

Универзитет у Београду

**Факултет организационих наука**

Лабораторија за софтверско инжењерство (СИЛАБ)

Предмет: Пројектовање софтвера

Тема: Софтверски систем за праћење обавеза у теретани у Јава окружењу

Студент: Едис Скендери 297/17 Ментор: Проф. Др Синиша Влајић

Београд, 2021.

Садржај

[1. Фаза прикупљања корисничких захтева 2](#_Toc113218368)

[1.1. Вербални опис 2](#_Toc113218369)

[1.2. Случајеви коришћења 2](#_Toc113218370)

[СК1: Случај коришћења – Креирање налога запосленима 3](#_Toc113218371)

[СК2: Случај коришћења – Додавање нове обавезе 4](#_Toc113218372)

[СК3: Случај коришћења – Претраживање обавеза 5](#_Toc113218373)

[СК4: Случај коришћења – Претраживање запослених 6](#_Toc113218374)

[СК5: Случај коришћења – Промена података о обавези 7](#_Toc113218375)

[СК6: Случај коришћења – Промена података о запосленом 8](#_Toc113218376)

[СК7: Случај коришћења – Креирање новог захтева (сложен ск) 9](#_Toc113218377)

[СК8: Случај коришћења – Промена захтева (сложен ск) 10](#_Toc113218378)

[СК9: Случај коришћења – Брисање захтева (сложен ск) 11](#_Toc113218379)

[СК10: Пријављивање администратора на систем 12](#_Toc113218380)

[2. Фаза анализе 13](#_Toc113218381)

[2.1. Системски дијаграми секвенци 13](#_Toc113218382)

[ДС1: Дијаграм секвенце случаја коришћења – Креирање налога запосленима 13](#_Toc113218383)

[ДС2: Дијаграм секвенце случаја коришћења – Додавање нове обавезе 14](#_Toc113218384)

[ДС3: Дијаграм секвенце случаја коришћења – Претраживање обавеза 15](#_Toc113218385)

[ДС4: Дијаграм секвенце случаја коришћења – Претраживање запослених 16](#_Toc113218386)

[ДС5: Дијаграм секвенце случаја коришћења – Измена података о обавези 17](#_Toc113218387)

[ДС6: Дијаграм секвенце случаја коришћења – Измена података о запосленима 19](#_Toc113218388)

[ДС7: Дијаграм секвенце случаја коришћења – Креирање захтева 21](#_Toc113218389)

[ДС8: Дијаграм секвенце случаја коришћења – Измена захтева 22](#_Toc113218390)

[ДС9: Дијаграм секвенце случаја коришћења – Брисање захтева 23](#_Toc113218391)

[ДС10: Дијаграм секвенце случаја коришћења – Пријављивање администратора на систем 24](#_Toc113218392)

[2.2. Понашање софтверског система – Дефисинање уговора о системским операцијама 26](#_Toc113218393)

[2.3. Структура софтверског система – Концептуални (доменски) модел 28](#_Toc113218394)

[2.4. Структура софтверског система – Релациони модел 28](#_Toc113218395)

[3. Пројектовање 32](#_Toc113218396)

[3.1. Пројектовање корисничког интерфејса 32](#_Toc113218397)

[3.1.1. Пројектовање екранских форми 33](#_Toc113218398)

[СК1: Случај коришћења – Креирање налога запосленима 35](#_Toc113218399)

[СК2: Случај коришћења – Додавање нове обавезе 37](#_Toc113218400)

[СК3: Случај коришћења – Претраживање обавеза 39](#_Toc113218401)

[СК4: Случај коришћења – Претраживање запослених 41](#_Toc113218402)

[СК5: Случај коришћења – Промена података о обавези 43](#_Toc113218403)

[СК6: Случај коришћења – Промена података о запосленом 46](#_Toc113218404)

[СК7: Случај коришћења – Креирање новог захтева (сложен ск) 49](#_Toc113218405)

[СК8: Случај коришћења – Промена захтева (сложен ск) 51](#_Toc113218406)

[СК9: Случај коришћења – Брисање захтева (сложен ск) 53](#_Toc113218407)

[СК10: Случај коришћења – Пријављивање администратора на систем 55](#_Toc113218408)

[3.1.2. Пројектовање контролера корисничког интерфејса 57](#_Toc113218409)

[3.2. Пројектовање апликационе логике 57](#_Toc113218410)

[3.2.1. Контролер аплиакционе логике 59](#_Toc113218411)

[3.2.2. Пословна логика 60](#_Toc113218412)

[3.2.3. Брокер базе података 68](#_Toc113218413)

[3.3. Пројектовање складишта података 69](#_Toc113218414)

[4. Имплементација 70](#_Toc113218415)

[5. Тестирање 73](#_Toc113218416)

# Фаза прикупљања корисничких захтева

## Вербални опис

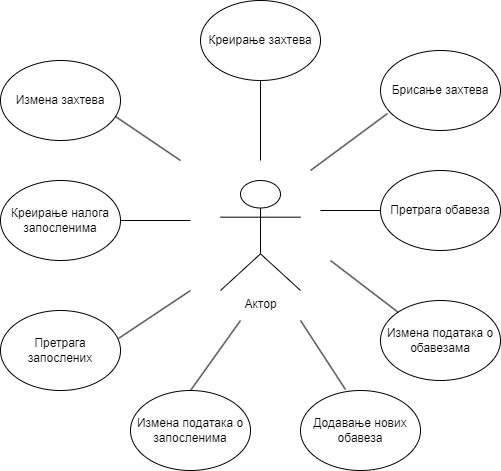
Задатак је направити десктоп апликацију за фирму која се бави пружањем услуга управљања теретаном у Јава програмском окружењу. Овај софтверски систем има за циљ да олакша организацију рада саме теретане. Помоћу апликације ће се водити евиденција о запосленима као и о врсти обавезе која се захтева од сваког запосленог. Тако ће запослени у сваком тренутку знати који се захтеви односе на њега и какве су му одговорности у току радног дана. Сваки запослени ће моћи да приступи апликацији уносом свог корисничког имена и лозинке.

Апликација омогућава креирање налога новим запосленима, додавање нових обавеза, као и повезивање запослених са обавезама за које су задужени помоћу захтева. Потребно је омогућити претрагу доступних обавеза, измену података самих обавеза, провера неизвршених обавеза као и увид у податке свих запослених.

## Случајеви коришћења

У овој апликацији постоји 10 случајева коришћења:

1. Креирање налога запосленима
2. Претрага запослених
3. Додавање нових обавеза
4. Претрага обавеза
5. Измена података о обавезама
6. Измена података о запосленима
7. Креирање захтева (сложен ск)
8. Измена захтева
9. Брисање захтева



*Слика 1 Случајеви коришћења*

### СК1: Случај коришћења – Креирање налога запосленима

#### Назив СК

Креирање новог налога

#### Актори СК

Администратор

#### Учесници СК

Администратор и систем (програм)

**Предуслов**: Систем је укључен и администратор je улогован под својом шифром. Систем приказује форму за рад са налогом. Учитана је листа општина.

#### Основни сценарио СК

1. Администратор **уноси** податке у нови налог. (АПУСО)
2. Администратор **контролише** да ли је коректно унео податке у нови налог. (АНСО)
3. Администратор **позива** систем да креира нови налог. (АПСО)
4. Систем **креира** нови налог. (СО)
5. Систем **приказује** администратору нови налог и поруку: “Систем је креирао нови налог“. (ИА)

Алтернативна сценарија

5.1 Уколико систем не може да креира нови налог он приказује администратору поруку: “Систем не може да креира нови налог”. (ИА)

### СК2: Случај коришћења – Додавање нове обавезе

#### Назив СК

Креирање нове обавезе

#### Актори СК

Администратор

#### Учесници СК

Администратор и систем (програм)

**Предуслов**: Систем је укључен и администратор je улогован под својом шифром. Систем приказује форму за рад са обавезом.

#### Основни сценарио СК

1. Администратор **уноси** податке у нову обавезу. (АПУСО)
2. Администратор **контролише** да ли је коректно унео податке у нову обавезу. (АНСО)
3. Администратор **позива** систем да креира нову обавезу. (АПСО)
4. Систем **креира** нову обавезу. (СО)
5. Систем **приказује** администратору нову обавезу и поруку: “Систем је креирао нову обавезу“. (ИА)

Алтернативна сценарија

5.1 Уколико систем не може да креира нову обавезу он приказује администратору поруку: “Систем не може да креира нову обавезу”. (ИА)

### СК3: Случај коришћења – Претраживање обавеза

#### Назив СК

Претраживање обавеза

#### Актори СК

Администратор

#### Учесници СК

Администратор и систем (програм)

**Предуслов**: Систем је укључен и администратор je улогован под својом шифром. Систем приказује форму за рад са обавезама.

#### Основни сценарио СК

1. Администратор **уноси** вредност по којој претражује обавезе. (АПУСО)
2. Администратор **позива** систем да нађе обавезе по задатој вредности. (АПСО)
3. Систем **тражи** обавезе по задатој вредности. (СО)
4. Систем приказује администратору податке о обавезама и поруку: “Систем је нашао обавезе по задатој вредности”. (ИА)

Алтернативна сценарија

* 1. Уколико систем не може да нађе обавезе он приказује администратору поруку: “Систем не може да нађе обавезе по задатој вредности”. (ИА)

### СК4: Случај коришћења – Претраживање запослених

#### Назив СК

Претраживање запослених

#### Актори СК

Администратор

#### Учесници СК

Администратор и систем (програм)

**Предуслов**: Систем је укључен и администратор je улогован под својом шифром. Систем приказује форму за рад са запосленима.

#### Основни сценарио СК

1. Администратор **уноси** вредност по којој претражује запослене. (АПУСО)
2. Администратор **позива** систем да нађе запослене по задатој вредности. (АПСО)
3. Систем **тражи** запослене по задатој вредности. (СО)
4. Систем приказује администратору податке о запосленима и поруку: “Систем је нашао запослене по задатој вредности”. (ИА)

Алтернативна сценарија

* 1. Уколико систем не може да нађе запослене он приказује администратору поруку: “Систем не може да нађе запосленог по задатој вредности”. (ИА)

### СК5: Случај коришћења – Промена података о обавези

#### Назив СК

Промена обавезе

#### Актори СК

Администратор

#### Учесници СК

Администратор и систем (програм)

**Предуслов**: Систем је укључен и администратор je улогован под својом шифром. Систем приказује форму за рад са обавезом.

#### Основни сценарио СК

1. Администратор **уноси** вредност по којој претражује обавезе. (АПУСО)
2. Администратор **позива** систем да нађе обавезе по задатој вредности. (АПСО)
3. Систем **тражи** обавезе по задатој вредности. (СО)
4. Систем приказује администратору обавезе и поруку: “Систем је нашао обавезе по задатој вредности”. (ИА)
5. Администратор **бира** обавезу. (АПУСО)
6. Администратор **позива** систем да учита обавезу. (АПСО)
7. Систем **учитава** обавезу. (СО)
8. Систем **приказује** администратору податке о обавези и поруку: “Систем је учитао обавезу.” (ИА)
9. Администратор **уноси (мења)** податке о обавези. (АПУСО)
10. Администратор **контролише** да ли је коректно унео податке о обавези. (АНСО)
11. Администратор **позива** систем да запамти податке о обавези. (АПСО)
12. Систем **памти** податке о обавези. (СО)
13. Систем **приказује** администратору запамћену обавезу и поруку: “Систем је запамтио обавезу.” (ИА)

Алтернативна сценарија

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

8.1 Уколико систем не може да учита обавезу он приказује администратору поруку: “Систем не може да учита обавезу”. Прекида се извршење сценариа. (ИА)

13.1 Уколико систем не може да запамти податке о обавези он приказује администратору поруку “Систем не може да запамти обавезу”. (ИА)

### СК6: Случај коришћења – Промена података о запосленом

#### Назив СК

Промена запосленог

#### Актори СК

Администратор

#### Учесници СК

Администратор и систем (програм)

**Предуслов**: Систем је укључен и администратор je улогован под својом шифром. Систем приказује форму за рад са запосленим. Учитана је листа општина.

#### Основни сценарио СК

1. Администратор **уноси** вредност по којој претражује запослене. (АПУСО)
2. Администратор **позива** систем да нађе запослене по задатој вредности. (АПСО)
3. Систем **тражи** запослене по задатој вредности. (СО)
4. Систем приказује администратору запослене и поруку: “Систем је нашао запослене по задатој вредности”. (ИА)
5. Администратор **бира** запосленог. (АПУСО)
6. Администратор **позива** систем да учита запосленог. (АПСО)
7. Систем **учитава** запосленог. (СО)
8. Систем **приказује** администратору податке о запосленом и поруку: “Систем је учитао запосленог.” (ИА)
9. Администратор **уноси (мења)** податке о запосленом. (АПУСО)
10. Администратор **контролише** да ли је коректно унео податке о запосленом. (АНСО)
11. Администратор **позива** систем да запамти податке о запосленом. (АПСО)
12. Систем **памти** податке о запосленом. (СО)
13. Систем **приказује** администратору запамћеног запосленог и поруку: “Систем је запамтио запосленог.” (ИА)

Алтернативна сценарија

4.1 Уколико систем не може да нађе запосленe он приказује администратору поруку: “Систем не може да нађе запосленог по задатој вредности”. Прекида се извршење сценариа. (ИА)

8.1 Уколико систем не може да учита запосленог он приказује администратору поруку: “Систем не може да учита запосленог”. Прекида се извршење сценариа. (ИА)

13.1 Уколико систем не може да запамти податке о запосленом он приказује администратору поруку “Систем не може да запамти запосленог”. (ИА)

### СК7: Случај коришћења – Креирање новог захтева (сложен ск)

#### Назив СК

Креирање новог захтева

#### Актори СК

Администратор

#### Учесници СК

Администратор и систем (програм)

**Предуслов**: Систем је укључен и администратор je улогован под својом шифром. Систем приказује форму за рад са захтевом. Учитане су листе запослених и обавеза.

#### Основни сценарио СК

1. Администратор **уноси** податке у нови захтев. (АПУСО)
2. Администратор **контролише** да ли је коректно унео податке у нови захтев. (АНСО)
3. Администратор **позива** систем да креира нови захтев. (АПСО)
4. Систем **креира** нови захтев. (СО)
5. Систем **приказује** администратору нови захтев и поруку: “Систем је креирао нови захтев“. (ИА)

Алтернативна сценарија

5.1 Уколико систем не може да креира нови захтев он приказује администратору поруку: “Систем не може да креира нови захтев”. (ИА)

### СК8: Случај коришћења – Промена захтева (сложен ск)

#### Назив СК

Промена захтева

#### Актори СК

Администратор

#### Учесници СК

Администратор и систем (програм)

**Предуслов**: Систем је укључен и администратор je улогован под својом шифром. Систем приказује форму за рад са захтевима. Учитана је листа захтева.

#### Основни сценарио СК

1. Администратор **бира** захтев из листе учитаних захтева. (АПУСО)
2. Систем **отвара** захтев (ИА).
3. Администратор **уноси (мења)** захтев. (АПУСО)
4. Администратор **контролише** да ли је коректно унео податке у захтев. (АНСО)
5. Администратор **позива** систем да запамти податке (нови статус) захтева. (АПСО)
6. Систем **памти** податке о захтевима. (СО)
7. Систем **приказује** администратору измењен захтев и поруку: “Систем је изменио захтев.” (ИА)

Алтернативна сценарија

7.1 Уколико систем не може да промени захтев он приказује администратору поруку: “Систем не може да промени захтев”. (ИА)

### СК9: Случај коришћења – Брисање захтева (сложен ск)

#### Назив СК

Брисање захтева

#### Актори СК

Администратор

#### Учесници СК

Администратор и систем (програм)

**Предуслов**: Систем је укључен и администратор je улогован под својом шифром. Систем приказује форму за рад са захтевима. Учитана је листа захтева.

