.println((String) odg.getGreska());
        throw new Exception((String) odg.getOdgovor());
    }
}

public ArrayList<OdabraniTrening> pretraziOdabraneTreninge(Korisnik kor, Trening tr, String status)
throws Exception {
    OdabraniTrening ot = new OdabraniTrening();
    if (kor != null) {
        ot.setKorisnik(kor);
    }
    if (tr != null) {
        ot.setTrening(tr);
    }
    if (!status.trim().isEmpty()) {
        ot.setStatus(status);
    }
    ZahtevKlijenta zk = new
ZahtevKlijenta(operacija.Operacija.PRETRAGA_ODABRANOG_TRENINGA, ot);
    komunikacija.Komunikacija.getInstance().posalji(zk);
    OdgovorServera odg = komunikacija.Komunikacija.getInstance().primi();
    if (odg.getStatus() == StatusOperacije.OK) {
        return (ArrayList<OdabraniTrening>) odg.getOdgovor();
    } else {
        System.out.println((String) odg.getGreska());
        throw new Exception((String) odg.getOdgovor());
    }
}

public Object izmeniOdabraniTrening(OdabraniTrening ot) throws Exception {

```

```

        ZahtevKlijenta zk = new ZahtevKlijenta(operacija.Operacija.IZMENA_ODABRANOG_TRENING,
ot);
        komunikacija.Komunikacija.getInstance().posalji(zk);
        OdgovorServera odg = komunikacija.Komunikacija.getInstance().primi();
        if (odg.getStatus() == StatusOperacije.OK) {
            return odg.getOdgovor();
        } else {
            System.out.println((String) odg.getGreska());
            throw new Exception((String) odg.getOdgovor());
        }
    }

    public void odajvaSaSistema() throws Exception {
        ZahtevKlijenta zk = new ZahtevKlijenta(operacija.Operacija.LOGOUT,
sesija.Sesija.getInstance().getRadnik());
        komunikacija.Komunikacija.getInstance().posalji(zk);
        OdgovorServera odg = komunikacija.Komunikacija.getInstance().primi();
        if (odg.getStatus() == StatusOperacije.KRAJ_RADAK) {
            return;
        } else {
            System.out.println((String) odg.getGreska());
            throw new Exception((String) odg.getOdgovor());
        }
    }
}

```

Пројектовање апликационе логике

Апликациони сервери треба да обезбеде сервисе који ће омогућити реализацију апликационе логике софтверског система. Пројектовани апликациони сервер садржи:

- део за комуникацију са клијентом
- контролер апликационе логике
- део за комуникацију са складиштем података (брокер базе података)
- део који садржи пословну логику.

Комуникација са клијентом

Део за комуникацију подиже серверски сокет који даље ослушкује мрежу. Када клијент успостави конекцију, сервер генерише нит која ће бити одговорна за двосмерну везу са клијентом. Клијент шаље захтев за извршење неке од системских операција, одговарајућа нит (додељена клијенту) прихвата захтев и прослеђује га до контролера апликационе логике. Након извршења системске операције, контролер враћа резултат извршења операције до нити која је задужена за тог клијента. Резултат се затим прослеђује клијенту. Комуникација између клијента и сервера се обавља разменом објекта класе **OdgovorServera** и класе **ZahtevKlijenta**. Операције које се користе приликом комуникације клијента и

сервера налазе се у класи **Operacije**. Статус операција које се шаљу кроз одговоре сервера пописане су помоћу еnumerације **StatusOperacije**.

```
public class OdgovorServera implements Serializable {

    private Object odgovor;
    private Object greska;
    private StatusOperacije status;

    public OdgovorServera() {
    }

    public OdgovorServera(Object odgovor, Object greska, StatusOperacije status) {
        this.odgovor = odgovor;
        this.greska = greska;
        this.status = status;
    }

    public Object getOdgovor() {
        return odgovor;
    }

    public void setOdgovor(Object odgovor) {
        this.odgovor = odgovor;
    }

    public StatusOperacije getStatus() {
        return status;
    }

    public void setStatus(StatusOperacije status) {
        this.status = status;
    }

    public Object getGreska() {
        return greska;
    }

    public void setGreska(Object greska) {
        this.greska = greska;
    }

}
```

```

public class ZahtevKlijenta implements Serializable {

    private int operacija;
    private Object parametar;

    public ZahtevKlijenta() {
    }

    public ZahtevKlijenta(int operacija, Object parametar) {
        this.operacija = operacija;
        this.parametar = parametar;
    }

    public Object getParametar() {
        return parametar;
    }

    public void setParametar(Object parametar) {
        this.parametar = parametar;
    }

    public int getOperacija() {
        return operacija;
    }

    public void setOperacija(int operacija) {
        this.operacija = operacija;
    }

}

public class Operacija {

    public static final int LOGIN = 0;
    public static final int LOGOUT = 1;
    public static final int REGISTRACIJA = 2;
    public static final int UNOS_KORISNIKA = 3;
    public static final int UNOS_TRENINGA = 4;
    public static final int PRETRAGA_KORISNIKA = 5;
    public static final int PRETRAGA_TRENINGA = 6;
    public static final int IZMENA_KORISNIKA = 7;
    public static final int IZMENA_TRENINGA = 8;
    public static final int VRATI_SVE_TRENINGE = 9;
    public static final int VRATI_SVE_KORISNIKE = 10;
    public static final int UNOS_ODABRANOG_TRENINGA = 11;
    public static final int PRETRAGA_ODABRANOG_TRENINGA = 12;
    public static final int IZMENA_ODABRANOG_TRENING = 13;

