**Comp 5590 Course Proposal**

**Spotify Improvement Application**

**Yuan Tao Zhou**

**Background**

Music has become an integral part of society nowadays. One of the biggest music service providers is Spotify. As of the second quarter of 2021, Spotify had 172 million premium subscribers worldwide. ("Spotify users - subscribers in 2021 | Statista", 2022) Spotify is a streaming service that provides music from virtually all artists with a membership system. As a musician I often practice with this service, one of the problems I ran into was when I ran a playlist and songs would be all over the place in terms of tempo which can be frustrating to adjust to. That got me thinking not only me, but lots of musicians might face the same problem. I have built the core functionality that would work for me as personal use. Now I want to expand the functionality to allow other users to use it too.

**Methods**

When I first started to work on the project, I imagined that I would have to involve some sort of tempo detection system, there are independent tools for that then I would have to figure out how to integrate that. Luckily for me, there is a Spotify API for developer ("Home | Spotify for Developers", 2022) which provides tons of information about the songs. So far, I have been working in Node.js to retrieve data from the endpoints provided on the website ("Web API Reference | Spotify for Developers", 2022). I intend to build the front end with React. Right now, I will login to the landing page, and it automatically sets a pre-determinant playlist and grabs all songs at a set tempo (bpm =100), or range of tempo (bpm= 110 – 130), and then spits out to a new playlist. I have followed the example login from the website ("Authorization Code Flow | Spotify for Developers", 2022) I would like other people to be able to login into their own account and select from all their playlists to perform the query.

I do not intend to keep the user’s data after their query. I have not explored the security issue, if I am allowed to manipulate another user’s playlist with my app. If I couldn’t not perform the action directly then I would have to incorporate a Database and API. In that case, I would use MongoDB and Express and have the user stay on my application and host the playlist with my own account. I don’t think it is possible to have my application provide direct service of playing the songs.

**Tentative schedule**

**Market research,** I will contact peers and use different method of survey to gather ideas that can be done to improve the Spotify service. **1 week**

**Understanding login in**, I am currently using a sample login from the Spotify website. At the time I didn’t understand how the login worked, after further education I have more understanding of the express framework. **1 week**

**Implement different user login**, currently the app only works for my account. I intend on creating another account to test the functionality and build a login page. I will be focusing on the functionality in this week, if there is not time to build the login page, I will delay that part with UI part. **1 week**

**Implement different playlist selection**, after the user log into their account all their playlists will display and onclick they will be directed to select a tempo range to filter to a new playlist. **2 weeks**

**Building an UI**

This part will be covering the general look of the application. The core functionality is completed. **1 week**

**Sorting a playlist by bpm ascending and descending order**

After the user select a playlist then they can select to order the playlist in ascending or descending order in term of tempo, then the sorted songs will be in a new playlist. **2 weeks**

**Build a mock suggestion system from the endpoint data**

One of the latest features from Spotify is suggesting songs the user might want to listen to. I want to build a similar system, I will use songs with similar bpm first and pull some of the data such as danceability, artists and pull suggesting songs for the user. **3 weeks**

**Implement one of the features suggested from market research**.

This will be the biggest variable in terms of this project, depending on the feature it can take longer or shorter. Some idea might be developed outside of the time frame of the school term. **2 weeks**

**Refine UI and deploying, hosting**

This will the finally touch up on the application, deploy it to the public there is a marketplace from Spotify developer, that will be the end goal, but if the app can be used by the public the project is considered done. **1 week**

**References**

Authorization Code Flow | Spotify for Developers. (2022). Retrieved 10 January 2022, from https://developer.spotify.com/documentation/general/guides/authorization/code-flow/

Home | Spotify for Developers. (2022). Retrieved 10 January 2022, from https://developer.spotify.com/

Spotify users - subscribers in 2021 | Statista. (2022). Retrieved 10 January 2022, from https://www.statista.com/statistics/244995/number-of-paying-spotify-subscribers/

Web API Reference | Spotify for Developers. (2022). Retrieved 10 January 2022, from https://developer.spotify.com/documentation/web-api/reference/#/