

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
PROGRAMŲ SISTEMŲ KATEDRA

## **Muzikos kaupimo programos MusiX projektas**

Laboratorinis darbas

|                |                                       |           |
|----------------|---------------------------------------|-----------|
| Atliko:        | 2 kurso 5 grupės studentai            |           |
|                | Goda Radlinskaitė, Gabrielė Žielytė   | (parašas) |
|                | Valdas Rakutis, Nedas Valentinovičius | (parašas) |
| Darbo vadovas: | dr. Vytautas Valaitis                 | (parašas) |

Vilnius – 2018

## TURINYS

|  |    |
|--|----|
| ANOTACIJA .....  | 3  |
| ĮVADAS .....   | 4  |
| 1. UŽDUOTYS IR JŲ VYKDYMO SCENARIJAI .....                           | 5  |
| 1.1. Panaudojimo atvejų diagrama .....                               | 5  |
| 1.2. Užduoties „Log in“ atlikimo veiksmų diagrama .....              | 6  |
| 1.3. Užduoties „Register“ atlikimo veiksmų diagrama .....            | 7  |
| 1.4. Užduoties „Create New Playlist“ atlikimo veiksmų diagrama ..... | 8  |
| 1.5. Užduoties „Delete playlist“ atlikimo veiksmų diagrama .....     | 9  |
| 1.6. Užduoties „Rename playlist“ atlikimo veiksmų diagrama .....     | 10 |
| 1.7. Užduoties „Add song“ atlikimo veiksmų diagrama .....            | 11 |
| 1.8. Užduoties „Edit song’s info“ atlikimo veiksmų diagrama .....    | 12 |
| 1.9. Užduoties „Delete song“ atlikimo veiksmų diagrama .....         | 13 |
| 1.10. Užduoties „Play song“ atlikimo veiksmų diagrama .....          | 14 |
| 1.11. Užduoties „Pause song“ atlikimo veiksmų diagrama .....         | 15 |
| 1.12. Užduoties „Log out“ atlikimo veiksmų diagrama .....            | 16 |
| 2. STRUKTŪRINIS PROGRAMŲ SISTEMOS MODELIS .....                      | 17 |
| 2.1. Klasių diagrama .....   | 17 |
| 2.2. Objektų diagrama .....  | 18 |
| 3. DINAMINIS PROGRAMŲ SISTEMOS MODELIS .....                         | 19 |
| 3.1. Prisijungimo ir registracijos prie programos procesas .....     | 19 |
| 3.2. Programos veikimo procesas .....                                | 20 |
| 3.3. Naujos dainos pridėjimo procesas .....                          | 21 |
| 3.4. Dainos tvarkymo procesas .....                                  | 22 |
| 4. PROGRAMŲ SISTEMOS KOMPONENTAI .....                               | 23 |
| 4.1. Nulinis lygmuo .....  | 23 |
| 4.2. Pirmas lygmuo .....   | 23 |
| 4.3. Antras lygmuo .....   | 24 |
| 4.4. Programos komponentų diagrama .....                             | 25 |
| 5. KOMPONENTŲ IŠSISKIRSTYMAS TINKLE .....                            | 26 |
| 5.1. Mazgai ir artefaktai .....                                      | 26 |
| REZULTATAI IR IŠVADOS .....  | 27 |
| PRIEDAI .....  | 27 |

## **Anotacija**

Šiame dokumente apibrėžta muzikos kaupimo platformos MusiX architektūra, pasitelkus 4+1 pjūvių modelį ir susidaro iš 5 dalių – užduočių ir jų vykdymo scenarijų, struktūrinio programos sistemos modelio, dinaminio programos sistemos modelio, programų sistemos komponentų bei komponentų išsiskirstymo tinkle. Šio darbo tikslas yra įvertinti programų sistemų inžinerijos teikiamą naudą projektų kūrimui ir klaidų bei problemų, kylančių programos kūrimo metu, kiekio mažinimui.

## **Įvadas**

Šio darbo tikslas yra suprojektuoti intuityvią, naudotojui aiškią ir nesunkiai naudojamą, greitai veikiančią muzikos kaupimo programą. Ši programa naudotojo pateiktą nuorodą į muzikinį kūrinį prideda į jo sudarytus grojaraščius ir leidžia jų klausytis ne tik platformoje, kurioje buvo įvykdytas kūrinio pasirinkimas, bet ir kitose platformose ar įrenginiuose. MusiX veiks Windows bei Android operacinėse sistemose, vartotojui prisijungus prie savo paskyros jo dainos bei grojaraščiai bus randami visuose programą palaikainčiuose įrenginiuose.

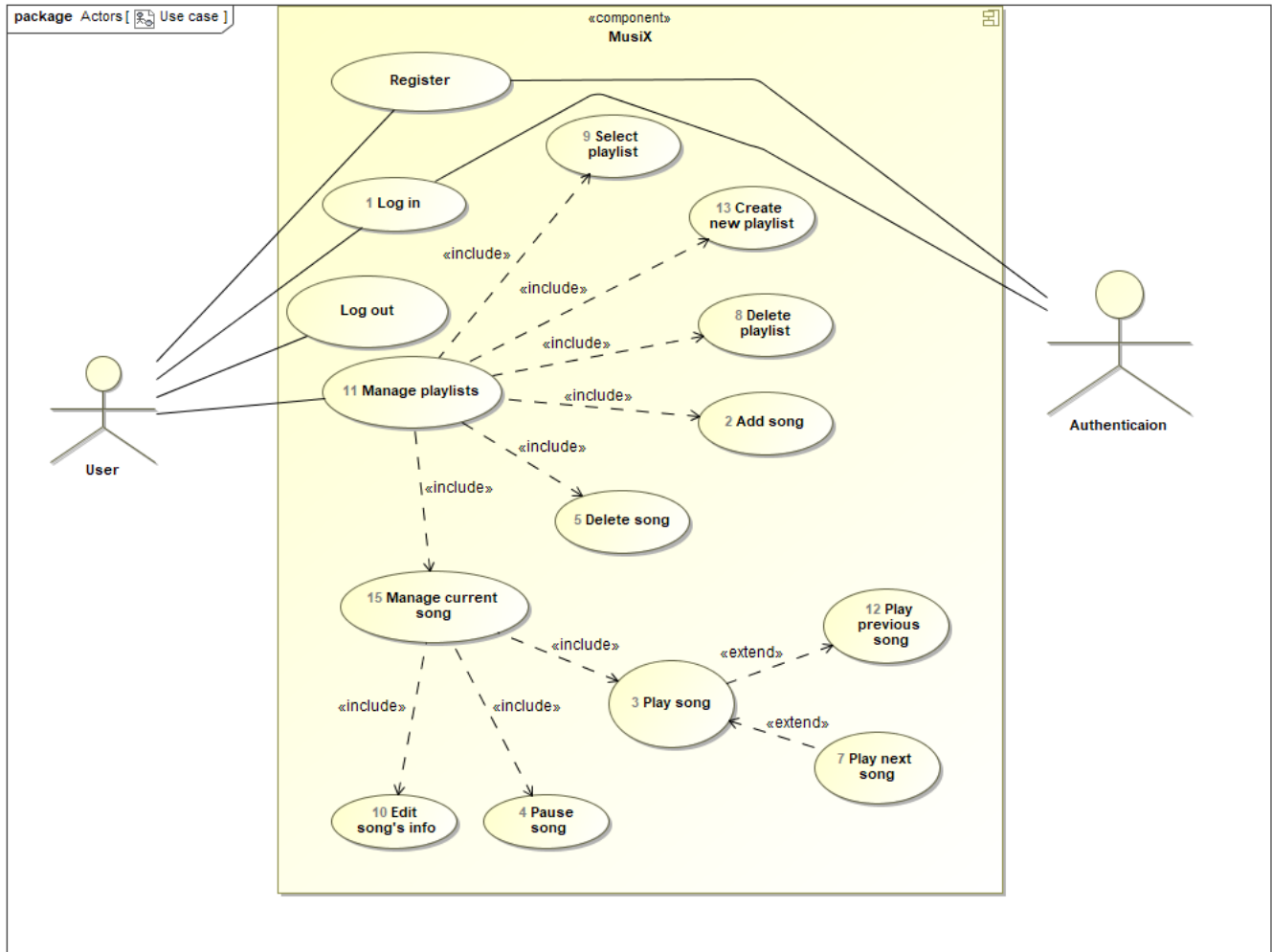
MusiX programa yra išskirtinė tuo, kad padaro tai, ko kitos šiuo metu egzistuojančios muzikos klausymo platformos nesugeba įgyvendinti ar įgyvendina limituotai – kaupti savąją dainų kolekciją iš įvairių šaltinių. Kai kurios platformos už jų siūlomos muzikinės bibliotekos papildymą pačių programos naudotojų turimais kūriniais prašo mėnesinių mokesčių ar yra išvis draudžiamos Lietuvoje bei daugelyje kitų šalių. Šis projektas bando pašalinti šiuos limitus, leisdamas nesunkiai kurti savą muzikos kolekciją be papildomų mokesčių ar draudimų.

Šiuo projektu siekiama palengvinti muzikos mėgėjų, saugančių savo muzikinę kolekciją daugybėje skirtingų vietų, gyvenimus leidžiant jiems apjungti įvairius muzikos ar kitų garso šaltinius. Tikimasi, kad galų gale MusiX bus pagrindinė bei mėgiamiausia ne tik Lietuvių, bet ir kitų šalių gyventojų muzikinė platforma.

# 1. Užduotys ir jų vykdymo scenarijai

Šiame skyriuje aprašomi programos panaudojimo atvejai bei jų įvykdymo sekų diagramos.