#### Основни сценарио СК

1. Администратор **бира** захтев из листе учитаних захтева. (АПУСО)
2. Систем **отвара** захтев (ИА).
3. Систем приказује администратору податке о захтевима који нису реализовани и поруку: “Систем је нашао захтеве по задатој вредности”. (ИА)
4. Администратор **бира** захтев који жели да обрише. (АПУСО)
5. Администратор **позива** систем да обрише одабрани захтев. (АПСО)
6. Систем **брише** захтев. (СО)
7. Систем **приказује** администратору поруку: “Систем је обрисао захтев.” (ИА)

Алтернативна сценарија

7.1 Уколико систем не може да обрише захтев он приказује администратору поруку: “Систем не може да обрише захтев”. (ИА)

### СК10: Пријављивање администратора на систем

**Назив СК**

Пријављивање администратора на систем

**Актори СК**

Администратор

**Учесници СК**

Администратор и систем (програм)

**Предуслов:** Систем је укључен и **приказује** форму за пријављивање актора.

**Основни сценарио СК**

1. Администратор **уноси** податке за пријаву. (АПУСО)

2. Администратор **контролише** да ли је коректно **унео** податке за пријаву. (АНСО)

3. Администратор **позива** систем да **пронађе** администратора са задатим подацима.(АПСО)

4. Систем **претражује** администраторе. (СО)

5. Систем **проналази** администратора. (ИА)

6. Систем **отвара** администраторску форму. (ИА)

**Алтернативна сценарија:**

5.1 Уколико систем није **пронашао** администратора, **приказује** администратору поруку: “Не постоји члан/администратор са тим креденцијалима ”, прекида се извршење сценарија. (ИА)

5.2 Уколико систем није успео да **пронађе** администратора услед прекида комуникације са сервером, систем **приказује** администратору поруку: “Комуникација између сервера је прекинута ”. (ИА)

# Фаза анализе

## Системски дијаграми секвенци

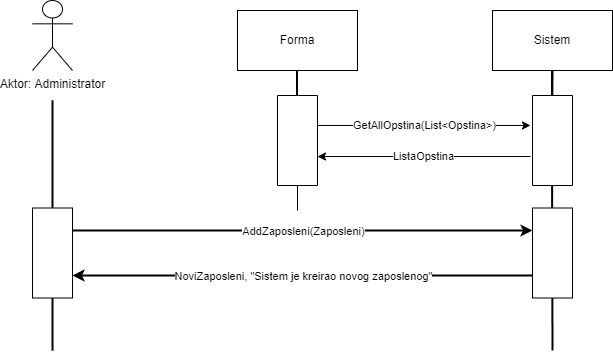
### ДС1: Дијаграм секвенце случаја коришћења – Креирање налога запосленима

1. Форма **позива** систем да учита листу општина. (АПСО)

2. Систем **враћа** форми листу општина. (ИА)

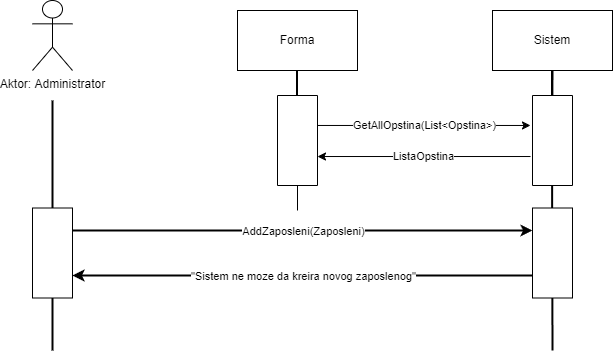
3. Администратор **позива** систем да креира новог запосленог. (АПСО)

4. Систем **приказује** администратору нови налог и поруку: „Систем је креирао новог запосленог“. (ИА)



*Алтернативна сценарија*

4.1 Уколико систем не може да креира корисника он приказује администратору поруку: „Систем не може да креира новог запосленог“. (ИА)



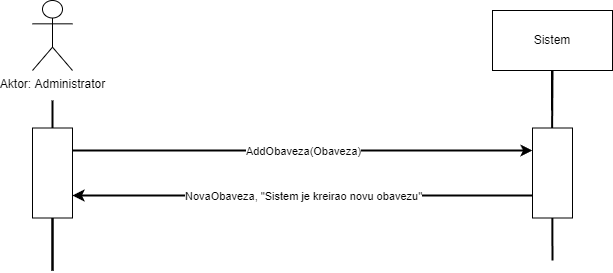
Са наведених секвенцних дијаграма уочавају се 2 системске операције:

1. *Signal* GetAllOpstina(List<Opstina>)
2. *Signal* AddZaposleni(Zaposleni)

### ДС2: Дијаграм секвенце случаја коришћења – Додавање нове обавезе

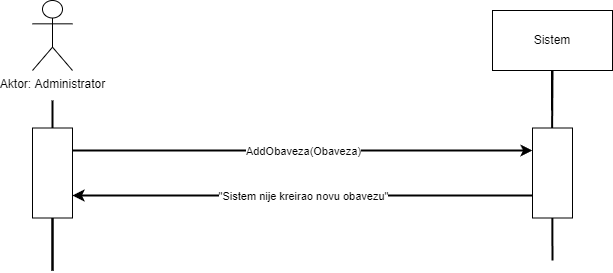
1. Администратор **позива** систем да креира нову обавезу. (АПСО)

2. Систем **приказује** администратору нову обавезу и поруку: „Систем је креирао нову обавезу“. (ИА)



*Алтернативна сценарија*

2.1. Уколико систем не може да креира услугу он приказује администратору поруку: „Систем не може да креира нову обавезу“. (ИА)



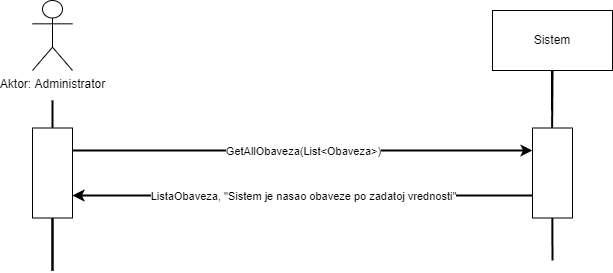
Са наведених секвенцних дијаграма уочава се једна системска операција:

1. *Signal* AddObaveza(Obaveza)

### ДС3: Дијаграм секвенце случаја коришћења – Претраживање обавеза

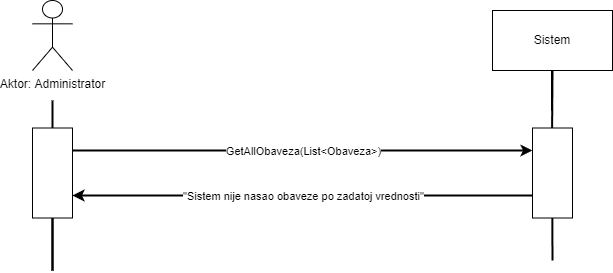
1. Администратор **позива** систем да нађе обавезе по задатој вредности. (АПСО)

2. Систем приказује администратору податке о обавезама и поруку: “Систем је нашао обавезе по задатој вредности”. (ИА)



*Алтернативна сценарија*

2.1. Уколико систем не може да нађе обавезе он приказује администратору поруку: „Систем не може да нађе обавезе по задатој вредности“. (ИА)



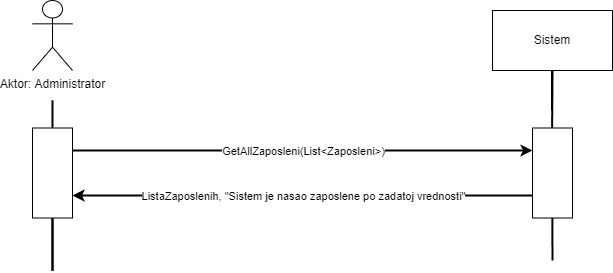
Са наведених секвенцних дијаграма уочава се једна системска операција:

1. *Signal* GetAllObaveza(List<Obaveza>)

### ДС4: Дијаграм секвенце случаја коришћења – Претраживање запослених

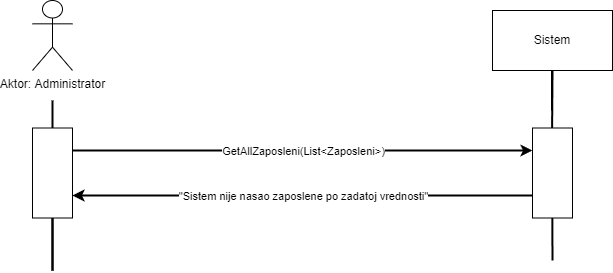
1. Администратор **позива** систем да нађе запослене по задатој вредности. (АПСО)

2. Систем приказује администратору податке о запосленима и поруку: “Систем је нашао запослене по задатој вредности”. (ИА)



*Алтернативна сценарија*

2.1. Уколико систем не може да нађе запослене он приказује администратору поруку: „Систем не може да нађе запосленог по задатој вредности“. (ИА)



Са наведених секвенцних дијаграма уочава се једна системска операција:

1. *Signal* GetAllZaposleni(List<Zaposleni>)

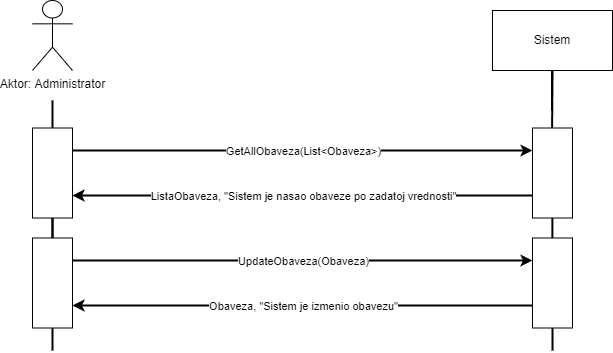
### ДС5: Дијаграм секвенце случаја коришћења – Измена података о обавези

1. Администратор **позива** систем да нађе обавезе по задатој вредности. (АПСО)

2. Систем приказује администратору обавезе и поруку: “Систем је нашао обавезе по задатој вредности”. (ИА)

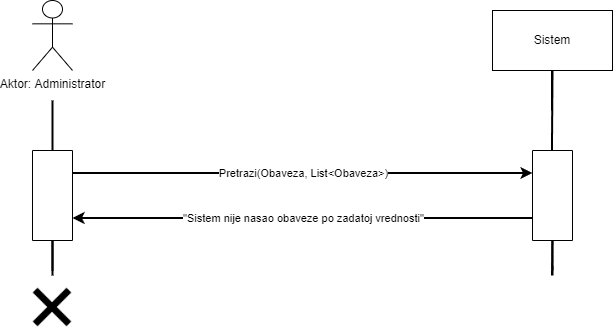
5. Администратор **позива** систем да запамти измењене податке о обавези. (АПСО)

6. Систем **приказује** администратору запамћену обавезу и поруку: “Систем је запамтио обавезу.” (ИА)

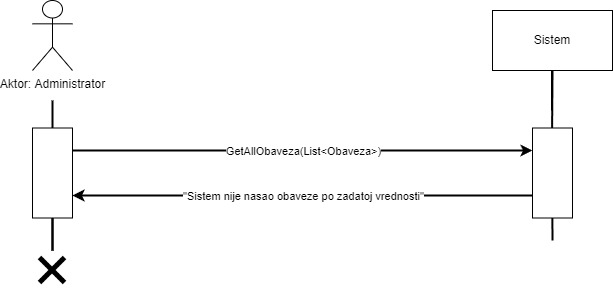


*Алтернативна сценарија*

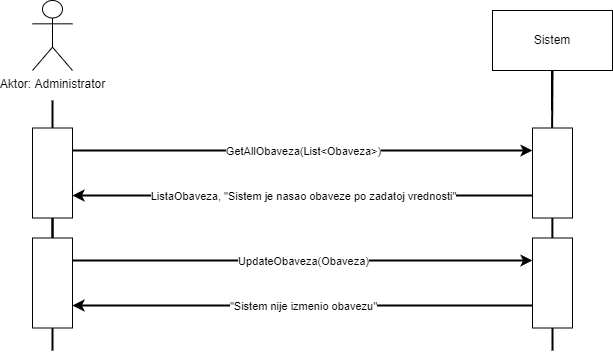
* 1. Уколико систем не може да нађе обавезе он приказује администратору поруку: „Систем не може да нађе обавезе по задатој вредности“. Прекида се извршавање сценарија. (ИА)



4.1. Уколико систем не може да учита обавезу он приказује администратору поруку „Систем не може да учита обавезу“. Прекида се извршавање сценарија. (ИА)



6.1. Уколико систем не може да запамти измењене податке о обавези он приказује администратору поруку „Систем не може да запамти обавезу“. (ИА)



Са наведених секвенцних дијаграма уочавају се 2 системске операције:

1. *Signal* GetAllObaveza(Obaveza, List<Obaveza>)
2. *Signal* UpdateObaveza(Obaveza)

### ДС6: Дијаграм секвенце случаја коришћења – Измена података о запосленима

1. Форма **позива** систем да учита листу општина. (АПСО)

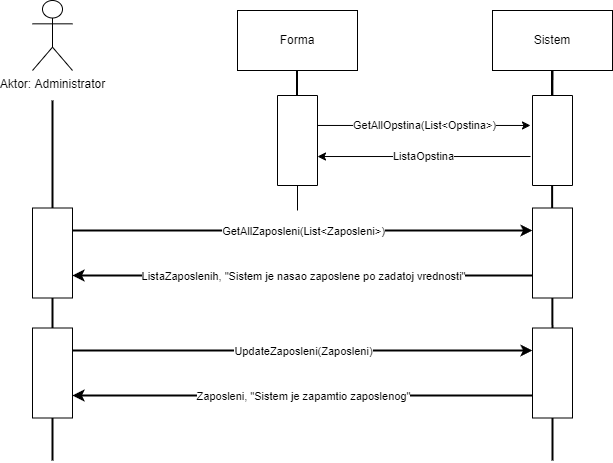
2. Систем **враћа** форми листу општина. (ИА)

3. Администратор **позива** систем да нађе запослене по задатој вредности. (АПСО)

4. Систем приказује администратору запослене и поруку: “Систем је нашао запослене по задатој вредности”. (ИА)

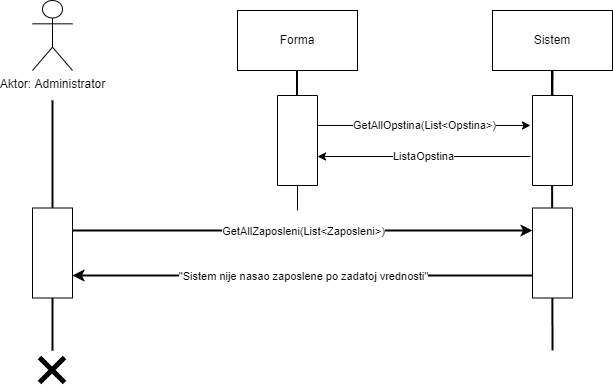
5. Администратор **позива** систем да запамти измењене податке о запосленом. (АПСО)

6. Систем **приказује** администратору запамћеног запосленог и поруку: “Систем је запамтио запосленог.” (ИА)

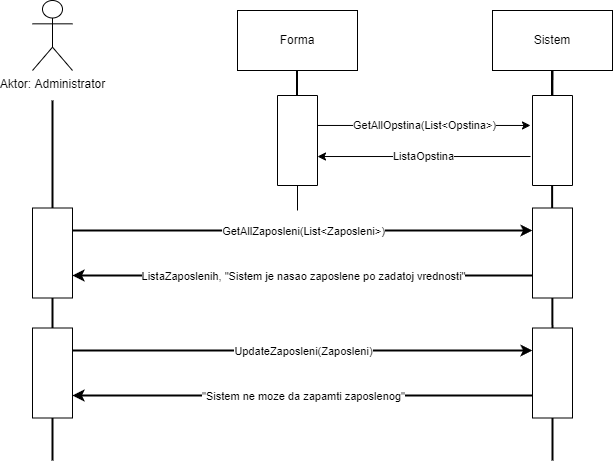


*Алтернативна сценарија*

4.1. Уколико систем не може да нађе запослене он приказује администратору поруку: „Систем не може да нађе запосленог по задатој вредности“. Прекида се извршавање сценарија. (ИА)



8.1. Уколико систем не може да запамти податке о запосленом он приказује администратору поруку „Систем не може да запамти запосленог“. (ИА)



Са наведених секвенцних дијаграма уочавају се четири системске операције:

1. *Signal* GetAllOpstina (List<Opstina>)
2. *Signal* GetAllZaposleni(List<Zaposleni>)
3. *Signal* UpdateZaposleni(Zaposleni)

### ДС7: Дијаграм секвенце случаја коришћења – Креирање захтева

1. Форма **позива** систем да учита листу запослених. (АПСО)

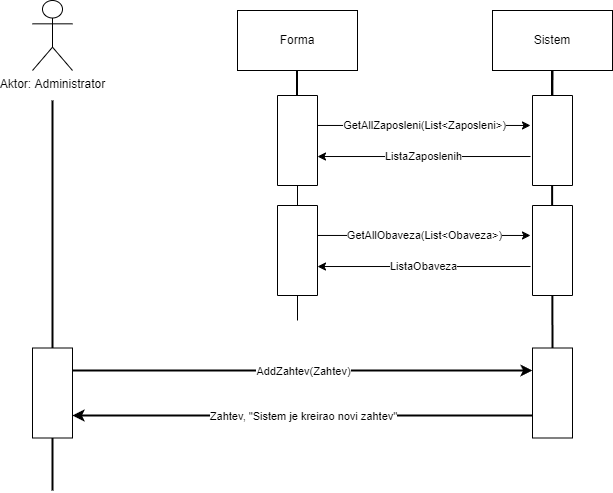
2. Систем **враћа** форми листу запослених. (ИА)

3. Форма **позива** систем да учита листу обавеза. (АПСО)

4. Систем **враћа** форми листу услуга. (ИА)

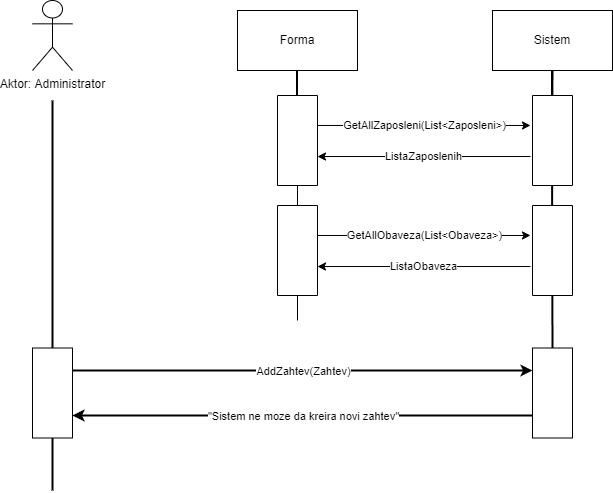
5. Администратор **позива** систем да креира нови захтев. (АПСО)

