SPOTIFOO PLAYER

Table of Contents

[1. About: 2](#_Toc109427809)

[2. Requirements 2](#_Toc109427810)

[2.1 Play Song 2](#_Toc109427811)

[2.2 Filter By Artist 2](#_Toc109427812)

[2.3 Filter By Album 2](#_Toc109427813)

[2.4 Filter By Genre 2](#_Toc109427814)

[2.5 Search 2](#_Toc109427815)

[3 Input 3](#_Toc109427816)

# About:

Spotifoo is command line interface application, that allows you to stream music instead of downloading it from an unknown source. The user can listen to the songs according to his choice and he can able to choose the song based on his interest. We can only navigate by typing a number from an option list and then press enter to execute the command. Typing 0 and pressing enter in any part of the application will send you back to the main menu. This application is providing features to users like display all the songs, by category (album, artist and genre) and searching the songs by name.

# Project Overview

The application is a command line interface (terminal) application that does not have a graphical user interface. The user only navigates by typing a number from an option list and then press enter to execute the command. Typing 0 and pressing enter in any part of the application will send you back to the main menu.

## Play Song

When pressing option #1 on the main menu, the app must show a list of songs by reading the text file located inside the asset folder. When the user chooses a specific song, the music file should open in the default music player installed in your computer. In addition, it should also open the corresponding album art. Or show a placeholder image if the album art is not found.

**Note:** Only open the album art, if the app can play the song.

## Filter By Artist

When pressing option #2 on the main menu, the app will replace the main menu options with a list of artists available based on the .txt file. Typing the number corresponding to an artist will in turn, show a list of songs that belong to this artist.

## Filter By Album

When pressing option #3 on the main menu, the app will replace the main menu options with a list of albums available based on the .txt file.

## Filter By Genre

When pressing option #4 on the main menu, the app should show a list of genres. The only four genres available are:

1. Metal
2. Pop
3. Reggaeton
4. Rock

## Search

When pressing option #5 on the main menu, the app should allow me to perform a search based solely on the name of the songs. The app must prompt the user to input some text. Then the app must display a list of songs that matches the whole text or part of the text inside the name of a song.   
Example: The assets folder mentioned in the section “what we provide” has these songs:

* Welcome to the Jungle (Guns and Roses)
* Welcome to the family (Avenged Sevenfold)

If the user types “welcome” it should return both results and any other that matches the text. But if the user types “welcome to the jungle” it should only return the song from Guns and Roses.

## Sample data

Spotifoo has permission from top artists around the world, to share their songs with us. Thus, we created a folder called **assets** with 2 sub folders. One for the music files with a bunch of .mp3 files and another for the album art in .png files.

We also provide a file called data.txt with the information of the songs with information about the song name, artist, album, genre, file name, and album image. Your app must read this file to display the list of songs and other features requested in the mandatory requirements listed below.

**Note:** The folder and text file, contain some invalid data on purpose. Do not try to edit the text inside data.txt or add missing files. Instead, write code to validate the errors.

