Универзитет у Београду

Факултет организационих наука

Лабораторија за софтверско инжењерство

**Семинарски рад из предмета**

**Пројектовање софтвера**

**Тема: Софтверски систем за праћење пројеката**

Студент:

Софија Опачић 342/17

Садржај

[Прикупљање корисничких захтева 5](#_Toc163085021)

[Вербални опис 5](#_Toc163085022)

[Случајеви коришћења 5](#_Toc163085023)

[СК1: Случај коришћења – Пријављивање на систем 7](#_Toc163085024)

[СК2 – Случај коришћења – Креирање новог запосленог 8](#_Toc163085025)

[СК3 – Случај коришћења – Измена података о запосленом 9](#_Toc163085026)

[СК4 – Случај коришћења – Креирање новог задатка 10](#_Toc163085027)

[СК5 – Случај коришћења – Измена података о задатку 11](#_Toc163085028)

[СК6: Случај коришћења – Унос новог пројекта 12](#_Toc163085029)

[СК7: Случај коришћења – Измена података о пројекту 13](#_Toc163085030)

[СК8: Случај коришћења – Увид и претрага пројеката 14](#_Toc163085031)

[Анализа 15](#_Toc163085032)

[Системски дијаграм секвенци 15](#_Toc163085033)

[ДС1: Дијаграм секвенци случаја коришћења – Пријављивање на систем 15](#_Toc163085034)

[ДС2: Дијаграм секвенци случаја коришћења : Креирање новог запосленог 17](#_Toc163085035)

[ДС3: Дијаграм секвенци случаја коришћења – Измена података о запосленом 19](#_Toc163085036)

[ДС4: Дијаграм секвенци случаја коришћења : Креирање новог задатка 21](#_Toc163085037)

[ДС5: Дијаграм секвенци – Измена података о задатку 22](#_Toc163085038)

[ДС6: Дијаграм секвенци случаја коришћења – Унос новог пројекта 24](#_Toc163085039)

[ДС7: Дијаграм секвенце случаја коришћења – Измена података о пројекту 26](#_Toc163085040)

[ДС8: Дијаграм секвенци случаја коришћења – Увид и претрага пројеката 29](#_Toc163085041)

[Понашање софтверског система – Дефинисање уговора о системским операцијама 31](#_Toc163085042)

[Структура софтверског система – Концептуални(доменски) модел 33](#_Toc163085043)

[Структура софтверског система – релациони модел 33](#_Toc163085044)

[Пројектовање 37](#_Toc163085045)

[Пројектовање корисничког интерфејса 38](#_Toc163085046)

[Пројектовање екранских форми 38](#_Toc163085047)

[СК1: Случај коришћења - Пријављивање на систем 39](#_Toc163085048)

[СК2 – Случај коришћења – Креирање новог запосленог 40](#_Toc163085049)

[СК3 – Случај коришћења – Измена података о запосленом 42](#_Toc163085050)

[СК4 – Случај коришћења – Креирање новог задатка 44](#_Toc163085051)

[СК5 – Случај коришћења – Измена података о задатку 46](#_Toc163085052)

[СК6: Случај коришћења – Унос новог пројекта 48](#_Toc163085053)

[СК7: Случај коришћења – Измена података о пројекту 51](#_Toc163085054)

[СК8: Случај коришћења – Увид и претрага пројеката 54](#_Toc163085055)

[Пројектовање контролера графичког интерфејса (GUI Controller) 57](#_Toc163085056)

[Пројектовање апликационе логике 57](#_Toc163085057)

[Контролер апликационе логике 58](#_Toc163085058)

[Пројектовање понашања софтверског система (Системске операције) 58](#_Toc163085059)

[Пројектовање структуре софтверског система (Доменске класе) 64](#_Toc163085060)

[Пројектовање складиша података 69](#_Toc163085061)

[Имплементација 71](#_Toc163085062)

[Тестирање 72](#_Toc163085063)

[Закључак 72](#_Toc163085064)

[Литература 73](#_Toc163085065)

# Прикупљање корисничких захтева

## Вербални опис

Потребно је направити софтверско решење које ће омогућити кориснику да лакше управља и прати рад на више пројеката.

Апликација омогућава пријављивање на систем. У завиности од типа корисника систем нуди различите опције.

Корисник- запослени има могућност увида у све пројекте на којима он ради као и његове личне задатке и могућност претраге пројеката по статусу (завршен, активан) или по типу задатка. Запослени може да мења податке о пројекту (назив, статус, додељивање задатака) само ако је он вођа тог пројекта.

Корисник- администратор поред горе наведених опција има могућност креирања нових запослених и њихову измену, креирања нових задатака и њихову измену као и креирања новог пројекта, измену података о пројекту и брисање пројекта (при брисању пројекта пројекат се брише са листе пројеката свих запослених који су радили на њему као и задаци запослених који су били везани за тај пројекат).

## Случајеви коришћења

Идентификовано је осам случајева коришћења:

1. Пријављивање на систем
2. Унос новог запосленог
3. Измена података о запосленом
4. Унос новог задатка
5. Измена података о задатку
6. Унос новог пројекта
7. Измена података о пројекту
8. Увид и претрага пројекта

A screenshot of a computer

Description automatically generated

Слика 1 : Модел случајева коришћења

### СК1: Случај коришћења – Пријављивање на систем

**Назив СК**

Пријављивање на систем

**Актори СК**

Запослени или Администратор

**Учесници СК**

Актор и систем (програм)

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

**Основни сценарио СК**

1. Актор **уноси** корисничко име у форму за пријављивање. (АПУСО)

2. Актор **контролише** да ли је коректно унео податке у форму за пријављивање. (АНСО)

3. Актор **позива** систем да пријави корисника. (АПСО)

4. Систем **памти** податке и пријављује актора. (СО)

5. Систем **приказује** актору главну форму програма. (ИА)

Алтернативна сценарија

5.1 Уколико систем не може да пронађе корисника са унетим корисничким именом и шифром он приказује актору поруку “ У систему не постоји корисник са унетим корисничким именом”. (ИА)

### СК2 – Случај коришћења – Креирање новог запосленог

**Назив СК**

Креирање новог запосленог

**Актори СК**

Администратор

**Учесници СК**

Администратор и систем (програм)

**Предуслов**: Систем је укључен и администратор je улогован под својим корисничким именом. Систем приказује форму за унос запосленог.