## 1.1. Panaudojimo atvejų diagrama

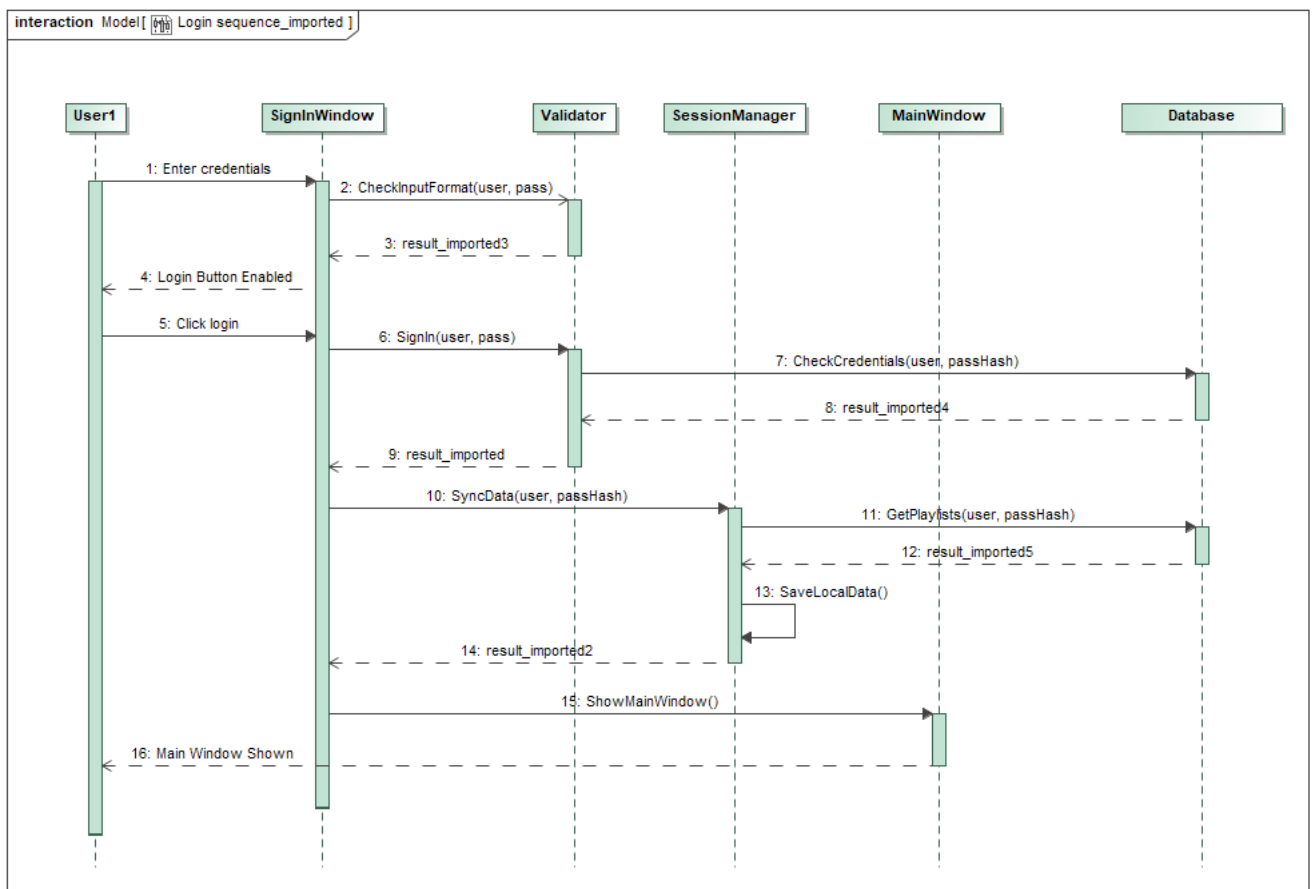


1 pav. Panaudojimo atvejai

1 - oje diagramoje matome, kokias užduotis programoje gali įvykdyti naudotojas bei naudotojo patikrinimo sistema. Naudotojas gali atlikti tokius veiksmus kaip:

- Prisijungti
- Užsiregistruoti
- Atlikti įvairius grojaraščių tvarkymo veiksmus
- Groti, sustabdyti, pridėti, trinti ir kitaip redaguoti turimas ar naujas dainas

## 1.2. Užduoties „Log in“ atlikimo veiksmų diagrama



2 pav. Prisijungimo diagrama

Diagramoje Nr. 2 matome, kas vyksta įvykdant prisijungimo uždavinį.

**Tikslas:** Prisijungti prie savo paskyros

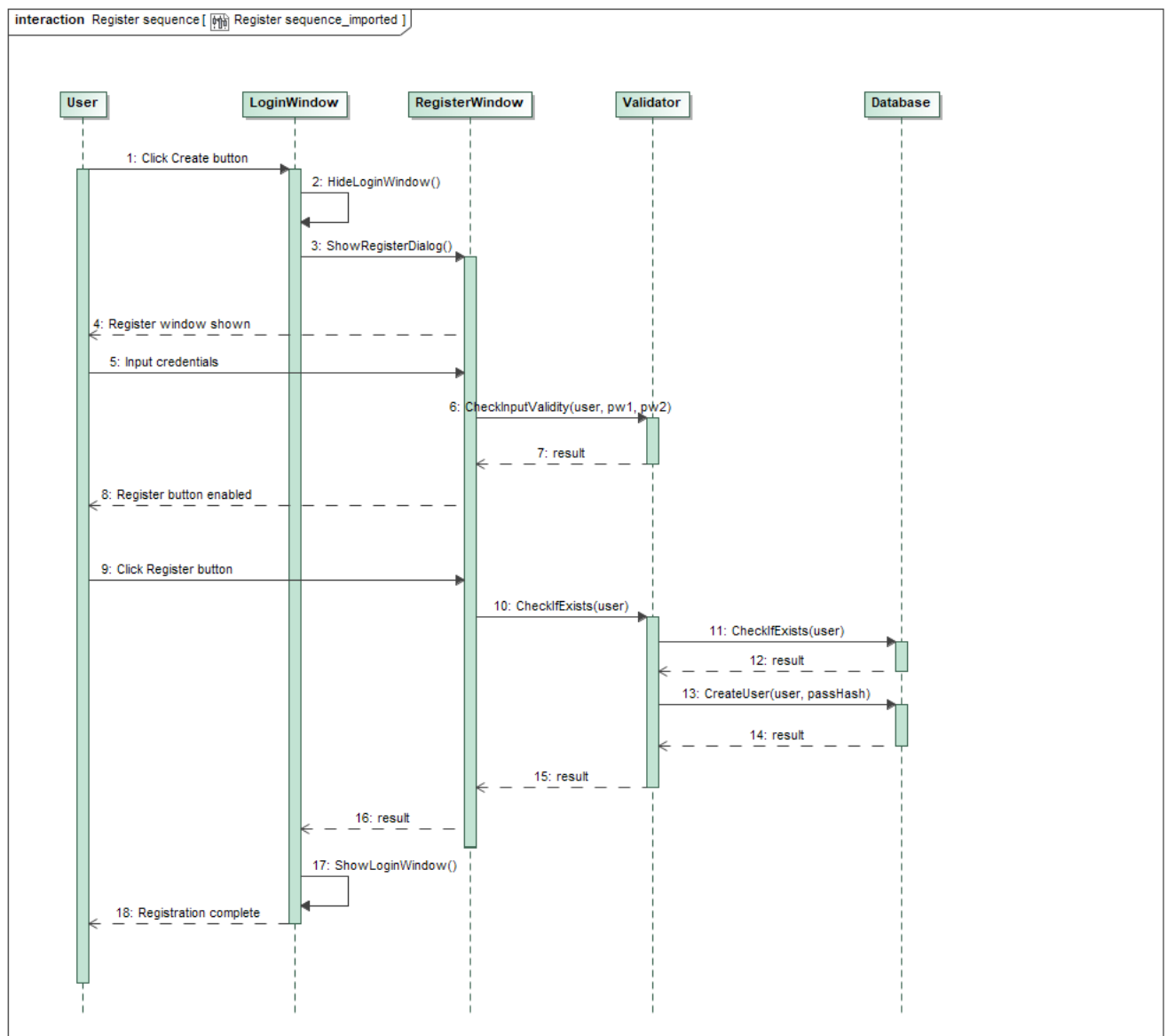
**Pirminis agentas:** Naudotojas

**„Prieš“ sąlyga:** Naudotojas atsidaręs prisijungimo langą

**„Po“ sąlyga:** Naudotojas prisijungia prie sistemos ir patenka į pagrindinį langą.

**Scenarijus:** Naudotojas įveda prisijungimo duomenis. Šie duomenys yra sulyginami su duombazėje laikomais duomenimis. Jei nerandamas atitikmuo, grįžtama į prisijungimo langą. Jei atitinkami duomenys aptinkami sistemoje, naudotojas yra prijungiamas prie sistemos ir jam yra parodomas pagrindinis langas.

### 1.3. Užduoties „Register“ atlikimo veiksmų diagrama



3 pav. Registracijos diagrama

Diagramoje Nr. 3 matome, kas vyksta įvykdam registracijos uždavinį.

**Tikslas:** Priregistruoti naują varototoją sistemoje.

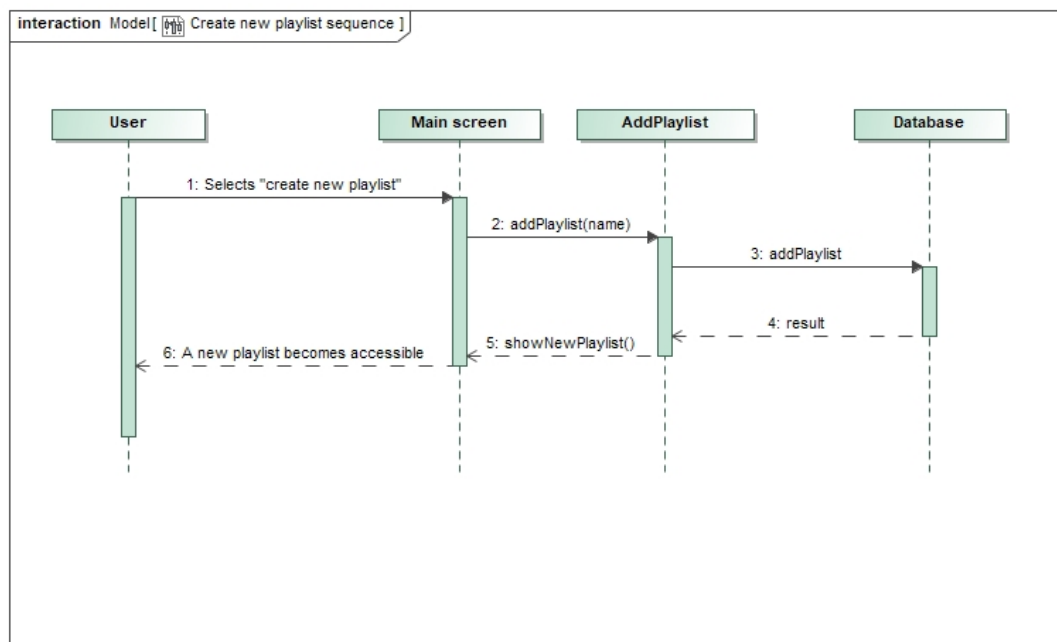
**Pirminis agentas:** Naudotojas

**„Prieš“ sąlyga:** Naudotojas atsidaręs registracijos langą

**„Po“ sąlyga:** Naudotojas užsiregistruoja ir gali sėkmnigai prisijungti prie sistemos.

**Scenarijus:** Naudotojas įveda registracijos lange prašomus duomenis. Šių formatas yra patikrinamas. Tuomet yra tikrinama, ar duomenų bazėje tokie duomenys dar neegzistuoja. Jei neegzistuoja, naudotojo duomenys išsaugomi duomenų bazėje ir parodomas prisijungimo langas.

## 1.4. Užduoties „Create New Playlist“ atlikimo veiksmų diagrama



4 pav. Naujo grojaraščio sukūrimas

Diagramoje Nr. 4 matome, kas vyksta įvykdant naujo grojaraščio sukūrimo uždavinį.

