MusiQuest

Rajat Rathi	160050015
Abhro Bhuniya	160050017
Tummidi Nikhil	160050096
Gurparkash Singh	160050112

OVERVIEW

MusiQuest will be a Flutter application, which primarily serves as a general purpose application to listen to / search for music. The app will provide a lot of user customisation so that the user can filter and play music easily and efficiently. *MusiQuest* will also contain a recommendation system, storing user-specific information and thus allowing the user to explore new music which will suit their taste.

FUNCTIONAL SPECIFICATIONS

- Users will be able to Login (new users can register) into the app.
- The user will be displayed lists of most popular (most viewed/searched), most liked, and most recent songs (common for all users).
- The user can play songs via an embedded YouTube window (our app will store every song's YouTube URL to stream), and the app will also have the provision to show lyrics along with the song being played (if available).
- The user can search over song name, part of lyrics, album or artist name, as well as genre while accounting for minor spelling mistakes.
- The user can create custom playlists which will persist unless deleted by the user (unlike queues, which expire after a session).
- The user can select a few songs and make them into a temporary queue to loop through them without going through the trouble to choose every next song. They will have an option to save this queue as a permanent playlist. These queues will expire after each session, if not saved as a playlist.
- The user can like/dislike songs, based on which we will recommend songs to them and which will be tallied for all users to display the most popular and most liked songs in user home.
- The user will also have a personal profile page, and also the option to deactivate/delete their account.

IF TIME PERMITS...

Apart from the above mentioned features that we plan to surely implement, there are certain features that we will look to implement if we have some time on our side. Following is the account of such points:

- ➔ **Voice Search:** There are some libraries that can be used to match audio and return results. We might look to work with these libraries so that a user can give a recording of a small part of an song and we can return the matching results.
- ➔ **Privacy of user playlists:** There could be an option to the users to make their playlist public, that is, viewable but unmalleable to others. Also, we can store 'priority' for each song to allow the user to reorder songs in the playlist when playing in cyclic order (as compared to the usual case where we sort by 'timestamp' when the song was added to the playlist).
- ➔ **Discussion Forum / Reviews:** This will enable users to add comments and reply to comments on Songs, Albums and Artists and review the song.

FUTURE SCOPE

There are several features that can be incorporated into our App, but we have decided to exclude them either because of the time crunch or because of lack of available data. Listed below is an account of such features along with the way to add them if required.

- ➔ **Information Window:** Most modern Music websites and Apps provide information and other trivia about various Songs, Artists and Albums, however, it would be nearly impossible for a small team to gather and maintain such a database. However, logistically speaking, provided such data, we can easily incorporate it into our Relational Schema.
- ➔ **Profanity Filtering:** Most discussion columns have automatic detection and removal of explicit language / racial comments. This can be added using Machine Learning.

TYPES OF USERS

- ➔ **Privileged Users (Admins):** Admins will be allowed to add and remove songs to and from the database. Admins can also remove (malicious) users, and under our current design, we will have a predefined set of admins added to the system in advance.
- ➔ **Normal Users:** The user does not have the rights to modify the song database. All functionalities described above will be available to the user.

INTERFACE SPECIFICATION

Login/Register Page

Will allow a new user to login/register or an admin to login with a username and password. On successful login, the user will be directed to their Home Page, or the admin will be directed to the Maintenance Page. This will be visible to both the user and the admin.

User Side:

Home Page

The Home Page will contain lists of the trending, most popular, most recently played songs, albums and artists. Options will be available to search and to access personal library.

My Library

My Library has options for creating and viewing playlists and favourite songs and artists.
It also contains recommendations specific for users.

Search Screen

Will allow the user to search for songs, albums, artists, lyrics and genre while accounting for minor spelling mistakes and displaying results based on popularity.

Profile Page

Will allow the user to modify username and password and give provision to delete the account.

