

УНИВЕРЗИТЕТ У БЕОГРАДУ  
ФАКУЛТЕТ ОРГАНИЗАЦИОНИХ НАУКА, ЛАБАРАТОРИЈА ЗА  
СОФТВЕРСКО ИНЖЕЊЕРСТВО

СЕМИНАРСКИ РАД

**ТЕМА: Софтверски систем за рад кладионице  
у Јава окружењу**

Ментор:  
др. Душан Савић

Студент:  
Драгољуб Ђурић, 2015/0124

Београд, 2019. године



# Садржај

Кориснички захтеви.....	10
Вербални опис.....	10
Случајеви коришћења.....	11
СК1: Случај коришћења – Пријава на систем.....	12
СК2: Случај коришћења – Уплата новог тикета.....	13
СК3: Случај коришћења – Претрага тикета.....	14
СК4: Случај коришћења – Брисање тикета.....	15
СК5: Случај коришћења – Исплата тикета.....	16
СК6: Случај коришћења – Креирање нове утакмице.....	17
СК7: Случај коришћења – Унос коначног резултата утакмице.....	18
СК8: Случај коришћења – Претрага пословних резултата.....	19
Анализа.....	20
Понашање софтверског система - Системски дијаграм секвенци.....	20
ДС1: Дијаграм секвенци случаја коришћења – Пријава на систем.....	20
ДС2: Дијаграм секвенци случаја коришћења – Уплата новог тикета.....	21
ДС3: Дијаграм секвенци случаја коришћења – Претрага тикета.....	22
ДС4: Дијаграм секвенци случаја коришћења – Брисање тикета.....	23
ДС5: Дијаграм секвенци случаја коришћења – Исплата тикета.....	25
ДС6: Дијаграм секвенци случаја коришћења – Креирање нове утакмице.....	27
ДС7: Дијаграм секвенци случаја коришћења – Унос коначног резултата утакмице.....	29
ДС8: Дијаграм секвенци случаја коришћења – Претрага пословних резултата.....	31
Понашање софтверског система - Дефинисање уговора о системским операцијама.....	33
Структура софтверског система – Концептуални(доменски) модел.....	36
Структура софтверског система – Релациони модел.....	37
Пројектовање.....	41
Архитектура софтверског система.....	41
Пројектовање корисничког интерфејса.....	41
Пројектовање екранских форми.....	42
СК1: Случај коришћења – Пријава на систем.....	43
СК2: Случај коришћења – Уплата новог тикета.....	45
СК3: Случај коришћења – Претрага тикета.....	47
СК4: Случај коришћења – Брисање тикета.....	49
СК5: Случај коришћења – Исплата тикета.....	52

СК6: Случај коришћења – Креирање нове утакмице.....	55
СК7: Случај коришћења – Унос коначног резултата утакмице.....	57
СК8: Случај коришћења – Претрага пословних резултата.....	61
Пројектовање контролера корисничког интерфејса.....	63
Пројектовање апликационе логике.....	64
Пројектовање контролера апликационе логике.....	64
Пројектовање пословне логике.....	64
Пројектовање понашања софтверског система – системске операције.....	64
Пројектовање структуре софтверског система – доменске класе.....	72
Пројектовање брокера базе података.....	73
Пројектовање складишта података.....	75
Коначна архитектура.....	76
Имплементација.....	77
Тестирање.....	78
Литература.....	79

## **Попис слика**

Слика 1: Концептуални модел софтверског система.....	36
Слика 2: Тронивојска архитектура.....	41
Слика 3: Структура корисничког интерфејса.....	41
Слика 4: Приказ форме за покретање сервера.....	42
Слика 5: Приказ форме за конектовање на сервер.....	42
Слика 6: Приказ форме за пријављивање на систем.....	43
Слика 7: Приказ поруке о успешном пријављивању.....	44
Слика 8: Приказ поруке о неуспешном пријављивању.....	44
Слика 9: Приказ форме за рад са тикетима - Уплата новог тикета.....	45
Слика 10: Приказ поруке о успешној уплати тикета.....	46
Слика 11: Приказ поруке о неуспешној уплати тикета.....	46
Слика 12: Приказ форме за рад са тикетима - Претрага тикета.....	47
Слика 13: Приказ форме након успешне претраге тикета.....	48
Слика 14: Приказ поруке о неуспешној претрази тикета.....	48
Слика 15: Приказ форме за рад са тикетима - Брисање тикета.....	49
Слика 16: Приказ форме након успешне претраге тикета.....	50
Слика 17: Приказ поруке о успешном брисању тикета.....	50
Слика 18: Приказ поруке о неуспешној претрази тикета.....	51
Слика 19: Приказ поруке о неуспешном брисању тикета.....	51
Слика 20: Приказ форме за рад са тикетима - Исплата тикета.....	52
Слика 21: Приказ форме након успешне претраге тикета.....	53
Слика 22: Приказ поруке о успешном брисању тикета.....	53
Слика 23: Приказ поруке о неуспешној претрази тикета.....	54
Слика 24: Приказ поруке о неуспешној исплати тикета.....	54
Слика 25: Приказ форме за рад са утакмицама - Креирање нове утакмице.....	55
Слика 26: Приказ поруке о успешном креирању утакмице.....	56
Слика 27: Приказ поруке о неуспешном креирању утакмице.....	56

Слика 28: Приказ форме за рад са утакмицама - Унос коначног резултата утакмице.....	57
Слика 29: Приказ форме након успешне претраге утакмице.....	58
Слика 30: Приказ поруке о успешном уносу резултата.....	59
Слика 31: Приказ поруке о неуспешној претрази утакмице.....	59
Слика 32: Приказ поруке о неуспешном уносу резултата.....	60
Слика 33: Приказ форме за рад са пословним резултатима.....	61
Слика 34: Приказ форме након успешне претраге пословних резултата.....	62
Слика 35: Приказ поруке о неуспешној претрази пословних резултата.....	62
Слика 36: Понашање софтверског система.....	65
Слика 37: Доменске класе.....	72
Слика 38: Перзистентни оквир.....	73
Слика 39: Структура софтверског система.....	74
Слика 40: Структура складишта података.....	75
Слика 41: Коначна архитектура софтверског система.....	76

## **Попис дијаграма**

Дијаграм 1: Случајеви коришћења.....	11
Дијаграм 2: DC1 - основни сценарио.....	20
Дијаграм 3: DC1 - алтернативни сценарио.....	20
Дијаграм 4: DC2 - основни сценарио.....	21
Дијаграм 5: DC2 - алтернативни сценарио.....	21
Дијаграм 6: DC3 - основни сценарио.....	22
Дијаграм 7: DC3 - алтернативни сценарио.....	22
Дијаграм 8: DC4 - основни сценарио.....	23
Дијаграм 9: DC4 - алтернативни сценарио.....	23
Дијаграм 10: DC4 - алтернативни сценарио.....	24
Дијаграм 11: DC5 - основни сценарио.....	25
Дијаграм 12: DC5 - алтернативни сценарио.....	25
Дијаграм 13: DC5 - алтернативни сценарио.....	26
Дијаграм 14: DC6 - основни сценарио.....	27
Дијаграм 15: DC6 - алтернативни сценарио.....	27
Дијаграм 16: DC7 - основни сценарио.....	29
Дијаграм 17: DC7 - алтернативни сценарио.....	29
Дијаграм 18: DC7 - алтернативни сценарио.....	30
Дијаграм 19: DC8 - основни сценарио.....	31
Дијаграм 20: DC8 - алтернативни сценарио.....	31
Дијаграм 21: УГ1: ПријаваНаСистем.....	66
Дијаграм 22: УГ2: ВратиСвеТипове.....	66
Дијаграм 23: УГ3: СачувајТикет.....	67
Дијаграм 24: УГ4: ВратиТикет.....	68
Дијаграм 25: УГ5: ОбришиТикет.....	68
Дијаграм 26: УГ6: ИсплатиТикет.....	69

Дијаграм 27: УГ7: ВратиСвеТимове.....	69
Дијаграм 28: УГ8: КреирајМеч.....	70
Дијаграм 29: УГ9: ВратиМечЗаУносРезултата.....	70
Дијаграм 30: УГ10: УносРезултата.....	71
Дијаграм 31: УГ11: ВратиРезултатеПословања.....	71

## **Попис таблица**

Табела 1: Табела User.....	37
Табела 2: Табела Team.....	37
Табела 3: Табела Match.....	38
Табела 4: Табела Tip.....	38
Табела 5: Табела Odds.....	39
Табела 6: Табела Ticket.....	39
Табела 7: Табела Has.....	40

## Кориснички захтеви

### Вербални опис

Потребно је креирати софтверски систем који ће омогућити запосленима Моцарт кладионице да лакше обављају свој посао. Корисници овог система су само запослени и пре почетка рада неопходно је њихово пријављивање на систем.

Систем води евиденцију о свим уплаћеним тикетима и пружа функционалности као што су: уплата тикета, претрага тикета, брисање тикета, исплата тикета. Сваки тикет садржи податке као што су: ID, уплаћену количина новца, укупна квота, могући добитак, листа одиграних утакмица са типовима и статус.

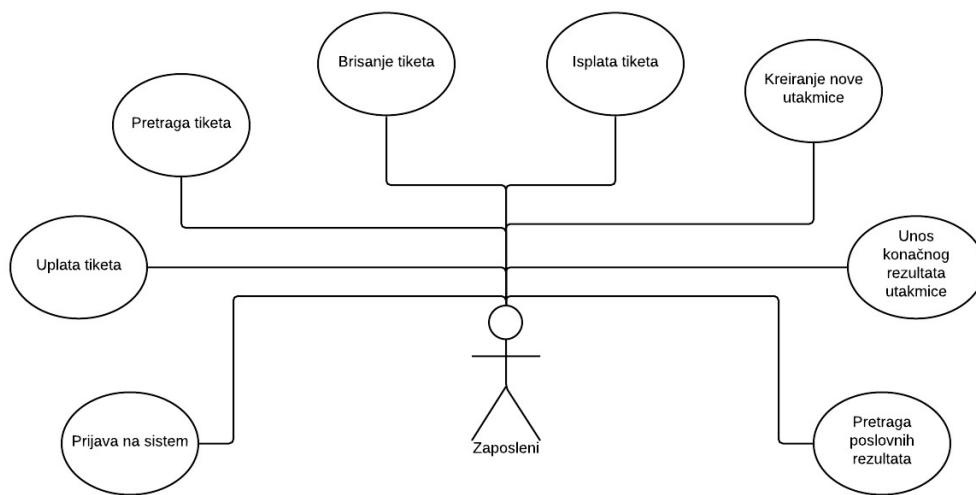
Такође потребно је да систем омогући додавање нове утакмице, уношење коначног резултата утакмице. Свака утакмица садржи следеће податке: ID, време почетка, домаћи тим, гостујући тим, коначни резултат.

Овим системом, сваком запосленом је омогућено да лакше води евиденцију о пословању кладионице.

## Случајеви коришћења

У овом систему постоје следећи случајеви коришћења:

- пријава на систем,
- уплата тикета(сложен),
- претрага тикета,
- брисање тикета,
- исплата тикета,
- креирање нове утакмице,
- унос коначног резултата утакмице,
- претрага пословних резултата



Дијаграм 1: Случајеви коришћења

## **СК1: Случај коришћења – Пријава на систем**

### **Назив СК**

Пријава на систем

### **Актори СК**

Запослени

### **Учесници СК**

Запослени и систем (програм)

**Предуслов:** Систем је укључен. Систем приказује форму за пријаву на систем.

### **Основни сценарио СК**

1. Запослени **уноси** корисничко име и лозинку. (АПУСО)
2. Запослени **контролише** да ли је коректно унео корисничко име и лозинку. (АНСО)
3. Запослени **позива** систем да се пријави на систем (провери податке). (АПСО)
4. Систем **проверава** податке о запосленом. (СО)
5. Систем **приказује** запосленом поруку: “Login successful! “. (ИА)

### **Алтернативна сценарија**

- 5.1. Уколико систем не може да нађе запосленог он приказује запосленом поруку: “Login unsuccessful!”. (ИА)

## **СК2: Случај коришћења – Уплата новог тикета**

### **Назив СК**

Уплата новог тикета

### **Актори СК**

Запослени

### **Учесници СК**

Запослени и систем (програм)

**Предуслов:** Систем је укључен и запослени је улогован под својом шифром. Систем приказује форму за рад са тикетима. Учитана је листа типова.

### **Основни сценарио СК**

1. Запослени **уноси** податке у нови тикет. (АПУСО)
2. Запослени **контролише** да ли је коректно унео податке. (АНСО)
3. Запослени **позива** систем да запамти податке о тикету. (АПСО)
4. Систем **памти** податке о тикету. (СО)
5. Систем **приказује** запосленом поруку: “Ticket is successfully saved!”. (ИА)

### **Алтернативна сценарија**

- 5.1. Уколико систем не може да запамти податке о тикету он приказује запосленом поруку: “Ticket is not saved!”. (ИА)

### **СК3: Случај коришћења – Претрага тикета**

#### **Назив СК**

Претрага тикета

#### **Актори СК**

Запослени

#### **Учесници СК**

Запослени и систем (програм)

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

#### **Основни сценарио СК**

1. Запослени **уноси** вредност по којој претражује тикет. (АПУСО)
2. Запослени **позива** систем да нађе тикет по задатој вредности. (АПСО)
3. Систем **тражи** тикет по задатој вредности. (СО)
4. Систем **приказује** запосленом податке о тикету. (ИА)

#### **Алтернативна сценарија**

- 4.1. Уколико систем не може да пронађе тикет он приказује запосленом поруку: “Ticket is not found!”.(ИА)

## **СК4: Случај коришћења – Брисање тикета**

### **Назив СК**

Брисање тикета

### **Актори СК**

Запослени

### **Учесници СК**

Запослени и систем (програм)

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

### **Основни сценарио СК**

1. Запослени **уноси** вредност по којој претражује тикет. (АПУСО)
2. Запослени **позива** систем да нађе тикет по задатој вредности. (АПСО)
3. Систем **тражи** тикет по задатој вредности. (СО)
4. Систем **приказује** запосленом тикет.(ИА)
5. Запослени **позива** систем да обрише тикет.(АПСО)
6. Систем **брисе** тикет. (СО)
7. Систем **приказује** запосленом поруку: “Ticket is successfully deleted!”. (ИА)

### **Алтернативна сценарија**

- 4.1. Уколико систем не може да нађе тикет он приказује запосленом поруку: “Ticket is not found!”. Прекида се извршавање сценарија. (ИА)
- 7.1. Уколико систем не може да обрише тикет он запосленом приказује поруку: “Ticket is not deleted!”. (ИА)

## **СК5: Случај коришћења – Исплата тикета**

### **Назив СК**

Исплата тикета

### **Актори СК**

Запослени

### **Учесници СК**

Запослени и систем (програм)

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

### **Основни сценарио СК**

1. Запослени **уноси** вредност по којој претражује тикет. (АПУСО)
2. Запослени **позива** систем да нађе тикет по задатој вредности. (АПСО)
3. Систем **тражи** тикет по задатој вредности. (СО)
4. Систем **приказује** запосленом тикет.(ИА)
5. Запослени **позива** систем да исплати тикет.(АПСО)
6. Систем **исплаћује** тикет. (СО)
7. Систем **приказује** запосленом поруку: “Ticket is successfully payed!”. (ИА)

### **Алтернативна сценарија**

- 4.1. Уколико систем не може да нађе тикет он приказује запосленом поруку: “Ticket is not found!”. Прекида се извршавање сценарија. (ИА)
- 7.1. Уколико систем не може да исплати тикет он запосленом приказује поруку: “Ticket is not payed!”. (ИА)

## **СК6: Случај коришћења – Креирање нове утакмице**

### **Назив СК**

Унос нове утакмице

### **Актори СК**

Запослени

### **Учесници СК**

Запослени и систем (програм)

**Предуслов:** Систем је укључен и запослени је улогован под својом шифром. Систем приказује форму за рад са утакмицама. Учитана је листа тимова.