}

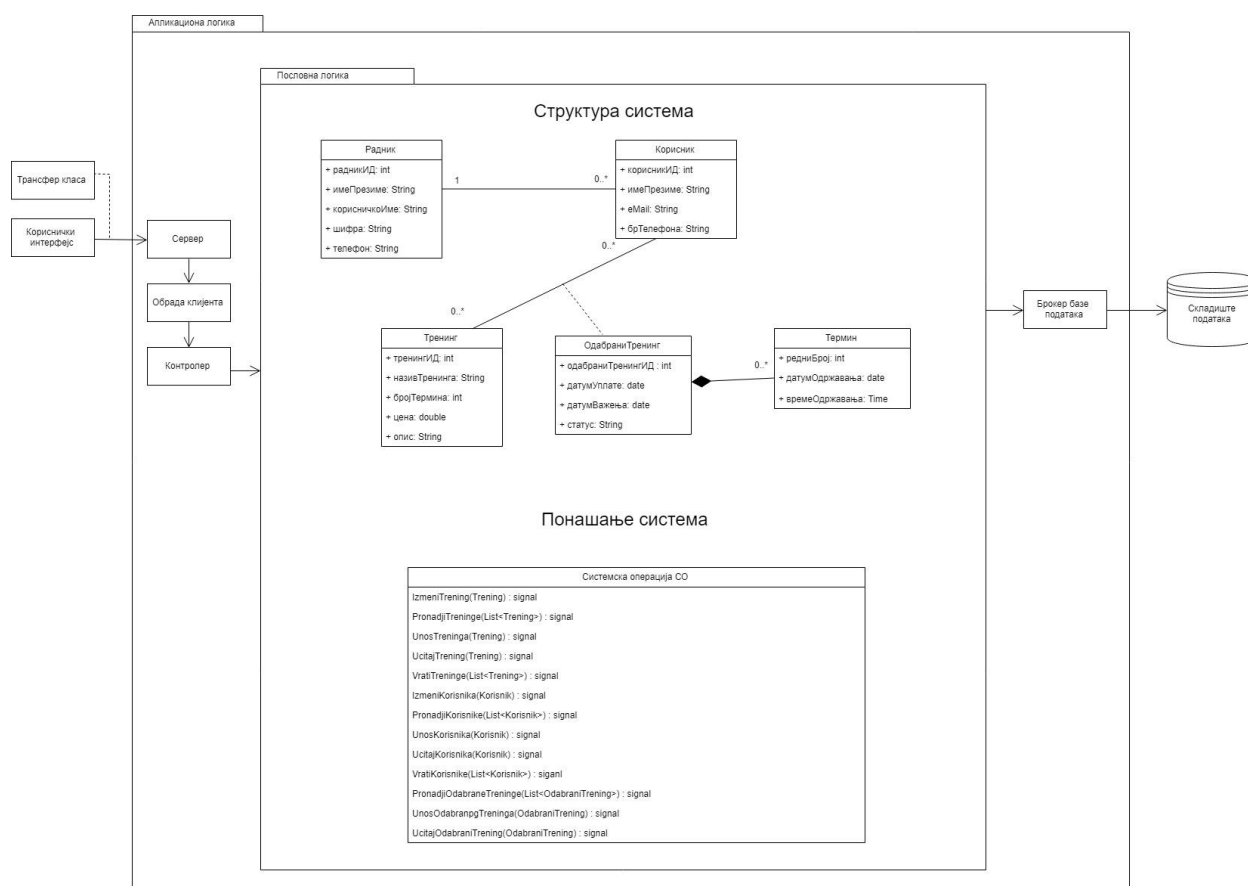
public enum StatusOperacije {
    OK, ERROR, KRAJ_RADA;
}

```

Контролер апликационе логике

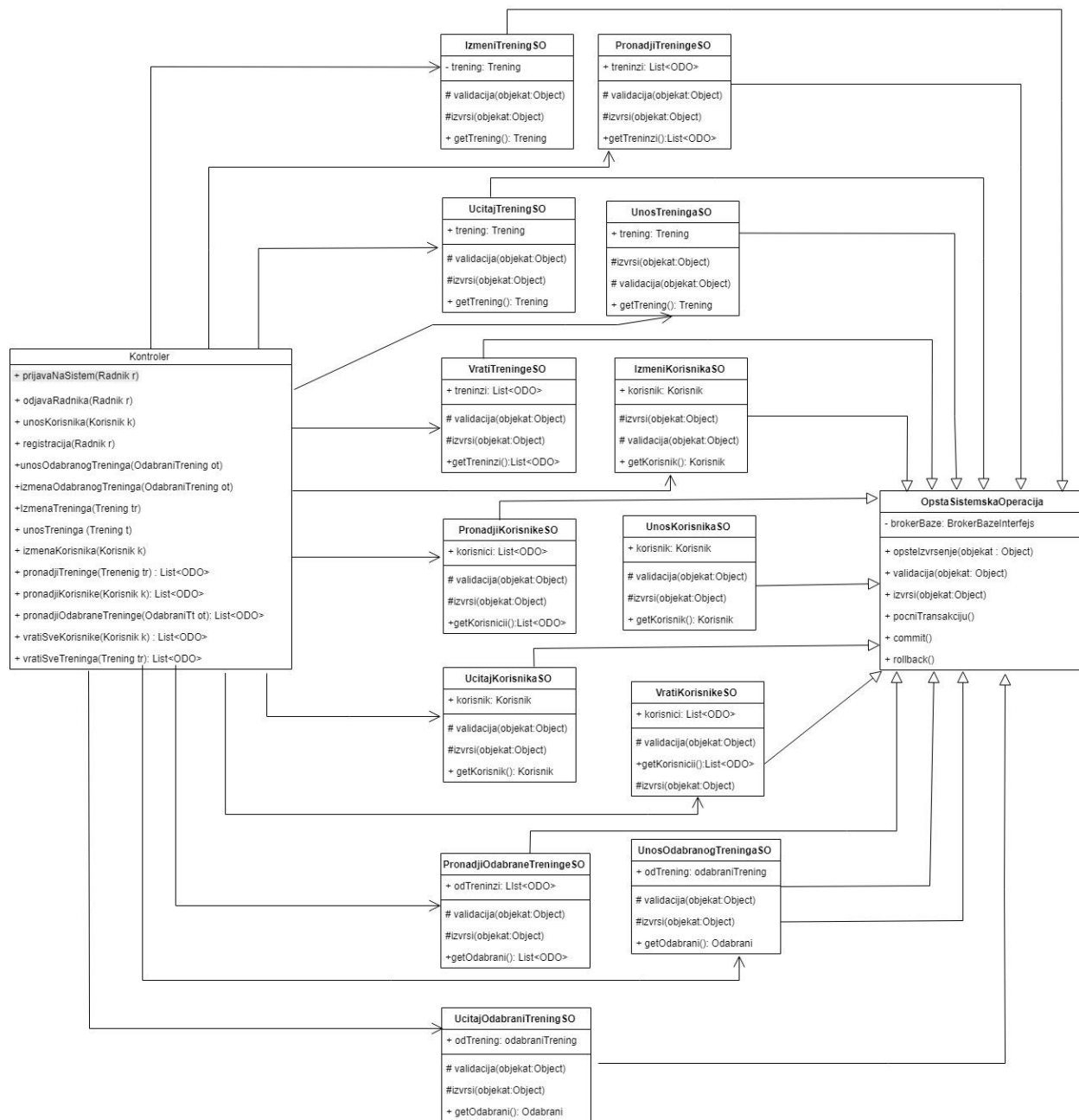
Контролер апликационе логике прихвата захтеве за извршење системских операција и исте прослеђује до конкретне системске операције. Након извршења системске операције, контролер прихвата одговор и враћа назад позиваоцу (нити клијента).

Како је у фазама прикупљања захтева и анализе дата спецификација структуре и понашања софтверског система, односно спецификација пословне логике софтверског система, следећа слика даје опис система након фазе пројектовања комуникације са клијентима и контролера апликационе логике.



Слика бр. 71. Архитектура софтверског система након пројектовања контролера апликационе логике

Преко класе **Kontroler** прихватају се захтеви од нити клијента за извршење системских операција, који се затим прослеђују до одговарајућих класа које су одговорне за извршење системских операција. За сваку системску операцију праве се софтверске класе које треба да реализују системску операцију.



Слика бр. 72. Класе које имплементирају системске операције наслеђују класу OpstaSO

