

Overview

This program is a music service provider for both artists and users. It is used for listening to music, creating music, and learning about the music industry.

User guide

Accounts and authentication

Start by entering 1 to login or 2 to sign up .

If entered 1, first input valid user ID or valid artist ID of 4 characters less, then input valid password corresponding to that ID to login to either their user account or artist account.

If entered 2, first input a valid unique user ID of 4 characters or less, then input password, and finally input name.

During any point in either the login page or the signup page, user can go back to main page by entering -1

If the user inputs anything else during main page, they exit the program

User

-After a successful login, users can enter 1 to start or 2 to end a listening session, 3 to search for songs and playlists, 4 to search for artists, 5 to logout, anything else exit the program.

-Whenever a song, playlist, or artist shows up on screen as a search result, user has the option to select them and perform the following actions on the song: 1 listen, 2 see more information about it, and 3 add it to a playlist; see all songs in a playlist; see all songs performed by the artist.

Artist

Once the artist logs in, they are prompted to enter 1 to add a new song, 2 to find top fans and playlists, and 4 to logout. If the artist decides to add a new song, they are prompted for the title and duration. If a song with the same name and duration exists, the artist is prompted to enter a 1 to proceed with the insertion. The artist is then prompted to enter any additional aids. If the aids are invalid, an error message is printed and they are returned to the artist's main menu.

Testing Strategy:

We used Aron Gu's testing data posted on eClass forum to test our program extensively. We put ourselves in the position of a (imperfect) user to test normal cases and as many edge cases as time permits. Each group member is also responsible for testing code produced by other members and providing instant feedback through our discord server.

We also tested the rubric cases presented to make sure that they produced the results wanted.

Group Break-Down Strategy

Rayan:

- Handle initial database entry (20 minutes)
- Login Functionality (4 hours)
- Signup Functionality (2 hours)

- Main user file (4 hours)
- Edge case testing on all above files + completed program (4 hours)

Yushu:

- User functionality (5 hours)
- Song functionality (6 hours)
- Edge case testing on the completed program(5 hours)

Gurman:

- Artist action
- Add Songs (2 hour)
- display top users and playlists (1 hour)
- debugging (5 hours)

Coordination strategy:

Discord server:

- Created a small server with all 3 members and called frequently at night to check on progress. A lot of messaging back and forth to decide on how to develop our work.

Meetings after class:

- All 3 members share the same morning classes, so we would talk after class to say what we need to do on the day and if we realized anything since the night before or since the last time we talked

Github:

- Created a repository for the project and committed over 100 times with many comments on what we fixed and what we updated. This means we made sure to break down the work into meaningful small problems so that figuring out bugs and later fixes would be very easy.

Design

Primary functions

main(): once valid input is placed into either login() or signup(), user either calls userMainMenu() or artistsMainMenu()

userMainMenu(): give user the options to call startSession(), endSession(), searchSongAndPlaylists(), or searchArtists()