### **Основни сценарио СК**

1. Запослени **уноси** податке о утакмици. (АПУСО)
2. Запослени **контролише** да ли је коректно унео податке о утакмици. (АНСО)
3. Запослени **позива** систем да запамти податке о утакмици. (АПСО)
4. Систем **памти** податке о утакмици. (СО)
5. Систем **приказује** запосленом поруку: “Match is successfully created! ID: “ и ID новокреиране утакмице. (ИА)

### **Алтернативна сценарија**

- 5.1. Уколико систем не може да запамти податке о утакмици он приказује запосленом поруку: “Match is not created!”.(ИА)

## **СК7: Случај коришћења – Унос коначног резултата утакмице**

### **Назив СК**

Унос коначног резултата утакмице

### **Актори СК**

Запослени

### **Учесници СК**

Запослени и систем (програм)

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

### **Основни сценарио СК**

1. Запослени **уноси** вредност по којој претражује утакмицу. (АПУСО)
2. Запослени **позива** систем да нађе утакмицу по задатој вредности. (АПСО)
3. Систем **тражи** утакмицу по задатој вредности. (СО)
4. Систем **приказује** запосленом утакмицу.(ИА)
5. Запослени **уноси** уноси податке о утакмици. (АПУСО)
6. Запослени **контролише** да ли је коректно унео податке. (АНСО)
7. Запослени **позива** систем да запамти податке.(АПСО)
8. Систем **памти** податке. (СО)
9. Систем **приказује** запосленом поруку: “Score is successfully inserted!”. (ИА)

### **Алтернативна сценарија**

- 4.1. Уколико систем не може да нађе утакмицу он приказује запосленом поруку: “Score is cannot be inserted!”. Прекида се извршавање сценарија. (ИА)
- 9.1. Уколико систем не може да запамти податке он приказује запослениу поруку: “Score is not inserted! ”. (ИА)

## **СК8: Случај коришћења – Претрага пословних резултата**

Претраживање пословних резултата

### **Актори СК**

Запослени

### **Учесници СК**

Запослени и систем (програм)

**Предуслов:** Систем је укључен и запослени је улогован под својом шифром. Систем приказује форму за рад са пословним резултатима.

### **Основни сценарио СК**

1. Запослени **уноси** вредност по којој претражује пословне резултате. (АПУСО)
2. Запослени **позива** систем да нађе пословне резултате по задатој вредности. (АПСО)
3. Систем **тражи** податке о пословним резултатима. (СО)
4. Систем **приказује** запосленом податке о пословним резултатима. (ИА)

### **Алтернативна сценарија**

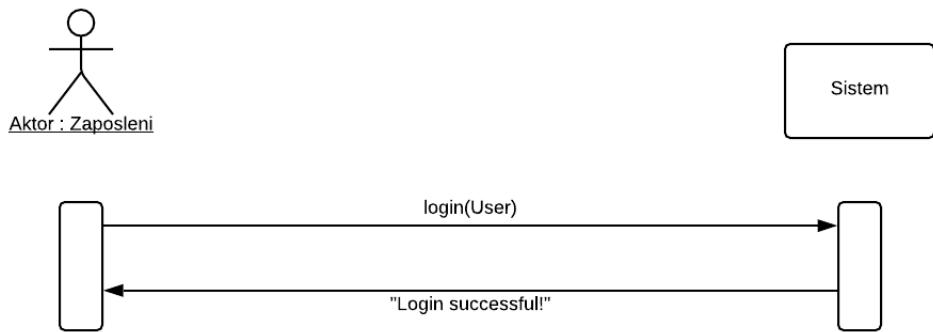
- 4.1. Уколико систем не може да нађе податке о пословним резултатима систем приказује запосленом поруку: “Business results are not found!”. (ИА)

## Анализа

### Понашање софтверског система - Системски дијаграм секвенци

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

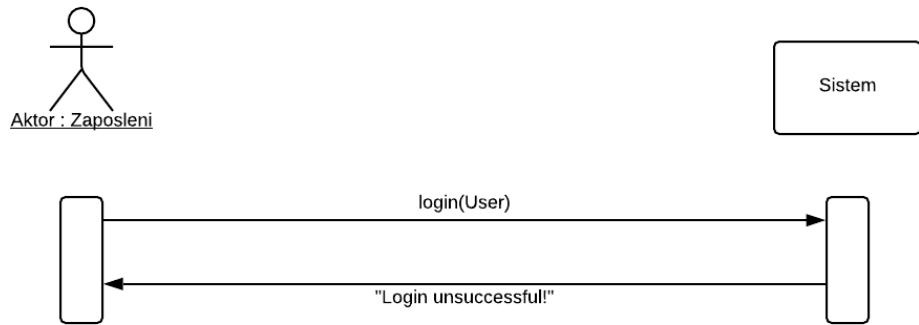
1. Запослени **позива** систем да се пријави на систем (провери податке). (АПСО)
2. Систем **приказује** запосленом поруку: "Login successful! ". (ИА)



Дијаграм 2: ДС1 - основни сценарио

#### Алтернативна сценарија

- 2.1. Уколико систем не може да нађе запосленог он приказује запосленом поруку: "Login unsuccessful!". (ИА)



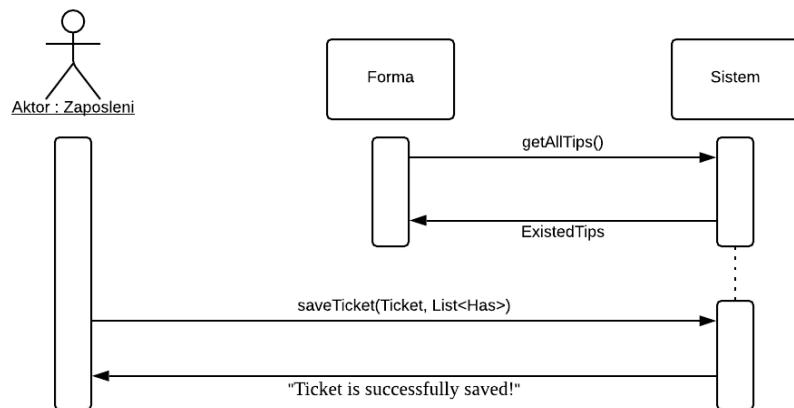
Дијаграм 3: ДС1 - алтернативни сценарио

Са наведених секвенцних дијаграма уочава се једна системска операција коју треба пројектовати:

1. signal login(User);

## ДС2: Дијаграм секвенци случаја коришћења – Уплата новог тикета

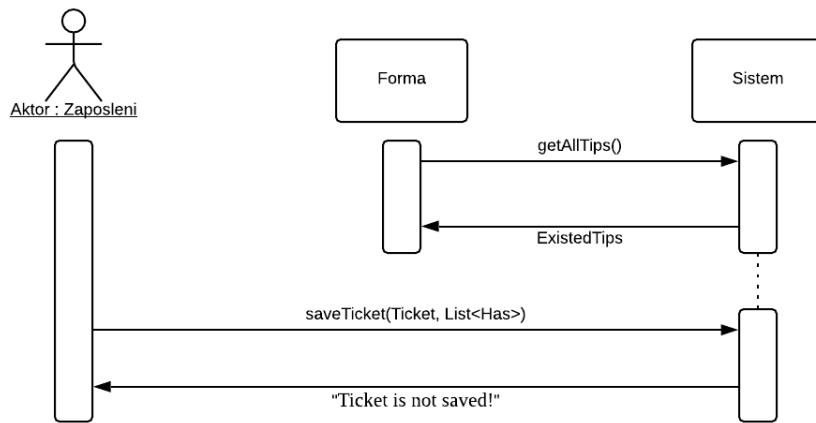
1. Запослени **позива** систем да запамти податке о тикету. (АПСО)
2. Систем **приказује** запосленом поруку: "Ticket is successfully saved!". (ИА)



Дијаграм 4: ДС2 - основни сценарио

### Алтернативна сценарија

- 2.1. Уколико систем не може да запамти податке о тикету он приказује запосленом поруку: "Ticket is not saved!". (ИА)



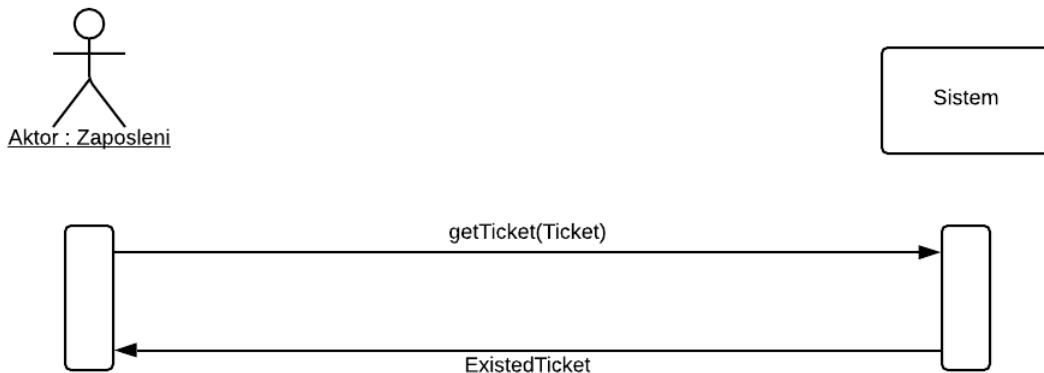
Дијаграм 5: ДС2 - алтернативни сценарио

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

1. signal `getAllTips();`
2. signal `saveTicket(Ticket, List<Has>);`

### ДСЗ: Дијаграм секвенци случаја коришћења – Претрага тикета

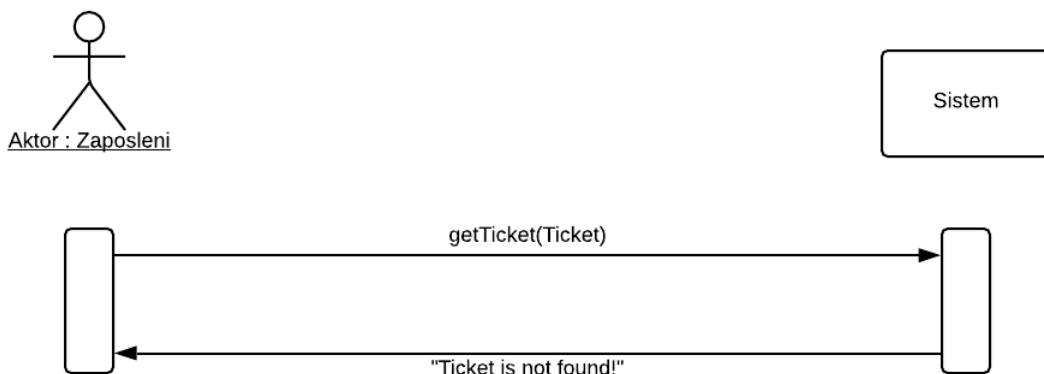
1. Запослени **позива** систем да нађе тикет по задатој вредности. (АПСО)
2. Систем **приказује** запосленом податке о тикету. (ИА)



Дијаграм 6: ДСЗ - основни сценарио

#### Алтернативна сценарија

- 2.1. Уколико систем не може да пронађе тикет он приказује запосленом поруку: "Ticket is not found!".(ИА)



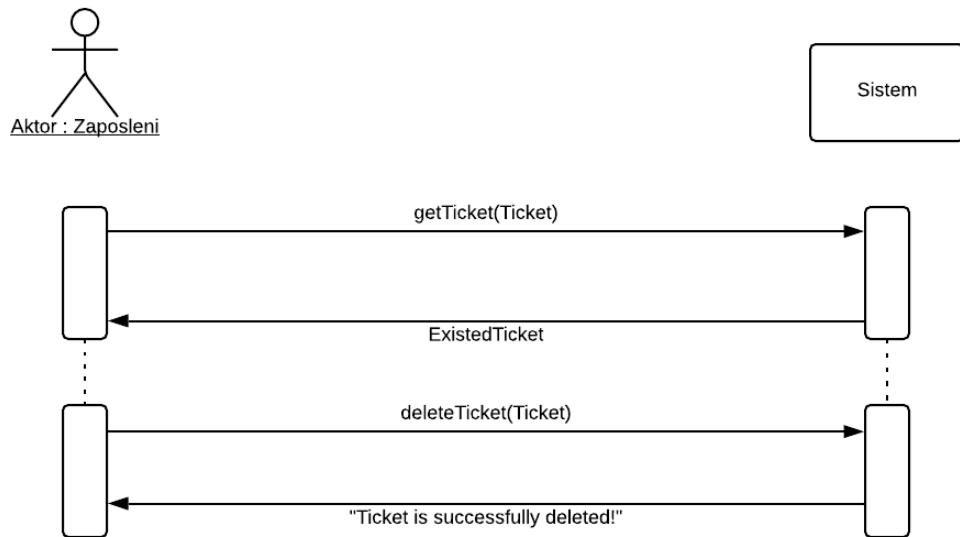
Дијаграм 7: ДСЗ - алтернативни сценарио

Са наведених секвенцних дијаграма уочава се једна системска операција коју треба пројектовати:

1. signal `getTicket(Ticket);`

#### ДС4: Дијаграм секвенци случаја коришћења – Брисање тикета

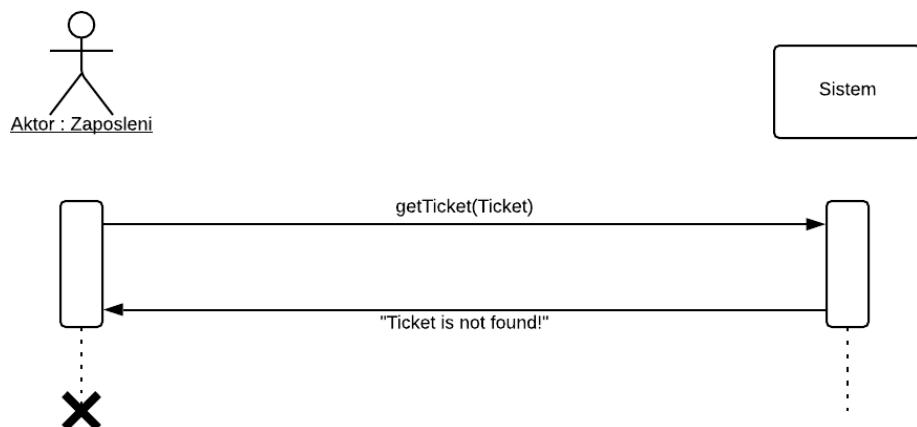
1. Запослени **позива** систем да нађе тикет по задатој вредности. (АПСО)
2. Систем **приказује** запосленом тикет.(ИА)
3. Запослени **позива** систем да обрише тикет.(АПСО)
4. Систем **приказује** запосленом поруку: "Ticket is successfully deleted!". (ИА)



Дијаграм 8: ДС4 - основни сценарио

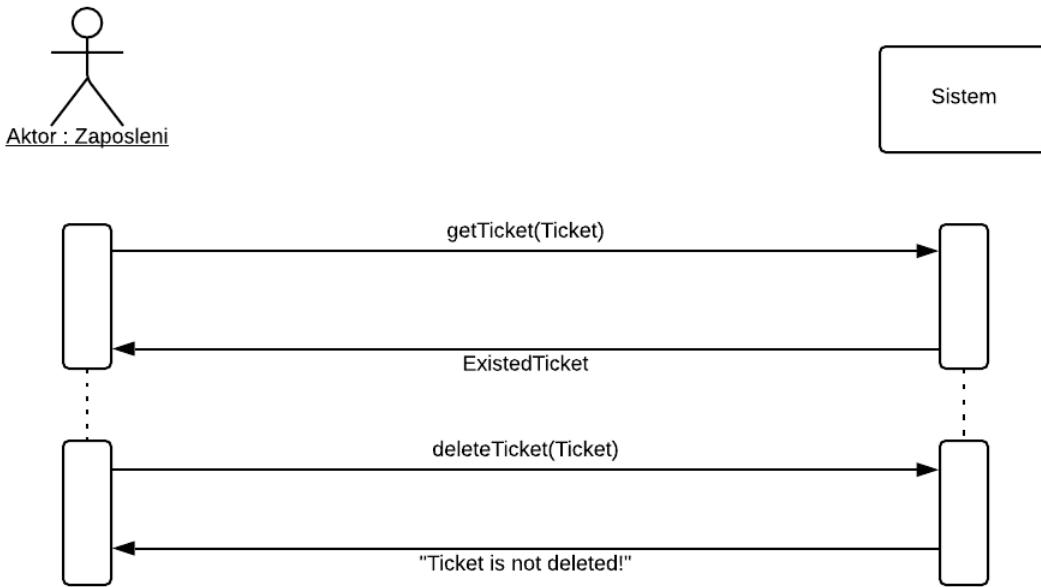
#### Алтернативна сценарија

- 2.1. Уколико систем не може да нађе тикет он приказује запосленом поруку: "Ticket is not found!". Прекида се извршавање сценарија. (ИА)



