General Overview

Welcome to the music streaming platform that's giving Spotify and Apple Music a run for its money. This program satisfies the basic needs of all users and achieves this with the following functions:

login() – This is where the magic starts. It is the barrier between man and the ability to stream music at one's own discretion. However, if you are an artist, login() is the gateway to building your discography and viewing your top fans. This function can distinguish between the ids of users and artists. If an id falls into both categories, the user can choose which category they fall into and login as. Which brings us to the next function, parse().

parse() – Milage may vary depending on whether you are a user or artist. Now that you've conquered the login() function, this is where your adventure begins. Users can start sessions, use the search song/playlist function, use the search artist function, and listen to their favourite hits once they've found them. After a session of listening to bangers users can end session manually, logout to log back in with a different id, or exit the program entirely. Artists can add songs for users to listen to, view their top 3 users who listen to their songs the longest time, and view their top 3 playlists that include the largest number of their songs. Artists may also logout to log back in with a different id or exit the program entirely.

startSession() – Starts a listening session for the user. Is triggered by either the user entering 'start' after login or the user playing a song if there is no ongoing session.

searchKeywordSong() – The search algorithm to find playlists and songs so the user can start listening to songs, add songs to playlists, or find info about the songs. Is triggered by the user entering 'ssp' after login.

keywordArtist() – The search algorithm to find artists and their songs so the user can start listening to songs, add songs to playlists, or find info about the songs. Is triggered by the user entering 'sa' after login.

selectSong() – Allows the user to listen to songs, add songs to playlists, or find info about the songs. If triggered after songs are displayed by the search algorithms.

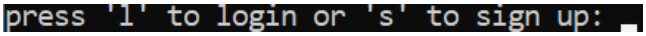
addSong() – Allows the artist to add a song to their discography. Is triggered by the artist by entering 'add' after login.

addPerformer() – Allows artist to add a performer to a song if the song includes multiple artists. Is optional by artist when adding a new song.

topFansPl() – Allows the artist to view top fans who listen to their songs and top playlists featuring their songs. Is triggered by the artist after entering 'top' after login.

A Small User Guide

Using this program should be relatively straight forward.



For new users press 's' to sign up. Then enter a unique uid that is less than 5 characters long, your name, and password for respective prompts to successfully create a new account.

Then press 'l' to login, and use the following format:

Username: *uid*

Password: *password*

```
type 'start' to start a listening session

type 'ssp' to search for a playlist or song

type 'exit' to exit and close the program

type 'end' to end the current session

type 'sa' to search for an artist

type 'logout' to log out
```

If you are logged in as a user, you will be given the options above. Follow the prompts and enjoy your listening experience. To play a song use either ‘ssp’ or ‘sa’ to find your desired song to play.

Note:

- You don’t need to enter ‘start’ to start a listening session, you can just search and play a song and a session will be automatically started.
- You don’t need to enter ‘end’ to end a listening session, you can logout or exit and the ongoing session will automatically end.

```
type 'add' to add a song
type 'top' to find top fans and playlists
type 'logout' to log out
type 'exit' to exit and close the program
```

If you are logged in as an artist, you will be given the options above. To add a song type ‘add’ and follow the prompts regarding title and duration.

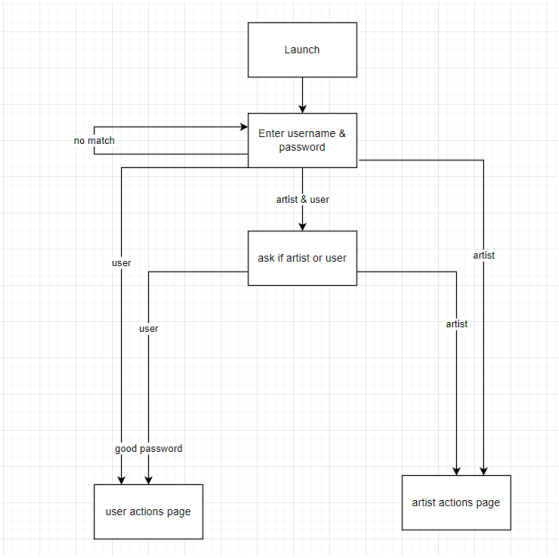
```
Would you like to add a performer?
 Press 'y' or 'n':
```

At this screen press y to add performers if there are multiple and ‘n’ if it’s just you who performed the song. Continue until all artists are added and your song has been successfully added.