**Tikslas:** Sukurti naują grojaraštį

**Pirminis agentas:** Naudotojas

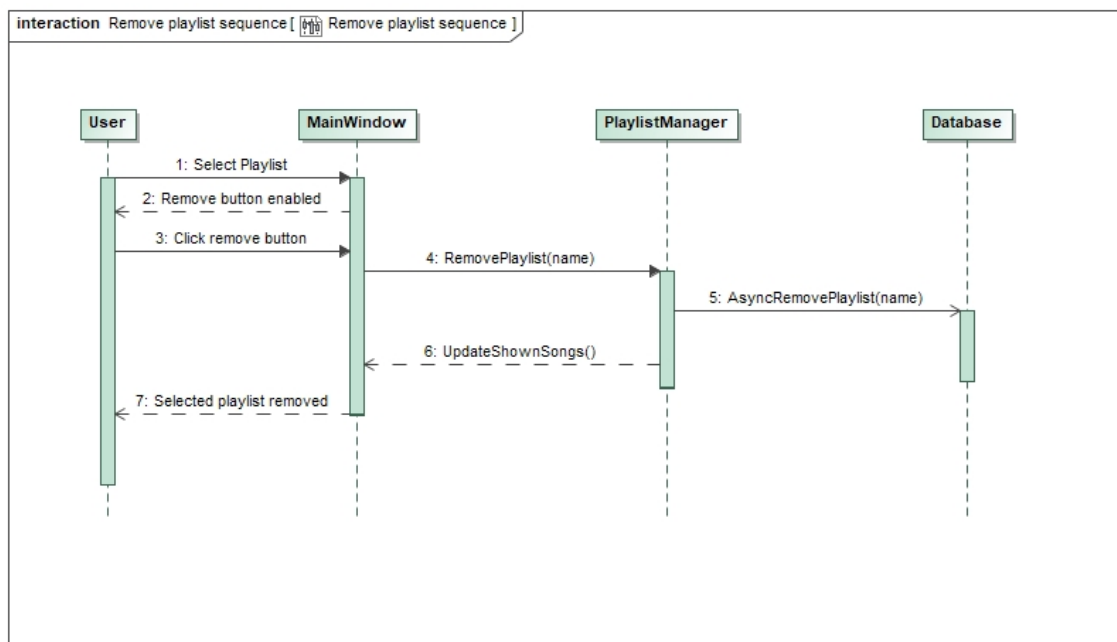
**„Prieš“ sąlyga:** Naudotojas nori susikurti naują grojaraštį

**„Po“ sąlyga:** Sukuriamas ir duomenų bazėje išsaugomas naujas grojaraštis.

**Scenarijus:** Naudotojas paspaudžia naujo grojaraščio pridėjimo mygtuką. Atsiradusiame teksto langelyje įrašo grojaraščio pavadinimą. Sistema patikrina, ar toks grojaraštis jau neegzistuoja, ir jei neegzistuoja tai prideda naują grojaraštį naudotojo duombazėje ir parodo jį naudojimosi aplinkoje.



## 1.5. Užduoties „Delete playlist“ atlikimo veiksmų diagrama



5 pav. Grojaraščio ištrynimasis

Diagramoje Nr. 5 matome, kas vyksta įvykdamas grojaraščio ištrynimo uždavinį.

**Tikslas:** Ištrinti grojaraštį

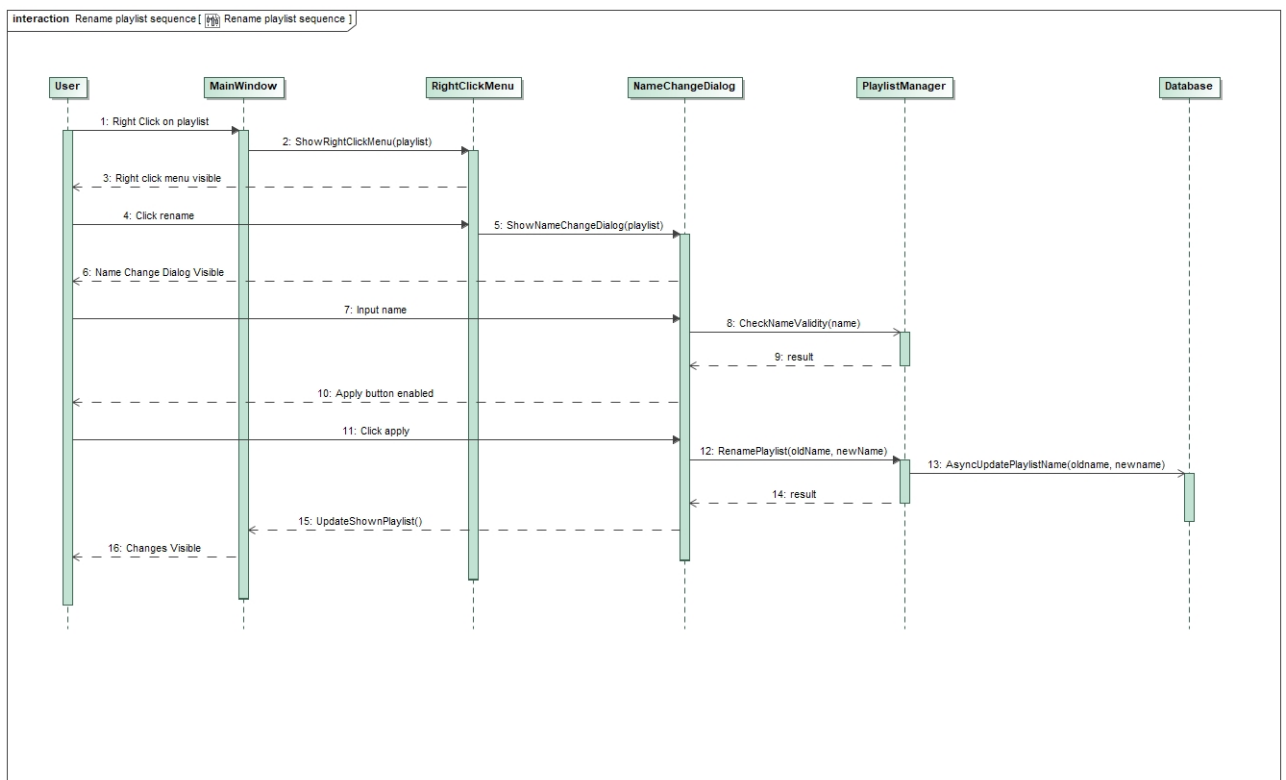
**Pirminis agentas:** Naudotojas

**„Prieš“ sąlyga:** Naudotojas atsidaręs pagrindinį langą.

**„Po“ sąlyga:** Naudotojas pagrindiniame lange nebemato ištrinto grojaraščio.

**Scenarijus:** Naudotojas pasirenka norimą ištrinti grojaraštį ir paspaudžia „Remove“ mygtuką. Lokaliai išrinamas grojaraštis ir į duomenų bazę nusiunčiama užklausa, kad ištrintų grojaraštį.

## 1.6. Užduoties „Rename playlist“ atlikimo veiksmų diagrama



6 pav. Grojaraščio pavadinimo keitimas

Diagramoje Nr. 6 matome, kas vyksta norint įvykdyti grojaraščio pervardijimo uždavinį.

**Tikslas:** Pakeisti grojaraščio pavadinimą.

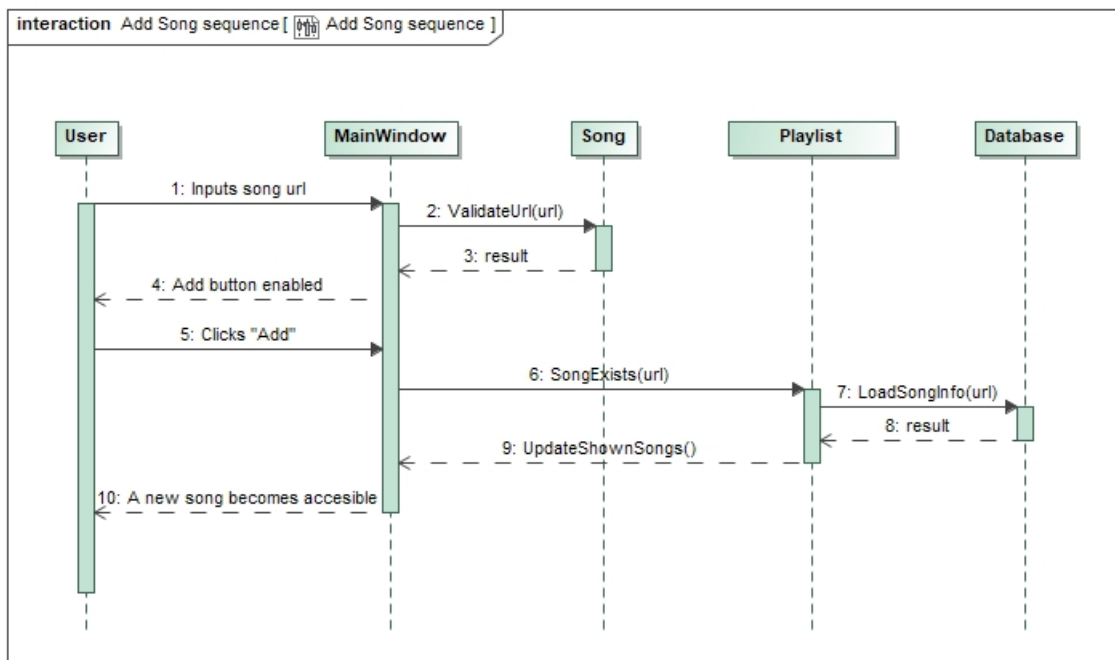
**Pirminis agentas:** Naudotojas

**„Prieš“ sąlyga:** Naudotojas atsidaręs pagrindinį langą.

**„Po“ sąlyga:** Naudotojas pagrindiniame lange mato pervadintą grojaraštį.

**Scenarijus:** Naudotojas dešiniu pelės klavišu spusteli ant grojaraščio, kurį nori pervadinti. Atsidariusiame meniu pasirenka opciją „Rename“. Atsidaro langas kuriame vartotojas gali įvesti naują pavadinimą. Su kiekvienu klavišo paspaudimu yra tikrinama ar pavadinimas yra teisingo formato. Įvedus teisingą pavadinimą galima paspausti mygtuką „Apply“, kuris tiek lokaliai, tiek duomenų bazėje atnaujina grojaraščio pavadinimą. Naudotojas pagrindiniame lange mato pasikeitimus.

## 1.7. Užduoties „Add song“ atlikimo veiksmų diagrama



7 pav. Naujos dainos pridėjimas

Diagramoje Nr. 7 matome, kas vyksta įvykdant naujos dainos pridėjimo uždavinį.