Дијаграм 9: ДС4 - алтернативни сценарио

4.1. Уколико систем не може да обрише тикет он запосленом приказује поруку: "Ticket is not deleted!". (ИА)



Дијаграм 10: ДС4 - алтернативни сценарио

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

1. signal `getTicket(Ticket)`;
2. signal `deleteTicket(Ticket)`;

### ДС5: Дијаграм секвенци случаја коришћења – Исплата тикета

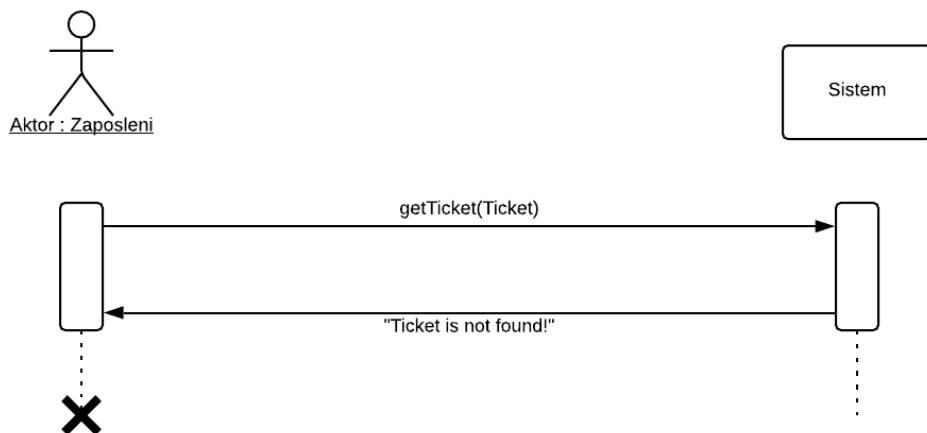
1. Запослени **позива** систем да нађе тикет по задатој вредности. (АПСО)
2. Систем **приказује** запосленом тикет.(ИА)
3. Запослени **позива** систем да исплати тикет.(АПСО)
4. Систем **приказује** запосленом поруку: "Ticket is successfully payed!". (ИА)



Дијаграм 11: ДС5 - основни сценарио

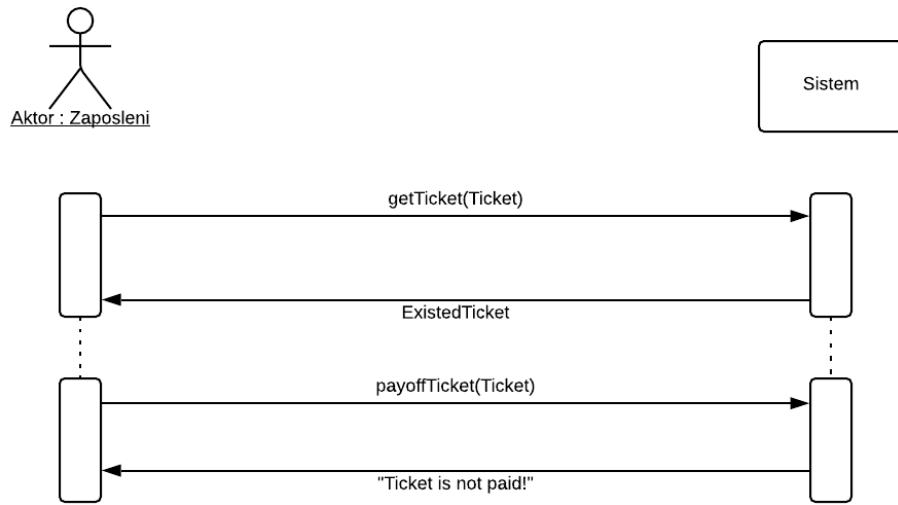
### Алтернативна сценарија

- 2.1. Уколико систем не може да нађе тикет он приказује запосленом поруку: "Ticket is not found!". Прекида се извршавање сценарија. (ИА)



Дијаграм 12: ДС5 - алтернативни сценарио

4.1. Уколико систем не може да исплати тикет он запосленом приказује поруку: "Ticket is not payed!". (ИА)



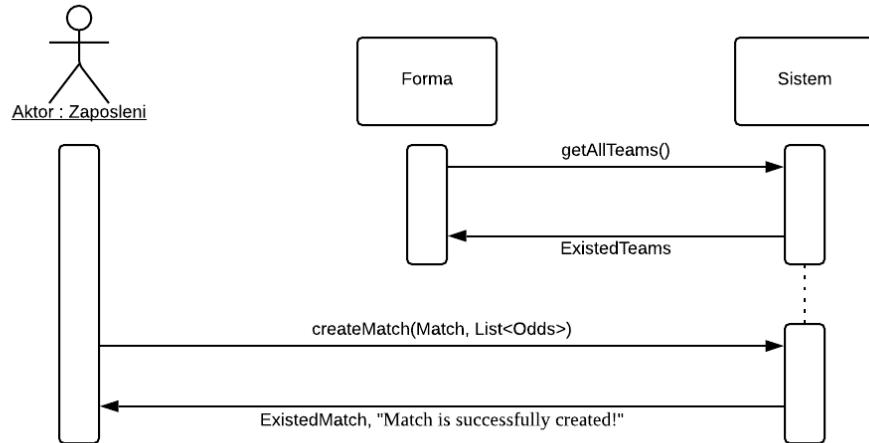
Дијаграм 13: ДС5 - алтернативни сценарио

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

1. signal `getTicket(Ticket)`;
2. signal `payoffTicket(Ticket)`;

## ДС6: Дијаграм секвенци случаја коришћења – Креирање нове утакмице

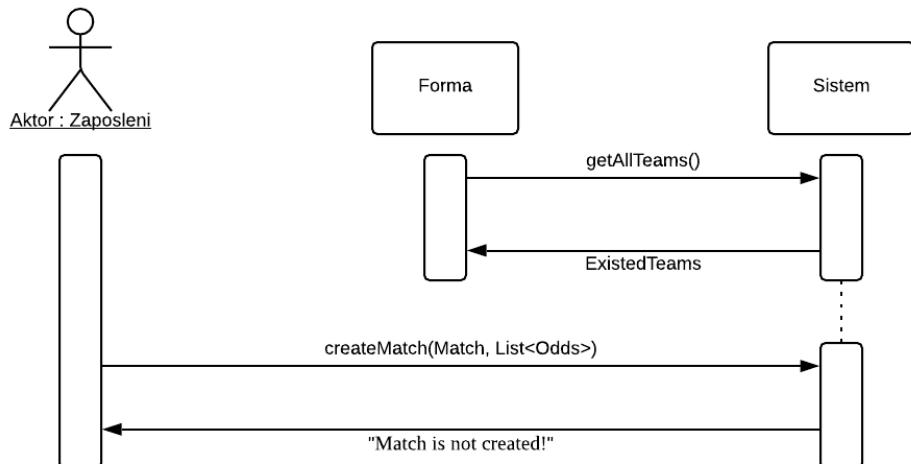
1. Запослени **позива** систем да запамти податке о утакмици. (АПСО)
2. Систем **приказује** запослену поруку: "Match is successfully created! ID: " и ID новокреиране утакмице. (ИА)



Дијаграм 14: ДС6 - основни сценарио

### Алтернативна сценарија

- 2.1. Уколико систем не може да запамти податке о утакмици он приказује запослену поруку: "Match is not created!".(ИА)



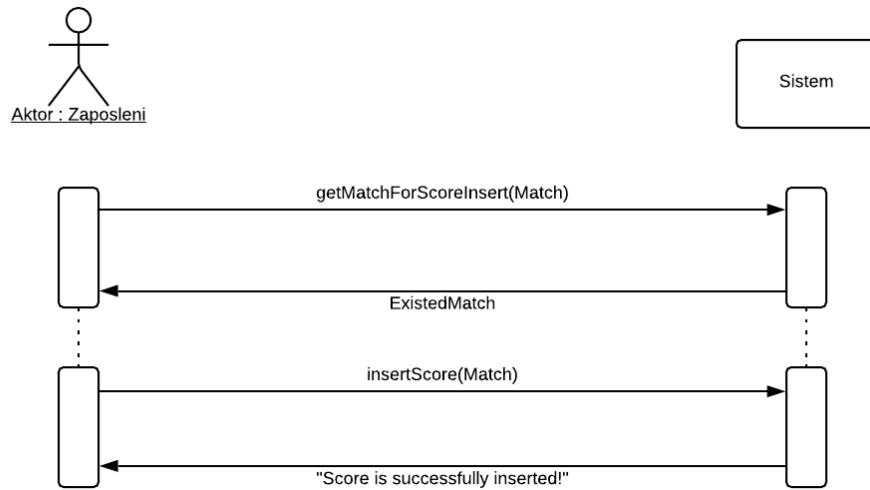
Дијаграм 15: ДС6 - алтернативни сценарио

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

1. signal getAllTeams();
2. signal createMatch(Match, List<Odds>);

## ДС7: Дијаграм секвенци случаја коришћења – Унос коначног резултата утакмице

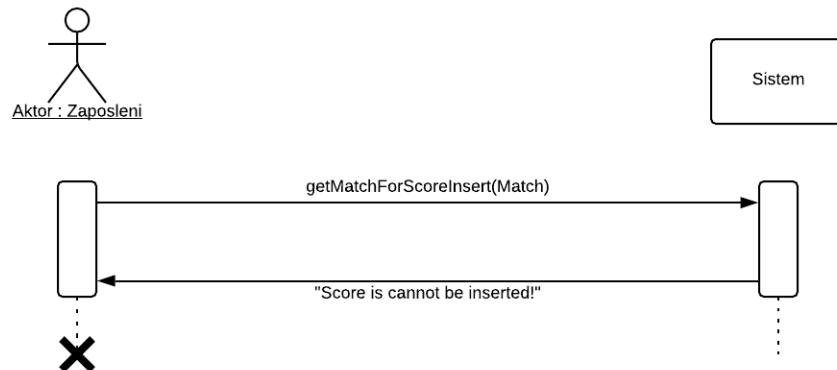
1. Запослени **позива** систем да нађе утакмицу по задатој вредности. (АПСО)
2. Систем **приказује** запосленом утакмицу.(ИА)
3. Запослени **позива** систем да запамти податке.(АПСО)
4. Систем **приказује** запосленом поруку: "Score is successfully inserted!". (ИА)



Дијаграм 16: ДС7 - основни сценарио

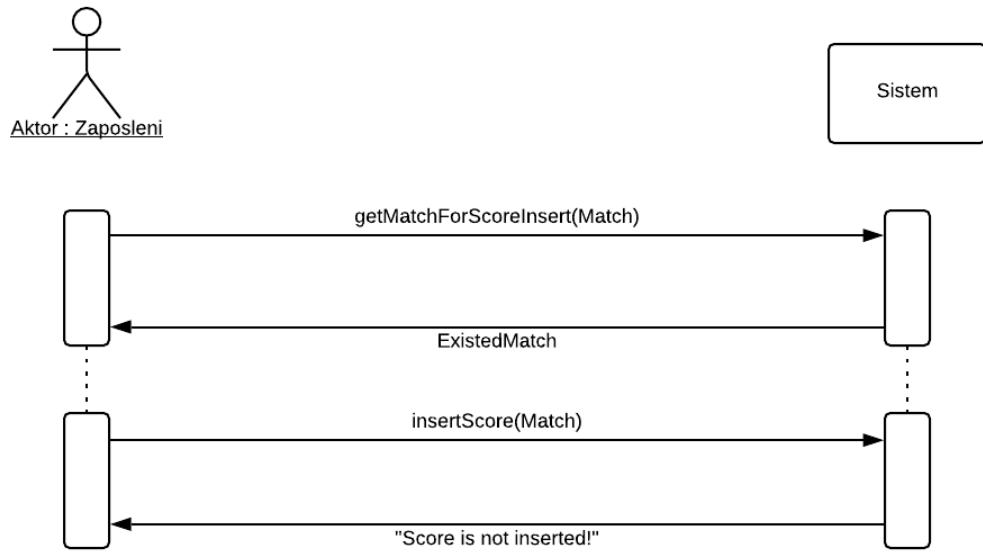
### Алтернативна сценарија

- 2.1. Уколико систем не може да нађе утакмицу он приказује запосленом поруку: "Score is cannot be inserted!". Прекида се извршавање сценарија. (ИА)



Дијаграм 17: ДС7 - алтернативни сценарио

4.1. Уколико систем не може да запамти податке он приказује запослениу поруку: "Score is not inserted! ". (ИА)



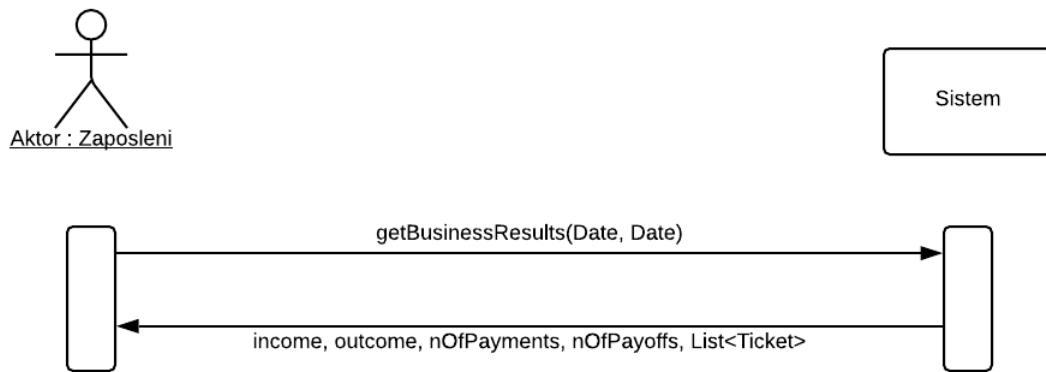
Дијаграм 18: ДС7 - алтернативни сценарио

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

1. signal getMatchForScoreInsert(Match);
2. signal insertScore(Match);

## ДС8: Дијаграм секвенци случаја коришћења – Претрага пословних резултата

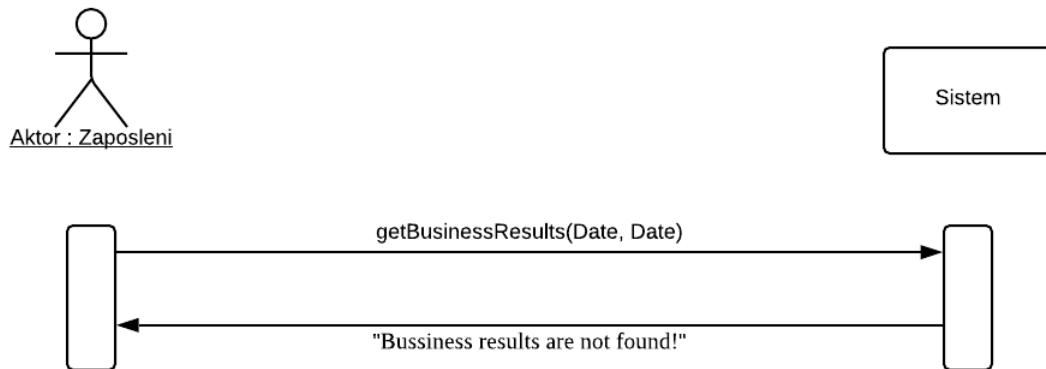
1. Запослени **позива** систем да нађе пословне резултате по задатој вредности. (АПСО)
2. Систем **приказује** запосленом податке о пословним резултатима. (ИА)



Дијаграм 19: ДС8 - основни сценарио

### Алтернативна сценарија

- 4.1. Уколико систем не може да нађе податке о пословним резултатима систем приказује запосленом поруку: “Bussiness results are not found!”. (ИА)