Playlist/Album Screen

Will contain a list of songs which are part of a particular playlist/album. Playlists owned by the user will have the provision to add/remove songs.

Queue Screen

This screen displays the temporary queue of songs that the user is listening to, with options to add songs to queue and save the queue as a user playlist.

Song Screen

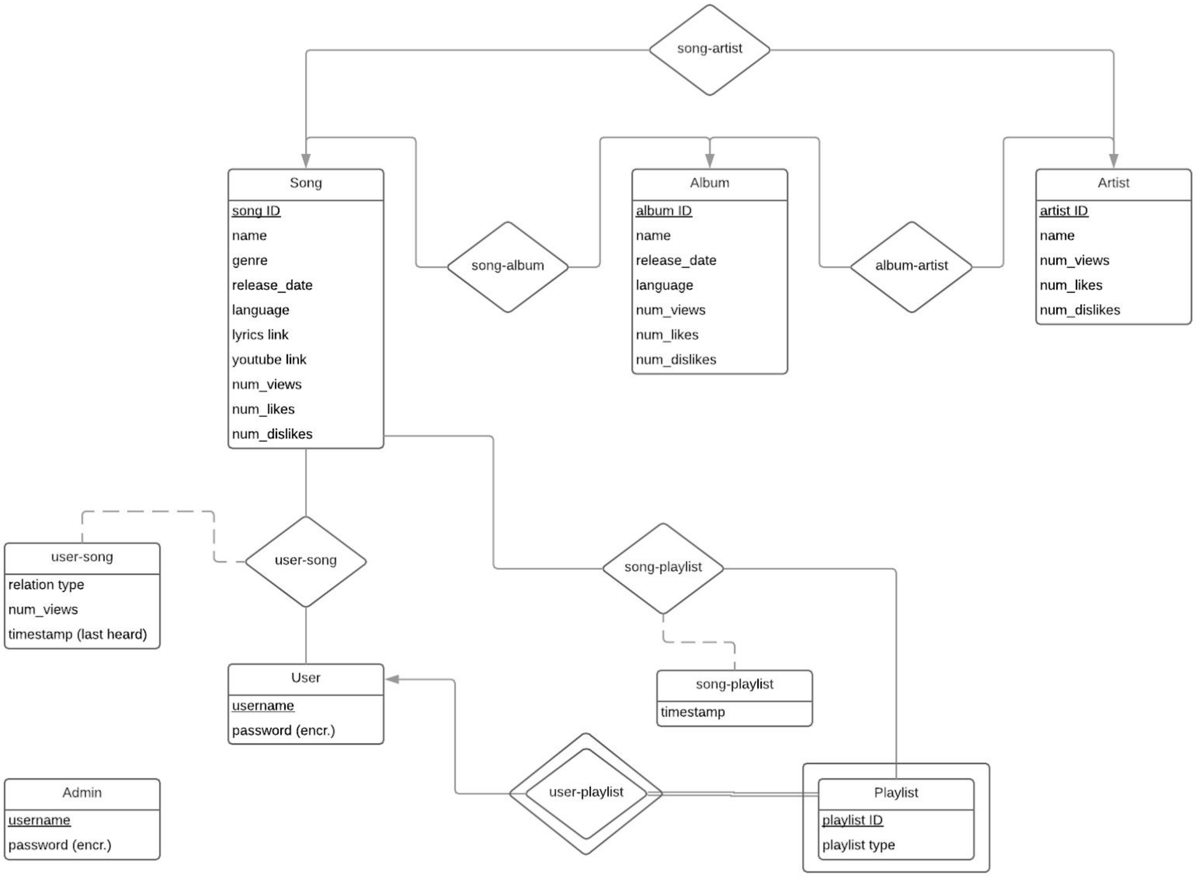
This screen will display song details (name, artist, etc.) and play the song via its YouTube URL.