**Tikslas:** Pridėti naują dainą prie grojaraščio

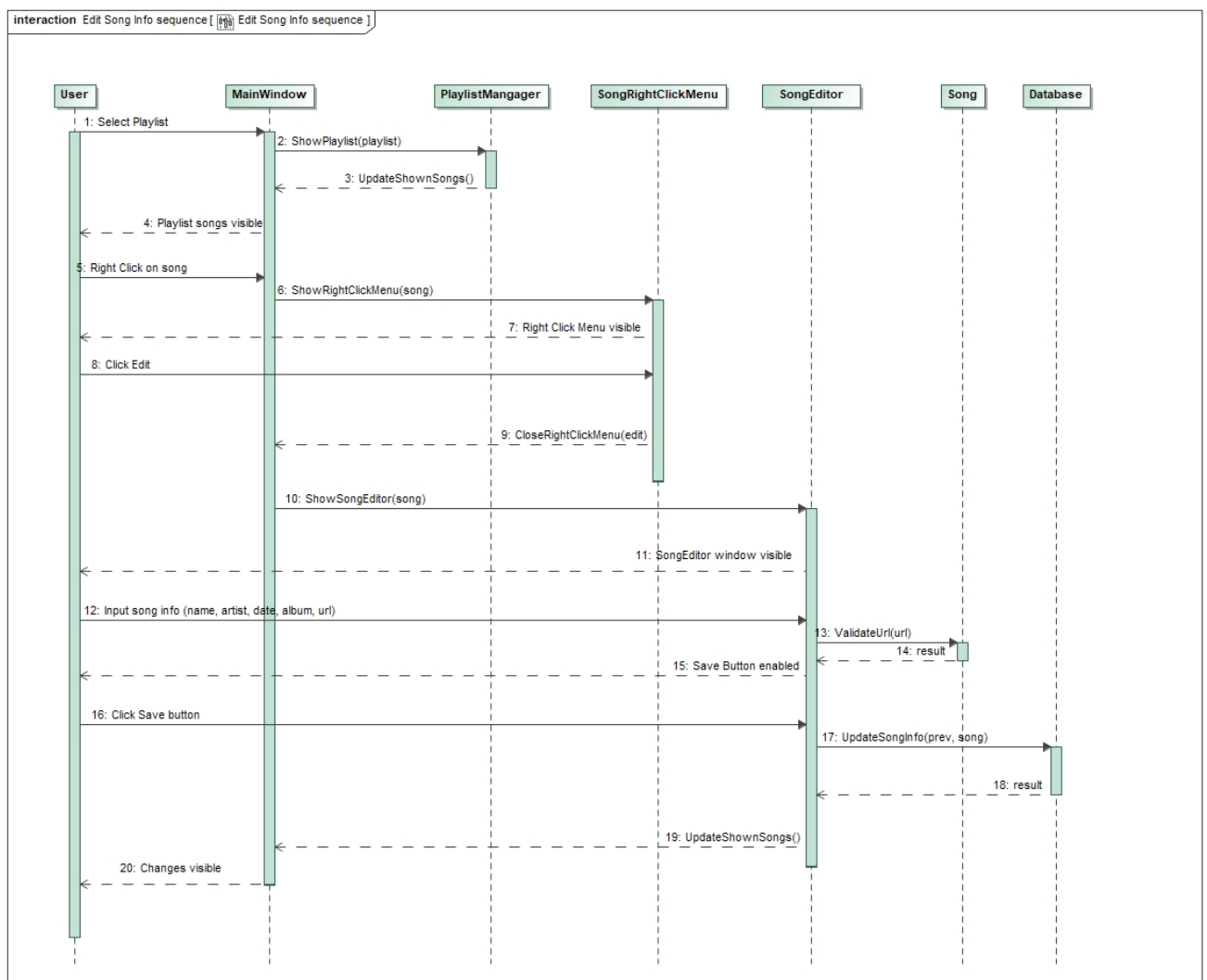
**Pirminis agentas:** Naudotojas

**„Prieš“ sąlyga:** Naudotojas atsidaręs pagrindinį langą.

**„Po“ sąlyga:** Naudotojas pagrindiniame lange mato naują dainą.

**Scenarijus:** Naudotojas pagrindiniame lange įveda dainos URL. Tuo metu tikrinama, ar iš šio URL galima atsisiųsti medijos failą. Jeigu galima, tada spustelėjus „Add“ mygtuką tikrinama, ar dabartiniame grojaraštyje tokia daina jau egzistuoja. Jeigu neegzistuoja, į duomenų bazę siunčiama užklausa su dainos URL. Jeigu duomenų bazė ieško tokios dainos ir grąžina jos duomenis (atlikėją, datą, albumą, pavadinimą). Galiausiai atnaujinamos ekrane rodomų dainų sąrašas.

## 1.8. Užduoties „Edit song's info“ atlikimo veiksmų diagrama



8 pav. Dainos duomenų pakeitimo diagrama

Diagramoje Nr. 8 matome, kas vyksta įvykdant dainos duomenų pakeitimo uždavinį.

**Tikslas:** Pakeisti dainos duomenis

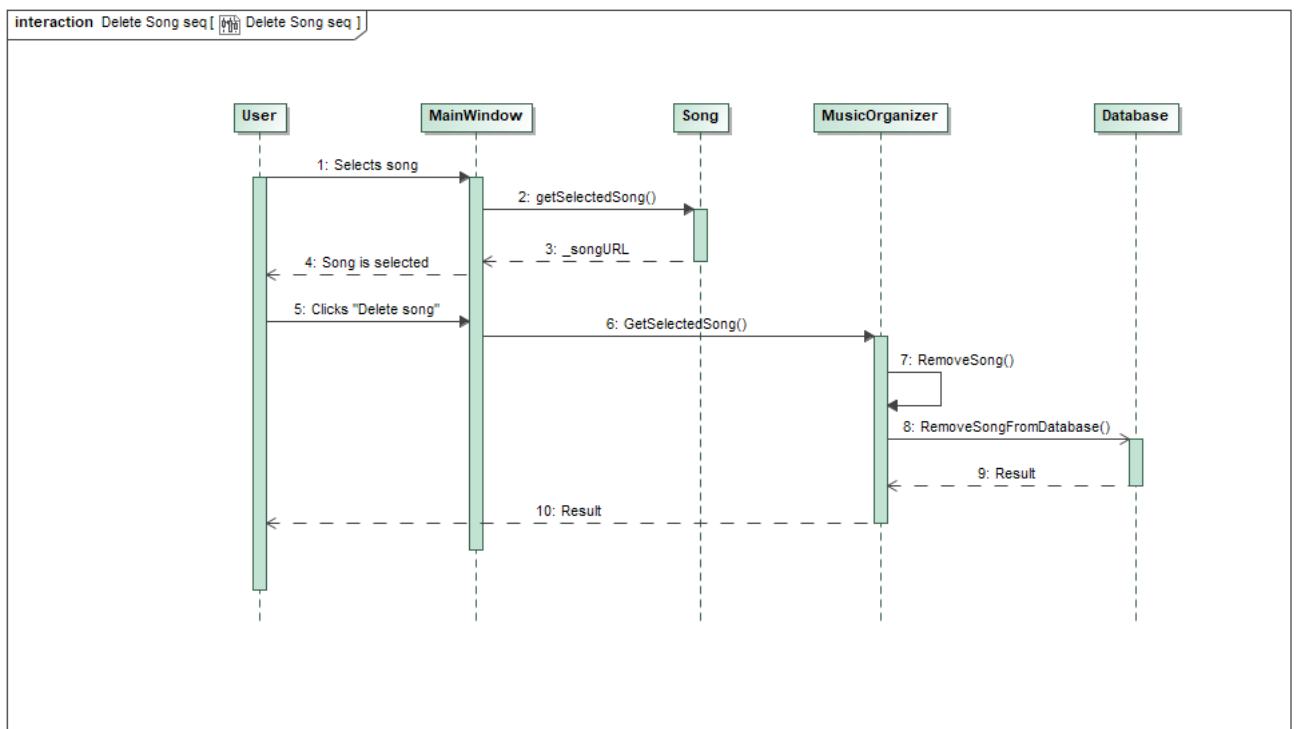
**Pirminis agentas:** Naudotojas

**„Prieš“ sąlyga:** Naudotojas atsidaręs pagrindinį langą.

**„Po“ sąlyga:** Naudotojas pagrindiniame lange mato atnaujintus duomenis.

**Scenarijus:** Naudotojas pasirenka grojaraštį kuriame nori pakeisti dainą. Dešiniu pelės klavišu paspaudus and norimos pakeisti dainos, atsidaro meniu, kuriame vartotojas spaudžia ant „Edit“ opcijos. Atsidaro langas, kuriame vartotojas gali pakeisti dainos duomenis. Pakeitus duomenis jie validuojami. Jeigu duomenys teisingi, duomenys išsaugojami duomenų bazėje ir užsidaro dainos informacijos modifikavimo langas.

## 1.9. Užduoties „Delete song“ atlikimo veiksmų diagrama



9 pav. Dainos pašalinimo diagrama

Diagramoje Nr. 9 matome, kas vyksta įvykdant dainos pašalinimo uždavinį.

**Tikslas:** Ištrinti dainą

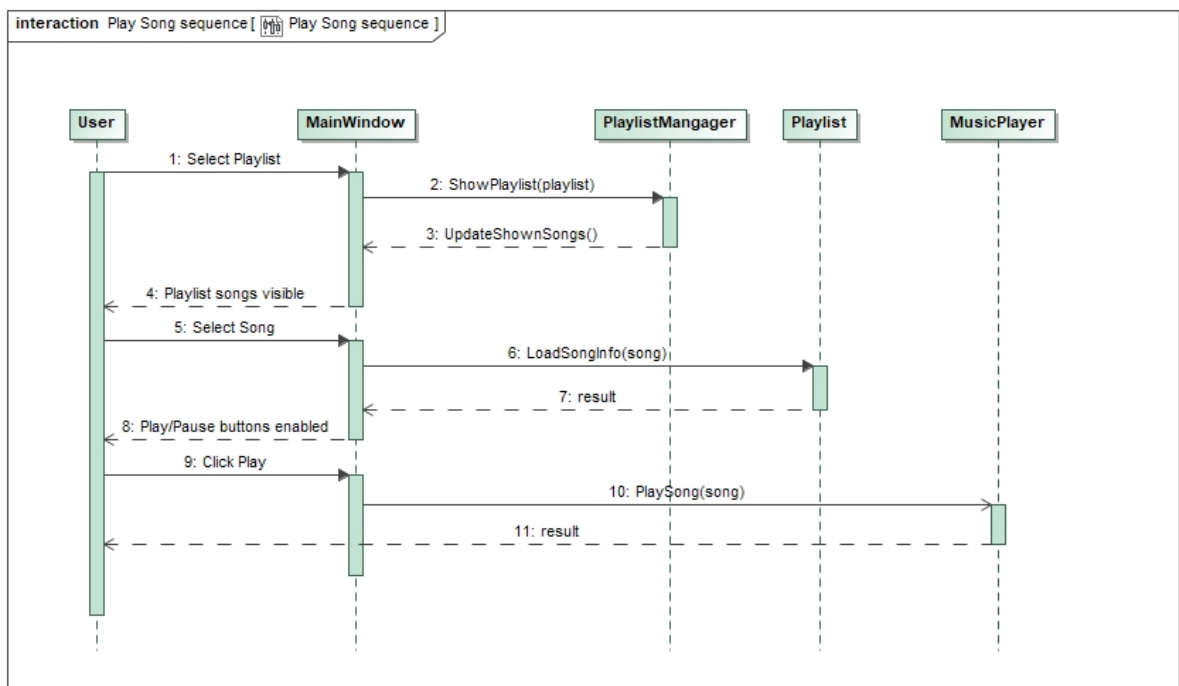
**Pirminis agentas:** Naudotojas

**„Prieš“ sąlyga:** Naudotojas atsidaręs pagrindinį langą.

**„Po“ sąlyga:** Naudotojas pasirinktam grojaraštį nebemato ištrintos dainos.

**Scenarijus:** Naudotojas pasirenka grojaraštį kuriame nori pakeisti dainą. Dešiniu pelės klavišu paspaudus and norimos pakeisti dainos, atsidaro meniu, kuriame vartotojas spaudžia ant „Remove“ opcijos. Pasirinktame grojaraštyje ištrinama daina, bei nusiunčiama užklausa į duomenų bazę, kad daina būtų ištrinta. Lange nebelieka ištrintos dainos.