**Основни сценарио СК**

1. Администратор **уноси** податке о запосленом (име, презиме, корисничко име). (АПУСО)
2. Администратор **контролише** да ли је коректно унео податке о запосленом. (АНСО)
3. Администратор **позива** систем да запамти податке о запосленом. (АПСО)
4. Систем **памти** податке о запосленом. (СО)
5. Систем **затвара** форму за унос запосленог и администратору приказује форму за приказ свих запослених. (ИА)

Алтернативна сценарија

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

### СК3 – Случај коришћења – Измена података о запосленом

**Назив СК**

Измена података о запосленом

**Актори СК**

Администратор

**Учесници СК**

Администратор и систем (програм)

**Предуслов**: Систем је укључен и администратор je улогован под својим корисничким именом. Систем приказује форму за рад са запосленима. Учитана је листа запослених.

**Основни сценарио СК**

1. Администратор **бира** запосленог чије податке жели да измени. (АПСО)
2. Систем **учитава** запосленог.(СО)
3. Систем **приказује** администратору запосленог и отвара форму за измену података. (ИА)
4. Администратор **уноси** **(мења)** податке о запосленом. (АПУСО)
5. Амдинистратор **контролише** да ли је коректно унео податке о запосленом. (АНСО)
6. Администратор **позива** систем да запамти податке о запосленом. (АПСО)
7. Систем **памти** податке о запосленом. (СО)
8. Систем **затвара** форму за измену података и приказује администратору форму са свим запосленима. (ИА)

Алтернативна сценарија

3.1 Уколико систем не може да учита запосленог он приказује адмнистратору поруку: “Систем не може да учита запосленог”. Прекида се извршење сценарија. (ИА)

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

### СК4 – Случај коришћења – Креирање новог задатка

**Назив СК**

Креирање новог задатка

**Актори СК**

Администратор

**Учесници СК**

Администратор и систем (програм)

**Предуслов**: Систем је укључен и администратор je улогован под својим корисничким именом. Систем приказује форму за унос новог задатка.

**Основни сценарио СК**

1. Администратор **уноси** податке о задатку (назив, опис). (АПУСО)
2. Администратор **контролише** да ли је коректно унео податке о задатку. (АНСО)
3. Администратор **позива** систем да запамти податке о задатку. (АПСО)
4. Систем **памти** податке о задатку. (СО)
5. Систем **затвара** форму за унос задатка и администратору приказује форму за приказ свих задатака. (ИА)

Алтернативна сценарија

5.1 Уколико систем не може да запамти податке о задатку он приказује администатору поруку “Систем не може да запамти задатак”. (ИА)

### СК5 – Случај коришћења – Измена података о задатку

**Назив СК**

Измена података о задатку

**Актори СК**

Администратор

**Учесници СК**

Администратор и систем (програм)

**Предуслов**: Систем је укључен и администратор je улогован под својим корисничким именом. Систем приказује форму за рад са задацима. Учитана је листа задатака.

**Основни сценарио СК**

1. Администратор **бира** задатак чије податке жели да измени. (АПСО)
2. Систем **учитава** задатак.(СО)
3. Систем **приказује** администратору задатак и отвара форму за измену података. (ИА)
4. Администратор **уноси** **(мења)** податке о задакту. (АПУСО)
5. Амдинистратор **контролише** да ли је коректно унео податке о задаку. (АНСО)
6. Администратор **позива** систем да запамти податке о задатку. (АПСО)
7. Систем **памти** податке о задацима. (СО)
8. Систем **затвара** форму за измену података и приказује администратору форму са свим задацима. (ИА)

Алтернативна сценарија

3.1 Уколико систем не може да учита задатак он приказује адмнистратору поруку: “Систем не може да учита задатак”. Прекида се извршење сценарија. (ИА)

8.1 Уколико систем не може да запамти податке о задатку он приказује администатору поруку “Систем не може да запамти задатак”. (ИА)

### СК6: Случај коришћења – Унос новог пројекта

**Назив СК**

Унос новог пројекта

**Актори СК**

Адмнистратор

**Учесници СК**

Администратор и систем (програм)

**Предуслов**: Систем је укључен . Тип корисника који је улогован је адмнистратор. Систем приказује форму за унос пројеката.

**Основни сценарио СК**

1. Адмнистратор **уноси** податке о пројекту (назив пројекта, статус, додељује запосленог као вођу пројекта и креира задатке за пројекат (назив, одговорни запослени)). (АПУСО)
2. Адмнистратор **контролише** да ли је коректно унео податке о пројекту. (АНСО)
3. Адмнистратор **позива** систем да запамти унете податке о пројекту. (АПСО)
4. Систем **памти** податке о пројекту. (СО)
5. Систем **затвара** форму за унос пројеката иадмнистратору **приказује** форму на којој је листа свих пројеката. (ИА)

Алтернативна сценарија

5.1 Уколико систем не може да запамти пројекат он приказује адмнистратору поруку: “Систем не може да запамти пројекат”. (ИА)

### СК7: Случај коришћења – Измена података о пројекту

**Назив СК**

Измена података о пројекту

**Актори СК**

Адмнистратор или Запослени

**Учесници СК**

Актор и систем (програм)

**Предуслов**: Систем је укључен . Систем приказује форму за рад са пројектима. Учитана је листа пројеката.

**Основни сценарио СК**

1. Актор **бира** пројекат. (АПСО)
2. Систем **учитава** пројекат.(СО)
3. Систем **приказује** актору форму за измену пројекта. (ИА)
4. Актор **уноси (мења)** све или неке податке о пројекту (назив пројекта, статус, додељује запосленог као вођу пројекта и креира задатке за пројекат (одговорни запослени, задаци)). (АПУСО)
5. Актор **контролише** да ли је коректно унео податке о пројекту. (АНСО)
6. Актор **позива** систем да запамти податке о пројекту. (АПСО)
7. Систем **памти** податке о пројекту. (СО)
8. Систем **затвара** форму за измену пројекта иактору **приказује** форму на којој је листа свих пројеката. (ИА)

Алтернативна сценарија

3.1 Уколико систем не може да учита пројекат он приказује актору поруку: “Систем не може да учита пројекат”. Прекида се извршење сценарија. (ИА)

8.1 Уколико систем не може да запамти податке о пројекту он приказује актору поруку “Систем не може да запамти пројекат”. (ИА)

### СК8: Случај коришћења – Увид и претрага пројеката

**Назив СК**

Увид и претрага пројеката и задатака

**Актори СК**

Запослени или Адмнистратор

**Учесници СК**

Актор и систем (програм)

