

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

**SAITYNO TAIKOMŲJŲ PROGRAMŲ PROJEKTAVIMAS**

**Projekto Ataskaita**

**Atliko:**

IFF-4/3 grupės studentas

Žilvinas Abromavičius

**Priėmė:**

dėst. P. Tamošiūnas

**KAUNAS, 2017**

Turinys

[1. Sprendžiamo uždavinio aprašymas 3](#_Toc500760080)

[1.1. Sistemos paskirtis 3](#_Toc500760081)

[1.2. Funkciniai reikalavimai 3](#_Toc500760082)

[2. Sistemos architektūra 3](#_Toc500760083)

[3. Vartotojo sąsajos projektas 4](#_Toc500760084)

[4. API Specifikacija 4](#_Toc500760085)

[5. API panaudojimo pavyzdžiai 4](#_Toc500760086)

[6. Išvados 4](#_Toc500760087)

# Sprendžiamo uždavinio aprašymas

# Sistemos paskirtis

Kursinio projekto tikslas – sukurti muzikos klausimo ir talpinimo platformą.

Veikimo principas – panašus kaip kitų muzikos platformų (Soundcloud, Mixcloud). Registruoti vartotojai gauna galimybę kelti savo muziką, klausyti kitų platformoje įkeltos muzikos, kurti grojaraščius, pamėgti tam tikrus muzikos kūrinius.

# Funkciniai reikalavimai

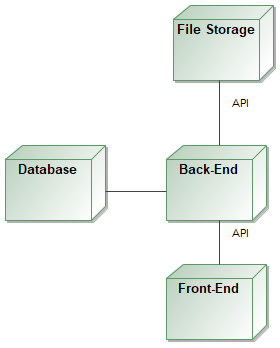
Žemiau nurodytas funkcinių reikalavimų sąrašas:

* Bet kokių platformoje esančių muzikos kūrinių klausymas.
* Muzikos paieška
* Galimybė pamėgti muzikos kūrinius, peržiūrėti pamėgtus kūrinius, klausytis jų kaip grojaraščio.
* Grojaraščių klausymas, kūrimas, redagavimas, šalinimas. Galimybė juos padaryti viešus, privačius.
* Galimybė sekti tam tikrus platformos vartotojus-kūrėjus, matyti jų veiklą naujienų sraute.
* Registracija ir prisijungimas, naudojant OAUTH2 (Google).
* Muzikos skirstymas į tam tikrus žanrus, tų žanrų peržiūra.

# Sistemos architektūra

Sistema susideda iš šių pagrindinių dalių:

* Front-End – React + Redux + Babel + Webpack.
* Back-End – ASP.NET Core 2.0 + Entity Framework.
* Failų saugojimo serveris – čia bus saugojami ir pasiekiami visi muzikos failai. Programos įgyvendinimo metu buvo nuspręsta naudoti Dropbox talpyklą.
* Duomenų bazė –saugojami platformos duomenys. Naudojamas Microsoft SQL Server.

