Mini-Project-1 Design Document

Group: Dylan Du, Navdeep Singh, Ty Greve
CMPUT 291, A1
November 3, 2022

High Level Overview of the System:

At a high level, our program consists of two main while loops. The outer while loop contains the login screen which displays the login menu where the user can select whether they want to register or log in. If a user fails the registration or login process, for example by entering empty strings for their username and password, the outer while loop restarts and the user is brought back to the login menu to try again.

Once the user enters a correct/valid username and password to log in, they are prompted with the option to log in as a user, or as an artist if necessary. The program then continues to a 2nd while loop, either the while loop corresponding to the user menu, or the loop corresponding to the artist's menu. A high level representation can be found in the appendix[1]. Notice only one of the artist or user menus can be accessed after logging in. And when logging out, the program breaks out of the user/artist menu, and cycles back to the outer while loop, displaying the login menu once again.

User Guide:

Starting the Program:

To start, navigate to the directory in which the program is in. This directory must also contain the database file (initialized with the schema that was provided in this assignment) that you wish to use. Open the terminal and type 'python3 main.py xxxx.db', where the first command line argument after typing 'python3' is the UAtify program file, and the second is the database file (replace xxxx with the name of the database file that you would like to use).

Login or Create Account:

You will be prompted with the login screen with two options or you may type 'q' to quit the program. To create an account, type 1, and then you will be prompted to enter a username, your name, and a password. Alternatively, if you already have an account you may type 2, and follow the prompts to enter your username and password. Following the successful account creation or verification, you will be at the main menu.

Main Menu:

At the main menu you will be prompted with four options, as well as typing 'q' to quit the program or 'l' to logout. Type 1 to start a session, type 2 to search for songs or playlists, type 3 to search for artists, type 4 to end your current session if you have one active.

Searching for songs or playlists:

You will be prompted to enter keywords related to the songs or playlists you are looking for to carry out your search. You may enter no keywords, one keyword, or many keywords separated by commas. You will then see a list of songs and playlists related to your keyword(s). You may type 'n' to view the next page of search results, or type the number above the song or playlist information to view all of the songs in that playlist or view more information about a song. Also, typing 'q' will allow you to return to the

main menu. After selecting a song or playlist, you will be prompted with the options to listen to a song, get info about a song, or add a song to a playlist, Also, you may type 'b' to return to the search results.

Searching for artists:

You will be prompted to enter keywords related to the artists, or songs they have performed, to carry out your search. You may enter no keywords, one keyword, or many keywords separated by commas. You will then see a list of artists related to your keyword(s). You may type 'n' to view the next page of search results, or type the number above the artist information to view all of the songs performed by that artist. Also, you may type 'q' to return to the main menu.

Selecting an artists discography (all of their songs):

After selecting an artist from the artist search page, of which you want to view all of their songs, you will see the songs performed by that artist, and four options. To listen, see more info about a song, or add a song to a playlist, start by typing the number located above the song information in the list of songs, followed by a comma, and then one of the following 'listen', 'info', or 'playlist'. Additionally, you may type 'b' to return to the search results.

Adding a song to a playlist:

After selecting the song a list of existing playlist created by you will be displayed with the playlist ID and title. You may type the playlist ID to add this song to the playlist, or type 'new' to create a new playlist and add the song to it. If you choose to create a new playlist you will be prompted to enter a playlist title.

Primary Functions:

process_login(option):

This function processes user input in the login screen when the user tries to log in. The only parameter is option, which is a string, either "User/Artist login", or "Register User". Then based on the input, the function then asks the user to enter a username/password, and then calls upon database_functions.py to perform the login tasks. A depiction of how this function works can be found in the appendix[1].

song_actions(songs, uid):

This major function includes an interface, and input processing for a user where the function displays the options for song actions, such as listening to a song, obtaining information from a song, and adding a song to a playlist.