Дијаграм 20: ДС8 - алтернативни сценарио

Са наведених секвенцних дијаграма уочава се једна системска операција коју треба пројектовати:

1. signal getBusinessResults(Date, Date);

Како резултат анализе сценарија добијено је укупно 11 системских операција које треба пројектовати:

1. signal LoginSO(User);
2. signal GetAllTipsSO();
3. signal SaveTicketSO(Ticket, List<Has>);
4. signal GetTicketSO(Ticket);
5. signal DeleteTicketSO(Ticket);
6. signal PayoffTicketSO(Ticket);
7. signal GetAllTeamsSO();
8. signal CreateMatchSO(Match, List<Odds>);
9. signal GetMatchForScoreInsertSO(Match);
10. signal InsertScoreSO(Match);
11. signal GetBusinessResultsSO(Date, Date);

## Понашање софтверског система - Дефинисање уговора о системским операцијама

### Уговор УГ1: ПријаваНаСистем

**Операција:** LoginSO(User): signal;

**Веза са СК:** СК1

**Предуслови:** /

**Постуслови:** /

### Уговор УГ2: ВратиСвеТипове

**Операција:** GetAllTipsSO(): signal;

**Веза са СК:** СК2

**Предуслови:** /

**Постуслови:** /

### Уговор УГ3: СачувајТикет

**Операција:** SaveTicketSO(Ticket, List<Has>): signal;

**Веза са СК:** СК2

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

**Постуслови:** Креиран је нови тикет.

### Уговор УГ4: ВратиТикет

**Операција:** GetTicketSO(Ticket): signal;

**Веза са СК:** СК3, СК4, СК5

**Предуслови:** /

**Постуслови:** /

### **Уговор УГ5: ОбришиТикет**

**Операција:** DeleteTicketSO(Ticket): signal;

**Веза са СК:** СК4

**Предуслови:** Структурна ограничења над објектом Тикет морају бити задовољена.

**Постуслови:** Тикет је обрисан.

### **Уговор УГ6: ИсплатиТикет**

**Операција:** PayoffTicketSO(Ticket): signal;

**Веза са СК:** СК5

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

**Постуслови:** Тикет је исплаћен.(Подаци о тикету су изменењени)

### **Уговор УГ7: ВратиСвеTimове**

**Операција:** GetAllTeamsSO(): signal;

**Веза са СК:** СК6

**Предуслови:** /

**Постуслови:** /

### **Уговор УГ8: КреирајМеч**

**Операција:** CreateMatchSO(Match, List<Odds>): signal;

**Веза са СК:** СК6

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

**Постуслови:** Креиран је нови меч.

### **Уговор УГ9: ВратиМечЗаУносРезултата**

**Операција:** GetMatchForScoreInsertSO(Match): signal;

**Веза са СК:** СК7

**Предуслови:** /

**Постуслови:** /

### **Уговор УГ10: УносРезултата**

**Операција:** InsertScoreSO(Match): signal;

**Веза са СК:** СК7

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

**Постуслови:** Унет је резултат за меч.(Подаци о мечу су изменењени)

### **Уговор УГ11: ВратиРезултатеПословања**

**Операција:** GetBusinessResultsSO(Date, Date): signal;

**Веза са СК:** СК8

**Предуслови:** /

**Постуслови:** /

## Структура софтверског система – Концептуални(доменски) модел



Слика 1: Концептуални модел софтверског система

## Структура софтверског система – Релациони модел

User(username, password, firstname, lastname)

Team(id, name)

Match(id, startTime, scoreInserted, goalsHome, goalsAway, *username*, *home*, *away*)

Tip(name)

Odds(id, name, value)

Ticket(id, paid, totalOdds, win, timeOfPayement, status, *username*)

Has(*matchId*, *name*, *ticketId*)

Tabela User		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Struktурно ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzav. atributa jedne tabele	Međuzav. atributa više tabele	
username	username	String	Not null			<b>INSERT</b> / <b>UPDATE</b> <b>CASCADES</b>
	password	String	Not null			Match, Ticket
	firstname	String	Not null			<b>DELETE</b>
	lastname	String	Not null			<b>RESTRICTED</b>
						Match, Ticket

Табела 1: Табела User

Tabela Team		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Struktурноogrаничење
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzav. atributa jedne tabele	Međuzav. atributa više tabele	
id	id	String	Not null			<b>INSERT</b> / <b>UPDATE</b> <b>RESTRICTED</b>
	name	String	Not null			Match
						<b>DELETE</b> <b>RESTRICTED</b> Match

Табела 2: Табела Team

Tabela Match		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzav. atributa jedne tabele	Međuzav. atributa više tabela	
	id	Integer	not null and >0, auto-increment			<b>INSERT</b> RESTRICTED User, Team
	startTime	Date	not null			<b>UPDATE</b> RESTRICTED Odds
	scoreInserted	Boolean	(default: false)			<b>DELETE</b> RESTRICTED Odds
	goalsHome	Integer				
	goalsAway	Integer				
	username	String	not null			
	home	String	not null			
	away	String	not null			

Табела 3: Табела Match

Tabela Tip		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzav. atributa jedne tabele	Međuzav. atributa više tabela	
	name	String	not null			<b>INSERT</b> / <b>UPDATE</b> RESTRICTED Odds <b>DELETE</b> RESTRICTED Odds

Табела 4: Табела Tip

Tabela Odds		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzav. atributa jedne tabele	Međuzav. atributa više tabela	
Atributi	id	Integer	not null and >0			<b>INSERT</b> RESTRICTED Match, Tip
	name	String	not null			<b>UPDATE</b> RESTRICTED Has
	value	Double	not null and >=1			<b>DELETE</b> RESTRICTED Has

Табела 5: Табела Odds

Tabela Ticket		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzav. atributa jedne tabele	Međuzav. atributa više tabela	
Atributi	id	Integer	not null and >0, auto-increment			<b>INSERT</b> RESTRICTED User
	paid	Double	not null and > 0			<b>UPDATE</b> CASCADES Has
	totalOdds	Double	not null and >=1			<b>DELETE</b> CASCADES Has
	win	Double	not null and >0			
	timeOfPayment	Date	not null			
	status	String	not null			
	username	String	not null			

Табела 6: Табела Ticket

Tabela Has		Prosto vrednosno ograničenje		Složeno vrednosno ograničenje		Strukturno ograničenje
Atributi	Ime	Tip atributa	Vrednost atributa	Međuzav. atributa jedne tabele	Međuzav. atributa više tabela	
	matchId	Integer	not null and >0			<b>INSERT</b> RESTRICTED Ticket, Odds
	name	String	not null			<b>UPDATE</b> RESTRICTED Ticket, Odds
	ticketId	Integer	not null and >0			<b>DELETE</b> /

Табела 7: Табела Has

## Пројектовање

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

## Архитектура софтверског система

У пројекту је коришћена тронивојска архитектура која се састоји од следећих нивоа:

- Корисничког интерфејса
- Апликационе логике
- Складишта података.

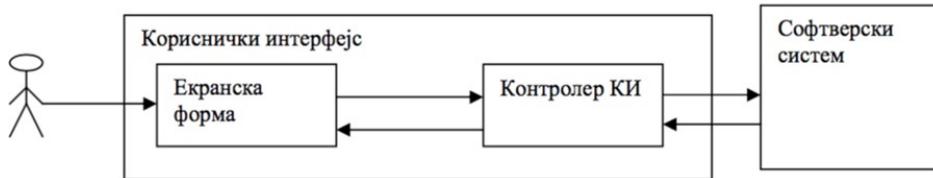


Слика 2: Тронивојска архитектура

Кориснички интерфејс се налази на клијентској страни, док су апликациона логика и складиште на страни сервера.

## Пројектовање корисничког интерфејса

Кориснички интерфејс представља реализацију улаза и/или излаза софтверског система. Кориснички интерфејс се састоји од екранских форми и контролера корисничког интерфејса. Екранска форма има улогу да приhvата податке које корисник уноси, приhvата догађаје које корисник прави, позива контролера корисничког интерфејса и приказује податке које добија као резултат од контролера.



Слика 3: Структура корисничког интерфејса

## Пројектовање екранских форми

Кориснички интерфејс је дефинисан преко скупа екранских форми. Сценарија коришћења екранских форми су директно повезана са сценаријима случајева коришћења. На следећим slikama су приказане почетне екранске форме клијентске и серверске апликације.



Слика 4: Приказ форме за покретање сервера



Слика 5: Приказ форме за конектовање на сервер

## СК1: Случај коришћења – Пријава на систем

### Назив СК

Пријава на систем

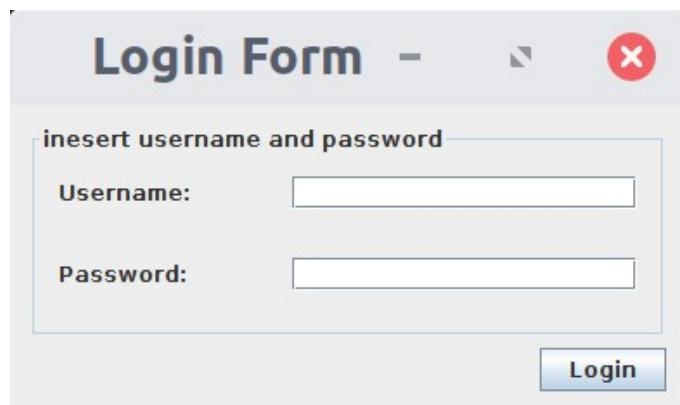
### Актори СК

Запослени

### Учесници СК

Запослени и систем (програм)

**Предуслов:** Систем је укључен. Систем приказује форму за пријаву на систем.



Слика 6: Пrikaz forme za prijavljivanje na sistem

### Основни сценарио СК

1. Запослени **уноси** корисничко име и лозинку. (АПУСО)

**Опис акције:** Запослени уноси вредности у поља са називима Username и Password.

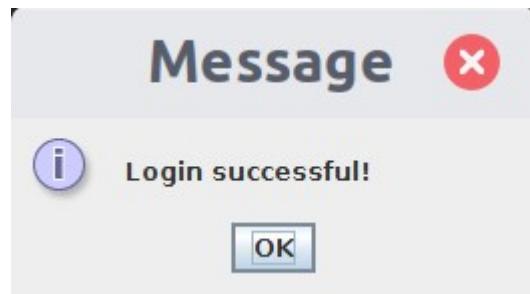
2. Запослени **контролише** да ли је коректно унео корисничко име и лозинку. (АХСО)

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

**Опис акције:** Запослени кликом на дугме „Login“ позива системску операцију LoginSO(User) која проверава податке и пријављује га на систем.

4. Систем **проверава** податке о запосленом. (СО)

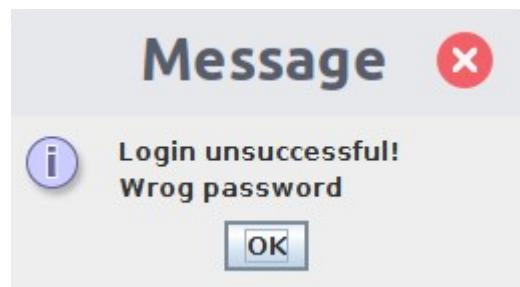
5. Систем **приказује** запосленом поруку: “Login successful! “. (ИА)



*Слика 7: Приказ поруке о успешном пријављивању*

#### **Алтернативна сценарија**

- 5.1. Уколико систем не може да нађе запосленог он приказује запосленом поруку: “Login unsuccessful!”. (ИА)



*Слика 8: Приказ поруке о неуспешном пријављивању*

## СК2: Случај коришћења – Уплата новог тикета

### Назив СК

Уплата новог тикета

### Актори СК

Запослени

### Учесници СК

Запослени и систем (програм)

**Предуслов:** Систем је укључен и запослени је улогован под својом шифром. Систем приказује форму за рад са тикетима. Учитана је листа типова.

The screenshot shows a software interface titled "New ticket". In the top left, it says "Current user: Pera Peric". Below that is a section for "Match and tip" with "Match ID" set to "1" and "Tip" set to "1". There are buttons for "Add", "Edit", and "Delete". To the right, there are fields for "Payment" and "Total odds:". At the bottom, there is a table titled "Ticket" with columns: ID, Home, Away, Start time, Tip, Odds. A "Save" button is located at the bottom right of the form area.

Слика 9: Приказ форме за рад са тикетима - Уплата новог тикета

### Основни сценарио СК

1. Запослени **уноси** податке у нови тикет. (АПУСО)

**Опис акције:** Запослени уноси вредност у поља са називима Payment и Match ID и бира тип из падајуће листе под називом Tip.

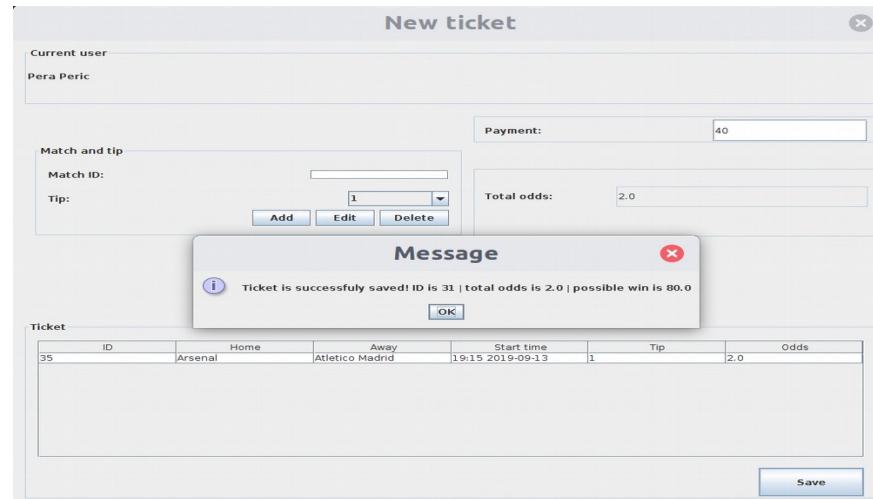
2. Запослени **контролише** да ли је коректно унео податке. (АНСО)

3. Запослени **позива** систем да запамти податке о тикету. (АПСО)

**Опис акције:** Запослени кликом на дугме „Save“ позива системску операцију SaveTicketSO(Ticket) која креира нови тикет и чува податке о тикету.

4. Систем **памти** податке о тикету. (СО)

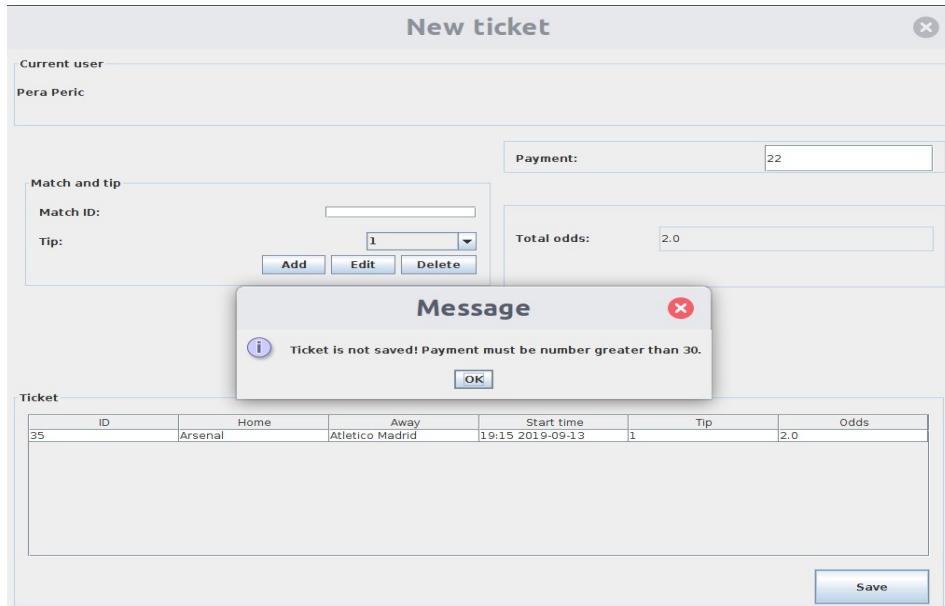
5. Систем **приказује** запосленом поруку: “Ticket is successfully saved!”. (ИА)



Слика 10: Приказ поруке о успешној уплати тикета

### Алтернативна сценарија

5.1. Уколико систем не може да запамти податке о тикету он приказује запосленом поруку: “Ticket is not saved!”. (ИА)



Слика 11: Приказ поруке о неуспешној уплати тикета

### СК3: Случај коришћења – Претрага тикета

#### Назив СК

Претрага тикета

#### Актори СК

Запослени

#### Учесници СК

Запослени и систем (програм)

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

Слика 12: Приказ форме за рад са тикетима - Претрага тикета

#### Основни сценарио СК

1. Запослени **уноси** вредност по којој претражује тикет. (АПУСО)

**Опис акције:** Запослени уноси вредност у поље под називом Ticket ID.

2. Запослени **позива** систем да нађе тикет по задатој вредности. (АПСО)

**Опис акције:** Запослени кликом на дугме „Search“ позива системску операцију GetTicketSO(Ticket).

3. Систем **тражи** тикет по задатој вредности. (СО)

#### 4. Систем приказује запосленом податке о тикету. (ИА)

The screenshot shows the 'Check ticket' application interface. At the top, it displays the current user as 'Pera Peric'. Below this is a search section titled 'Search Ticket' with a 'Ticket ID' input field containing '27' and a 'Search' button. To the right, there are fields for 'Payment' (3434.0), 'Date' (05/08 2019-07-12), and 'Employee' (Pera Peric). Further down, it shows 'Total odds' (2.0) and 'Possible win' (6868.0). A table titled 'Ticket' lists a single row: ID 29, Home Manchester City, Away Juventus, Start time 03:37 2019-07-04, Tip 2, Odds 2.0, and Final score 1 - 2. At the bottom, a green bar indicates the status as 'Winning'.