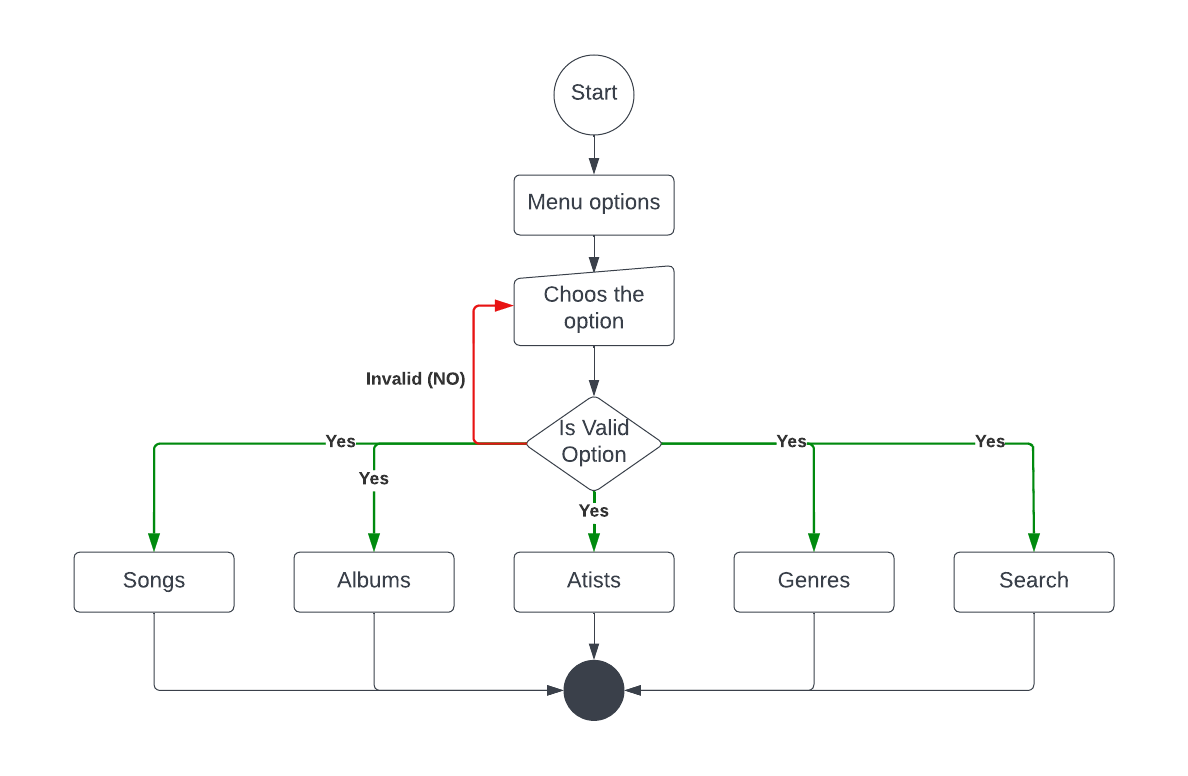
# Implementation and Design

## Technologies

## Design and Flow diagram

## Screens

Navigation:



Models:

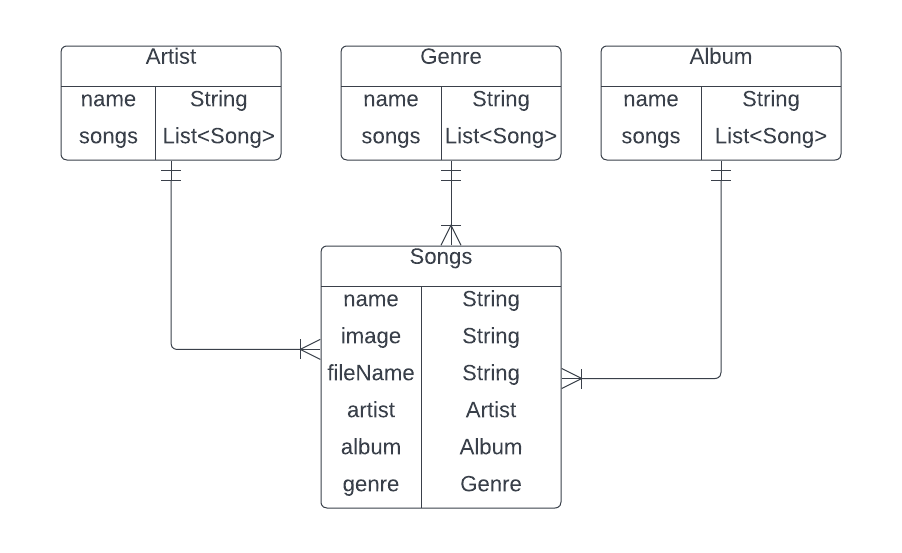
In our Spotifoo App, we have 4 models. They are

