a)Overview:

This program is designed to provide music streaming services to both normal users and artists. Users should be able to perform the four major actions: Start session, search for songs and playlist, search for artists, and end sessions. When a song is selected, the user can choose to either listen to it, display information, or add it to his/her playlist. If an artist has a user account as well, he/she will be able to choose to login as artist or user. If login as an artist, he/she should be able to choose to add songs or find his/her three most popular playlist as well as the three top fans of him/her.

Small guide to start the program:

To start the program, simply pass the file name as a command line argument. For example, run it as : python3 mini_project1.py mini.db

You will then be asked to choose to login or register. If you choose to login, simply input your id and passwords to login as user/artist. If you have both accounts, you will be prompted to choose to login as user/artist. To register as a user, you should input a username that is within 4 characters, as well as the password. After a successful registration, you will directly enter the main user login surface. After that, follow the instructions shown on the screen, to input the corresponding keywords to choose services you want.

b)Detailed design of major functions:

connect()

Enable us to connect databases through python. It also enforces foreign keys and uses sys.argv to allow user pass filename as command line argument..

main_screen()

This portion keeps guiding the user/artist to input an id and password. First time users can choose to register as a new user. It also allows people to exit the program. When exit, every existing session will be closed.

login()

This part checks whether inputted id and password match one of the user or artist. If it does not match any of them, the user/artist will be asked to input again. If one has the same id in both artists and users tables, will be asked to choose to login as one of them.

register()

To enable people to register as new users. It prompts people to enter a valid 4 characters id and password. ID can be case insensitive. When a new account is created, will display create message.

user_login(uid)

When a user successfully logs in, will check if they have an unclosed session. If that is the case, update the global value s_num, which is the session number. Then users will be prompted to choose their four actions or logout. If they logout without closing their session, store their id and session number in two separate lists.

start_session(uid)

To open a session for users to allow them to listen to music later. If there is no existing session, create one for the user, session number based on the largest session number of the past session number and add one. If there is no session before, the new session number will be one. If there is a running session, the user will not be able to open the session.

search_SandP(uid)

This allows users to search for songs or playlists that match to their keywords. Each keyword will be matched at most one. The result will be sorted based on the matching time of their keywords. For each result, id, title,duration will be shown, and five of them in one page. Users can also choose any results shown in the screen or press x to exit, 0 to the next page. If a song is selected, the user can perform a song action, which will be discussed later. If a playlist is selected, all songs in this playlist will be shown, users can choose them to perform song actions.

search_artist(uid)

This allows the user to search artists based on their inputted keywords. The results will be shown based on the most matching keywords. Results will be displayed at most 5 in each page, users can choose any of these five or press 0 to the next page, x to exit. When an artist was selected, any songs performed by him/her will be displayed, and users can now choose any of those songs to perform song actions.

end_session(uid)

Users should be able to end their session by themselves. If there is no running session, reject it. If a session was successfully closed, a message will be displayed to notify the user.

song_select(uid,sid,title)

Prompt user input as user's action, user can choose if they want to listen the song

or see more information about the song or add the song to playlist

listen_query(uid:str, sid:int, title)

When a song is selected for listening, a listening event is recorded within the current session of the user (if a session has already started for the user) or within a new session (if not). A listening event is recorded by either inserting a row to table listen or increasing the listen count in this table by 1.

get_more_info(sid:str)

Get more information about the song. More information for a song is the names of artists who performed it in addition to id, title and duration of the song as well as the names of playlists the song is in (if any).

add_to_pl_ui(sid, uid)

adding a song to a playlist, showing what playlists the user has. The song can be added to an existing playlist owned by the user (if any) or to a new playlist. When it is added to a new playlist, a new playlist would be created with a unique new id and the uid set to the id of the user and a title will be obtained from input.

artist_login(aid)

When an artist successfully logs in, he/she should be able to choose to either add a song or find_top, which will be discussed later.

add_song(aid)

It prompts the artist to enter a title and duration of a song. If it matches any songs within the database, artists will be asked whether they want to add this song as a new song. To implement a new song, a new song id will be added based on the largest existing song id and plus one. If there are no songs, create a song id as one.

find_top(aid)

Find top fans and playlists. The artist would list the top 3 users who listen to their songs the longest time and the top 3 playlists that include the largest number of their songs. If there are less than 3 such users or playlists, fewer number of users and playlists can be returned.

c)Testing strategies:

For each major session, test them normally first, and check the database to see if they match them. Then try to do some action that will not be normal, including but not limited to:

Ensure users can logout and exit the program. Try to login with invalid id and passwords. Try to login, search without any input. Try to start/end a session while there is running/no session. Try to logout without closing the session, then login as another user and then logout again to login the previous account to see if the session is still running. Try to press 0 while the page reaches the end when displaying results.

(d) your group work break-down strategy.:

At the end, the code was divided into two parts.

Login screen, User section (Start session search for songs and playlists, search for artist, and end the session) for JiahaoYu (ccid: jiahao20).

Song actions (listen to song, find more information, add song to playlist) and artists action (add a song, find top fans and playlist) for Yile Ma(ccid: yile).

Aksshat discussed with us at the beginning of the mini-project release.

Aksshat Khanna (ccid:aksshat) was supposed to finish the artist action part, but he failed to work on it. JiahaoYu have been trying to use emails, googledoc, whatsapp to reach other members. Only Yile Ma(ccid:yile) responded and updated the code. The other group member Aksshat Khanna(ccid:aksshat) did not update anything during the last couple days. Instead, he tried to contact us 6 hours before the deadline but still with no update on the code. Since the deadline is coming, we have to work on the code with the rest of us. Hence he (Aksshat Khanna) did not work on this code eventually. What he did was that he had few discussions with us. Chat histories stored on the phone.