Универзитет у Београду

Факултет организационих наука

Лабораторија за софтверско инжењерство

Семинарски рад из предмета

Пројектовање софтвера

Тема: Софтверски систем намењен за рад воћарског друштва

“ВоћариПН” у .NET окружењу

Професор: Студент:

проф. др Синиша Влајић Маријана Милошевић 2017/1104

Београд, 2020. година

Садржај

[Кориснички захтеви 1](#_Toc100023017)

[Вербални опис 1](#_Toc100023018)

[1. СЛУЧАЈЕВИ КОРИШЋЕЊА 2](#_Toc100023019)

[СК1: Случај коришћења – Пријављивање корисника на систем 3](#_Toc100023020)

[СК2: Случај коришћења – Унос новог власника 4](#_Toc100023021)

[СК3: Случај коришћења – Претрага власника 5](#_Toc100023022)

[СК4: Случај коришћења – Измена података о власнику 6](#_Toc100023023)

[СК5: Случај коришћења – Брисање власника 7](#_Toc100023024)

[СК6: Случај коришћења – Унос нове парцеле 8](#_Toc100023025)

[СК7: Случај коришћења – Претрага парцела 9](#_Toc100023026)

[СК8: Случај коришћења – Измена података о парцели 10](#_Toc100023027)

[СК9: Случај коришћења – Унос новог купца 11](#_Toc100023028)

[СК10: Случај коришћења – Унос нове евиденције о продаји 12](#_Toc100023029)

[СК11: Случај коришћења – Претрага евиденција о продаји 13](#_Toc100023030)

[2. АНАЛИЗА 14](#_Toc100023031)

[2.1 Понашање софтверског система - Дијаграм секвенци случаја коришћења 14](#_Toc100023032)

[ДС1: Дијаграм секвенци случаја коришћења - Пријављивање на систем 15](#_Toc100023033)

[ДС2: Дијаграм секвенци случаја коришћења - Унос новог власника 16](#_Toc100023034)

[ДС3: Дијаграм секвенци случаја коришћења - Претрага власника 18](#_Toc100023035)

[ДС4: Дијаграм секвенци случаја коришћења – Измена података о власнику 20](#_Toc100023036)

[ДС5: Дијаграм секвенци случаја коришћења - Брисање власника 23](#_Toc100023037)

[ДС6: Дијаграм секвенци случаја коришћења – Унос нове парцеле 26](#_Toc100023038)

[ДС7: Дијаграм секвенци случаја коришћења – Претрага парцела 29](#_Toc100023039)

[ДС8: Дијаграм секвенци случаја коришћења – Измена парцеле 31](#_Toc100023040)

[ДС9: Дијаграм секвенци случаја коришћења – Унос новог купца 34](#_Toc100023041)

[ДС10: Дијаграм секвенци случаја коришћења – Унос нове евиденције продаје 36](#_Toc100023042)

[ДС11 : Дијаграм секвенци случаја коришћења – Претага евиденција продаје 38](#_Toc100023043)

[2.2 Понашање софтверског система – Дефинисање уговора о системским операцијама 41](#_Toc100023044)

[2.3 Структура софтверског система - Концептуални (доменски) модел 46](#_Toc100023045)

[2.4 Релациони модел 47](#_Toc100023046)

[3. ПРОЈЕКТОВАЊЕ 53](#_Toc100023047)

[3.1 Архитектура софтверског система 53](#_Toc100023048)

[3.2. Пројектовање корисничког интерфејса 54](#_Toc100023049)

[СК1: Случај коришћења – Пријављивање корисника на систем 55](#_Toc100023050)

[СК2: Случај коришћења – Унос новог власника 56](#_Toc100023051)

[СК3: Случај коришћења – Претрага власника 57](#_Toc100023052)

[СК4: Случај коришћења – Измена података о власнику 59](#_Toc100023053)

[СК5: Случај коришћења – Брисање власника 61](#_Toc100023054)

[СК6: Случај коришћења – Унос нове парцеле 63](#_Toc100023055)

[СК7: Случај коришћења – Претрага парцела 65](#_Toc100023056)

[СК8: Случај коришћења – Измена података о парцели 67](#_Toc100023057)

[СК9: Случај коришћења – Унос новог купца 69](#_Toc100023058)

[СК10: Случај коришћења – Унос нове евиденције о продаји 71](#_Toc100023059)

[СК11: Случај коришћења – Претрага евиденција о продаји 74](#_Toc100023060)

[3.2.2 Пројектовање контролера корисничког интерфејса 77](#_Toc100023061)

[3.3 Пројектовање апликационе логике 78](#_Toc100023062)

[3.3.1 Контролер апликационе логике 78](#_Toc100023063)

[3.3.2 Системске операције 82](#_Toc100023064)

[3.3.3 Доменске класе 95](#_Toc100023065)

[3.3.4 Брокер базе података 96](#_Toc100023066)

[3.3.5 Пројектовање складишта података 99](#_Toc100023067)

[4. ИМПЛЕМЕНТАЦИЈА 102](#_Toc100023068)

[5. ТЕСТИРАЊЕ 104](#_Toc100023069)

[6. ЛИТЕРАТУРА 105](#_Toc100023070)

## Кориснички захтеви

## Вербални опис

Потребно је креирати апликацију за праћење рада удружења воћара у општини Параћин „ВоћариПН“. Апликација треба да омогући преглед, претрагу и евиденцију парцела на територији општине са различитим културама, власнике и податке истих као и све реализоване куповине од стране екстерних лица која се чувају у бази.

Претпоставка је да једна парцела садржи једну културу и да је једна парцела припада једном власнику.

Редовним ажурирањем података, за сваку парцелу у бази потребно је садржати власника, површину, врсту културе и место у коме се налази.

Поред тога потребно је водити евиденцију о подацима регистрованих власника, омогућити унос новог власника, измену података о власнику, претрагу свих власника и брисање података о власницима који више немају никакав удео.

Апликација треба да омогући унос нове парцеле, претрагу, као и могућност измене детаља парцеле.

Купац представља екстерног корисника који на захтев потражње одређене културе добија информације на основу претраге и уколико је остварена куповина, његови подаци као и подаци о куповини се чувају у бази.

# СЛУЧАЈЕВИ КОРИШЋЕЊА

У конкретном случају, идентификовани су следећи случајеви коришћења:

1. СК1: Пријављивање корисника
2. СК2: Унос новог власника
3. СК3: Претрага власника
4. СК4: Измена података о власнику
5. СК5: Брисање власника
6. СК6: Унос нове парцеле
7. CK7: Претрага парцела
8. СК8: Измена података о парцели
9. СК9: Унос новог купца
10. СК10: Унос нове евиденције о продаји
11. СК11: Претрага евиденција о продаји

Модел СК се може представити преко следећег дијаграма случајева коришћења:

Diagram

Description automatically generated

*Дијаграм слушаја коришћења*

## СК1: Случај коришћења – Пријављивање корисника на систем

**Назив СК**

*Пријављивање корисника на систем*

**Актори СК**

*Корисник*

**Учесници СК**

*Корисник* и систем (програм)

Предуслов: Систем је укључен. Систем приказује форму за пријављивање на систем.

**Основни сценарио СК:**

1. *Корисник* уноси податке потребне за пријављивање. (АПУСО)

2. *Корисник* контролише да ли је коректно унео податке потребне за пријављивање. (АНСО)

3. *Корисник* позива систем да се пријави (провери податке). (АПСО)

4. Систем проверава податке о кориснику. (СО)

5. Систем приказује кориснику поруку: **“Успешно пријављивање на систем! “.** (ИА)

**Алтернативна сценарија**

5.1. Уколико систем не може да нађе корисника, он приказује кориснику поруку: **“Неуспешно пријављивање на систем!”.** (ИА)

## СК2: Случај коришћења – Унос новог власника

**Назив СК**

Унос *новог власника*

**Актори СК**

*Корисник*

**Учесници СК**

*Корисник* и систем (програм)

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за унос *власника*. Учитана је листа *места.*

**Основни сценарио СК:**

1. *Корисник* позива систем да креира *новог власника*. (АПСО)
2. Систем креира *новог власника*. (СО)
3. Систем приказује поруку *кориснику*: **„Систем је успешно креирао власника!“**. (ИА)
4. *Корисник* уноси податке о *новом власнику*. (АПУСО)
5. *Корисник* контролише да ли је коректно унео податке о *новом власнику*. (АНСО)
6. *Корисник* позива систем да запамти податке о *новом власнику*. (АПСО)
7. Систем памти податке о *новом власнику*. (СО)
8. Систем приказује *кориснику* поруку: **„Систем је запамтио власника!“.** (ИА)

**Алтернативна сценарија**

3.1. Уколико систем не може да *креира власника*, систем приказује поруку: **„Систем не може да креира власника!“.** (ИА)

8.1. Уколико систем не може да запамти податке о *новом власнику* он приказује *кориснику* поруку **„Систем не може да запамти власника!“**. (ИА)

## СК3: Случај коришћења – Претрага власника

**Назив СК**

Претрага *власника*

**Актори СК**

*Корисник*

**Учесници СК**

*Корисник* и систем (програм)

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за претрагу *власника*.

