- Title: Sound Sphere
- Who: Eric Fithian, Owen Vangermeersch, Ryan Wagster, Yuhe Zou
- Project Description:

Sound Sphere is a social media music-sharing platform. The project can be described as if Instagram and Spotify had a baby. The purpose of the platform is to cultivate a music-sharing and discovery platform for users.

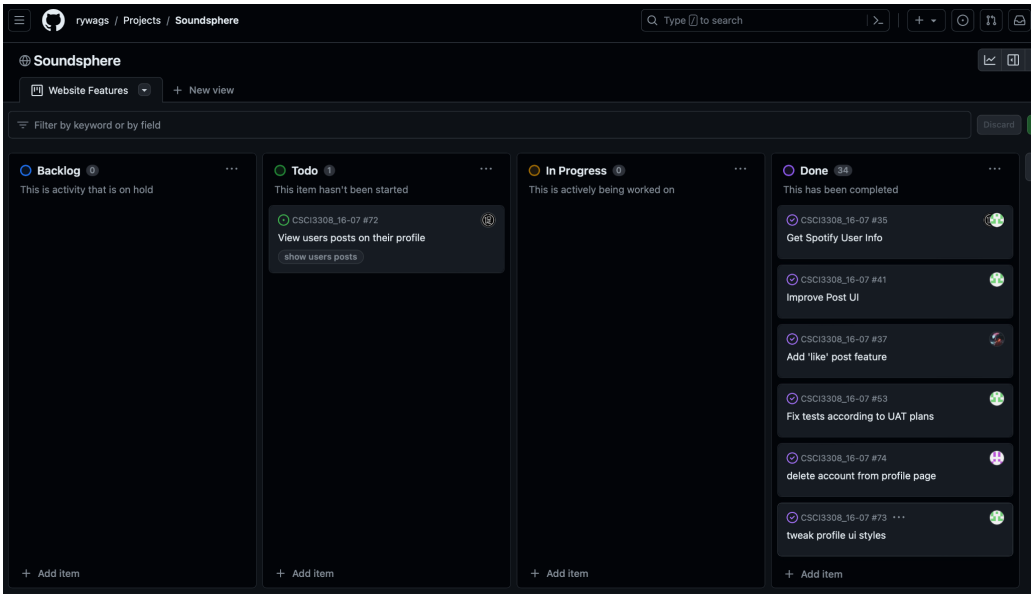
The functionality of the end product includes various features. To start, the user can create an account and link that account to their Spotify account. This allows access to the user's top songs, and top artists, as well as other data like their profile picture. This is all done through the Spotify API. When a user registers an account, their profile page is automatically updated with their current top three songs and top three artists.

In terms of sharing music, Users can create a song post. When they do this they enter the Spotify song link and add a comment/description for their post. Only people who follow that user will see that post. Alternatively, users can search for other users and view their posts on their profile before following that user.

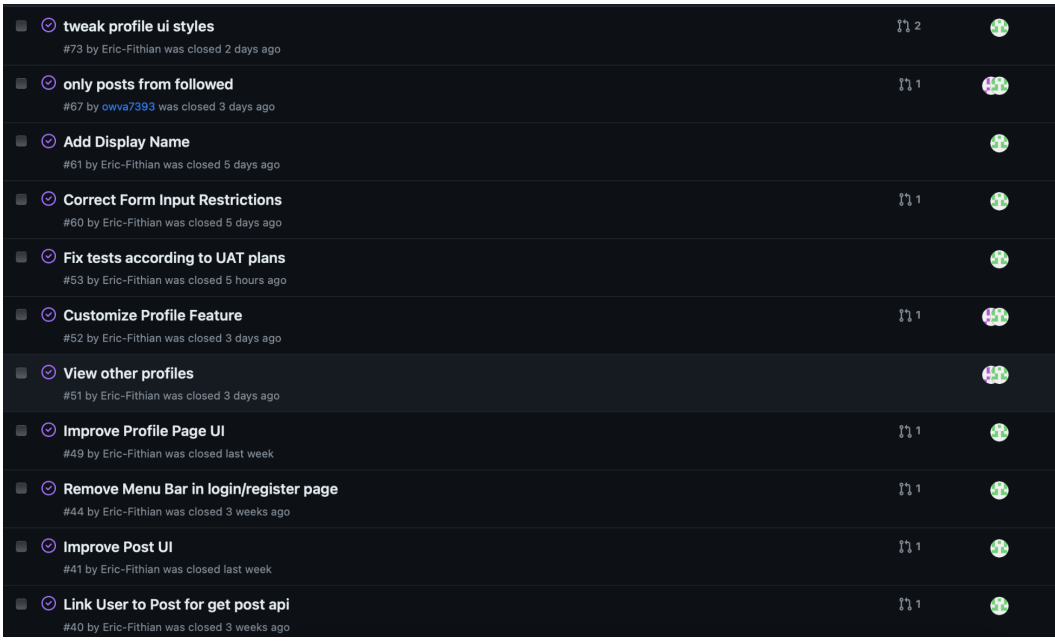
In terms of discovering music, users have a home/feed page that contains a running list of posts from the people they follow. They interact with these posts by liking, commenting, going to the song link, and listening to the song. They can view other people's comments as well as the total number of likes for that post.