Слика 13: Приказ форме након успешне претраге тикета

#### Алтернативна сценарија

4.1. Уколико систем не може да пронађе тикет он приказује запосленом поруку: "Ticket is not found!".(ИА)

The screenshot shows the 'Check ticket' application interface. At the top, it displays the current user as 'Pera Peric'. Below this is a search section titled 'Search Ticket' with a 'Ticket ID' input field containing '33' and a 'Search' button. A message dialog box titled 'Message' appears, stating 'Ticket is not found! Ticket 33 does not exist' with an 'OK' button. The background of the main window is mostly empty, indicating no results were found.

Слика 14: Приказ поруке о неуспешној претрази тикета

## СК4: Случај коришћења – Брисање тикета

### Назив СК

Брисање тикета

### Актори СК

Запослени

### Учесници СК

Запослени и систем (програм)

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

Слика 15: Приказ форме за рад са тикетима -  
Брисање тикета

### Основни сценарио СК

1. Запослени **уноси** вредност по којој претражује тикет. (АПУСО)

**Опис акције:** Запослени уноси вредност у поље под називом Ticket ID.

2. Запослени **позива** систем да нађе тикет по задатој вредности. (АПСО)

**Опис акције:** Запослени кликом на дугме „Search“ позива системску операцију GetTicketSO(Ticket).

3. Систем **тражи** тикет по задатој вредности. (СО)

4. Систем **приказује** запосленом тикет.(ИА)

The screenshot shows a 'Delete ticket' dialog box. At the top, it displays the current user as 'Pera Peric'. Below that is a 'Search Ticket' section with a 'Ticket ID' input field containing '31' and a 'Search' button. To the right, there are fields for 'Payment' (40.0), 'Date' (20:19 2019-08-29), and 'Employee' (Pera Peric). Further down, there are fields for 'Total odds' (2.0) and 'Possible win' (80.0). A table titled 'Ticket' lists a single row with ID 35, Home team Arsenal, Away team Atletico Madrid, Start time 19:15 2019-09-13, Tip 1, Odds 2.0, and Final score //. At the bottom, a status bar says 'Not finished' and contains a 'Delete' button.

Слика 16: Приказ форме након успешне претраге тикета

5. Запослени **позива** систем да обрише тикет.(АПСО)

**Опис акције:** Запослени кликом на дугме „Delete“ позива системску операцију DeleteTicketSO(Ticket).

6. Систем **бриси** тикет. (CO)

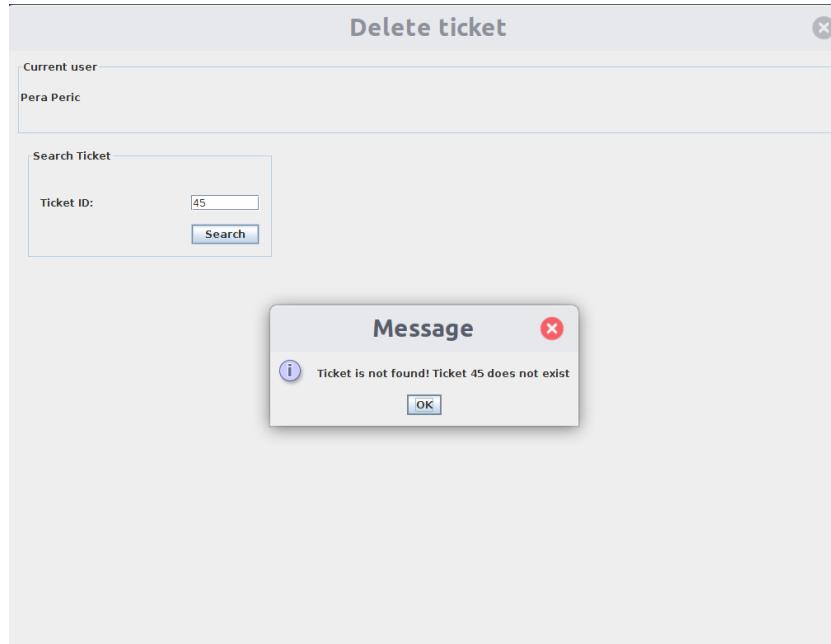
7. Систем **приказује** запосленом поруку: “Ticket is successfully deleted!”. (ИА)

This screenshot is identical to Screenshot 16, showing the 'Delete ticket' dialog with ticket ID 31 selected. However, a 'Message' dialog box has appeared in the center, stating 'Ticket 31 is successfully deleted!' with an 'OK' button. The rest of the interface remains the same, including the table and status bar.

Слика 17: Приказ поруке о успешном брисању тикета

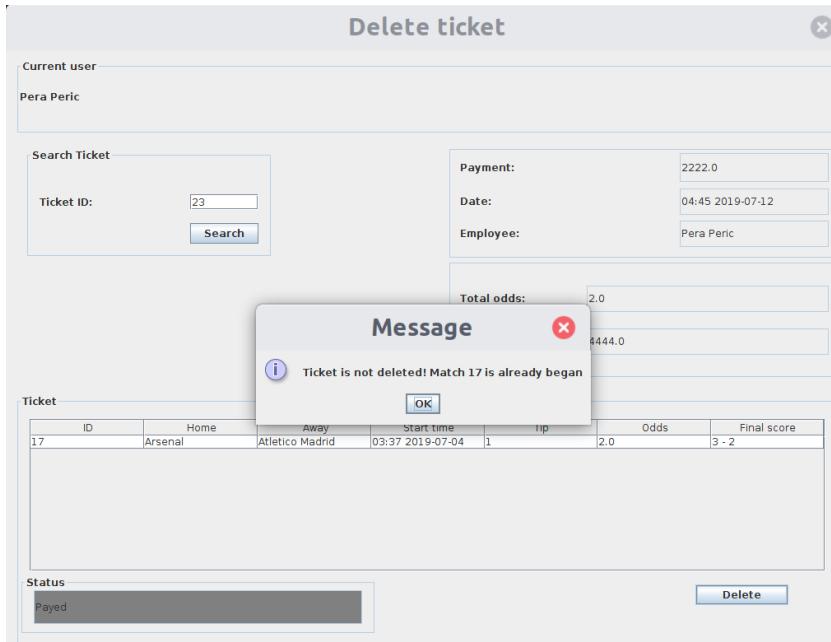
## Алтернативна сценарија

4.1. Уколико систем не може да нађе тикет он приказује запосленом поруку: “Ticket is not found!”. Прекида се извршавање сценарија. (ИА)



Слика 18: Приказ поруке о неуспешној претрази тикета

7.1. Уколико систем не може да обрише тикет он запосленом приказује поруку: “Ticket is not deleted!”. (ИА)



Слика 19: Приказ поруке о неуспешном брисању тикета

## СК5: Случај коришћења – Исплата тикета

### Назив СК

Исплата тикета

### Актори СК

Запослени

### Учесници СК

Запослени и систем (програм)

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

Слика 20: Приказ форме за рад са тикетима -  
Исплата тикета

### Основни сценарио СК

1. Запослени уноси вредност по којој претражује тикет. (АПУСО)

**Опис акције:** Запослени уноси вредност у поље под називом Ticket ID.

2. Запослени позива систем да нађе тикет по задатој вредности. (АПСО)

**Опис акције:** Запослени кликом на дугме „Search“ позива системску операцију GetTicketSO(Ticket).

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

4. Систем **приказује** запосленом тикет.(ИА)

The screenshot shows the 'Payoff ticket' window. At the top, it displays the current user as 'Pera Peric'. Below this is a search section with a 'Ticket ID' input field containing '13' and a 'Search' button. To the right, there are fields for 'Payment' (60.0), 'Date' (03:32 2019-07-11), and 'Employee' (Pera Peric). Further down, 'Total odds' (2.0) and 'Possible win' (120.0) are shown. A table labeled 'Ticket' lists a single row: ID 14, Home Arsenal, Away Atletico Madrid, Start time 03:37 2019-07-04, Tip 1, Odds 2.0, and Final score 3 - 2. At the bottom, a green progress bar is labeled 'Winning' and a blue 'Payoff' button is visible.

*Слика 21: Приказ форме након успешне претраге тикета*

5. Запослени **позива** систем да исплати тикет.(АПСО)

**Опис акције:** Запослени кликом на дугме „Payoff“ позива системску операцију PayoffTicketSO(Ticket).

6. Систем **исплаћује** тикет. (СО)

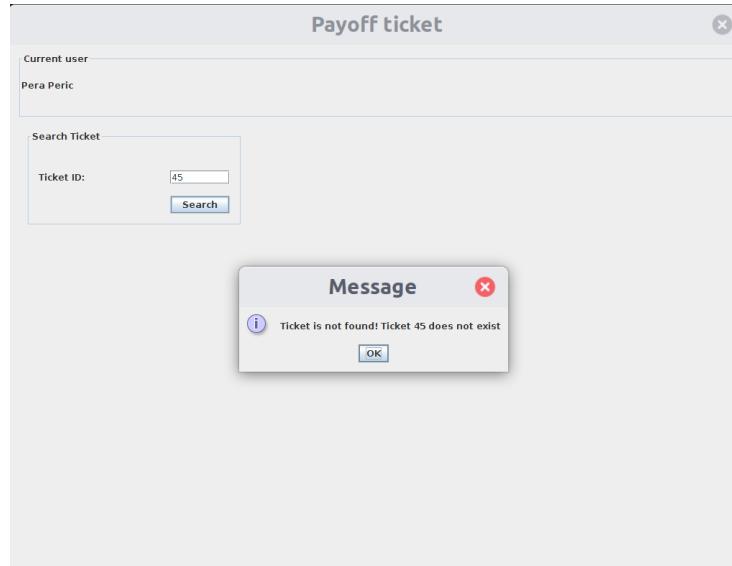
7. Систем **приказује** запосленом поруку: “Ticket is successfully payed!”. (ИА)

This screenshot is identical to Screenshot 21, showing the 'Payoff ticket' window with a successful payment. However, a modal dialog box titled 'Message' is overlaid in the center. It contains the text 'Ticket 13 is successfully payed! Total winning is 120.0' and an 'OK' button. The rest of the interface elements are visible beneath the message box.

*Слика 22: Приказ поруке о успешном брисању тикета*

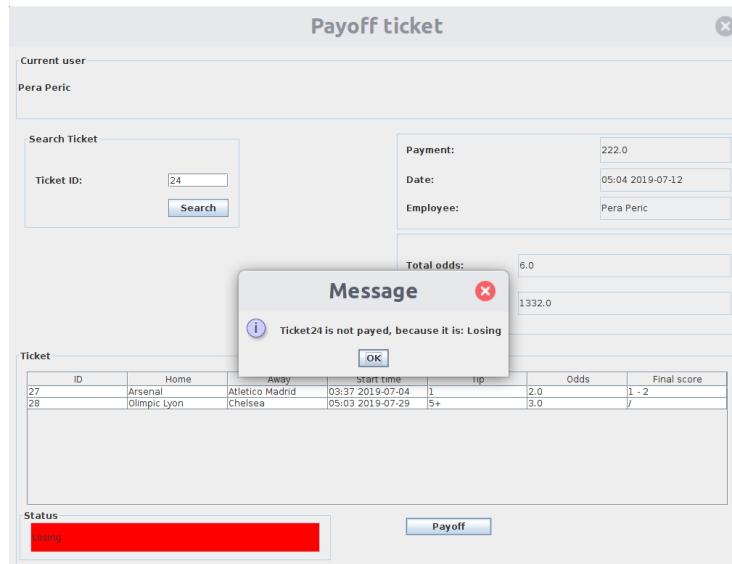
## Алтернативна сценарија

4.1. Уколико систем не може да нађе тикет он приказује запосленом поруку: “Ticket is not found!”. Прекида се извршавање сценарија. (ИА)



Слика 23: Приказ поруке о неуспешној претрази тикета

7.1. Уколико систем не може да исплати тикет он запосленом приказује поруку: “Ticket is not payed!”. (ИА)



Слика 24: Приказ поруке о неуспешној исплати тикета

## СК6: Случај коришћења – Креирање нове утакмице

### Назив СК

Унос нове утакмице

### Актори СК

Запослени

### Учесници СК

Запослени и систем (програм)

**Предуслов:** Систем је укључен и запослени је улогован под својом шифром. Систем приказује форму за рад са утакмицама. Учитана је листа тимова.

The screenshot shows a 'New match' dialog box. At the top, it displays the current user as 'Pera Peric'. Below that, there are two dropdown menus for 'Teams': 'HOME' is set to 'Arsenal' and 'AWAY' is set to 'Atletico Madrid'. Under the 'Odds' section, there are two groups of input fields. The first group, 'Winner', contains three options: '1' (with an empty input field), 'x' (with an empty input field), and '2' (with an empty input field). The second group, 'Number of goals', contains four options: '0-1' (with an empty input field), '2-3' (with an empty input field), '3+' (with an empty input field), and '5+' (with an empty input field). At the bottom left is a 'Time' field showing '8/30/19 3:37:26 AM'. To the right of the time field is a 'Create Match' button. The entire dialog has a light gray background and a white main content area.

Слика 25: Приказ форме за рад са утакмицама - Креирање нове утакмице

### Основни сценарио СК

1. Запослени **уноси** податке о утакмици. (АПУСО)

**Опис акције:** Запослени бира гостујући и домаћи тим из падајућих листи са називима HOME и AWAY, затим уноси вредност у поља са називима 1, x, 2, 0-1, 2-3, 3+, 5+, на крају бира датум из поља за избор датума под називом Time.

2. Запослени **контролише** да ли је коректно унео податке о утакмици. (АНСО)
3. Запослени **позива** систем да запамти податке о утакмици. (АПСО)

**Опис акције:** Запослени кликом на дугме „Create Match“ позива системску операцију CreateMatchSO(Match, List<Odds>).

4. Систем **памти** податке о утакмици. (СО)
5. Систем **приказује** запосленом поруку: "Match is successfully created! ID: " и ID новокреиране утакмице. (ИА)



*Слика 26: Приказ поруке о успешном креирању утакмице*

#### **Алтернативна сценарија**

- 5.1. Уколико систем не може да запамти податке о утакмици он приказује запосленом поруку: "Match is not created!".(ИА)



*Слика 27: Приказ поруке о неуспешном креирању утакмице*

## **СК7: Случај коришћења – Унос коначног резултата утакмице**

### **Назив СК**

Унос коначног резултата утакмице

### **Актори СК**

Запослени

### **Учесници СК**

Запослени и систем (програм)

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

The screenshot shows a modal dialog titled "Insert result". At the top right is a red "X" button. Below the title is a section labeled "Current User" containing the name "Pera Peric". At the bottom left is a label "Match ID:" followed by a text input field and a "Search" button to its right.