**Основни сценарио СК**

1. *Корисник* уноси вредности по којима претражује *власнике*. (АПУСО)
2. *Корисник* позива систем да пронађе *власнике* по задатим вредностима. (АПСО)
3. Систем претражује *власнике* по задатим вредностима. (СО)
4. Систем приказује *кориснику* пронађене *власнике* и поруку: **„Систем је пронашао власнике по задатом критеријуму“.** (ИА)
5. *Корисник* бира *власника* чије податке жели да прикаже. (АПУСО)
6. *Корисник* позива систем да прикаже податке о изабраном *власнику*. (АПСО)
7. Систем приказује податке о изабраном *власнику* и поруку: **„Систем је учитао власника!“.** (ИА)

**Алтернативни сценарио СК**

4.1. Уколико систем не може да пронађе *власснике* он *кориснику* приказује поруку: **„Не постоје власници за задати критеријум!“**. (ИА)

7.1. Уколико систем не може да учита *власника* по задатим критеријумима, приказује поруку **„Систем не може да учита власника!“**. (ИА)

## СК4: Случај коришћења – Измена података о власнику

**Назив СК**

Измена података о *власнику*

**Актори СК**

*Корисник*

**Учесници СК**

*Корисник* и систем (програм)

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за претрагу *власника*. Учитана је листа *места.*

**Основни сценарио СК**

1. *Корисник* уноси вредности по којима претражује *власнике*. (АПУСО)
2. *Корисник* позива систем да пронађе *власнике* по задатим вредностима. (АПСО)
3. Систем претражује *власнике* по задатим вредностима. (СО)
4. Систем приказује *кориснику* пронађене *власнике* и поруку: **„Систем је пронашао власнике по задатом критеријуму“.** (ИА)
5. *Корисник* бира *власника* чије податке жели да измени. (АПУСО)
6. *Корисник* позива систем да прикаже податке о изабраном *власнику*. (АПСО)
7. Систем налази *власника*. (СО)
8. Систем приказује податке о изабраном *власнику* и поруку: **„Систем је учитао власника!“.** (ИА)
9. *Корисник* мења податке о *власнику*. (АПУСО)
10. *Корисник* контролише да ли је коректно унео податке. (АНСО)
11. *Корисник* позива систем да запамти податке о *власнику*. (АПСО)
12. Систем памти податке о *власнику*. (СО)
13. Систем приказује *кориснику* поруку: **„Систем је запамтио власника!“**. (ИА)

**Алтернативни сценарио СК**

4.1. Уколико систем не може да пронађе *власника* по задатим критеријумима, приказује поруку **„Систем не може да пронађе *власнике* за задати критеријум!“**. (ИА)

8.1. Уколико систем не може да учита *власника* по задатим критеријумима, приказује поруку **„Систем не може да учита власника!“**. (ИА)

12.1. Уколико систем не може да запамти *власника*, приказује поруку **„Систем не може да запамти власника!“.** (ИА)

## СК5: Случај коришћења – Брисање власника

**Назив СК**

Брисање *власника*

**Актори СК**

*Корисник*

**Учесници СК**

*Корисник* и систем (програм)

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за претрагу *власника*.

**Основни сценарио СК**

1. *Корисник* уноси вредности по којима претражује *власнике*. (АПУСО)
2. *Корисник* позива систем да пронађе *власнике* по задатим вредностима. (АПСО)
3. Систем претражује *власнике* по задатим вредностима. (СО)
4. Систем приказује *кориснику* пронађене *власнике* и поруку: **„Систем је пронашао власнике по задатом критеријуму“.** (ИА)
5. *Корисник* бира *власника* којег жели да обрише. (АПУСО)
6. *Корисник* позива систем да прикаже податке о изабраном *власнику*. (АПСО)
7. Систем учитава *власника*. (СО)
8. Систем приказује податке о изабраном *власнику* и поруку: **„Систем је учитао власника!“.** (ИА)
9. *Корисник* позива систем да обрише податке о изабраном *власнику*. (АПСО)
10. Систем брише податке о *власнику.* (СО)
11. Систем брише изабраног *власника* и приказује поруку: **„Систем је обрисао власника!“**. (ИА)

**Алтернативни сценарио СК**

4.1. Уколико систем не може да пронађе *власника*  по задатим критеријумима, приказује поруку „**Систем не може да пронађе власнике за задати критеријум!“**. (ИА)

7.1. Уколико систем не може да учита *власника* по задатим критеријумима, приказује поруку **„Систем не може да учита власника!“**. (ИА)

11.1. Уколико систем не може да обрише *власника*, приказује поруку **„Систем не може да обрише власника”**. (ИА)

## СК6: Случај коришћења – Унос нове парцеле

**Назив СК**

Унос *новe парцеле*

**Актори СК**

*Корисник*

**Учесници СК**

*Корисник* и систем (програм)

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за унос *парцеле*. Учитана је листа *власника, култура и места.*

**Основни сценарио СК**

1. *Корисник* позива систем да креира *нову парцелу*. (АПСО)
2. Систем креира *нову парцелу*. (СО)
3. Систем приказује поруку *кориснику*: **„Систем је успешно креирао нову парцелу“**. (ИА)
4. *Корисник* уноси податке за *нову парцелу*. (АПУСО)
5. *Корисник* контролише да ли је коректно унео податке о *новој парцели*. (АНСО)
6. *Корисник* позива систем да запамти податке о *новој парцели*. (АПСО)
7. Систем памти податке о *новој парцели*. (СО)
8. Систем приказује поруку *кориснику*: **„Систем је запамтио парцелу“.** (ИА)

**Алтернативни сценарио СК**

3.1. Уколико систем не може да креира *парцелу*, систем приказује поруку: **„Систем не може да креира парцелу!“**. (ИА)

8.1. Уколико систем не може да запамти *стан*, систем приказује поруку: **„Систем не може да запамти парцелу!“**. (ИА)

## СК7: Случај коришћења – Претрага парцела

**Назив СК**

Претрага *парцела*

**Актори СК**

*Корисник*

**Учесници СК**

*Корисник* и систем (програм)

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за претрагу *парцела*.

**Основни сценарио СК**

1. *Корисник* уноси вредности по којима претражује *парцеле*. (АПУСО)
2. *Корисник* позива систем да пронађе *парцеле* по задатим вредностима. (АПСО)
3. Систем претражује *парцеле* по задатим вредностима. (СО)
4. Систем приказује *кориснику* пронађене *парцеле* и поруку: **„Систем је пронашао парцеле по задатом критеријуму“.** (ИА)
5. *Корисник* бира *парцелу* чије податке жели да прикаже. (АПУСО)
6. *Корисник* позива систем да прикаже податке о изабраној *парцели*. (АПСО)
7. Систем учитава *парцелу*. (СО)
8. Систем приказује податке о изабраној *парцели* и поруку: **„Систем је учитао парцелу!“.** (ИА)

**Алтернативни сценарио СК**

4.1. Уколико систем не може да пронађе *парцелу* по задатим критеријумима, приказује поруку **„Систем не може да пронађе парцелу за задати критеријум!“**. (ИА)

8.1. Уколико систем не може да пронађе *парцелу* по задатим критеријумима, приказује поруку **„Систем не може да учита парцелу!“**. (ИА)

## СК8: Случај коришћења – Измена података о парцели

**Назив СК**

Измена података о *парцели*

**Актори СК**

*Корисник*

**Учесници СК**

*Корисник* и систем (програм)

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за претрагу *парцела*. Учитана је листа *власника, типова парцела и места.*

**Основни сценарио СК**

1. *Корисник* уноси вредности по којима претражује *парцеле*. (АПУСО)
2. *Корисник* позива систем да пронађе *парцеле* по задатим вредностима. (АПСО)
3. Систем претражује *парцеле* по задатим вредностима. (СО)
4. Систем приказује *кориснику* пронађене *парцеле* и поруку: **„Систем је пронашао парцеле по задатом критеријуму“.** (ИА)
5. *Корисник* бира *парцелу* чије податке жели да измени. (АПУСО)
6. *Корисник* позива систем да прикаже податке о изабраној *парцели*. (АПСО)
7. Систем учитава *парцелу*. (СО)
8. Систем приказује податке о изабраној *парцели* и поруку: **„Систем је учитао парцелу!“.** (ИА)
9. *Корисник* мења податке о *парцели*. (АПУСО)
10. *Корисник* контролише да ли је коректно унео податке. (АНСО)
11. *Корисник* позива систем да запамти податке о *парцели*. (АПСО)
12. Систем памти податке о *парцели*. (СО)
13. Систем приказује *кориснику* поруку: **„Систем је запамтио парцелу!“**. (ИА)

**Алтернативни сценарио СК**

4.1. Уколико систем не може да пронађе *парцелу* по задатим критеријумима, приказује поруку **„Систем не може да пронађе *парцелу* за задати критеријум!“**. (ИА)

8.1. Уколико систем не може да учита *парцелу* по задатим критеријумима, приказује поруку **„Систем не може да учита парцелу!“**. (ИА)

13.1. Уколико систем не може да запамти *парцелу*, приказује поруку **„Систем не може да запамти парцелу!“.** (ИА)

## СК9: Случај коришћења – Унос новог купца

**Назив СК**

Унос *новог купца*

**Актори СК**

*Корисник*

**Учесници СК**

*Корисник* и систем (програм)

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за унос *купца*. Учитана је листа *места.*

**Основни сценарио СК**

1. *Корисник* позива систем да креира *новог купца*. (АПСО)
2. Систем креира *новог купца.* (СО)
3. Систем приказује поруку *кориснику*: **„Систем је успешно креирао новог купца“**. (ИА)
4. *Корисник* уноси податке за *новог купца*. (АПУСО)
5. *Корисник* контролише да ли је коректно унео податке о *новом купцу*. (АНСО)
6. *Корисник* позива систем да запамти податке о *новом купцу*. (АПСО)
7. Систем памти податке о *новом купцу*. (СО)
8. Систем приказује поруку *кориснику*: **„Систем је запамтио купца“.** (ИА)

**Алтернативни сценарио СК**

3.1. Уколико систем не може да креира *купца*, систем приказује поруку: **„Систем не може да креира купца!“**. (ИА)

8.1. Уколико систем не може да запамти *купца*, систем приказује поруку: **„Систем не може да запамти купца!“**. (ИА)

## СК10: Случај коришћења – Унос нове евиденције о продаји

**Назив СК**

Унос *нове евиденције о продаји*

**Актори СК**

*Корисник*

**Учесници СК**

*Корисник* и систем (програм)

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за унос *нове евиденције о продаји.* Учитана је листа *купаца* и *култура.* Након избора културе, учитана је листа *парцела* за изабрану културу.

**Основни сценарио СК**

1. *Корисник* бира купца за којег жели да унесе *нову евиденцију о продаји.* (АПУСО)
2. *Корисник* бира културу за коју жели да унесе *нову евиденцију о продаји.* (АПУСО)
3. *Корисник* уноси податке о *новој евиденцији о продаји*. (АПУСО)
4. *Корисник* контролише да ли је коректно унео податке о *новој евиденцији о продаји*. (АНСО)
5. *Корисник* позива систем да запамти податке о *новој евиденцији о продаји.* (АПСО)
6. Систем памти податке о *новој евиденцији о продаји.* (СО)
7. Систем приказује *кориснику* поруку „Систем је успешно извршио чување!“. (ИА)

**Алтернативни сценарио СК**

6.1. Уколико систем не може да запамти податке о *новој продаји*, приказује поруку**: „Систем не може да изврши чување“.** (ИА)

## СК11: Случај коришћења – Претрага евиденција о продаји

**Назив СК**

Претрага *евиденција о родаји*

**Актори СК**

*Корисник*

**Учесници СК**

*Корисник* и систем (програм)

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за претрагу *евиденција о продаји*.

