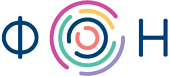
УНИВЕРЗИТЕТ У БЕОГРАДУ

ФАКУЛТЕТ ОРГАНИЗАЦИОНИХ НАУКА



Лабораторија за софтверско инжењерство

Пројектовање софтвера

-семинарски рад-

Тема:   
Софтверски систем за организацију музичких догађаја  
у програмском језику C#

Ментор: Студент:

др Синиша Влајић Јанко Гашић 134/2016

др Душан Савић

маст. Инж Татјана Стојановић

Београд, 2020

Sadržaj

[1 ПРИКУПЉАЊЕ ЗАХТЕВА ОД КОРИСНИКА 1](#_Toc83284060)

[1.1 Вербални опис 1](#_Toc83284061)

[1.2 Случајеви коришћења 1](#_Toc83284062)

[СК1: Случај коришћења – Пријављивање организатора 2](#_Toc83284063)

[СК2: Случај коришћења – Унос извођача 3](#_Toc83284064)

[СК3: Случај коришћења – Измена извођача 4](#_Toc83284065)

[СК4: Случај коришћења – Брисање извођача 5](#_Toc83284066)

[СК5: Случај коришћења – Унос музичког догађаја (Сложен СК) 6](#_Toc83284067)

[СК6: Случај коришћења – Измена музичког догађаја (Сложени СК) 7](#_Toc83284068)

[СК7: Случај коришћења – Брисање музичког догађаја 8](#_Toc83284069)

[СК8: Случај коришћења – Унос наступа (Сложен СК) 9](#_Toc83284070)

[2 Анализа 10](#_Toc83284071)

[2.1 Понашање софтверског система – системски дијаграм секвенци 10](#_Toc83284072)

[ДС1: Дијаграм секвенци случаја коришћења – Пријављивање организатора 10](#_Toc83284073)

[ДС2: Дијаграм секвенци случаја коришћења – Унос извођача 11](#_Toc83284074)

[ДС3: Дијаграм секвенци случаја коришћења – Измена извођача 12](#_Toc83284075)

[ДС4: Дијаграм секвенци случаја коришћења – Брисање извођача 14](#_Toc83284076)

[ДС5: Дијаграм секвенци случаја коришћења – Унос музичког догађаја 15](#_Toc83284077)

[ДС6: Дијаграм секвенци случаја коришћења – Измена музичког догађаја 17](#_Toc83284078)

[ДС7: Дијаграм секвенци случаја коришћења – Брисање музичког догађаја 19](#_Toc83284079)

[ДС8: Дијаграм секвенци случаја коришћења – Унос наступа 20](#_Toc83284080)

[2.2 Понашање софтверског система – дефинисање уговора о системским операцијама 22](#_Toc83284081)

[Уговор УГ1: ПријавиОрганизатора 22](#_Toc83284082)

[Уговор УГ2: СацувајИзводјаца 22](#_Toc83284083)

[Уговор УГ3: ВратиИзводјаце 22](#_Toc83284084)

[Уговор УГ4: ОбрисиИзводјаца 22](#_Toc83284085)

[Уговор УГ5: ВратиЛокације 23](#_Toc83284086)

[Уговор УГ6: СацувајМузицкиД 23](#_Toc83284087)

[Уговор УГ7: ВратиМузицкеД 23](#_Toc83284088)

[Уговор УГ8: ОбрисиМузицкиД 23](#_Toc83284089)

[Уговор УГ9: ВратиБинеОвогМузицкогД 23](#_Toc83284090)

[Уговор УГ10: СацувајНаступ 23](#_Toc83284091)

[2.3 Структура софтверског система – Концептуални (доменски) модел 24](#_Toc83284092)

[2.4 Stuktura softverskog sistema – Relacioni model 25](#_Toc83284093)

[3 ПРОјектовање 28](#_Toc83284094)

[3.1 Пројектовање корисничког интерфејса 28](#_Toc83284095)

[СК1: Случај коришћења – Пријављивање организатора 29](#_Toc83284096)

[СК2: Случај коришћења – Унос извођача 31](#_Toc83284097)

[СК3: Случај коришћења – Измена извођача 32](#_Toc83284098)

[СК4: Случај коришћења – Брисање извођача 34](#_Toc83284099)

[СК5: Случај коришћења – Унос музичког догађаја (Сложен СК) 35](#_Toc83284100)

[СК6: Случај коришћења – Измена музичког догађаја (Сложени СК) 36](#_Toc83284101)

[СК7: Случај коришћења – Брисање музичког догађаја 38](#_Toc83284102)

[СК8: Случај коришћења – Унос наступа (Сложен СК) 39](#_Toc83284103)

[3.2 Пројектовање контролера корисничког интерфејса 41](#_Toc83284104)

[3.3 Пројектовање апликационе логике 41](#_Toc83284105)

[Пројектовање контролера апликационе логике 41](#_Toc83284106)

[Пословна логика 42](#_Toc83284107)

# ПРИКУПЉАЊЕ ЗАХТЕВА ОД КОРИСНИКА

## Вербални опис

Потребно је направити софтверски систем који организатору омогућава планирање и организацију музичких догађаја. Организатор ће бити једино овлашћено лице које има права да ради са апликацијом.

За почетак, неопходно је обезбедити пријаву организатора на систем. Уносом корисничког имена и лозинке корисник добија приступ апликацији.

Организатор ће имате три опције: рад са музичким догађајима, рад са извођачима и унос наступа. Биће могућ унос новог извођача и мењање постојећих. Поред тога, корисник ће бити у могућности да обрише из базе извођача зе случај да његове услуге више неће бити потребне.

Такође, поред извођача, организатор ће бити у могућности да уноси, мења, и брише музичке догађаје у случају да они буду отказани. Приликом уноса музичког догађаја, организатор ће уносити и бине које ће се налазити на догађају.

## Случајеви коришћења

1. Пријављивање организатора
2. Унос извођача
3. Измена извођача
4. Брисање извођача
5. Унос музичког догађаја
6. Измена музичког догађаја
7. Брисање музичког догађаја
8. Унос наступа



Slika 1 - Slučajevi korišćenja

### СК1: Случај коришћења – Пријављивање организатора

**Назив СК**

Пријављивање организатора  
  
**Актори СК**

Организатор

**Учесници СК**

Организатор и систем (програм)

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

**Основни сценарио СК**

1. Организатор **уноси** податке за пријављивање (корисничкно име и лозинку). (АПУСО)
2. Организатор **контролише** да ли је исправно унео податке за пријављивање.(АНСО)
3. Организатор **позива** систем да пронађе корисника са датим подацима.(АПСО)
4. Систем **претражује** кориснике. (СО)
5. Систем **приказује** организаторупоруку: ‘Успешна пријава!‘’ и омогућава приступ систему. (ИА)

Алтернативна сценарија

5.1 Уколико систем не може да нађе корисника приказује поруку: “Неуспешна пријава!”.(ИА)

### СК2: Случај коришћења – Унос извођача

**Назив СК**

Унос извођача

**Актори СК**

Организатор

**Учесници СК**

Организатор и систем (програм)

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

**Основни сценарио СК**

1. Организатор **уноси** податке о извођачу. (АПУСО)
2. Организатор **контролише** да ли је коректно унео податке о извођачу. (АНСО)
3. Организатор **позива** систем да запамти податке о извођачу. (АПСО)
4. Систем **памти** податке о извођачу. (СО)
5. Систем **приказује** организатору запамћеног извођача и поруку: “Додат је нови извођач по имену [име]“. (ИА)

Алтернативна сценарија

5.1 Уколико систем не може да креира извођача он приказује организатору поруку: “Систем не може да креира извођача”. Прекида се извршење сценарија. (ИА)

### СК3: Случај коришћења – Измена извођача

**Назив СК**

Измена извођача

**Актори СК**

Организатор

**Учесници СК**

Организатор и систем (програм)

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

**Основни сценарио СК**

1. Организатор **бира** извођача ког жели да измени. (АПУСО)
2. Организатор **позива** систем да учита податке о одабраном извођачу. (АПСО)
3. Систем **учитава** податке о одабраном извођачу. (СО)
4. Систем **приказује** организатору податке о извођачу и поруку: “Спремно за измену”. (ИА)
5. Организатор **мења** податке о извођачу. (АПУСО)
6. Организатор **контролише** да ли је коректно унео податке о извођачу. (АНСО)
7. Организатор **позива** систем да запамти податке о извођачу. (АПСО)
8. Систем **памти** податке о извођачу. (СО)
9. Систем **приказује** организатору запамћеног извођача и поруку: “Направљене су измене над извођачем: [Име]” (ИА)

Алтернативна сценарија

4.1 Уколико систем не може да учита податке о извођачу он приказује организатору поруку: “Систем не може да учита извођача”. Прекида се извршење сценарија. (ИА)

9.1 Уколико систем не може да запамти податке о извођачу он приказује организатору поруку “Систем не може да запамти извођача”. (ИА)

### СК4: Случај коришћења – Брисање извођача

**Назив СК**

Брисање извођача

**Актори СК**

Организатор

**Учесници СК**

Организатор и систем (програм)

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

**Основни сценарио СК**

1. Организатор **бира** извођача ког жели да измени. (АПУСО)
2. Организатор **позива** систем да обрише извођача. (АПСО)
3. Систем **деактивира** извођача. (СО)
4. Систем **приказује** организатору поруку: “Систем је обрисао извођача.” (ИА)

Алтернативна сценарија

4.1 Уколико систем не може да деактивира извођача он приказује организатору поруку “Систем не може да обрише извођача”. (ИА)

### СК5: Случај коришћења – Унос музичког догађаја (Сложен СК)

**Назив СК**

Унос музичког догађаја

**Актори СК**

Организатор

**Учесници СК**

Организатор и систем (програм)

**Предуслов**: Систем је укључен и организатор је улогован под својом шифром. Систем приказује форму за рад са музичким догађајем. Систем је учитао листу локација.