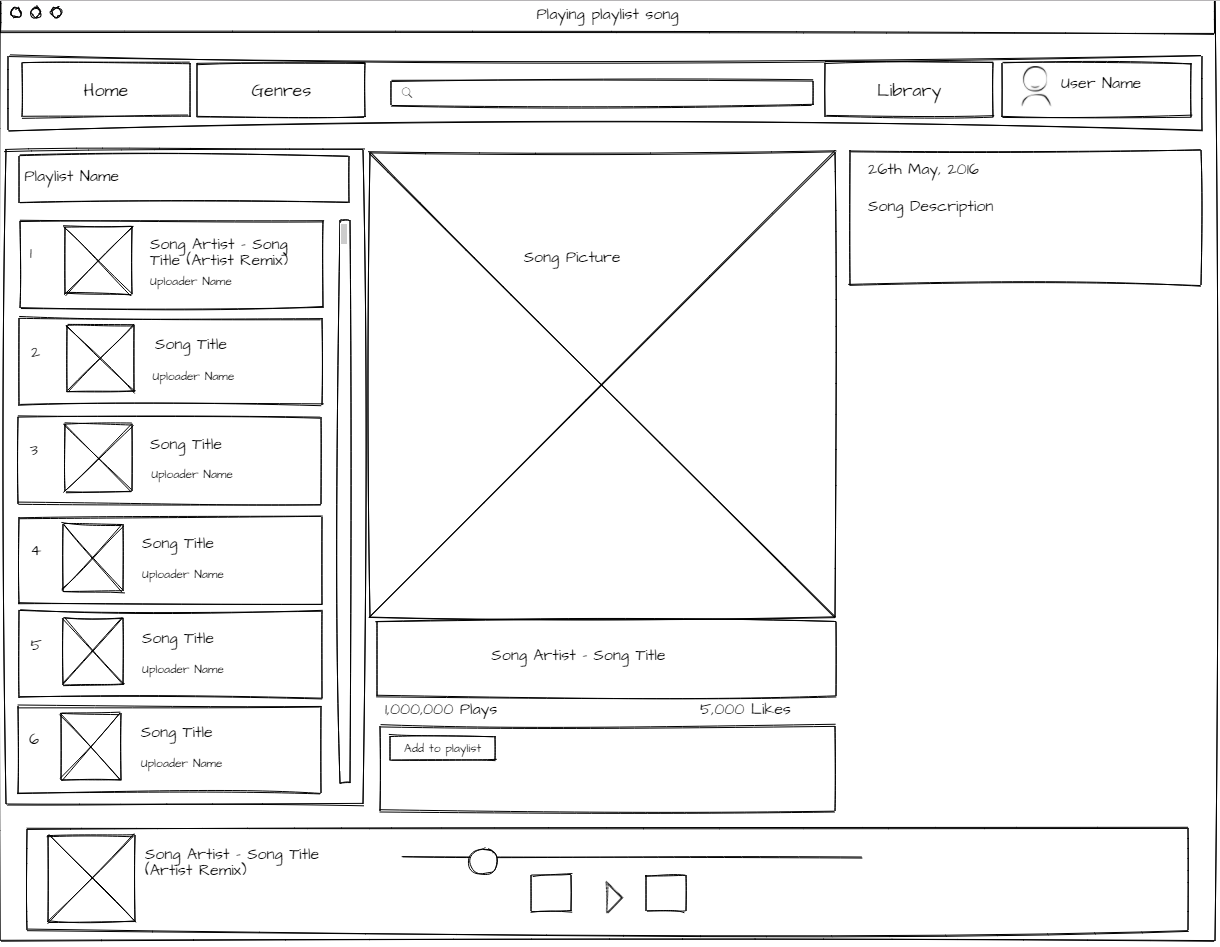


1 pav. Sistemos architektūra

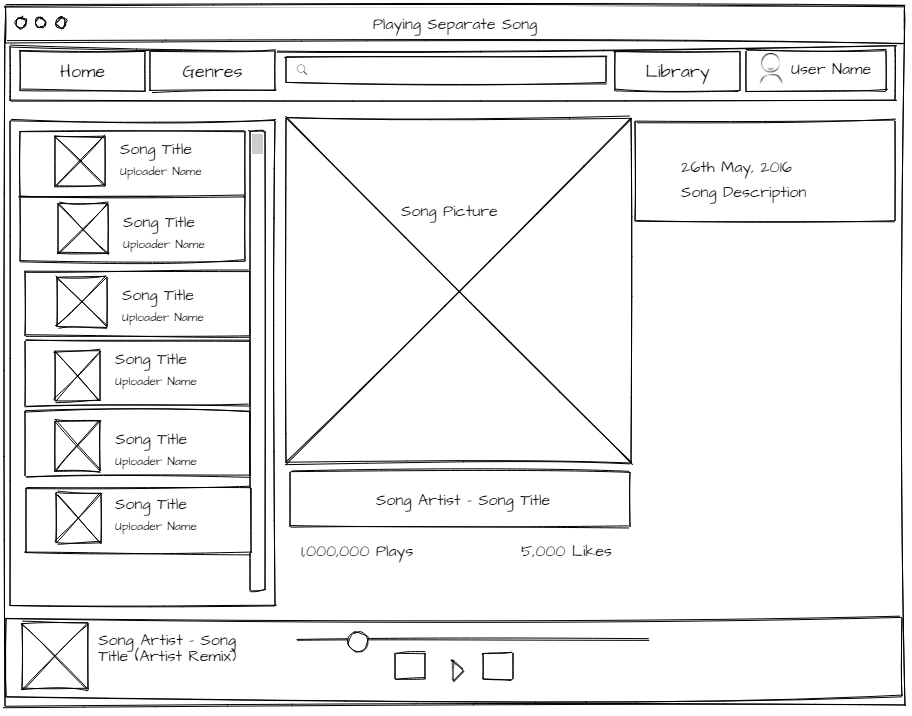
# Vartotojo sąsajos projektas

Vartotojo sąsajos projektavimui buvo naudojamas įrankis MockFlow. Buvo sukurti vartotojo sąsajos prototipai su šiuo įrankiu, vėliau pagal sukurtus prototipus buvo kuriama vartotojo sąsaja.

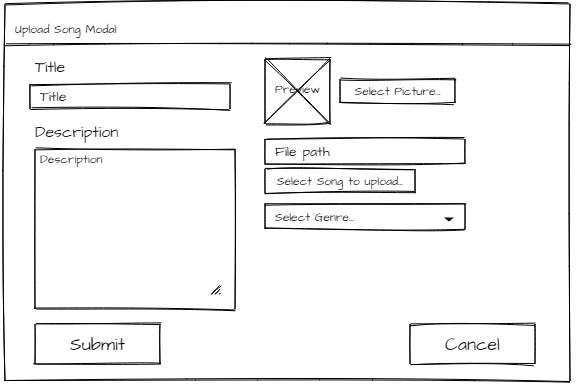
Žemiau pateikti sukurti vartotojo sąsajos prototipai:



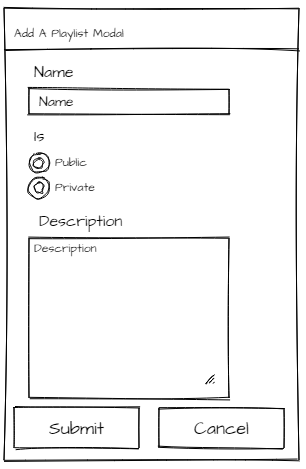
2 pav. Tam tikros grojaraščio dainos grojimo lango prototipas



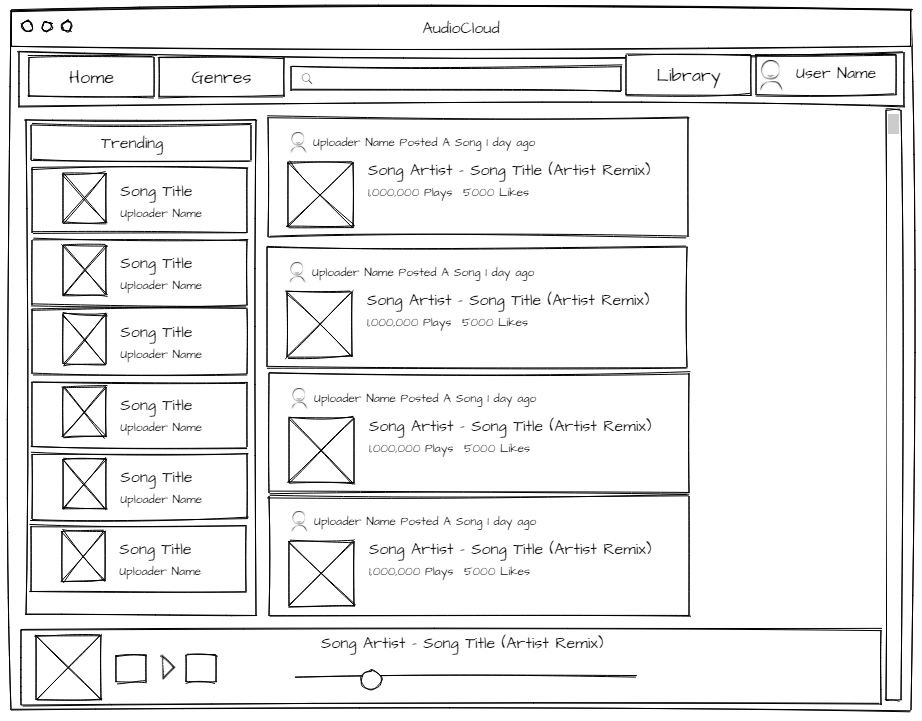
3 pav. Atskiros dainos grojimo lango prototipas