**Предуслов**: Систем је укључен . Актори су улоговани. Систем приказује форму за пројекте. Учитана је листа пројеката.

**Основни сценарио СК**

1. Актор **уноси** вредности према којима претражује пројекат. (АПУСО)
2. Актор **позива** систем да нађе пројекат по задатим вредностима. (АПСО)
3. Систем **тражи** пројекат по задатој вредности. (СО)
4. Систем приказује актору пројекте по задатим вредностима (ИА)

Алтернативна сценарија

4**.**1 Уколико систем не може да пронађе пројекат он приказује актору поруку “Систем не може да пронађе пројекат ”. (ИА)

### 

# Анализа

## Системски дијаграм секвенци

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

1. Актор **позива** систем да пријави корисника. (АПСО)
2. Систем **приказује** актору главну форму програма. (ИА)

A black and white screen

Description automatically generated

Алтернативна сценарија

2.1 Уколико систем не може да пронађе корисника са унетим корисничким именом и шифром он приказује актору поруку “ У систему не постоји корисник са унетим корисничким именом“

A black and white screen

Description automatically generated

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

1. Сигнал **find(username)**

### ДС2: Дијаграм секвенци случаја коришћења : Креирање новог запосленог

1. Администратор **позива** систем да запамти податке о запосленом. (АПСО)
2. Систем **затвара** форму за унос запосленог и администратору приказује форму за приказ свих запослених. (ИА)

A black rectangular object with white rectangles

Description automatically generated

Алтернативна сценарија:

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

A black rectangular object with white rectangles

Description automatically generated

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

1. Сигнал  **save(employee)**

### ДС3: Дијаграм секвенци случаја коришћења – Измена података о запосленом

1. Форма **позива** систем да учита листу запослених. (АПСО)
2. Систем **враћа** форми листу запослених.(ИА)
3. Администратор **бира** запосленог чије податке жели да измени. (АПСО)
4. Систем **приказује** администратору запосленог и отвара форму за измену података. (ИА)
5. Администратор **позива** систем да запамти податке о запосленом. (АПСО)
6. Систем **затвара** форму за измену података и приказује администратору форму са свим запосленима. (ИА)

A screenshot of a phone

Description automatically generated

Алтернативна сценарија

4.1 Уколико систем не може да учита запосленог он приказује адмнистратору поруку: “Систем не може да учита запосленог”. Прекида се извршење сценарија. (ИА)

A screenshot of a phone

Description automatically generated

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

**A screenshot of a phone

Description automatically generated**

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

1. Сигнал **getList(<List>Employees)**
2. Сигнал **EmployeeForm(Employee)**
3. Сигнал **save(Employee)**

### ДС4: Дијаграм секвенци случаја коришћења : Креирање новог задатка

1. Администратор **позива** систем да запамти податке о задатку. (АПСО)
2. Систем **затвара** форму за унос задатка и администратору приказује форму за приказ свих задатака. (ИА)

A black rectangular object with white rectangles

Description automatically generated

Алтернативна сценарија

* 1. Уколико систем не може да запамти податке о задатку он приказује администатору поруку “Систем не може да запамти задатак”. (ИА)

A black rectangular object with white rectangles

Description automatically generated

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

1. Сигнал **save(task)**

### ДС5: Дијаграм секвенци – Измена података о задатку

1. Форма **позива** систем да учита листу задатака. (АПСО)
2. Систем **враћа** форми листу задатака.(ИА)
3. Администратор **бира** задатак чије податке жели да измени. (АПСО)
4. Систем **приказује** администратору задатак и отвара форму за измену података. (ИА)
5. Администратор **позива** систем да запамти податке о задатку. (АПСО)
6. Систем **затвара** форму за измену података и приказује администратору форму са свим задацима. (ИА)

A screenshot of a phone

Description automatically generated

Алтернативна сценарија

4.1 Уколико систем не може да учита задатак он приказује адмнистратору поруку: “Систем не може да учита задатак”. Прекида се извршење сценарија. (ИА)

A screenshot of a phone

Description automatically generated

* 1. Уколико систем не може да запамти податке о задатку он приказује администатору поруку “Систем не може да запамти задатак”. (ИА)

A screenshot of a phone

Description automatically generated

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

1. Сигнал **getList(<List>Tasks)**
2. Сигнал **TaskForm(Task)**
3. Сигнал **save(task)**

### ДС6: Дијаграм секвенци случаја коришћења – Унос новог пројекта

1. Форма **позива** систем да учита листу запослених. (АПСО)
2. Систем **враћа** форми листу запослених.(ИА)
3. Форма **позива** систем да учита листу задатака. (АПСО)
4. Систем **враћа** форми листу задатака.(ИА)
5. Адмнистратор **позива** систем да запамти унете податке о пројекту. (АПСО)
6. Систем **затвара** форму за унос пројеката иадмнистратору **приказује** форму на којој је листа свих пројеката. (ИА)

A screenshot of a phone

Description automatically generated

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

A screenshot of a phone

Description automatically generated

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

1. Сигнал **getList(<List>Еmployees)**
2. Сигнал **getList(<List>Tasks)**
3. Сигнал **save(project)**

### ДС7: Дијаграм секвенце случаја коришћења – Измена података о пројекту

1. Форма **позива** систем да учита листу пројеката. (АПСО)
2. Систем **враћа** форми листу пројеката.(ИА)
3. Форма **позива** систем да учита листу запослених. (АПСО)
4. Систем **враћа** форми листу запослених.(ИА)
5. Форма **позива** систем да учита листу задатака. (АПСО)
6. Систем **враћа** форми листу задатака.(ИА)
7. Актор **бира** пројекат. (АПСО)
8. Систем **приказује** актору форму за измену пројекта. (ИА)
9. Актор **позива** систем да запамти податке о пројекту. (АПСО)
10. Систем **затвара** форму за измену пројекта иактору **приказује** форму на којој је листа свих пројеката. (ИА)

A black background with white lines

Description automatically generated

Алтернативна сценарија

* 1. Уколико систем не може да учита пројекат он приказује актору поруку: “Систем не може да учита пројекат”. Прекида се извршење сценарија. (ИА)

A black background with white lines

Description automatically generated

* 1. Уколико систем не може да запамти податке о пројекту он приказује актору поруку “Систем не може да запамти пројекат”. (ИА)