6. Систем **приказује** администратору нови захтев и поруку: „Систем је креирао нови захтев“. (ИА)



*Алтернативна сценарија*

6.1. Уколико систем не може да креира захтев он приказује администратору поруку: „Систем не може да креира нови захтев“. (ИА)

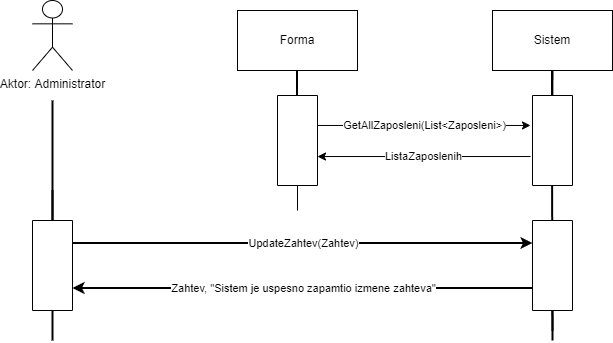


Са нaведених секвенцних дијаграма уочавају се три системске операције:

1. *Signal* GetAllZaposleni(List<Zaposleni>)
2. *Signal* GetAllObaveza(List<Obaveza>)
3. *Signal* AddZahtev(Zahtev)

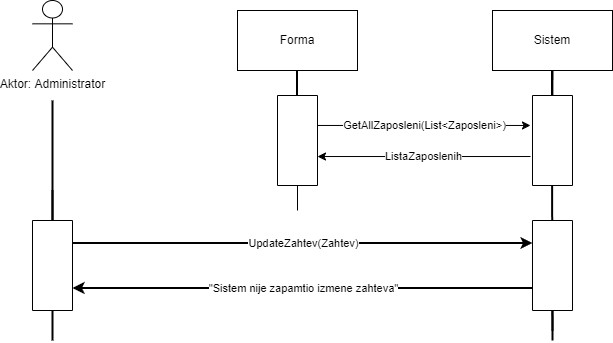
### ДС8: Дијаграм секвенце случаја коришћења – Измена захтева

1. Форма **позива** систем да учита листу запослених. (АПСО)
2. Систем **враћа** форми листу запослених. (ИА)
3. Администратор **позива** систем да запамти измене захтева. (АПСО)
4. Систем **приказује** администратору запамћене измене захтева и поруку: “Систем je запамтио измене захтева”.(ИА)



*Алтернативна сценарија*

2.1. Уколико систем не може да пронађе захтеве по задатој вредности приказује администратору поруку „Систем не може да нађе захтевe по задатој вредности“. Прекида се извршавање сценарија. (ИА)

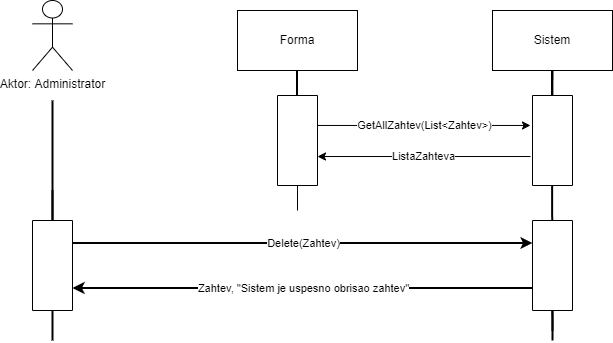


Са наведених секвенцних дијаграма уочавају се три системске операције:

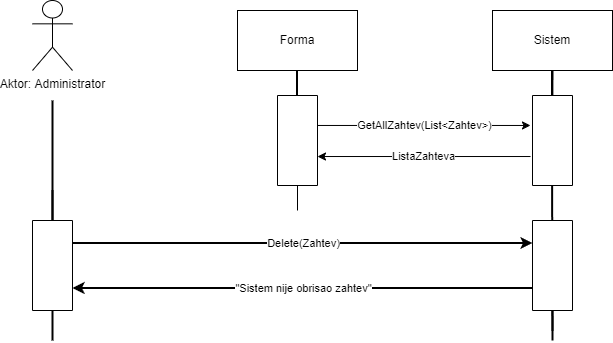
1. Signal GetAllZaposleni(List<Zaposleni>)
2. Signal UpdateZahtev(Zahtev)

### ДС9: Дијаграм секвенце случаја коришћења – Брисање захтева

1. Форма **позива** систем да учита листу запослених. (АПСО)
2. Систем **враћа** форми листу запослених. (ИА)
3. Администратор **позива** систем да обрише одабрани захтев. (АПСО)
4. Систем **приказује** администратору поруку: “Систем je успешно обрисао захтев”.(ИА)



6.1. Уколико систем не може да обрише захтев он приказује администратору поруку „Систем не може да обрише захтев“. (ИА)



Са наведених секвенцних дијаграма уочавају се четири системске операције:

1. *Signal* GetAllZahtev(List<Zahtev>)
2. Signal Delete(Zahtev)

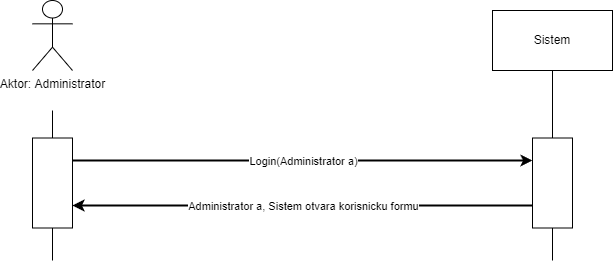
### ДС10: Дијаграм секвенце случаја коришћења – Пријављивање администратора на систем

**Основни сценарио СК**

1. Администратор **позива** систем да пронађе администратора са задатим подацима.(АПСО)

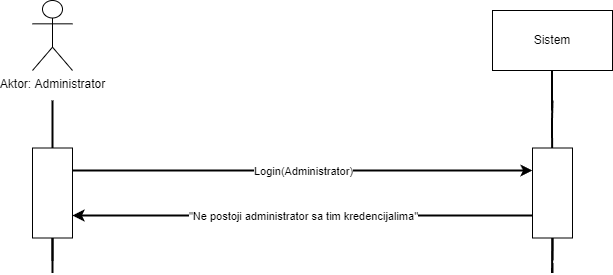
2. Систем **проналази** актора. **(**ИА)

3. Систем **отвара** администраторску форму и **приказује** актору пријављеног администратора. (ИА)



**Алтернативна сценарија:**

2.1 Уколико систем није **пронашао** актора, **приказује** актору поруку: „Не постоји администратор са тим крединцијалима.”. (ИА)



Са наведених секвенцних дијаграма уочава се једна системска операција:

1. *Signal* Login(Zaposleni)

На основу анализе сценарија добијено је 18 системских операција:

1. *Signal* SOGetAllOpstina (List<Opstina>)
2. Signal SOGetAllZaposleni (List<Zaposleni>)
3. Signal SOGetAllZahtev (List<Zahtev>)
4. Signal SOGetAllObaveza (List<Obaveza>)
5. *Signal* SOAddZaposleni (Zaposleni)
6. *Signal* SOUpdateZaposleni (Zaposleni)
7. Signal SOAddObaveza (Obaveza)
8. *Signal* SOUpdateObaveza (Obaveza)
9. Signal SOAddZahtev (Zahtev)
10. Signal SOUpdateZahtev (Zahtev)
11. Signal SODeleteZahtev (Zahtev)
12. Signal SOLogin (Administrator)
13. Signal SOAddPromenaObaveze ()
14. Signal SOGetAllStavkaZahteva (List<StavkaZahteva>)

## Понашање софтверског система – Дефисинање уговора о системским операцијама

1. Уговор UG1: SOGetAllOpstina(List<Opstina>): signal;

Веза са СК: СК1, СК6

Предуслови:

Постуслови:

1. Уговор UG2: SOGetAllZaposleni(List<Zaposleni>): signal;

Веза са СК: СК7, СК8

Предуслови: /

Постуслови: /

1. Уговор UG3: SOGetAllZahtev(List<Zahtev>): signal;

Веза са СК: СК9

Предуслови: /

Постуслови: /

1. Уговор UG4: SOGetAllObaveza(List<Obaveza>): signal;

Веза са СК: СК7

Предуслови: /

Постуслови: /

1. Уговор UG5: SOAddZaposleni(Zaposleni): signal;

Веза са СК: СК1

Предуслови: Вредносна и структурна ограничења над објектом Zaposleni морају бити

задовољена.

Постуслови: Креиран је нови запослени.

1. Уговор UG6: SOUpdateZaposleni(Zaposleni): signal;

Веза са СК: СК6

Предуслови: Вредносна и структурна ограничења над објектом Zaposleni морају бити

задовољена.

Постуслови: Запамћен је нови запослени.

1. Уговор UG7: SOAddObaveza(Obaveza): signal;

Веза са СК: СК2

Предуслови: Вредносна и структурна ограничења над објектом Obaveza морају бити

задовољена.

Постуслови: Креирана је нова обавеза.

1. Уговор UG8: SOUpdateObaveza(Obaveza): signal;

Веза са СК: СК5

Предуслови: Вредносна и структурна ограничења над објектом Obaveza морају бити

задовољена.

Постуслови: Подаци о обавези су запамћени.

1. Уговор UG9: SOAddZahtev(Zahtev): signal;

Веза са СК: СК7

Предуслови: Вредносна и структурна ограничења над објектом Zahtev морају бити

задовољена.

Постуслови: Креиран је нови захтев.

1. Уговор UG10: SOUpdateZahtev(Zahtev): signal;

Веза са СК: СК8

Предуслови: Вредносна и структурна ограничења над објектом Obaveza морају бити

задовољена.

Постуслови: Подаци о обавези су запамћени.

1. Уговор UG11: SODeleteZahtev(Zahtev): signal;

Веза са СК: СК9

Предуслови: Вредносна и структурна ограничења над објектом Zahtev су задовољена

Постуслови: Обрисан је захтев.

1. Уговор UG12: SOLogin(Administrator): signal;

Веза са СК: СК10

Предуслови: /

Постуслови: Администратор је улогован.

1. Уговор UG13: SOAddPromenaObaveze(Zahtev): signal;

Веза са СК: СК5

Предуслови: /

Постуслови: /

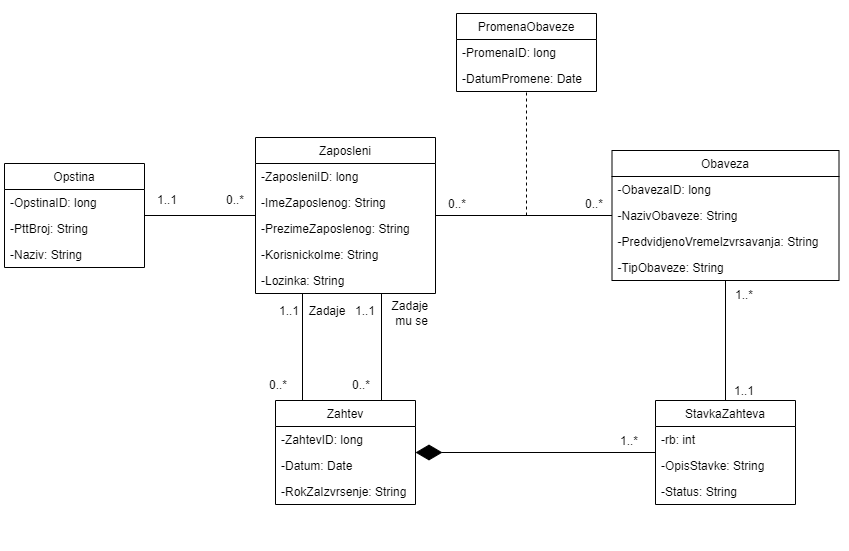
1. Уговор UG14: SOGetAllStavkaZahteva(Zahtev): signal;

Веза са СК: СК8

Предуслови: /

Постуслови: /

## Структура софтверског система – Концептуални (доменски) модел



## Структура софтверског система – Релациони модел

Opstina (​OpstinaID​, PttBroj, Naziv)

Zaposleni (ZaposleniID, Ime, Prezime, KorisnickoIme, Lozinka, *OpstinaID*)

Obaveza (ObavezaID, Naziv, PredvidjenoVremeIzvrsavanja, TipObaveze)

PromenaObaveze(PromenaID, *ObavezaID, ZaposleniID*, DatumPromene)

Zahtev (ZahtevID, Datum, RokZaIzvrsenje, *ZaposleniID*)

StavkaZahteva (rb*, ZahtevID*, OpisStavke, Status)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela Opstina | | Prosto vrednosno ogranicenje | | Slozeno vrednosno ogranicenje | | Strukturno ogranicenje |
| Atributi | Ime | Tip atributa | Vrednost atributa | Medjuzavisnost atributa jedne tabele | Medjuzavisnost atributa vise tabela | INSERT /  UPDATE CASCADES Zaposleni  DELETE RESTRICTED Zaposleni |
| IDOpstine | Integer | Not null and >0 |  |  |
| Naziv | String | Not null |  |  |
| PttBroj | String | Not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela Zaposleni | | Prosto vrednosno ogranicenje | | Slozeno vrednosno ogranicenje | | Strukturno ogranicenje |
| Atributi | Ime | Tip atributa | Vrednost atributa | Medjuzavisnost atributa jedne tabele | Medjuzavisnost atributa vise tabela | INSERT RESTRICTED Opstina  UPDATE RESTRICTED Opstina, CASCADES Zahtev  DELETE RESTRICTED Zahtev |
| IDZaposlenog | Integer | Not null and >0 |  |  |
| Ime | String | Not null |  |  |
| Adresa | String | Not null |  |  |
| Tip | Integer | Not null |  |  |
| IDOpstine | Integer | Not null and >0 |  |  |

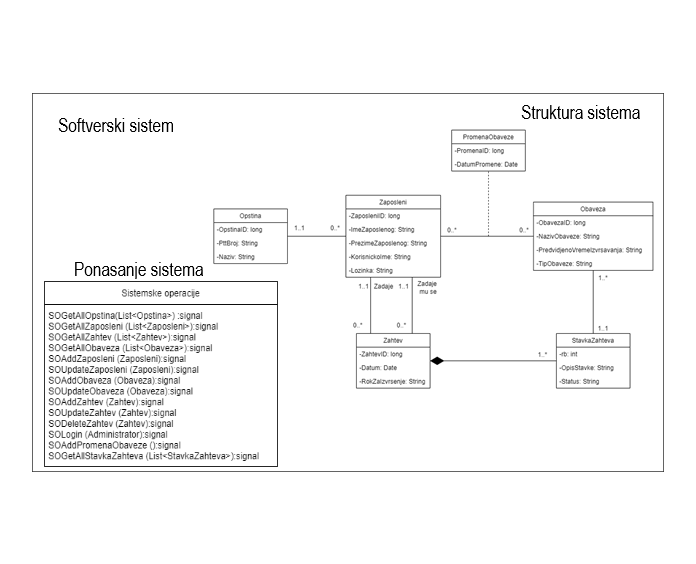
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela Obaveza | | Prosto vrednosno ogranicenje | | Slozeno vrednosno ogranicenje | | Strukturno ogranicenje |
| Atributi | Ime | Tip atributa | Vrednost atributa | Medjuzavisnost atributa jedne tabele | Medjuzavisnost atributa vise tabela | INSERT /  UPDATE CASCADES  StavkaZahteva  DELETE  StavkaZahteva, CASCADES  PromenaObaveze |
| IDObaveze | Integer | Not null and >0 |  |  |
| Naziv | String | Not null |  |  |
| IDKategorije | Integer | Not null and >0 |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela Zahtev | | Prosto vrednosno ogranicenje | | Slozeno vrednosno ogranicenje | | Strukturno ogranicenje |
| Atributi | Ime | Tip atributa | Vrednost atributa | Medjuzavisnost atributa jedne tabele | Medjuzavisnost atributa vise tabela | INSERT RESTRICTED Zaposleni  UPDATE CASCADES StavkaZahteva  DELETE CASCADES StavkaZahteva |
| IDZahteva | Integer | Not null and >0 |  |  |
| Datum | Date | Not null |  |  |
| Status | Integer | Not null |  |  |
| IDZaposlenog | Integer | Not null and > 0 |  |  |
| IDObaveze | Integer | Not null and >0 |  |  |

# 

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tabela StavkaZahteva | | Prosto vrednosno ogranicenje | | Slozeno vrednosno ogranicenje | | Strukturno ogranicenje |
| Atributi | Ime | Tip atributa | Vrednost atributa | Medjuzavisnost atributa jedne tabele | Medjuzavisnost atributa vise tabela | INSERT RESTRICTED Zahtev, obaveza  UPDATE RESTRICTED Zahtev, obaveza  DELETE / |
| rb | Integer | Not null and >0 |  |  |
| OpisStavke | Integer | Not null and >0 |  |  |
| Status | String | Not null |  |  |

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



# Пројектовање

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

Архитектура софтверског система

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

1. Кориснички интерфејс
2. Апликациона логика
3. Складиште података

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



## Пројектовање корисничког интерфејса

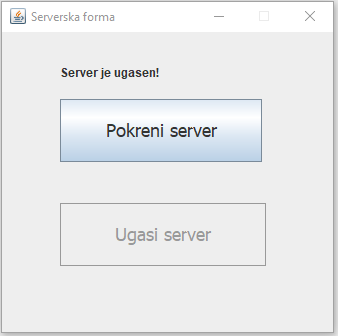
Кориснички интерфејс представља реализацију улаза и/или излаза софтверског система и његову структуру чине екранска форма и контролер корисничког интерфејса.