**Основни сценарио СК**

1. Организатор **уноси** податке о музичком догађају и бинама. (АПУСО)
2. Организатор **контролише** да ли је коректно унео податке о музичком догађају и бинама. (АНСО)
3. Организатор **позива** систем да запамти податке о музичком догађају и бинама. (АПСО)
4. Систем **памти** податке о музичком догађају и бинама. (СО)
5. Систем **приказује** организатору запамћени догађај и поруку: “Систем је запамтио податке о музичком догађају“. (ИА)

Алтернативна сценарија

5.1 Уколико систем не може да запамти податке о музичком догађају он приказује организатору поруку “Систем не може да запамти податке о музичком догађају ”. (ИА)

### СК6: Случај коришћења – Измена музичког догађаја (Сложени СК)

**Назив СК**

Измена музичког догађаја

**Актори СК**

Организатор

**Учесници СК**

Организатор и систем (програм)

**Предуслов**: Систем је укључен и организатор је улогован под својом шифром. Систем приказује форму за рад са музичким догађајем. Систем је учитао листу локација.

**Основни сценарио СК**

1. Организатор **бира** музички догађај ког жели да измени. (АПУСО)
2. Организатор **позива** систем да учита податке о одабраном музичком догађају. (АПСО)
3. Систем **учитава** податке о одабраном музичком догађају. (СО)
4. Систем **приказује** организатору податке о музичком догађају и поруку: “Музички догађај је спреман за измену”. (ИА)
5. Организатор **мења** податке о музичком догађају. (АПУСО)
6. Организатор **контролише** да ли је коректно унео податке о музичком догађају. (АНСО)
7. Организатор **позива** систем да запамти податке о музичком догађају. (АПСО)
8. Систем **памти** податке о музичком догађају. (СО)
9. Систем **приказује** организатору запамћени догађај и поруку: “Напревљене су измене над музичким догађајем: [Име]” (ИА)

Алтернативна сценарија

9.1 Уколико систем не може да запамти податке о музичком догађају он приказује организатору поруку “Систем не може да запамти музички догађај”. (ИА)

### СК7: Случај коришћења – Брисање музичког догађаја

**Назив СК**

Брисање музичког догађаја

**Актори СК**

Организатор

**Учесници СК**

Организатор и систем (програм)

**Предуслов**: Систем је укључен и организатор је улогован под својом шифром. Систем приказује форму за рад са музичким догађајем.

**Основни сценарио СК**

1. Организатор **бира** музички догађај кој жели да измени. (АПУСО)
2. Организатор **позива** систем да обрише музички догађај. (АПСО)
3. Систем **деактивира** музички догађај. (СО)
4. Систем **приказује** организатору поруку: “Систем је обрисао музички догађај.” (ИА)

Алтернативна сценарија

4.1 Уколико систем не може да деактивира музички догађај он приказује организатору поруку “Систем не може да обрише музички догађај”. (ИА)

### СК8: Случај коришћења – Унос наступа (Сложен СК)

**Назив СК**

Унос наступа

**Актори СК**

Организатор

**Учесници СК**

Организатор и систем (програм)

**Предуслов**: Систем је укључен и организатор је улогован под својом шифром. Систем приказује форму за рад са наступом. Систем је учитао листу музичких догађаја.

**Основни сценарио СК**

1. Организатор **бира** музички догађај. (АПУСО)
2. Организатор **позива** систем да потврди музички догађај. (АПСО)
3. Систем **учитава** податке о одабраном музичком догађају (бине) и извођачима. (СО)
4. Систем **приказује** организатору музички догађај и поруку: “Систем је пронашао бине музичког догађаја [Име]“. (ИА)
5. Организатор **уноси** податке о наступу. (АПУСО)
6. Организатор **контролише** да ли је коректно унео податке о наступу. (АНСО)
7. Организатор **позива** систем да запамти податке о наступ. (АПСО)
8. Систем **памти** податке о наступу. (СО)
9. Систем **приказује** организатору запамћени догађај и поруку: “Додат је нов наступ извођача [Име извођача] на догађају [Назив догађаја]“. (ИА)

Алтернативна сценарија

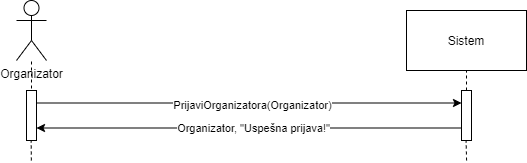
9.1 Уколико систем не може да запамти податке о наступу он приказује организатору поруку “Систем не може да запамти податке о наступу ”. (ИА)

# Анализа

## Понашање софтверског система – системски дијаграм секвенци

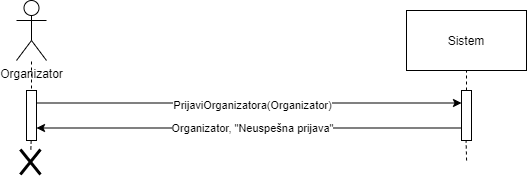
### ДС1: Дијаграм секвенци случаја коришћења – Пријављивање организатора

1. Организатор **позива** систем да пронађе корисника са датим подацима. (АПСО)
2. Систем **приказује** организаторупоруку: ‘Успешна пријава!‘’ и омогућава приступ систему. (ИА)

****

**Алтернативна сценарија**

2.1 Уколико систем не може да нађе корисника приказује поруку: “Неуспешна пријава!”.(ИА)

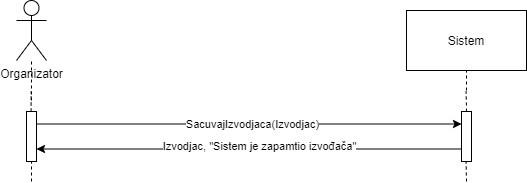


Са наведених секвенцних дијаграма уочавају се системске операције које треба пројектовати:

1. signal **PrijaviOrganizatora**(Organizator)

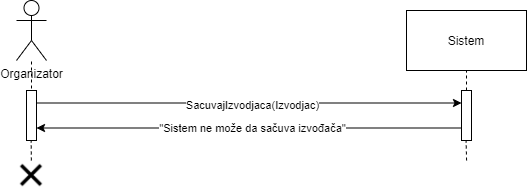
### ДС2: Дијаграм секвенци случаја коришћења – Унос извођача

1. Организатор **позива** систем да запамти податке о извођачу. (АПСО)
2. Систем **приказује** организатору запамћени и поруку: “Систем је запамтио извођача“. (ИА)



**Алтернативна сценарија**

2.1 Уколико систем не може да креира извођача он приказује организатору поруку: “Систем не може да запамти извођача”. Прекида се извршење сценарија. (ИА)

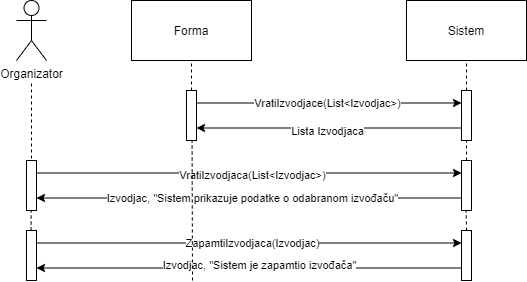


Са наведених секвенцних дијаграма уочавају се системске операције које треба пројектовати:

1. signal **SacuvajIzvodjaca**(Izvodjac)

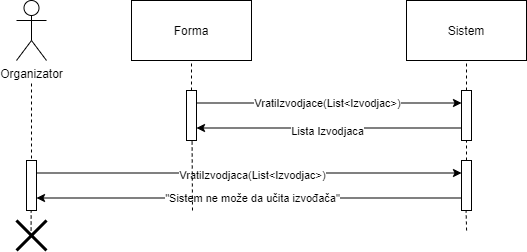
### ДС3: Дијаграм секвенци случаја коришћења – Измена извођача

1. Форма **позива** систем да учита податке о извођачима. (АПСО)
2. Систем **приказује** организатору податке о извођачу и поруку: “Систем приказује податке о одабраном извођачу”. (ИА)
3. Организатор **позива** систем да запамти податке о извођачу. (АПСО)
4. Систем **приказује** организатору запамћени и поруку: “Систем је запамтио извођача.” (ИА)

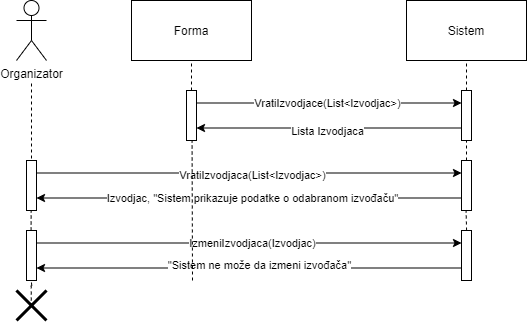


**Алтернативна сценарија**

2.1 Уколико систем не може да учита податке о извођачу он приказује организатору поруку: “Систем не може да учита извођача”. Прекида се извршење сценарија. (ИА)



4.1 Уколико систем не може да запамти податке о извођачу он приказује организатору поруку “Систем не може да запамти измењеног извођача”. (ИА)

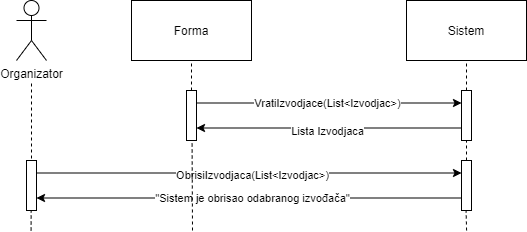


Са наведених секвенцних дијаграма уочавају се системске операције које треба пројектовати:

1. signal **VratiIzvodjace**()  
2. signal **ZapamtiIzvodjaca**(Izvodjac)

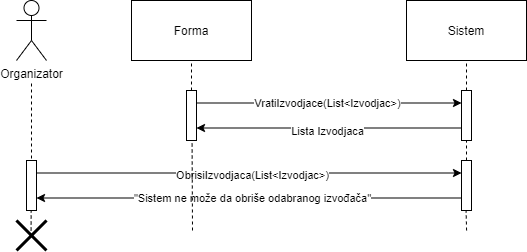
### ДС4: Дијаграм секвенци случаја коришћења – Брисање извођача