**Основни сценарио СК**

1. *Корисник* уноси вредности по којима претражује *евиденције о продаји*. (АПУСО)
2. *Корисник* позива систем да пронађе *евиденције о продаји* по задатим вредностима. (АПСО)
3. Систем претражује *евиденције о продаји* по задатим вредностима. (СО)
4. Систем приказује *кориснику* пронађене *евиденције о продаји* и поруку: **„Систем је пронашао евиденције о продаји за задати критеријум!“**. (ИА)
5. *Корисник* бира *продају* чије податке жели да прикаже. (АПУСО)
6. *Корисник* позива систем да прикаже податке о изабраној *продаји*. (АПСО)
7. Систем учитава *продаје*. (СО)
8. Систем приказује податке о изабраној *продаји* и поруку: **„Систем је учитао продају!“.** (ИА)

**Алтернативни сценарио СК**

4.1. Уколико систем не може да пронађе евиденцију о *продаји* за изабрани критеријум, приказује поруку: **„Не постоје евиденције о продаји за задати критеријум!“.** (ИА)

8.1. Уколико систем не може да учита *продају* по задатим критеријумима, приказује поруку **„Систем не може да учита продају!“**. (ИА)

# 2. АНАЛИЗА

Након фазе прикупљања захтева, прелази се на следећу фазу, фазу анализе. Ова фаза описује логичку структуру и понашање софвтерског система, тј. пословну логику софтверског система.

• Понашање описујемо помоћу:

- системских дијаграма секвенци

- уговора о системским операцијама

• Структуру описујемо помоћу:

- концептуалног модела

- релационог модела

## 2.1 Понашање софтверског система - Дијаграм секвенци случаја коришћења

Понашање система се може описати преко УМЛ-ових секвенцних дијаграма [Larman], односно преко дијаграма сарадње[JPRS].

Системски дијаграм секвенци приказује, за издвојени сценарио СК, догађаје у одређеном редоследу, који успостављају интеракцију измену актора и софтверског система.

Догађај који направи актор је побуда за позив системске операције. Прецизније речено, догађај који направи актор прихвата прималац догађаја који након тога позива системску операцију. То значи да актор не позива системску операцију непосредно већ то чини прекo посредника (примаоца догађаја). Позив системске операције указује на интеракцију измену актора и система. За догађај који представља побуду за позив СО се често каже да је то системски догађај.

## ДС1: Дијаграм секвенци случаја коришћења - Пријављивање на систем

**Основни сценарио СК:**

1. *Корисник* уноси податке потребне за пријављивање. (АПУСО)

2. *Корисник* контролише да ли је коректно унео податке потребне за пријављивање. (АНСО)

3. *Корисник* позива систем да се пријави (провери податке). (АПСО)

4. Систем проверава податке о кориснику. (СО)

5. Систем приказује кориснику поруку: **“Систем је успешно пријавио корисника! “.** (ИА)

Diagram

Description automatically generated

**Алтернативна сценарија**

5.1. Уколико систем не може да нађе корисника, он приказује кориснику поруку: **“Неуспешно пријављивање корисника на систем!”.** (ИА)

**Diagram

Description automatically generated**

## ДС2: Дијаграм секвенци случаја коришћења - Унос новог власника

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за унос *власника*. Учитана је листа *места.*

**Основни сценарио СК:**

1. *Корисник* позива систем да креира *новог власника*. (АПСО)
2. Систем приказује поруку *кориснику*: **„Систем је успешно креирао власника!“**. (ИА)
3. *Корисник* позива систем да запамти податке о *новом власнику*. (АПСО)
4. Систем приказује *кориснику* поруку: **„Систем је запамтио власника!“.** (ИА)



**Алтернативна сценарија**

2.1. Уколико систем не може да *креира власника*, систем приказује поруку: **„Систем не може да креира власника!“.** (ИА)



4.1. Уколико систем не може да запамти податке о *новом власнику* он приказује *кориснику* поруку **„Систем не може да запамти власника!“**. (ИА)



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

1. *Signal* **VratiListuMesta**(List<Mesto>)
2. *Signal* **KreirajVlasnika**(Vlasnik)
3. *Signal* **ZapamtiVlasnika**(Vlasniik)

## ДС3: Дијаграм секвенци случаја коришћења - Претрага власника

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за претрагу *власника*.

**Основни сценарио СК**

1. *Корисник* позива систем да пронађе *власнике* по задатим вредностима. (АПСО)
2. Систем приказује *кориснику* пронађене *власнике* и поруку: **„Систем је пронашао власнике по задатом критеријуму“.** (ИА)
3. *Корисник* позива систем да прикаже податке о изабраном *власнику*. (АПСО)
4. Систем приказује податке о изабраном *власнику* и поруку: **„Систем је учитао власника!“.** (ИА)

**Diagram

Description automatically generated**

**Алтернативни сценарио СК**

2.1. Уколико систем не може да пронађе *власснике* он *кориснику* приказује поруку: **„Не постоје власници за задати критеријум!“**. (ИА)

Graphical user interface, text, application, email

Description automatically generated

4.1. Уколико систем не може да учита *власника* по задатим критеријумима, приказује поруку **„Систем не може да учита власника!“**. (ИА)

Diagram

Description automatically generated

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

1. *Signal* **PretraziVlasnike**(Kriterijum, List<Vlasnik>)
2. *Signal* **PrikaziVlasnika**(Vlasnik)

## ДС4: Дијаграм секвенци случаја коришћења – Измена података о власнику

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за претрагу *власника*. Учитана је листа *места.*

**Основни сценарио СК**

1. *Корисник* позива систем да пронађе *власнике* по задатим вредностима. (АПСО)
2. Систем приказује *кориснику* пронађене *власнике* и поруку: **„Систем је пронашао власнике по задатом критеријуму“.** (ИА)
3. *Корисник* позива систем да прикаже податке о изабраном *власнику*. (АПСО)
4. Систем приказује податке о изабраном *власнику* и поруку: **„Систем је учитао власника!“.** (ИА)
5. *Корисник* позива систем да запамти податке о *власнику*. (АПСО)
6. Систем приказује *кориснику* поруку: **„Систем је запамтио власника!“**. (ИА)

Diagram

Description automatically generated

**Алтернативни сценарио СК**

* 1. Уколико систем не може да пронађе *власника* по задатим критеријумима, приказује поруку **„Систем не може да пронађе *власнике* за задати критеријум!“**. (ИА)

**Graphical user interface, text, application, email

Description automatically generated**

* 1. Уколико систем не може да учита *власника* по задатим критеријумима, приказује поруку **„Систем не може да учита власника!“**. (ИА)

**Diagram

Description automatically generated**

* 1. Уколико систем не може да запамти *власника*, приказује поруку **„Систем не може да запамти власника!“.** (ИА)

Diagram

Description automatically generated

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

1. *Signal* **UcitajListuMesta**(List<Mesto>)
2. *Signal* **PretraziVlasnike**(Kriterijum, List<Vlasnik>)
3. *Signal* **PrikaziVlasnika**(Vlasnik)
4. *Signal* **ZapamtiVlasnika**(Vlasnik)

## ДС5: Дијаграм секвенци случаја коришћења - Брисање власника

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за претрагу *власника*.

**Основни сценарио СК**

1. *Корисник* позива систем да пронађе *власнике* по задатим вредностима. (АПСО)
2. Систем приказује *кориснику* пронађене *власнике* и поруку: **„Систем је пронашао власнике по задатом критеријуму“.** (ИА)
3. *Корисник* позива систем да прикаже податке о изабраном *власнику*. (АПСО)
4. Систем приказује податке о изабраном *власнику* и поруку: **„Систем је учитао власника!“.** (ИА)
5. *Корисник* позива систем да обрише податке о изабраном *власнику*. (АПСО)
6. Систем брише изабраног *власника* и приказује поруку: **„Систем је обрисао власника!“**. (ИА)

Diagram

Description automatically generated

**Алтернативни сценарио СК**

2.1. Уколико систем не може да пронађе *власника*  по задатим критеријумима, приказује поруку „**Систем не може да пронађе власнике за задати критеријум!“**. (ИА)

**Graphical user interface, text, application, email

Description automatically generated**

4.1. Уколико систем не може да пронађе *власника* по задатим критеријумима, приказује поруку **„Систем не може да учита власника!“**. (ИА)

Diagram

Description automatically generated

6.1. Уколико систем не може да обрише *власника*, приказује поруку **„Систем не може да обрише власника”**. (ИА)

Diagram

Description automatically generated

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

1. *Signal* **PretraziVlasnike**(Kriterijum, List<Vlasnik>)
2. *Signal* **PrikaziVlasnika**(Vlasnik)
3. *Signal* **ObrisiVlasnike**(Vlasnik)

### 

## ДС6: Дијаграм секвенци случаја коришћења – Унос нове парцеле

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за унос *парцеле*. Учитана је листа *власника, типова парцела и места.*

**Основни сценарио СК**

1. *Корисник* позива систем да креира *нову парцелу*. (АПСО)
2. Систем приказује поруку *кориснику*: **„Систем је успешно креирао нову парцелу“**. (ИА)
3. *Корисник* позива систем да запамти податке о *новој парцели*. (АПСО)
4. Систем приказује поруку *кориснику*: **„Систем је запамтио парцелу“.** (ИА)



**Алтернативни сценарио СК**

2.1. Уколико систем не може да креира *парцелу*, систем приказује поруку: **„Систем не може да креира парцелу!“**. (ИА)



4.1. Уколико систем не може да запамти *парцелу*, систем приказује поруку: **„Систем не може да запамти парцелу!“**. (ИА)



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

1. *Signal* **VratiListuMesta**(List<Mesto>)
2. *Signal* **VratiListuVlasnika**(List<Vlasnik>)
3. *Signal* **VratiListuKultura**(List<Kultura>)
4. *Signal* **KreirajParcelu**(Parcela)
5. *Signal* **ZapamtiParcelu**(Parcela)

## ДС7: Дијаграм секвенци случаја коришћења – Претрага парцела

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за претрагу *парцела*.