```

public abstract class OpstaSistemskaOperacija {

    protected BrokerBazeInterfejs brokerBaze;

    public OpstaSistemskaOperacija() {
        brokerBaze = new BrokerBaze();
    }

    public final void opsteIzvršenje(Object objekat) throws Exception {
        try {
            validacija(objekat);
            try {
                pocniTransakciju();
                izvrši(objekat);
                commit();
            } catch (Exception ex) {
                rollback();
                throw ex;
            }
        } catch (Exception ex) {
            ex.printStackTrace();
            throw ex;
        }
    }

    protected abstract void validacija(Object objekat) throws Exception;

    private void pocniTransakciju() throws Exception {
        BazaKonekcija.getInstance().pocniTransakciju();
    }

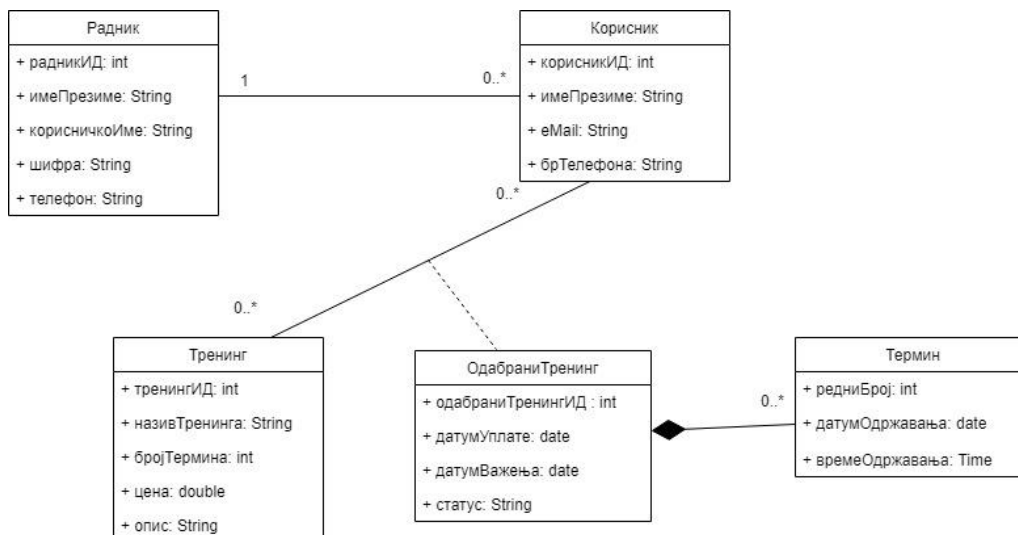
    protected abstract void izvrši(Object objekat) throws Exception;

    private void commit() throws Exception {
        BazaKonekcija.getInstance().commit();
    }

    private void rollback() throws Exception {
        BazaKonekcija.getInstance().rollback();
    }
}

```

Проектовање структуре софтверског система



Слика бр. 73. Концептуални модел система

Свака класа има приватна поља атрибута, геттере и сеттере за те атрибуте, беспараметарски конструктор, као и параметарски конструктор. Пример класа **Korisnik**:

```
public class Korisnik implements OpstiDomenskiObjekat, Serializable {

    private int korisnikID;
    private String imePrezime;
    private String eMail;
    private String telefon;
    private Radnik radnik;

    public Korisnik() {
    }

    @Override
    public String toString() {
        return imePrezime;
    }

    public Korisnik(int korisnikID, String imePrezime, String eMail, String telefon, Radnik radnik) {
        this.korisnikID = korisnikID;
        this.imePrezime = imePrezime;
        this.eMail = eMail;
        this.telefon = telefon;
        this.radnik = radnik;
    }

    public Korisnik(String imePrezime, String eMail, String telefon, Radnik radnik) {
        this.imePrezime = imePrezime;
        this.eMail = eMail;
        this.telefon = telefon;
        this.radnik = radnik;
    }
}
```