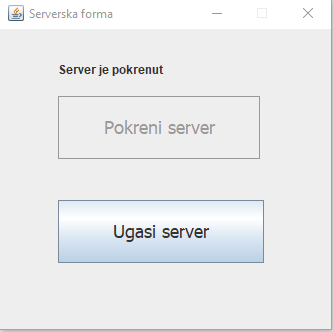
### Пројектовање екранских форми

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

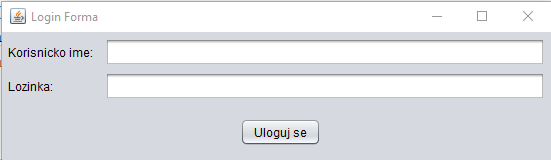
На серверској страни програма пројектована је корисничка форма која изгледа пре активације овако:



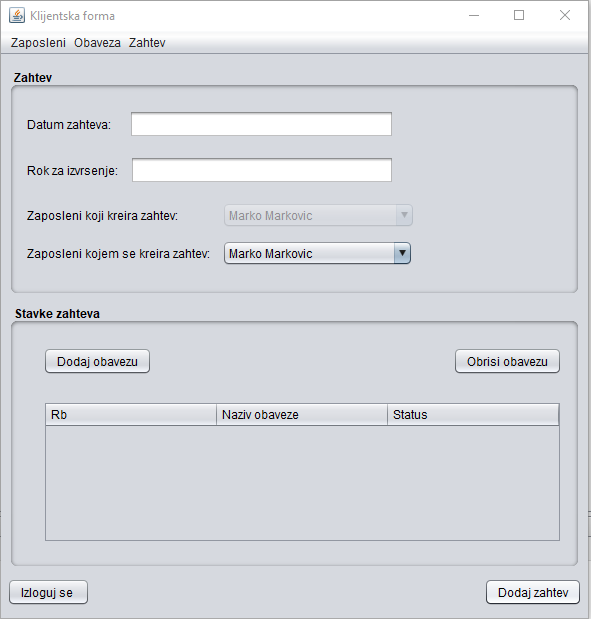
Након активације, серверска форма изгледа овако:



На клијентској страни прво је потребно улоговати се како би се доспело у могућност коришћења апликације. Форма за логовање изгледа овако:



Након логовања, систем приказује клијенту главну екранску форму из које се може доћи до осталих екранских форми.



### СК1: Случај коришћења – Креирање налога запосленима

#### Назив СК

Креирање новог налога

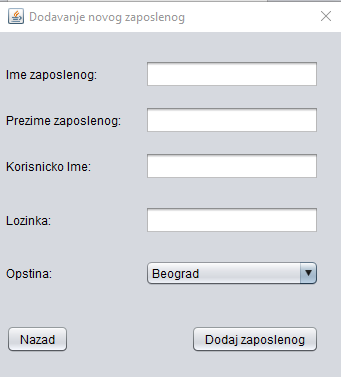
#### Актори СК

Администратор

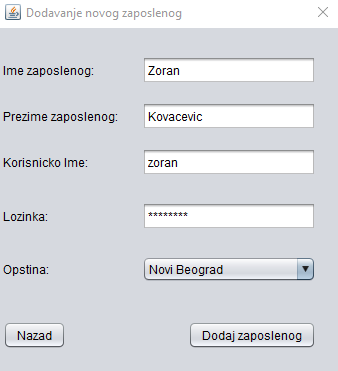
#### Учесници СК

Администратор и систем (програм)

**Предуслов**: Систем је укључен и администратор je улогован под својом шифром. Систем приказује форму за рад са налогом. Учитана је листа општина где се налазе теретане.

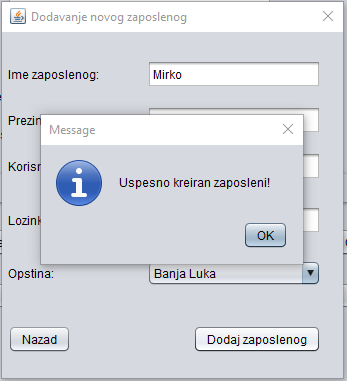


#### Основни сценарио СК

1. Администратор **уноси** податке у нови налог. (АПУСО)
2. Администратор **контролише** да ли је коректно унео податке у нови налог. (АНСО)
3. Администратор **позива** систем да креира нови налог. (АПСО)

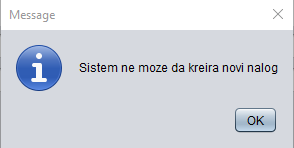
Опис акције: Администратор кликом на дугме „Dodaj zaposlenog“ позива системску операцију **AddZaposleni(Zaposleni)**

1. Систем **креира** нови налог. (СО)
2. Систем **приказује** администратору нови налог и поруку: “Систем је креирао нови налог“. (ИА)



Алтернативна сценарија

5.1 Уколико систем не може да креира нови налог он приказује администратору поруку: “Систем не може да креира нови налог”. (ИА)



### СК2: Случај коришћења – Додавање нове обавезе

#### Назив СК

Креирање нове обавезе

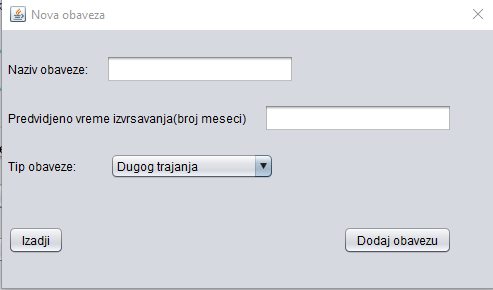
#### Актори СК

Администратор

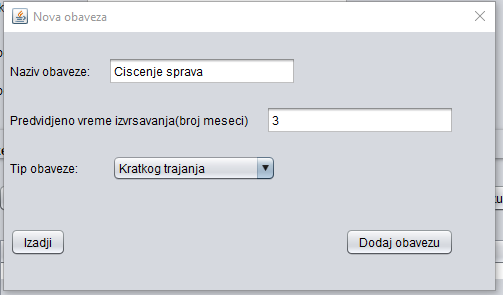
#### Учесници СК

Администратор и систем (програм)

**Предуслов**: Систем је укључен и администратор je улогован под својом шифром. Систем приказује форму за рад са обавезом. Учитана је листа типова обавеза.

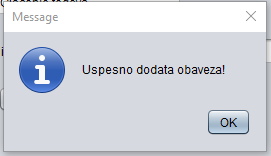


#### Основни сценарио СК

1. Администратор **уноси** податке у нову обавезу. (АПУСО)
2. Администратор **контролише** да ли је коректно унео податке у нову обавезу. (АНСО)
3. Администратор **позива** систем да креира нову обавезу. (АПСО)

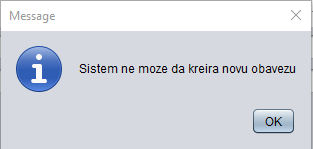
Опис акције: Администратор кликом на дугме „Dodaj obavezu“ позива системску операцију **KreirajUslugu(Usluga)**

1. Систем **креира** нову обавезу. (СО)
2. Систем **приказује** администратору нову обавезу и поруку: “Систем је креирао нову обавезу“. (ИА)



Алтернативна сценарија

5.1 Уколико систем не може да креира нову обавезу он приказује администратору поруку: “Систем не може да креира нову обавезу”. (ИА)



### СК3: Случај коришћења – Претраживање обавеза

#### Назив СК

Претраживање обавеза

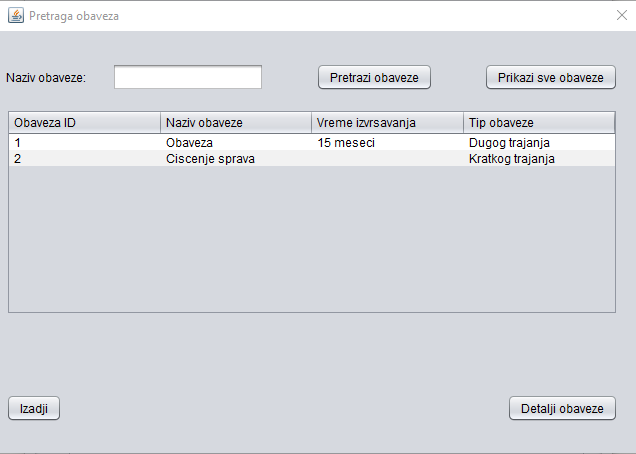
#### Актори СК

Администратор

#### Учесници СК

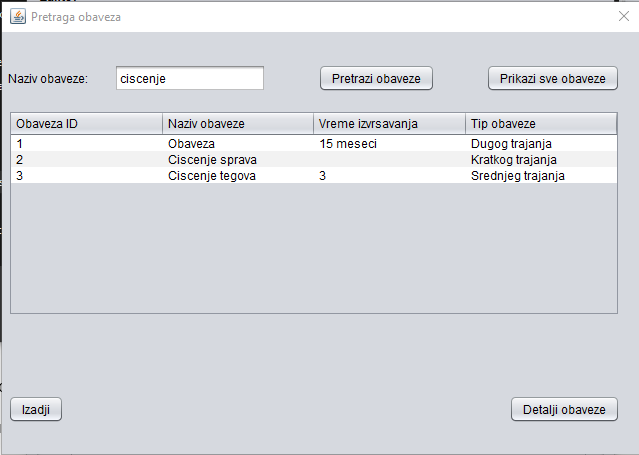
Администратор и систем (програм)

**Предуслов**: Систем је укључен и администратор je улогован под својом шифром. Систем приказује форму за рад са обавезама.



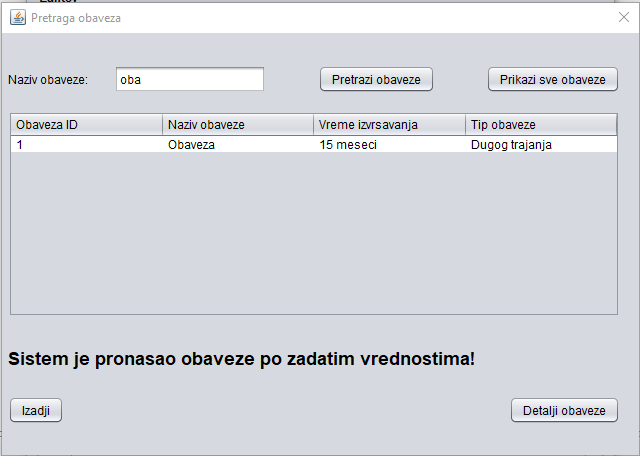
#### Основни сценарио СК

1. Администратор **уноси** вредност по којој претражује обавезе. (АПУСО)



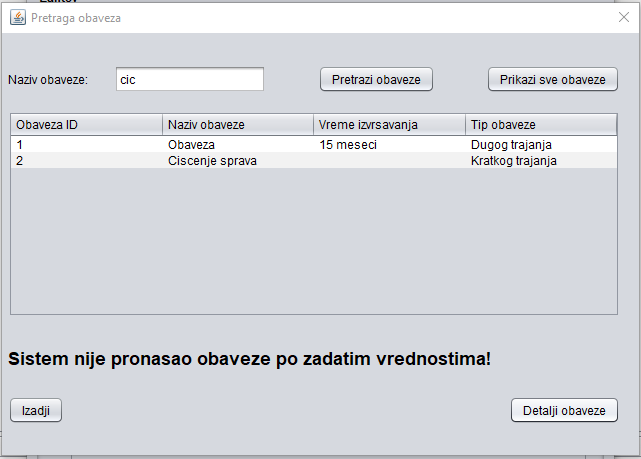
1. Администратор **позива** систем да нађе обавезе по задатој вредности. (АПСО)

Опис акције: Администратор кликом на дугме „Pretrazi obaveze“ позива системску операцију **GetAllObaveza(List<Obaveza>)**

1. Систем **тражи** обавезе по задатој вредности. (СО)
2. Систем приказује администратору податке о обавезама и поруку: “Систем је нашао обавезе по задатој вредности”. (ИА)

Алтернативна сценарија

* 1. Уколико систем не може да нађе обавезе он приказује администратору поруку: “Систем не може да нађе обавезе по задатој вредности”. (ИА)



### СК4: Случај коришћења – Претраживање запослених

#### Назив СК

Претраживање запослених

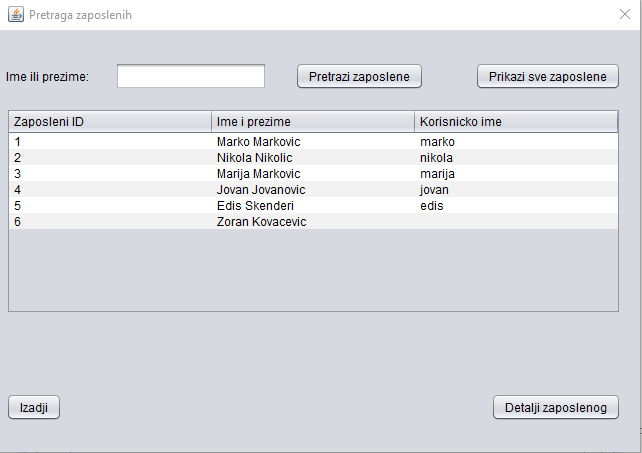
#### Актори СК

Администратор

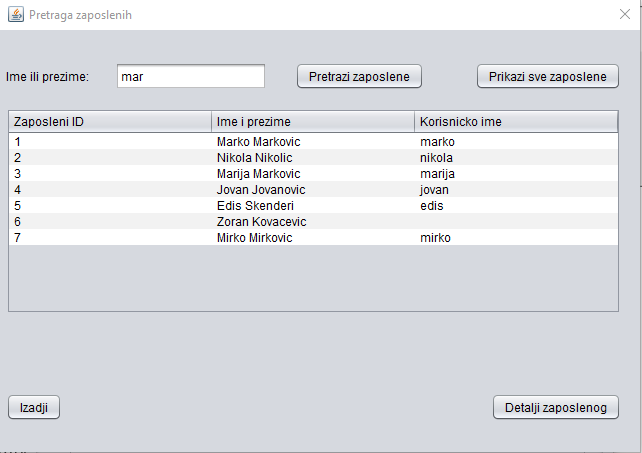
#### Учесници СК

Администратор и систем (програм)

**Предуслов**: Систем је укључен и администратор je улогован под својом шифром. Систем приказује форму за рад са запосленима.

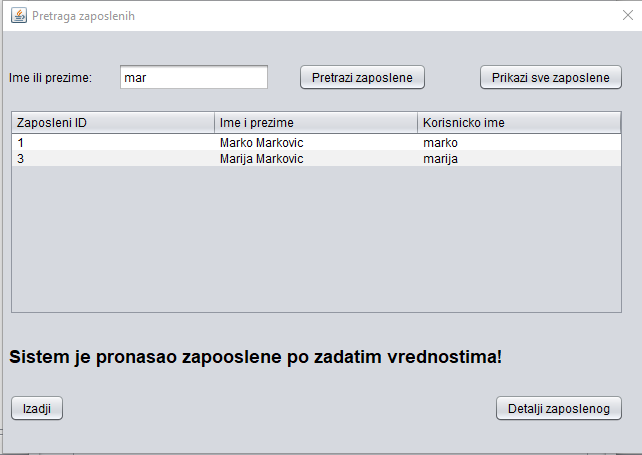


#### Основни сценарио СК

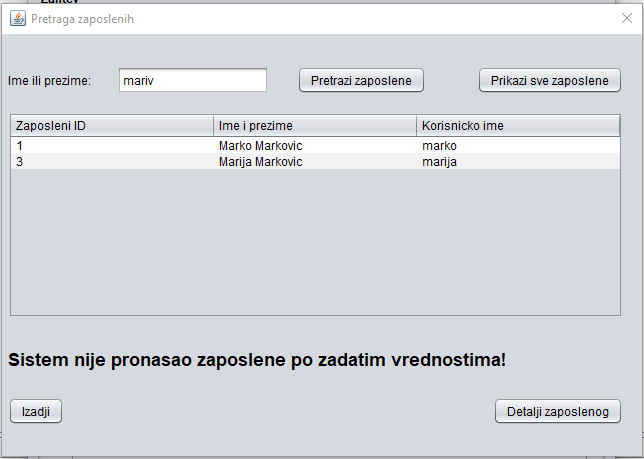
1. Администратор **уноси** вредност по којој претражује запослене. (АПУСО)
2. Администратор **позива** систем да нађе запослене по задатој вредности. (АПСО)

Опис акције: Администратор кликом на дугме „Pretrazi zaposlene“ позива системску операцију **GetAllZaposleni(List<Zaposleni>)**

1. Систем **тражи** запослене по задатој вредности. (СО)
2. Систем приказује администратору податке о запосленима и поруку: “Систем је нашао запослене по задатој вредности”. (ИА)



Алтернативна сценарија

* 1. Уколико систем не може да нађе запослене он приказује администратору поруку: “Систем не може да нађе запосленог по задатој вредности”. (ИА)

### СК5: Случај коришћења – Промена података о обавези

#### Назив СК

Промена обавезе

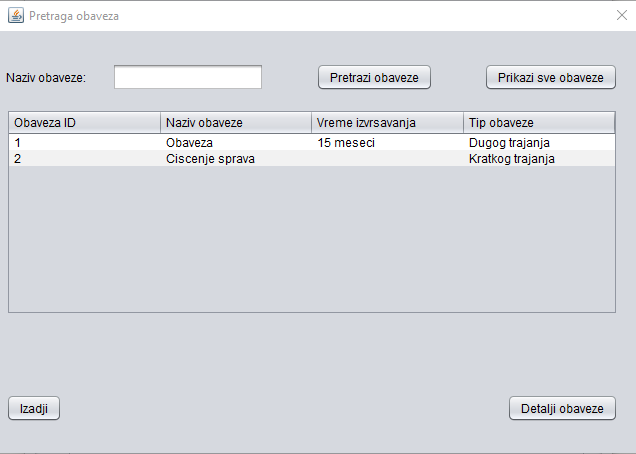
#### Актори СК

Администратор

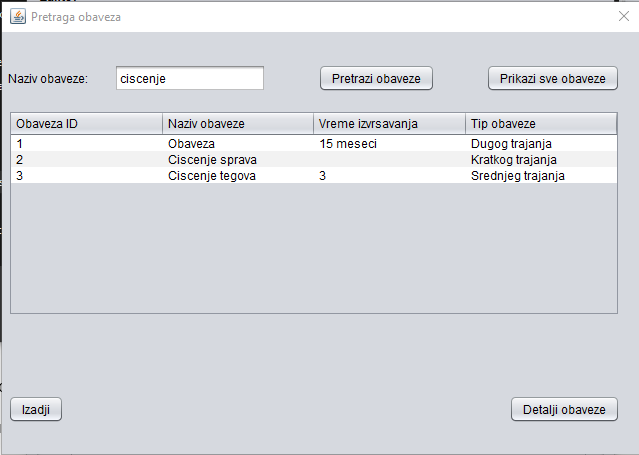
#### Учесници СК

Администратор и систем (програм)

**Предуслов**: Систем је укључен и администратор je улогован под својом шифром. Систем приказује форму за рад са обавезама.