*Слика 28: Приказ форме за рад са утакмицама  
- Унос коначног резултата утакмице*

## Основни сценарио СК

1. Запослени **уноси** вредност по којој претражује утакмицу. (АПУСО)

**Опис акције:** Запослени уноси вредност у поље под називом Match ID.

2. Запослени **позива** систем да нађе утакмицу по задатој вредности. (АПСО)

**Опис акције:** Запослени кликом на дугме „Search“ позива системску операцију GetMatchForScoreInsertSO(Match).

3. Систем **тражи** утакмицу по задатој вредности. (СО)

4. Систем **приказује** запосленом утакмицу.(ИА)

The screenshot shows a modal dialog titled "Insert score". At the top left is a "Current User" section with "Pera Peric" listed. Below it is a "Match ID:" field containing "20" with a "Search" button to its right. The main area is titled "Teams" and contains two columns: "HOME" and "AWAY". Under "HOME", a dropdown menu shows "Benfica". Between the "HOME" and "AWAY" sections are two empty square input fields. Under "AWAY", a dropdown menu shows "Porto". At the bottom right of the dialog is a large red button labeled "Insert Score".

Слика 29: Приказ форме након успешне претраге утакмице

5. Запослени **уноси** уноси податке о утакмици. (АПУСО)

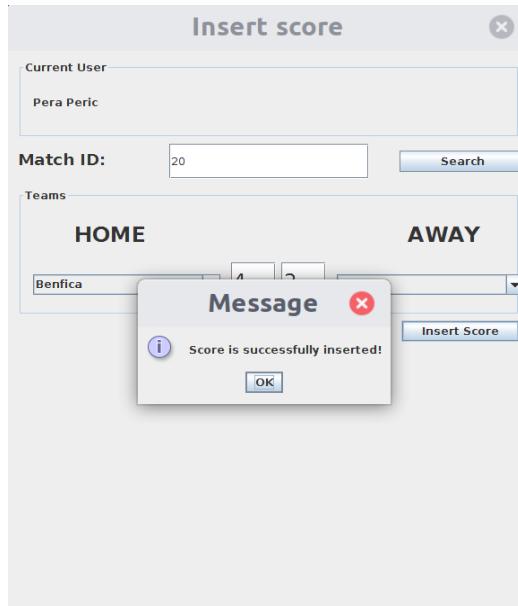
**Опис акције:** Запослени уноси вредност у поља која се налазе поред поља HOME и AWAY.

6. Запослени **контролише** да ли је коректно унео податке. (АНСО)

7. Запослени **позива** систем да запамти податке.(АПСО)

**Опис акције:** Запослени кликом на дугме „Insert Score“ позива системску операцију InsertScoreSO(Match).

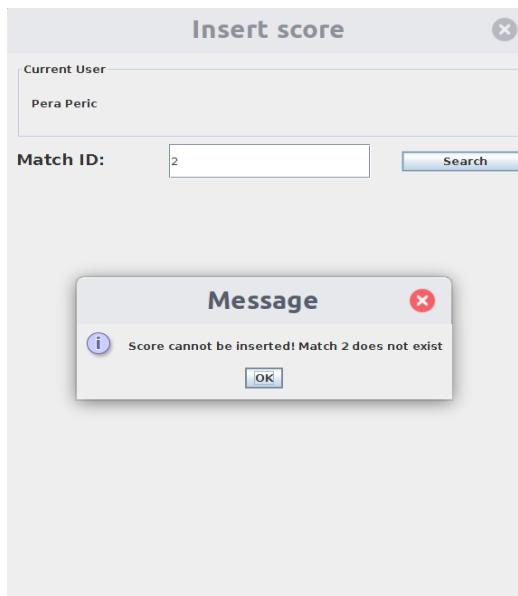
8. Систем **памти** податке. (CO)
9. Систем **приказује** запосленом поруку: “Score is successfully inserted!”. (ИА)



*Слика 30: Приказ поруке о успешном уносу резултата*

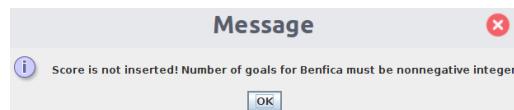
#### **Алтернативна сценарија**

- 4.1. Уколико систем не може да нађе утакмицу он приказује запосленом поруку: “Score cannot be inserted!”. Прекида се извршавање сценарија. (ИА)



*Слика 31: Приказ поруке о неуспешној претрази утакмице*

9.1. Уколико систем не може да запамти податке он приказује запослениу поруку: “Score is not inserted! ”. (ИА)



Слика 32: Приказ поруке о неуспешном уносу резултата

## СК8: Случај коришћења – Претрага пословних резултата

Претраживање пословних резултата

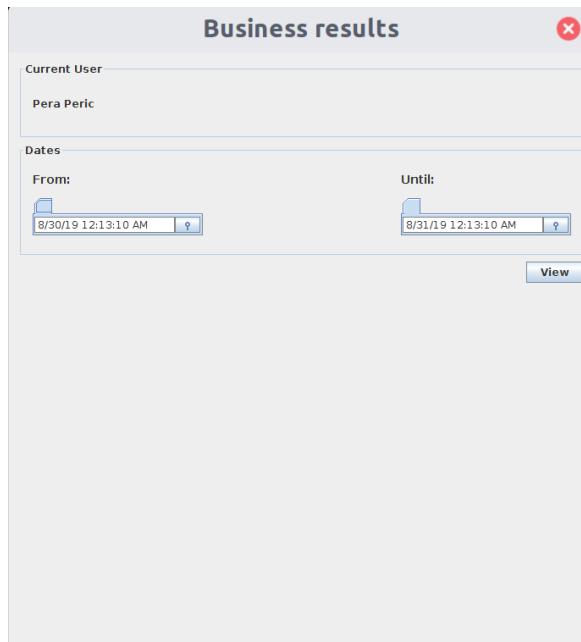
### Актори СК

Запослени

### Учесници СК

Запослени и систем (програм)

**Предуслов:** Систем је укључен и запослени је улогован под својом шифром. Систем приказује форму за рад са пословним резултатима.



Слика 33: Приказ форме за рад са пословним резултатима

### Основни сценарио СК

1. Запослени **уноси** вредност по којој претражује пословне резултате. (АПУСО)

- Опис акције:** Запослени бира датум из поља за избор датума са називима From и Until.
2. Запослени **позива** систем да нађе пословне резултате по задатој вредности. (АПСО)

**Опис акције:** Запослени кликом на дугме „View“ позива системску операцију GetBusinessResultsSO(Date, Date).

3. Систем **тражи** податке о пословним резултатима. (СО)

4. Систем приказује запосленом податке о пословним резултатима. (ИА)

The screenshot shows the 'Business results' window. In the 'Current User' section, it says 'Pera Peric'. Under 'Dates', 'From' is set to '8/30/19 12:14:00 AM' and 'Until' is set to '8/31/19 12:14:00 AM'. A 'View' button is present. The 'Tickets' section contains a table with 16 rows of data:

ID	Paid	Total odds	Possible win	Time of payment	Status	User
2	50.0	3.0	150.0	22:53 2019-07-10	Losing	Pera Peric
3	40.0	3.0	120.0	22:55 2019-07-10	Losing	Pera Peric
4	70.0	5.0	350.0	22:55 2019-07-10	Payed	Pera Peric
6	60.0	5.0	300.0	23:24 2019-07-10	Winning	Pera Peric
8	60.0	3.0	180.0	23:27 2019-07-10	Losing	Pera Peric
9	333.0	3.0	999.0	23:30 2019-07-10	Payed	Pera Peric
10	222.0	2.0	444.0	23:58 2019-07-10	Payed	Pera Peric
13	60.0	2.0	120.0	03:32 2019-07-11	Payed	Pera Peric
14	40.0	4.0	160.0	02:33 2019-07-12	Losing	Pera Peric
16	200.0	6.0	1200.0	02:54 2019-07-12	Losing	Pera Peric

The 'Results' section displays summary statistics:

- Total income: 8520.0
- Total outcome: 6497.0
- Number of payments: 21
- Number of payoffs: 7

Слика 34: Приказ форме након успешне претраге пословних резултата

### Алтернативна сценарија

4.1. Уколико систем не може да нађе податке о пословним резултатима систем приказује запосленом поруку: "Business results are not found!". (ИА)

The screenshot shows the 'Business results' window. In the 'Current User' section, it says 'Pera Peric'. Under 'Dates', 'From' is set to '8/30/19 12:14:21 AM' and 'Until' is set to '9/11/19 12:14:00 AM'. A 'View' button is present. A modal dialog box titled 'Message' appears, containing the text: 'Business results are not found! Dates cannot be in future' with an 'OK' button.

Слика 35: Приказ поруке о неуспешној претрази пословних резултата

## Пројектовање контролера корисничког интерфејса

Контролер корисничког интерфејса одговоран је да:

- Прихвата податке које шаље екранска форма;
- Конвертује податке у објекат који представља улазни аргумент који прихвата СО;
- Шаље захтев за извршење системске операције;
- Прихвата објекат који настаје као резултат извршења системске операције;
- Конвертује објекат у податке графичких елемената.

Контролер има улогу посредника између софтверског система и екранских форми.

## Пројектовање апликационе логике

Апликациона логика може да има различите улазно-излазне репрезентације и она нема знања о томе где се налази кориснички интерфејс. Апликациони сервери треба да обезбеде сервисе који ће омогућити реализацију апликационе логике софтверског система.

## Пројектовање контролера апликационе логике

У пројекту улогу контролера апликационе логике има класа *Server* која подиже серверски сокет који ослушкује мрежу. Када клијентски сокет успостави везу са серверским сокетом тада класа *Server* генерише посебну нит која је задужена за двосмерну комуникацију са клијентом.

Класа *Server* садржи листу свих клијената и одговорна је за њихово прекидање приликом затварања серверског сокета. Класа *ClientThread* је задужена да прихвата захтеве добијене од клијената и да захтев за извршење системске операције проследи до класе *Controller*, која затим преусмерава захтев до класе која је одговорна за извршење системске операције. Након извршења системске операције, резултат се враћа до *Controller-a*, а затим и до класе *ClientThread* која тај резултат шаље назад до клијента.

Комуникације између клијента и сервера је реализована разменом *transfer* објектата. Захтев за извршење системске операције се шаље преко објекта *Request* класе, док се резултат пакује у објекат класе *Response*.

## Пројектовање пословне логике

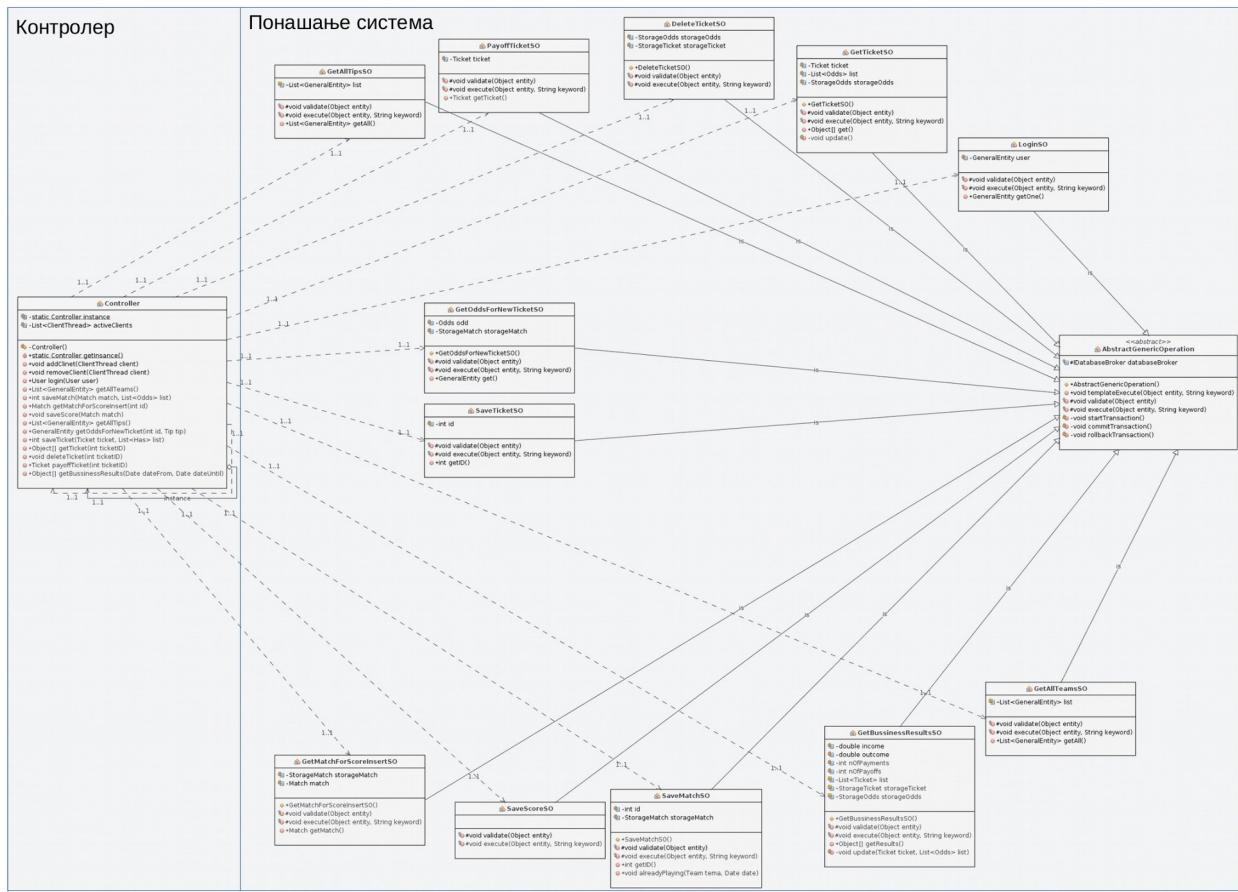
Пословна логика је описана са структуром (доменским класама) и понашањем (системским операцијама).

### Пројектовање понашања софтверског система – системске операције

Пројектовање понашања софтверског система укључује пројектовање опште класе и конкретних класа које су одговорне за извршење системских операција. Конкретне класе наслеђују класу *AbstractGenericOperation* која дефинише алгоритам за извршење СО преко методе *templateExecute*.

Пре извршења системске операције проверава се предуслов уколико постоји и отвара се трансакција. Уколико дође до изузетка приликом извршења системске операције, поништава се трансакција (rollback), у супротном се потврђује (commit).

Методе које су заједничке свим системским операцијама су имплементиране у класи *AbstractGenericOperation*. Методе специфичне за извршење СО су апстрактне (*validate*, *execute*) и свака конкретна класа мора дати имплементацију наведених метода.



