

**Title:**

Melodie

**Who:**

Alexander Brimhall

Daniil Garusov

Bella Longo

Casey Rudzki

Gabo Sambo

Jaden Snell

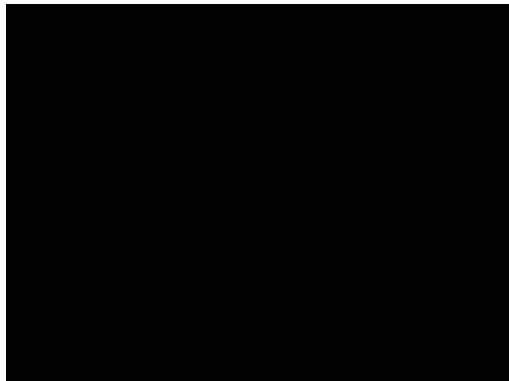
**Project Description:**

The purpose of Melodie is to add a community aspect to the listening experience. It takes in user data using the spotify API and displays statistics to the user through their profile. They will be able to see their top tracks and artists as well as add snippets to their bio. Another aspect of Melodie is the ability to view their friends' listening habits through the communal sharing of data through the website. Users will also be able to view a leaderboard of all users of the website regarding the amount of music people on the website listen to, the number of unique albums listened to, as well as the number of unique artists listened to.

The web application will allow music enthusiasts to connect with others fostering an inclusive environment for the users. It gives the opportunity to further develop the consumer's listening taste and share an intimate aspect of their life with their friends. The interactive interface immerses the user into their world of music as well as the music of others. Melodie's social media-esque properties make it the next generation of music collaboration.

**Project Tracker:**

<https://github.com/users/bellalongo/projects/1>

**Video:**

[https://drive.google.com/file/d/1SYXFkdXbMMrjsZOHqLwMEE\\_liJkeW1K-/view?usp=sharing](https://drive.google.com/file/d/1SYXFkdXbMMrjsZOHqLwMEE_liJkeW1K-/view?usp=sharing)

VCS:

<https://github.com/bellalongo/group7project>

### Contributions:

**Alexander Brimhall:** I assisted in using the spotify api to allow users to log in to our site, I created the functionality for users to search songs on the music page. Using Daniil's work pulling music data from spotify I got the music to play and pause on the music page. Lastly I modified some of the tables in the database to get snippets and posts to be associated with the user that created them.

**Daniil Garusov:** I created the log in functionality, integrated the music playing from the spotify api and made the snippets play on the profile page. I did alot of github merge conflict resolution.

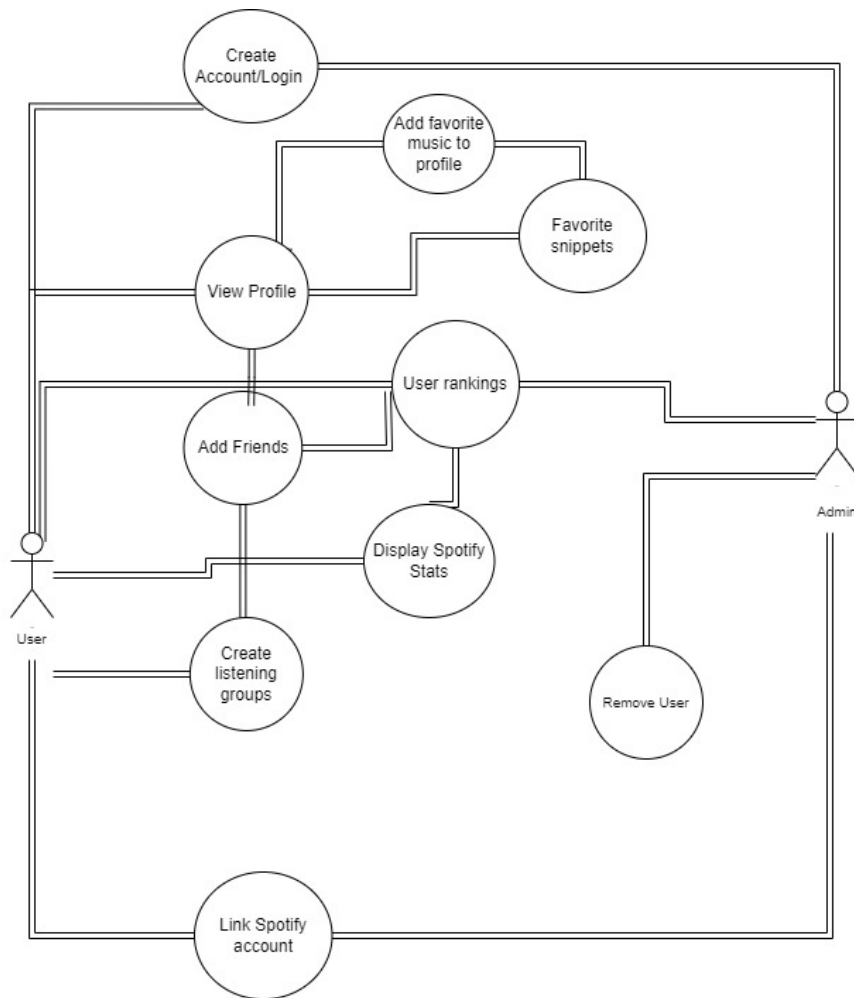
**Bella Longo -** I made the website design, as well as the home page functionality. I worked to have it so that when a user makes a post, it is stored under their username, so to generate a homepage, the user's friends and their posts are parsed thorough. I also had the top tracks in the US, as well as top new songs display on the homepage as well.

**Casey Rudzki:** I worked on the profile page. I worked on adding the user's info to the profile page, using the spotify api so that the page could display the user's name, profile picture, and then display their top 5 tracks.

**Gabo Sambo:** In this project I helped more on the frontend development Mainly on the html end I made the skeleton for some pages then we progressed to using current data from API's in our home page we used two API's the Spotify API and I also implemented the Billboard API.

**Jaden Snell:** I worked on both the friends page and adding the artists and friends columns to the home page using our database as well as part of the design of the home page. I also added the functionality of adding a friend and the ability to delete a user with their user\_id.

## Use Case Diagram:



## Test Results:

### Test Features:

1. Login functionality
2. Playing the snippets
3. Updating the feed

### Login functionality

- Test Case:
  - Make sure the login request sends to spotify
  - Make sure the account gets created if registration is made
  - Test Data Used: spotify username and password
  - Test Environment: Login page and spotify pop up
  - Test Results: User has registered account and it is linked to spotify
  - Testers: Our group and other groups in the course

### Playing the snippets

- Test Case:
  - The snippet plays when the user presses play
  - Test Data Used: the user and the snippet stored in the database
  - Test environment: profile page
  - Test results: the music plays
  - Testers: people from another group

### Updating the feed

- Test Case:
  - The feed updates when a user posts
  - Test Data Used: the user, the user's friends, and the post that the user uploads
  - Test Environment: feed page
  - Test Results: the feed updates
  - Testers: Another group using the spotify API

### Observations

#### Doing

- Logging in to spotify, and then navigating to the home page to view the feed, trying to click on the top tracks, then adding friends, and snippets

#### Reasoning

- There is an assumption that some things on the website are buttons, and they want to add music to their profile

#### Behavior

- Yes, the user added their top snippets and add top tracks

#### Reason for Deviation

- Assumed some things were buttons

#### Changes made

- Adjusted the UI to clear up confusion

### **Deployment:**

To deploy the app, run the command docker-compose up in your terminal, then go to localhost:3000 to access the application.