## 1.10. Užduoties „Play song“ atlikimo veiksmų diagrama



10 pav. Dainos pradėjimo groti diagrama

Diagramoje Nr. 10 matome, kas vyksta įvykdant dainos pradėjimo groti uždavinį.

**Tikslas:** Paleisti dainą

**Pirminis agentas:** Naudotojas

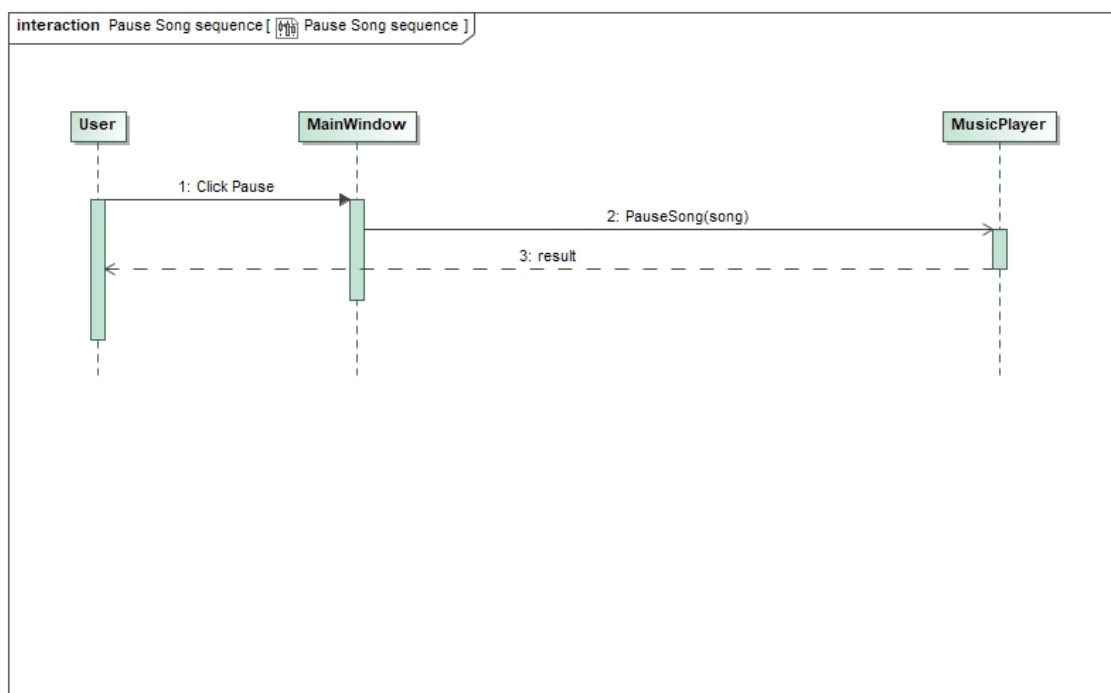
**„Prieš“ sąlyga:** Naudotojas atsidaręs pagrindinį langą.

**„Po“ sąlyga:** Naudotojas girdi norimą dainą.

**Scenarijus:** Naudotojas pasirenka grojaraštį kuriame yra norima daina, tada pasirenka dainą.

Leidžiami spausti „Play/Pause“ mygtukai, paspaudus „Play“ – daina pradeda groti.

### 1.11. Užduoties „Pause song“ atlikimo veiksmų diagrama



11 pav. Dainos pristabdymo diagrama

Diagramoje Nr. 11 matome, kas vyksta įvykdant dainos pristabdymo uždavinį.

**Tikslas:** Sustabdyti grojančią dainą dainą

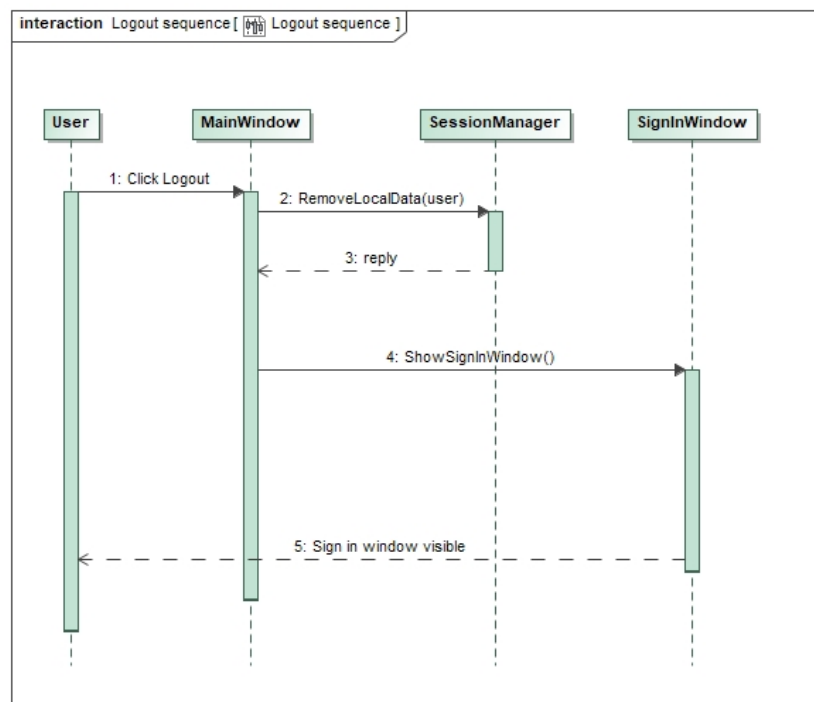
**Pirminis agentas:** Naudotojas

**„Prieš“ sąlyga:** Naudotojas atsidaręs pagrindinį langą ir girdi grojančią dainą.

**„Po“ sąlyga:** Naudotojas nebegirdi grojančios dainos.

**Scenarijus:** Naudotojas spusteli „Pause“ mygtuką. Grojanti daina sustoja.

## 1.12. Užduoties „Log out“ atlikimo veiksmų diagrama



12 pav. Atsijungimo nuo sistemos diagrama

Diagramoje Nr. 12 matome, kas vyksta įvykdant atsijungimo nuo sistemo uždavinį.

**Tikslas:** Atsijungti nuo savo paskyros

**Pirminis agentas:** Naudotojas

**„Prieš“ sąlyga:** Naudotojas atsidaręs pagrindinį langą.

**„Po“ sąlyga:** Naudotojas mato prisijungimo langą.

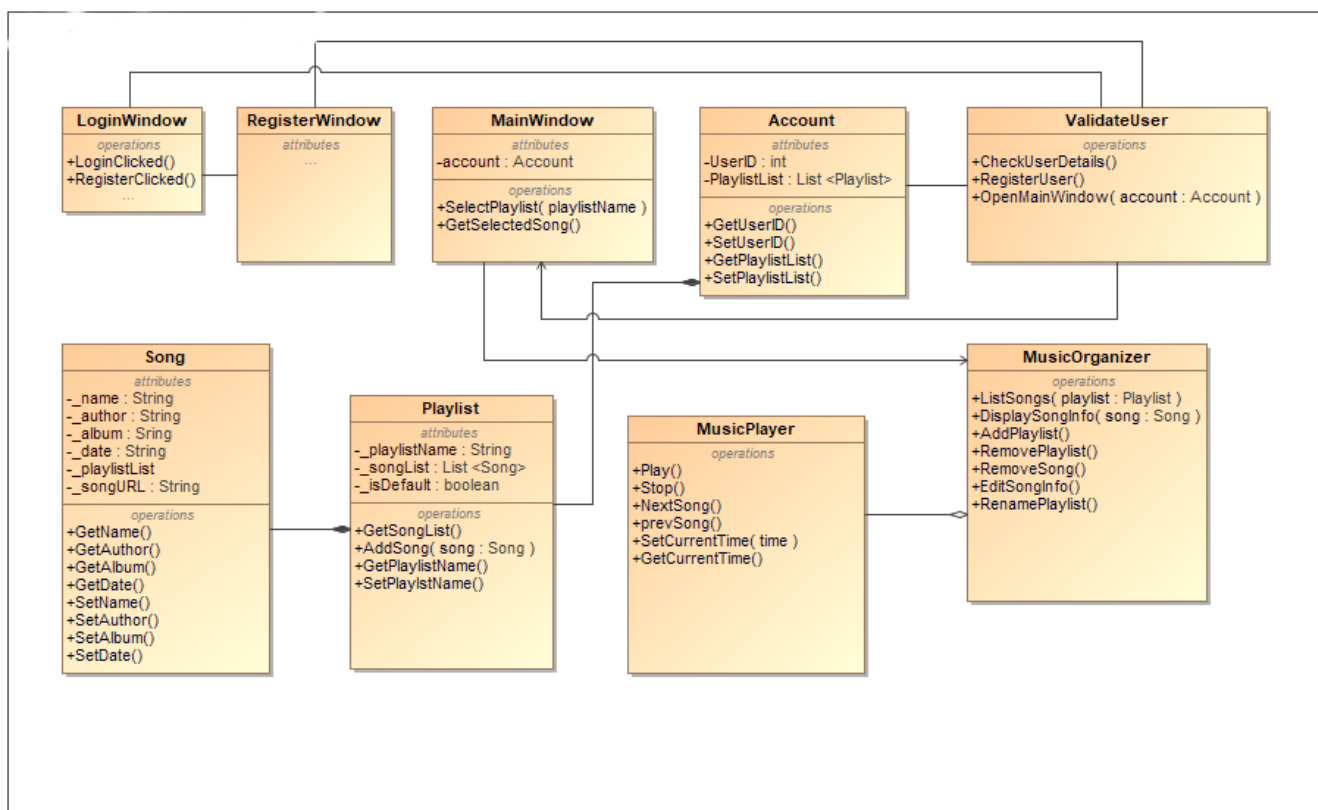
**Scenarijus:** Naudotojas pagrindiniame lange spusteli „Log out“ mygtuką. Ištrinami lokaliai išsaugoti vartotojo duomenys kaip playlistai, atsisiųstos dainos. Uždaromas pagrindinis langas ir atidaromas prisijungimo langas.



## 2. Struktūrinis programų sistemos modelis

Šiame skyriuje programinė sistema bus aprašoma iš struktūrinės pusės, daugiau dėmesio skiriant sistemą įgyvendinančioms klasėms.