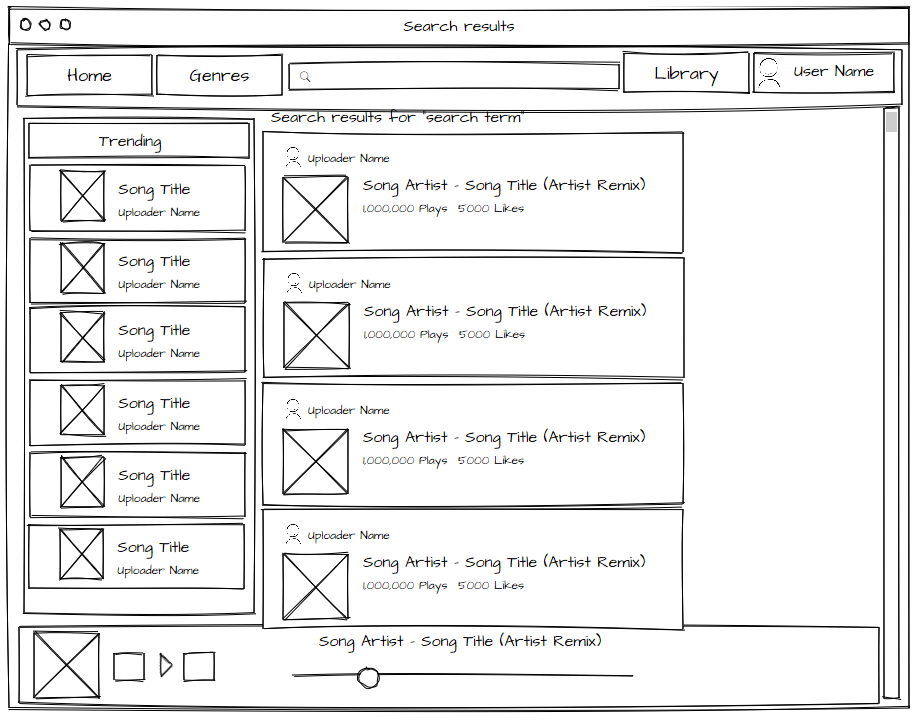
4 pav. Dainos sukūrimo modalinio lango prototipas



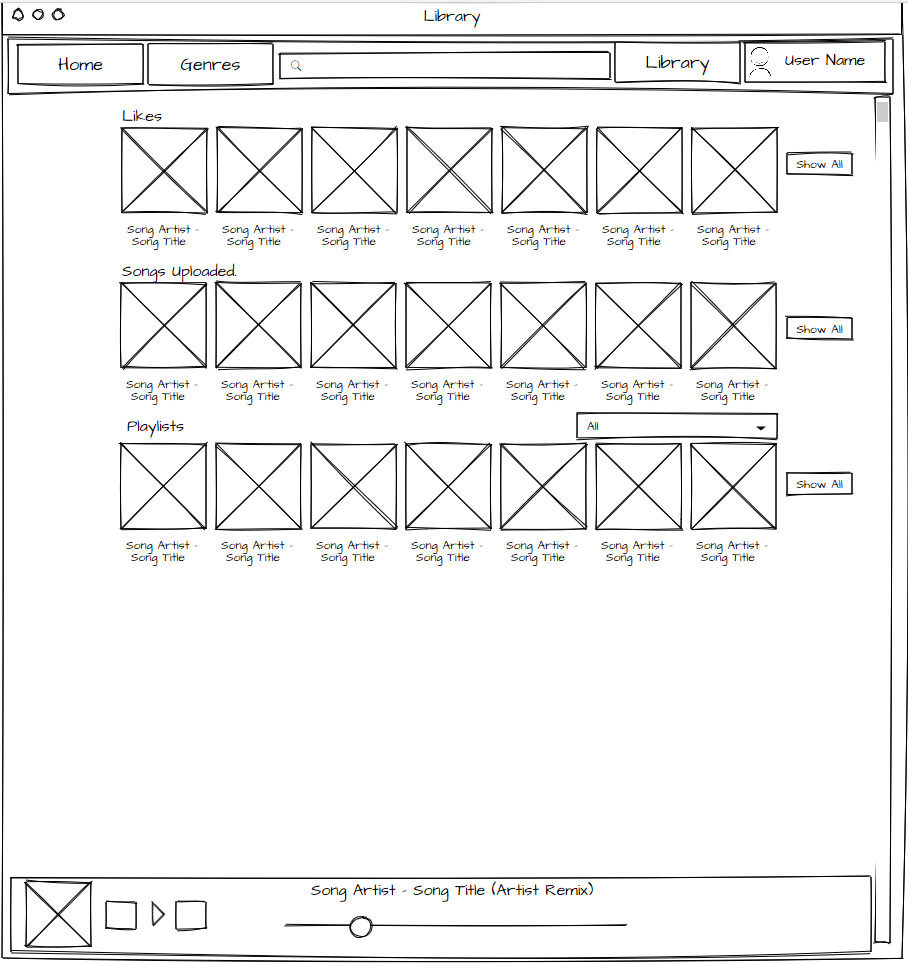
5 pav. Grojaraščio sukūrimo modalinio lango prototipas