```

public int getKorisnikID() {
    return korisnikID;
}

public void setKorisnikID(int korisnikID) {
    this.korisnikID = korisnikID;
}

public String getImePrezime() {
    return imePrezime;
}

public void setImePrezime(String imePrezime) {
    this.imePrezime = imePrezime;
}

public String geteMail() {
    return eMail;
}

public void seteMail(String eMail) {
    this.eMail = eMail;
}

public String getTelefon() {
    return telefon;
}

public void setTelefon(String telefon) {
    this.telefon = telefon;
}

public Radnik getRadnik() {
    return radnik;
}

public void setRadnik(Radnik radnik) {
    this.radnik = radnik;
}

@Override
public String vratiVrednostiAtributa() {
    return korisnikID + ", " + (imePrezime == null ? null : "" + imePrezime + "") + ", " + (eMail == null
? null : "" + eMail + "") + ", " + (telefon == null ? null : "" + telefon + "") + ", " + (radnik == null ? null :
"" + radnik.getRadnikID() + "");
}

@Override
public String postaviVrednostAtrbuta() {
    return "korisnikID=" + korisnikID + ", imePrezime=" + (imePrezime == null ? null : "" + imePrezime
+ "") + ", email=" + (eMail == null ? null : "" + eMail + "") + ", telefon=" + (telefon == null ? null : "" +
telefon + "") + ", radnikID=" + (radnik == null ? null : "" + radnik.getRadnikID() + "");
}

```

```

@Override
public String vratiImeKlase() {
    return "Korisnik";
}

@Override
public String vratiUslovZaNadjiSlog() {
    return "korisnikID=" + korisnikID;
}

@Override
public String vratiUslovZaNadjiSlogove() {
    String rez = "";
    if (imePrezime != null) {
        rez += "k.imePrezime=" + imePrezime + " AND ";
    }
    if (korisnikID != 0) {
        rez += "k.korisnikID=" + korisnikID + " AND ";
    }
    if (radnik != null && radnik.getImePrezime() != null) {
        rez += "r.imePrezime LIKE '%" + radnik.getImePrezime() + "%' AND ";
    }
    rez = rez.substring(0, rez.length() - 4);
    return "where " + rez;
}

@Override
public String vratiImeKolone(int brojKolone) {
    String[] kolone = {"korisnikID", "imePrezime", "email", "telefon", "radnikID"};
    return kolone[brojKolone];
}

@Override
public ArrayList<OpstiDomenskiObjekat> napuniSve(ResultSet rs) throws Exception {
    ArrayList<OpstiDomenskiObjekat> lista = new ArrayList<>();
    try {
        while (rs.next()) {
            int id = rs.getInt("k.korisnikID");
            String ime = rs.getString("k.imePrezime");
            String eMail = rs.getString("k.email");
            String telefon = rs.getString("k.telefon");
            int radnikID = rs.getInt("k.radnikID");
            String ImeRadnika = rs.getString("r.imePrezime");
            String korIme = rs.getString("r.username");
            String sifra = rs.getString("r.password");
            String telefonRadnika = rs.getString("r.telefon");
            Radnik r = new Radnik(radnikID, ImeRadnika, korIme, sifra, telefonRadnika);
            Korisnik k = new Korisnik(id, ime, eMail, telefon, r);
            lista.add(k);
        }
        return lista;
    } catch (Exception ex) {

```

```

        throw new Exception("Greska prilikom uzimanja vrednosti iz baze.");
    }
}

@Override
public void ubaci(String nazivAtributa, Object vrednostAtributa) {
    switch (nazivAtributa) {
        case "korisnikID": setKorisnikID((int) vrednostAtributa);
            break;
        case "imePrezime": setImePrezime((String) vrednostAtributa);
            break;
        case "eMail": seteMail((String) vrednostAtributa);
            break;
        case "telefon": setTelefon((String) vrednostAtributa);
            break;
        case "radnikID": setRadnik((Radnik) vrednostAtributa);
            break;
    }
}

@Override
public String vratiUslovZaSortiranje() {
    return "ORDER BY k.imePrezime ASC";
}

@Override
public String vratiTabeluSaUslovomSpajanja() {
    return " k INNER JOIN radnik r ON k.radnikID = r.radnikID";
}