1. Организатор **позива** систем да учита податке о извођачима. (АПСО)
2. Систем **приказује** кориснику све податке о извођачу и поруку: “Систем приказује податке о одабраном извођачу“. (ИА)
3. Организатор **позива** систем да брише извођача. (АПСО)
4. Систем **приказује** организатору поруку: “Систем је обрисао извођача.” (ИА)



**Алтернативна сценарија**

4.1 Уколико систем не може да учита податке о извођачу он приказује организатору поруку: “Систем не може да учита извођача”. Прекида се извршење сценарија. (ИА)

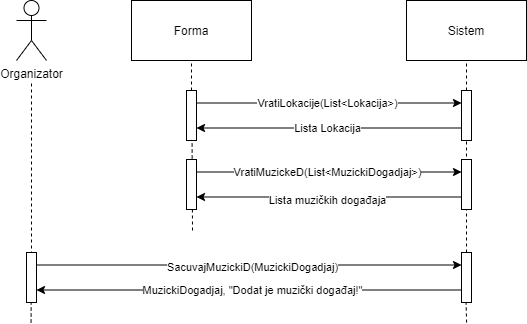


Са наведених секвенцних дијаграма уочавају се системске операције које треба пројектовати:

1. signal **VratiIzvodjace**()  
2. signal **ObrisiIzvodjaca**(Izvodjac)

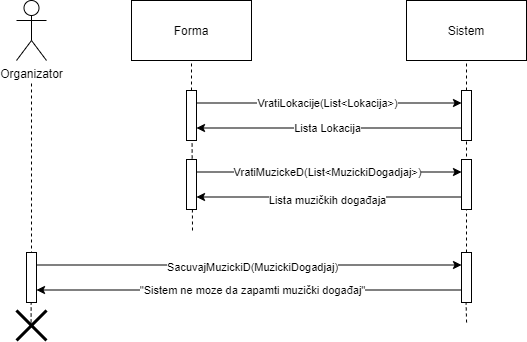
### ДС5: Дијаграм секвенци случаја коришћења – Унос музичког догађаја

1. Форма **позива** систем да учита листу локација. (АПСО)
2. Систем враћа форми листу локација. (ИА)
3. Организатор **позива** систем да запамти податке о музичком догађају. (АПСО)
4. Систем **приказује** организатору запамћени и поруку: “Додат је музички догађај“. (ИА)



**Алтернативна сценарија**

6.1 Уколико систем не може да запамти музички догађај он приказује организатору поруку: “Систем не може да запамти музички догађај”. Прекида се извршење сценарија. (ИА)

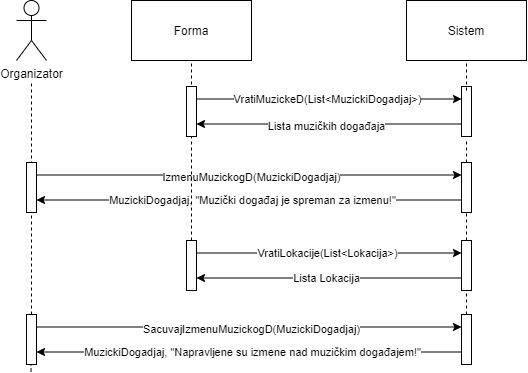


Са наведених секвенцних дијаграма уочавају се системске операције које треба пројектовати:

1. signal **VratiLokacije**()  
2. signal **VratiMuzickeD**()  
3. signal **SacuvajMuzickiD** (MuzickiDogadjaj)

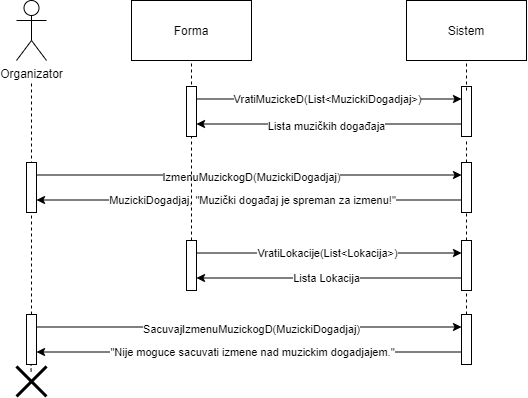
### ДС6: Дијаграм секвенци случаја коришћења – Измена музичког догађаја

1. Форма **позива** систем да учита листу музичких догађаја. (АПСО)
2. Систем враћа форми листу музичких догађаја. (ИА)
3. Организатор **позива** систем да учита податке о музичком догађају. (АПСО)
4. Систем **обавештава** организатора о успешном учитавању и поруку: “Музички догађај је спреман за измену“. (ИА)
5. Форма **позива** систем да учита листу локација. (АПСО)
6. Систем враћа форми листу локација. (ИА)
7. Организатор **позива** систем да запамти измењене податке о музичком догађају. (АПСО)
8. Систем **приказује** организатору запамћени и поруку: “ Направљене су измене над музичким догађајем“. (ИА)



**Алтернативна сценарија**

8.1 Уколико систем не може да запамти музички догађај он приказује организатору поруку: “Није могуће сачувати измене над музичким догађајем”. Прекида се извршење сценарија. (ИА)

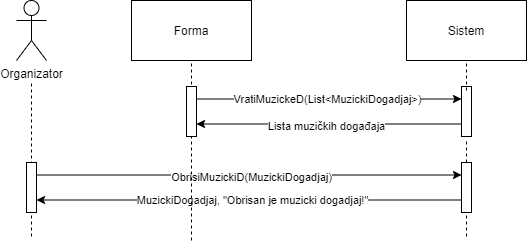


Са наведених секвенцних дијаграма уочавају се системске операције које треба пројектовати:

1. signal **VratiLokacije**()  
2. signal **VratiMuzickeD**()  
3. signal **SacuvajMuzickiD** (MuzickiDogadjaj)

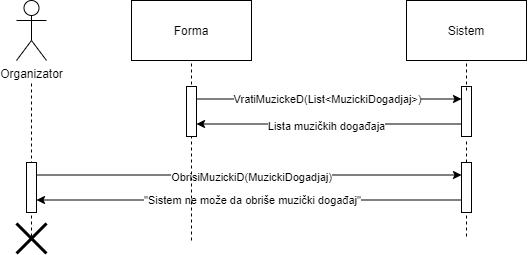
### **ДС7: Дијаграм секвенци случаја коришћења – Брисање музичког догађаја**

1. Форма **позива** систем да учита листу музичких догађаја. (АПСО)
2. Систем **приказује** листу музичких догађаја.
3. Организатор **позива** систем да обрише музички догађај. (АПСО)
4. Систем **приказује** организатору поруку: “Систем је обрисао музички догађај.” (ИА)



**Алтернативна сценарија**

4.1 Уколико систем не може да обрише музички догађај он приказује организатору поруку: “Систем не може да обрише музички догађај”. Прекида се извршење сценарија. (ИА)

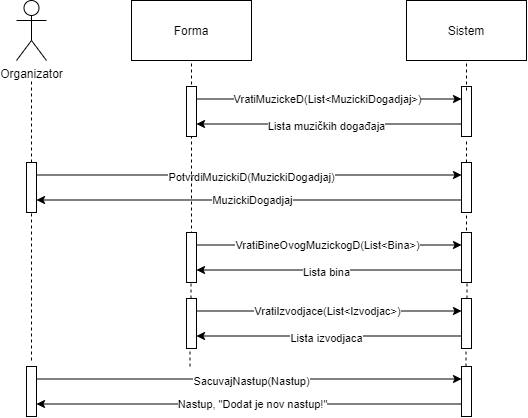


Са наведених секвенцних дијаграма уочавају се системске операције које треба пројектовати:

1. signal **VratiMuzickeD**()  
2. signal **ObrisiMuzickiD**(MucickiDogadjaj)

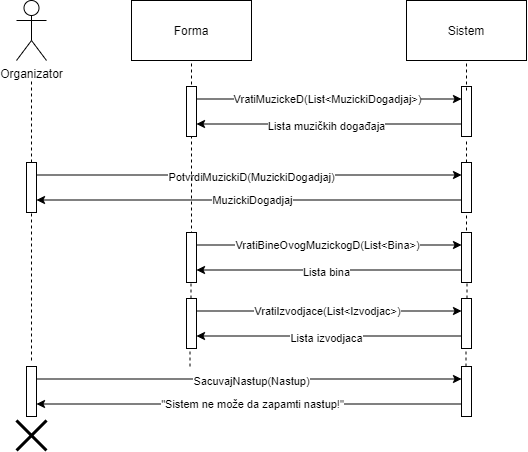
### ДС8: Дијаграм секвенци случаја коришћења – Унос наступа

1. Форма **позива** систем да учита листу музичких догађаја. (АПСО)
2. Систем враћа форми листу музичких догађаја. (ИА)
3. Организатор **позива** систем да одабере за измену музички догађај. (АПСО)
4. Систем **приказује** организатору одабрани музички догађај. (ИА)
5. Форма **позива** систем да учита листу бина. (АПСО)
6. Систем враћа форми листу бина. (ИА)
7. Форма **позива** систем да учита листу извођача. (АПСО)
8. Систем враћа форми листу извођача. (ИА)
9. Организатор **позива** систем да запамти податке о наступу. (АПСО)
10. Систем **приказује** организатору запамћени и поруку: “Додат је нов наступ“. (ИА)



**Алтернативна сценарија**

10.1 Уколико систем не може да запамти нов наступ он приказује организатору поруку: “Систем не може да запамти нов наступ”. Прекида се извршење сценарија. (ИА)



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

1. signal **VratiMuzickeD**()  
2. signal **VratiBineOvogMuzickogD**(MuzickiDogadjaj)  
3. signal **VratiIzvodjace**()  
4. signal **SacuvajNastup**(Nastup)

## Понашање софтверског система – дефинисање уговора о системским операцијама

Као резултат анализе сценарија, добијене су следеће системске операције које треба пројектовати:

1. signal **PrijaviOrganizatora**(Organizator)  
2. signal **SacuvajIzvodjaca**(Izvodjac)  
3. signal **VratiIzvodjace**()  
4. signal **ObrisiIzvodjaca**(Izvodjac)  
5. signal **VratiLokacije**()  
6. signal **SacuvajMuzickiD**(MuzickiDogadjaj)  
7. signal **VratiMuzickeD**()  
8. signal **ObrisiMuzickiD** (MuzickiDogadjaj)  
9. signal **VratiBineOvogMuzickogD**(MuzickiDogadjaj)  
10. signal **SacuvajNastup**(Nastup)

### Уговор УГ1: PrijaviOrganizatora

Операција: PrijaviOrganizatora(Organizator) : organizator;  
Веза са СК: СК1  
Предуслови: - /  
Постуслови: - /

### Уговор УГ2: SacuvajIzvodjaca

Операција: SacuvajIzvodjaca(Izvodjac) : signal;  
Веза са СК: СК2, СК3  
Предуслови: Вредносна и структурна ограничења над објектом Изводјац морају бити задовољена  
Постуслови: Подаци о извођачу су сачувани

### Уговор УГ3: VratiIzvodjace

Операција: VratiIzvodjace() : List<Izvodjac>;  
Веза са СК: СК3, СК4, СК8  
Предуслови: - /  
Постуслови: - /

### Уговор УГ4: ObrisiIzvodjaca

Операција: ObrisiIzvodjaca(Izvodjac) : signal;  
Веза са СК: СК4  
Предуслови: - Вредносна и структурна ограничења над објектом Извођач морају бити задовољена  
Постуслови: - Изабрани извођач је деактивиран

### Уговор УГ5: VratiLokacije

Операција: VratiLokacije() : List<Lokacija>;  
Веза са СК: СК5, СК6, СК8  
Предуслови: - /  
Постуслови: - /

### Уговор УГ6: SacuvajMuzickiD

Операција: SacuvajMuzickiD(MuzickiDogadjaj) : signal;  
Веза са СК: СК5, СК6  
Предуслови: Вредносна и структурна ограничења над објектом Догадјај морају бити задовољена  
Постуслови: Унет је нови музички догађај

### Уговор УГ7: VratiMuzickeD

Операција: VratiMuzickeD(): List<MuzickiDogadjaj>;  
Веза са СК: СК5, СК6, СК7  
Предуслови: - /  
Постуслови: - /

### Уговор УГ8: ObrisiMuzickiD

Операција: ObrisiMuzickiD(MuzickiDogadjaj) : signal;  
Веза са СК: СК7  
Предуслови: - Вредносна и структурна ограничења над објектом Догадјај морају бити задовољена  
Постуслови: - Изабрани догађај је деактивиран

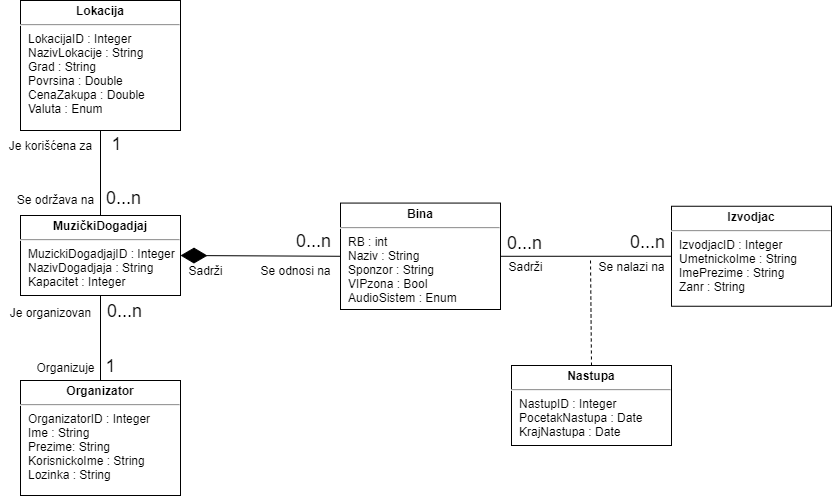
### Уговор УГ9: VratiBineOvogMuzickogD

Операција: VratiBineOvogMuzickogD(MuzickiDogadjaj) : List<Bina >;  
Веза са СК: СК8  
Предуслови: - /  
Постуслови: - /

### Уговор УГ10: SacuvajNastup

Операција: SacuvajNastup(Nastup) : signal;  
Веза са СК: СК8  
Предуслови: Вредносна и структурна ограничења над објектом Наступ морају бити задовољена  
Постуслови: Унет је нови наступ

## Структура софтверског система – Концептуални (доменски) модел



Слика 2 - Концептуални модел

## Stuktura softverskog sistema – Relacioni model

**MuzickiDogadjaj** (MuzickiDogadjajID, NazivDogadjaja, Kapacitet, *OrganizatorID*, *LokacijaID*)  
**Organizator** (OrganizatorID, Ime, Prezime, KorisnickoIme, Lozinka)  
**Lokacija** (LokacijaID, NazivLokacije, Grad, Povrsina, CenaZakupa, Valuta)  
**Bina** (*MuzickiDogadjajID*, RB, Naziv, Sponzor, VIPzona, AudioSistem)  
**Izvodjac** (IzvodjacID, UmetnickoIme, Ime, Prezime, Zanr)  
**Nastup** (NastupID, PocetakNastupa, KrajNastupa, *MuzickiDogadjajID*, *RB*, *IzvodjacID*)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Табела :** MuzickiDogadjaj | | **Просто вредносно ограничење** | | **Сложено вредносно ограницење** | | **Структурно ограничење** |
| **Атрибути** | **Име** | **Тип атрибута** | **Вредност атрибута** | **Међузависност атрибута једне табеле** | **Међузависност атрибута више табела** | **INSERT** RESTRICTED Organizator, Lokacija  **UPDATE** RESTRICTED Organizator, Lokacija CASCADES Bina  **DELETE** CASCADES Bina |
| MuzickiDogadjajID | Integer | not null and >0 |  |  |
| NazivDogadjaja | String | not null |  |  |
| Kapacitet | Integer | not null and >0 |  |  |
| OrganizatorID | Integer | not null and >0 |  |  |
| LokacijaID | Integer | not null and >0 |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Табела:** Organizator | | **Просто вредносно ограничење** | | **Сложено вредносно ограницење** | | **Структурно ограничење** |
| **Атрибути** | **Име** | **Тип атрибута** | **Вредност атрибута** | **Међузависност атрибута једне табеле** | **Међузависност атрибута више табела** | **INSERT** /  **UPDATE** CASCADES MuzickiDogadjaj  **DELETE** RESTRICTED MuzickiDogadjaj |
| OrganizatorID | Integer | not null and >0 |  |  |
| Ime | String | not null |  |  |
| Prezime | String | not null |  |  |
| KorisnickoIme | String | not null |  |  |
| Lozinka | String | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Табела:** Lokacija | | **Просто вредносно ограничење** | | **Сложено вредносно ограницење** | | **Структурно ограничење** |
| **Атрибути** | **Име** | **Тип атрибута** | **Вредност атрибута** | **Међузависност атрибута једне табеле** | **Међузависност атрибута више табела** | **INSERT** /  **UPDATE** CASCADES MuzickiDogadjaj  **DELETE** RESTRICTED MuzickiDogadjaj |
| LokacijaID | Integer | not null and >0 |  |  |
| NazivLokacije | String | not null |  |  |
| Grad | String | not null |  |  |
| Povrsina | Integer | not null and >0 |  |  |
| CenaZakupa | Double | not null and >0 |  |  |
| Valuta | Enum | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Табела:** Bina | | **Просто вредносно ограничење** | | **Сложено вредносно ограницење** | | **Структурно ограничење** |
| **Атрибути** | **Име** | **Тип атрибута** | **Вредност атрибута** | **Међузависност атрибута једне табеле** | **Међузависност атрибута више табела** | **INSERT** RESTRICTED MuzickiDogadjaj  **UPDATE** RESTRICTED MuzickiDogadjaj CASCADES Nastup  **DELETE** RESTRICTED Nastup |
| MuzickiDogadjajID | Integer | not null and >0 |  |  |
| RB | Integer | not null and >0 |  |  |
| Naziv | String | not null |  |  |
| Sponzor | String | not null |  |  |
| VIPzona | Bool | not null |  |  |
| AudioSistem | Enum | not null |  |  |

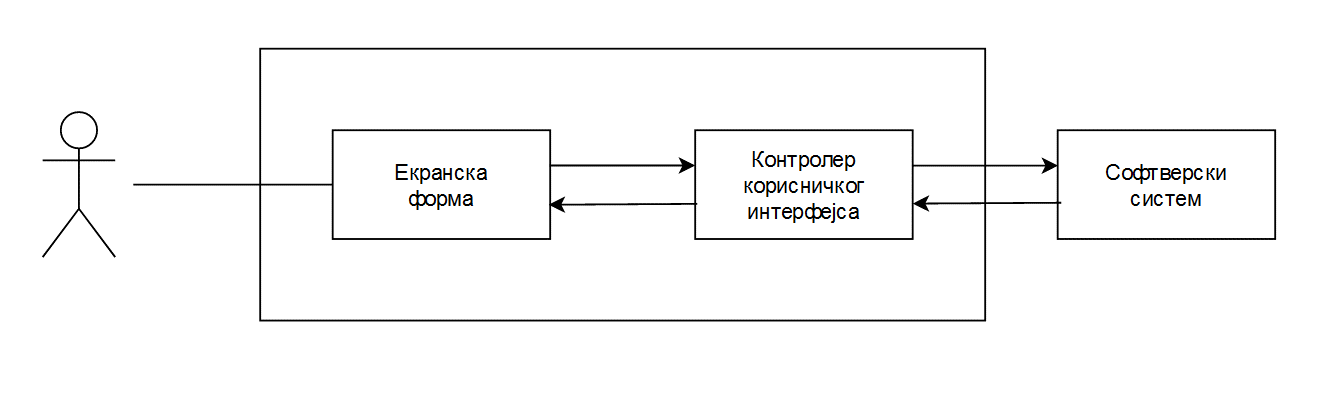
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Табела:** Izvodjac | | **Просто вредносно ограничење** | | **Сложено вредносно ограницење** | | **Структурно ограничење** |
| **Атрибути** | **Име** | **Тип атрибута** | **Вредност атрибута** | **Међузависност атрибута једне табеле** | **Међузависност атрибута више табела** | **INSERT** /  **UPDATE** CASCADES Nastup  **DELETE** RESTRICTED Nastup |
| IzvodjacID | Integer | not null and >0 |  |  |
| UmetnickoIme | String | not null |  |  |
| ImePrezime | String | not null |  |  |
| Zanr | String | not null |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Табела:** Nastup | | **Просто вредносно ограничење** | | **Сложено вредносно ограницење** | | **Структурно ограничење** |
| **Атрибути** | **Ime** | **Date** | **Vrednost atributa** | **Међузависност атрибута једне табеле** | **Међузависност атрибута више табела** | **INSERT** RESTRICTED Bina, Izvodjac  **UPDATE** RESTRICTED Bina, MuzickiDogadjaj  **DELETE** / |
| NastupID | Integer | not null and >0 |  |  |
| MuzickiDogadjajID | Integer | not null and >0 |  |  |
| RB | Integer | not null and >0 |  |  |
| IzvodjacID | Integer | not null and >0 |  |  |
| PocetakNastupa | Date | not null |  |  |
| KrajNastupa | Date | not null |  |  |