It interacts with the database in a similar way with process_login(), by calling functions from the database_functions.py which involves another level of abstraction. A schema would be similar to one shown in the appendix[1].

Other functions in the database_functions.py file serve as individual queries that do not involve a lot of complexity, which allows us to easily test and implement our queries into the main.py file.

Testing Strategy:

We structured our code so that there is an intermediary file, database_functions.py, which conceptually sits between our main python file and the database. This allows for individual query testing of the database_functions queries before we implement them into the main.py file. A conceptual schema of our file structures can be found in the appendix[2].

Our general strategy for testing our program is a two step process.

Firstly, we test our individual queries in the database_functions.py file, which contains functions such as end_session(uid, sno). To verify that our queries are correct, we created our own data set and were able to understand what the correct output should be. There are also many assert statements across our database_functions.py file that allow us to implicitly debug throughout the design process. For instance, when an artist is adding a song with the same title and duration, we made the design decision to completely reject it, which is consistent with the project specifications (Oct 23: Clarifications) as title and duration are together unique. And our assert statements in the insert_song() function allows us to ensure our program runs correctly.

Secondly, we then test our main.py file by interacting with our interface, and observing our queries that result from using the interface are the same as when we individually tested our queries. For instance, using the main.py interface to query top 3 fans of an artist, and then comparing the result to our individual query tests of the top3_users() function.

Some example scenarios tested:

- In our login screen, we ensured that the user does not input empty strings as username/password.
- We tested extraneous input that is 'incorrect'. The safe_input() function handles that.
- We also ensured that users or artists cannot insert songs that would break primary key integrity.
The add song to playlist menu does not have the go back or cancel option
- You are currently allowed to make a user with a username as the letter 'q'. This does not allow you to use the functionality of quitting the program while logging in or registering an account with any of those fields as the letter 'q'