**Основни сценарио СК**

1. *Корисник* позива систем да пронађе *парцеле* по задатим вредностима. (АПСО)
2. Систем приказује *кориснику* пронађене *парцеле* и поруку: **„Систем је пронашао парцеле по задатом критеријуму“.** (ИА)
3. *Корисник* позива систем да прикаже податке о изабраној *парцели*. (АПСО)
4. Систем приказује податке о изабраној *парцели* и поруку: **„Систем је пронашао парцелу!“.** (ИА)



**Алтернативни сценарио СК**

* 1. Уколико систем не може да пронађе *парцелу* по задатим критеријумима, приказује поруку **„Систем не може да пронађе парцелу за задати критеријум!“**. (ИА)



4.1. Уколико систем не може да пронађе *парцелу* по задатим критеријумима, приказује поруку **„Систем не може да пронађе парцелу!“**. (ИА)



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

1. *Signal* **PretraziParcele**(Kriterijum, List<Parcela>)
2. *Signal* **PrikaziParcelu**(Parcela)

## ДС8: Дијаграм секвенци случаја коришћења – Измена парцеле

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за претрагу *парцела*. Учитана је листа *власника, типова парцела и места.*

**Основни сценарио СК**

1. *Корисник* позива систем да пронађе *парцеле* по задатим вредностима. (АПСО)
2. Систем приказује *кориснику* пронађене *парцеле* и поруку: **„Систем је пронашао парцеле по задатом критеријуму“.** (ИА)
3. *Корисник* позива систем да прикаже податке о изабраној *парцели*. (АПСО)
4. Систем приказује податке о изабраној *парцели* и поруку: **„Систем је пронашао парцелу!“.** (ИА)
5. *Корисник* позива систем да запамти податке о *парцели*. (АПСО)
6. Систем приказује *кориснику* поруку: **„Систем је запамтио парцелу!“**. (ИА)



**Алтернативни сценарио СК**

* 1. Уколико систем не може да пронађе *парцелу* по задатим критеријумима, приказује поруку **„Систем не може да пронађе *парцелу* за задати критеријум!“**. (ИА)



* 1. Уколико систем не може да пронађе *парцелу* по задатим критеријумима, приказује поруку **„Систем не може да пронађе парцелу!“**. (ИА)



6.1. Уколико систем не може да запамти *парцелу*, приказује поруку **„Систем не може да запамти парцелу!“.** (ИА)



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

1. *Signal* **UcitajListuMesta**(List<Mesto>)
2. *Signal* **UcitajListuVlasnika**(List<Vlasnik>)
3. *Signal* **UcitajListuKultura**(List<Kultura>)
4. *Signal* **PretraziParcele**(Kriterijum, List<Parcela>)
5. *Signal* **PrikaziParcelu**(Parcela)
6. *Signal* **ZapamtiParcelu**(Parcela)

## ДС9: Дијаграм секвенци случаја коришћења – Унос новог купца

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за унос *купца*. Учитана је листа *места.*

**Основни сценарио СК**

1. *Корисник* позива систем да креира *новог купца*. (АПСО)
2. Систем приказује поруку *кориснику*: **„Систем је успешно креирао новог купца“**. (ИА)
3. *Корисник* позива систем да запамти податке о *новом купцу*. (АПСО)
4. Систем приказује поруку *кориснику*: **„Систем је запамтио купца“.** (ИА)



**Алтернативни сценарио СК**

2.1. Уколико систем не може да креира *купца*, систем приказује поруку: **„Систем не може да креира купца!“**. (ИА)



4.1. Уколико систем не може да запамти *купца*, систем приказује поруку: **„Систем не може да запамти купца!“**. (ИА)



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

1. *Signal* **VratiListuMesta**(List<Mesto>)
2. *Signal* **KreirajKupca**(Kupac)
3. *Signal* **ZapamtiKupca**(Kupac)

## 

## ДС10: Дијаграм секвенци случаја коришћења – Унос нове евиденције продаје

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за унос *нове евиденције о продаји.* Учитана је листа *парцела* и *купаца.*

**Основни сценарио СК**

1. *Корисник* позива систем да запамти податке о *новој евиденцији о продаји.* (АПСО)
2. Систем приказује *кориснику* поруку „Систем је успешно извршио чување!“. (ИА)



**Алтернативни сценарио СК**

* 1. Уколико систем не може да запамти податке о *новој продаји*, приказује поруку**: „Систем не може да изврши чување“.** (ИА)



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

1. *Signal* **VratiListuVlasnika**(List<Vlasnik>)
2. *Signal* **VratiListuParcela**(List<Parcela>)
3. *Signal* **ZapamtiProdaje**(Kupac)

## ДС11 : Дијаграм секвенци случаја коришћења – Претага евиденција продаје

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за претрагу *евиденција о продаји*.

**Основни сценарио СК**

1. *Корисник* позива систем да пронађе *евиденције о продаји* по задатим вредностима. (АПСО)
2. Систем приказује *кориснику* пронађене *евиденције о продаји* и поруку: **„Систем је пронашао евиденције о продаји за задати критеријум!“**. (ИА)
3. *Корисник* позива систем да прикаже податке о изабраној *продаји*. (АПСО)
4. Систем приказује податке о изабраној *продаји* и поруку: **„Систем је учитао продају!“.** (ИА)

Diagram

Description automatically generated

**Алтернативни сценарио СК**

2.1. Уколико систем не може да пронађе евиденцију о *продаји* за изабрани критеријум, приказује поруку: **„Систем не може да пронађе продаје!“.** (ИА)



4.1. Уколико систем не може да пронађе *продају* по задатим критеријумима, приказује поруку **„Систем не може да учита продају!“**. (ИА)

Diagram

Description automatically generated

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

1. *Signal* **PretraziProdaje**(Kriterijum, List<Prodaja>)
2. *Signal* **PrikaziProdaju**(Prodaja)

Као резултат анализе сценарија добијено је укупно 20 системских операција које треба пројектовати:

1. *Signal* **PrijaviKorisnika**(Korisnik)
2. *Signal* **VratiListuMesta**(List<Mesto>)
3. *Signal* **KreirajVlasnika**(Vlasnik)
4. *Signal* **ZapamtiVlasnika**(Vlasniik)
5. *Signal* **PretraziVlasnike**(Kriterijum, List<Vlasnik>)
6. *Signal* **PrikaziVlasnika**(Vlasnik)
7. *Signal* **ObrisiVlasnikа**(Vlasnik)
8. *Signal* **VratiListuVlasnika**(List<Vlasnik>)
9. *Signal* **VratiListuKultura**(List<Kultura>)
10. *Signal* **KreirajParcelu**(Parcela)
11. *Signal* **ZapamtiParcelu**(Parcela)
12. *Signal* **PretraziParcele**(Kriterijum, List<Parcela>)
13. *Signal* **PrikaziParcelu**(Parcela)
14. *Signal* **KreirajKupca**(Kupac)
15. *Signal* **ZapamtiKupca**(Kupac)
16. *Signal* **VratiListuParcela**(List<Parcela>)
17. *Signal* **VratiListuKupaca**(List<Kupac>)
18. *Signal* **ZapamtiProdaje**(Kupac)
19. *Signal* **PretraziProdaje**(Kriterijum, List<Prodaja>)
20. *Signal* **PrikaziProdaju**(Prodaja)

## Понашање софтверског система – Дефинисање уговора о системским операцијама

**Уговор УГ1: Пријави корисника**

**Операција: PrijaviSe(Korisnik): signal;**

**Веза са СК: СК1**

**Предуслови:-**

**Постуслови: Корисник је пријављен на систем.**

**Уговор УГ2: Врати листу места**

**Операција**: **VratiListuMesta**(List<Mesto>): *Signal*;

**Веза са СК**: СК2, СК4, СК6, СК8, СК9

**Предуслови: -**

**Постуслови: -**

**Уговор УГ3: Креирај Власника**

**Операција:** **KreirajVlasnika**(Vlasnik): *Signal*;

**Веза са СК:** СК2

**Предуслови**: Просто вредносно ограничење над објектом Власник мора бити задовољено. Структурно ограничење над објектом Власник мора бити задовољено.

**Постуслови**: Креиран је нови Валсник.

**Уговор УГ4: Запамти Власника**

**Операција:** **ZapamtiVlasnika**(Vlasniik): *Signal*;

**Веза са СК:** СК2, СК4

**Предуслови**: Просто вредносно ограничење над објектом Власник мора бити задовољено. Структурно ограничење над објектом Власник мора бити задовољено.

**Постуслови**: Подаци о Власнику су запамћени.

**Уговор УГ5: Претражи Власнике**

**Операција**: **PretraziVlasnike**(Kriterijum, List<Vlasnik>): *Signal*;

**Веза са СК**: СК3, СК4, СК5

**Предуслови: -**

**Постуслови: -**

**Уговор УГ6: Прикажи власника**

**Операција**: **PrikaziVlasnika**(Vlasnik): *Signal*;

**Веза са СК**: СК3, СК4, СК5

**Предуслови: -**

**Постуслови: -**

**Уговор УГ7: Обриши власника**

**Операција**: **ObrisiVlasnika**(Vlasnik): *Signal*;

**Веза са СК**: СК5

**Предуслови: -**

**Постуслови: -**

**Уговор УГ8: Врати листу власника**

**Операција**: **VratiListuVlasnika**(List<Vlasnik>): *Signal*;

**Веза са СК**: СК6, СК8

**Предуслови: -**

**Постуслови: -**

**Уговор УГ9: Врати листу култура**

**Операција**: **VratiListuKultura** (List<Kultura>): *Signal*;

**Веза са СК**: СК6, СК8

**Предуслови: -**

**Постуслови: -**

**Уговор УГ10: Креирај Парцелу**

**Операција:** **KreirajParcelu**(Parcela): *Signal*;

**Веза са СК:** СК6

**Предуслови**: Просто вредносно ограничење над објектом Парцела мора бити задовољено. Структурно ограничење над објектом Парцела мора бити задовољено.

**Постуслови**: Креирана је нова парцела.

**Уговор УГ11: Запамти Парцелу**

**Операција:** **ZapamtiParcelu**(Parcela): *Signal*;

**Веза са СК:** СК6, СК8

**Предуслови**: Просто вредносно ограничење над објектом Парцела мора бити задовољено. Структурно ограничење над објектом Парцела мора бити задовољено.