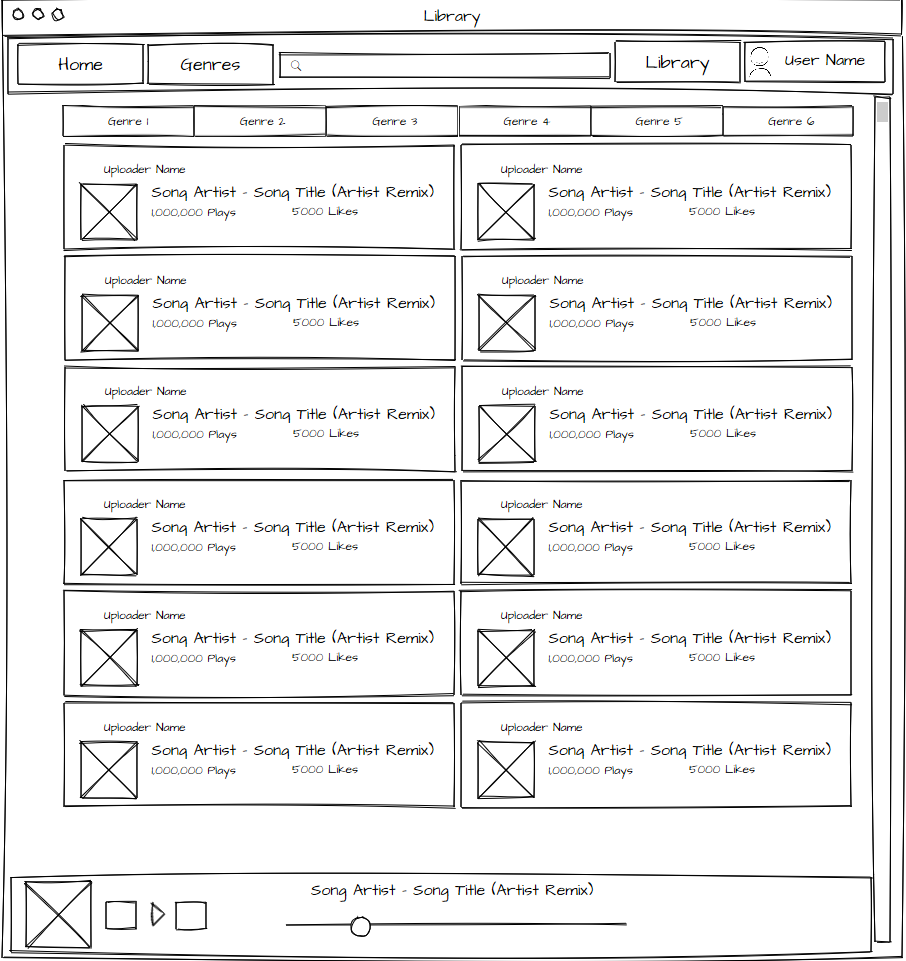
6 pav. Naujienų puslapio prototipas



7 pav. Paieškos rezultatų lango prototipas

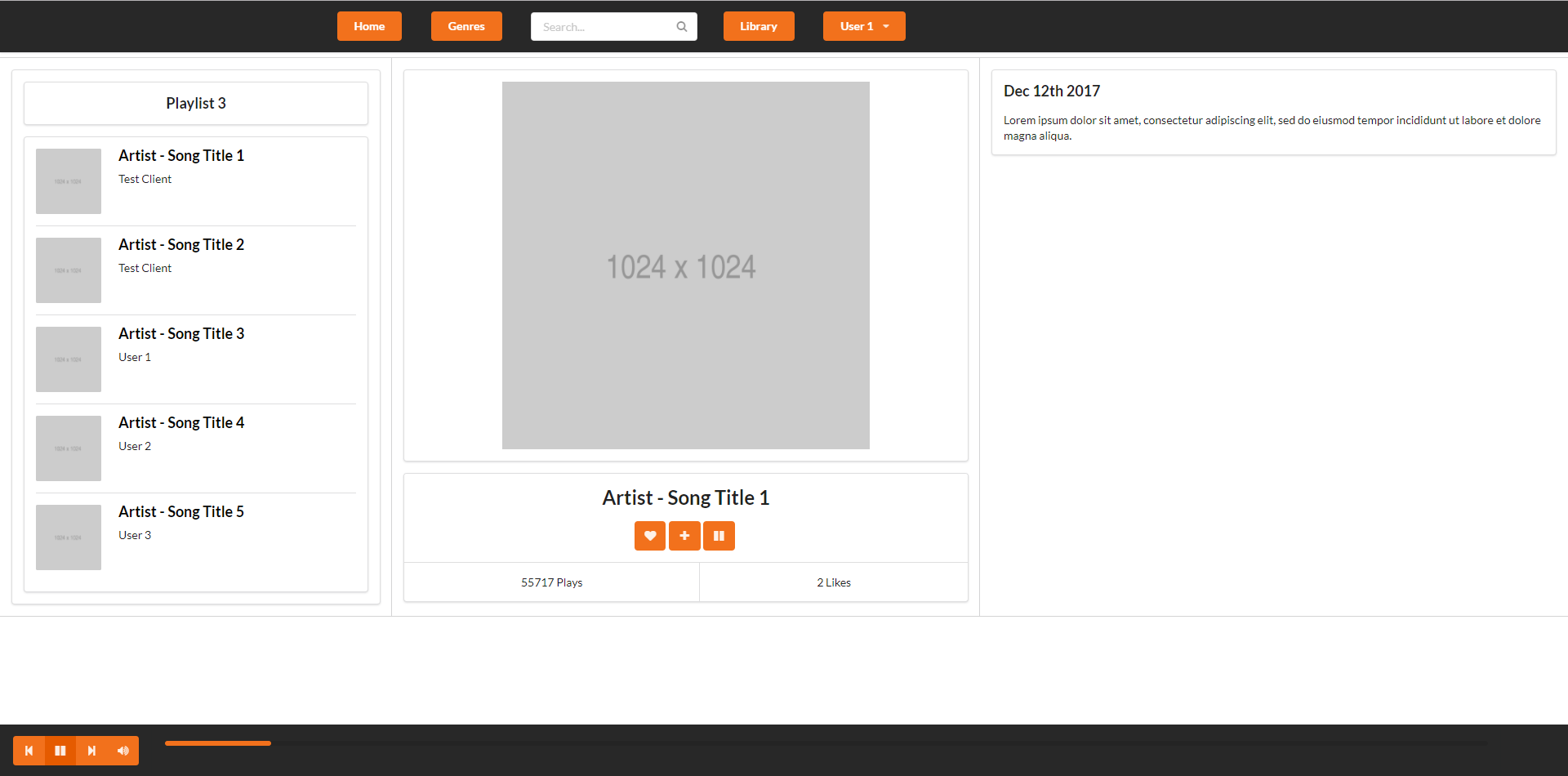


8 pav. Vartotojo bibliotekos puslapio prototipas