A black background with white lines

Description automatically generated

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

1. Сигнал **getList(<List>Projects)**
2. Сигнал **getList(<List>Еmployees)**
3. Сигнал **getList(<List>Tasks)**
4. Сигнал **ProjectForm(Project)**
5. Сигнал **save(project)**

### ДС8: Дијаграм секвенци случаја коришћења – Увид и претрага пројеката

1. Форма **позива** систем да учита листу пројеката. (АПСО)
2. Систем **враћа** форми листу пројеката.(ИА)
3. Актор **позива** систем да нађе пројекат по задатим вредностима. (АПСО)
4. Систем приказује актору пројекте по задатим вредностима (ИА)

A screenshot of a phone

Description automatically generated

Алтернативна сценарија

* 1. Уколико систем не може да пронађе пројекат он приказује актору поруку “Систем не може да пронађе пројекат ”. (ИА)

A screenshot of a phone

Description automatically generated

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

1. Сигнал **getList(<List>Projects)**
2. Сигнал **getList(<List>Projects, params)**

## Понашање софтверског система – Дефинисање уговора о системским операцијама

1. Уговор **UG1**: find

**Операција: find(username):** сигнал

**Веза са СК:** СК1

**Предуслови:** *Подаци о објекту Zaposleni морају да се слажу са одацима у бази*

**Постуслови:** *Корисник(Запослени) је пријављен на систем*

1. Уговор **UG2**: saveEmployee

**Операција:** **save(employee):** сигнал

**Веза са СК:** СК2, CK3

**Предуслови:** *Структурна и вредносна ограничења над Employee објектом морају бити задовољена*

**Постуслови:** *Сачувани су подаци о запосленом*

1. Уговор **UG3**: getListEmployees

**Операција: getList(<List>Employees):** сигнал

**Веза са СК:** СК3,CK6,CK7

**Предуслови:**

**Постуслови:** *Учитана је листа запослених*

1. Уговор **UG4**: EmployeeForm

**Операција:** **EmployeeForm(Employee):** сигнал

**Веза са СК:** СК3

**Предуслови:**

**Постуслови:** *Учитана је форма за рад са запосленим*

1. Уговор **UG5**: saveTask

**Операција: save(task):** сигнал

**Веза са СК:** СК4, CK5

**Предуслови:** *Структурна и вредносна ограничења над Task објектом морају бити задовољена*

**Постуслови:** *Сачувани су подаци о задатку*

1. Уговор **UG6**: getListTask

**Операција: getList(<List>Tasks):** сигнал

**Веза са СК:** СК5, CK6, CK7

**Предуслови:**

**Постуслови:** *Учитана је листа задатака*

1. Уговор **UG7**: TaskForm

**Операција:** **TaskForm(Task):** сигнал

**Веза са СК:** СК5

**Предуслови:**

**Постуслови:** *Учитана је форма за рад са задацима*

1. Уговор **UG8**: saveProject

**Операција:** **save(project):** сигнал

**Веза са СК:** СК6, CK7

**Предуслови:** *Структурна и вредносна ограничења над Project објектом морају бити задовољена*

**Постуслови:** *Сачувани су подаци о пројекту*

1. Уговор **UG9**: getList

**Операција:** **getList(<List>Projects):** сигнал

**Веза са СК:** СК7,CK8

**Предуслови:**

**Постуслови:** *Учитана је листа пројеката*

1. Уговор **UG10**: ProjectForm

**Операција: ProjectForm(Project):** сигнал

**Веза са СК:** СК7

**Предуслови:**

**Постуслови:** *Учитана је форма за рад са пројектима*

1. Уговор **UG11**: getList2

**Операција:** **getList(<List>Projects, params):** сигнал

**Веза са СК:** СК8

**Предуслови:**

**Постуслови:** *Учитана је листа пројеката по задатим параметрима*

## Структура софтверског система – Концептуални(доменски) модел

A black background with white text

Description automatically generated

## Структура софтверског система – релациони модел

zaposleni(id, ime, prezime, username, tip, obrisan)

zadaci(id, naziv, opis, obrisan)

projekti(id, naziv,obrisan, aktivan, *zaposleniID)*

projekatstavke(id, *projekatID, zadatakID, zaposleniID*, obrisan)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela PROJEKTI** | | **PROSTO VREDNOSNO OGRANIČENJE** | | **SLOŽENO VREDNOSNO OGRANIČENJE** | | **STRUKTURNO OGRANIČENJE** |
| **ATRIBUTI** | **IME** | **TIP ATRIBUTA** | **VREDNOST ATRIBUTA** | **MEĐUZAVISNOT ATRIBUTA JEDNE TABELE** | **MEĐUZAVISNOT ATRIBUTA VIŠE TABELA** | **INSERT** /  **UPDATE** CASCADES projekatstavke  **DELETE** CASCADES projekatstavke |
| id | int | Not null and >0 |  |  |
| naziv | String | Not null |  |  |
| zaposleniID | int | Not null and >0 |  |  |
| obrisan | boolean | 0,1  Default 0 |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela ZADATAK** | | **PROSTO VREDNOSNO OGRANIČENJE** | | **SLOŽENO VREDNOSNO OGRANIČENJE** | | | **STRUKTURNO OGRANIČENJE** |
| **ATRIBUTI** | **IME** | **TIP ATRIBUTA** | **VREDNOST ATRIBUTA** | **MEĐUZAVISNOT ATRIBUTA JEDNE TABELE** | **MEĐUZAVISNOT ATRIBUTA VIŠE TABELA** | | **INSERT** /  **UPDATE** CASCADES projekatstavke  **DELETE** RESTRICTED projekatstavke |
| id | int | Not null and >0 |  | |  |
| naziv | String | Not null |  | |  |
| opis | String | Not null |  | |  |
| orbisan | boolean | 0,1  Default 0 |  | |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela ZAPOSLENI** | | **PROSTO VREDNOSNO OGRANIČENJE** | | **SLOŽENO VREDNOSNO OGRANIČENJE** | | **STRUKTURNO OGRANIČENJE** |
| **ATRIBUTI** | **IME** | **TIP ATRIBUTA** | **VREDNOST ATRIBUTA** | **MEĐUZAVISNOT ATRIBUTA JEDNE TABELE** | **MEĐUZAVISNOT ATRIBUTA VIŠE TABELA** | **INSERT** /  **UPDATE** CASCADES projekatstavke  **DELETE** RESTRICTED projekatstavke |
| id | int | Not null and >0 |  |  |
| ime | String | Not null |  |  |
| prezime | String | Not null |  |  |
| tip | boolean | 0,1  Default 0 |  |  |
| obrisan | boolean | 0,1  Default 0 |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tabela PROJEKATSTAVKE** | | **PROSTO VREDNOSNO OGRANIČENJE** | | **SLOŽENO VREDNOSNO OGRANIČENJE** | | **STRUKTURNO OGRANIČENJE** |
| **ATRIBUTI** | **IME** | **TIP ATRIBUTA** | **VREDNOST ATRIBUTA** | **MEĐUZAVISNOT ATRIBUTA JEDNE TABELE** | **MEĐUZAVISNOT ATRIBUTA VIŠE TABELA** | **INSERT** RESTRICTED projekti,  zaposleni,  zadaci  **UPDATE** /  **DELETE** / |
| id | int | Not null and >0 |  |  |
| projekatID | int | Not null and >0 |  |  |
| zaposleniID | int | Not null and >0 |  |  |
| zadatakID | int | Not null and >0 |  |  |
| obrisan | boolean | 0,1  Default 0 |  |  |