**Постуслови**: Подаци о Парцели су запамћени.

**Уговор УГ12: Претражи Парцеле**

**Операција**: **PretraziParcele**(Kriterijum, List<Parcela>): *Signal*;

**Веза са СК**: СК7, СК8

**Предуслови: -**

**Постуслови: -**

**Уговор УГ13: Прикажи Парцелу**

**Операција**: **PrikaziParcelu**(Parcela): *Signal*;

**Веза са СК**: СК7, СК8

**Предуслови: -**

**Постуслови: -**

**Уговор УГ14: Креирај Купца**

**Операција:** **KreirajKupca**(Kupac): *Signal*;

**Веза са СК:** СК9

**Предуслови**: Просто вредносно ограничење над објектом Купац мора бити задовољено. Структурно ограничење над објектом Купац мора бити задовољено.

**Постуслови**: Креиран је нови купац.

**Уговор УГ15: Запамти Купца**

**Операција:** **ZapamtiKupca**(Kupac): *Signal*;

**Веза са СК:** СК9

**Предуслови**: Просто вредносно ограничење над објектом Купац мора бити задовољено. Структурно ограничење над објектом Купац мора бити задовољено.

**Постуслови**: Подаци о Купцу су запамћени.

**Уговор УГ16: Врати листу парцела**

**Операција**: **VratiListuParcela**(List<Parcela>): *Signal*;

**Веза са СК**: СК10

**Предуслови: -**

**Постуслови: -**

**Уговор УГ17: Врати листу Купаца**

**Операција**: **VratiListuKupaca**(List<Kupac>): *Signal*;

**Веза са СК**: СК10

**Предуслови: -**

**Постуслови: -**

**Уговор УГ18: Запамти Продаје**

**Операција:** **ZapamtiProdaje**(Kupac): *Signal*;

**Веза са СК:** СК10

**Предуслови**: Просто вредносно ограничење над објектом Продаја мора бити задовољено. Структурно ограничење над објектом Продаја мора бити задовољено.

**Постуслови**: Подаци о Продаји су запамћени.

**Уговор УГ19: Претражи Продаје**

**Операција**: **PretraziProdaje**(Kriterijum, List<Prodaja>): *Signal*;

**Веза са СК**: СК11

**Предуслови: -**

**Постуслови: -**

**Уговор УГ20: Прикажи продају**

**Операција**: **PrikaziProdaju**(Prodaja): *Signal*;

**Веза са СК**: СК11

**Предуслови: -**

**Постуслови: -**

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

Diagram, schematic

Description automatically generated

Text

Description automatically generated with low confidence

## 2.4 Релациони модел

**Kupac**(ID, Ime, Prezime, Email, Telefon, *MestoID*)

**Vlasnik**(ID, Ime, Prezime, Email, Telefon, *MestoID*)

**Mesto**(ID, Naziv)

**Kultura**(ID, Naziv)

**Parcela**(ID, Broj, Opis, Povrsina, *KulturaID, VlasnikID, MestoID*)

**Prodaja**(*ParcelaID, KupacID*, Datum, JedinicnaCena, Kolicina, Ukupno)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Табела Власник** | | **Просто вредносно ограничење** | | **Сложено вредносно ограничење** | | **Структурно**  **ограничење** |
| **Атрибути** | **Име** | **Тип**  **атрибута** | **Вредност**  **атрибута** | **Међузав.**  **Атрибута**  **Једне**  **табеле** | **Међузав.**  **Атрибута**  **Више**  **табела** | INSERT  /  UPDATE  CASCADES  Parcela  DELETE  RESTRICTED  Parcela |
| ID | integer | not null  and >0 |  |  |
| ime | string | not null |  |  |
| prezime | string | not null |  |  |
| email | string | not null |  |  |
| telefon | string | not null |  |  |
| mestoID | integer | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Табела Купац** | | **Просто вредносно ограничење** | | **Сложено вредносно ограничење** | | **Структурно**  **ограничење** |
| **Атрибути** | **Име** | **Тип**  **атрибута** | **Вредност**  **атрибута** | **Међузав.**  **Атрибута**  **Једне**  **табеле** | **Међузав.**  **Атрибута**  **Више**  **табела** | INSERT  /  UPDATE  CASCADES  Prodaja  DELETE  RESTRICTED  Prodaja |
| ID | integer | not null  and >0 |  |  |
| ime | string | not null |  |  |
| prezime | string | not null |  |  |
| email | string | not null |  |  |
| telefon | string | not null |  |  |
| mestoID | integer | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Табела Место** | | **Просто вредносно ограничење** | | **Сложено вредносно ограничење** | | **Структурно**  **ограничење** |
| **Атрибути** | **Име** | **Тип**  **атрибута** | **Вредност**  **атрибута** | **Међузав.**  **Атрибута**  **Једне**  **табеле** | **Међузав.**  **Атрибут**  **Више**  **табела** | INSERT  /  UPDATE  CASCADES  Kupac, Vlasnik, Parcela  DELETE  RESTRICTED  Kupac, Vlasnik, Parcela |
| ID | integer | not null  and >0 |  |  |
| naziv | string | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Табела Култура** | | **Просто вредносно ограничење** | | **Сложено вредносно ограничење** | | **Структурно**  **ограничење** |
| **Атрибути** | **Име** | **Тип**  **атрибута** | **Вредност**  **атрибута** | **Међузав.**  **Атрибута**  **Једне**  **табеле** | **Међузав.**  **Атрибут**  **Више**  **табела** | INSERT  /  UPDATE  CASCADES  Parcela  DELETE  RESTRICTED  Parcela |
| ID | integer | not null  and >0 |  |  |
| naziv | string | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Табела Парцела** | | **Просто вредносно ограничење** | | **Сложено вредносно ограничење** | | **Структурно**  **ограничење** |
| **Атрибути** | **Име** | **Тип**  **атрибута** | **Вредност**  **атрибута** | **Међузав.**  **Атрибута**  **Једне**  **табеле** | **Међузав.**  **Атрибута**  **Више**  **табела** | INSERT  RESTRICTED  Mesto, Vlasnik, Kultura  UPDATE  CASCADES  Prodaja  RESTRICTED  Mesto, Vlasnik, Kultura  DELETE  RESTRICTED  Prodaja |
| ID | integer | not null  and >0 |  |  |
| broj | string | not null |  |  |
| opis | string |  |  |  |
| povrsina | double | not null |  |  |
| kulturaID | integer | not null |  |  |
| vlasnikID | integer | not null |  |  |
| mestoID | integer | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Табела Продаја** | | **Просто вредносно ограничење** | | **Сложено вредносно ограничење** | | **Структурно**  **ограничење** |
| **Атрибути** | **Име** | **Тип**  **атрибута** | **Вредност**  **атрибута** | **Међузав.**  **Атрибута**  **Једне**  **табеле** | **Међузав.**  **Атрибута**  **Више**  **табела** | INSERT  RESTRICTED  Kupac, Parcela  UPDATE  RESTRICTED  Kupac, Parcela  DELETE  / |
| parcelaID | integer | not null  and >0 |  |  |
| kupacID | integer | not null  and >0 |  |  |
| Datum | date | not null  and >0 |  |  |
| JedinicnaCena | double | >=0 |  |  |
| Kolicina | double | >=0 |  |  |
| Ukupno | double | >=0 |  |  |

# 3. ПРОЈЕКТОВАЊЕ

## 3.1 Архитектура софтверског система

* У оквиру фазе пројектовања описује се физичка структура и понашање софтверског система тј. архитектура софтверског система. У овом семинарском раду је коришћена класична трослојна архитектура, која се састоји од:
* Корисничког интерфејса
* Апликационе логике
* Складишта података

Korisnički interfejs

Aplikaciona logika

Skladište podataka

**I sloj**

**II sloj**

**III sloj**

**Softverski sistem**

*Трослојна архитектура*

Skladište podataka

**Korisnički interfejs**

Ekranska forma

KontrolerKI

**Aplikaciona logika**

KontrolerAL

Poslovna logika

Database broker

*Трослојна архитектура – детаљнији приказ*

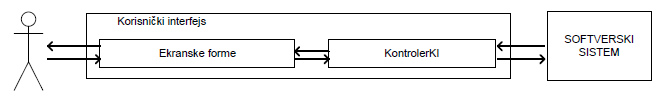
## 

## 3.2. Пројектовање корисничког интерфејса

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

Кориснички интерфејс састоји се из:

* Екранске форме,
* Контролера корисничког интерфејса.



*Структура корисничког интерфејса*

### 

## СК1: Случај коришћења – Пријављивање корисника на систем

**Назив СК**

*Пријављивање корисника на систем*

**Актори СК**

*Корисник*

**Учесници СК**

*Корисник* и систем (програм)

Предуслов: Систем је укључен. Систем приказује форму за пријављивање на систем.

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

**Основни сценарио СК:**

1. *Корисник* уноси податке потребне за пријављивање. (АПУСО)

2. *Корисник* контролише да ли је коректно унео податке потребне за пријављивање. (АНСО)

3. *Корисник* позива систем да се пријави (провери податке). (АПСО)

4. Систем проверава податке о кориснику. (СО)

5. Систем приказује кориснику поруку: **“Успешно пријављивање на систем! “.** (ИА)

**Алтернативна сценарија**

5.1. Уколико систем не може да нађе корисника, он приказује кориснику поруку: **“Неуспешно пријављивање на систем!”.** (ИА)

Graphical user interface, application

Description automatically generated

#### 

## СК2: Случај коришћења – Унос новог власника

**Назив СК**

Унос *новог власника*

**Актори СК**

*Корисник*

**Учесници СК**

*Корисник* и систем (програм)

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за унос *власника*. Учитана је листа *места.*

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

**Основни сценарио СК:**

1. *Корисник* позива систем да креира *новог власника*. (АПСО)
2. Систем креира *новог власника*. (СО)
3. Систем приказује поруку *кориснику*: **„Систем је успешно креирао власника!“**. (ИА)
4. *Корисник* уноси податке о *новом власнику*. (АПУСО)
5. *Корисник* контролише да ли је коректно унео податке о *новом власнику*. (АНСО)
6. *Корисник* позива систем да запамти податке о *новом власнику*. (АПСО)
7. Систем памти податке о *новом власнику*. (СО)
8. Систем приказује *кориснику* поруку: **„Систем је запамтио власника!“.** (ИА)

**Алтернативна сценарија**

3.1. Уколико систем не може да *креира власника*, систем приказује поруку: **„Систем не може да креира власника!“.** (ИА)

8.1. Уколико систем не може да запамти податке о *новом власнику* он приказује *кориснику* поруку **„Систем не може да запамти власника!“**. (ИА)

## СК3: Случај коришћења – Претрага власника

**Назив СК**

Претрага *власника*

**Актори СК**

*Корисник*

**Учесници СК**

*Корисник* и систем (програм)

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за претрагу *власника*.