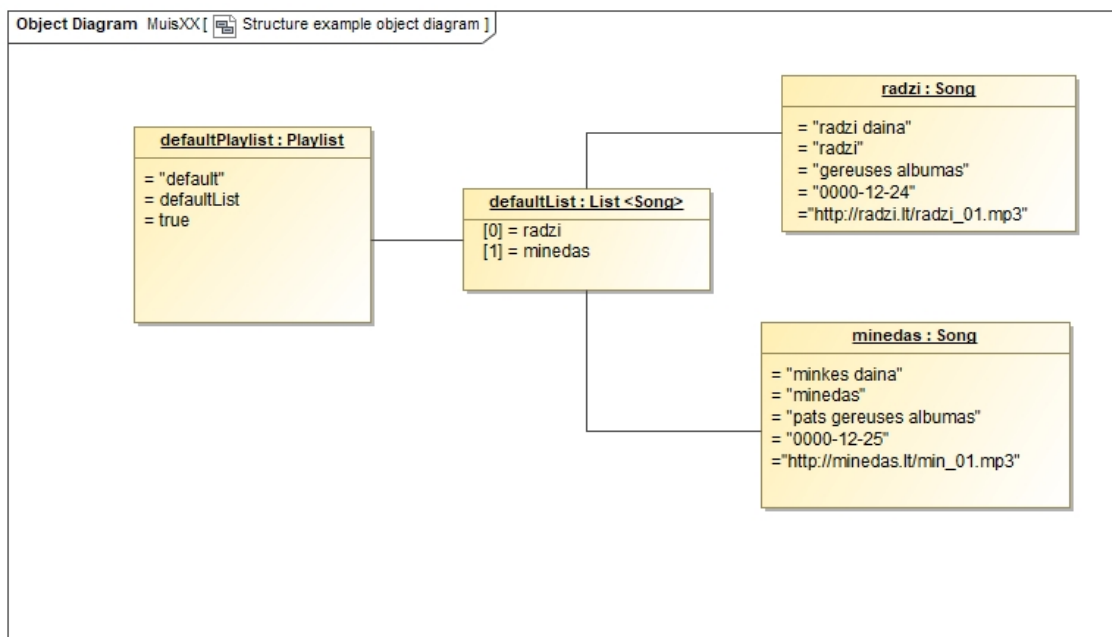
### 2.1. Klasių diagrama



13 pav. Programą sudarančios klasės ir jų ryšiai

Klasių diagramoje, pateiktoje priede Nr.13, matome programoje veikiančias klases bei jų ryšius. Itin svarbi čia yra MusicOrganizer klasė, kuri valdo didžiąją dalį programos veikimo ir yra iššaukiama iškart po sėkmingo prisijungimo. Ji šaukia bei kuria kitus objektus, tokius kaip naujai pridėtos dainos ar grojaraščiai, ir jas nuolatos sinchronizuoja su duomenų baze.

## 2.2. Objektų diagrama



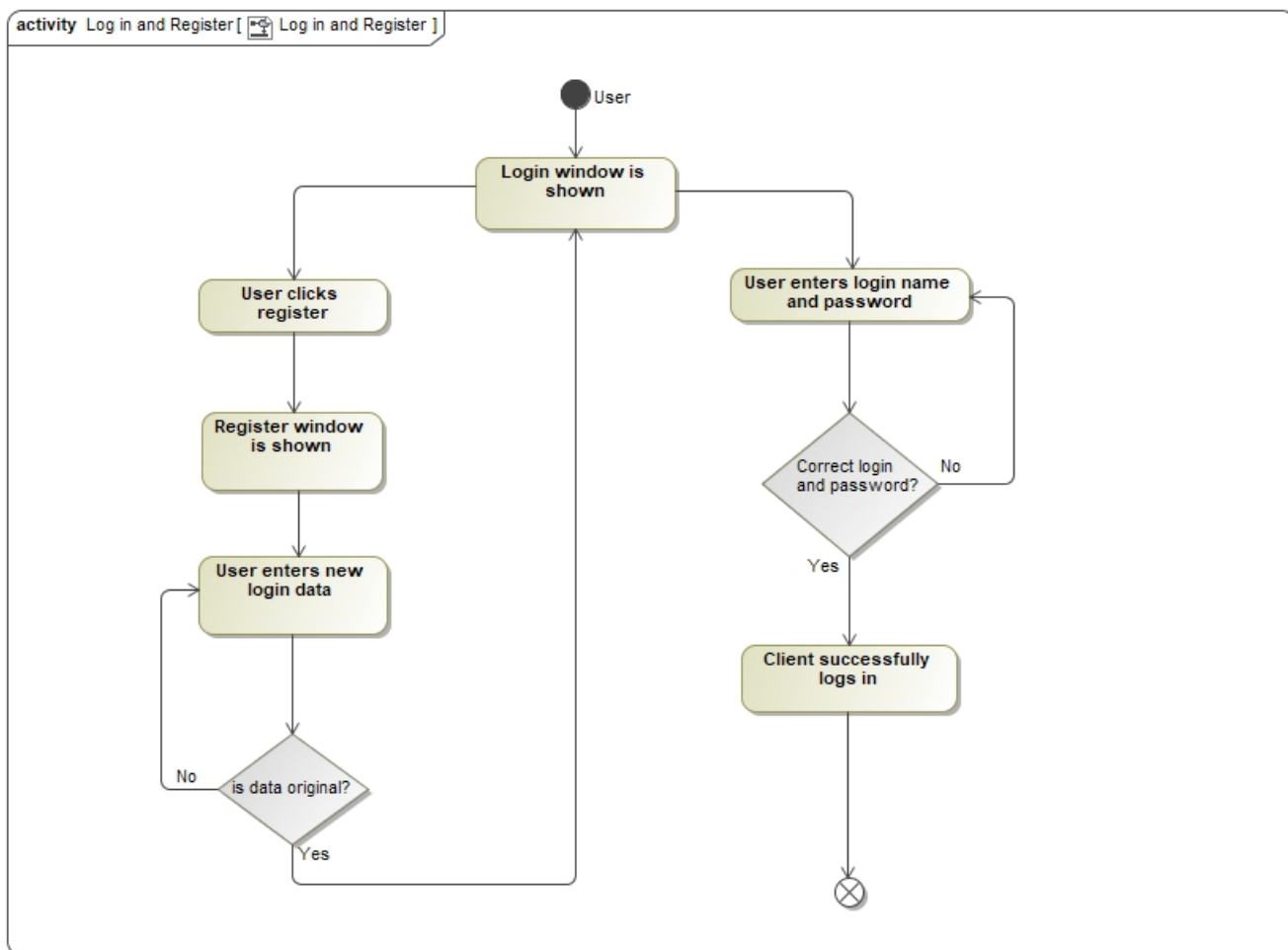
14 pav. Objektai ir jų ryšiai

Pavyzdyje, pavaizduotame diagramoje Nr.14 matome, kaip vyksta dainos saugojimas loka-lioje atmintyje dainos veikimo metu. Naudotojo turimos dainos iš duomenų bazės yra įtraukiamos į sąrašą, kuris gali būti priskirtas įvairiems grojaraščiams (šiuo atveju dainų sąrašas yra priskiria-mas Numatytajam grojaraščiui). Panašiu būdu yra saugomos ir kitos dainos, ir naudotojo turimi grojaraščiai. List'e taip pat turetų būti laikoma ir informacija apie dainas ar grojaraščius.

### 3. Dinaminis programų sistemos modelis

Šiame skyriuje programinė sistema bus aprašoma iš elgsenos jos vykdymo metu pusės išskaidant tam tikras funkcijas į smulkesnes diagramas.

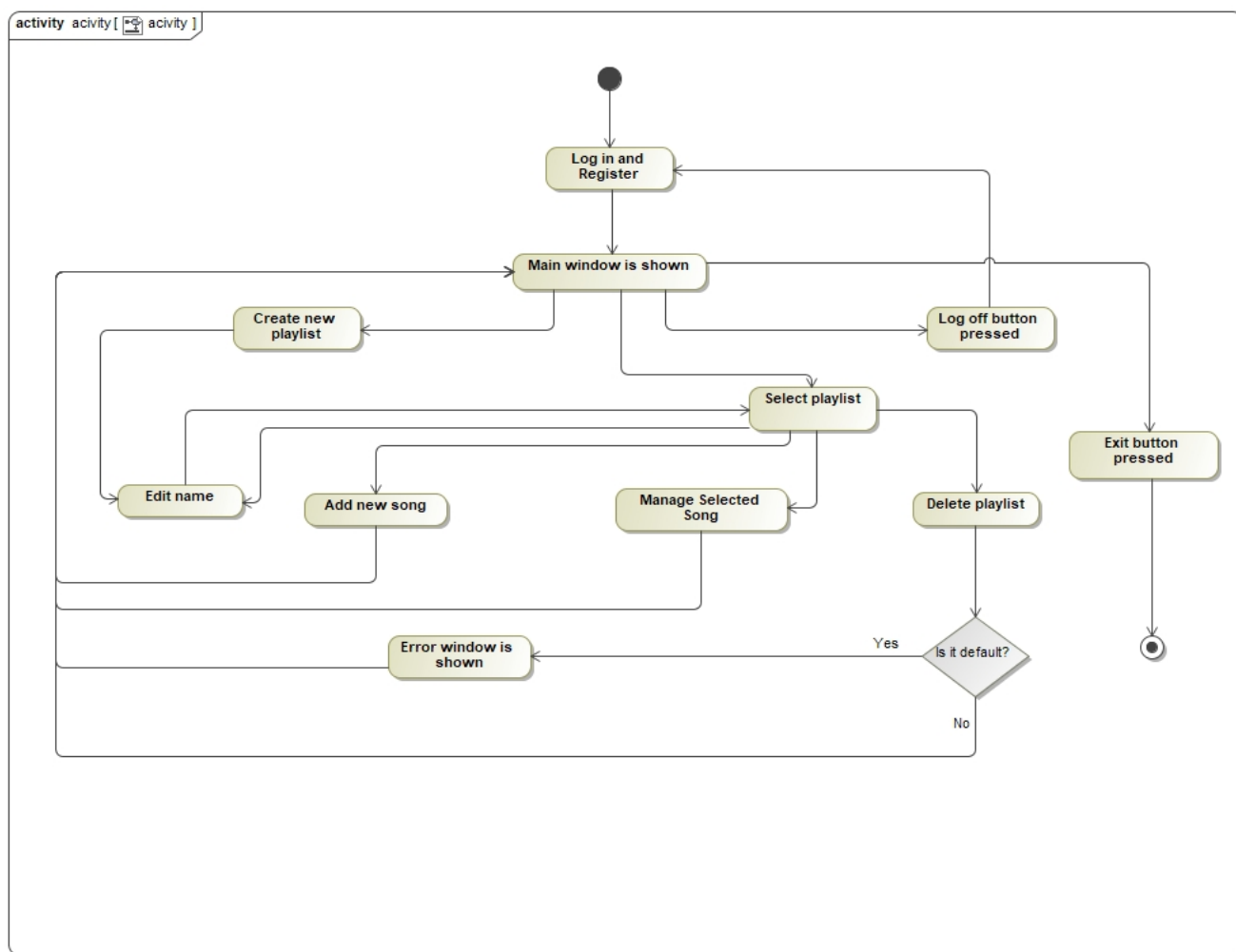
#### 3.1. Prisijungimo ir registracijos prie programos procesas



15 pav. Prisijungimo ar registracijos procesas

Diagramoje Nr.15 matome, kaip vyksta naudotojo prisijungimas ar registracija. Kadangi naudotojas visuomet yra prašomas prisijungti, šie veiksmai visuomet bus vykdomi, norint patekti į pagrindinį programos langą. Svabu pastebėti, kad po sėkmingos registracijos naudotojas yra siunčiamas į pagrindinį langą ir jam reikia praeiti visą prisijungimo procedūrą.