# ПРОјектовање

## Пројектовање корисничког интерфејса

Кориснички интерфејс се састоји од екранских форми и контролера корисничког интерфејса. Екранска форма има улогу да прихвата податке које корисник уноси, прихвата догађаје које корисник прави, позива контролера корисничког интерфејса и приказује податке које добија као резултат од контролера. Корисничка форма нема референцу на Домен.



Сценарији у којима се користе екранске форме директно су повезане са сценаријима случајева коришћења.

### СК1: Случај коришћења – Пријављивање организатора

**Назив СК**

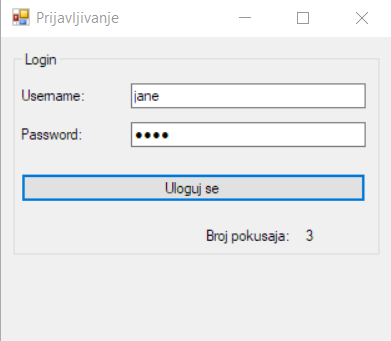
Пријављивање организатора  
  
**Актори СК**

Организатор

**Учесници СК**

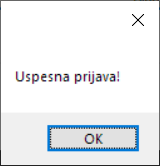
Организатор и систем (програм)

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



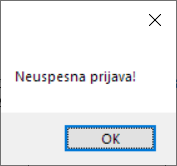
**Основни сценарио СК**

1. Организатор **уноси** податке за пријављивање (корисничкно име и лозинку). (АПУСО)
2. Организатор **контролише** да ли је исправно унео податке за пријављивање.(АНСО)
3. Организатор **позива** систем да пронађе корисника са датим подацима.(АПСО)
4. Систем **претражује** кориснике. (СО)
5. Систем **приказује** организаторупоруку: ‘Успешна пријава!‘’ и омогућава приступ систему. (ИА)



Алтернативна сценарија

5.1 Уколико систем не може да нађе корисника приказује поруку: “Неуспешна пријава!”.(ИА)



### СК2: Случај коришћења – Унос извођача

**Назив СК**

Унос извођача

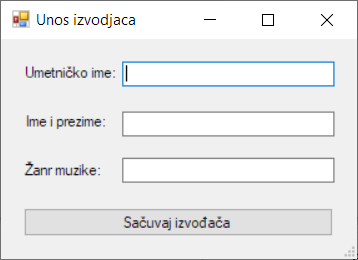
**Актори СК**

Организатор

**Учесници СК**

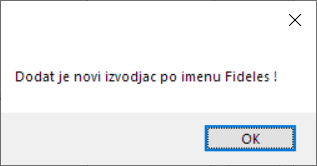
Организатор и систем (програм)

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



**Основни сценарио СК**

1. Организатор **уноси** податке о извођачу. (АПУСО)
2. Организатор **контролише** да ли је коректно унео податке о извођачу. (АНСО)
3. Организатор **позива** систем да запамти податке о извођачу. (АПСО)
4. Систем **памти** податке о извођачу. (СО)
5. Систем **приказује** организатору запамћеног извођача и поруку: “Додат је нови извођач по имену [име]“. (ИА)



Алтернативна сценарија

5.1 Уколико систем не може да креира извођача он приказује организатору поруку: “Систем не може да креира извођача”. Прекида се извршење сценарија. (ИА)

### СК3: Случај коришћења – Измена извођача

**Назив СК**

Измена извођача

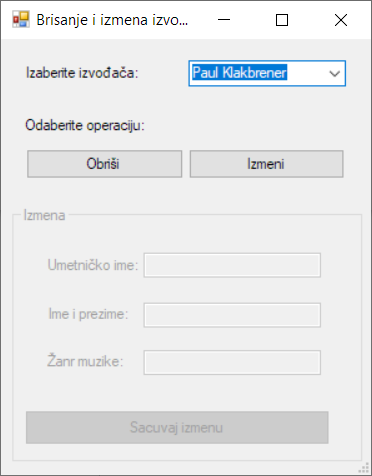
**Актори СК**

Организатор

**Учесници СК**

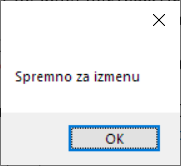
Организатор и систем (програм)

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

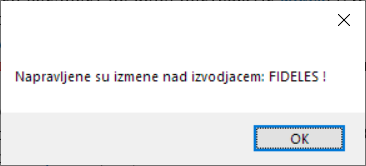


**Основни сценарио СК**

1. Организатор **бира** извођача ког жели да измени. (АПУСО)
2. Организатор **позива** систем да учита податке о одабраном извођачу. (АПСО)
3. Систем **учитава** податке о одабраном извођачу. (СО)
4. Систем **приказује** организатору податке о извођачу и поруку: “Спремно за измену”. (ИА)



1. Организатор **мења** податке о извођачу. (АПУСО)
2. Организатор **контролише** да ли је коректно унео податке о извођачу. (АНСО)
3. Организатор **позива** систем да запамти податке о извођачу. (АПСО)
4. Систем **памти** податке о извођачу. (СО)
5. Систем **приказује** организатору запамћеног извођача и поруку: “Направљене су измене над извођачем: [Име]” (ИА)



Алтернативна сценарија

4.1 Уколико систем не може да учита податке о извођачу он приказује организатору поруку: “Систем не може да учита извођача”. Прекида се извршење сценарија. (ИА)

9.1 Уколико систем не може да запамти податке о извођачу он приказује организатору поруку “Систем не може да запамти извођача”. (ИА)

### СК4: Случај коришћења – Брисање извођача

**Назив СК**

Брисање извођача

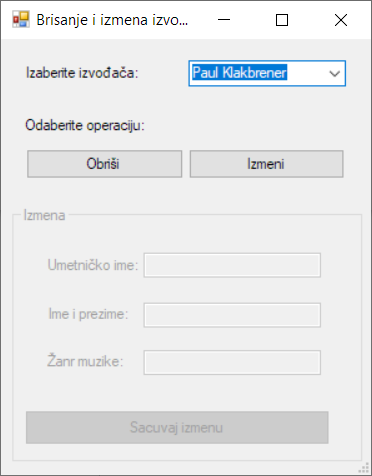
**Актори СК**

Организатор

**Учесници СК**

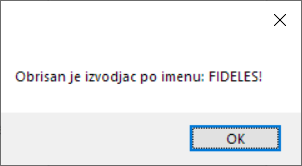
Организатор и систем (програм)

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



**Основни сценарио СК**

1. Организатор **бира** извођача ког жели да измени. (АПУСО)
2. Организатор **позива** систем да обрише извођача. (АПСО)
3. Систем **деактивира** извођача. (СО)
4. Систем **приказује** организатору поруку: “Систем је обрисао извођача.” (ИА)



Алтернативна сценарија

4.1 Уколико систем не може да деактивира извођача он приказује организатору поруку “Систем не може да обрише извођача”. (ИА)

### СК5: Случај коришћења – Унос музичког догађаја (Сложен СК)

**Назив СК**

Унос музичког догађаја

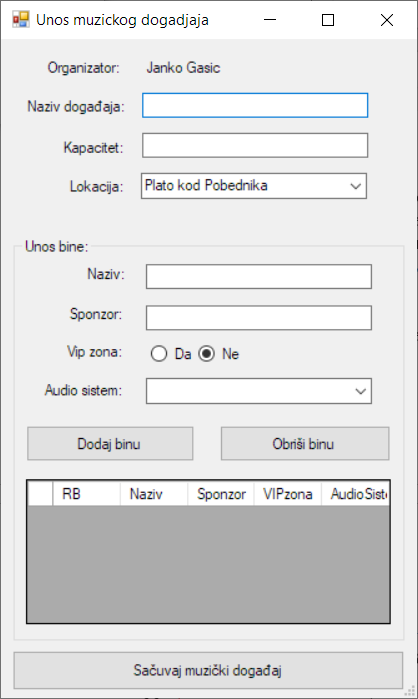
**Актори СК**

Организатор

**Учесници СК**

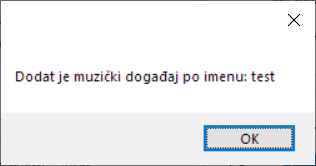
Организатор и систем (програм)

**Предуслов**: Систем је укључен и организатор је улогован под својом шифром. Систем приказује форму за рад са музичким догађајем. Систем је учитао листу локација.