songAction() is called whenever user selects a song

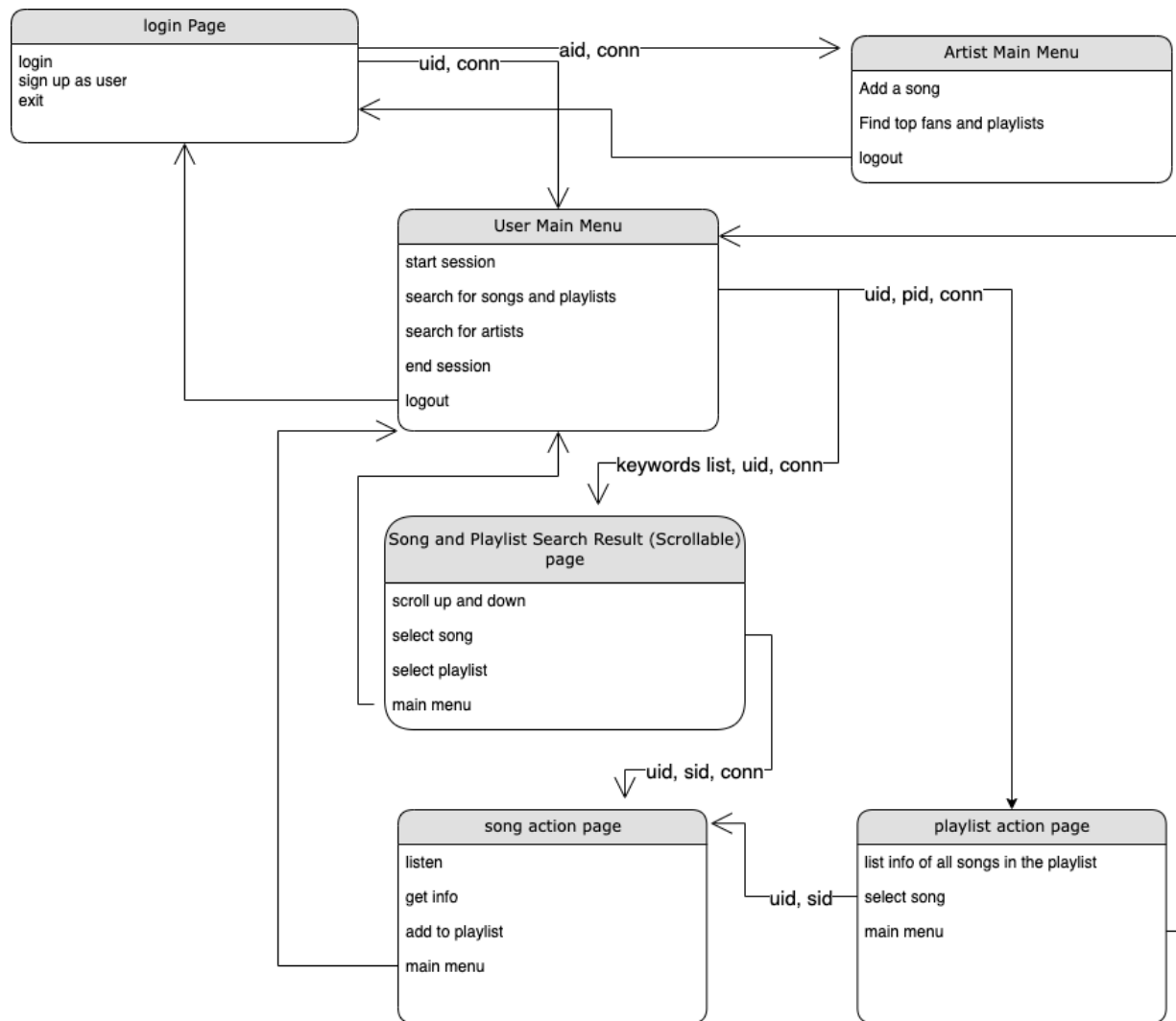
artistsMainMenu(): gives the artist the options to add songs and show top fans and playlists

displayUsersPlaylists(): display top users and playlists

addSong(): add new song

Features not mandated by the specification:

When user attempts to add a song to a playlist, validate that the playlist belongs to the user because users are not supposed to add songs to playlists they don't own



Bugs (24)

Main: (4 bugs)

- In case of artist logging in, the variable needed was not initialized inside the scope of that case. Quickly initialized the variable to match the needed output.
- After recalling the main() function, having the database entry be inside the main function made it so it asked for the database entry again. Changed it to a global variable so it would only be called once.
- In the case a database name was not provided in the command line arguments, it was providing an error of not having any database to connect to. Added a try catch to take the database name in that case.
- Sign up was accepting empty fields for username,password, and name. Validated to remove that ability

Accounts and Authentication: (4 bugs)

- Tried to use rowcount to implement checking if ID is valid but it caused errors for cases of having no matches with the given ID. Decided to change it by checking if the fetch had results.
- Got stuck in both the login page and sign up page so remembered to place an option to go back to the main menu screen.
- Forgot input case insensitivity on ID so I got invalid ID even when the ID was the same (with different cases of course). Added `.lower()` to the user entry, and `lower()` to the cursor execution on the ID to make them match every time regardless of casing
- Was closing the connection after finishing sign up, which caused a connection bug when entering the user main menu. Had to remove it and make sure to close connection only at the end of main

User: (8 bugs)

- scrolling() breaks when the list to display is empty: fixed by checking the list size
- sessions newly created could not be accessed: fix with `conn.commit()` after creating a session
- conn was closed in some functions and caused future queries to raise an error: fixed by only close conn once in the main
- circular import between scrolling and searchArtist: resolved by giving searchArtist its own artistScrolling function
- cannot logout from secondary calls of `userMainMenu()`: fixed by bubbling up the return value
- Newly signed up users create their first session with None sno: fixed by replacing None with 0
- duration not shown when trying to see all song of artist
- if search result is empty, still print an empty list and is scrollable: fixed by checking size of search result

Artist (8):

- Forgot to Validating extra aids,
- crash when empty list of top users or playlists,
- Forgot to commit changes when adding new songs with `conn.commit()`,
- Forgot to make top artists and playlists display in one option,
- Make add song work for empty songs table,
- Stop the user from entering songs if aid conflicts with additional aids,
- Stop the user from entering non integer duration,
- Stop the user from entering empty song name,