Admin Side:

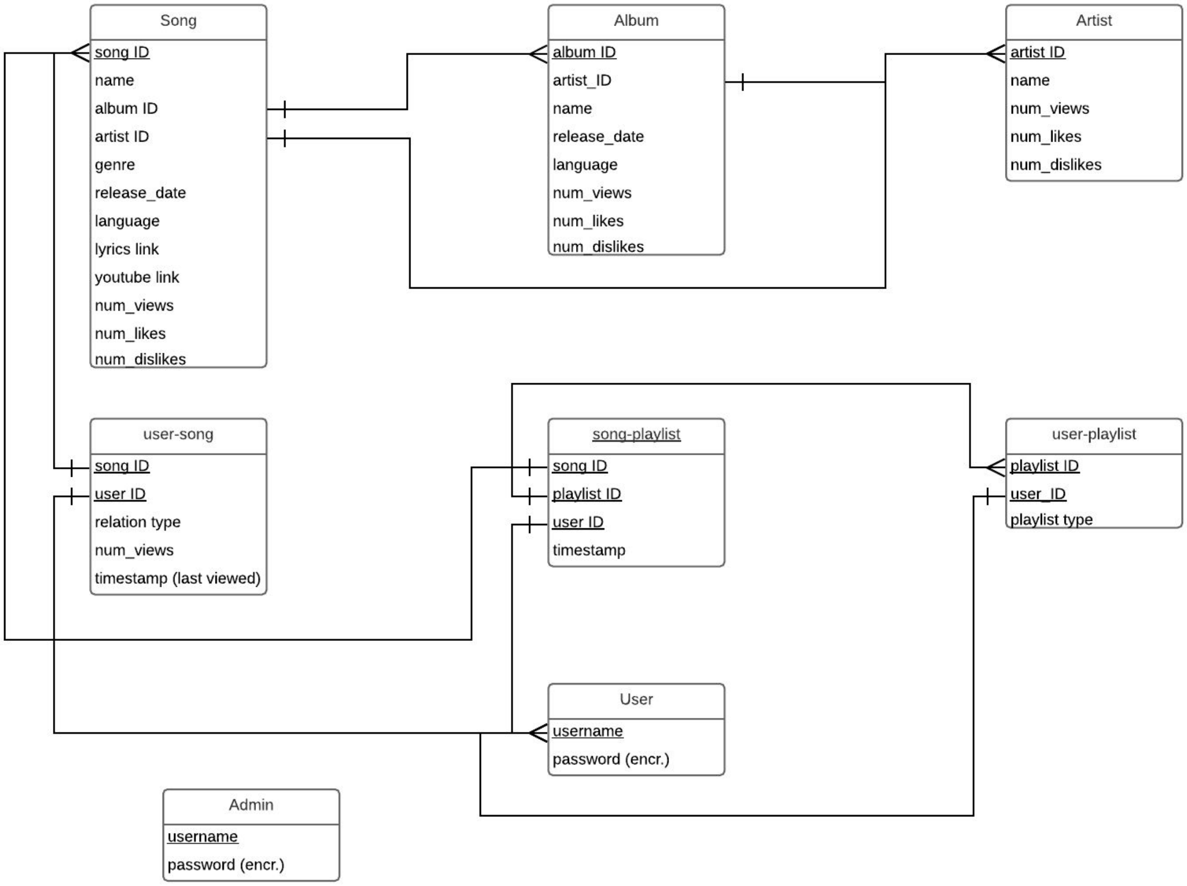
Maintenance Page

Will have form functionality to add/ delete songs, albums or artists from the backend to the application. Will also have a search box with autocomplete to search for specific users (maybe malicious) to remove them.

ER DIAGRAM



SCHEMA (TABLE DESIGN)



PLANS FOR TESTING

We have already looked up datasets of more than 75000 songs along with relevant tags and would use these songs in the database for the app. We will create multiple users and test the various features like if the most listened to and most liked songs are shown appropriately to the users, and more importantly if the recommendation system works satisfactorily taking into account which songs the users listened to and liked.