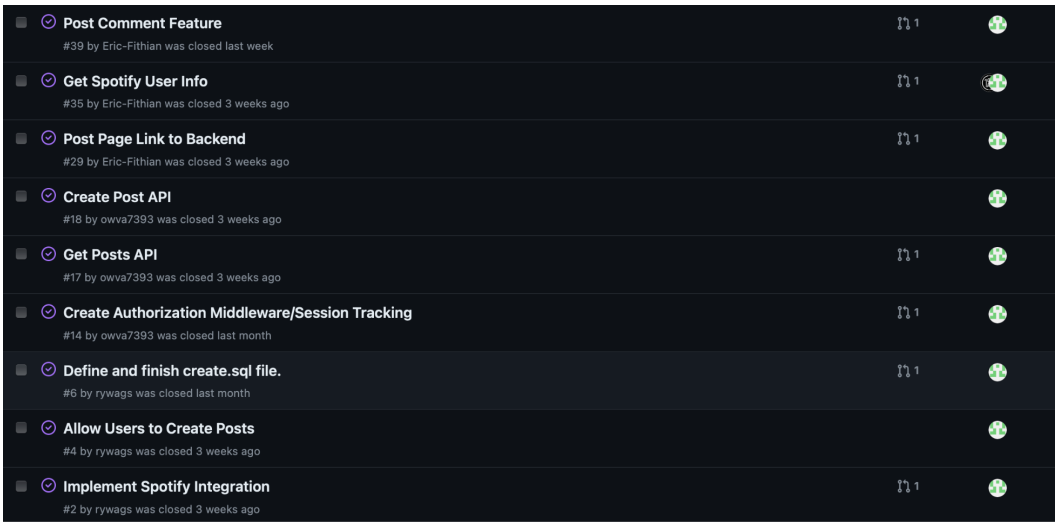
Users can discover/find other users using the search users functionality. This allows users to visit the profile of any user and see their top songs, top artists, and their most recent posts.

- Project Tracker - GitHub project board:
    - https://github.com/users/rywags/projects/1

- 

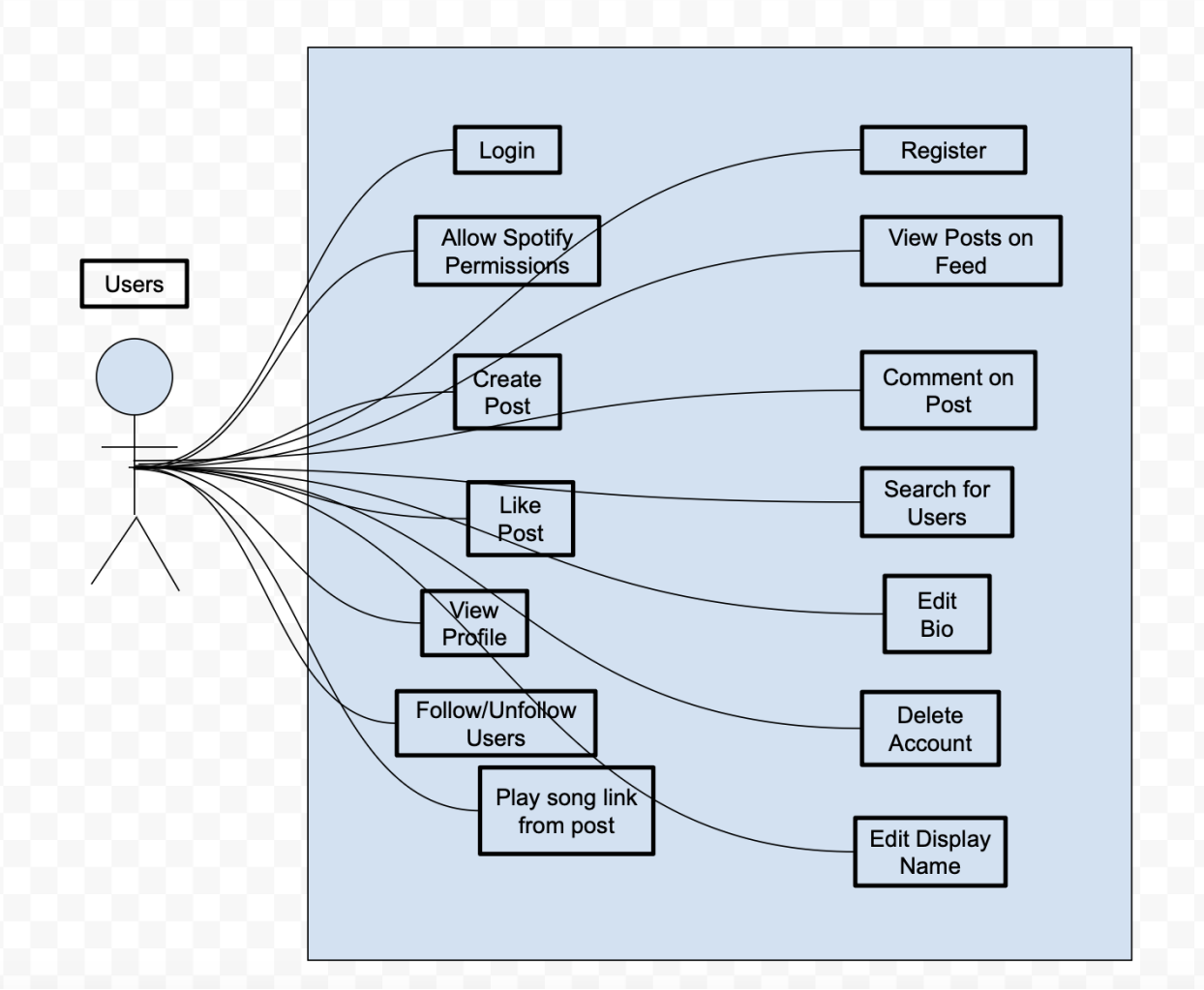
- Video: (video is 6 minutes long, stop at 5 minutes if necessary)
  - https://drive.google.com/file/d/1_dZf0CznOjWnn7Sylk5fGqJ3JS_EIYpZ/view?usp=sharing
- VCS: https://github.com/rywags/CSCI3308_16-07
- Contributions:
  - Eric Fithian - Features worked on:
    - Spotify API 0Auth flow and API integration
      - Key management features
      - Get song api
      - Get user account information api
      - Get top tracks api
      - Get top artists api
      - Sign into spotify and request permissions
    - Database design
      - Sql init script
    - Session management
      - Authorization middleware
    - Create Post Feature
    - View Posts on feed Feature
    - Comment on Post Feature
    - Like Post Feature
    - View Profile Page Feature
      - Profile Info Header
      - Profile Top Songs
      - Profile Top Artists