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

A computer screen shot of a computer

Description automatically generated

# Пројектовање

Фаза пројектовања ближе нам описује структуру и понашање система. Да бисмо то постигли, неопходно је пројектовати кориснички интерфејс (екранске форме за унос података, GUI координатор, бројни специфични контролери), апликациону логику (главни контролер апликационе и пословне логике), перзистентни оквир за рад са подацима (брокер базе података, апстрактни и конкретни репозиторијуми итд.) и саме базе података (завршено у потпуности у претходној фази).

Архитектура софтверског система је имплементирана према MVC патерну и трослојна је:

* Кориснички интерфејс
* Апликациона логика
* Складиште података

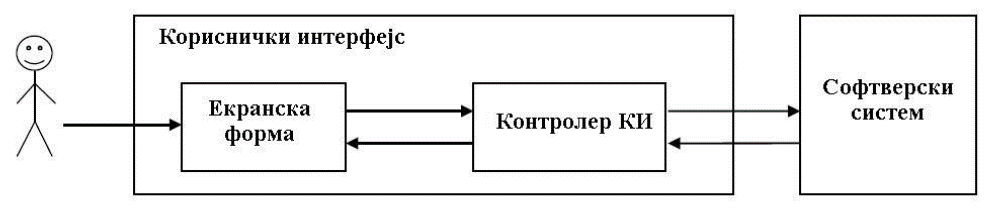
Коришћена је и клијент-сервер архитектура, те се први слој (кориснички интерфејс) одиграва на клијентској страни, а остали слојеви се извршавају на серверу.



Слика 2: Тронивојска архитектура

## Пројектовање корисничког интерфејса

Кориснички интерфејс представља улазно-излазну компоненту софтверског система (познату и као UI компонента – од енглеских речи "user interface"). Састоји се од следећих елемената: • Екрански обрасци • GUI контролери (контролери корисничког интерфејса) Одлуку који контролер треба позвати доноси компонента под називом "ГУИ координатор".



## Пројектовање екранских форми

Серверска страна програма нема форму.

### СК1: Случај коришћења - Пријављивање на систем

**Назив СК**

Пријављивање на систем

**Актори СК**

Запослени или Администратор

**Учесници СК**

Актор и систем (програм)

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

A screenshot of a computer

Description automatically generated

**Основни сценарио СК**

1. Актор **уноси** корисничко име у форму за пријављивање. (АПУСО)

2. Актор **контролише** да ли је коректно унео податке у форму за пријављивање. (АНСО)

A screenshot of a computer

Description automatically generated

1. Актор **позива** систем да пријави корисника. (АПСО)

(Опис акције:кликом на дугме “Prijava”, позива се системска операција find(username))

4. Систем **памти** податке и пријављује корисника. (СО)

5. Систем **приказује** актору главну форму програма. (ИА)

A screenshot of a computer

Description automatically generated

Алтернативна сценарија

5.1 Уколико систем не може да пронађе корисника са унетим корисничким именом и шифром он приказује актору поруку “ У систему не постоји корисник са унетим корисничким именом”. (ИА)

A screenshot of a computer

Description automatically generated

### СК2 – Случај коришћења – Креирање новог запосленог

**Назив СК**

Креирање новог запосленог

**Актори СК**

Администратор

**Учесници СК**

Администратор и систем (програм)

**Предуслов**: Систем је укључен и администратор je улогован под својим корисничким именом. Систем приказује форму за унос запосленог.

A screenshot of a computer

Description automatically generated

**Основни сценарио СК**

1. Администратор **уноси** податке о запосленом (име, презиме, корисничко име). (АПУСО)

A screenshot of a computer

Description automatically generated

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

(Опис акције: Кликом на дугме “Sačuvaj“ позива се системка операција save(employee))

1. Систем **памти** податке о запосленом. (СО)
2. Систем **затвара** форму за унос запосленог и администратору приказује форму за приказ свих запослених. (ИА)

A screenshot of a computer

Description automatically generated

Алтернативна сценарија

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

A screenshot of a computer

Description automatically generated

### СК3 – Случај коришћења – Измена података о запосленом

**Назив СК**

Измена података о запосленом

**Актори СК**

Администратор

**Учесници СК**

Администратор и систем (програм)

**Предуслов**: Систем је укључен и администратор je улогован под својим корисничким именом. Систем приказује форму за рад са запосленима. Учитана је листа запослених.

A screenshot of a computer

Description automatically generated

**Основни сценарио СК**

1. Администратор **бира** запосленог чије податке жели да измени. (АПСО)

(Опис акције: притиском на запосленог па затим на дугме “Izmeni” позива се ситемска операција EmployeeForm(employee))

1. Систем **учитава** запосленог.(СО)
2. Систем **приказује** администратору запосленог и отвара форму за измену података. (ИА)

A screenshot of a computer

Description automatically generated

1. Администратор **уноси** **(мења)** податке о запосленом. (АПУСО)

A screenshot of a computer

Description automatically generated

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

(Опис акције: Кликом на дугме “Sačuvaj“ позива се системка операција save(employee))

1. Систем **памти** податке о запосленом. (СО)
2. Систем **затвара** форму за измену података и приказује администратору форму са свим запосленима. (ИА)

A screenshot of a computer

Description automatically generated

Алтернативна сценарија

* 1. Уколико систем не може да учита запосленог он приказује адмнистратору поруку: “Систем не може да учита запосленог”. Прекида се извршење сценарија. (ИА)

A screen shot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

### СК4 – Случај коришћења – Креирање новог задатка

**Назив СК**

Креирање новог задатка

**Актори СК**

Администратор

**Учесници СК**

Администратор и систем (програм)

**Предуслов**: Систем је укључен и администратор je улогован под својим корисничким именом. Систем приказује форму за унос новог задатка.