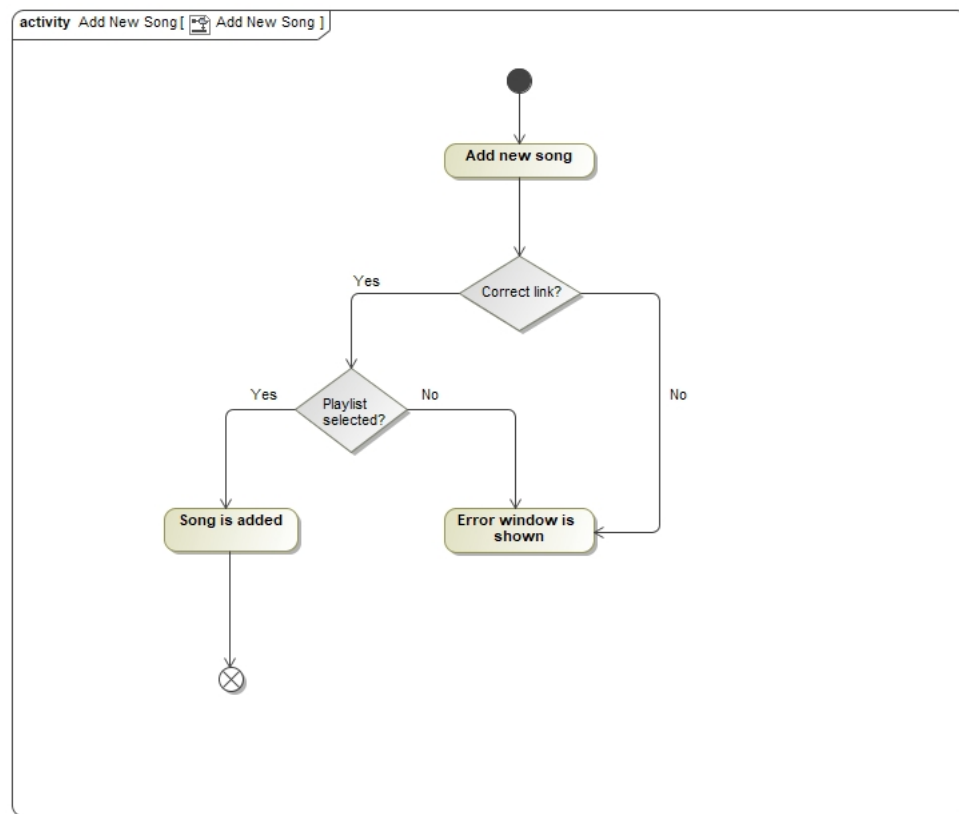
### 3.2. Programos veikimo procesas



16 pav. Programos veikimo procesas

Diagrama Nr.16 vaizduoja pagrindinio programos veikimo procesą. Kaip matome, didžioji dalis veiksmų veda atgal į pagrindinį langą, nes didžioji dalis veiksmo vyksta būtent čia. Programa baigia darbą tik paspaudus išjungimo mygtuką. Atsijungus nuo paskyros grįžtama atgal į prisijungimo langą. Detalesni naujos dainos pridėjimo bei pasirinktos dainos tvarkymo grafikai matomi 17 ir 18 diagramose atitinkamai.

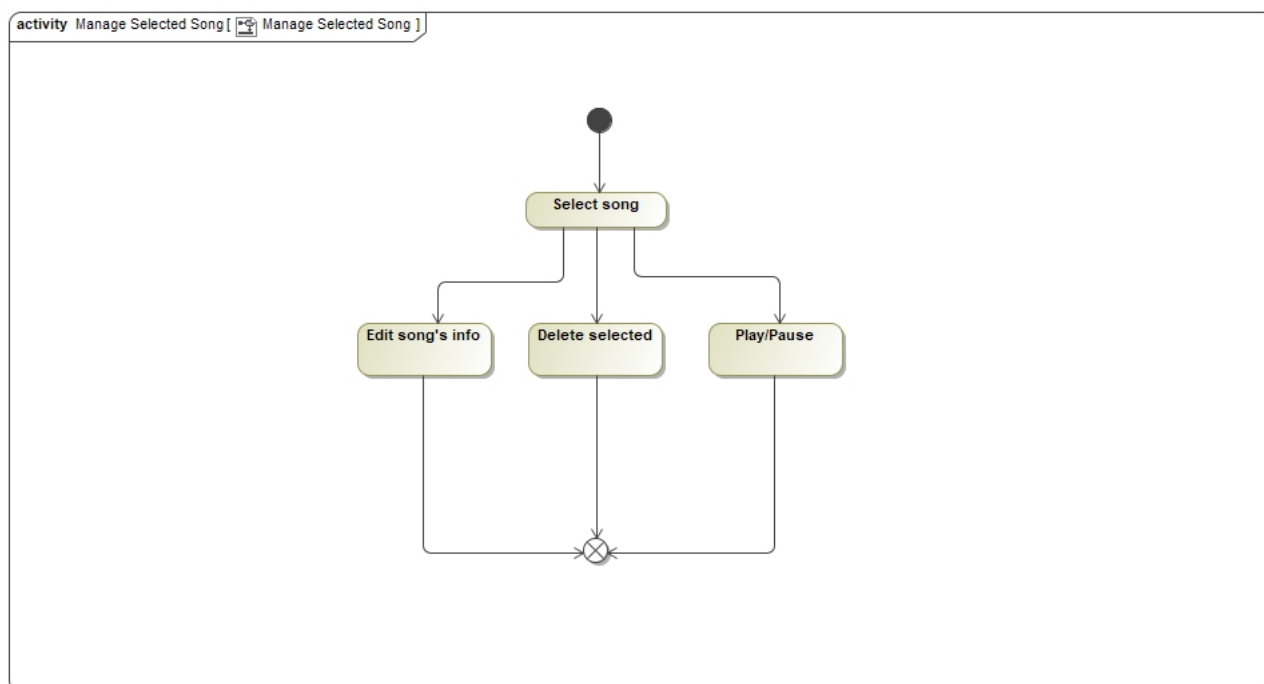
### 3.3. Naujos dainos pridėjimo procesas



17 pav. Dainos pridėjimo procesas

Diagramoje Nr.17 matome dainos pridėjimo procesą. Pridedant naują dainą tikrinami du dalykai – ar nurodyta nuroda yra teisinga bei palaikoma sistemos ir ar buvo pasirinktas grojaraštis. Jei bent vienas iš šių dalykų yra neigiamas, dainos pridėjimo procesas yra nutraukiamas.

### 3.4. Dainos tvarkymo procesas



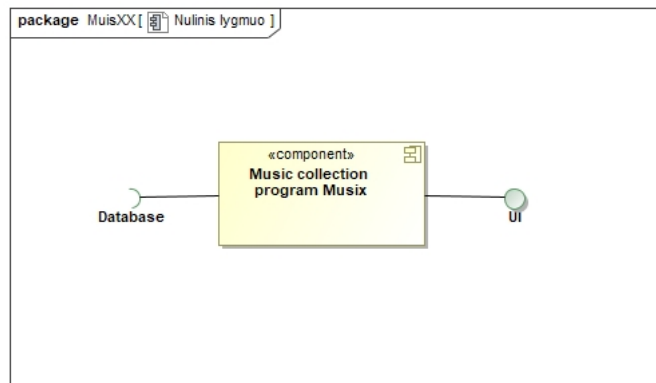
18 pav. Dainos tvarkymo procesas

Diagrama Nr.18 rodo galimybes, kurias galima daryti pasirinkus dainą. Į tai įeina ir dainos informacijos keitimas, ir dainos trynimasis. Šios dvi operacijos glaudžiai bendrauja su duomenų baze ir jų veikimas matomas diagramose Nr.8 bei 9 atitinkamai. Trečiasis pasirinkimas yra paprastas dainos pradėjimas groti ar jos sustabdymas. Šie veiksmai detalčiau yra aprašomi diagramose Nr.10 bei 11.

## 4. Programų sistemos komponentai

Šiame skyriuje programinė sistema bus aprašoma taip pat iš struktūrinės pusės, tik šį kartą daugiau dėmesio bus skiriama klasių skirstymui į komponentus, ryšio tarp jų apibrėžimui.

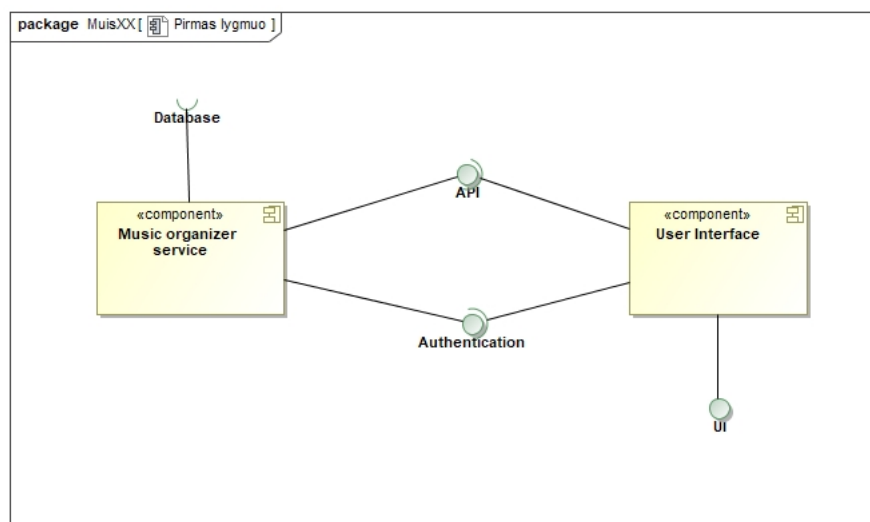
### 4.1. Nulinis lygmuo



19 pav. Nulinio lygmens komponentai

19 pav. pavaizduota bendriausia komponentų diagrama nuliniame lygyje. Ją sudaro vienas komponentas – pagrindinė MusiX sistema, kuri yra atsakinga už naudotojo matomą naudojimosi aplinką, naudotojo veiksmų apdorojimą bei bendravimą su duomenų baze.

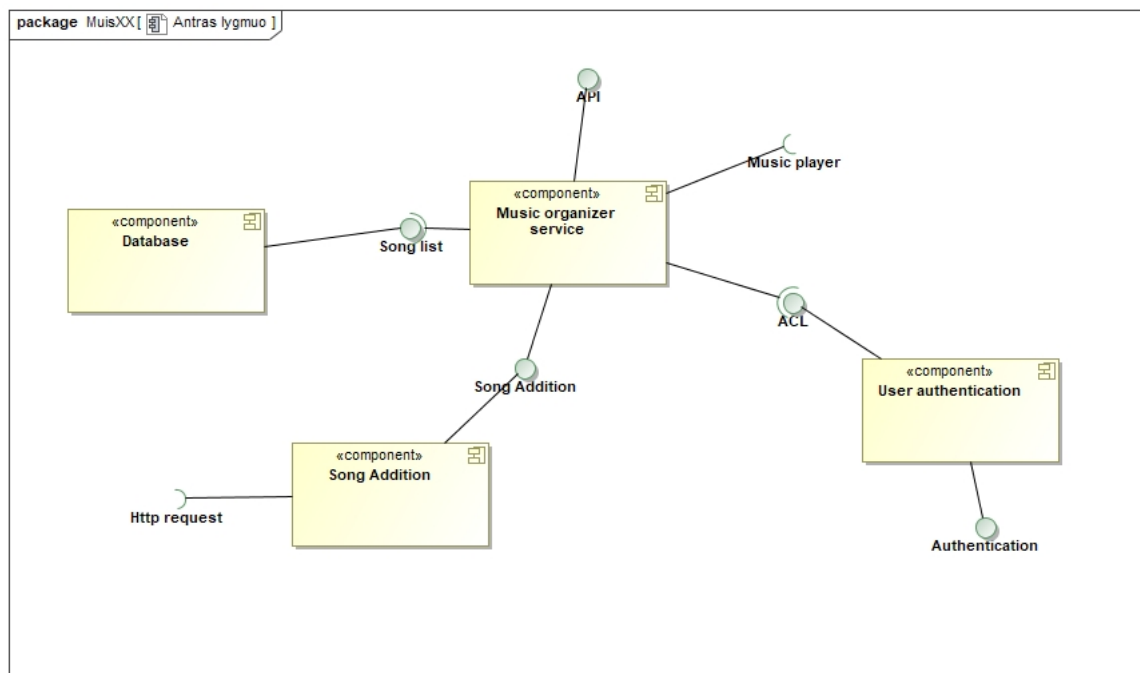
### 4.2. Pirmas lygmuo



20 pav. Pirmo lygmens komponentai

20 pav. parodytas pirmas sistemos komponentų lygmuo, kuriame MusiX sistema, pavaizduota 19 pav., yra išskaidoma į smulkesnes dalis. Atsiranda komponentas, atsakingas už vartotojo sąsajos vaizdavimą bei apdorojimą, ir kitas komponentas, atsakingas už bendravimą su duomenų baze.