Table

Description automatically generatedGraphical user interface

Description automatically generated with medium confidence

**Основни сценарио СК**

1. *Корисник* уноси вредности по којима претражује *власнике*. (АПУСО)
2. *Корисник* позива систем да пронађе *власнике* по задатим вредностима. (АПСО)
3. Систем претражује *власнике* по задатим вредностима. (СО)
4. Систем приказује *кориснику* пронађене *власнике* и поруку: **„Систем је пронашао власнике по задатом критеријуму“.** (ИА)
5. *Корисник* бира *власника* чије податке жели да измени. (АПУСО)
6. *Корисник* позива систем да прикаже податке о изабраном *власнику*. (АПСО)
7. Систем приказује податке о изабраном *власнику* и поруку: **„Систем је пронашао власника!“.** (ИА)

**Алтернативни сценарио СК**

4.1. Уколико систем не може да пронађе *власснике* он *кориснику* приказује поруку: **„Не постоје власници за задати критеријум!“**. (ИА)

7.1. Уколико систем не може да пронађе *власника* по задатим критеријумима, приказује поруку **„Систем не може да пронађе власника!“**. (ИА)

**Graphical user interface, application

Description automatically generated**

## СК4: Случај коришћења – Измена података о власнику

**Назив СК**

Измена података о *власнику*

**Актори СК**

*Корисник*

**Учесници СК**

*Корисник* и систем (програм)

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за претрагу *власника*. Учитана је листа *места.*

Graphical user interface, table

Description automatically generated

Graphical user interface, application

Description automatically generated

**Основни сценарио СК**

1. *Корисник* уноси вредности по којима претражује *власнике*. (АПУСО)
2. *Корисник* позива систем да пронађе *власнике* по задатим вредностима. (АПСО)
3. Систем претражује *власнике* по задатим вредностима. (СО)
4. Систем приказује *кориснику* пронађене *власнике* и поруку: **„Систем је пронашао власнике по задатом критеријуму“.** (ИА)
5. *Корисник* бира *власника* чије податке жели да измени. (АПУСО)
6. *Корисник* позива систем да прикаже податке о изабраном *власнику*. (АПСО)
7. Систем налази *власника*. (СО)
8. Систем приказује податке о изабраном *власнику* и поруку: **„Систем је пронашао власника!“.** (ИА)
9. *Корисник* мења податке о *власнику*. (АПУСО)
10. *Корисник* контролише да ли је коректно унео податке. (АНСО)
11. *Корисник* позива систем да запамти податке о *власнику*. (АПСО)
12. Систем памти податке о *власнику*. (СО)
13. Систем приказује *кориснику* поруку: **„Систем је запамтио власника!“**. (ИА)

**Алтернативни сценарио СК**

4.1. Уколико систем не може да пронађе *власника* по задатим критеријумима, приказује поруку **„Систем не може да пронађе *власнике* за задати критеријум!“**. (ИА)

8.1. Уколико систем не може да пронађе *власника* по задатим критеријумима, приказује поруку **„Систем не може да пронађе власника!“**. (ИА)

12.1. Уколико систем не може да запамти *власника*, приказује поруку **„Систем не може да запамти власника!“.** (ИА)

## СК5: Случај коришћења – Брисање власника

**Назив СК**

Брисање *власника*

**Актори СК**

*Корисник*

**Учесници СК**

*Корисник* и систем (програм)

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за претрагу *власника*.

Graphical user interface, table

Description automatically generated

Graphical user interface

Description automatically generated

**Основни сценарио СК**

1. *Корисник* уноси вредности по којима претражује *власнике*. (АПУСО)
2. *Корисник* позива систем да пронађе *власнике* по задатим вредностима. (АПСО)
3. Систем претражује *власнике* по задатим вредностима. (СО)
4. Систем приказује *кориснику* пронађене *власнике* и поруку: **„Систем је пронашао власнике по задатом критеријуму“.** (ИА)
5. *Корисник* бира *власника* чије податке жели да измени. (АПУСО)
6. *Корисник* позива систем да прикаже податке о изабраном *власнику*. (АПСО)
7. Систем налази *власника*. (СО)
8. Систем приказује податке о изабраном *власнику* и поруку: **„Систем је пронашао власника!“.** (ИА)
9. *Корисник* позива систем да обрише податке о изабраном *власнику*. (АПСО)
10. Систем брише податке о *власнику.* (СО)
11. Систем брише изабраног *власника* и приказује поруку: **„Систем је обрисао власника!“**. (ИА)

**Алтернативни сценарио СК**

4.1. Уколико систем не може да пронађе *власника*  по задатим критеријумима, приказује поруку „**Систем не може да пронађе власнике за задати критеријум!“**. (ИА)

7.1. Уколико систем не може да пронађе *власника* по задатим критеријумима, приказује поруку **„Систем не може да пронађе власника!“**. (ИА)

11.1. Уколико систем не може да обрише *власника*, приказује поруку **„Систем не може да обрише власника”**. (ИА)

## СК6: Случај коришћења – Унос нове парцеле

**Назив СК**

Унос *новe парцеле*

**Актори СК**

*Корисник*

**Учесници СК**

*Корисник* и систем (програм)

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за унос *парцеле*. Учитана је листа *власника, типова парцела и места.*

Graphical user interface, application

Description automatically generatedGraphical user interface

Description automatically generated

**Основни сценарио СК**

1. *Корисник* позива систем да креира *нову парцелу*. (АПСО)
2. Систем креира *нову парцелу*. (СО)
3. Систем приказује поруку *кориснику*: **„Систем је успешно креирао нову парцелу“**. (ИА)
4. *Корисник* уноси податке за *нову парцелу*. (АПУСО)
5. *Корисник* контролише да ли је коректно унео податке о *новој парцели*. (АНСО)
6. *Корисник* позива систем да запамти податке о *новој парцели*. (АПСО)
7. Систем памти податке о *новој парцели*. (СО)
8. Систем приказује поруку *кориснику*: **„Систем је запамтио парцелу“.** (ИА)

**Алтернативни сценарио СК**

3.1. Уколико систем не може да креира *парцелу*, систем приказује поруку: **„Систем не може да креира парцелу!“**. (ИА)

8.1. Уколико систем не може да запамти *стан*, систем приказује поруку: **„Систем не може да запамти парцелу!“**. (ИА)

## СК7: Случај коришћења – Претрага парцела

**Назив СК**

Претрага *парцела*

**Актори СК**

*Корисник*

**Учесници СК**

*Корисник* и систем (програм)

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за претрагу *парцела*.

Table

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Основни сценарио СК**

1. *Корисник* уноси вредности по којима претражује *парцеле*. (АПУСО)
2. *Корисник* позива систем да пронађе *парцеле* по задатим вредностима. (АПСО)
3. Систем претражује *парцеле* по задатим вредностима. (СО)
4. Систем приказује *кориснику* пронађене *парцеле* и поруку: **„Систем је пронашао парцеле по задатом критеријуму“.** (ИА)
5. *Корисник* бира *парцелу* чије податке жели да прикаже. (АПУСО)
6. *Корисник* позива систем да прикаже податке о изабраној *парцели*. (АПСО)
7. Систем претражује *парцелу*. (СО)
8. Систем приказује податке о изабраној *парцели* и поруку: **„Систем је пронашао парцелу!“.** (ИА)

**Алтернативни сценарио СК**

4.1. Уколико систем не може да пронађе *парцелу* по задатим критеријумима, приказује поруку **„Систем не може да пронађе парцелу за задати критеријум!“**. (ИА)

8.1. Уколико систем не може да пронађе *парцелу* по задатим критеријумима, приказује поруку **„Систем не може да пронађе парцелу!“**. (ИА)

Graphical user interface, application

Description automatically generated

## СК8: Случај коришћења – Измена података о парцели

**Назив СК**

Измена података о *парцели*

**Актори СК**

*Корисник*

**Учесници СК**

*Корисник* и систем (програм)

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за претрагу *парцела*. Учитана је листа *власника, типова парцела и места.*

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

**Основни сценарио СК**

1. *Корисник* уноси вредности по којима претражује *парцеле*. (АПУСО)
2. *Корисник* позива систем да пронађе *парцеле* по задатим вредностима. (АПСО)
3. Систем претражује *парцеле* по задатим вредностима. (СО)
4. Систем приказује *кориснику* пронађене *парцеле* и поруку: **„Систем је пронашао парцеле по задатом критеријуму“.** (ИА)
5. *Корисник* бира *парцелу* чије податке жели да прикаже. (АПУСО)
6. *Корисник* позива систем да прикаже податке о изабраној *парцели*. (АПСО)
7. Систем претражује *парцелу*. (СО)
8. Систем приказује податке о изабраној *парцели* и поруку: **„Систем је пронашао парцелу!“.** (ИА)
9. *Корисник* мења податке о *парцели*. (АПУСО)
10. *Корисник* контролише да ли је коректно унео податке. (АНСО)
11. *Корисник* позива систем да запамти податке о *парцели*. (АПСО)
12. Систем памти податке о *парцели*. (СО)
13. Систем приказује *кориснику* поруку: **„Систем је запамтио парцелу!“**. (ИА)

**Алтернативни сценарио СК**

4.1. Уколико систем не може да пронађе *парцелу* по задатим критеријумима, приказује поруку **„Систем не може да пронађе *парцелу* за задати критеријум!“**. (ИА)

8.1. Уколико систем не може да пронађе *парцелу* по задатим критеријумима, приказује поруку **„Систем не може да пронађе парцелу!“**. (ИА)

13.1. Уколико систем не може да запамти *парцелу*, приказује поруку **„Систем не може да запамти парцелу!“.** (ИА)