A screenshot of a computer

Description automatically generated

**Основни сценарио СК**

1. Администратор **уноси** податке о задатку (назив, опис). (АПУСО)

A screenshot of a computer

Description automatically generated

1. Администратор **контролише** да ли је коректно унео податке о задатку. (АНСО)
2. Администратор **позива** систем да запамти податке о задатку. (АПСО)

(Опис акције: Кликом на дугме “Sačuvaj“ позива се системка операција save(task))

1. Систем **памти** податке о задатку. (СО)
2. Систем **затвара** форму за унос задатка и администратору приказује форму за приказ свих задатака. (ИА)

A screenshot of a computer

Description automatically generated

Алтернативна сценарија

* 1. Уколико систем не може да запамти податке о задатку он приказује администатору поруку “Систем не може да запамти задатак”. (ИА)

A screenshot of a computer

Description automatically generated

### СК5 – Случај коришћења – Измена података о задатку

**Назив СК**

Измена података о задатку

**Актори СК**

Администратор

**Учесници СК**

Администратор и систем (програм)

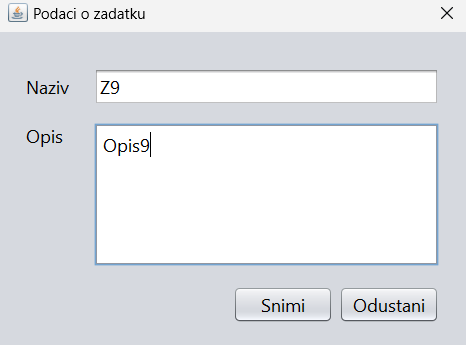
**Предуслов**: Систем је укључен и администратор je улогован под својим корисничким именом. Систем приказује форму за рад са задацима. Учитана је листа задатака.

**Основни сценарио СК**

1. Администратор **бира** задатак чије податке жели да измени. (АПСО)

(Опис акције: притиском на запосленог па затим на дугме “Izmeni” позива се ситемска операција TaskForm(task))

1. Систем **учитава** задатак.(СО)
2. Систем **приказује** администратору задатак и отвара форму за измену података. (ИА)



1. Администратор **уноси** **(мења)** податке о задакту. (АПУСО)

A screenshot of a computer

Description automatically generated

1. Амдинистратор **контролише** да ли је коректно унео податке о задаку. (АНСО)
2. Администратор **позива** систем да запамти податке о задатку. (АПСО)

(Опис акције: Кликом на дугме “Sačuvaj“ позива се системка операција save(task))

1. Систем **памти** податке о задацима. (СО)
2. Систем **затвара** форму за измену података и приказује администратору форму са свим задацима. (ИА)

A screenshot of a computer

Description automatically generated

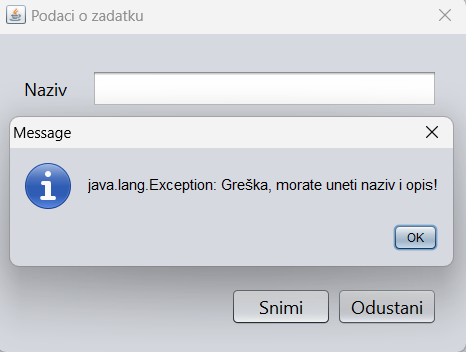
Алтернативна сценарија

* 1. Уколико систем не може да учита задатак он приказује адмнистратору поруку: “Систем не може да учита задатак”. Прекида се извршење сценарија. (ИА)

A screen shot of a computer

Description automatically generated

* 1. Уколико систем не може да запамти податке о задатку он приказује администатору поруку “Систем не може да запамти задатак”. (ИА)



### СК6: Случај коришћења – Унос новог пројекта

**Назив СК**

Унос новог пројекта

**Актори СК**

Адмнистратор

**Учесници СК**

Администратор и систем (програм)

**Предуслов**: Систем је укључен . Тип корисника који је улогован је адмнистратор. Систем приказује форму за унос пројеката.

A screenshot of a computer

Description automatically generated

**Основни сценарио СК**

1. Адмнистратор **уноси** податке о пројекту (назив пројекта, статус, додељује запосленог као вођу пројекта и креира задатке за пројекат (назив, одговорни запослени)). (АПУСО)

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Адмнистратор **контролише** да ли је коректно унео податке о пројекту. (АНСО)
2. Адмнистратор **позива** систем да запамти унете податке о пројекту. (АПСО)

(Опис акције: Кликом на дугме “Sačuvaj“ позива се системка операција save(project))

1. Систем **памти** податке о пројекту. (СО)
2. Систем **затвара** форму за унос пројеката иадмнистратору **приказује** форму на којој је листа свих пројеката. (ИА)

A screenshot of a computer

Description automatically generated

Алтернативна сценарија

5.1 Уколико систем не може да запамти пројекат он приказује адмнистратору поруку: “Систем не може да запамти пројекат”. (ИА)

A screenshot of a computer

Description automatically generated

### СК7: Случај коришћења – Измена података о пројекту

**Назив СК**

Измена података о пројекту

**Актори СК**

Адмнистратор или Запослени

**Учесници СК**

Адмнистратор и систем (програм)

**Предуслов**: Систем је укључен. Систем приказује форму за рад са пројектима. Учитана је листа пројеката.

A screenshot of a computer

Description automatically generated

**Основни сценарио СК**

1. Актор **бира** пројекат. (АПСО)
2. (Опис акције: притиском на запосленог па затим на дугме “Izmeni” позива се ситемска операција ProjectForm(project))
3. Систем **учитава** пројекат.(СО)
4. Систем **приказује** актору форму за измену пројекта. (ИА)

A screenshot of a computer

Description automatically generated

1. Актор **уноси (мења)** све или неке податке о пројекту (назив пројекта, статус, додељује запосленог као вођу пројекта и креира задатке за пројекат (одговорни запослени, задаци)). (АПУСО)

A screenshot of a computer

Description automatically generated

1. Актор **контролише** да ли је коректно унео податке о пројекту. (АНСО)
2. Актор **позива** систем да запамти податке о пројекту. (АПСО)