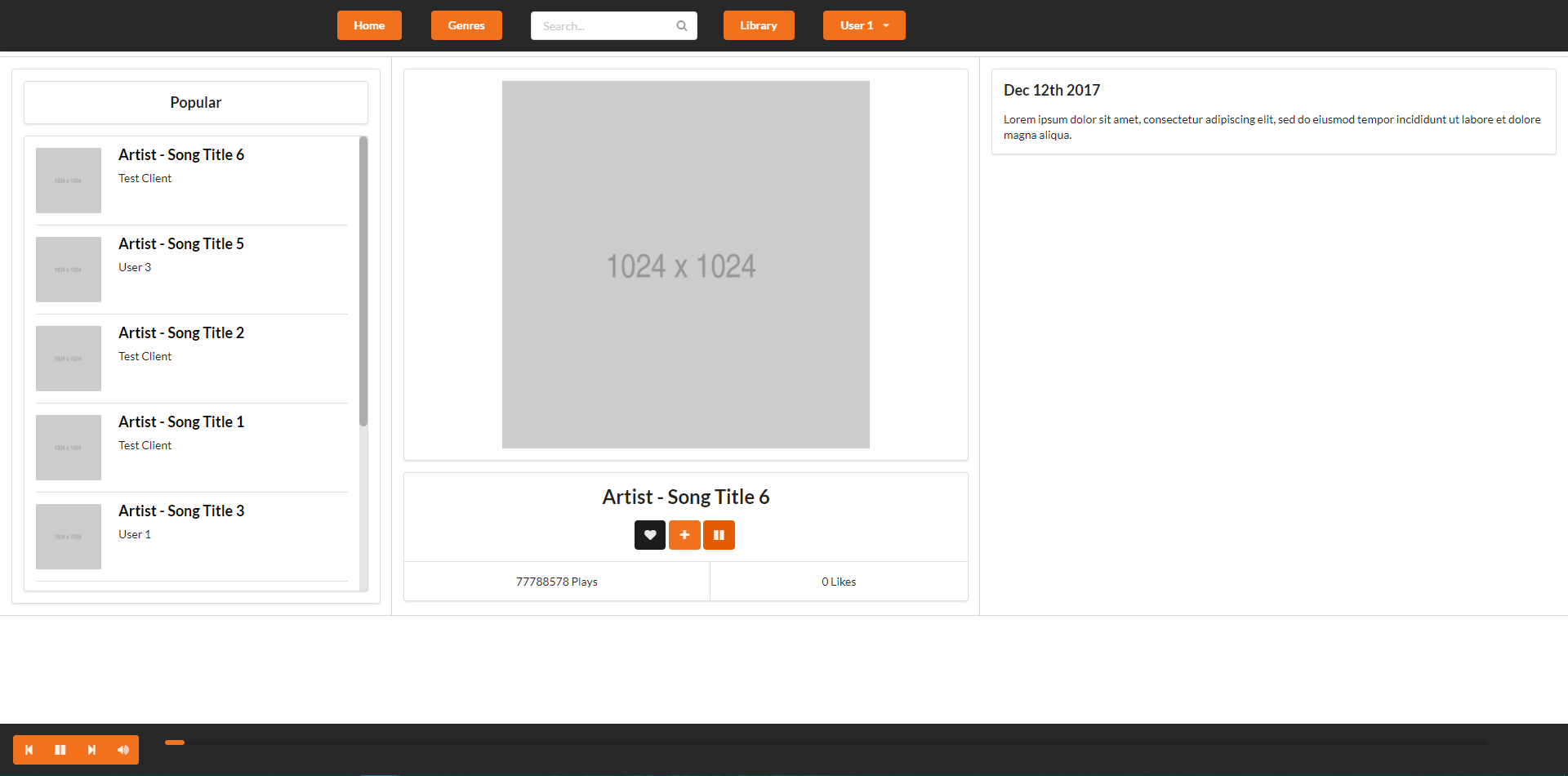


9 pav. Žanrų puslapio prototipas

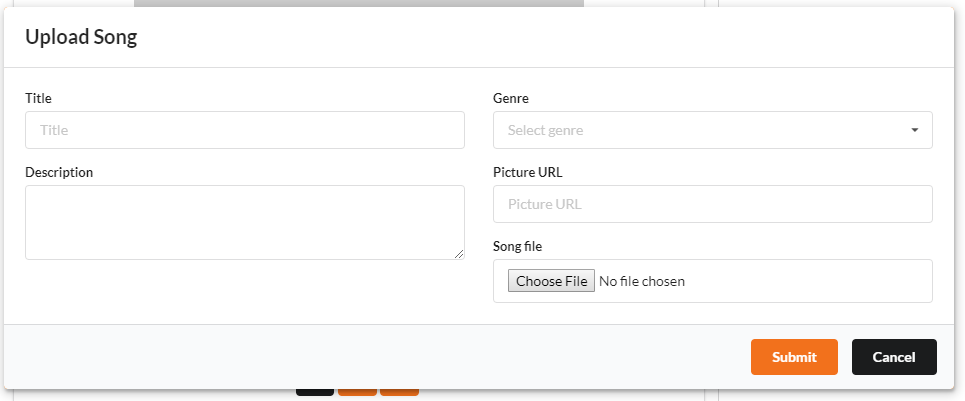
Pagal prototipus buvo įgyvendinama puslapio išvaizda. Grafinės sąsajos kūrimui buvo naudojamas Semantic UI React karkasas. Žemiau pateikti