Song.java

Artists.java

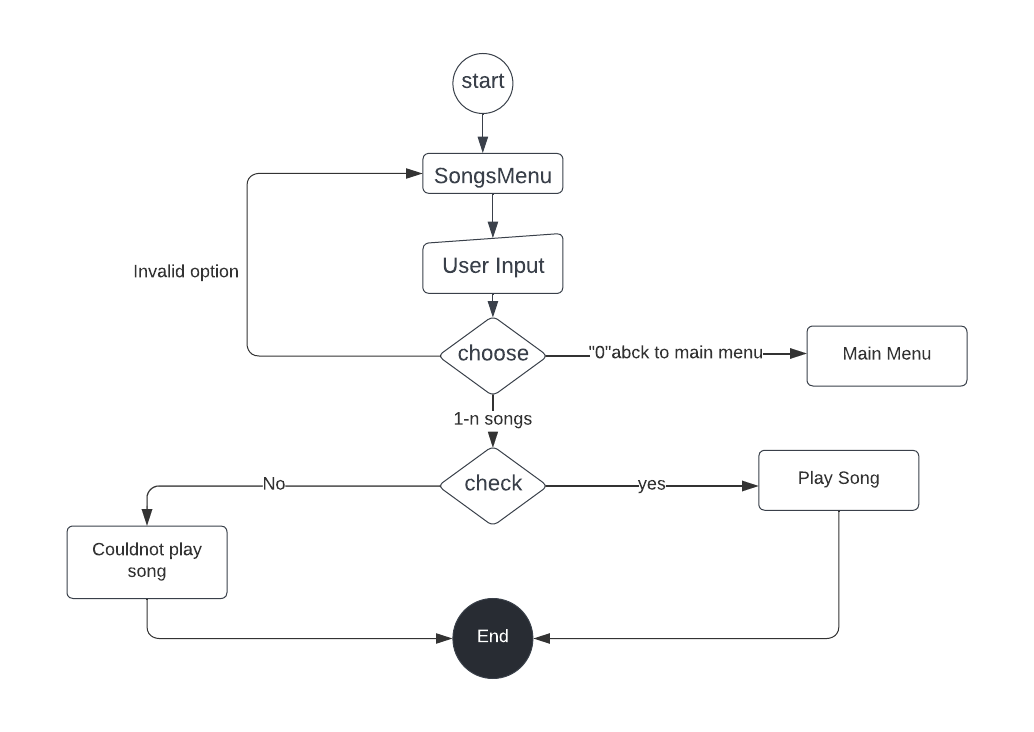
Albums.java

Genres.java



In this I have created models for each of the items in the menu. The navigation picture of each model is shown below.

Song.java

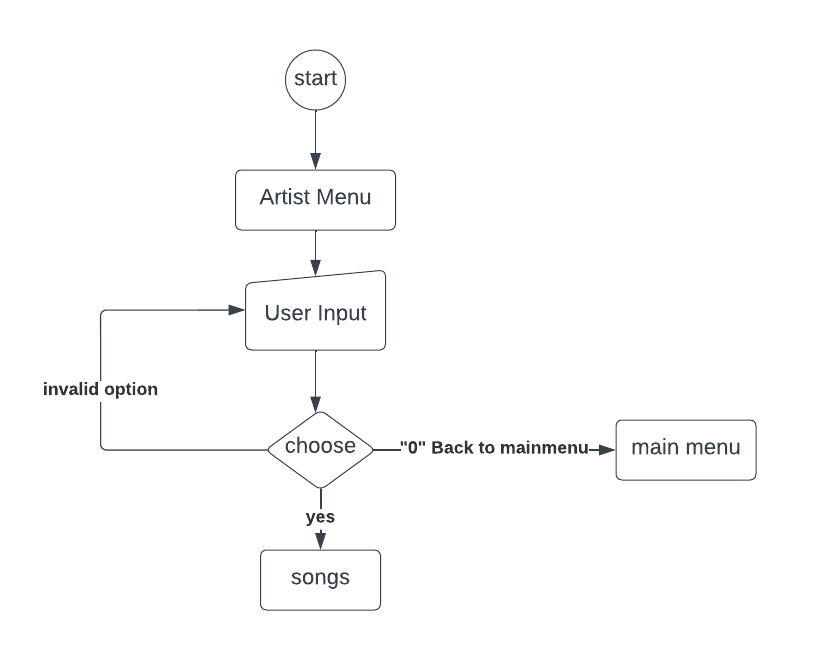


Above is the UML diagram of the model song. Here the user can select any song to his choice. He

enters the option of the song and the song plays. After that, it exits out of the application. Whenever

user clicks invalid option, exception occurs and he is asked to choose another option.

