

P1 Proposal: Song Searcher

Brief Project Description:

Song Searcher helps users to find and organize songs for a party playlist. It lists songs by their danceability, and supports finding songs with a similar level of danceability. The app makes use of a Red-Black Tree to efficiently insert the song data and retrieve it in a sorted manner based on a danceability score. The data is loaded from a local data file that the user specifies.

Representative Tasks Performed Using this Application:

1. Find a song of a specific danceability level.
 2. Report the average danceability of all songs in a data set.
-

Backend Developer (BD) Role

The backend developer writes code that reads in a set of songs from a CSV file, inserts them into a Red-Black Tree, and accesses them based on commands from the frontend. To read in the data, the backend developer develops a java class that defines a single song object and that can be inserted into the Red-Black Tree using each song's danceability score as the key. The backend supports data sets formatted in the same way as this [song dataset from kaggle](#) in CSV format. The frontend accesses the backend's functionality through method calls on the backend object.

Interface Design Responsibilities:

- An interface for a class that defines a single song and exposes song properties required by the frontend: artist, title, year, genre, and danceability score.
- An interface for the backend that exposes the required functionality to the frontend: read data from a file, list the average danceability of all songs, get a list of songs with minimum danceability that is at or above a specified threshold.

Presentation Responsibilities:

After integration, the backend developer demonstrates a search with the app to find the songs with minimum danceability score at or above 86.

Frontend Developer (FD) Role

The frontend developer writes code that drives an interactive loop of prompting the user to select a command, then requests any required details about that command from the user, and then displays the results of the command. The commands to include are: a command to specify (and load) a data file, a command to list all songs that have a specific danceability score, a command that shows the average danceability score in the loaded dataset, and a command to exit the app. The results are computed with the help of the BD's code. When the user enters invalid input, instructive feedback about what they should enter is displayed.

Interface Design Responsibilities:

- An interface for a class that implements the functionality of the frontend for the app. The class has a constructor that accepts a reference to the backend and a `java.util.Scanner` instance to read user input. It also has a method that starts the main command loop for the user. To allow for easier testing of the frontend, the command loop is broken down into a separate method for the main menu and each of the sub menus.

Presentation Responsibilities:

After integration, the frontend developer will record a video demonstration of the task: List the average danceability of all songs in a data set.