## СК9: Случај коришћења – Унос новог купца

**Назив СК**

Унос *новог купца*

**Актори СК**

*Корисник*

**Учесници СК**

*Корисник* и систем (програм)

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за унос *купца*. Учитана је листа *места.*

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

**Основни сценарио СК**

1. *Корисник* позива систем да креира *новог купца*. (АПСО)
2. Систем креира *новог купца.* (СО)
3. Систем приказује поруку *кориснику*: **„Систем је успешно креирао новог купца“**. (ИА)
4. *Корисник* уноси податке за *новог купца*. (АПУСО)
5. *Корисник* контролише да ли је коректно унео податке о *новом купцу*. (АНСО)
6. *Корисник* позива систем да запамти податке о *новом купцу*. (АПСО)
7. Систем памти податке о *новом купцу*. (СО)
8. Систем приказује поруку *кориснику*: **„Систем је запамтио купца“.** (ИА)

**Алтернативни сценарио СК**

3.1. Уколико систем не може да креира *купца*, систем приказује поруку: **„Систем не може да креира купца!“**. (ИА)

8.1. Уколико систем не може да запамти *купца*, систем приказује поруку: **„Систем не може да запамти купца!“**. (ИА)

## СК10: Случај коришћења – Унос нове евиденције о продаји

**Назив СК**

Унос *нове евиденције о продаји*

**Актори СК**

*Корисник*

**Учесници СК**

*Корисник* и систем (програм)

Graphical user interface, application

Description automatically generated**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за унос *нове евиденције о продаји.* Учитана је листа *парцела* и *купаца.*

Graphical user interface, application

Description automatically generated

**Основни сценарио СК**

1. *Корисник* бира купца за којег жели да унесе *нову евиденцију о продаји.* (АПУСО)
2. *Корисник* уноси податке о *новој евиденцији о продаји*. (АПУСО)
3. *Корисник* контролише да ли је коректно унео податке о *новој евиденцији о продаји*. (АНСО)
4. *Корисник* позива систем да запамти податке о *новој евиденцији о продаји.* (АПСО)
5. Систем памти податке о *новој евиденцији о продаји.* (СО)
6. Систем приказује *кориснику* поруку „Систем је успешно извршио чување!“. (ИА)

**Алтернативни сценарио СК**

6.1. Уколико систем не може да запамти податке о *новој продаји*, приказује поруку**: „Систем не може да изврши чување“.** (ИА)

Graphical user interface, application

Description automatically generated

## СК11: Случај коришћења – Претрага евиденција о продаји

**Назив СК**

Претрага *евиденција о родаји*

**Актори СК**

*Корисник*

**Учесници СК**

*Корисник* и систем (програм)

**Предуслов:** Систем је укључен и *корисник* је пријављен под својом шифром. Систем приказује форму за претрагу *евиденција о продаји*.

Graphical user interface, text, application

Description automatically generated

Table

Description automatically generated

Graphical user interface, table

Description automatically generated

**Основни сценарио СК**

1. *Корисник* уноси вредности по којима претражује *евиденције о продаји*. (АПУСО)
2. *Корисник* позива систем да пронађе *евиденције о продаји* по задатим вредностима. (АПСО)
3. Систем претражује *евиденције о продаји* по задатим вредностима. (СО)
4. Систем приказује *кориснику* пронађене *евиденције о продаји* и поруку: **„Систем је пронашао евиденције о продаји за задати критеријум!“**. (ИА)
5. *Корисник* бира *продају* чије податке жели да прикаже. (АПУСО)
6. *Корисник* позива систем да прикаже податке о изабраној *продаји*. (АПСО)
7. Систем претражује *продаје*. (СО)
8. Систем приказује податке о изабраној *продаји* и поруку: **„Систем је пронашао продају!“.** (ИА)

**Алтернативни сценарио СК**

4.1. Уколико систем не може да пронађе евиденцију о *продаји* за изабрани критеријум, приказује поруку: **„Не постоје евиденције о продаји за задати критеријум!“.** (ИА)

8.1. Уколико систем не може да пронађе *продају* по задатим критеријумима, приказује поруку **„Систем не може да пронађе продају!“**. (ИА)

Graphical user interface, text, application

Description automatically generated

### 3.2.2 Пројектовање контролера корисничког интерфејса

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

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

## 3.3 Пројектовање апликационе логике

### 3.3.1 Контролер апликационе логике

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

Клијент шаље захтев за извршење неке од СО до одговарајуће нити (коју смо назвали “Nit klijenta”), која је повезана са тим клијентом. “Nit klijenta” прима захтев и даље га преусмерава до класа које су одговорне за извршење СО. Након извршења СО резултат се враћа до апликационе логике, односно до “Niti klijenta”, која тај резултат шаље назад до клијента.

**Пример специфичног контролера апликационе логике**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Domen;

using System.Net;

using System.Net.Sockets;

using System.Runtime.Serialization.Formatters.Binary;

using System.Threading;

namespace Klijent

{

public class Komunikacija

{

TcpClient klijent;

NetworkStream tok;

BinaryFormatter formater;

public bool poveziSeNaServer()

{

try

{

klijent = new TcpClient("localhost", 20000);

tok = klijent.GetStream();

formater = new BinaryFormatter();

return true;

}

catch (Exception)

{

return false;

}

}

public void Kraj()

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.Kraj;

formater.Serialize(tok, transfer);

}

public Object KreirajKupca()

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.KreirajKupca;

transfer.TransferObjekat = new Kupac();

formater.Serialize(tok, transfer);

return ((TransferKlasa)formater.Deserialize(tok)).Rezultat;

}

public Object ZapamtiKupca(Kupac k)

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.ZapamtiKupca;

transfer.TransferObjekat = k;

formater.Serialize(tok, transfer);

return ((TransferKlasa)formater.Deserialize(tok)).Rezultat;

}

public Object KreirajVlasnika()

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.KreirajVlasnika;

transfer.TransferObjekat = new Vlasnik();

formater.Serialize(tok, transfer);

return ((TransferKlasa)formater.Deserialize(tok)).Rezultat;

}

public Object ZapamtiVlasnika(Vlasnik v)

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.ZapamtiVlasnika;

transfer.TransferObjekat = v;

formater.Serialize(tok, transfer);

return ((TransferKlasa)formater.Deserialize(tok)).Rezultat;

}

public Object PrikaziVlasnika(Vlasnik v)

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.PrikaziVlasnika;

transfer.TransferObjekat = v;

formater.Serialize(tok, transfer);

return ((TransferKlasa)formater.Deserialize(tok)).Rezultat;

}

public Object PretraziVlasnike(Vlasnik v)

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.PretraziVlasnike;

transfer.TransferObjekat = v;

formater.Serialize(tok, transfer);

return ((TransferKlasa)formater.Deserialize(tok)).Rezultat;

}

public Object ObrisiVlasnika(Vlasnik v)

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.ObrisiVlasnika;

transfer.TransferObjekat = v;

formater.Serialize(tok, transfer);

return ((TransferKlasa)formater.Deserialize(tok)).Rezultat;

}

public Object KreirajParcelu()

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.KreirajParcelu;

transfer.TransferObjekat = new Parcela();

formater.Serialize(tok, transfer);

return ((TransferKlasa)formater.Deserialize(tok)).Rezultat;

}

public Object ZapamtiParcelu(Parcela p)

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.ZapamtiParcelu;

transfer.TransferObjekat = p;

formater.Serialize(tok, transfer);

return ((TransferKlasa)formater.Deserialize(tok)).Rezultat;

}

public Object PretraziParcele(Parcela p)

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.PretraziParcele;

transfer.TransferObjekat = p;

formater.Serialize(tok, transfer);

return ((TransferKlasa)formater.Deserialize(tok)).Rezultat;

}

public Object PrikaziParcele(Parcela p)

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.PrikaziParcelu;

transfer.TransferObjekat = p;

formater.Serialize(tok, transfer);

return ((TransferKlasa)formater.Deserialize(tok)).Rezultat;

}

public Object VratiSveTipoveParcela()

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.VratiSveTipove;

transfer.TransferObjekat = new TipParcele();

formater.Serialize(tok, transfer);

return ((TransferKlasa)formater.Deserialize(tok)).Rezultat;

}

public Object VratiSveVlasnike()

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.VratiSveVlasnike;

transfer.TransferObjekat = new Vlasnik();

formater.Serialize(tok, transfer);

return ((TransferKlasa)formater.Deserialize(tok)).Rezultat;

}

public Object VratiSveKupce()

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.VratiSveKupce;

transfer.TransferObjekat = new Kupac();

formater.Serialize(tok, transfer);

return ((TransferKlasa)formater.Deserialize(tok)).Rezultat;

}

public Object VratiSvaMesta()

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.VratiSvaMesta;

transfer.TransferObjekat = new Mesto();

formater.Serialize(tok, transfer);

return ((TransferKlasa)formater.Deserialize(tok)).Rezultat;

}

public Object zapamtiProdaje(Kupac k)

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.ZapamtiProdaje;

transfer.TransferObjekat = k;

formater.Serialize(tok, transfer);

return ((TransferKlasa)formater.Deserialize(tok)).Rezultat;

}

public Object PretraziProdaje(Prodaja p)

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.PretraziProdaje;

transfer.TransferObjekat = p;

formater.Serialize(tok, transfer);

return ((TransferKlasa)formater.Deserialize(tok)).Rezultat;

}

public Object PrikaziProdaju(Prodaja p)

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.PrikaziProdaju;

transfer.TransferObjekat = p;

formater.Serialize(tok, transfer);

return ((TransferKlasa)formater.Deserialize(tok)).Rezultat;

}

public Object VratiSveParcele()

{

TransferKlasa transfer = new TransferKlasa();

transfer.Operacija = Operacije.VratiSveParcele;

transfer.TransferObjekat = new Parcela();

formater.Serialize(tok, transfer);

return ((TransferKlasa)formater.Deserialize(tok)).Rezultat;

}

}

}

### 3.3.2 Системске операције

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

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

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using Biblioteka;

using Sesija;

namespace SistemskeOperacije

{

public abstract class OpstaSO

{

public Object izvrsiSO(OpstiDomenskiObjekat odo)

{

Object rezultat = null;

Broker.dajSesiju().otvoriKonekciju();

Broker.dajSesiju().zapocniTransakciju();

try

{

rezultat = Izvrsi(odo);

Broker.dajSesiju().potvrdiTransakciju();

}

catch (Exception)

{

Broker.dajSesiju().ponistiTransakciju();

}

finally

{

Broker.dajSesiju().zatvoriKonekciju();

}

return rezultat;

}

public abstract object Izvrsi(OpstiDomenskiObjekat odo);

}

}

Опис методе: Пре извршења системске операције отвара се конекција и покреће трансакција. Уколико дође до изузетака приликом извршења системске операције, поништава се трансакција (*rollback*), у супротном се потврђује (*commit*).