įgyvendintos grafinės sąsajos paveikslėliai.



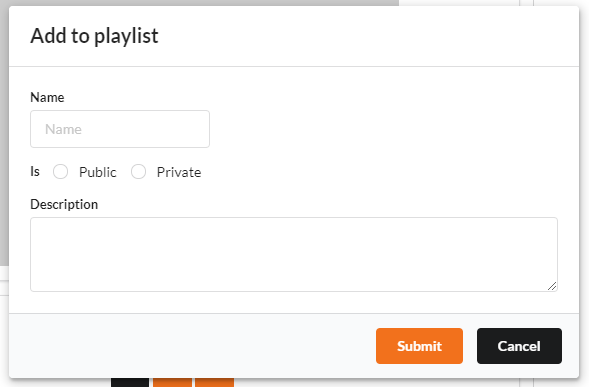
10 pav. Tam tikros grojaraščio dainos grojimo puslapis



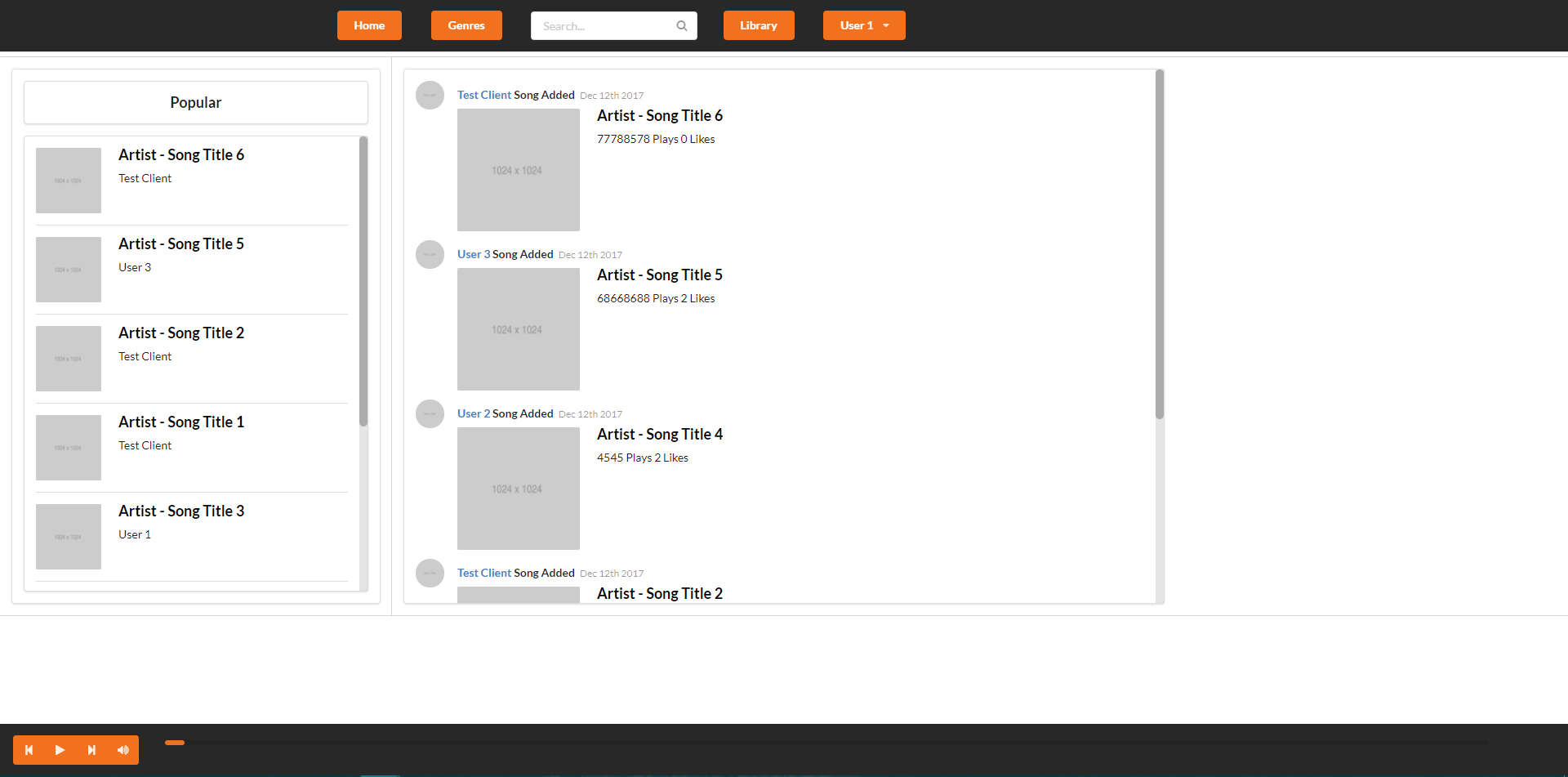
11 pav. Atskiros dainos grojimo puslapis