(Опис акције: Кликом на дугме “Sačuvaj“ позива се системка операција save(project))

1. Систем **памти** податке о пројекту. (СО)
2. Систем **затвара** форму за измену пројекта иактору **приказује** форму на којој је листа свих пројеката. (ИА)

A screenshot of a computer

Description automatically generated

Алтернативна сценарија

* 1. Уколико систем не може да учита пројекат он приказује актору поруку: “Систем не може да учита пројекат”. Прекида се извршење сценарија. (ИА)

A screen shot of a computer

Description automatically generated

8.1 Уколико систем не може да запамти податке о пројекту он приказује актору поруку “Систем не може да запамти пројекат”. (ИА)

A screenshot of a computer

Description automatically generated

### СК8: Случај коришћења – Увид и претрага пројеката

**Назив СК**

Увид и претрага пројеката и задатака

**Актори СК**

Запослени или Адмнистратор

**Учесници СК**

Актор и систем (програм)

**Предуслов**: Систем је укључен . Актори су улоговани. Систем приказује форму за пројекте. Учитана је листа пројеката.

A screenshot of a computer

Description automatically generated

**Основни сценарио СК**

1. Актор **уноси** вредности према којима претражује пројекат. (АПУСО)

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Актор **позива** систем да нађе пројекат по задатим вредностима. (АПСО)
2. (Опис акције:сваким избором неке вредности (активан, запослени, задатак, обрисани) позива се системска операција refresh())
3. Систем **тражи** пројекат по задатој вредности. (СО)
4. Систем приказује актору пројекте по задатим вредностима (ИА)

A screenshot of a computer

Description automatically generated

Алтернативна сценарија

* 1. Уколико систем не може да пронађе пројекат он приказује актору поруку “Систем не може да пронађе пројекат ”. (ИА)

A screen shot of a computer

Description automatically generated

## Пројектовање контролера графичког интерфејса (GUI Controller)

Контролер корисничког интерфејса (GUI контролер) је одговоран за следеће задатке:

* Прима податке које екранска форма шаље.
* Претвара податке графичких елемената у објекат који представља улаз у системску операцију која ће бити позвана.
* Шаље захтев за извршење до сервера апликације, који тај захтев даље обрађује.
* Прима објекат који представља излаз из софтверског система настао као резултат извршења системске операције.
* Претвара објекат у податке компатибилне са графичким интерфејсом, који као такви могу бити приказани.

## Пројектовање апликационе логике

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

* Контролера апликационе логике: Он формира серверски сокет који ослушкује мрежу и чека да се клијент повеже. Директно комуницира са клијентом. Улога му је да прими захтеве за извршење системских операција од клијента и проследи их пословној логици.
* Пословне логике: Овде се налази структура (доменске класе, апстрактне класе и интерфејси) и понашање (системске операције).
* Брокера базе података: Ова компонента служи за комуникацију између пословне логике и базе података. Она може се састојати од једне или више класа, интерфејса, апстрактних класа итд. (перзистентни оквир).

## Контролер апликационе логике

Део за комуникацију креира софтверски сокет који ослушкује мрежу и чека клијенте. Када сокет клијента успостави везу са сервером, тада сервер креира нит за рад са клијентом *(ClientThread*).

Целокупан софтверски систем је реализован коришћењем клијент-сервер архитектуре. На серверској страни је нит сервера (*ServerThread*) која садржи објекат класе *ServerSocket*. Ова нит је увек у могућности да прихвати клијента методом *serverSocket.accept()* чиме се креира нови објекат класе *Socket* и нова нит клијента на серверу(*ClientThread*). Када клијент пошаље захтев, одговарајућа клијентска нит (објекат класе *ClientThread*) прихвата захтев, тумачи га, и шаље даље у систем на извршавање. Након што се изврши операција, oдговарајућa нит за обраду клијентских захтева *(ClientThread*) шаље одговор назад клијенту.

## Пројектовање понашања софтверског система (Системске операције)

Сваки од претходно дефинисаних уговора дефинише једну системску операцију и ово представља понашање софтверског система. За сваку системску операцију је неопходно направити концептуална решења која су директно повезана са логиком проблема.

За сваки уговор пројектује се једно концептуално решење.

