

Ben Wedeen

Kyle Kells

Arjun Lakshmi Narasimhan

CSCI 4448 Project 6 Update

### **Advanced Spotify Search**

**<https://github.com/Larjun/Advanced-Spotify-Search>**

#### **Status Summary:**

##### **- Work Done:**

Ben Wedeen

- Implemented log-in page w/ authentication, side navigation bar, playlist data display and general structure and style for the main application page

Arjun

- Implemented functional track searching element, implemented ability to view track attributes such as Acousticness, Energy, Danceability, and many others provided by the API...

Kyle

- implemented top navigation bar which displays the user's Spotify username and profile picture

##### **- Changes and Issues Encountered:**

Instead of using NodeJS, we utilized React for our project. We ran into small issues with proper authentication and permissions to view certain information on user profiles, but we were able to solve those issues. We also ran into the problem of having to disclose

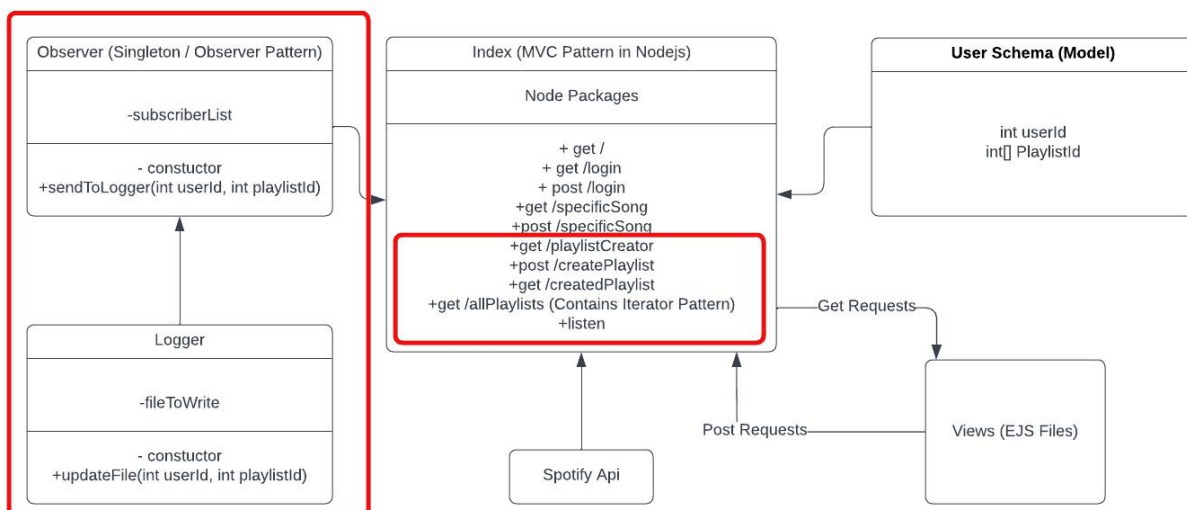
our personal spotify login credentials in order to access the API together. We fixed this by creating a spotify account dedicated to the project.

### - Use of Design Patterns:

MVC is utilized through our Node.js project structure. The view is our UI, how our user interacts with our system. The model, or how our data is organized are JSON files obtained from the spotify API. The controller is all of the logic that handles Get and Post requests. The rest of our design patterns will be utilized in the process of creating playlists and storing user information as shown in the class diagram below.

### Updated Class Diagram:

The parts of the diagram enclosed in red rectangles have not yet been implemented.



### Plans for Next Iteration:

In our next iteration, we will utilize an iterator pattern to allow the user to create playlists and create an observer pattern that tracks user behavior. We also want to work towards our goal of making attribute criteria searchable. While Spotify API does not explicitly enable this, we want to try to sort through songs randomly to collect songs that fit the user's specified criteria. Our original plan was to connect to MongoDB to store users, but due to unfamiliarity with React, we are unsure if we could complete that within the provided time frame along with other features.