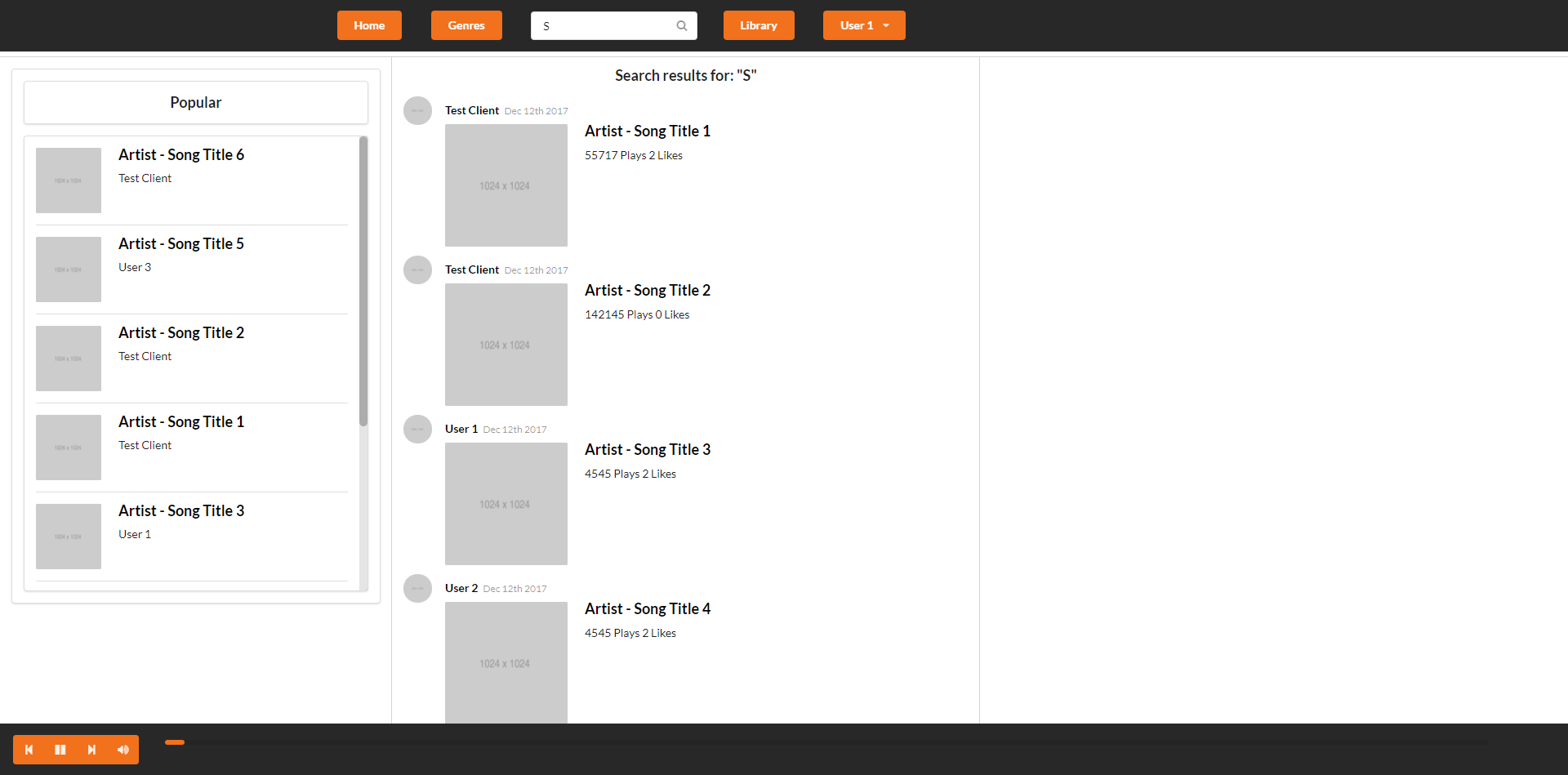
12 pav. Dainos sukūrimo modalinis langas