### 4.3. Antras lygmuo

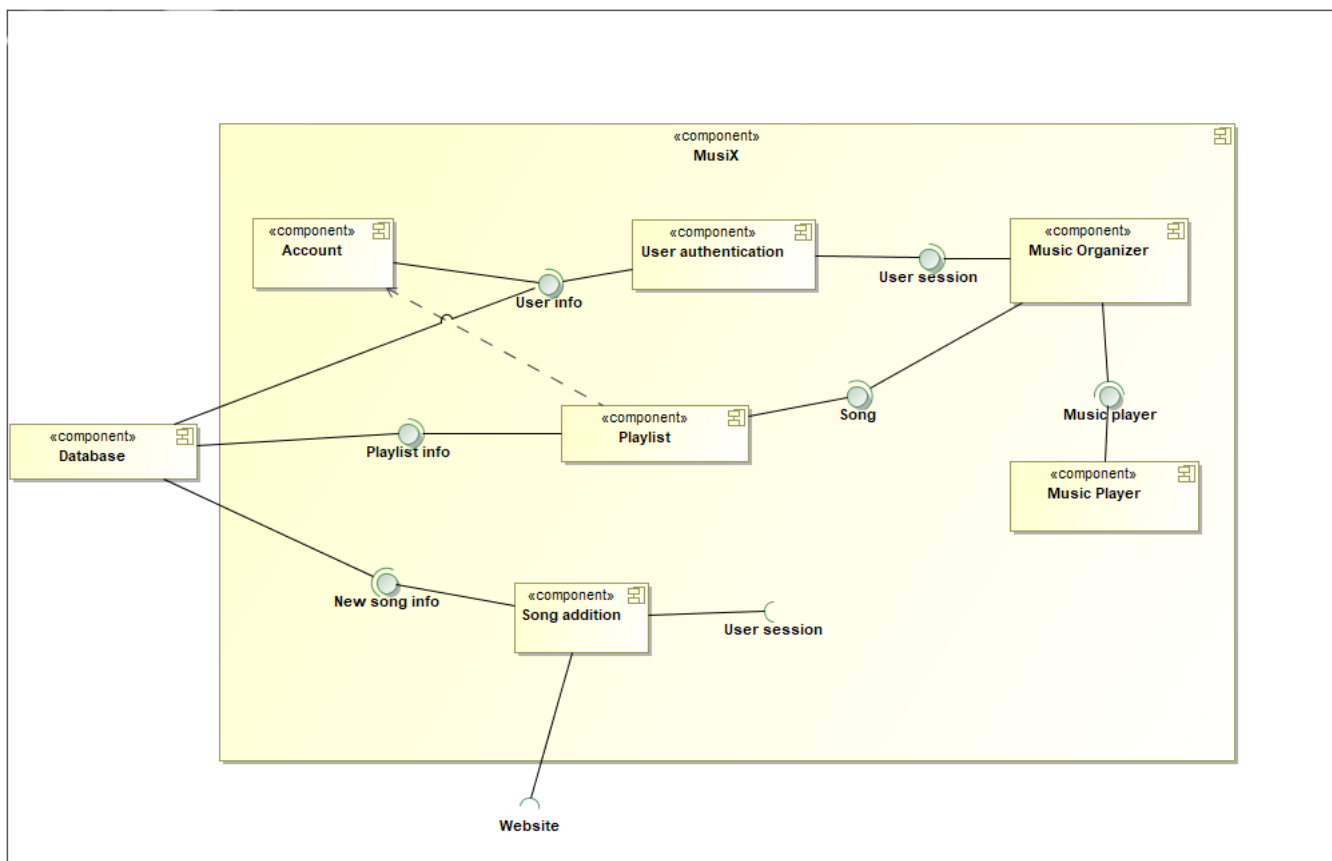


21 pav. Antro lygmens komponentai

21 pav. matomi dar detalesni atskirų komponentų ryšiai. Muzikos organizavimo komponentas yra visus komponentus siejanti grandis, su kuria vienaip ar kitaip susiję visi kiti komponentai. Dainos pridėjimo bei autentifikacijos komponentai siunčia užklausas į išorines instancijas, iš kurių turi gauti vienokius ar kitokius atsakymus prieš toliau tęsiant darbą ar grąžinant atsaką muzikos organizavimo komponentui.



#### 4.4. Programos komponentų diagrama



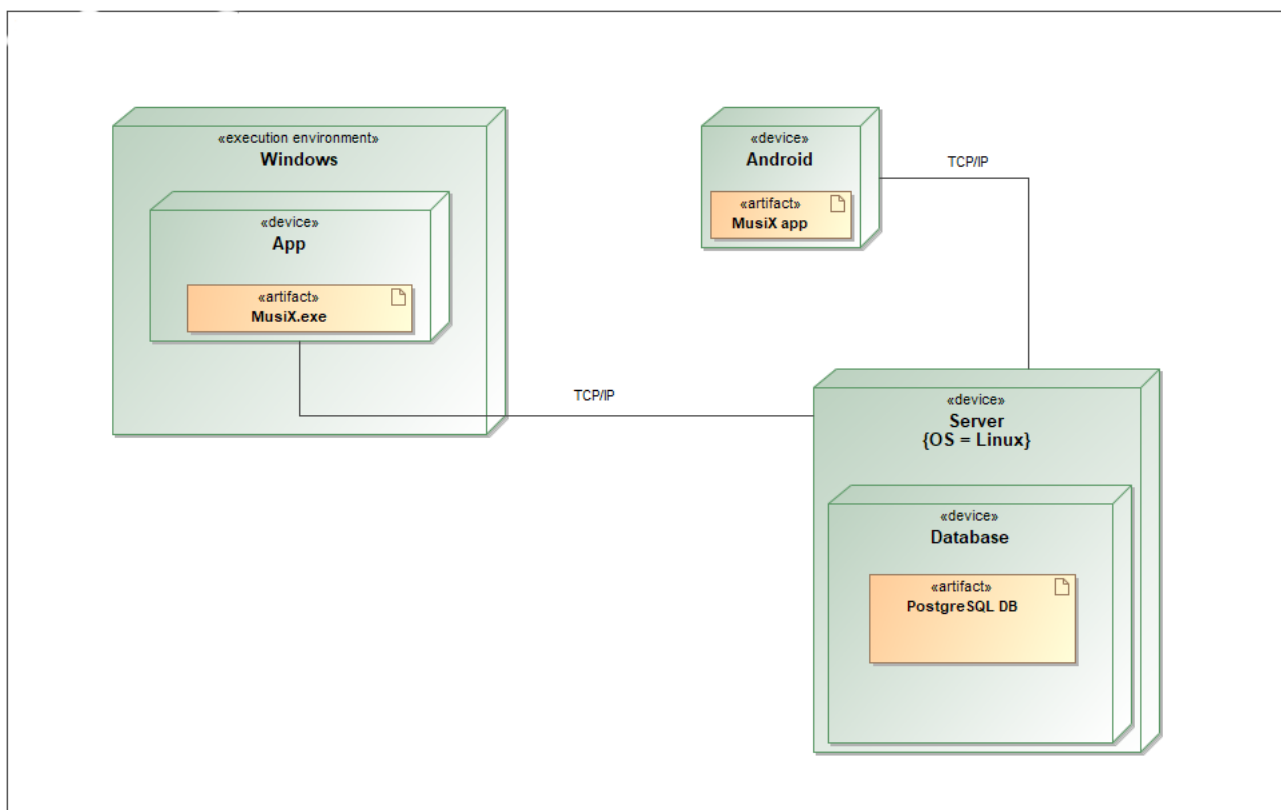
22 pav. Programoje veikiantys komponentai ir jų ryšiai

Komponentų diagrama, matoma 22 pav., parodo platų visų komponentų bendravimo tarpusavyje bei su duomenų baze vaizdą. Čia matome, kad duomenų baze iškart po pokyčių programoje (dainų pridėjimas, grojaraščių sukūrimas ar pakeitimas) gauna informaciją ir yra atnaujinama. Taip ji visuomet išlieka atnaujinta ir net netikėto programinės įrangos darbo pabaigos atveju duomenys nedingsta.

## 5. Komponentų išsiskirstymas tinkle

Šiame skyriuje bus apibrėžiama programinės sistemos naudojama įranga bei komunikacija tarp tinklo mazgų.

### 5.1. Mazgai ir artefaktai



23 pav. Programos bei serveriai, patenkantys į šį projektą

Diagramoje Nr.23 matoma, kad visai sistemai itin svarbi duomenų bazė veiks ant Linux operacinėje sistemoje veikiančio serverio su įrašyta PostgreSQL duomenų baze, kuri yra greita bei patikima. Su šia duombaze didžiąją laiko dalį ryšį palaikys ir Windows, ir Android operacinių sistemų MusiX versijos. Tai užtikrins, kad net naudotojui nusprendus kelių minučių bėgyje pakeisti įrenginius, per kuriuos klausosi šios platformos teikiamos muzikos, jo muzikinė kolekcija visuomet išliks atnaujinta

## Rezultatai ir išvados

Buvo atlikta projekto MusiX analizė pagal 4+1 pjūvį, apžvelgiant visa programų sistemą 5-iais skirtingais aspektais: užduotimis ir jų vykdymo scenarijais, struktūriniu bei diniminiu sistemos modeliais, programų sistemos komponentais ir jų išsiskirstymu tinkle. Detaliai aprašyta sistema tiek iš naudotojo, tiek iš programuotojo perspektyvų. Sukurtas programos prototipas turi padėti prie šio projekto dirbantiems programuotojams suvokti, kaip turi atrodyti vartotojo sąsaja bei veikti įvairios funkcijos.

MusiX turėtų pakeisti, kaip paprasti žmonės kaupia bei klausosi muzikos. Šios programos dėka savos muzikinės kolekcijos kaupimas tampa nesudėtingas, intuityvus bei pasiekiamas vi-  
sur, kur yra interneto ryšys bei Windows ar Android operacines sistemas naudojančios prietaisai. Autoriai tikisi, kad Lietuvos bei pasaulio gyventojai įvertins šios programos pranašumus prieš konkurentus ir pasirinks būtent šį produktą kaip savo pagrindinę muzikos klausymosi platformą.