Group Work Breakdown/Responsibilities

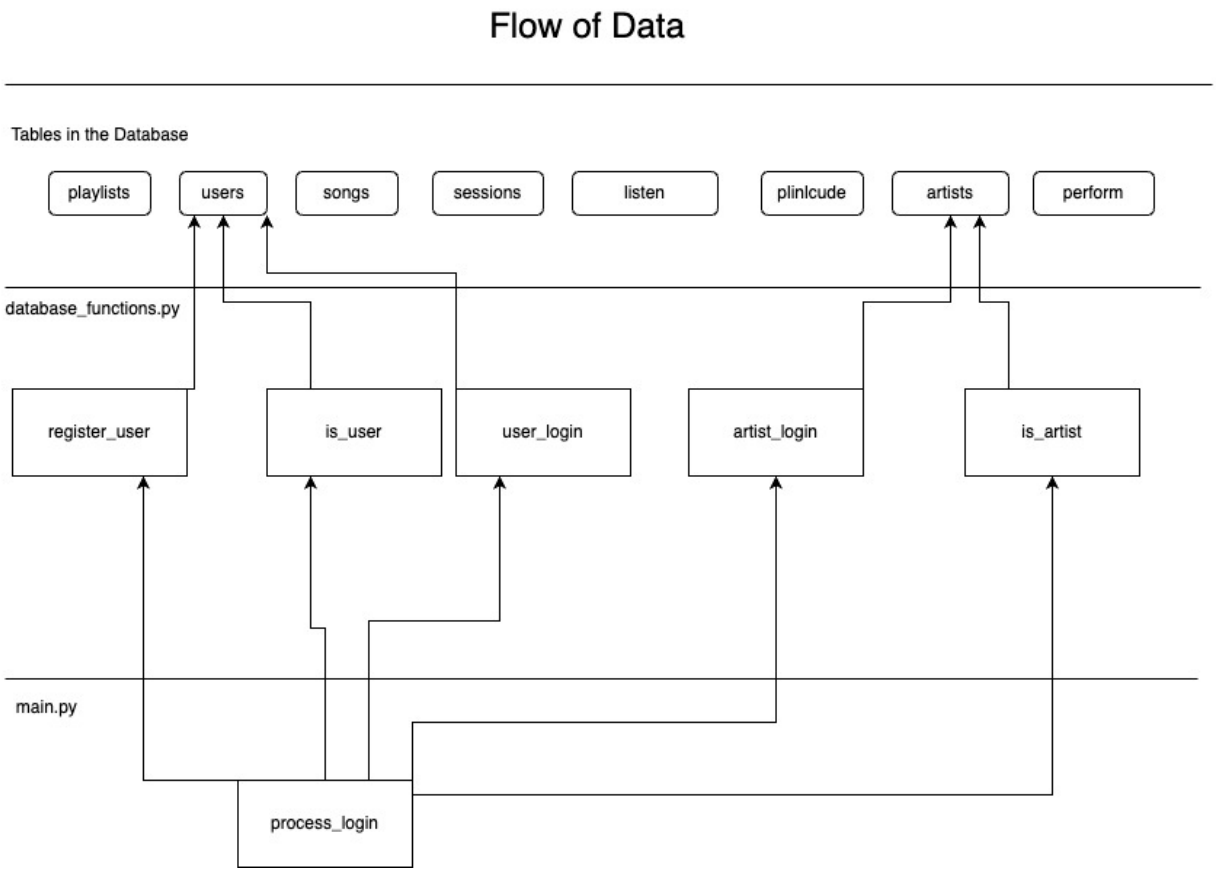
1. Dylan Du: 18 hours
    a. 5 hours → Full login screen implementation/Design
    b. 2 hours → login processing for user and artist
    c. 5 hours → Both Artist actions
        i. 1. Add a song
        ii. 2. Find top fans any playlists
    d. 2 hours → Interface for both user and artist
    e. 1 hour → Close all open sessions when exiting the program.
    f. 3 hours → Design Document except for user guide
1. Navdeep Singh: 11 hours
    a. 2 hours → adding the ability to start and end a session
    b. 3 hours → implementing user action 3 (using keywords to search for an artist)
    c. 3 hours → implementing user action 2 (using keywords to search for a song or playlist)
    d. 3 hours → implementing layout for song actions (how everything is displayed), as well as song action 2 (retrieving information about a song)
2. Ty Greve: 8 hours
    a. 2 hours → implementing song action 3

b. 2 hours → worked on managing current session and closing active sessions upon quitting the program
c. 2 hours → wrote the user guide for the report, tested edge cases for all input options, searched for bugs in the code
d. 2 hours → implementing song action 1

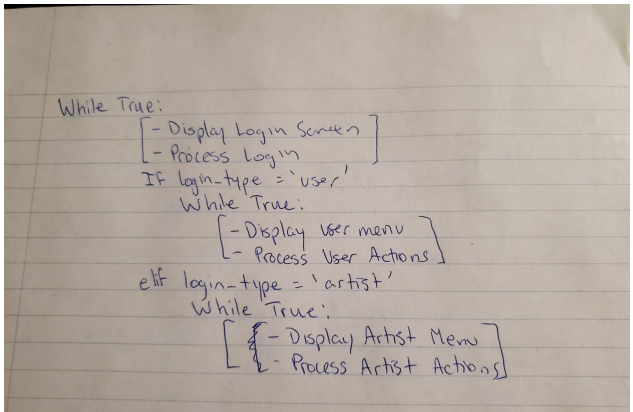
Method of coordination:

Mostly communication through text, updates in person when we have labs/classes together. As well as communication through GitHub pull requests and clear documentation.

Appendix



(2) This is an illustration of the flow of data in our program from a particular function instance, process_login(), called in the main.py file. A more in-depth explanation can be found under the testing strategy section.



(1) Here is a high level representation of our code. The outer while loop displays the Login Screen, and the inner while loop represents the user menu and artist menu, respectively.