@Override
public OpstiDomenskiObjekat napuni(ResultSet rs) throws Exception {
    Korisnik k = new Korisnik();
    try {
        while (rs.next()) {
            int id = rs.getInt("k.korisnikID");
            String ime = rs.getString("k.imePrezime");
            String eMail = rs.getString("k.email");
            String telefon = rs.getString("k.telefon");
            int radnikID = rs.getInt("k.radnikID");
            String ImeRadnika = rs.getString("r.imePrezime");
            String korIme = rs.getString("r.username");
            String sifra = rs.getString("r.password");
            String telefonRadnika = rs.getString("r.telefon");
            Radnik r = new Radnik(radnikID, ImeRadnika, korIme, sifra, telefonRadnika);
            k = new Korisnik(id, ime, eMail, telefon, r);
        }
        return k;
    } catch (Exception ex) {
        throw new Exception("Neusplo citanje iz baze");
    }
}
}

```

Пројектовање системских операција

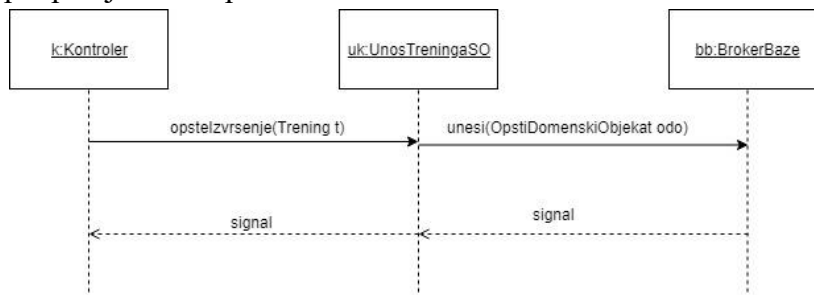
Уговор УГ1: KreirajTrening(Trening)

Операција: KreirajTrening(Trening): signal;

Веза са СК: СК1

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

Постуслови: Креиран је нови тренинг.



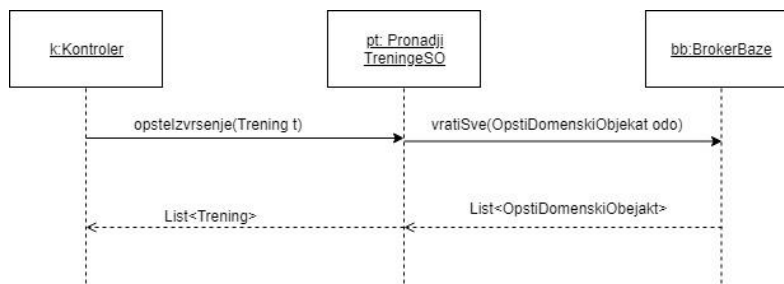
Уговор УГ2: PronadjiTreninge(ZadataVrednost, List<Trening>)

Операција: PronadjiTreninge(ZadataVrednost, List<Trening>): signal;

Веза са СК: СК2, СК3

Предуслови: /

Постуслови: /



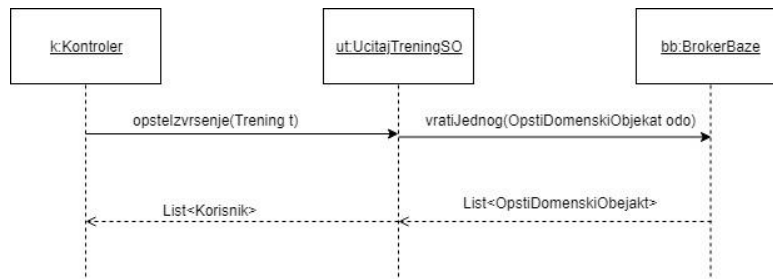
Уговор УГ3: UcitajTrening(Trening)