**Основни сценарио СК**

1. Организатор **уноси** податке о музичком догађају и бинама. (АПУСО)
2. Организатор **контролише** да ли је коректно унео податке о музичком догађају и бинама. (АНСО)
3. Организатор **позива** систем да запамти податке о музичком догађају и бинама. (АПСО)
4. Систем **памти** податке о музичком догађају и бинама. (СО)
5. Систем **приказује** организатору запамћени догађај и поруку: “Систем је запамтио податке о музичком догађају“. (ИА)



Алтернативна сценарија

5.1 Уколико систем не може да запамти податке о музичком догађају он приказује организатору поруку “Систем не може да запамти податке о музичком догађају ”. (ИА)

### СК6: Случај коришћења – Измена музичког догађаја (Сложени СК)

**Назив СК**

Измена музичког догађаја

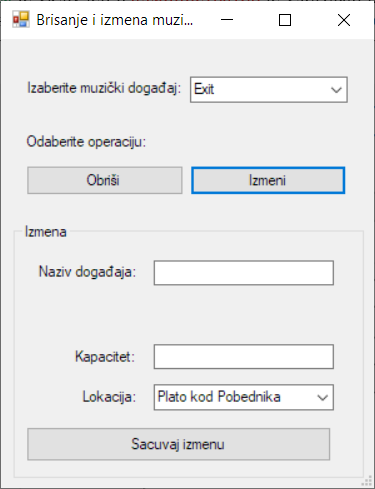
**Актори СК**

Организатор

**Учесници СК**

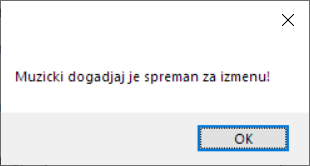
Организатор и систем (програм)

**Предуслов**: Систем је укључен и организатор је улогован под својом шифром. Систем приказује форму за рад са музичким догађајем. Систем је учитао листу локација.

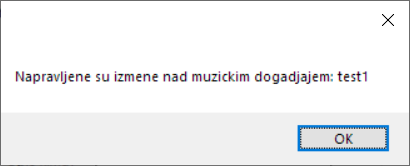


**Основни сценарио СК**

1. Организатор **бира** музички догађај ког жели да измени. (АПУСО)
2. Организатор **позива** систем да учита податке о одабраном музичком догађају. (АПСО)
3. Систем **учитава** податке о одабраном музичком догађају. (СО)
4. Систем **приказује** организатору податке о музичком догађају и поруку: “Музички догађај је спреман за измену”. (ИА)



1. Организатор **мења** податке о музичком догађају. (АПУСО)
2. Организатор **контролише** да ли је коректно унео податке о музичком догађају. (АНСО)
3. Организатор **позива** систем да запамти податке о музичком догађају. (АПСО)
4. Систем **памти** податке о музичком догађају. (СО)
5. Систем **приказује** организатору запамћени догађај и поруку: “Напревљене су измене над музичким догађајем: [Име]” (ИА)



Алтернативна сценарија

9.1 Уколико систем не може да запамти податке о музичком догађају он приказује организатору поруку “Систем не може да запамти музички догађај”. (ИА)

### СК7: Случај коришћења – Брисање музичког догађаја

**Назив СК**

Брисање музичког догађаја

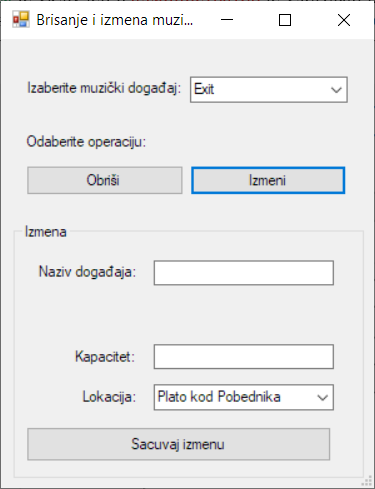
**Актори СК**

Организатор

**Учесници СК**

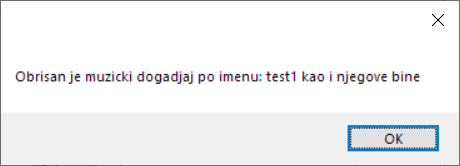
Организатор и систем (програм)

**Предуслов**: Систем је укључен и организатор је улогован под својом шифром. Систем приказује форму за рад са музичким догађајем.



**Основни сценарио СК**

1. Организатор **бира** музички догађај кој жели да измени. (АПУСО)
2. Организатор **позива** систем да обрише музички догађај. (АПСО)
3. Систем **деактивира** музички догађај. (СО)
4. Систем **приказује** организатору поруку: “Систем је обрисао музички догађај.” (ИА)



Алтернативна сценарија

4.1 Уколико систем не може да деактивира музички догађај он приказује организатору поруку “Систем не може да обрише музички догађај”. (ИА)

### СК8: Случај коришћења – Унос наступа (Сложен СК)

**Назив СК**

Унос наступа

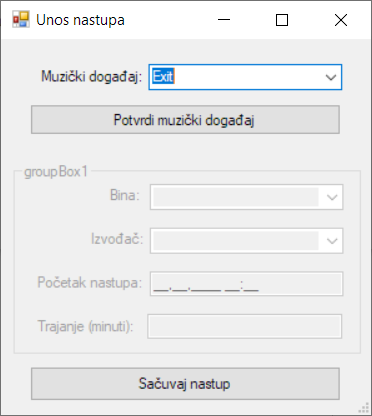
**Актори СК**

Организатор

**Учесници СК**

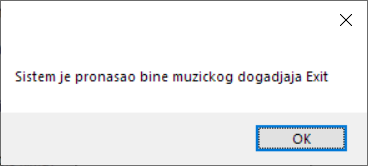
Организатор и систем (програм)

**Предуслов**: Систем је укључен и организатор је улогован под својом шифром. Систем приказује форму за рад са наступом. Систем је учитао листу музичких догађаја.

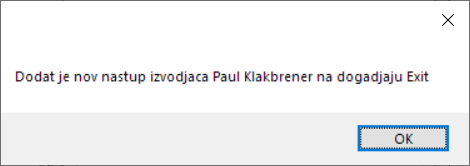


**Основни сценарио СК**

1. Организатор **бира** музички догађај. (АПУСО)
2. Организатор **позива** систем да потврди музички догађај. (АПСО)
3. Систем **учитава** податке о одабраном музичком догађају (бине) и извођачима. (СО)
4. Систем **приказује** организатору музички догађај и поруку: “Систем је пронашао бине музичког догађаја [Име]“. (ИА)



1. Организатор **уноси** податке о наступу. (АПУСО)
2. Организатор **контролише** да ли је коректно унео податке о наступу. (АНСО)
3. Организатор **позива** систем да запамти податке о наступ. (АПСО)
4. Систем **памти** податке о наступу. (СО)
5. Систем **приказује** организатору запамћени догађај и поруку: “Додат је нов наступ извођача [Име извођача] на догађају [Назив догађаја]“. (ИА)



Алтернативна сценарија

9.1 Уколико систем не може да запамти податке о наступу он приказује организатору поруку “Систем не може да запамти податке о наступу ”. (ИА)

## Пројектовање контролера корисничког интерфејса

Контролер корисничког интерфејса је одговоран за:

* Прихватање података које шаље форма
* Конвертовање податак у објекат који представља улазни аргумент системске операције
* Шаље захтев и прослеђује објекат комуникацији
* Прихвата објекат који настје као одговор сервера
* Конвертује објекат и његове елементе у графичке елементе и приказује их на форми

Контролер има улогу посредника између екранских форми и комуникације, прецизније екранских форми и самог система. За сваку екрануску форму постоји по један имплементирани контролер.

## Пројектовање апликационе логике

У оквиру апликационе логике пројектују се контролер апликационе логике, пословна логика и брокер базе података.

### Пројектовање контролера апликационе логике

Када говоримо о пројектовању апликационе логике говоримо конкретно о класама *Сервер*, *НитКлијента* и *Контролер*.

#### Сервер

*Сервер* је задужен за подизање серверског сокета који ослушкује мрежу. У моменту када клијентски сокет успостави везу са серверским сокетом, класа сервер ће генерисати посебну нит која служи за двосмерну комуникацију између клијента и сервера. Такође сервер ће генерисати нову класу *НитКлијента* и њој ће проследити новогенерисану нит за комуникацију. Класа *сервер* са клијентом има комуникацију тачно једном, када се он први пут повеже на сервер.

#### НитКлијента

*НитКлијента*, након што је генерисана и након што је примила нит преко које ће комуницирати са клијентом има задатак да преко те нити врши комуникацију. Константно ће очекивати од корисника предефинисану класу *Захтев* у којој ће бити објекат и операција која треба да се изврши над тим објектом. Док ће као одговор слати класу *Одговор* која у себи има тражени објекат, поруку и сигнал. Након што *НитКлијента* прими објекат и претвори га у одговарајући тип, а то зависи у односу на операцију коју треба и извршити, она прослеђује објекат у одговарајућу медоту *Контролера*.

#### Контролер

За разлику од НитиКлијената која постоји по једна за сваког корисника, контролер постоји само један једини за сваку нит клијента. Он од *НитиКлијента* прима одговарајући објекат и у зависности од операције која је прочитана у *НитиКлијента,* позива одговарајућу системску операцију коју треба извршити. Такође, *Контролер* након извршене операције, прослеђује *НитиКлијента* одговор како би га она вратила кориснику.

### Пословна логика

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

#### Пројектовање понашања система – Системске операције

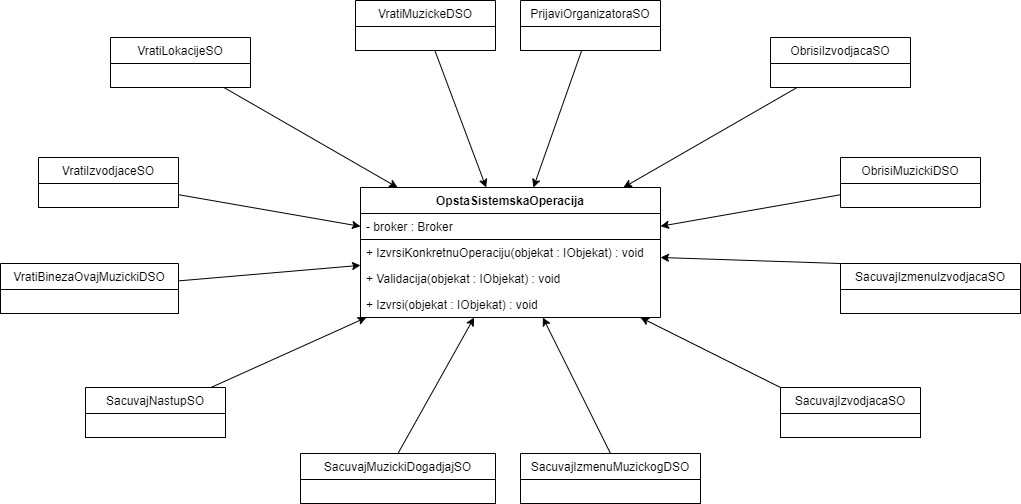
Пројектовање понашања система укључује пројектовање опште класе и подкласа које представљају системсе операције и одговорне су за њихова извршења. Те подкласе наслеђују *OpstuSistemskuOperaciju* која дефинише алгоритам извршавања системски операција.

