CPSC436I Project Proposal

Group Members: Martin Palanca, Sherry Zhang, Yancey Yang

Overview:

SoundBubble is a platform that leverages Spotify's public API to build logs of a user's recently played tracks and makes this data visible to other users. Users can create and join groups to refine which users' song logs appear on their feed, as well as rate songs using an 'thumbs up' and 'thumbs down' scale.

Project Description

This project is intended for users who are subscribed to the music platform Spotify. It will allow users, and other users within their "Group", to view each other's song logs and interact with the logs by "Up Voting" or "Down Voting" a particular log. The type of data we will be storing are Songs along with User Information. This data will be able to be shared among users to access within the group. Additional functionality that we may add/remove based on time constraints is the ability to recommend a played song to another user, and adding commenting functionality to a particular song log for users to provide their opinions (e.g. "This song is awesome!", "This song is lame").

Project Task Requirements

Minimal requirements
1) Retrieve a user's recently played song log
2) Create a user interface for interacting with Spotify song logs
3) The ability to "Up Vote" and "Down Vote" a particular song log
4) The ability to add users to a private group
5) User log in capabilities

Standard requirements
1) Be able to clear a user's posted song logs
2) Album cover appears as an icon for the song log
3) If song logs are large, paginate the song log
4) Create a clickable link on a song log to redirect the user to the spotify song page

Stretch requirements
1) Adding song log commenting functionality
2) Adding the ability for a user to recommend a played song to another user

Breakdown of Minimal Requirements
1) Retrieve a user's recently played song log
   a) Complete a technical analysis of Spotify's public API to determine available functionality and limitations

b) Implement retrieval of a user's recently played songs using the public Spotify API
c) Store song logs
d) Parse song data into a Song object to retrieve relevant data (e.g. song name, artist, timestamp, hyperlink, etc).

2) Create a user interface
a) Create a main/home page with navigation to other pages (about, song logs, group, etc).
b) Create an interactive dynamic song log dashboard that updates on refresh
c) Implement design of the UI so that offered functionality is intuitive

Prototype 1
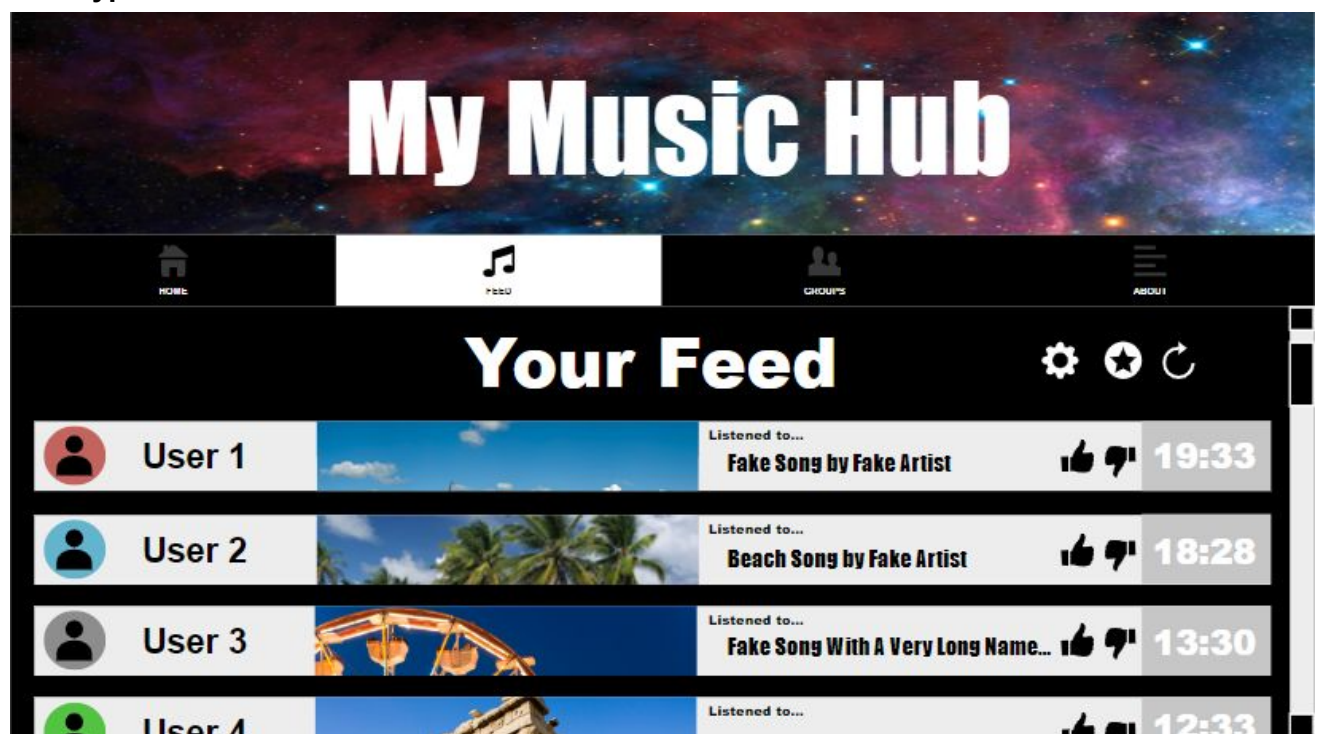
This sketch offers an early view of our music dashboard and the features it is intended to have. They include:

- Song logs from various users assembled into a "Feed" with most recently played songs appearing at the top
- An icon representing the user and a photograph of the user if applicable
- A photograph of the album cover next to the song name.
- The ability to upvote or downvote a user's played song
- Depending on time constraints, the name and artist may act as a hyperlink to the playable song on Spotify's platform
- Timestamped songs

Prototype 2
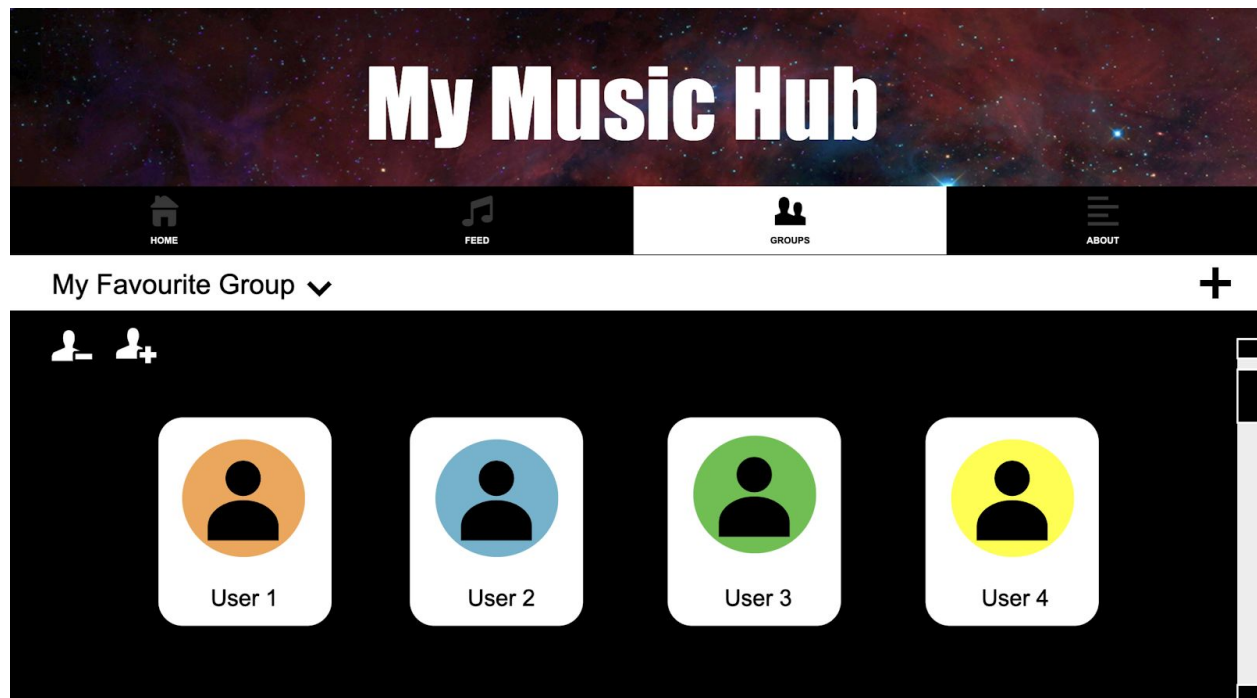
Functionalities on this page:
- Lists all groups the current user is in
- Lists all users currently in the selected group
- Add a user to the group
- Remove a member from the group
- Create a new group
- Depending on time constraints, clicking on a user from this page may show a list of songs they upvoted/downvoted