Слика 36: Понашање софтверског система

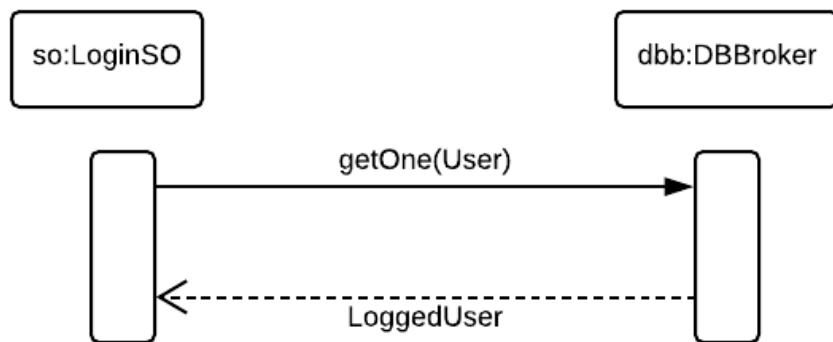
**Уговор УГ1: ПријаваНаСистем**

**Операција:** LoginSO(User): signal;

**Веза са СК:** СК1

**Предуслови:** /

**Постуслови:** /



*Дијаграм 21: УГ1: ПријаваНаСистем*

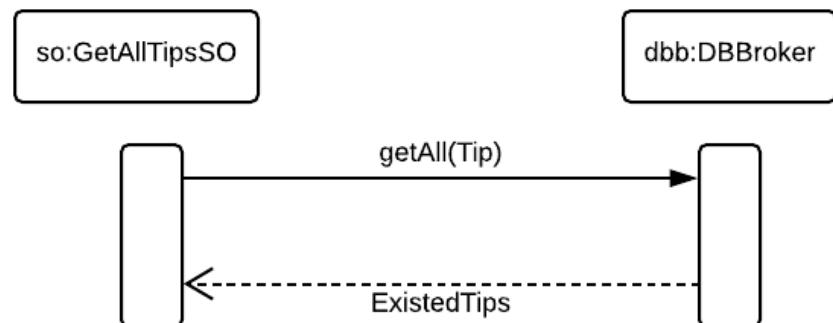
**Уговор УГ2: ВратиСвеТипове**

**Операција:** GetAllTipsSO(): signal;

**Веза са СК:** СК2

**Предуслови:** /

**Постуслови:** /



*Дијаграм 22: УГ2: ВратиСвеТипове*

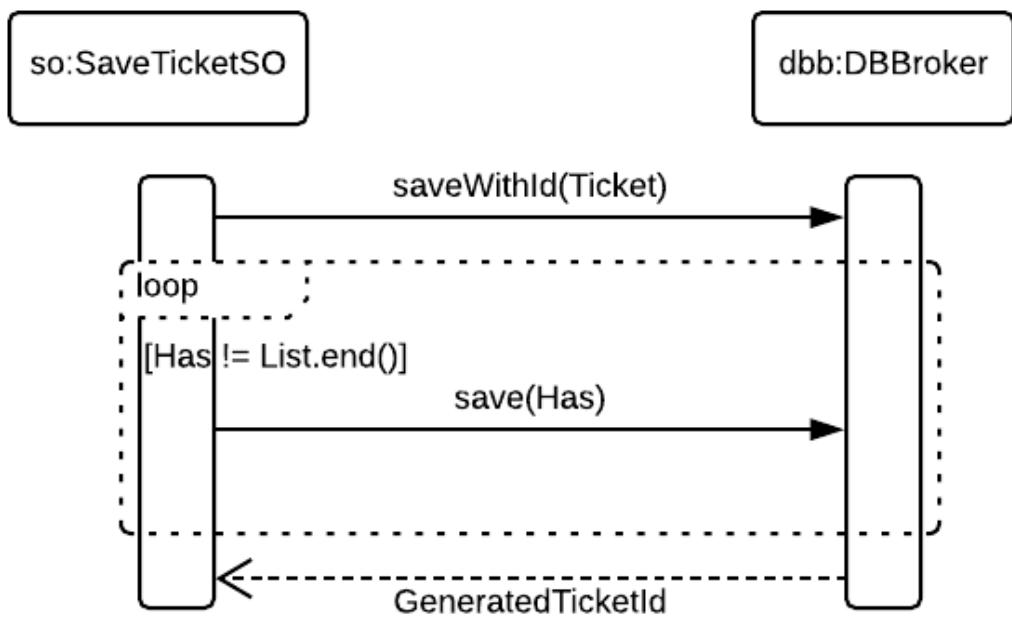
### Уговор УГ3: СачувајТикет

Операција: SaveTicketSO(Ticket, List<Has>): signal;

Веза са СК: СК2

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

Постуслови: Креиран је нови тикет.



Дијаграм 23: УГ3: СачувајТикет

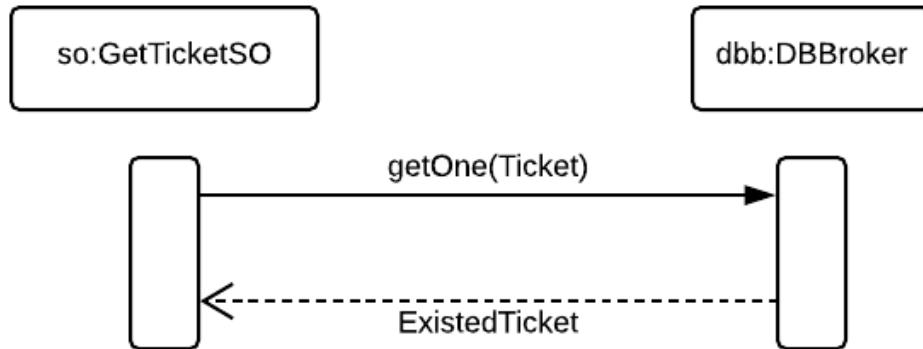
**Уговор УГ4: ВратиТикет**

**Операција:** GetTicketSO(Ticket): signal;

**Веза са СК:** СК3, СК4, СК5

**Предуслови:** /

**Постуслови:** /



*Дијаграм 24: УГ4: ВратиТикет*

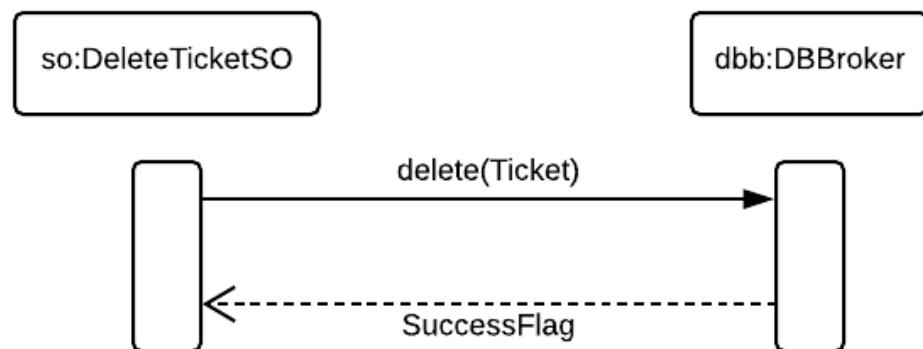
**Уговор УГ5: ОбришиТикет**

**Операција:** DeleteTicketSO(Ticket): signal;

**Веза са СК:** СК4

**Предуслови:** Структурна ограничења над објектом Тикет морају бити задовољена.

**Постуслови:** Тикет је обрисан.



*Дијаграм 25: УГ5: ОбришиТикет*

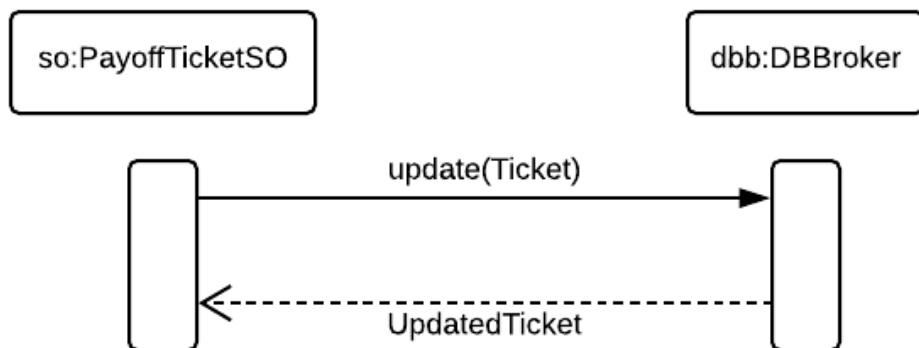
### Уговор УГ6: ИсплатиТикет

Операција: PayoffTicketSO(Ticket): signal;

Веза са СК: СК5

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

Постуслови: Тикет је исплаћен.(Подаци о тикету су изменjeni)



Дијаграм 26: УГ6: ИсплатиТикет

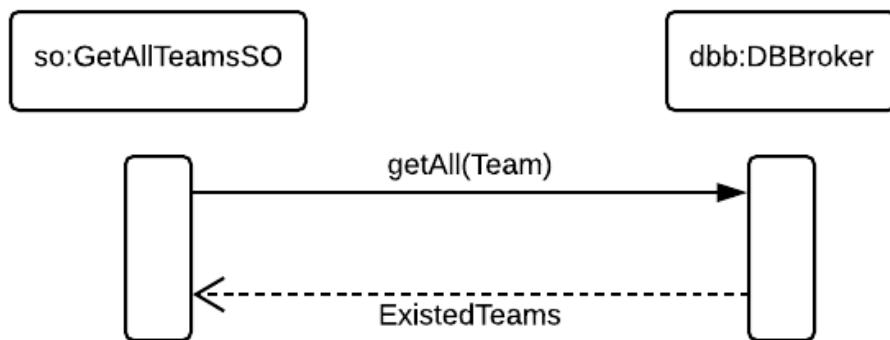
### Уговор УГ7: ВратиСвеТимове

Операција: GetAllTeamsSO(): signal;

Веза са СК: СК6

Предуслови: /

Постуслови: /



Дијаграм 27: УГ7: ВратиСвеТимове

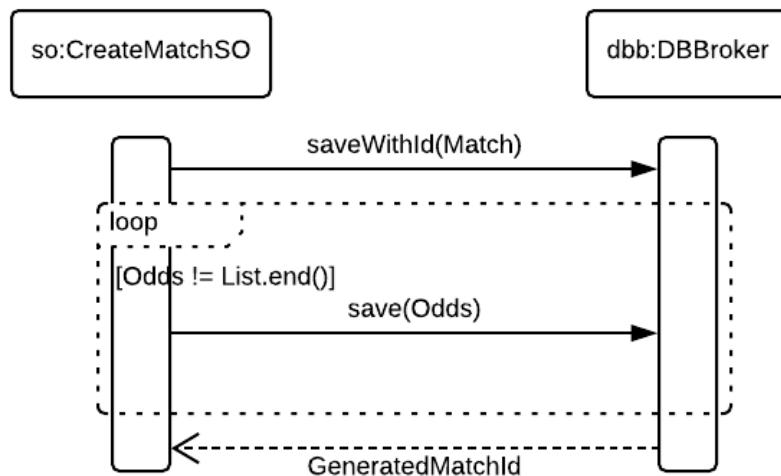
### Уговор УГ8: КреирајМеч

Операција: **CreateMatchSO**(Match, List<Odds>): signal;

Веза са СК: СК6

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

Постуслови: Креиран је нови меч.



Дијаграм 28: УГ8: КреирајМеч

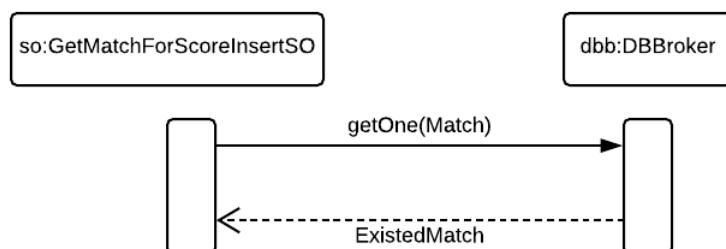
### Уговор УГ9: ВратиМечЗаУносРезултата

Операција: **GetMatchForScoreInsertSO**(Match): signal;

Веза са СК: СК7

Предуслови: /

Постуслови: /



Дијаграм 29: УГ9: ВратиМечЗаУносРезултата

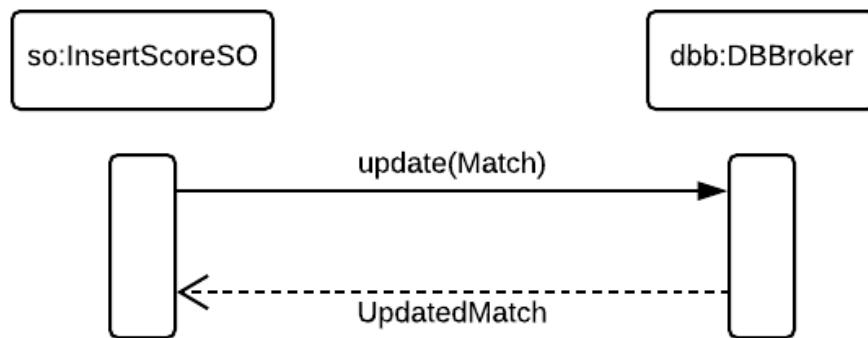
### Уговор УГ10: УносРезултата

Операција: InsertScoreSO(Match): signal;

Веза са СК: СК7

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

Постуслови: Унет је резултат за меч.(Подаци о мечу су изменjeni)



Дијаграм 30: УГ10: УносРезултата

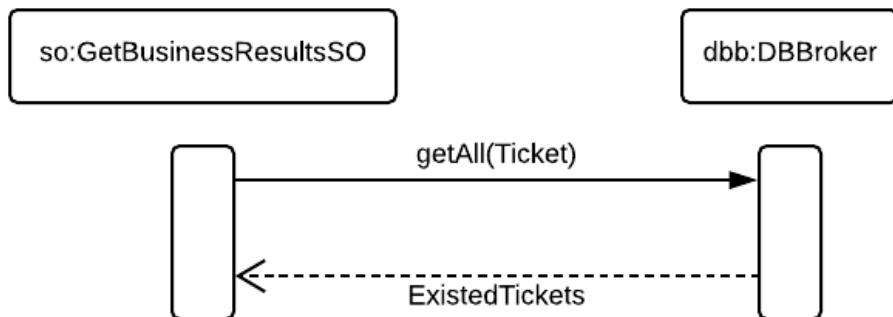
### Уговор УГ11: ВратиРезултатеПословања

Операција: GetBusinessResultsSO(Date, Date): signal;

Веза са СК: СК8

Предуслови: /

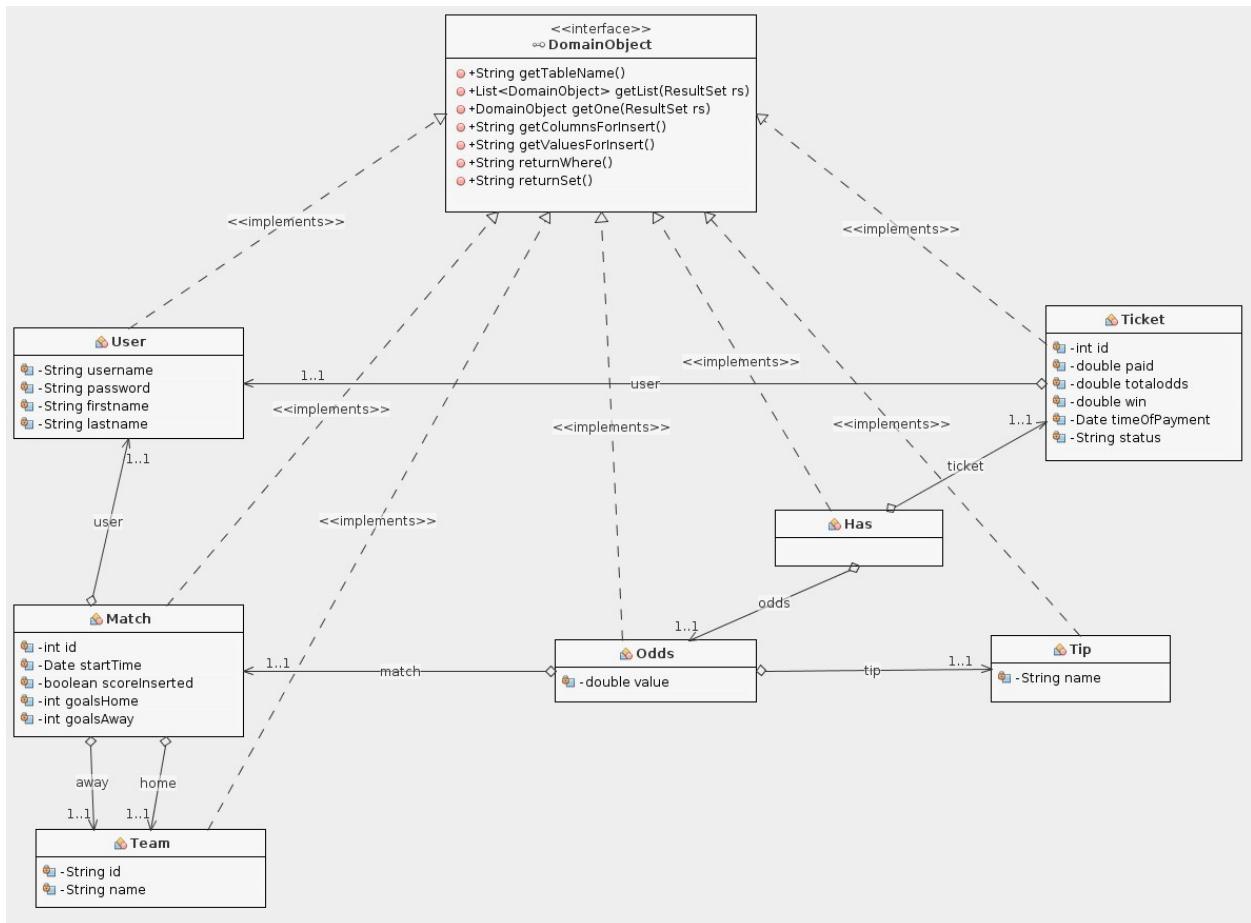
Постуслови: /



Дијаграм 31: УГ11: ВратиРезултатеПословања

## Пројектовање структуре софтверског система – доменске класе

На основу концептуалних класа праве се софтверске класе структуре.