To view top 3 fans in terms of total listening duration and top 3 playlists in terms of number of songs that feature you, enter ‘top’.

A Detailed Design

login() flowchart:



User functions:

startSession():

- Get the max session number of existing sessions from the user and increment it by 1 for the new session.
- Insert values (user id, session number, current time) into sessions table.

searchKeywordSong():

- Prompts for search input
- Compares the search input to titles of all songs and playlists that contain the search input.
- Displays the songs and allows the user to select a song by entering the index of displayed songs.
- Gets the sid of the selected song and calls selectSong(sid).
- If there are no results, back to user options.

keywordArtist():

- Prompts for search input
- Compares the search input to names of all artists that contain the search input.
- Displays the artists and allows the user to select an artist by entering the index of displayed artists.
- Gets the aid of the selected artist and calls selectArtist(aid).
- If there are no results, back to user options.

selectArtist(aid):

- Allows user to select the artist by entering the index of displayed artists.
- Displays the songs by the artist and allows the user to select a song by entering the index of displayed songs.
- Gets the sid of the selected song and calls selectSong(sid).

selectSong(sid):

- Gives the user the option to add a song to playlist, listen to a song, get info about a song, or exit the program.

listenSong(sid):

- If there is no ongoing session, then start a session before listening to song.

songInfo(sid):

- Displays performers of the song and playlists that contain the song.

exitProgram():

- If there is an ongoing session, then end the session before exiting the program.

logout():

- If there is an ongoing session, then end the session before taking the user back to the login screen.

Artist functions:

getFansPl():

- Gets top 3 fans and top 3 playlists.
- With respect to greatest total listen time for fans.
- With respect to number of occurrences of songs made by the artist that is logged in.

addSong():

- Adds a song with the option of adding additional performers on the song on top of the artist that is logged in.

Testing Strategy

Once we had a beta version of our music streaming program, we primarily tested out our program by dividing different functions amongst ourselves. The first thing we tested for was any errors that were raised by inputs, and we fixed those accordingly and consistently pushed updated versions of our code onto a git repository. Once there were no errors, we made sure that what was happening in the database also reflected what was going on in the code, whether that was users creating a new account, users having a listening session, artists adding a song, etc. With a combination of print statements, insert statements, and connection commits, we were able to achieve expected results for each query. As for testing the program, we tested for things from basic edge cases to trying to break the program on purpose. After many bug fixes we finally went over SQL injections making sure that our program was impregnable, and we made passwords invisible.

Group Work Break-Down

Othman:
- Implemented UI integration: 4 hrs
- Implemented Start Session, Search Song and Playlist functions: 2 hrs
- Implemented addSong() and topFans() function for artists: 1 hrs
- Worked on paginating search results: 1 hr
- Debugging/Testing, day 1: 4 hrs
- Debugging/Testing, day 2: 4 hrs
- Debugging/Testing, day 3: 4 hrs
- Wrote Small User Guide and Group Strategy/Breakdown section of report: 1 hr
- Total Time: 21 hrs

James:
- Set up GitHub, initialized database: 0.5 hrs
- Artist Search Function and End Session function: 1.5 hrs
- Setting up database test values and database population script: 4 hrs
- Worked on paginating search results: 1 hr
- Debugging/Testing, day 1: 4 hrs
- Debugging/Testing, day 2: 4 hrs
- Debugging/Testing, day 3: 4 hrs
- Wrote General Overview in report: 1 hr
- Total Time: 20 hrs

Eric:
- Implemented addPerformer and selectSong function: 2 hrs
- Implemented parse() function and selectArtist function: 1.5 hrs
- Implemented Song Info Function: 1hr
- Helped on paginating search results: 1 hr
- Debugging/Testing, day 1: 4 hrs
- Debugging/Testing, day 2: 4 hrs
- Debugging/Testing, day 3: 4 hrs
- Wrote Design and Testing Strategy sections of the report: 2 hours
- Total Time: 19.5 hrs