*OpstaSistemskaOperacija* креира једну инстанцу *Brokera* и има референцу ка њој. Она садржи апстрактне методе *IzvrisKonkretnuOperaciju* и *Validacija*. Ове методе морају бити имплеметиране и у њима мора експлицитно бити наведено на који начин ће се оне извршавати у свакој системској операцији која наслеђује општу. Алготирам извршавања налази се у методи *Izvrsi* која је public и која се позива из *контролера* (*Контролер* има референцу на *OpstuSistemskuOperaciju*).

Метода *Izvrsi* у себи садржи алгоритам извршавања који се састоји од:

* *Validacija*(objekat) – проверава се да ли је општи објекат који се прима као параметар типа који препознаје конкретна системска апликација
* *Broker*.*OtvoriKonekciju*() – отвара се конекција између базе и брокера
* *Broker*.*PokreniTransakciju*() – покреће се трансакција
* *IzvrsiKonkretnuOperaciju*(objekat) – извршава се конкретна операција над објектом који је прослеђен. Касније ће бити више речено о самим операцијама
* *Broker*.*CommitTransakcije*() – уколико није дошло до грешака, целокуп на трансакција се комитује у базу податакља
* *Broker*.*RollbackTransakcije*() – налази се у catch делу, извршава се уколико се догодина нека грешка при претходним операцијама. Поништава комплетну трансакцију и шаље серверу, а касније клијенту одговарајучу поруку.
* *Broker*.*ZatvoriKonekciju*() – налази се у finally делу, увек се извршава било да је трансакција прошла са или без грешака. Увек затвара конекцију која се отворила у другом кораку.

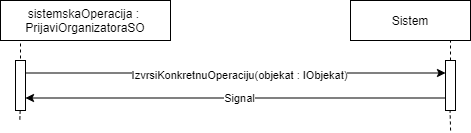
За сваку системску операцију треба направити концептуално решење које је директо повезано са логиком самог проблема. Начин на који ће се одређени проблем решити је дефинисам у апстрактној методи *IzvrsiKonkretnuOperaciju(objekat).* За сваку СО ова медота је различито имплементирана и у њој се позива одговарајућа метода *брокера* која решава дати проблем или даје одговор на конкретан упит.



Slika 3 - Sistemske operacije

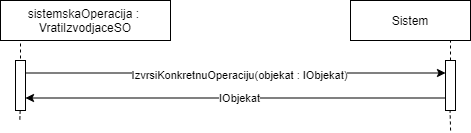
### Уговор УГ1: PrijaviOrganizatora

Операција: PrijaviOrganizatora(Organizator) : organizator;  
Веза са СК: СК1  
Предуслови: - /  
Постуслови: - /



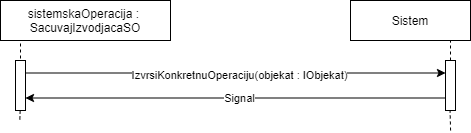
### Уговор УГ2: SacuvajIzvodjaca

Операција: SacuvajIzvodjaca(Izvodjac) : signal;  
Веза са СК: СК2, СК3  
Предуслови: Вредносна и структурна ограничења над објектом Изводјац морају бити задовољена  
Постуслови: Подаци о извођачу су сачувани



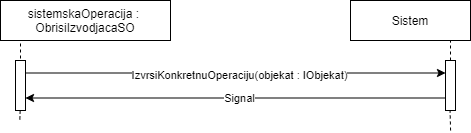
### Уговор УГ3: VratiIzvodjace

Операција: VratiIzvodjace() : List<Izvodjac>;  
Веза са СК: СК3, СК4, СК8  
Предуслови: - /  
Постуслови: - /



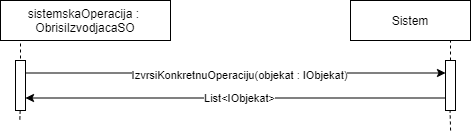
### Уговор УГ4: ObrisiIzvodjaca

Операција: ObrisiIzvodjaca(Izvodjac) : signal;  
Веза са СК: СК4  
Предуслови: - Вредносна и структурна ограничења над објектом Извођач морају бити задовољена  
Постуслови: - Изабрани извођач је деактивиран



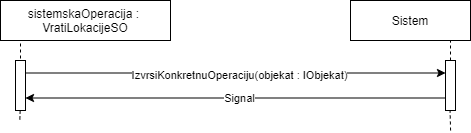
### Уговор УГ5: VratiLokacije

Операција: VratiLokacije() : List<Lokacija>;  
Веза са СК: СК5, СК6, СК8  
Предуслови: - /  
Постуслови: - /



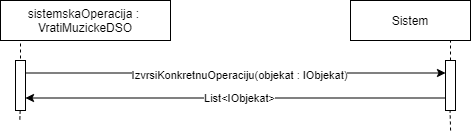
### Уговор УГ6: SacuvajMuzickiD

Операција: SacuvajMuzickiD(MuzickiDogadjaj) : signal;  
Веза са СК: СК5, СК6  
Предуслови: Вредносна и структурна ограничења над објектом Догадјај морају бити задовољена  
Постуслови: Унет је нови музички догађај



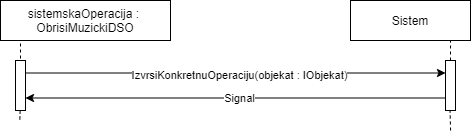
### Уговор УГ7: VratiMuzickeD

Операција: VratiMuzickeD(): List<MuzickiDogadjaj>;  
Веза са СК: СК5, СК6, СК7  
Предуслови: - /  
Постуслови: - /



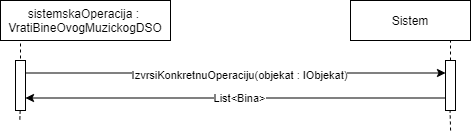
### Уговор УГ8: ObrisiMuzickiD

Операција: ObrisiMuzickiD(MuzickiDogadjaj) : signal;  
Веза са СК: СК7  
Предуслови: - Вредносна и структурна ограничења над објектом Догадјај морају бити задовољена  
Постуслови: - Изабрани догађај је деактивиран



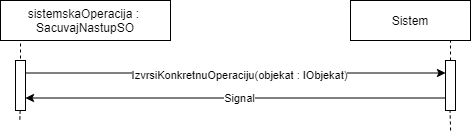
### Уговор УГ9: VratiBineOvogMuzickogD

Операција: VratiBineOvogMuzickogD(MuzickiDogadjaj) : List<Bina >;  
Веза са СК: СК8  
Предуслови: - /  
Постуслови: - /



### Уговор УГ10: SacuvajNastup

Операција: SacuvajNastup(Nastup) : signal;  
Веза са СК: СК8  
Предуслови: Вредносна и структурна ограничења над објектом Наступ морају бити задовољена  
Постуслови: Унет је нови наступ



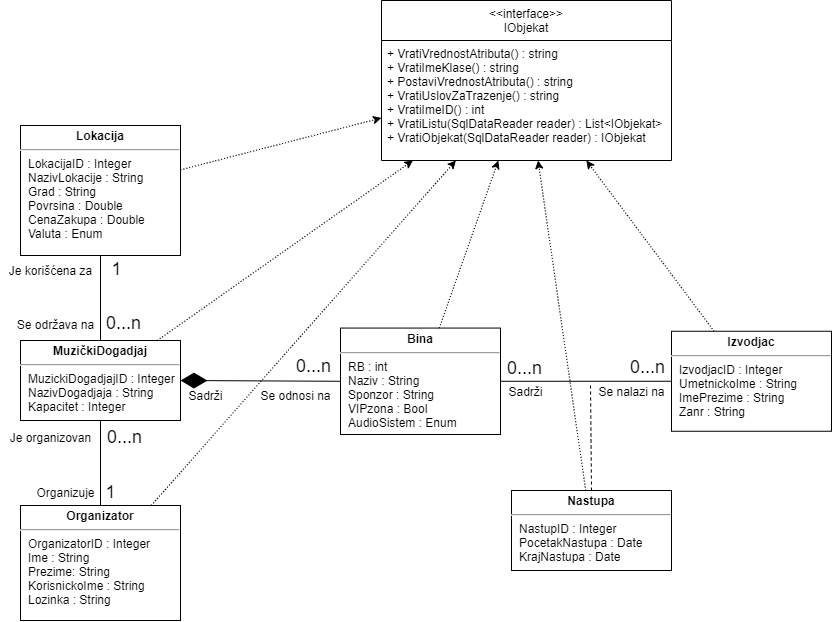
## Пројектовање структуре софтверског ситема – доменске класе

На основу структуре софтверског система, односно концептуалног модела праве се софтверске класе.

Свака класа доменска класа наслеђује *interfeace* коју у себи има наведене називе метода које свака доменска класа мора да имплементира, осим класа *Zahtev* и *Odgovor*. Медоте које су дефинисане у *interfejsu* односно класи *IObjekat* су:

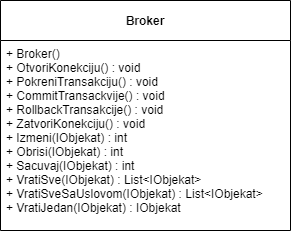
* string VratiVrednostAtributa();
* string VratiImeKlase();
* string PostaviVrednostAtributa();
* string VratiUslovZaTrazenje();
* string VratiImeID();
* List<IObjekat> VratiListu(SqlDataReader reader);
* IObjekat VratiObjekat(SqlDataReader reader);

Остале класе биће приказане на наредном дијаграму:



## Брокер базе података

Брокер базе података је одговоран за комуникацију између пословне логике и складишта података. Ову улогу врши класа *Broker.*



Све методе унутар класе *Broker* су пројектоване тако да буду генеричке. Овиме је обезбеђена униформност и није потребно пројектовати посебне методе за сваку доменску класу. Улазни аргумент сваке методе је интерфејс који свака доменска класа мора да имплементира.