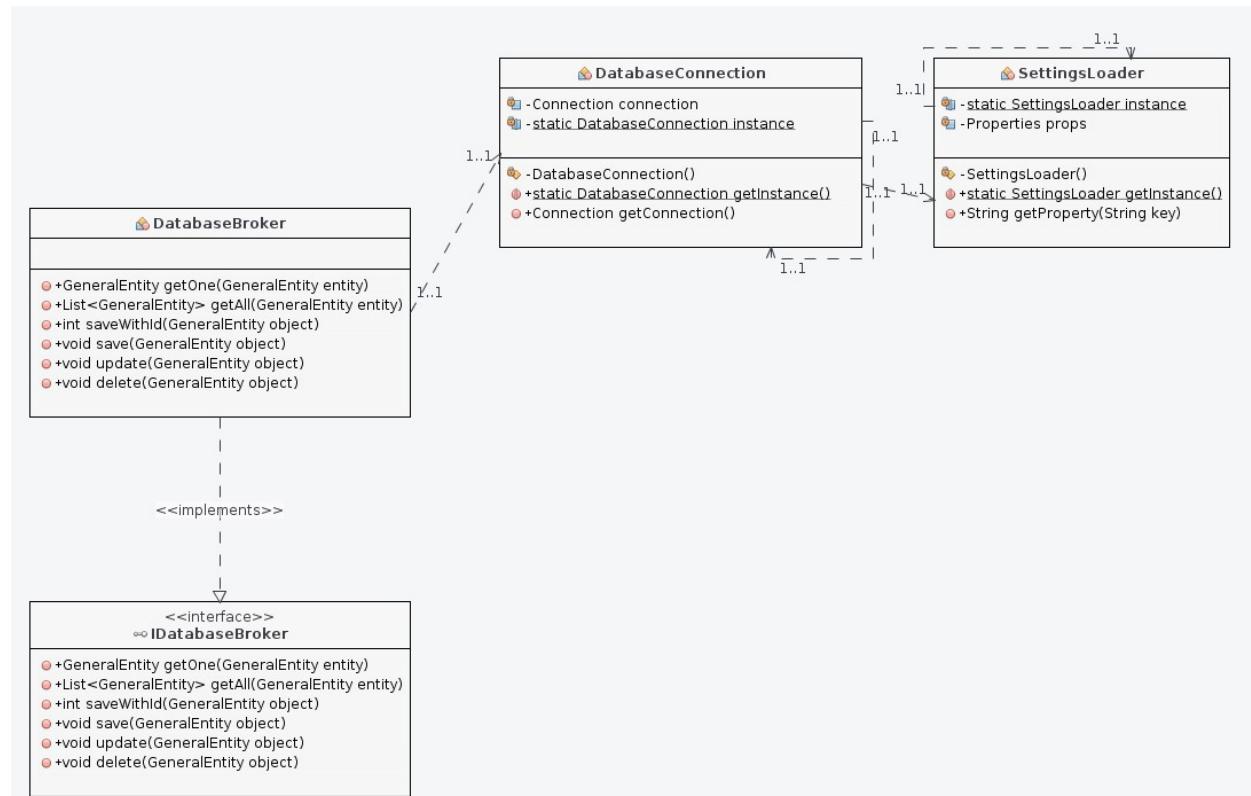


Слика 37: Доменске класе

## Пројектовање брокера базе података

Брокер базе података је одговоран за комуникацију између пословне логике и складишта података. Улогу брокера базе података има класа *DatabaseBroker*.

Класа *DatabaseBroker* обезбеђује перзистентни оквир објектима доменских класа које се чувају у бази података.



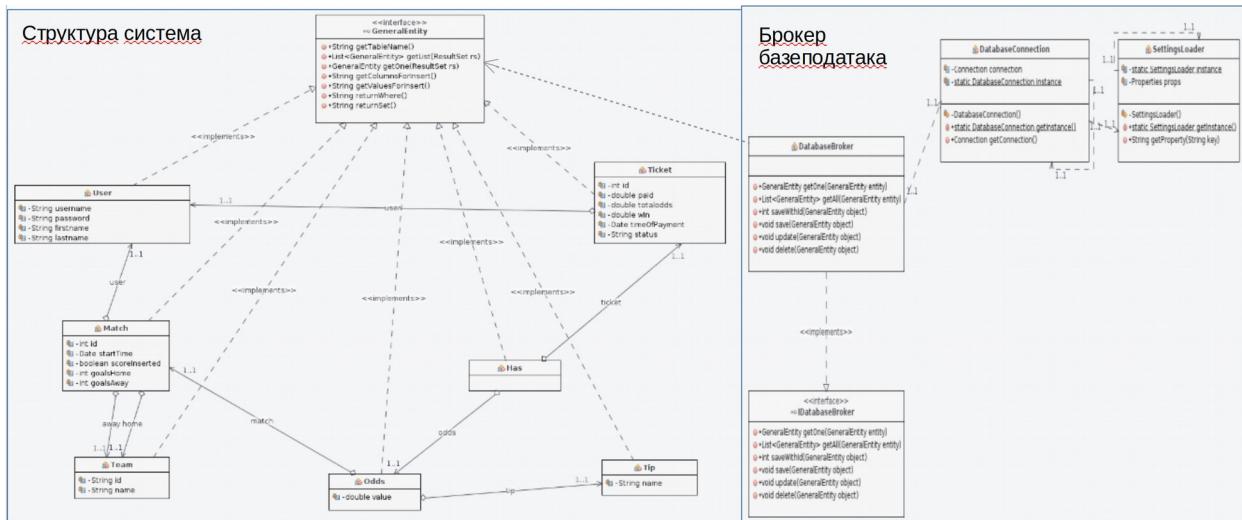
Слика 38: Перзистентни оквир

Класа *DatabaseBroker* имплементира интерфејс *IDatabaseBroker*:

```
public interface IDatabaseBroker {  
    public GeneralEntity getOne(GeneralEntity entity) throws SQLException;  
    public List<GeneralEntity> getAll(GeneralEntity entity) throws SQLException;  
    public int saveWithId(GeneralEntity object) throws SQLException;  
    public void save(GeneralEntity object) throws SQLException;  
    public void update(GeneralEntity object) throws SQLException;  
    public void delete(GeneralEntity object) throws SQLException;  
}
```

Све методе DatabaseBroker класе су уројени увоне као тенеричке, шако да не постоји имплементација појединачних метода за сваку доменску класу. Као улазни параметар тенеричких метода простирује се инверфејс које све доменске класе морају имплементирати, а што је инверфејс GeneralEntity.

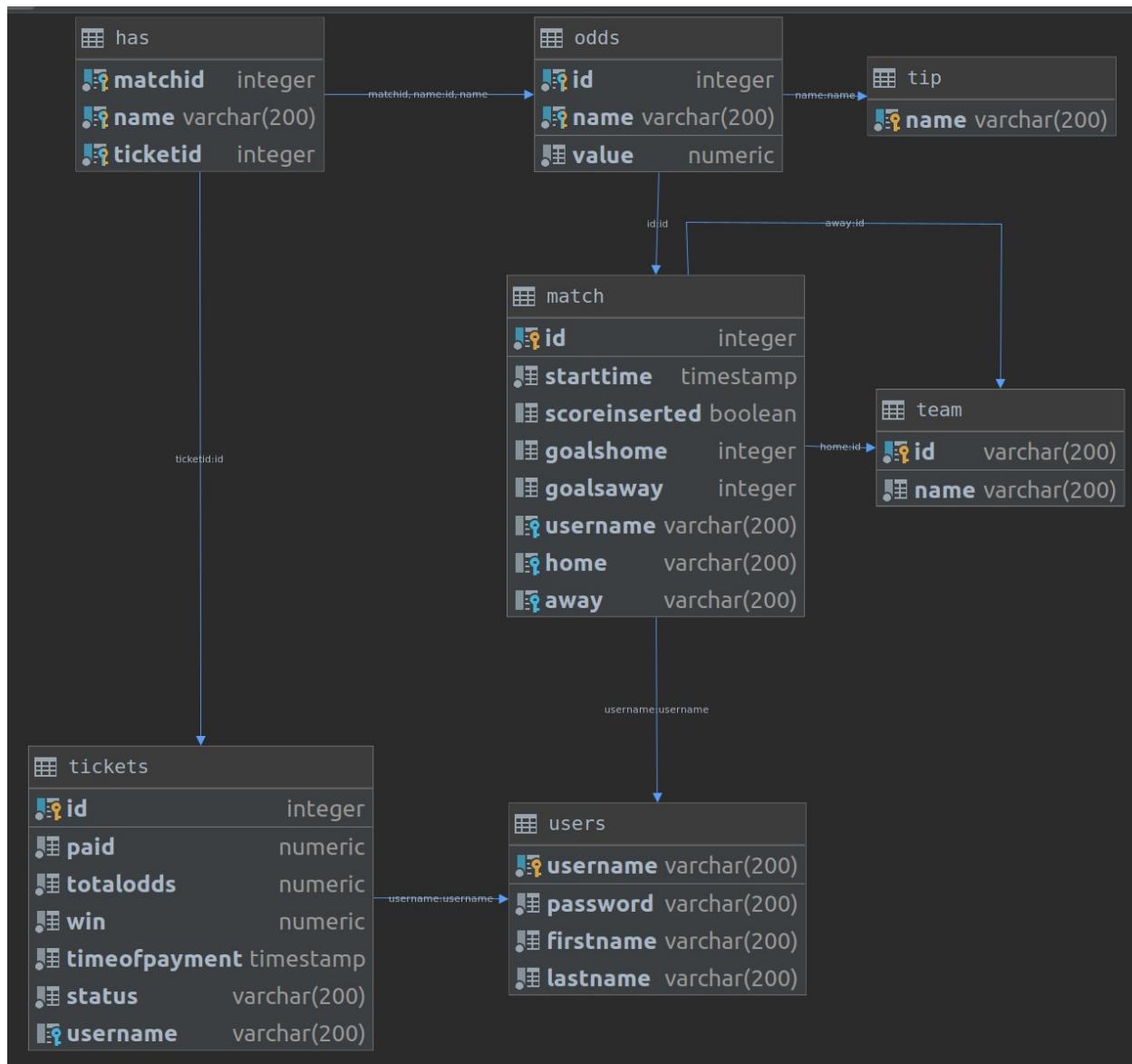
```
public interface GeneralEntity extends Serializable{
    public String getTableName();
    public List<GeneralEntity> getList(ResultSet rs) throws SQLException;
    public GeneralEntity getOne(ResultSet rs) throws SQLException;
    public String getColumnsForInsert();
    public String getValuesForInsert();
    public String returnWhere();
    public String returnSet();
}
```



Слика 39: Структура софтверског система

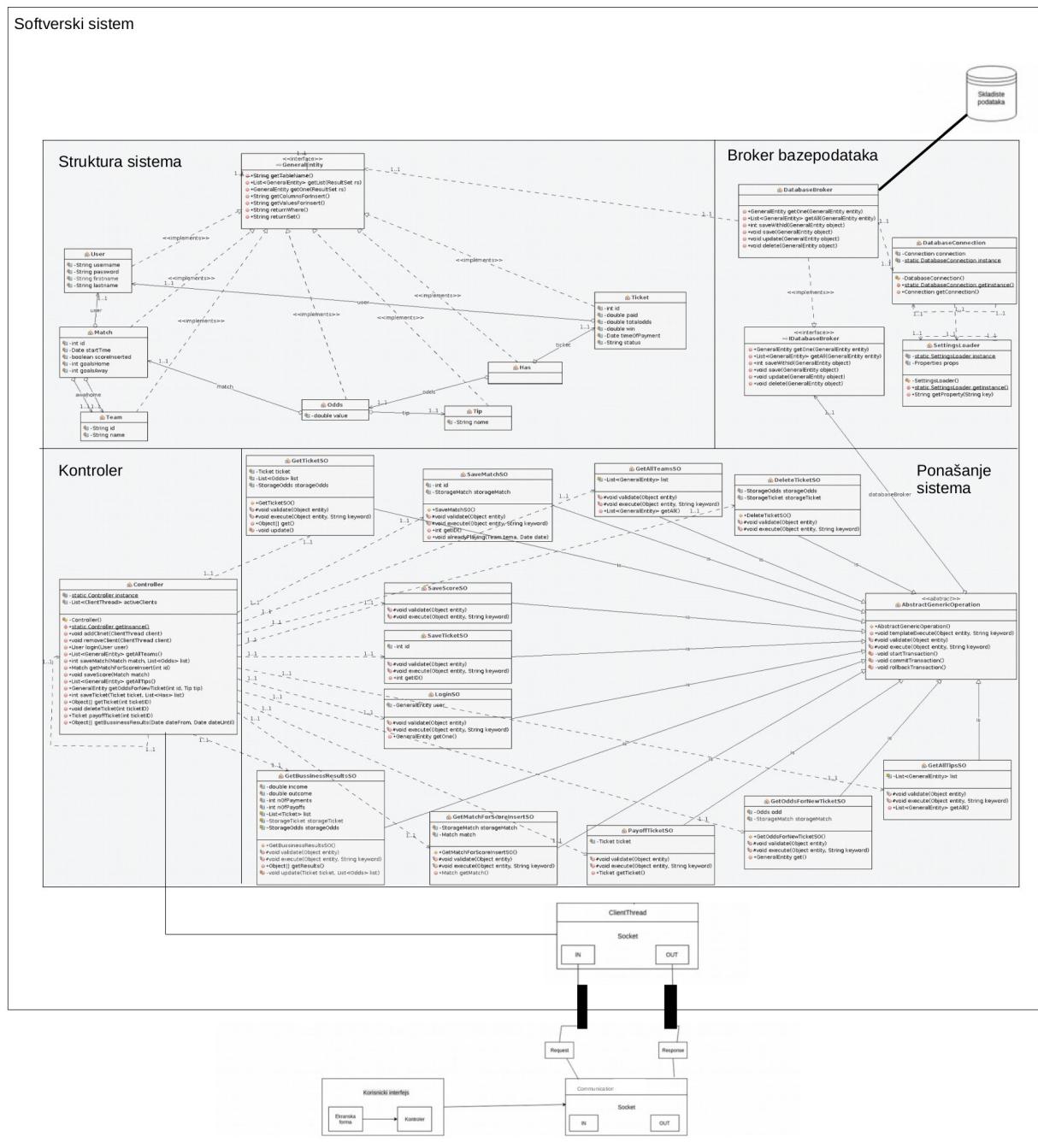
## Пројектовање складишта података

На основу софтверских класа структуре пројектоване су табеле (складишта података) релационог система за управљање базом података (PostgreSQL). Назив базе је psprojekat1 и садржи седам табела.



Слика 40: Структура складишта података

## Коначна архитектура



Слика 41: Коначна архитектура софтверског система

## Имплементација

Софтверски систем је реализован у програмском језику Java. Пројектован је као клијент-сервер апликација. Као развојно окружење коришћен је NetBeans IDE 8.1., систем за управљање базом података је релациони (PostgreSQL).

У оквиру пројекта постоје 3 потпројекта:

- Серверска страна пројекта
- Клијентска страна пројекта
- Заједнички пројекат.

## Тестирање

Извршено је мануелно тестирање свих случајева коришћења, као и валидација које су очекиване приликом уноса података. Валидације су тестиране уношењем исправних и погрешних података. Случајеви коришћења су постепено тестирани са све више активних клијената.

На основу извршених тестирања уклоњени су уочени недостаци.

## Литература

[1] др С. Влајић, Пројектовање софтвера, Београд, 2015.