

## Project Milestone #2

Group 101-4

Team Name: CAMIX

Members: Austin Cha, Cam Fields, Isha Karki, Xiyuan Liu, Max Van Sickle

### Project Features list

\*\*\*INSERT BULLET POINTS OF MAJOR FEATURES, include brief description per bullet

- **Queuing music in a priority queue** - this is determined by the people connected
- **Generate temporary sessions using codes** - The application can produce group codes/ QR codes for multiple users to join a protected group session which is terminated after use.
- **Security** - The application asks the users to login in order to receive all functions of the application. All the login information will be encoding and saved in the server.
- **Saving and updating the music resources into the database(API)** - the front end of the application will be connected with the music resources in the database. We are starting out with a static database with a limited song pool in order to get a working music sharing application and then we could try to piggyback off of another services API to get a larger song pool.
- **Personal profiles/networking** - Users' unique logins can allow them to create a profile with which they can connect with their friends.
- **Personal preferences and genre grouping** - Gives the users a more personalized experience. Allows the host to set restrictions like repeating songs, genre/type of music/environment, and control who's in the group/session.

### Requirements

#### 1) Queuing

Non Functional:

- Able to queue at least 100 songs.
- The queue will automatically update after each vote in no more than 1 second.
- The voting system will be very intuitive with either a swipe system or obvious icons indicating up-voting or down-voting.
- Each member of the group will be able to participate in the voting system as well as adding new songs to the queue
- Both the queue and the voting system will be accessible in the same screen

Functional:

- Each user will have access to many songs and add any song to the queue.
- If there is a song already in the queue it cannot be added again (no spamming songs)
- Once the song is in the queue, the users will be able to up-vote or down-vote different songs which will produce the dynamic queuing system. The more up-votes, the higher in the queue.

## **2) Temporary Sessions**

Non-Functional:

- Light version requirements: up to 16 users in one session (subject to change).
- Industry version requirements: up to 1,000 users in one session (subject to change).
- No expected delay, or 'lagging' as the user count gets higher.
- The time it takes to join a session must be no more than 2 seconds.
- You will not be able to join multiple sessions at once.

Functional:

- Every member of each session will have the same level of authority/ access with respect to the scope of the group.

Except for these specifics:

- The host of the temporary session will have admin power to kick anyone from the session.
- Skipping and de-queuing songs is to be determined.

## **3) Security**

Non functional:

- Having a hash table to store all the usernames and the passwords
- Using an encryption algorithm to improve the level of security. (Data Encryption Standard, RSA...)
- Validate Input Data before storing it to the server.

Functional:

- Visitor can access the application with limited function.
- User will have an interface to log in their account with password.

## **4) Connect to database/music service API with song pool**

Non functional:

- Seamless connection between local and server communications.
- At least 50 songs will be in the initial database.
- Users will not have direct access to the database, only through searching/ exploring for songs and/or adding them to the queue.

Functional:

- Hosting a database to house all .mp3 files / subscribing to amazon data hosting service.

## **5) Personal profiles/networking**

Non functional:

- Having some type of username/unique ID to be able to add friends and connect
- Customization of personal profiles (i.e. profile picture, display name, etc)

Functional:

- Possibly a graph data structure to connect profiles together as friends

## **6) Preferences and genres**

Non functional:

- Using linked lists to group similar songs together
- People will be able to view who's in the current group and host will be able to kick users out of the group
- Settings will include autoplay of suggested songs if queue is played through or to loop back to original queue. Also, a device list for bluetooth devices
- Anonymous mode allows user to add songs to the queue without showing username

Functional:

- App will be able to recommend songs based on what's currently in the queue and if the host sets a theme

## Project Plan

This gantt chart is subject to change as milestone requirements are released and target dates for phases are either over or underestimated.