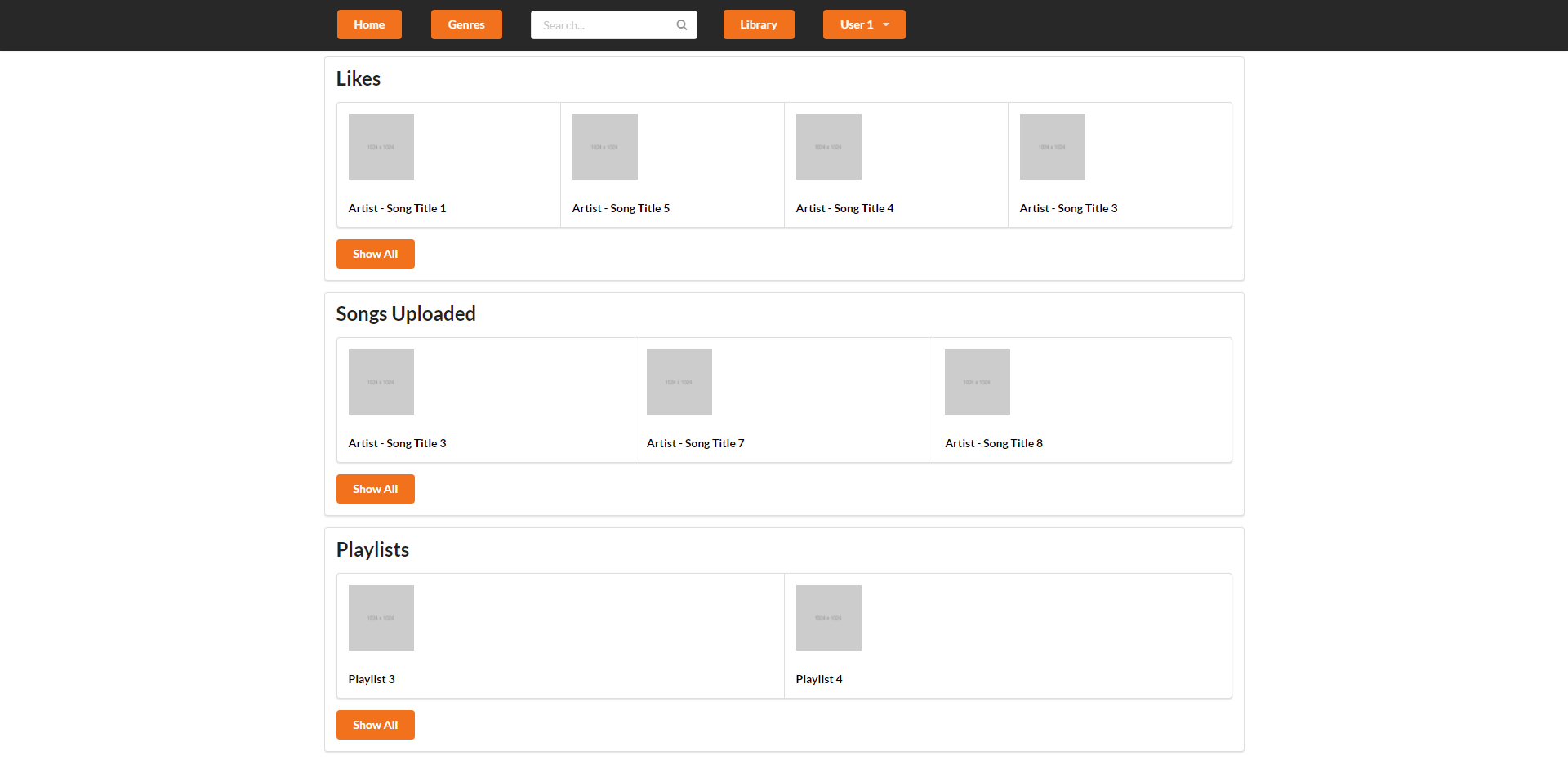
13 pav. Grojaraščio sukūrimo modalinis langas



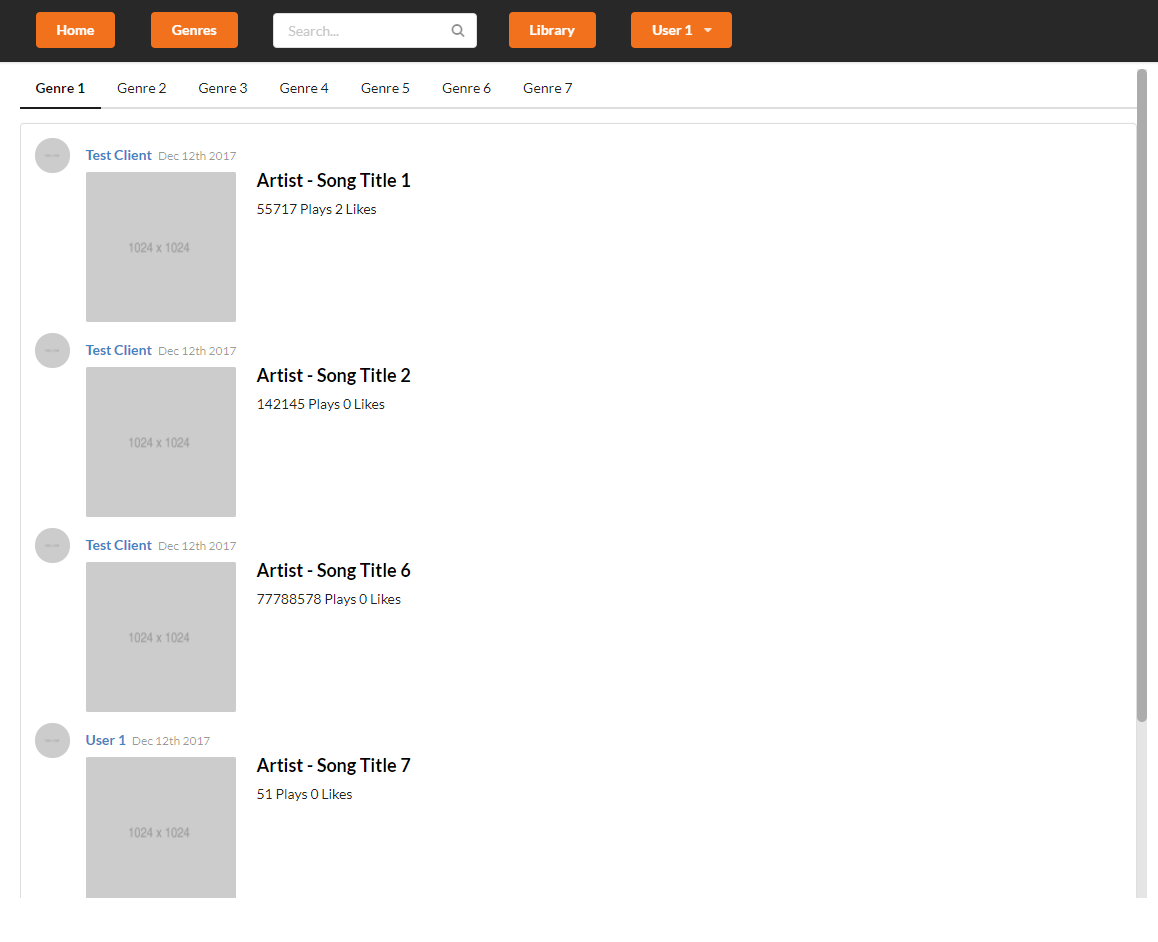
14 pav. Naujienų langas



15 pav. Paieškos rezultatų puslapis



16 pav. Vartotojo bibliotekos puslapis



17 pav. Žanrų puslapis

# API Specifikacija

# API panaudojimo pavyzdžiai

# Išvados