MusiQueue

Product Backlog

Coordinator: Ramya Vulimiri

Brian Hanford, Vincent Maggioli, Neil Amin,
Sairam Galla, Samuel Kuhns

Problem Statement

Typically for a bluetooth connection to a speaker, in order to play your own music you must first kick the person who is currently on the speaker off of it and then connect yourself to the speaker. Our application will allow users to connect to a queue, playing on one person's device (a.k.a. The hub) and enter music into a queue in the hub. The music will then play from the speaker after the song that was currently playing in real time, removing each song as it is played. Users are able to vote on and interact with the hub.

Background Information

The members of our team have all experienced the problem of deciding who plays music in any sort of social gathering. Whether it be in your apartment, or at a larger gathering everyone has their own taste in music. Currently, there is no application that perfectly addresses this issue and allows for every user's music to be represented. In CS 307, we addressed this problem with an android app, but we were not satisfied. First of all, IOS users are still completely alienated from our app. Secondly, because we use YouTube, the hub owner's phone screen must stay unlocked to allow music to continually play. We address this by implementing our application on web. This will mostly benefit hub owners, but may prove to be convenient for general users as well.

Environment

We will use HTML for the static pages and Angular 2 for the user interactions on the front end. We are using Firebase for our backend and database, so we will be storing users, the playlists and number of votes for each song from the hubs on Firebase.

Requirements

Functional

Backlog ID	Functional Requirements	Hours	Status
1	As a user, I would like to be able to sign in to MusiQueue using my Google account	4	Completed in Sprint 1
2	As a user, I would like to change my username	2	Completed in Sprint 1
3	As a user, I would like to connect to a hub by searching	6	Incomplete: moved to Sprint 2
4	As a user, I would like to add songs to the playlist in a hub	4	Completed in Sprint 1
5	As a user, I would like to vote on songs in a hub	4	Completed in Sprint 1
6	As a user, I would like for songs to re order based on their number of votes	8	Completed in Sprint 1
7	As a user, I would like to give my hub a name and protect it with a 4 digit pin password	4	Completed in Sprint 1
8	As a hub administrator, I would like to be able to remove songs from the hub playlist	4	Planned for Sprint 2

9	As a hub administrator, I would like to kick people off my hub	2	Planned for Sprint 2
10	As a user, I would like to connect to a hub near me.	8	Incomplete: moved to Sprint 2
11	As a user, I would like to sign out out of my Google account	2	Completed in Sprint 1
12	As a user, I would like to see which hubs are near me	4	Incomplete: moved to Sprint 2
13	As a user, I would like to leave a hub	3	Planned for Sprint 2
14	As a user, I would like to pull up a YouTube search in app	6	Completed in Sprint 1
15	As a hub administrator, I would like to see the current video playing on my screen	2	Completed in Sprint 1
16	As a user, I would like to automatically be granted access when returning to a hub I have previously provided the correct pin for	8	Planned for Sprint 2
17	As a developer, I would like to inspect my code, unit inspect my code, and inspect the design of my code.	100	Planned for Sprint 1 & 2
	Total	171	

Non-functional

Backlog ID	**Non-Functional Requirements**	**Hours**	**Status**
18	As a user, I would like personal information to remain secure and private	8	Planned for Sprint 2
19	As a developer, I would like to learn how to set up a local	4	Completed in Sprint

	development environment for using ng and AngularJS		1
20	As a developer, I would like to learn how to use AngularJS	20	Completed in Sprint 1
21	As a developer, I would like to learn how to use Firebase	10	Completed in Sprint 1
22	As an administrator, I would like all songs to be searched and selected from YouTube	4	Completed in Sprint 1
	Total	46	

Use Cases

Case: Google login

Action	**System Response**
1. Click Google button	2. Open Google login
3. Input credentials	4. Login user
	5. Close Google login

Case: Change username

Action	**System Response**
1. Return to nickname screen	
2. Enter desired nickname in text box	
3. Press enter button	4. Change associated username
	5. Display create/join hub screen

Case: Add song to playlist

Action	**System Response**
1. Create new hub	2. Pull up hub admin view
3. Search for video in search bar	4. Pull up top 5 search results
5. Select desired video	6. Add video to playlist

	7. Return to playlist view
	8. Begin playing video

Case: Upvote a song in a playlist

<u>**Action**</u>	<u>**System Response**</u>
1. Create a hub	2. Display admin hub view
3. Search and select a video	4. Enter video into playlist
	5. Return to admin hub view
6. Press upvote button on song	7. Change upvote count
	8. Change button color

Case: Downvote a song in a playlist

<u>**Action**</u>	<u>**System Response**</u>
1. Create a hub	2. Display admin hub view
3. Search and select a video	4. Enter video into playlist
	5. Return to admin hub view
6. Press downvote button on song	7. Change downvote count
	8. Change button color

Case: Reorder playlist after a vote is cast

<u>**Action**</u>	<u>**System Response**</u>
1. Create new hub	
2. Search for and enter three songs	3. Enter songs and update playlist
4. Enter upvote for third song	5. Make third song now second

Case: Create hub with name and pin password

<u>**Action**</u>	<u>**System Response**</u>
1. Click the create hub button.	2. Pull up hub creation view
3. Enter desired hub name and pin	4. Pull up hub admin view

Case: Remove song from playlist

Action	**System Response**
1. Create new hub	2. Pull up hub admin view
3. Enter two songs into playlist	4. Update playlist after each video
5. Select red minus button and hit “remove”	6. Delete video from playlist
	7. Update playlist view

Case: Kicking somebody from hub

Action	**System Response**
1. Create new hub	2. Pull up hub admin view
3. Join hub on another device with new account	4. Update users list
5. Select red minus button and hit “remove” on user	6. Remove user from user list

Case: Connect to nearby hub

Action	**System Response**
1. Select to join hub from create/join hub screen	2. Pull up join hub options view
3. Navigate to nearby hubs view	4. Display nearby hubs
5. Select desired hub from list	6. Connect to hub
	7. Pull up user or admin hub view

Case: Sign out of Google account

Action	**System Response**
1. Navigate to create/join hubs screen	
2. Select sign out button	3. Pull up user login screen

Case: Checking which hubs are nearby

Action	System Response
1. Select to join hub from create/join hub screen	2. Pull up join hub options view
3. Navigate to nearby hubs view	4. Display nearby hubs

Case: Leaving a hub

Action	System Response
1. User presses back button	2. Prompt the user yes or no dialog
3. Press yes or no to leave or not leave hub	4. Pull up create/join hub view (yes)
	5. Stay in current hub (no)

Case: Searching YouTube in app

Action	System Response
1. Select “join hub” from create/join hubs view	
2. Search for a hub by name in search bar, press enter	3. Pull up similar hubs to search

Case: Watching current video on screen

Action	System Response
1. Create new hub	
2. Enter two new videos into queue	
3. Current video ends	4. Pull next video from playlist
	5. Load and play video on screen