Операција: UcitajTrening(Trening): signal;

Веза са СК: СК2, СК3

Предуслови: /

Постуслови: /



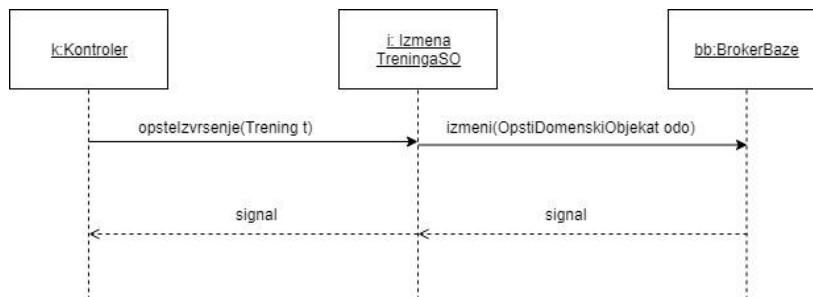
Уговор УГ4: PromeniTrening(Trening)

Операција: PromeniTreninge(Trening): signal;

Веза са СК: СК3

Предуслови: Тренинг мора постојати у систему да би био измењен. Вредносна и структурна ограничења над објектом *Trening* морају бити задовољена

Постуслови: Измењени су подаци о тренингу.



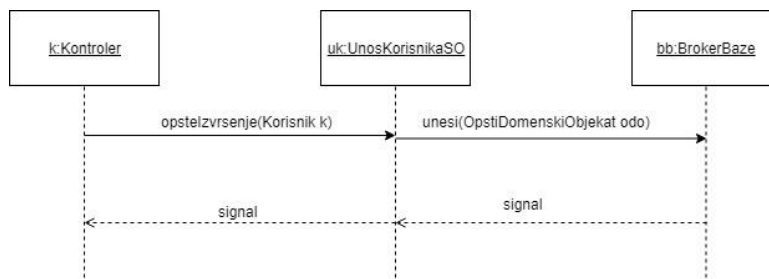
Уговор УГ5: KreirajKorisnika(Korisnik)

Операција: KreirajKorisnika(Korisnik): signal;

Веза са СК: СК4

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

Постуслови: Креиран је нови корисник.



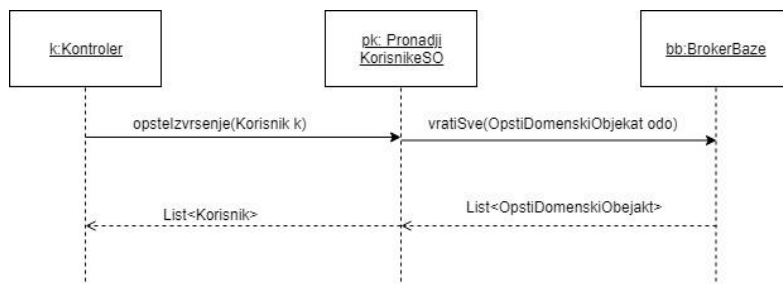
Уговор УГ6: PronadjiKorisnike(ZadataVrednost, List<Korisnik>)

Операција: PronadjiKorisnike(ZadataVrednost): signal;

Веза са СК: СК4, СК5, СК6

Предуслови: /

Постуслови: /



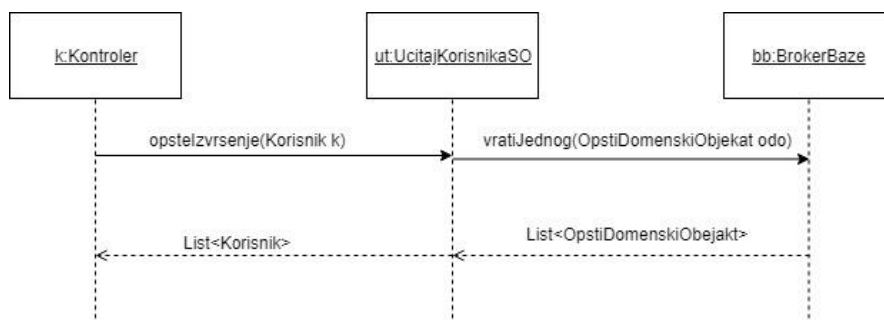
Уговор УГ7: UcitajKorisnika(Korisnik)

Операција: UcitajKorisnika(Korisnik): signal;

Веза са СК: СК4, СК5, СК6

Предуслови: /

Постуслови: /



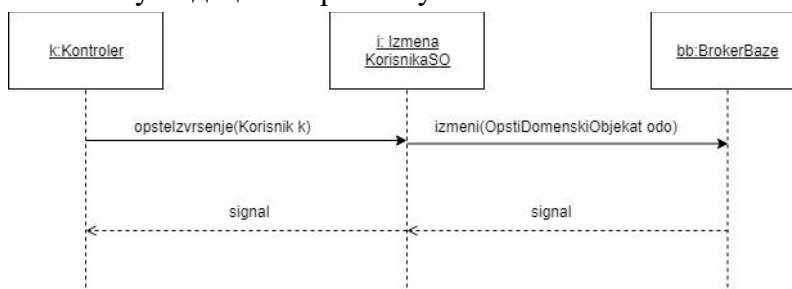
Уговор УГ8: PromeniKorisnika(Korisnik)

Операција: PromeniKorisnika(Korisnik): signal;

Веза са СК: СК6

Предуслови: Корисник мора постојати у систему да би био измењен. Вредносна и структурна ограничења над објектом *Korisnik* морају бити задовољена

Постуслови: Измењени су подаци о корсинику.



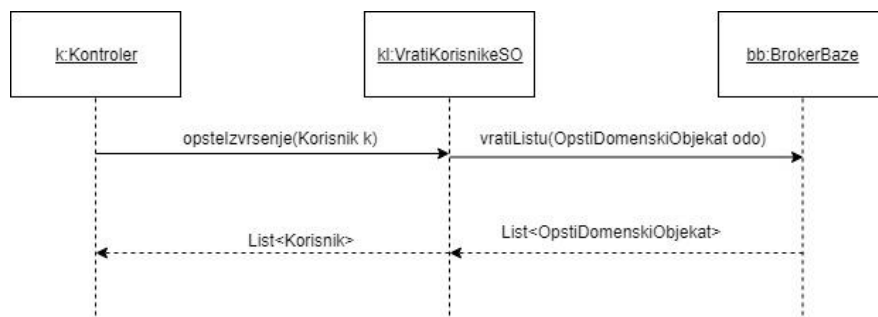