- Login and Register Form Restrictions
- Issues I Worked On/Completed: (I have the green profile picture)
    - 

    - 

- Owen Vangermeersch: Features Worked On:
    - First Iteration of Homepage - static
    - Updated menu/navbar with proper names and API routes
    - Helped implement the CSS stylesheet
    - Search Users Function
        - Front-end
        - API routes
        - Updated viewing of other profiles to link with the search
    - Follow user function
        - Database (User-user)

        - Following/followers counters on the profile
        - Profile Edit feature
        - Profile Delete Feature
        - Created Logo
    - Yuhe Zou:
        - I mostly worked on the foreground of our website. I created the register page where people can register for new accounts by their email and password. I also created the Make Post page where users can make song posts using a Spotify song URL and their comments.
    - Ryan Wagster: Features Worked On:
        - CSS stylesheet for global styles on website
        - Created webpage background using logo
        - Hosted webpage icon using logo and attached to head
        - First iteration of profile page
        - First iteration of login page
        - First iteration of header, menu, and footer
        - Attached user's posts to user's profile
        - Adjusting minor bugs across any pages
- Use Case Diagram:



- Test results:

1. Login Feature Testing

Test Case 1: Successful Login

- **Results:**
  - All users successfully logged in using the provided credentials.
  - Redirection to the homepage was smooth and without delay.
- **Observations:**
  - No issues were observed during this test.
  - The login process was intuitive and user-friendly.

Test Case 2: Unsuccessful Login

- **Results:**
    - Users were unable to log in with incorrect credentials, as expected.
    - The error message for incorrect credentials was clear and visible.
- **Observations:**
    - Users liked the clear communication of the login error.

2. Registration Feature Testing

Test Case: Successful Registration

- **Results:**
    - New users were able to register successfully.
    - Users were correctly redirected to the login page after registration.
- **Observations:**
    - A few users found the registration form slightly lengthy.
    - Could be better to redirect users to home page after registration

3. Delete User Feature Testing

Test Case: Successful Account Deletion

- **Results:**
    - Users could delete their accounts without issues.
    - Account deletion led to the expected redirection to the login page.
- **Observations:**
    - Some users hesitated before confirming the deletion, suggesting the need for a more clear way to demonstrate that they are deleting their account.
    - The process was found to be straightforward by most testers.

- Deployment:

The app was deployed on Azure for a limited period of time for Lab 13: http://recitation-16-team-07.eastus.cloudapp.azure.com:3000. In order to run the app, you can find step by step instructions in the ReadMe within this repository: https://github.com/rywags/CSCI3308_16-07