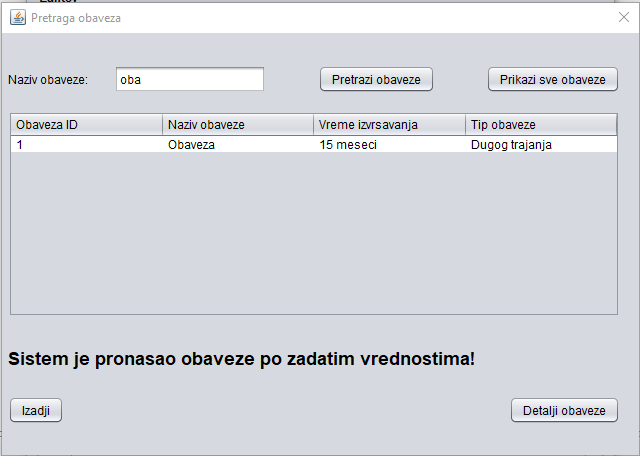


#### Основни сценарио СК

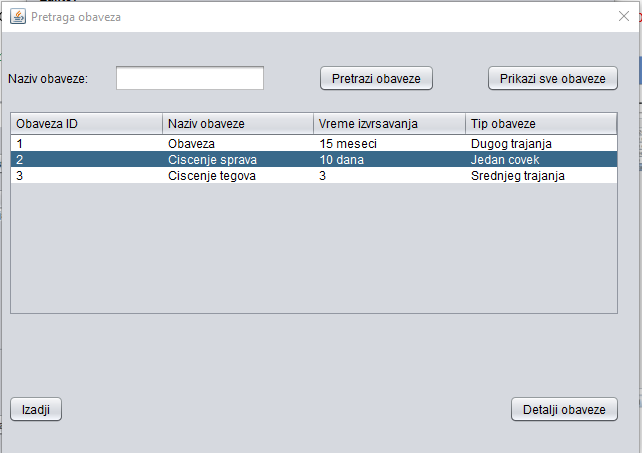
1. Администратор **уноси** вредност по којој претражује обавезе. (АПУСО)
2. Администратор **позива** систем да нађе обавезе по задатој вредности. (АПСО)

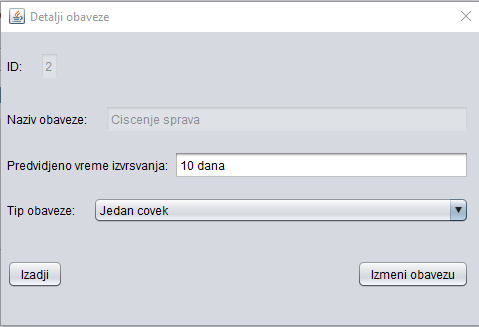
Опис акције: Администратор кликом на дугме „Pretrazi obaveze“ позива системску операцију **GetAllObaveza(List<Obaveza>)**

1. Систем **тражи** обавезе по задатој вредности. (СО)
2. Систем приказује администратору обавезе и поруку: “Систем је нашао обавезе по задатој вредности”. (ИА)



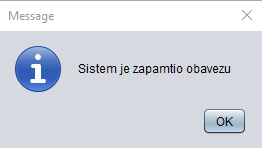
1. Администратор **бира** обавезу. (АПУСО)



1. Администратор **уноси (мења)** податке о обавези. (АПУСО)
2. Администратор **контролише** да ли је коректно унео податке о обавези. (АНСО)
3. Администратор **позива** систем да запамти податке о обавези. (АПСО)

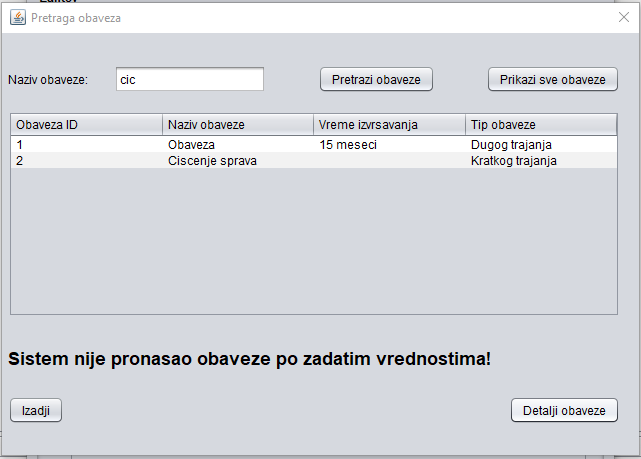
Опис акције: Администратор кликом на дугме „Izmeni obavezu“ позива системску операцију **Update(Obaveza)**

1. Систем **памти** податке о обавези. (СО)
2. Систем **приказује** администратору запамћену обавезу и поруку: “Систем је запамтио обавезу.” (ИА)

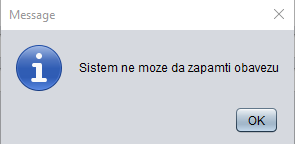


Алтернативна сценарија

* 1. Уколико систем не може да пронађе обавезе по задатој вредности он приказује администратору поруку: “Систем не може да нађе обавезе по задатој вредности”. (ИА)



10.1 Уколико систем не може да запамти податке о обавези он приказује администратору поруку “Систем не може да запамти обавезу”. (ИА)



### СК6: Случај коришћења – Промена података о запосленом

#### Назив СК

Промена запосленог

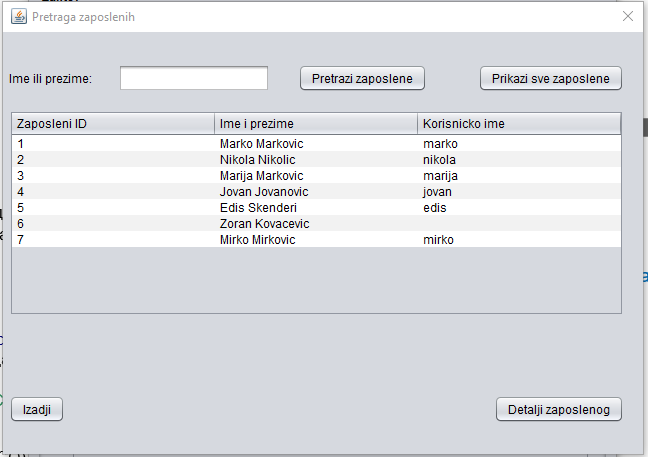
#### Актори СК

Администратор

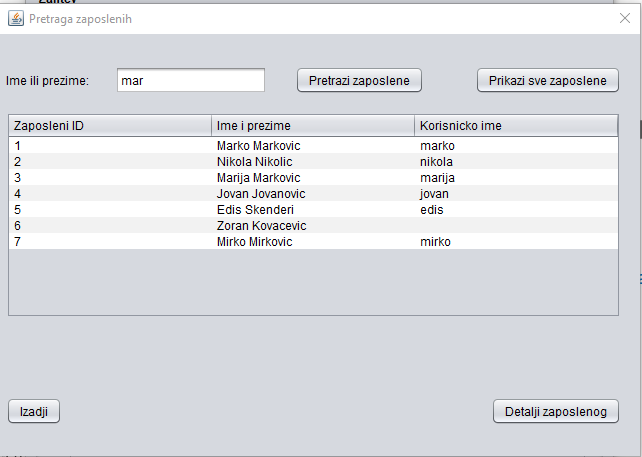
#### Учесници СК

Администратор и систем (програм)

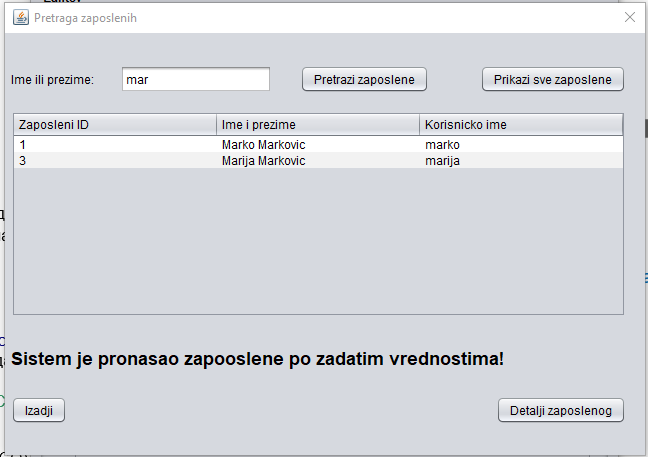
**Предуслов**: Систем је укључен и администратор je улогован под својом шифром. Систем приказује форму за рад са запосленим. Учитана је листа општина.

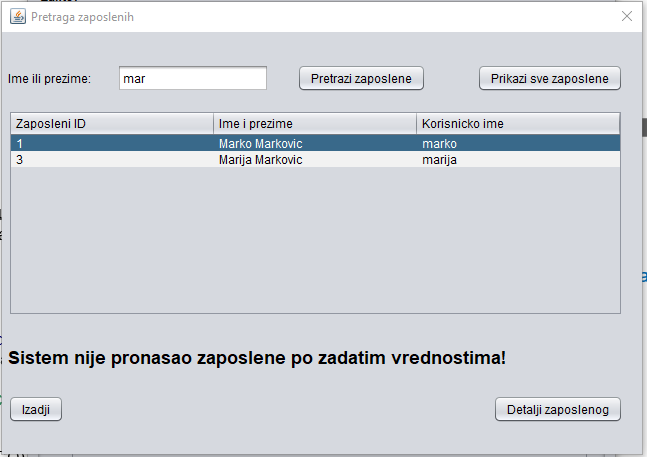


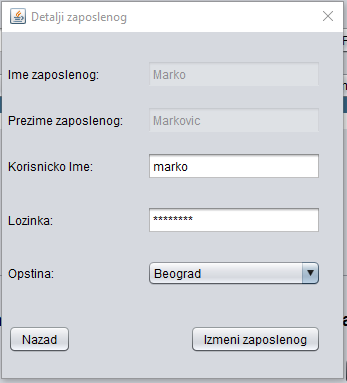
#### Основни сценарио СК

1. Администратор **уноси** вредност по којој претражује запослене. (АПУСО)
2. Администратор **позива** систем да нађе запослене по задатој вредности. (АПСО)

Опис акције: Администратор кликом на дугме „Pretrazi zaposlene“ позива системску операцију **GetAllZaposleni(List<Zaposleni>)**

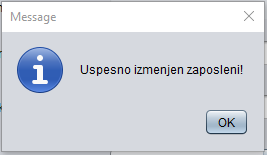
1. Систем **тражи** запослене по задатој вредности. (СО)
2. Систем приказује администратору запослене и поруку: “Систем је нашао запослене по задатој вредности”. (ИА)
3. Администратор **бира** запосленог. (АПУСО)



1. Администратор **уноси (мења)** податке о запосленом. (АПУСО)
2. Администратор **контролише** да ли је коректно унео податке о запосленом. (АНСО)
3. Администратор **позива** систем да запамти податке о запосленом. (АПСО)

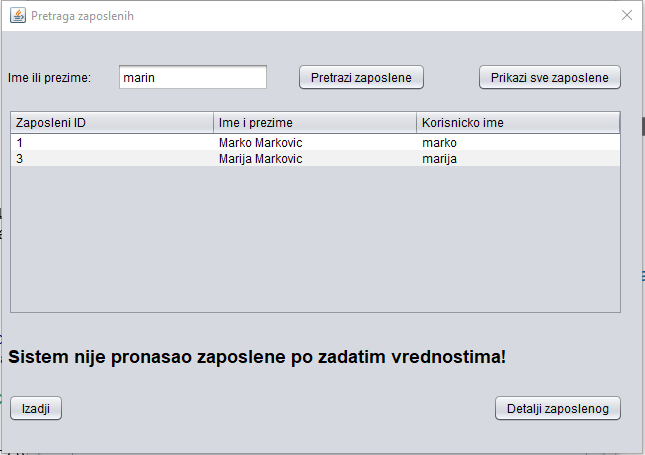
Опис акције: Администратор кликом на дугме „Izmeni zaposlenog“ позива системску операцију **Update(Zaposleni)**

1. Систем **памти** податке о запосленом. (СО)
2. Систем **приказује** администратору запамћеног запосленог и поруку: “Систем је запамтио запосленог.” (ИА)

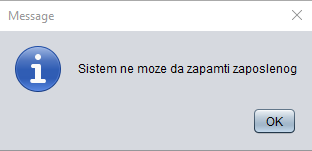


Алтернативна сценарија

* 1. Уколико систем не може да нађе запосленe он приказује администратору поруку: “Систем не може да нађе запосленог по задатој вредности”. Прекида се извршење сценариа. (ИА)



13.1 Уколико систем не може да запамти податке о запосленом он приказује администратору поруку “Систем не може да запамти запосленог”. (ИА)



### СК7: Случај коришћења – Креирање новог захтева (сложен ск)

#### Назив СК

Креирање новог захтева

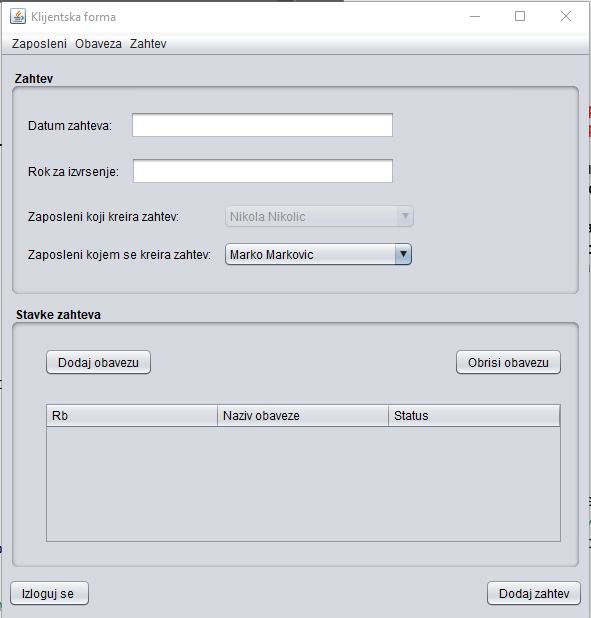
#### Актори СК

Администратор

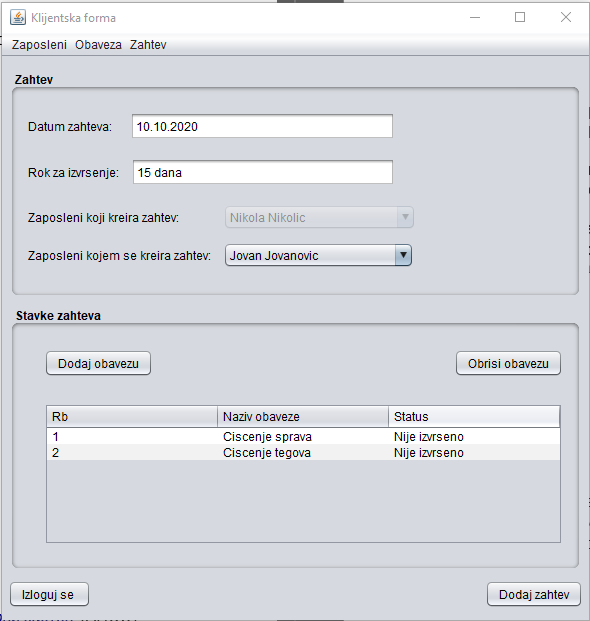
#### Учесници СК

Администратор и систем (програм)

**Предуслов**: Систем је укључен и администратор je улогован под својом шифром. Систем приказује форму за рад са захтевом. Учитане су листе запослених и обавеза.