1. Уговор **UG1**: find

**Операција: find(username):** сигнал

**Веза са СК:** СК1

**Предуслови:** *Подаци о објекту Zaposleni морају да се слажу са одацима у бази*

**Постуслови:** *Корисник(Запослени) је пријављен на систем*

*A black screen with white text

Description automatically generated*

1. Уговор **UG2**: saveEmployee

**Операција:** **save(employee):** сигнал

**Веза са СК:** СК2, CK3

**Предуслови:** *Структурна и вредносна ограничења над Employee објектом морају бити задовољена*

**Постуслови:** *Сачувани су подаци о запосленом*

*A black screen with white text

Description automatically generated*

1. Уговор **UG3**: getListEmployees

**Операција: getList(<List>Employees):** сигнал

**Веза са СК:** СК3,CK6,CK7

**Предуслови:**

**Постуслови:** *Учитана је листа запослених*

*A black screen with white text

Description automatically generated*

1. Уговор **UG4**: EmployeeForm

**Операција:** **EmployeeForm(Employee):** сигнал

**Веза са СК:** СК3

**Предуслови:**

**Постуслови:** *Учитана је форма за рад са запосленим*

*A black screen with white text

Description automatically generated*

1. Уговор **UG5**: saveTask

**Операција: save(task):** сигнал

**Веза са СК:** СК4, CK5

**Предуслови:** *Структурна и вредносна ограничења над Task објектом морају бити задовољена*

**Постуслови:** *Сачувани су подаци о задатку*

*A black screen with white squares

Description automatically generated*

1. Уговор **UG6**: getListTask

**Операција: getList(<List>Tasks):** сигнал

**Веза са СК:** СК5, CK6, CK7

**Предуслови:**

**Постуслови:** *Учитана је листа задатака*

*A black screen with white squares

Description automatically generated*

1. Уговор **UG7**: TaskForm

**Операција:** **TaskForm(Task):** сигнал

**Веза са СК:** СК5

**Предуслови:**

**Постуслови:** *Учитана је форма за рад са задацима*

*A black screen with a white text

Description automatically generated*

1. Уговор **UG8**: saveProject

**Операција:** **save(project):** сигнал

**Веза са СК:** СК6, CK7

**Предуслови:** *Структурна и вредносна ограничења над Project објектом морају бити задовољена*

**Постуслови:** *Сачувани су подаци о пројекту*

*A black screen with white text

Description automatically generated*

1. Уговор **UG9**: getList

**Операција:** **getList(<List>Projects):** сигнал

**Веза са СК:** СК7,CK8

**Предуслови:**

**Постуслови:** *Учитана је листа пројеката*

*A black screen with white text

Description automatically generated*

1. Уговор **UG10**: ProjectForm

**Операција: ProjectForm(Project):** сигнал

**Веза са СК:** СК7

**Предуслови:**

**Постуслови:** *Учитана је форма за рад са пројектима*

*A black screen with white text

Description automatically generated*

1. Уговор **UG11**: getList2

**Операција:** **getList(<List>Projects, params):** сигнал

**Веза са СК:** СК8

**Предуслови:**

**Постуслови:** *Учитана је листа пројеката по задатим параметрима*

*A black screen with white text

Description automatically generated*

Све брокер класе наслеђују класу *DBBase* и *DBBroker,* које праве конекцију са базом и извршавају упите. Свака контролер класа преко класе *Communication* шаље захтеве и прима одговоре од сервера.

## Пројектовање структуре софтверског система (Доменске класе)

На основу концептуалног модела се креирају доменске класе.

A black background with white text

Description automatically generated

Индентификоване су следеће класе:

1. Класа *Project*

A screen shot of a computer program

Description automatically generated

Top of Form

1. Класа *Employee*

A screen shot of a computer code

Description automatically generated

1. Класа *Task*

A computer code with text

Description automatically generated

1. Класа *ProjectItem*

A computer screen shot of a program code

Description automatically generated

Све наведене класе такође имају имплементиране *get* и *set* методе као и *override toString* методе.

Поред ових, додате су и следеће класе:

1. Класа *ProjectSimple*

A close-up of a white background

Description automatically generated

1. Класа *Constants*, који одређује тип системске операције коју клијент жели да постигне, као и неке друге параметре

A screenshot of a computer program

Description automatically generated

1. Класа *ClientRequest*, која представља захтев корисника

A screen shot of a computer code

Description automatically generated

1. Класа *ServerResponse*, која представља серверски одговор на захтев корисника

A screen shot of a computer program

Description automatically generated

Све наведене класе, сем класе *Constants*, такође имају имплементиране *get* и *set* методе као и *override toString* методе.

## Пројектовање складиша података

На основу одрађеног релационог модела, и пропратних ограничења, пројектоване су табеле базе података са којима комуницира наш софтверски систем:

1. Табела zaposleni

A screenshot of a computer

Description automatically generated

1. Табела zadaci

A screenshot of a computer

Description automatically generated

1. Табела projekti

A screenshot of a computer

Description automatically generated

1. Табела projekatstavke

A screenshot of a computer

Description automatically generated

За комуникацију са базом података задужена је класа DBBroker која прави конекцију са базом, шање јој упите и прима одговоре. Такође постоји класа DBBase коју наслеђују калсе EmployeeBroker, TaskBroker, ProjectBroker, ProjectItemBroker које представљају брокере за све доменске класе које су горе наведене.

На основу претходних целина, може се саставити цела архитектура софтверског система.

A screenshot of a computer

Description automatically generated

# Имплементација

Oвај софтверски систем је реализован у програмском језику Јава, коришћењем интегрисаног развојног окружења NetBeans. Систем за управљање базом података је MySQL, а алат за рад са МySQL базом је SqlYog.

Пројекти из којих се састоји ова апликација су:

1. Server
2. Client
3. Common

Пројекат *Server* садржи 11 класа које се налазе у три пакета.

Први пакет је *controller* у ком се налазе класе *EmployeeControler, TaskController, ProjectController* које комуницирају са одговарајућим брокером базе података у зависности о ком доменском обкејту је реч.

Други пакет је *db* у ком се налазе класе *DBBase, DBBroker, EmployeeBroker, TaskBroker, ProjectBroker, ProjectItemBroker.* Класа *DBBroker* прави конекцију са базом и шаље упите које добија од осталих брокер класа. Класу *DBBroker* наслеђују класе *EmployeeBroker, TaskBroker, ProjectBroker, ProjectItemBroker* и она служи за прављење метода које ће све ове класе користити за припремање упита и добијања података из базе. Класе *EmployeeBroker, TaskBroker, ProjectBroker, ProjectItemBroker* припремају упите везане за доменски обкејат који је у њиховом називу.

Трећи пакет је *threads* који садржи калсе *ServerThread, ClientThread. ServerThread* покреће сервер и ослушкује да ли постоји неки клијент. Када прихвати клијента креира нову нит *ClientThread* која наставља рад са клијентом, прихвата захтеве, шаље их дање у систем на извршавање а затим прослеђује назад системски одговор.

Пројекат *Client* садржи 19 класа које се налазе у два пакета и један додатни пакет у ком се налазе три слике.

Први пакет је *controller* и садржи класе *EmployeeController*, *TaskController*, *ProjectController*. Свака од ових класа користећи инстанцу калсе *Communication(*која се истоа налази у овом пакету*)* комуницира са сервером и шање корисничке захтеве.

Други пакет је *forms* и он садржи све класе форми и све моделе табела који су потребни да би се приказао кориснички интерфејс.

Трећи пакет *pictures* садржи три слике које се налазе као иконице на главној форми програма(*MainForm*).

Пројекат *Common* садржи 8 каласа које се налазе у два пакета.

Први пакет је *domen* који садржи све доменске објекте и класу за константе које се користе у програму.

Други пакет, *transfer,* садржи две класе преко којих клијент и сервер комуницирају. Једна је клијентски захтев а друга серверски одговор.

# Тестирање

У фази тестирања, тестиран је сваки од имплементираних случајева коришћења. Приликом тестирања сваког случаја коришћења, поред унетих правилних података, уношени су и неправилни подаци да би се утврдио резултат извршења. Након фазе тестирања, софтвер је спреман за коришћење од стране крајњег корисника.

# Закључак

Тренутно развијено софтверско решење јесте применљиво, али такође оставља пуно простора, да се коришћењем савремених технологија, побољшају и отклоне потенцијални недостаци, као и простора да се надограде нове функционалности које би задовољиле потребе корисника, пруживши му већу употребну вредност.

# Литература

”ПРОЈЕКТОВАЊЕ СОФТВЕРА СКРИПТА- радни материјал, вер. 1.3”, др Синиша Влајић, Београд 2020