Мотоде класе брокер:

public class Broker

{

private SqlConnection connection;

private SqlTransaction transaction;

public Broker() {}

public int Izmeni(IObjekat objekat) {}

public List<IObjekat> VratiSveSaUslovom(IObjekat objekat) {}

public int Obrisi(IObjekat objekat) {}

public List<IObjekat> VratiSve(IObjekat objekat) {}

public int Sacuvaj(IObjekat objekat) {}

public bool sacuvajFilm(IObjekat objekat) {}

public bool sacuvajRecenziju(IObjekat objekat) {}

public int DajSledeciID(IObjekat objekat) {}

public void OtvoriKonekciju() {}

public void PokreniTransakciju() {}

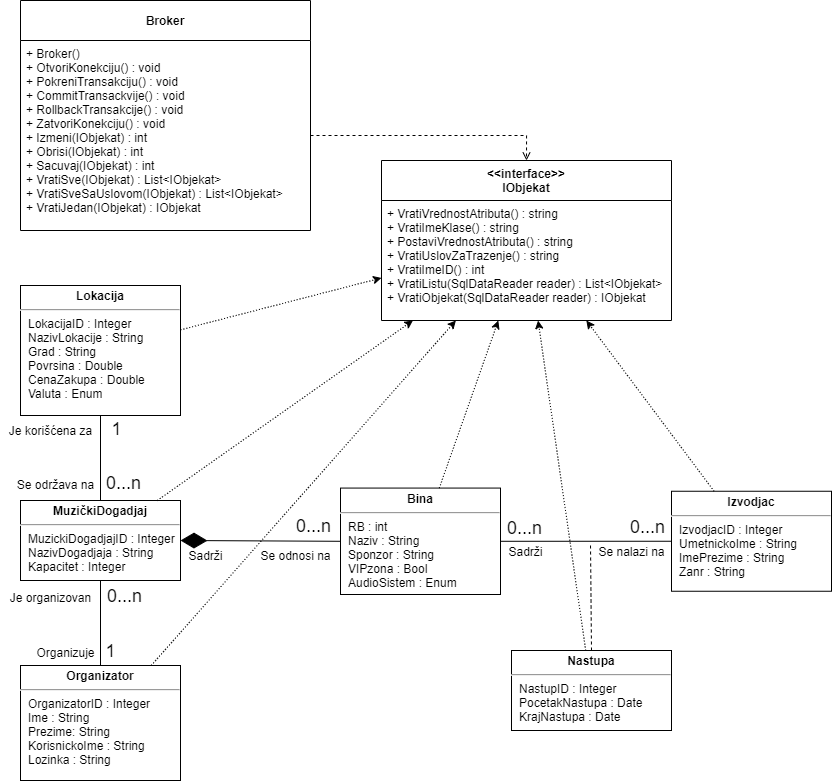
public void CommitTransakcije() {}

public void RollbackTransakcije() {}

public void ZatvoriKonekciju() {}

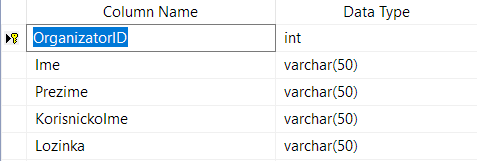
public IObjekat VratiJedan(IObjekat objekat) {}

}

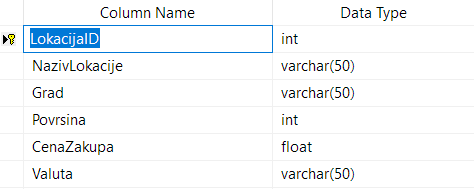


## Пројектовање складишта података

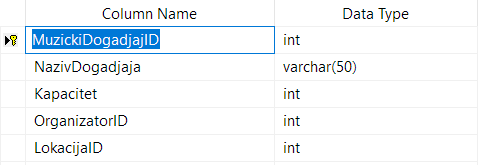
Складиште података је пројектована на основу структуре софтверског система.



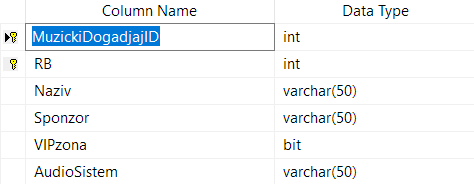
Табела Организатор



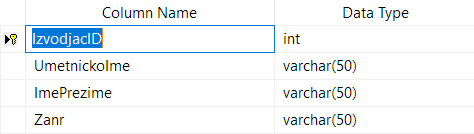
Табела Локација



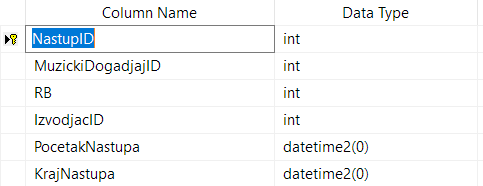
Табела Музички догађај



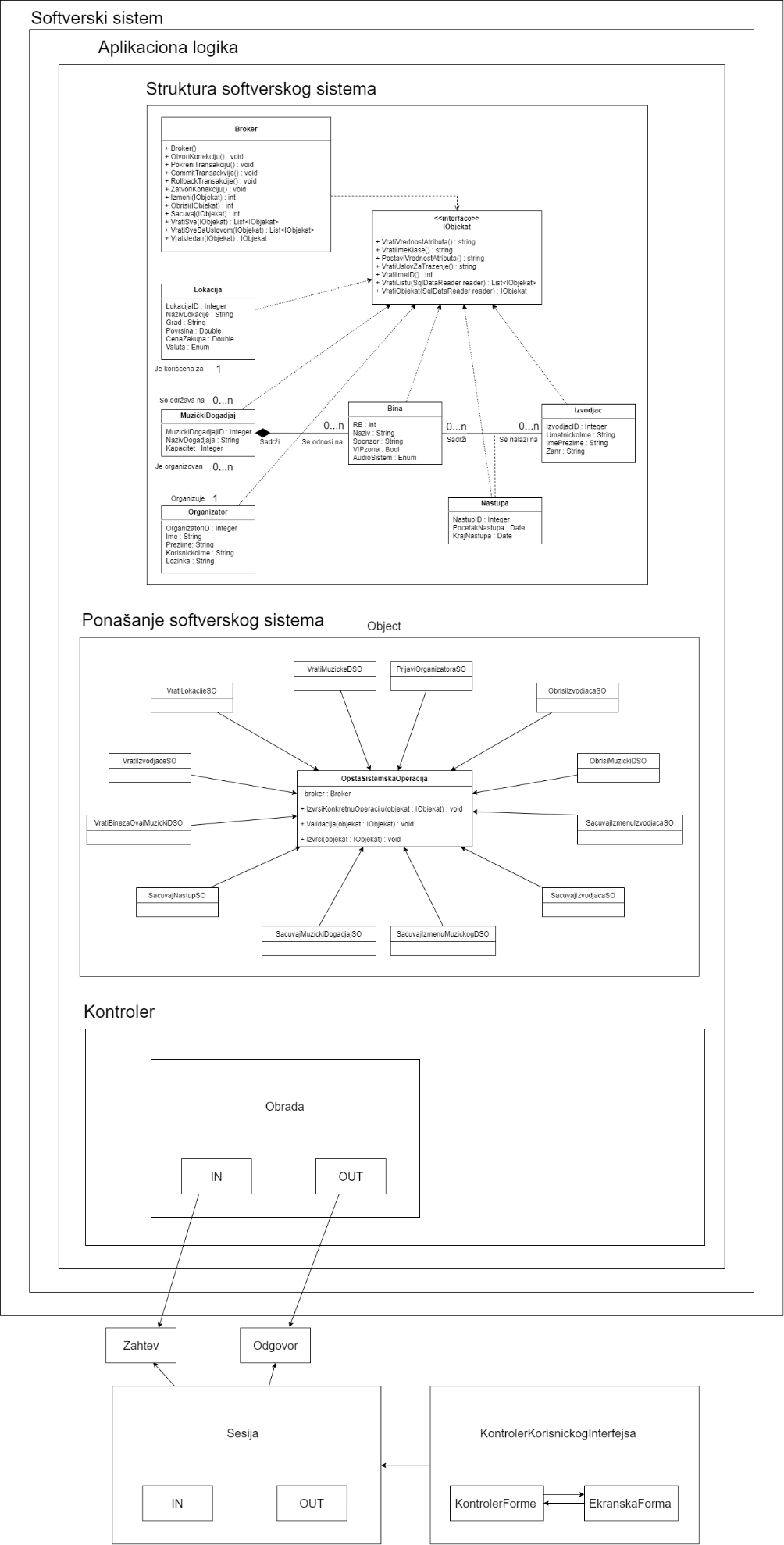
Табела Бина



Табела Извођач



Табела Наступ



# Имплементација

Софтверски систем је реализован у програмском језику C#. Пројектован је као клијент-сервер апликација. Као развојно окружење коришћен је Visual Studio 2019. База података имплементирана је преко MySQL-a док је као развојно окружење за базу података кориштен Microsoft SQL Server Management studio 18.

Систем се сатоји од шест пројеката:

* Брокер
* Домен
* Контролер
* Сервер
* СистемскеОперације
* Клијент – Mузички догађај

# Тестирање

Апликација је тестирана мануелно. Покретањем апликације и уношењем неисправних података, како би се утврдило да ли постоји грешка код валидације. Такође, уношене су и исправне вредности како би се тестирали сви случајеви коришћења и правилан упис у базу података.

Током тестирања нису кориштени алати за тестирање. Након тестирања установљено је да апликација нема грешака и да испуњава сва очекивања и случајеве кориштења.

# Литература

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