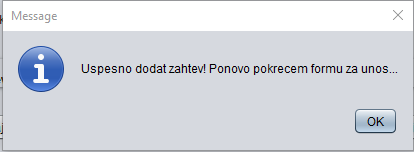


#### Основни сценарио СК

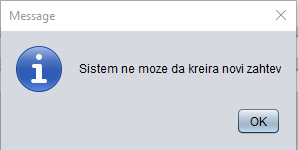
1. Администратор **уноси** податке у нови захтев. (АПУСО)
2. Администратор **контролише** да ли је коректно унео податке у нови захтев. (АНСО)
3. Администратор **позива** систем да креира нови захтев. (АПСО)

Опис акције: Администратор кликом на дугме „Dodaj zahtev“ позива системску операцију **AddZahtev(Zahtev)**

1. Систем **креира** нови захтев. (СО)
2. Систем **приказује** администратору нови захтев и поруку: “Систем је креирао нови захтев“. (ИА)



Алтернативна сценарија

5.1 Уколико систем не може да креира нови захтев он приказује администратору поруку: “Систем не може да креира нови захтев”. (ИА)

### СК8: Случај коришћења – Промена захтева (сложен ск)

#### Назив СК

Промена захтева

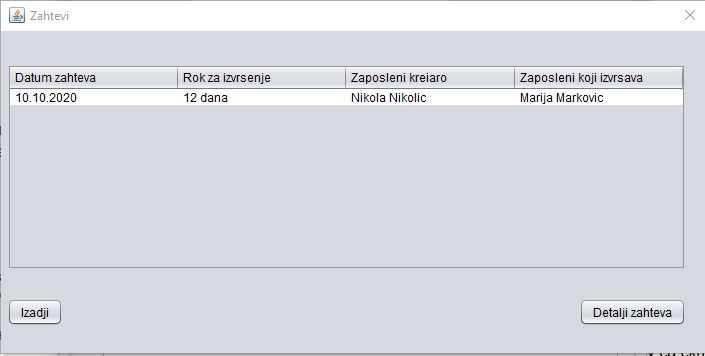
#### Актори СК

Администратор

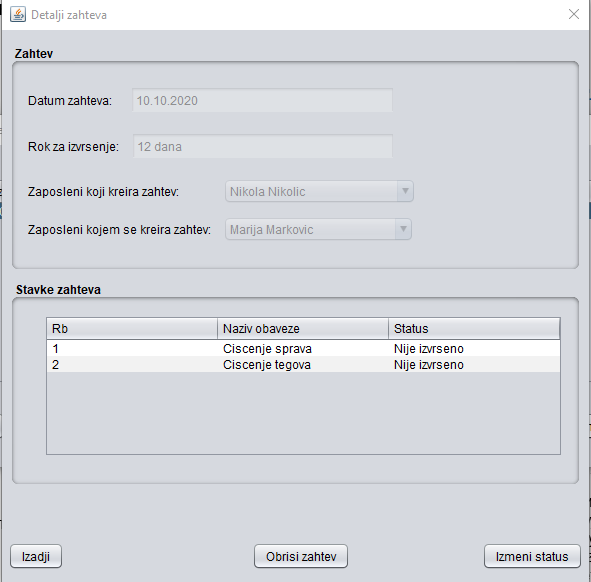
#### Учесници СК

Администратор и систем (програм)

**Предуслов**: Систем је укључен и администратор je улогован под својом шифром. Систем приказује форму за рад са захтевима. Учитана је листа захтева.

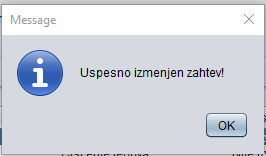


#### Основни сценарио СК

1. Администратор **бира** захтев из листе учитаних захтева. (АПУСО)
2. Систем **отвара** захтев (ИА).
3. Администратор **уноси (мења)** захтев. (АПУСО)
4. Администратор **позива** систем да запамти податке захтева. (АПСО)

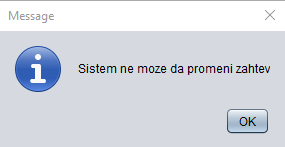
Опис акције: Администратор кликом на дугме „Izmeni zahtev“ позива системску операцију **Update(Zahtev)**

1. Систем **памти** податке о захтевима. (СО)
2. Систем **приказује** администратору измењен захтев и поруку: “Систем је изменио захтев.” (ИА)



Алтернативна сценарија

5.1 Уколико систем не може да промени захтев он приказује администратору поруку: “Систем не може да промени захтев”. (ИА)



### СК9: Случај коришћења – Брисање захтева (сложен ск)

#### Назив СК

Брисање захтева

#### Актори СК

Администратор

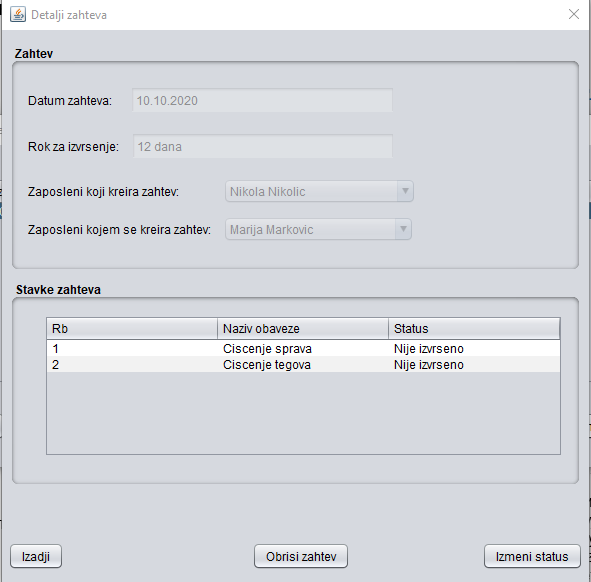
#### Учесници СК

Администратор и систем (програм)

**Предуслов**: Систем је укључен и администратор je улогован под својом шифром. Систем приказује форму за рад са захтевима. Учитана је листа захтева.

#### Основни сценарио СК

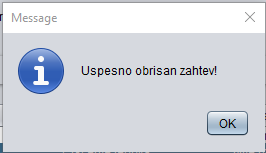
1. Администратор **бира** захтев из листе учитаних захтева. (АПУСО)
2. Систем **отвара** захтев (ИА).



1. Администратор **позива** систем да обрише одабрани захтев. (АПСО)

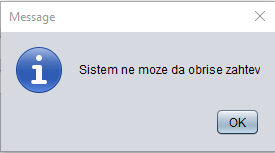
Опис акције: Администратор кликом на дугме „Obrisi zahtev“ позива системску операцију **Delete(Zahtev)**

1. Систем **брише** захтев. (СО)
2. Систем **приказује** администратору поруку: “Систем је обрисао захтев.” (ИА)



Алтернативна сценарија

7.1 Уколико систем не може да обрише захтев он приказује администратору поруку: “Систем не може да обрише захтев”. (ИА)



### СК10: Случај коришћења – Пријављивање администратора на систем

#### Назив СК

Пријављивање администратора на систем

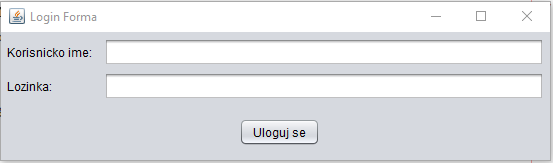
#### Актори СК

Администратор

#### Учесници СК

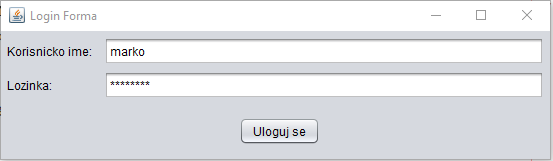
Администратор и систем (програм)

**Предуслов**: Систем је укључен и **приказује** форму за пријављивање администратора.



#### Основни сценарио СК

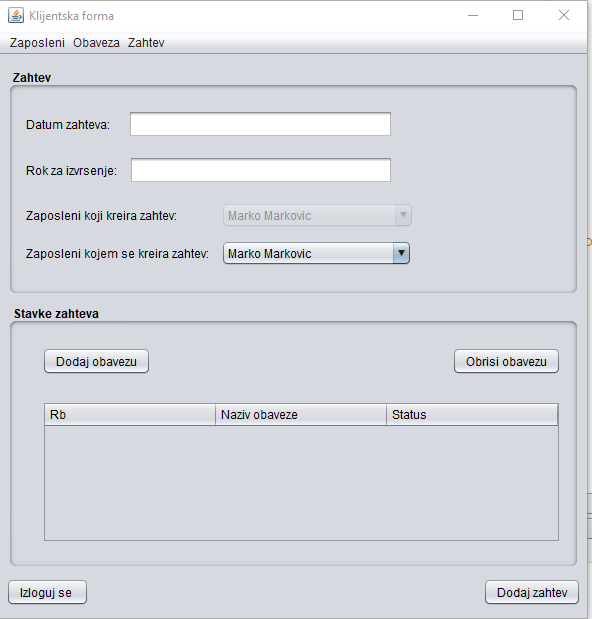
1. Администратор **уноси** податке за пријаву. (АПУСО)



1. Администратор **контролише** да ли је коректно унео податке за пријаву. (АНСО)
2. Администратор **позива** систем да **пронађе** администратора са унетим подацима.(АПСО)

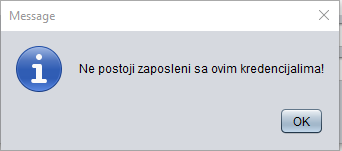
Опис акције: Администратор кликом на дугме „Login“ позива системску операцију **Login(Zaposleni)**

1. Систем **претражује** администраторе. (СО)
2. Систем **проналази** администратора. (ИА)
3. Систем **отвара** администраторску форму. (ИА)



Алтернативна сценарија

5.1 Уколико систем не може да пронађе администратора са задатим креденцијалима он приказује администратору поруку: “Систем не може да пронађе администратора са овим креденцијалима”. (ИА)



### Пројектовање контролера корисничког интерфејса

За сваку екранску форму пројектује се контролер корисничког интерфејса који има следеће одговорности:

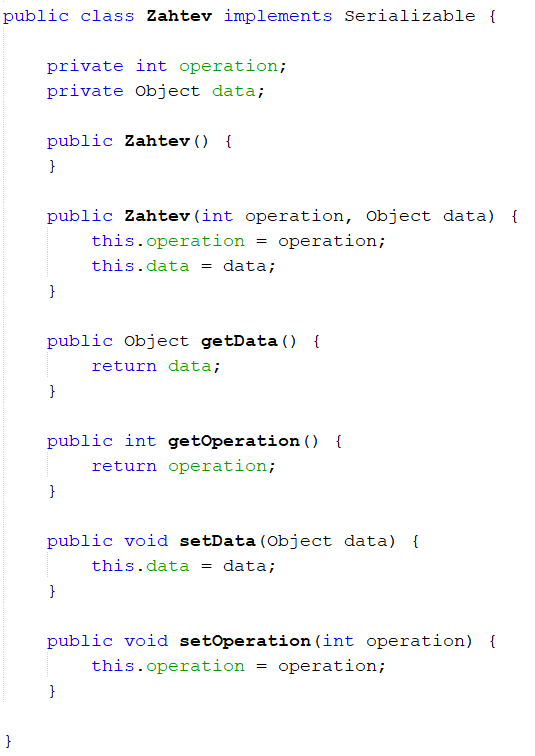
1. Прихвата графичке објекте од екранске форме
2. Конвертује податке који се налазе у графичким објектима у доменске објекте који ће бити прослеђени преко мреже до апликационог сервера
3. Конвертује доменске објекте у графичке објекте и прослеђује их до екранске форме

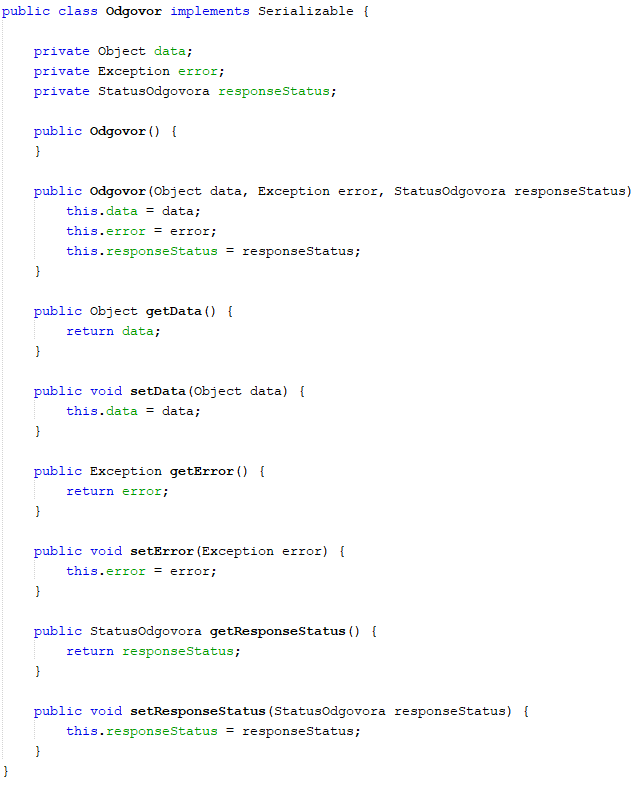
## Пројектовање апликационе логике

Апликациона логика, која се налази на апликационом серверу, обухвата:

1. Комуникацију са клијентом
2. Контролер апликационе логике
3. Пословну логику
4. Комуникацију са складиштем података

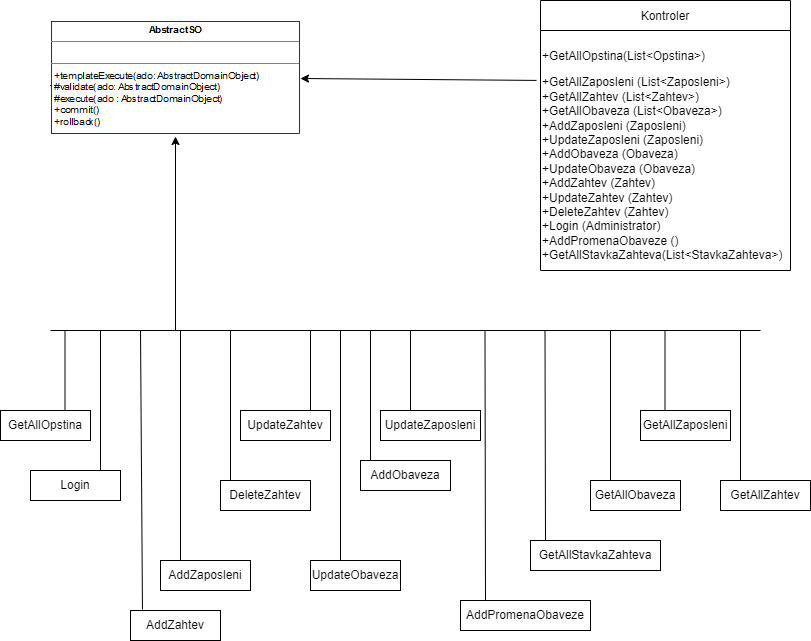
Комуникација са клијентом се имплементира преко сокета. Креира се серверски сокет на одређеном порту који ослушкује мрежу и чека захтев за успостављање конекције. Када клијентски сокет успостави конекцију са серверским сокетом, на серверу се креира нит преко које се одвија двосмерна комуникација између клијента и сервера. Та комуникација се обавља разменом објеката класа Zahtev и Odgovor.





### Контролер аплиакционе логике

Контролер апликцаионе логике добија захтев од нити која је повезана са клијентом да изврши одређену системску операцију, и тај захтев он прослеђује до класа које су одговорне за извршење системских операција. Након што се системска операција изврши, контролер добија резултат извршења те операције и прослеђује га до нити клијента.



### Пословна логика

За сваку системску операцију треба направити концептуална решења која су директно повезана са логиком проблема.

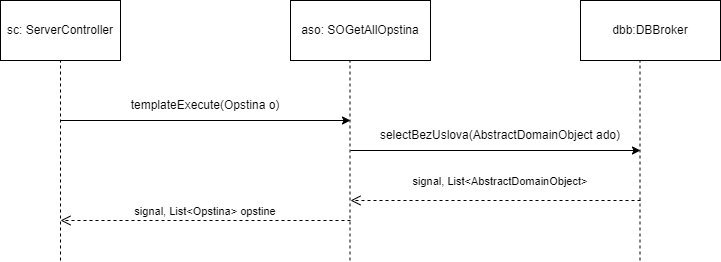
За сваки уговор пројектује се концептуално решење.