**Уговор УГ1: Пријави корисника**

**Операција: PrijaviSe(Korisnik): signal;**

**Веза са СК: СК1**

**Предуслови:-**

**Постуслови: Корисник је пријављен на систем.**

**Diagram

Description automatically generated with medium confidence**

**Уговор УГ1: Врати листу места**

**Операција**: **VratiListuMesta**(List<Mesto>): *Signal*;

**Веза са СК**: СК2, СК4, СК6, СК8, СК9

**Предуслови: -**

**Постуслови: -**



**Уговор УГ2: Креирај Власника**

**Операција:** **KreirajVlasnika**(Vlasnik): *Signal*;

**Веза са СК:** СК2

**Предуслови**: Просто вредносно ограничење над објектом Власник мора бити задовољено. Структурно ограничење над објектом Власник мора бити задовољено.

**Постуслови**: Креиран је нови Валсник.



**Уговор УГ3: Запамти Власника**

**Операција:** **ZapamtiVlasnika**(Vlasniik): *Signal*;

**Веза са СК:** СК2, СК4

**Предуслови**: Просто вредносно ограничење над објектом Власник мора бити задовољено. Структурно ограничење над објектом Власник мора бити задовољено.

**Постуслови**: Подаци о Власнику су запамћени.



**Уговор УГ4: Претражи Власнике**

**Операција**: **PretraziVlasnike**(Kriterijum, List<Vlasnik>): *Signal*;

**Веза са СК**: СК3, СК4, СК5

**Предуслови: -**

**Постуслови: -**



**Уговор УГ5: Прикажи власника**

**Операција**: **PrikaziVlasnika**(Vlasnik): *Signal*;

**Веза са СК**: СК3, СК4, СК5

**Предуслови: -**

**Постуслови: -**



**Уговор УГ6: Обриши Власника**

**Операција:** **ObrisiVlasnikа**(Vlasnik):сигнал;

**Веза са СК**: СК5

**Предуслови**: Структурно ограничење над објектом Власник мора бити задовољено.

**Постуслови**: Власник је обрисан.



**Уговор УГ7: Врати листу власника**

**Операција**: **VratiListuVlasnika**(List<Vlasnik>): *Signal*;

**Веза са СК**: СК6, СК8

**Предуслови: -**

**Постуслови: -**



**Уговор УГ8: Врати листу типова парцела**

**Операција**: **VratiListuKultura**(List<Kultura>): *Signal*;

**Веза са СК**: СК6, СК8

**Предуслови: -**

**Постуслови: -**

Diagram

Description automatically generated

**Уговор УГ9: Креирај Парцелу**

**Операција:** **KreirajParcelu**(Parcela): *Signal*;

**Веза са СК:** СК6

**Предуслови**: Просто вредносно ограничење над објектом Парцела мора бити задовољено. Структурно ограничење над објектом Парцела мора бити задовољено.

**Постуслови**: Креирана је нова парцела.



**Уговор УГ10: Запамти Парцелу**

**Операција:** **ZapamtiParcelu**(Parcela): *Signal*;

**Веза са СК:** СК6, СК8

**Предуслови**: Просто вредносно ограничење над објектом Парцела мора бити задовољено. Структурно ограничење над објектом Парцела мора бити задовољено.

**Постуслови**: Подаци о Парцели су запамћени.



**Уговор УГ11: Претражи Парцеле**

**Операција**: **PretraziParcele**(Kriterijum, List<Parcela>): *Signal*;

**Веза са СК**: СК7, СК8

**Предуслови: -**

**Постуслови: -**



**Уговор УГ12: Прикажи Парцелу**

**Операција**: **PrikaziParcelu**(Parcela): *Signal*;

**Веза са СК**: СК7, СК8

**Предуслови: -**

**Постуслови: -**



**Уговор УГ13: Креирај Купца**

**Операција:** **KreirajKupca**(Kupac): *Signal*;

**Веза са СК:** СК9

**Предуслови**: Просто вредносно ограничење над објектом Купац мора бити задовољено. Структурно ограничење над објектом Купац мора бити задовољено.

**Постуслови**: Креиран је нови купац.



**Уговор УГ14: Запамти Купца**

**Операција:** **ZapamtiKupca**(Kupac): *Signal*;

**Веза са СК:** СК9

**Предуслови**: Просто вредносно ограничење над објектом Купац мора бити задовољено. Структурно ограничење над објектом Купац мора бити задовољено.

**Постуслови**: Подаци о Купцу су запамћени.



**Уговор УГ15: Врати листу парцела**

**Операција**: **VratiListuParcela**(List<Parcela>): *Signal*;

**Веза са СК**: СК10

**Предуслови: -**

**Постуслови: -**



**Уговор УГ16: Врати листу Купаца**

**Операција**: **VratiListuKupaca**(List<Kupac>): *Signal*;

**Веза са СК**: СК10

**Предуслови: -**

**Постуслови: -**



**Уговор УГ17: Запамти Продаје**

**Операција:** **ZapamtiProdaje**(Kupac): *Signal*;

**Веза са СК:** СК10

**Предуслови**: Просто вредносно ограничење над објектом Продаја мора бити задовољено. Структурно ограничење над објектом Продаја мора бити задовољено.

**Постуслови**: Подаци о Продаји су запамћени.



**Уговор УГ18: Претражи Продаје**

**Операција**: **PretraziProdaje**(Kriterijum, List<Prodaja>): *Signal*;

**Веза са СК**: СК11

**Предуслови: -**

**Постуслови: -**



**Уговор УГ19: Прикажи продају**

**Операција**: **PrikaziProdaju**(Prodaja): *Signal*;

**Веза са СК**: СК11

**Предуслови: -**

**Постуслови: -**



*Diagram

Description automatically generated*

*Класе које су одговорне за SO наслеђују класу OpstaSO*

### 

### 3.3.3 Доменске класе

На основу концептуалних класа праве се софтверске класе структуре. Свака класа садржи приватна поља атрибута, getter-e, setter-e за исте, конструкторе (било безпараметризовани било параметризовани).

Доменске класе имплементирају интерпретирају интерфејс OpstiDomenskiObjekat како би омогућили лакше имплементирање метода Broker-a. На тај начин Broker прима интерфејс уместо самих класа и олакшава креирање генеричких упита.

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Domen

{

public interface OpstiDomenskiObjekat

{

string nazivTabele { get; }

string primarniKljuc { get; }

string uslovPrimarni { get; }

string uslovOstalo { get; }

string izmena { get; }

string unos { get; }

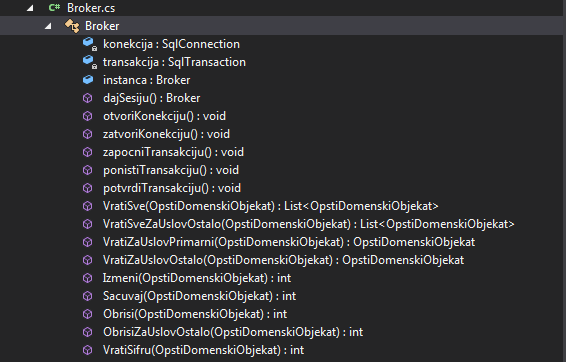
OpstiDomenskiObjekat procitaj(DataRow red);

}

}

### 3.3.4 Брокер базе података

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



*Брокер*

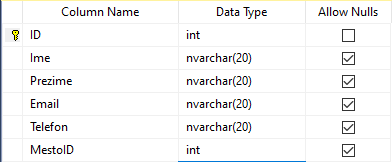
Diagram

Description automatically generated

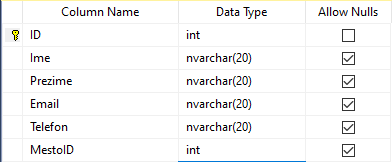
*Слика: Database брокер класа се повезује са класом OpstiDomenskiObjekat*

### 3.3.5 Пројектовање складишта података

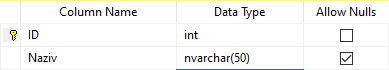
На основу структуре софтверских класа пројектоване су табеле (складишта података) релационог система за управљање базом података. У овом раду је коришћен MSSQL.



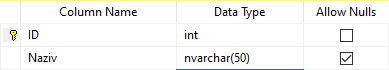
**КУПАЦ**



**ВЛАСНИК**



**МЕСТО**

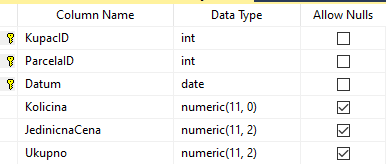


**КУЛТУРА**

**Table

Description automatically generated**

**ПАРЦЕЛА**



**ПРОДАЈА**

**Diagram

Description automatically generated**

*Сликa: ASSTBP Архитектура софт. Система након пројектовања табела базе података*

# 4. ИМПЛЕМЕНТАЦИЈА

Софтверски систем, резултат овог рада, развијен је у програмском језику C#. Систем је пројектован као клијент-сервер. Као развојно окружење коришћен је Visual Stiudio 2017. Као систем за управљање базом података коришћен је MSSQL. На основу архитектуре софтверског система добијене су следеће софтверске класе:

Text

Description automatically generatedText

Description automatically generatedGraphical user interface, text

Description automatically generated

*Слика Имплементација*

# 5. ТЕСТИРАЊЕ

Након имплементације уследило је тестирање сваког од случајева коришћења.

Тестирање је обављено тако што су уношени и валидни и невалидни подаци. Утврђене грешке или слабије функционалности су исправљене. Закључак је да софтверски систем у потпуности функционише и испуњава све задате критеријуме.

# 6. ЛИТЕРАТУРА

Влајић, С. (2015). Пројектовање софтвера (Скрипта). Београд