(a) a general overview of your system with a small user guide,

The general overview of the system gives a high-level introduction and may include a diagram showing the flow of data between different components; this can be useful for both users and developers of your application.

This is a command line app that allows user to login to perform different activities depends on the user login type. There are two types of accounts user account and artist account.

In the Login module, user first is prompted with a log in screen. User may choose to log in or create a new account by entering the option index into command line. When log-in option is chosen, user will be prompt to entering username and password. If the login is successful, the user will be directed to User module or Artist module based on the account type. If the login matches both account types, the user will be prompt to choose which type of account to log into.

If the user dose not have an existing account, the user can choose to create a new User account in the log in module. Note: Artist type account can not be created by user.

In the User module, user have 3 main activities: start session, search for songs and playlists, and search for artists.

Start session will start a new session which will end upon the user exiting User module.

Search for songs and playlists takes space separated keywords as command line input. A list of songs and playlists matching the keywords will be shown with the row with higher matching count on top of the list. The list is displayed in a downward paginated format with maximum 5 rows per page. The user can then select a row in the list. If the selected row contains a song, the user will be prompt to perform a SongAction(see below for detail) on the song. If the selected row contains a playlist, a list of songs in that playlist will be display. Selecting one of the playlist song also leads to a SongAction.

Search for Artists is similar to Search for songs and playlists. It takes space separated keywords in command line. Artists who’s name or song titles contains any of those keywords will be display in the same downward paginated format with higher matching count rows on top. Selecting an artist will display all the songs the artist performs. Selecting a song will also leads to SongAction.

SongAction: when a song is selected, the user can perform any of these three actions:

1. Listen to it: the song will be added to the current session. If there is none, an new session is started.

2. See more info: returns more details about the selected song.

3. Add to playlist: user can choose to add the song to an existing playlist or add it into a newly created playlist

In the Artist module, the user has 2 main activities: add a song and find top fans.

Add a song will add a song into database with song title, duration, and performer(s) given.

Find top fans will return top 3 users who listen to artist's songs the longest time, and top 3 playlists that include the largest number of their songs

(b)

We had 5 main modules for this system: userActivity, login, artistActivity, Main, and sql\_commands.

login controls all the log in activities. It manages user log in and register user new account, and returns the user type, user id and if user wants to exit the program or not.

artistActivity module is used to manage all user activities for the logged in user.

userActivity module is used to manage all artist activities for the logged in artiest.

Sql\_commands contain all the sql queries used in all other models.

Main is the part where we connect all modules together.

Here is our designed pseudo code for main.py:

While true:

(uId, isArtist)= login();

if(isArtist) {

artist(uId)

} else {

user(uId)

}

(c)

We used failure testing and error guessing strategy and try to achieve branch and statement coverage while testing. Unit testing was performed throughout the development process. We had 4 end-to-end use cases for testing.

Scenario 1: log in as a user and complete all user activities

Scenario 2: log in as an artist and complete all artist actives

Scenario 3: register as a new user and complete all user activities.

In each scenario, failure testing and error guessing is performed by entering invalid inputs and testing edge cases.

Test cases were able to cover all statements and branches. We discover 3 bugs during final testing. The nature of all the bugs were the lack of input constraints on the user.

(d) your group work break-down strategy.

Split of tasks:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tasks** | **Assignee** | **Progress** | **Time Spent** |
| Log in module | Qi Zhou | Done | 4 hrs |
| User Activity module | Tianyuan Fang | Done | 12hrs |
| Artist Activity module | Qi Zhou | Done | 1hr |
| System design | All group members | Done | 1.3hrs |
| Documentation | All group members | Done | 30min |
| Database commands | All group members | Done | 2hrs |

Both of coordination:

Use discord to communicate and hold design meetings. Push code to GitHub for version control.