1. Уговор UG1: SOGetAllOpstina (List<Opstina>): signal;

Веза са СК: СК1, СК6

Предуслови:/

Постуслови:/

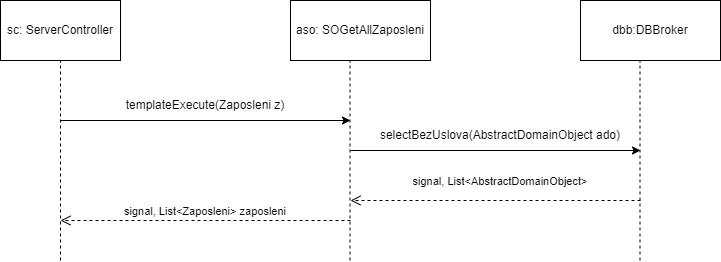


1. Уговор UG2: SOGetAllZaposleni (List<Zaposleni>): signal;

Веза са СК: СК7, СК8

Предуслови: /

Постуслови: /

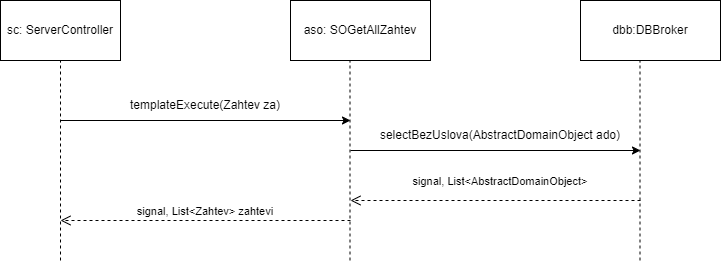


1. Уговор UG3: SOGetAllZahtev (List<Zahtev>): signal;

Веза са СК: СК9

Предуслови: /

Постуслови: /

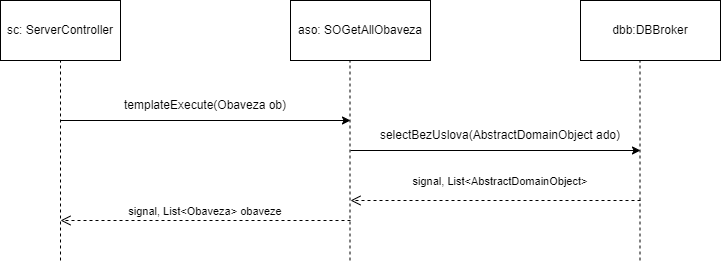


1. Уговор UG4: SOGetAllObaveza (List<Obaveza>): signal;

Веза са СК: СК7

Предуслови: /

Постуслови: /



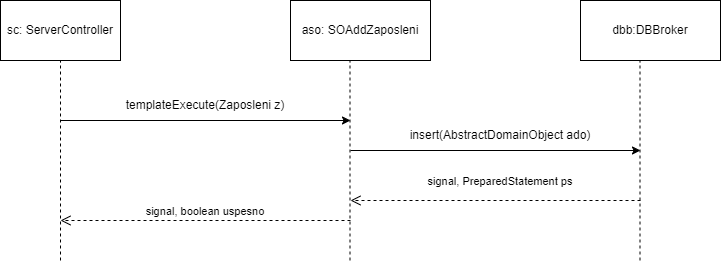
1. Уговор UG5: SOAddZaposleni (Zaposleni): signal;

Веза са СК: СК1

Предуслови: Вредносна и структурна ограничења над објектом Zaposleni морају бити

задовољена.

Постуслови: Креиран је нови запослени.



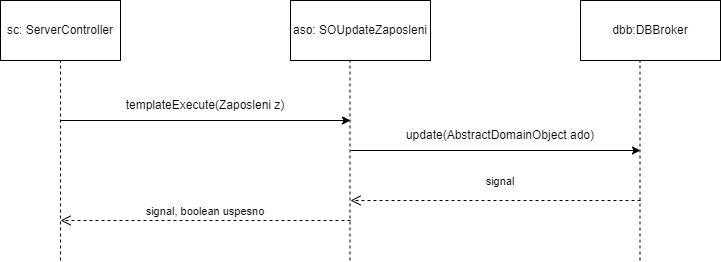
1. Уговор UG6: SOUpdateZaposleni (Zaposleni): signal;

Веза са СК: СК6

Предуслови: Вредносна и структурна ограничења над објектом Zaposleni морају бити

задовољена.

Постуслови: Запамћен је нови запослени.



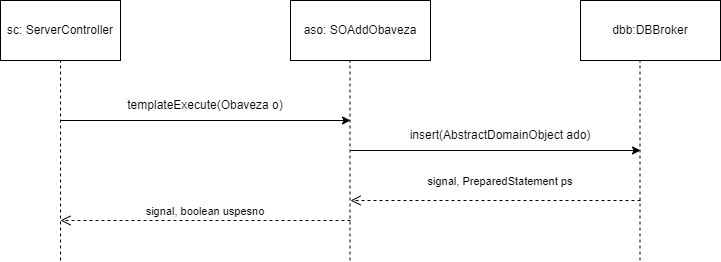
1. Уговор UG7: SOAddObaveza (Obaveza): signal;

Веза са СК: СК2

Предуслови: Вредносна и структурна ограничења над објектом Obaveza морају бити

задовољена.

Постуслови: Креирана је нова обавеза.



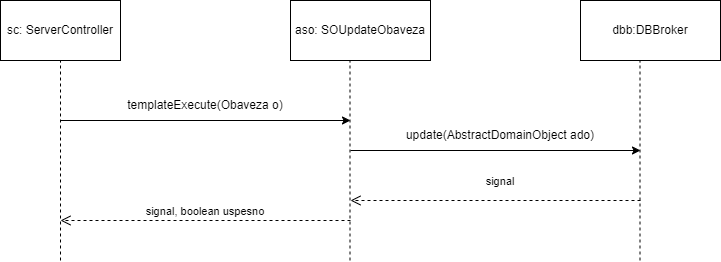
1. Уговор UG8: SOUpdateObaveza (Obaveza): signal;

Веза са СК: СК5

Предуслови: Вредносна и структурна ограничења над објектом Obaveza морају бити

задовољена.

Постуслови: Подаци о обавези су запамћени.



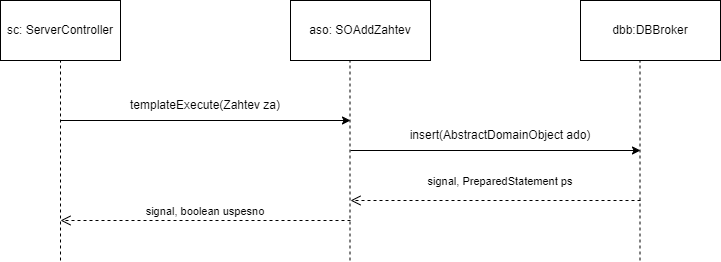
1. Уговор UG9: SOAddZahtev (Zahtev): signal;

Веза са СК: СК7

Предуслови: Вредносна и структурна ограничења над објектом Zahtev морају бити

задовољена.

Постуслови: Креиран је нови захтев.



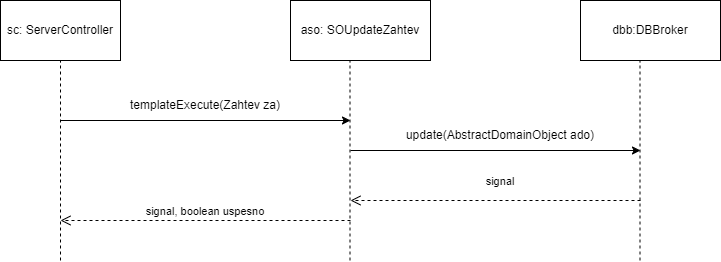
1. Уговор UG10: SOUpdateZahtev (Zahtev): signal;

Веза са СК: СК8

Предуслови: Вредносна и структурна ограничења над објектом Obaveza морају бити

задовољена.

Постуслови: Подаци о обавези су запамћени.

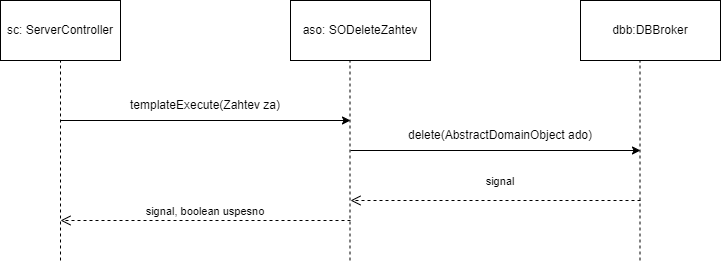


1. Уговор UG11: SODeleteZahtev (Zahtev): signal;

Веза са СК: СК9

Предуслови: Вредносна и структурна ограничења над објектом Zahtev су задовољена

Постуслови: Обрисан је захтев.

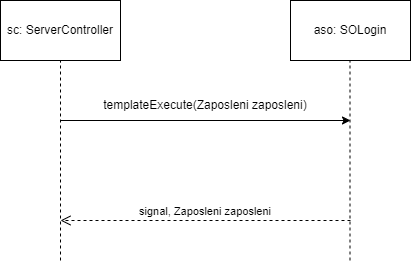


1. Уговор UG12: SOLogin(Administrator): signal;

Веза са СК: СК10

Предуслови: /

Постуслови: Администратор је улогован.

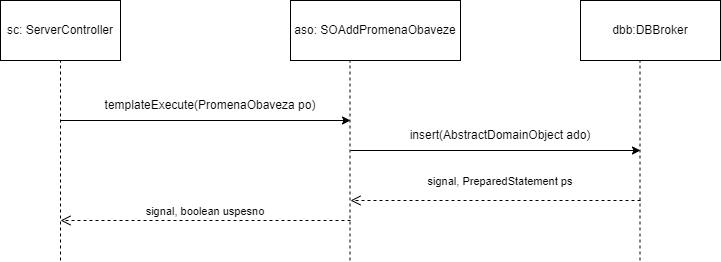


1. Уговор UG13: SOAddPromenaObaveze(Zahtev): signal;

Веза са СК: СК5

Предуслови: /

Постуслови: /

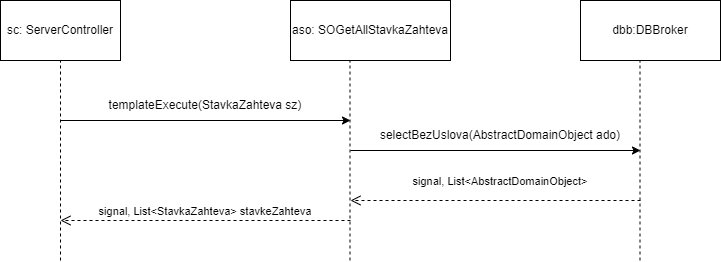


1. Уговор UG14: SOGetAllStavkaZahteva(Zahtev): signal;

Веза са СК: СК8

Предуслови: /

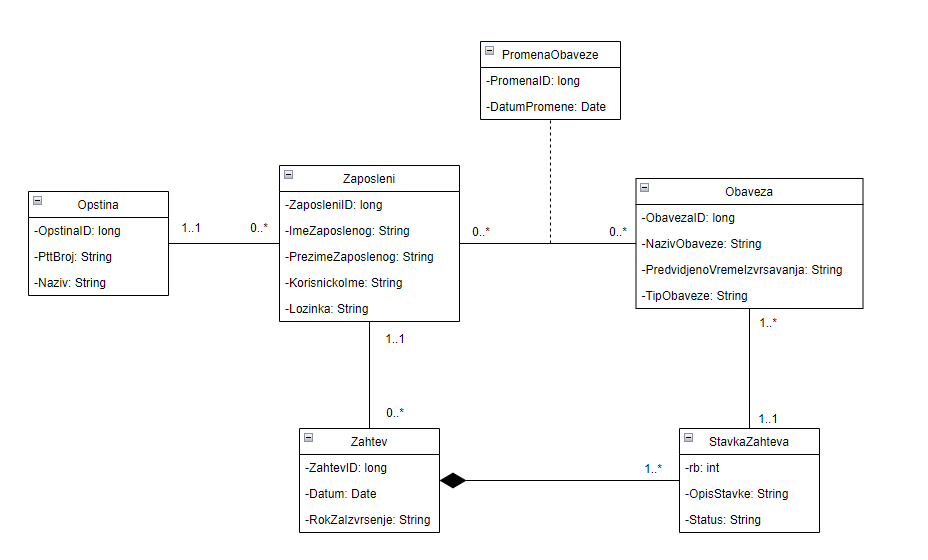
Постуслови: /



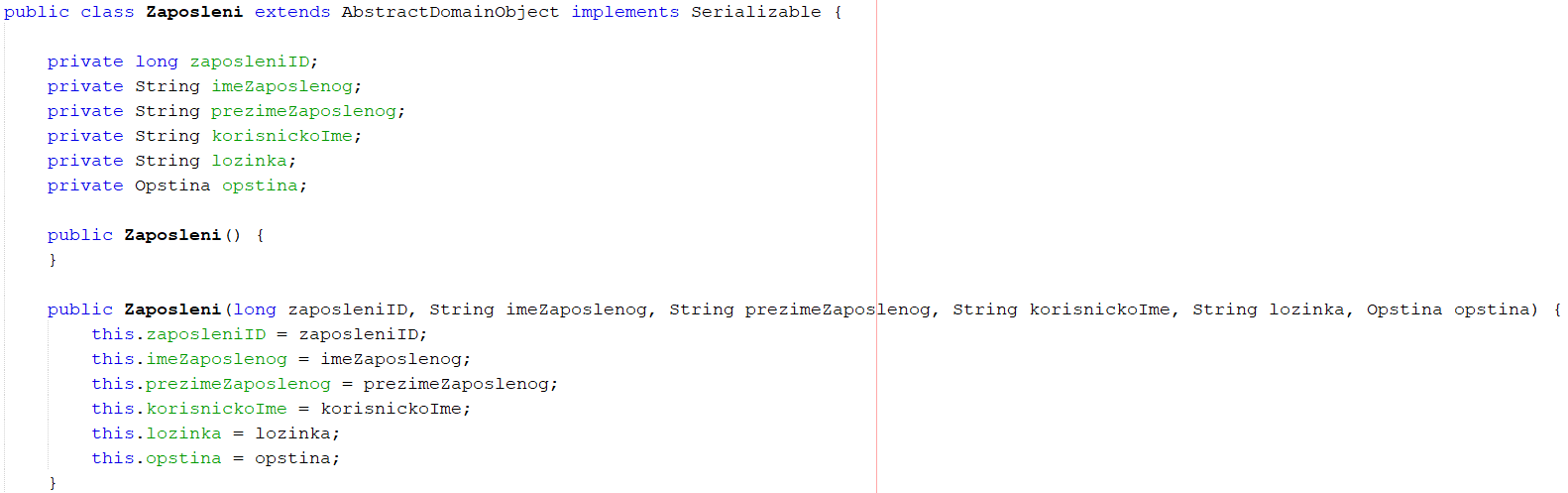
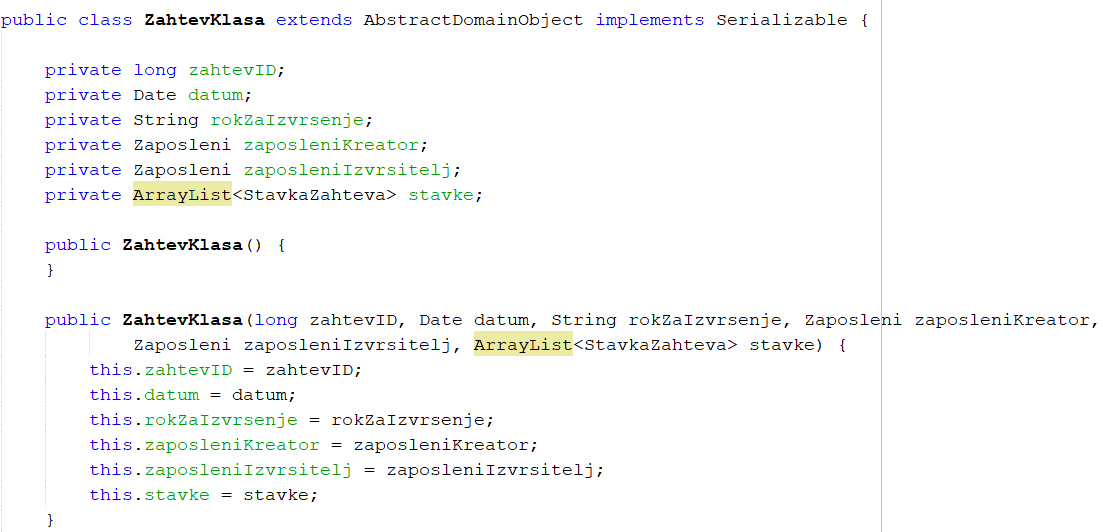
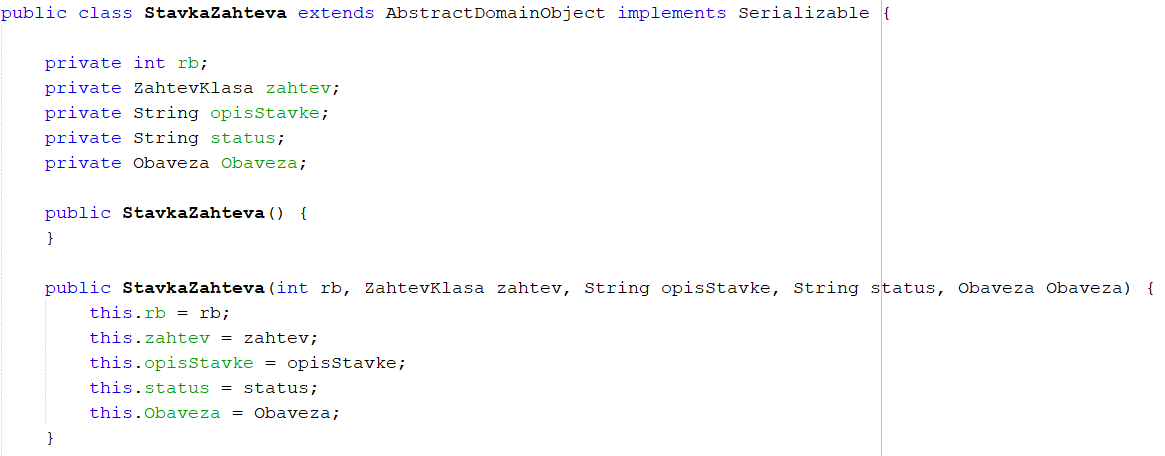
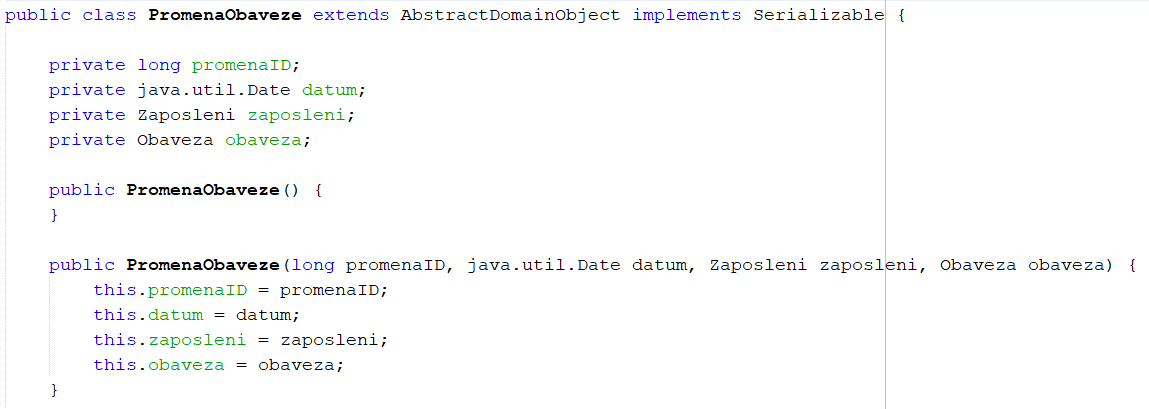
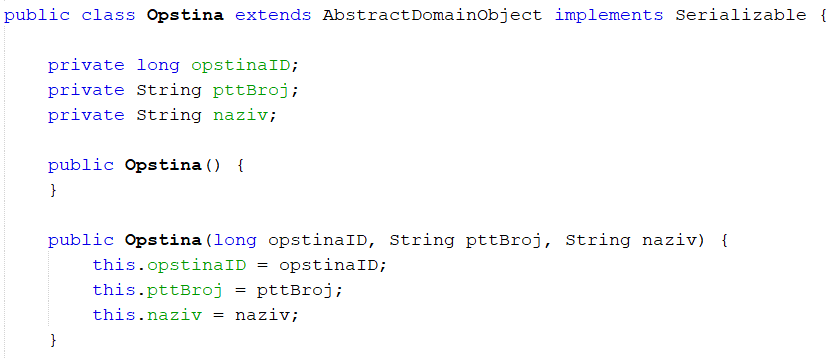
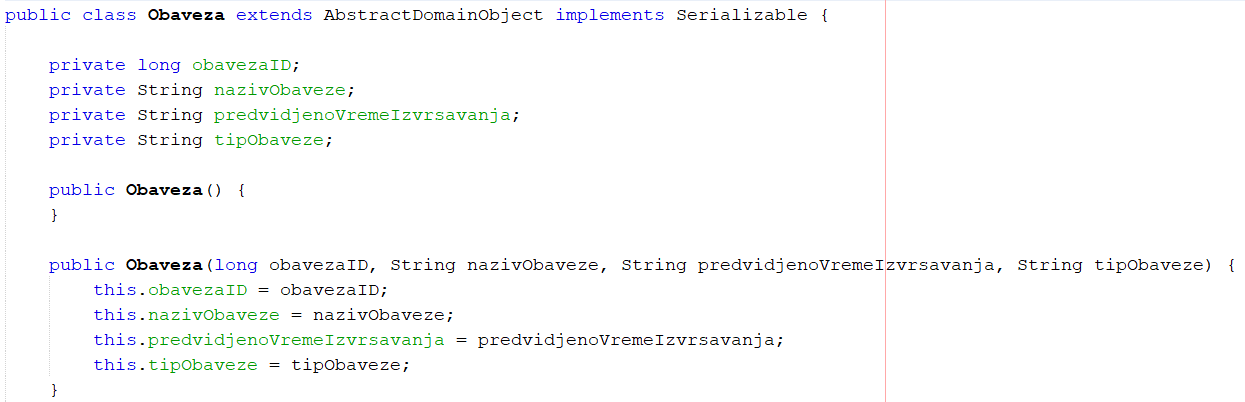
**Пројектовање структуре софтверског система**

