CMPUT 291– MINI PROJECT 1 REPORT

Kunaal Gupta, Aidan Gironella and Shubhkarman Jaura

1. ***Overview of our system with a small user guide***

The system we have created acts like a streaming service referred to as UAtify (mentioned in Assignment 1) that keeps information about songs, artists, users and playlists. This service allows the users to log-in or sign-up, start and end a music session, search for songs, playlists, and artists, listen to songs, add them to playlists, and gain more information about songs. The system also allows artists to add a song to the system and find their top fans and playlists.

Upon running our application, the user will be prompted to enter the full name of a .db file to interact with. Next, the users will be presented with a login screen. They can provide correct credentials to log in either as a user or artist or sign-up as a new user.

If logged-in a user, the “user” main menu will open where the users can create a listening session (updates the session table), search for songs and playlists (interacts with the songs and playlists table in the database), take action on the songs (updates the playlist, listen and plinclude tables), search for artists and close a session (updates the session table) as well. In the song/playlist search function and song action functions, the user can input “/exit” at any time to return to the main menu (may need to use “/exit” twice, depending on where the user is in the function).

If logged-in as an artist, the “artist” main menu will open where they can add a new song (updates the songs and perform tables in the database) and find a list of their top fans and playlists (interacts with the songs, listen and perform tables).

2. ***A detailed design of our software with a focus on the components required to deliver the major functions of our application***

Primarily our application has three main components:

1. User/Artist Authentication
2. User Session
3. Artist Session

User/Artist Authentication:

a) Main function: The main function of the code is responsible for implementing the log-in screen. On the log-in screen the user or artist can log into the system given that they provide a correct user/artist id and matching password. Our system uses the information provided by

the user and matches it with the information stored in the data-base to give right access to the users and artists.

b) Sign-up function: The system also allows the unregistered users to sign up by inserting new users' information into the users table in our database.

User Session: The user session is composed of several crucial functions relevant to the successful functioning of our application. It includes the following major function:

a) Start Session: The function allows the user to be able to start a session. For each session, a session number unique for the user is assigned by our system and it updates the session table in our database to store the new session created by the user with the current date as the start date.

b) Search for Songs and Playlists: This function takes the input of however many keywords the user would like, and returns all songs and playlists with any of the keywords in the title, sorted by the number of keyword matches. 5 results are displayed at a time, allowing the user to type "more" to see the next 5 results, or to pick a result number to get more actions. If a song is selected, a new screen will appear, allowing the user to pick a song action as outlined in the next paragraph. If a playlist is selected from the original results list, the program will list all songs in that playlist. This list of songs can also be selected. If at any point the user wishes to leave this functionality, they may simply type "/exit" and the program will return to the main menu.

c) Song Actions: There are three available song actions: listen to the song (only if the user has an active session), get more information about the song, and add the song to a playlist. Within this third action is an option allowing the user to create new playlists. It will also not let you add the same song to the same playlist twice.

d) Search for Artists: The function uses UserId (as a parameter) to request one or more unique keywords from the user in order to retrieve the details of all artists from the connected database who have the matching keywords in their name or the title of their song(s) in descending order of the matching keywords. It then shows 5 rows at a

time based on the user's response, whether he wants to see more matching artists or look into a specific artist. Along with the artist's information, the function returns a complete list of all the songs performed by that artist which is retrieved by search_song function. The user can select a song from this list and perform a song action, as discussed further below.

e) End a Session: Once the user is done using the user, they have the ability to end a session or log-out and any of their opened sessions will automatically be closed. Users can also terminate the whole application if they want.

Artist Session: The artists session is composed of two main function:

a) Adding a song: When the artists log into our system they have the privilege to add a song if it does not already exist in our database for that particular artist with the provided song title and duration. The artists must provide the list of any additional artists for who have performed with them.

b) Finding Top Fans and Playlists: This function takes the artist's ID as a parameter and processes It to display the data of top three users who listen to his/her songs the longest time as well as the data of top three playlists that include the most of his/her songs.

3. Testing Strategy

Throughout the completion of the program, special attention has been given to correct execution of the code and handling the edge/unexpected scenarios. Thus, each group-member was primarily responsible for ensuring the proper execution of their part. For the song/playlist searching the program was rigorously tested with various different approaches. The requirement to sort the results by the number of keyword matches greatly complicated our task, and ultimately we settled on a design that constructs an SQL query as a string and then executes the query, allowing all of our data processing to take place in SQL instead of Python. To test this, we gave the system multiple different combinations of keywords and cross-referenced the program's results with what we expected from going through our data tables by hand. Once this was working properly, we had to test both selecting songs and playlists. To test the song selection/actions, we tried inputting different selection numbers that we knew would not work, such as 0, a negative number, or a number higher than our results count. Knowing that the selection worked, we could then test our

three song actions. The main things to test here were if we could listen to a song without an active session, by attempting to listen to a song both with and without an active session, and if we could add the same song to a playlist twice. To test the playlist functionality (listing all songs in a selected playlist), we simply queried our database in SQLite since it was a relatively straight-forward query and ensured the results were the same.

4. ***Group Work Break-down Strategy***

We are three members in a group, therefore, we decided to break down the project into sub-parts. To keep the project on track, we actively communicated with each other on the WhatsApp group chat and ensured that expectations and responsibilities were set crystal clear. Moreover, as a team, we had group meetings where we clarified our doubts, seeked help when required and offered a helping hand to other group members to ensure the successful completion of the project.

Shubhkarman Jaura: was responsible for the User-Authentication, Start and end session along with adding the song functionality for the artists. He has also completed the majority of the project report as well. He has spent an estimate of 23 hours on this project.

Aidan Gironella: was responsible for implementing the search for songs/playlists functionality, search_songs_playlists(), as well as the song actions, song_action(). Also created the GitHub repository for the project. About 23 hours spent on this project.

Kunaal Gupta: was responsible for implementing the search for artists functionality along with finding top fans and playlists for the artists. Devoted about 22 hours to think & code for this project.