Artists

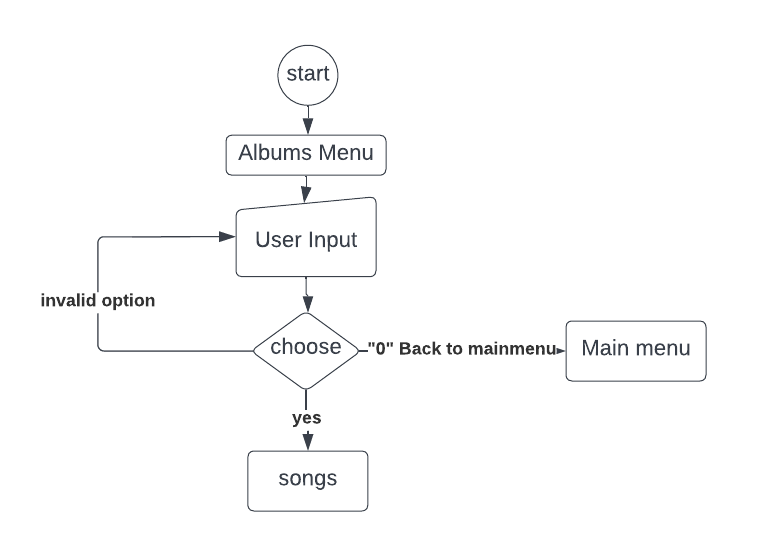


Above is the UML diagram of Artist model. Here the user enters the option and looks at the

Artists that are available. He then selects the required artist and plays the song. If the user

enters invalid number, he gets an exception.

Albums

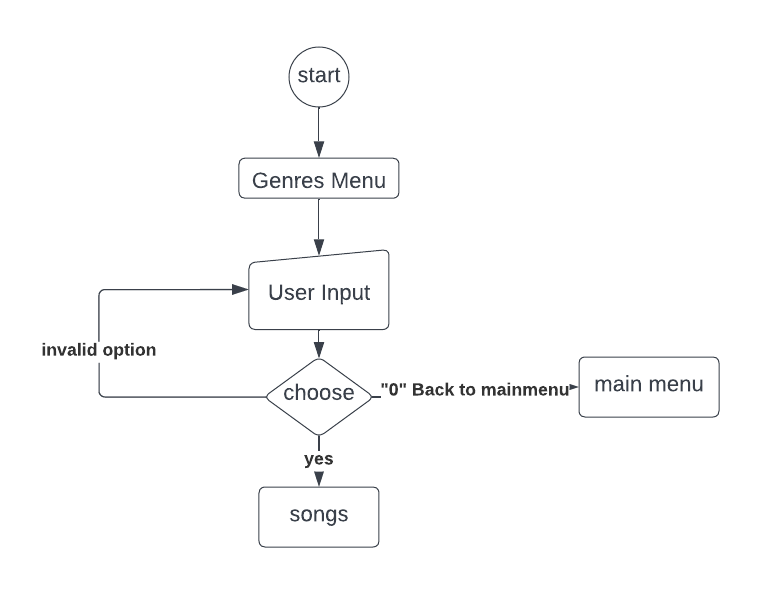


Above is the UML diagram of Albums model. Here the user enters the input and gets all the

Albums that are available in the application. He selects the required option and listens to the songs.

However, he gets an invalid exception, if he enters wrong input.

Genres

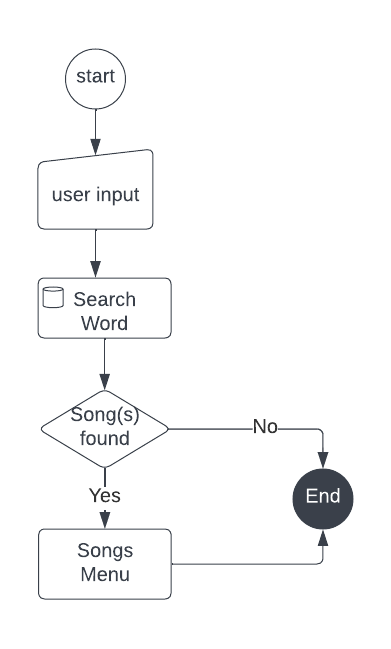


This is the model diagram of the Genres model. Here, the User enters the input and gets the Genres

that are available in the application. He then, selects the requires option and listen to the songs. He

gets an exception, when he enters the wrong option.

Search



Above is the UML diagram for the search option. Here the User can search for a song in the

application. However, if he enters wrong name of the song, he gets exit from the application. When

he enters name of the song, the name of the song that matches the search will be displayed from

albums, artists and genres.