На основу концептуалних класа израђују се доменске класе.

Концептуалне класе:



Доменске класе:



### Брокер базе података

Класа DBBroker посредује у свим операцијама са базом података. То је интерфејс који наслеђује интерфејс Repository са следећим операцијама

* public List<AbstractDomainObject> select()
* public List<OpstiDomenskiObjekat> selectBezUslova(AbstractDomainObject ado)
* public void insert(AbstractDomainObject ado)
* public void update(AbstractDomainObject ado)
* public void delete(AbstractDomainObject ado)

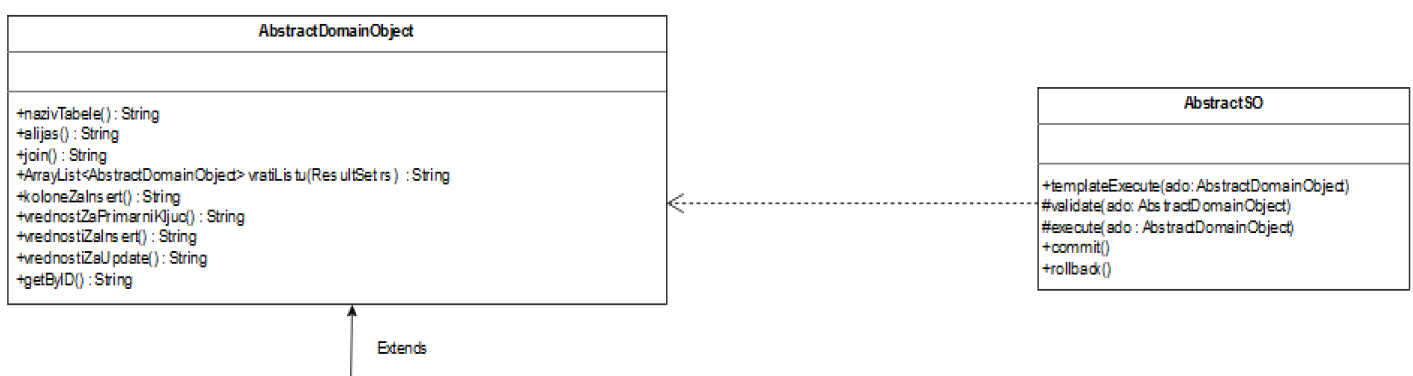
На ове операције које се односе на доменске класе, класа DBBroker додаје и имплементира операције за комуникацију са базом података:

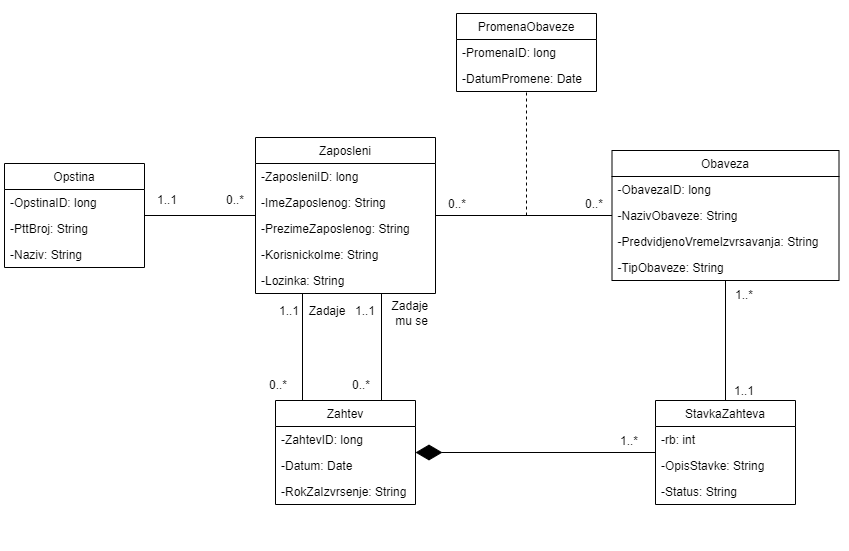
* public void commit()
* public void rollback()

Овај интерфејс имплементира класа DBBroker која даје имплементацију метода Repository интерфејса. Све те методе су генеричке, односно могу примити било који доменски објекат као параметар. То се постиже дефинисањем апстрактне класе AbstractDomainObject која садржи следеће методе:

* public abstract String nazivTabele();
* public abstract String alijas();
* public abstract String spajanje();
* public abstract ArrayList<AbstractDomainObject> selectLista(ResultSet rs);
* public abstract String koloneZaInsert();
* public abstract String primarniKljuc();
* public abstract String vrednostiZaInsert();
* public abstract String vrednostiZaUpdate();
* public abstract String id();

Свака доменска класа имплементира ове методе на одговарајући начин.

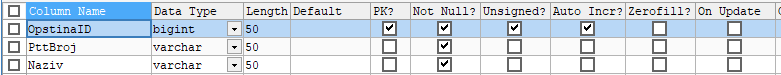




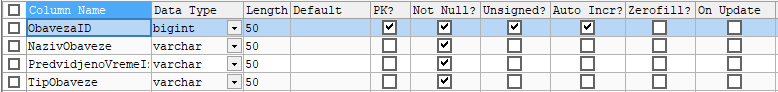
## Пројектовање складишта података

На основу доменских класа, пројектоване су табеле релационог система за управљање базом података. Систем за управљање базом података који је коришћен јесте MySQL.

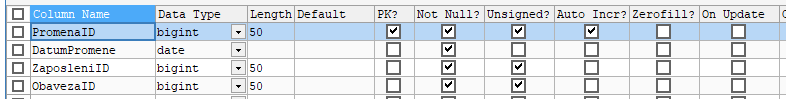
Табела Opstina



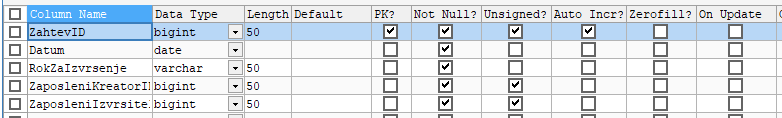
Табела Obaveza



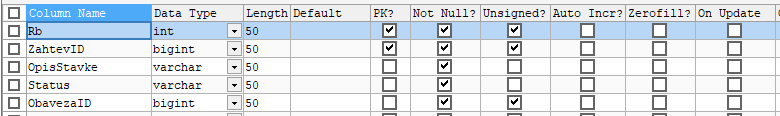
Табела PromenaObaveze



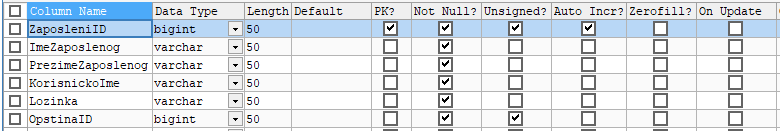
Табела Zahtev

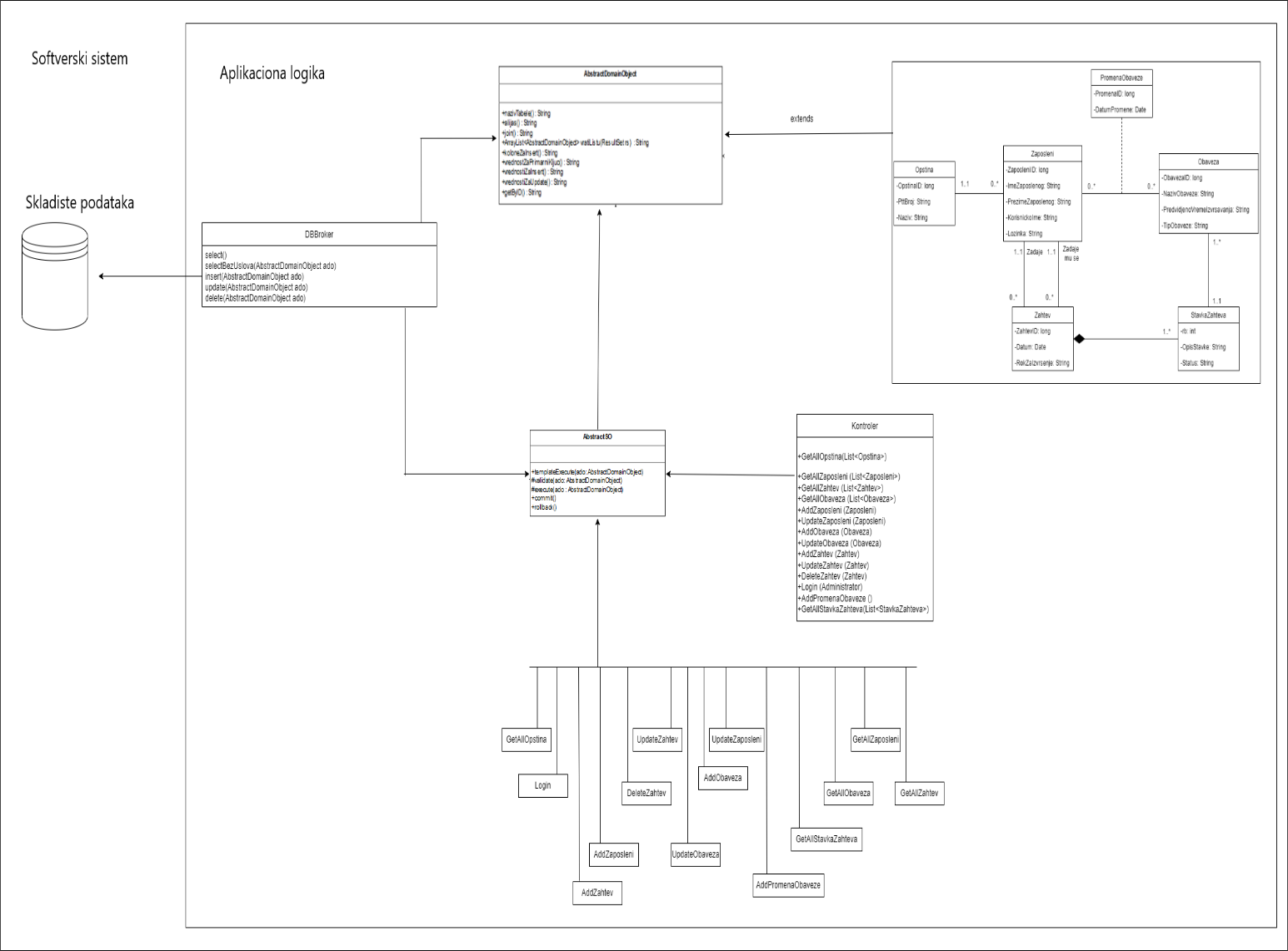


Табела StavkaZahteva



Табела Zaposleni

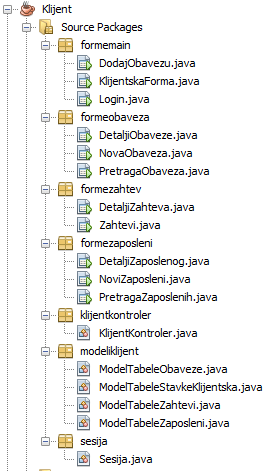


На основу претходних одељака, добија се цела архитектура софтверског система за праћење издавања услуга обављања кућних послова.

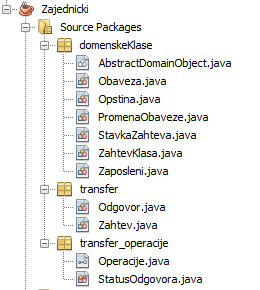
# Имплементација

Софтверски систем је развијан у програмском језику “Java”. Систем је пројектован као клијент-сервер апликација. Као развојно окружење коришћен је NetBeansIDE 12.0. На основу архитектуре софтверског система добијене су следеће софтверске класе:

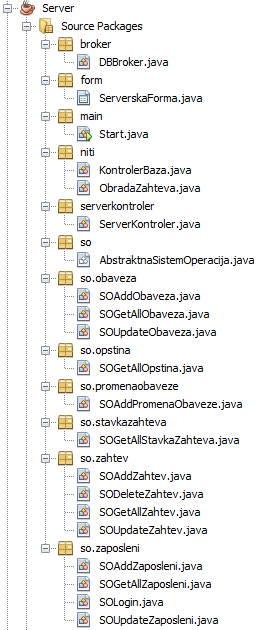
* Клијент



* Заједнички пројекат



* Сервер



# 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. Након фазе тестирања, софтвер је спреман за коришћење од стране крајњег корисника.