Уговор УГ9: VratilistuKorisnika(List<Korisnik>)

Операција: VratilistuKorisnika(List<Korisnik>): signal;

Веза са СК: СК7

Предуслови: /

Постуслови: /



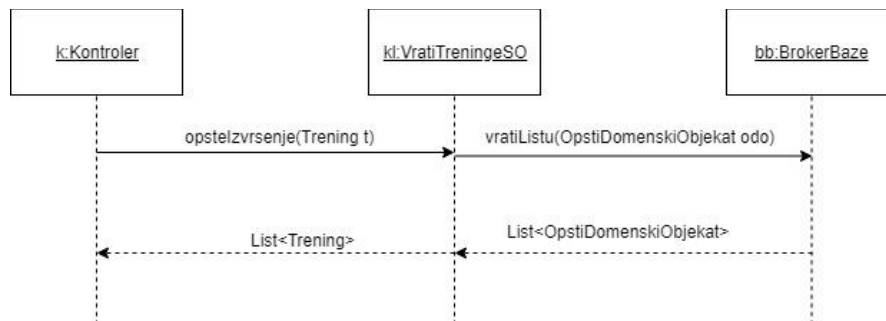
Уговор УГ10: VратиListuTreninga(List<Trening>)

Операција: VратиListuTreninga(List<Trening>): signal;

Веза са СК: СК7

Предуслови: /

Постуслови: /



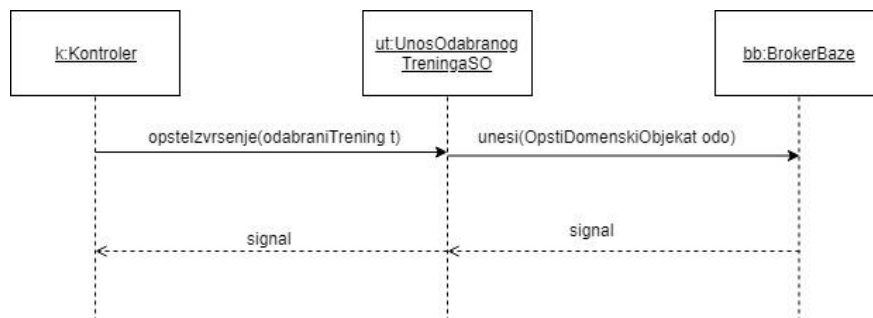
Уговор УГ11: KreirajOdabraniTrening(OdabraniTrening)

Операција: KreirajOdabraniTrening (OdabraniTrening): signal;

Веза са СК: СК7

Предуслови: Вредносна и структурна ограничења над објектом *OdabraniTrening* морају бити задовољена. У систему морају постојати клијенти и тренинзи.

Постуслови: Креиран је нови одабрани тренинг.



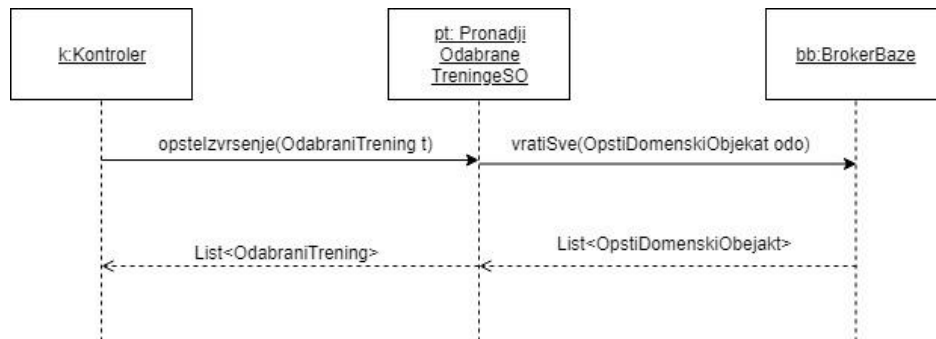
Уговор УГ12: PronadjiOdabraniTrening(ZadataVrednost, List<OdabraniTrening>)

Операција: PronadjiOdabraniTrening(ZadataVrednost, List<Trening>): signal;

Веза са СК: CK8

Предуслови: /

Постуслови: /



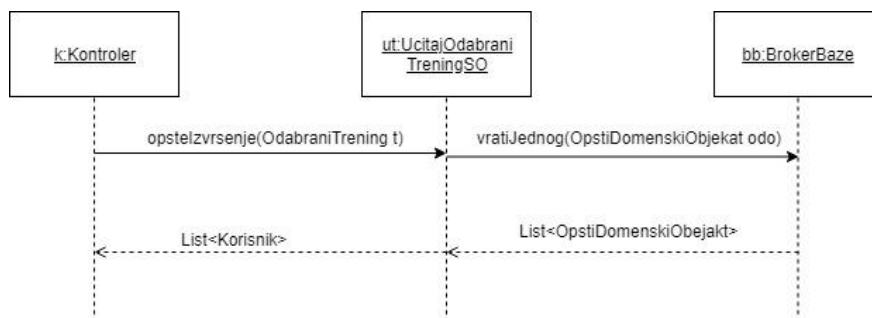
Уговор УГ13: UcitajOdabraniTrening(OdabraniTrening)

Операција: UcitajOdabraniTrening(OdabraniTrening): signal;

Веза са СК: CK8

Предуслови: /

Постуслови: /



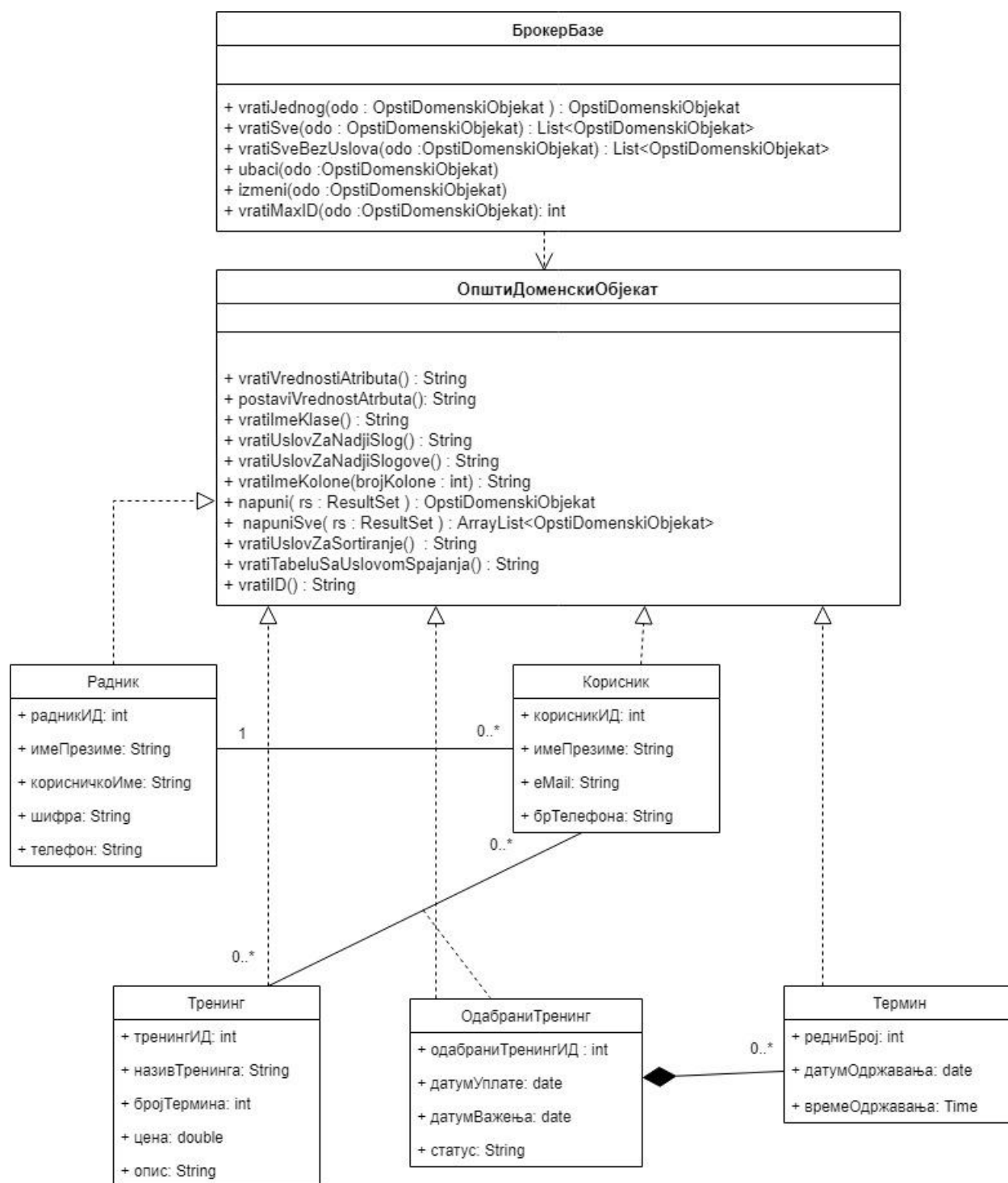
Брокер базе података

Класа **BrokerBazeInterfejs** се пројектује како би се обезбедио перзистентни сервис објектима доменских класа који се чувају у бази података. Ову класу наслеђује класа **BrokerBaze**. Ова класа посредује у свим операцијама над базом података и реализује следеће методе:

```
public OpstiDomenskiObjekat vratiJednog(OpstiDomenskiObjekat odo);  
public List<OpstiDomenskiObjekat> vratiSve(OpstiDomenskiObjekat odo);  
public List<OpstiDomenskiObjekat> vratiSveBezUslova(OpstiDomenskiObjekat odo);  
public void ubaci(OpstiDomenskiObjekat odo);  
public void izmeni(OpstiDomenskiObjekat odo);  
public int vratiMaxID(OpstiDomenskiObjekat odo);
```

Све методе **BrokerBaze** класе су пројектоване као генеричке, што значи да могу да прихвате различите доменске објекте преко параметара како не би било потребе да се у самом **BrokerBaze** -у имплементирају појединачне методе за сваку доменску класу и беспотребно умножава код. Ово је остварено дефинисањем интерфејса **OpstiDomenskiObjekat** кога имплементирају (и све његове методе) све доменске класе:

```
public interface OpstiDomenskiObjekat extends Serializable {  
    String vratiVrednostiAtributa();  
    String postaviVrednostAtrbuta();  
    String vratiImeKlase();  
    String vratiUslovZaNadjiSlog();  
    String vratiUslovZaNadjiSlogove();  
    String vratiImeKolone(int brojKolone);  
    public OpstiDomenskiObjekat napuni(ResultSet rs) throws Exception;  
    public ArrayList<OpstiDomenskiObjekat> napuniSve(ResultSet rs) throws Exception;  
    public void ubaci(String nazivAtributa, Object vrednostAtributa);  
    public String vratiUslovZaSortiranje() throws Exception;  
    public String vratiTabeluSaUslovomSpajanja();  
    String vratiID();  
}
```



Пројектовање складишта података

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

Табела **Радник**:

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update
<input type="checkbox"/>	radnikID	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	imePrezime	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	username	varchar	30		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	password	varchar	30		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	telefon	varchar	30		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Табела **Корисник**:

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update
<input type="checkbox"/>	korisnikID	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	imePrezime	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	email	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	telefon	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	radnikID	int	11		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Табела **Тренинг**:

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update
<input type="checkbox"/>	treningID	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	nazivTreninga	varchar	100		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	opis	varchar	500		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	brojTermina	int	11		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	cena	double			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Табела **ОдабраниТренинг**:

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update
<input type="checkbox"/>	odabraniID	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	datumUplate	date			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	datumVazenja	date			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	status	varchar	30		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	korisnikID	int	11		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	treningID	int	11		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Табела **Термин**:

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	On Update
<input type="checkbox"/>	redniBroj	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	odabraniTreningID	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	datum	date			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	vreme	time			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Имплементација

Софтверски систем развијен је у програмском језику Java и пројектован је као клијент-сервер. Као развојно окружење коришћен је NetBeans IDE 8.2, а као систем за управљање базом података коришћен је SQLyog Community.

Читав систем је реализован у три пројекта: **TeretanaServer**, **TeretanaKlijent** и **TeretanaZajdnicki**. На основу архитектуре софтверског система добијене су следеће софтверске класе:

TeretanaServer:

- forma/FormaServer.java
- forma/model/TableModelUlogovani.java
- kontroler/Kontroler.java
- niti/NitKlijenta.java
- niti/PokretanjeServera.java
- podesavanja/PodesavanjaBaze.java
- skladiste/BrokerBazeInterfejs.java
- skladiste/baza/BrokerBaze.java
- skladiste/baza/konekcija/BazaKonekcija.java
- so/OpstaSistemskaOperacija.java
- so/korisnik/IzmenaKorisnikaSO.java
- so/korisnik/PronadjiKorisnikeSO.java
- so/korisnik/UcitajKorisnikaSO.java
- so/korisnik/UnosKorisnikaSO.java
- so/korisnik/VratiKorisnikeSO.java
- so/odabraniTrening/IzmenaOdabranogTreningaSO.java
- so/odabraniTrening/PronadjiOdabraniTreningSO.java
- so/odabraniTrening/UcitajOdabraniTreningSO.java
- so/odabraniTrening/UnosOdabraniTreningOS.java
- so/radnik/PrijavljivanjeSO.java
- so/radnik/RegistracijaSO.java
- so/trening/IzmenaTreningaSO.java
- so/trening/PronadjiTreningeSO.java
- so/trening/UcitajTreningSO.java
- so/trening/UnosTreningaSO.java
- so/trening/VratiTreningeSO.java
- start/Start.java

TeretanaKlijent:

- forma/Glavna/FormaGlavna.java
- forma/Korisnik/IzmenaKorisnika.java
- forma/Korisnik/PretragaKorisnika.java
- forma/Korisnik/PrikazKorisnik.java
- forma/Korisnik/UnosKorisnika.java
- forma/Korisnik/model/TableModelKorisnik.java
- forma/LoginReg/FormaLogin.java
- froma/LoginReg/Registracija.java
- forma/OdabraniTrening/IzmenaOdabranogTreninga.java
- forma/OdabraniTrening/PretragaOdabranogTreninga.java
- forma/OdabraniTrening/PrikazOdabranogTreninga.java
- forma/OdabraniTrening/UnosOdabranogTreninga.java
- forma/OdabraniTrening/model/TableModelOdabraniTreninzi.java
- forma/OdabraniTrening/TableModelTermini.java
- forma/Trening/IzmeniTrening.java
- forma/Trening/PretragaTreninga.java
- forma/Trening/PrikazTreninga.java
- forma/Trening/UnosTreninga.java
- forma/Trening/model/TableModelTrening.java
- komunikacija/Komunikacija.java
- kontroler/Kontroler.java
- sesija/Sesija.java
- start/Start.java

TeretanaZajednicki:

- domen/Korisnik.java
- domen/OdabraniTrening.java
- domen/OpstiDomenskiObjekat.java
- domen/Radnik.java
- domen/Termin.java
- domen/Trening.java
- operacija/Operacija.java
- operacija/StatusOperacije.java
- podesavanja/PodesavanjaKonekcije.java
- transfer/OdgovorSevera.java
- transfer/ZahtevKlijenta.java

Тестирање

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

Литература

проф. др Синиша Влајић, *Пројектовање софтвера*, Београд 2015.