MusiQueue

Team 19 - Sprint 1 Planning Document

Vincent Maggioli, Samuel Kuhns, John Stimac, Sairam Galla, Brian Hanford, David McGary

Sprint Overview

During this sprint, we hope to get or backend server and database set up. We want the communications from Android to server to database to work smoothly. We will also create a very basic frontend to communicate with the server to test the interactions. We will also implement the ability to search and queue up youtube videos taken from the youtube API.

Scrum Master: Vincent Maggioli

Meeting Plan: Monday/Friday during CS 307 hours (11:30-12:20), First week we will meet at 5:30 instead.

Risks and Challenges: This sprint is laying the groundwork for the other sprints. Should we have any serious errors at the end of this sprint it will drastically slow progress in the following sprints. Also, our UI is still somewhat clunky, and we aren't optimizing it this sprint. This is banking on our backend working well enough initially to have time to work on our prototype of the Android application.

Current Sprint Detail

User Story 1

As a user, I would like to be able to queue youtube videos as songs.

Task	Time	Owner
Create activity to play YouTube videos	5 hours	Vincent Maggioli
Create connection to YouTube Android API	5 hours	Vincent Maggioli
Set up home queue screen to mimic design sketches	5 hours	Vincent Maggioli
Create activity for users after they have connected to a hub	5 hours	Vincent Maggioli

Acceptance Criteria:

- Given that our app implements the YouTube API, the user should be able to search for a song via keywords in the app.
- Given that our app implements the YouTube API, the user will be presented with a list of YouTube videos that match their search.
- Given that our app implements the YouTube API, the user will be able to enqueue videos via a link or a search.

User Story 2

As a user, I would like to be able to search for a song by name, see results, then add one to the queue.

Task	Time	Owner
Implement ability to search for YouTube videos in app	5 hours	Brian Harford
Implement ability to play videos after selecting a video from a search	5 hours	Vincent Maggioli

Acceptance Criteria

- Given that the search function is implemented, when a user searches for a YouTube video, they will be presented with a list of videos that match their search.
- Given that the selection for videos feature has been implemented, when a user selects a video after searching for it, the video will be enqueued and the user will be taken back to the queue screen.
- Given that the check for duplicate videos is in place, upon attempting to enqueue a video that is already in the queue, the user will get a message saying that the video is already in the queue, and the video will not be added.

User Story 3

As a user, I would like to see the songs currently in the queue in priority order.

Task	Time	Owner
Learn Android Studio	10 hours	Sam Kuhns
Reorder queue with each new element	10 hours	Sam Kuhns
Display queue in proper order with scrolling list	10 hours	Sam Kuhns

Acceptance Criteria:

- Given that the songs are added and voting is implemented, each song in the database should have an updated vote score
- Given that the songs are added and voting is implemented, the display of the songs should change based on the current score for each.
- Given that the songs are added and voting is implemented, the order of the songs displayed, should reflect the vote score given to each.

User Story 4

As a user, I would like to vote up (or down) songs to raise (or lower) them in the queue.

Task	Time	Owner
Implement the button click to increment number of upvotes in the database.	5 hours	John Stimac
Implement algorithm for ordering songs based on number of upvotes, downvotes, and time added	10 hours	John Stimac
Implement logic to prevent a user from voting twice on a song.	5 hours	John Stimac

Acceptance Criteria:

- Given that the front end button and the back end are implemented, the score for each video should update after a vote is submitted.

User Story 5

The application will connect to a server to implement the queue.

Task	Time	Owner
Learn how to do server interactions with app and database	10 hours	Brian Hanford
Set up a server with AWS	2 hours	Brian Hanford
Make sure all administrators are added to users who are allowed to access the server without allowing unwanted users.	3 hours	Brian Hanford
Set up communication between the app and the server	7 hours	Brian Hanford
Set up communication between the database and the server	5 hours	Brian Hanford

Acceptance Criteria:

- Given that the server is setup on AWS, all users should be able to access the AWS server page and update/manipulate it.
- Given that communication between the app and the server is correctly setup, we should be able to send http requests to the server and receive data back from the server for updating information on the app.
- Given that communication between the database and the server is correctly setup, the database should be able to receive sql statements from the server and send back result sets for the server to arrange and distribute as necessary.

User Story 6

As an administrator, I would like to store and view all active hubs with their queues on a server.

Task	Time	Owner
Every hub that is created by a user is added to the server.	10 hours	Sairam Galla
Learn database/server/Android interactions	10 hours	Sairam Galla
Learn Android Studio	10 hours	Sairam Galla

Acceptance Criteria:

- Given that a hub is created by a user, it should be added to the server.
- Given that a hub is active, it should be viewable by the administrator.

User Story 7

As a user, I would like to connect to a hub by the hub name and password.

Task	**Time**	**Owner**
Implement the ability to enter a hub name and password	6 hours	David McGary
Check whether hub exists	5 hours	David McGary
Check whether password is valid	4 hours	David McGary
Have a lock out feature after too many attempts	5 hours	David McGary
Learning Android Studio and resolving issues with Git	10 hours	David McGary

Acceptance Criteria:

- There should be an activity that shows after selecting “Connect to an existing hub”
- There should be clear error checking notifications

Functional Requirements

- As an administrator, I would like to have hubs auto delete after a few days of inactivity
- As an administrator, I would like to store and view all active hubs with their queues on a server.
- As an administrator, I would like to identify individual hubs by phone IDs.
- ~~As a user, I would like to connect to a hub by the hub name and password.~~
- As a user, I would like to find a nearby hub based on my current location, time permitting.
- ~~As a user, I would like to be able to sign in to MusiQueue securely (automatically with phone pin).~~
- As a user, I would like to have an option to reconnect to a recent hub without having to type the name and password again.
- ~~As a user, I would like to type in my name when connecting to a hub.~~
- As a user, I would like to see the songs currently in the queue in priority order.
- As a user, I would like to have the queue reorder as songs get voted on.
- As a user, I would like to have songs leave the queue after they get played.
- As a user, I would like to see who put each song in the hub.
- As a user, I would like to see how many votes each song in the queue has, time permitting.
- ~~As a user, I would like to not allow duplicates in the queue.~~
- ~~As a user, I would like to be able to queue youtube videos as songs.~~
- ~~As a user, I would like to be able to search for a song by name, see results, then add one to the queue.~~
- ~~As a user, I would like to be able to add youtube videos by their url.~~