Project Proposal for Capstone One
Prepared by John Mack
January 2020 Springboard Cohort
Prepared March 5, 2020

Project Summary:

This application will result in better music discovery by intelligently seeding the Spotify recommendation algorithm via Spotify's API. Put more simply, this app will music lovers to wave a magic wand and get back a collection of amazing music that they'll love.

Project Background:

Spotify has a robust and well-respected music recommendation engine and a vast music catalog. Spotify has developed technology that breaks down a song into both its musical components (tempo, detected speech, and emotional dimensions like happy and sad) and its catalog history (plays, likes, additions to playlists, etc.). The combination of these factors allows for a very robust and precise music recommendation experience.

One shortcoming of the traditional Spotify platforms (standalone app, web player, widgets, etc.) is the inability to specifically determine the seed criterion during a search for new music. The Spotify API allows the manipulation of at least some of these inputs when compared to their default recommendations. This app will leverage that capability to result in an exciting music discovery experience for users of this app.

App Demographics:

This app will primarily target Spotify premium subscribers as this app will be built on the Spotify service and would integrate well with automatic playlist creation and other Spotify user-level features. However, this app will also function with 30 second previews of resultant songs where no Spotify account is needed (if this functionality will still be supported by Spotify).

The common core demographic of this app's users is that they are music lovers who are passionate about new music discovery *and* have spent some time in the past trying to curate their perfect music library.

App Source Data:

This app will make extensive use of the Spotify API. The central unit of data this app will deal in is a Spotify song. Also, a list of several songs can become a Spotify playlist. Lastly, an individual Spotify account holder can both add and delete songs from different playlists and create or delete playlists themselves. The Spotify API currently allows for most of this functionality.

Project Design Considerations:

- The Spotfiy API is undergoing some changes… it is not certain which API endpoints will remain after these changes. There is no indication that these changes will impact this project's viability but we are at risk of an API change on Spotify's end.

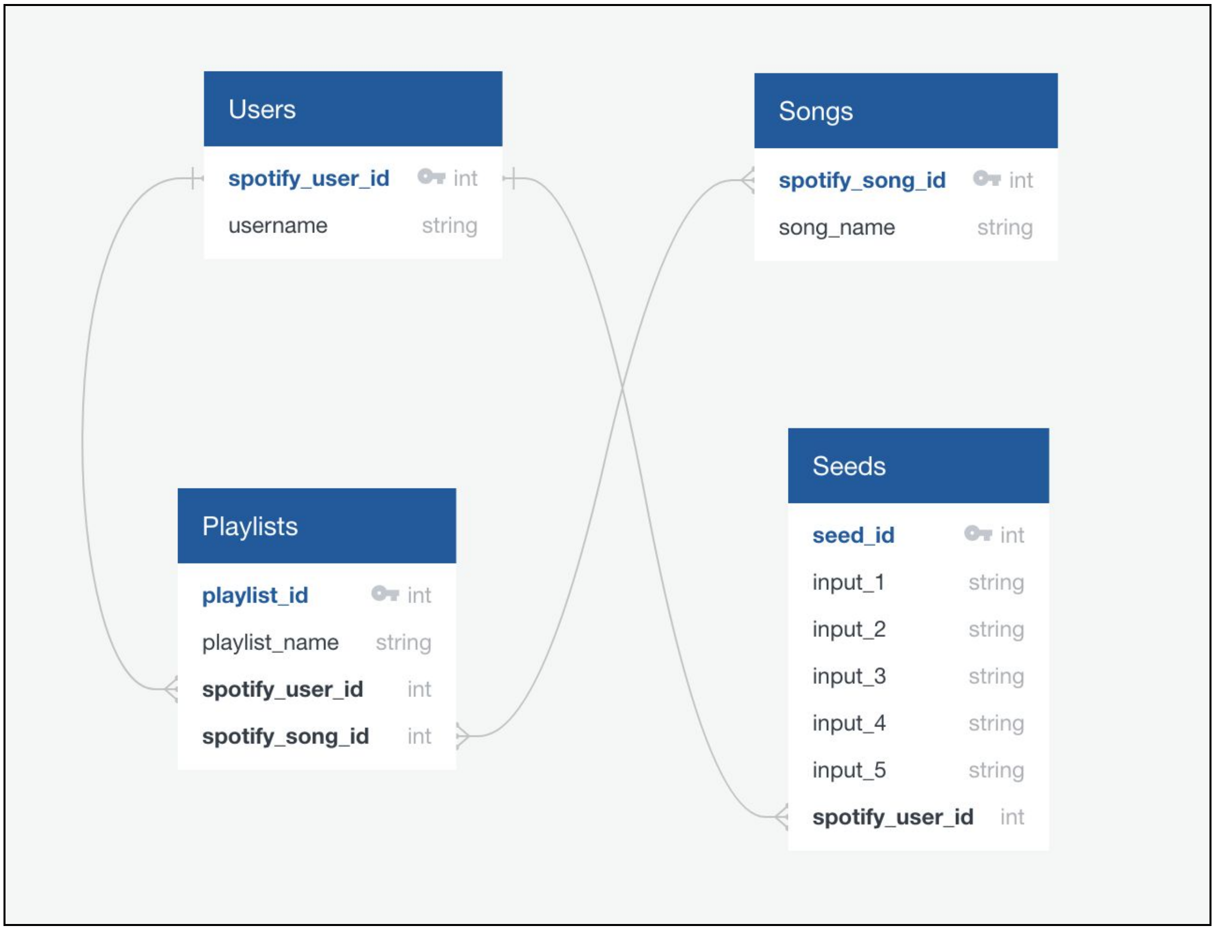
- A Spotify user that wants to integrate this app with their Spotify account will need to login to Spotify's authentication system through a Spotify widget they'll be redirected to. However, a user must also grant permission for this app to modify their Spotify song and playlist related data. This won't won't result in the storage of any sensitive data but will require user trust.

- At minimum, this app will take seed inputs (music genres, artists, songs, moods, temporal descriptors, etc.) and output a playlist of some kind (Spotify playlist, a widget that allows 30 second previews, linking to those songs' music videos on YouTube, etc.). Additional desired functionality would include the importing and editing of these Spotify playlists to a user's Spotify account.

- The user will begin their flow through our single page app on a home page featuring the prominent option to either login to their Spotify account or stay logged out. Then, the user can select up to five seed inputs (Spotify's current limit) via an HTML form. The results will then be displayed on this first page along with contextually dependant options involving exporting playlists, listening to the music, etc.

- One aspect of this app that enhances its characterization as more than a simple CRUD app is the how the implementations of the seed options are presented to the user. Rather than just copy and paste the descriptors Spotify uses, this app will obfuscate some of this from the user and try to deliver the type of songs the user is really searching for. As one example, three of Spotify's seeds include "acousticness," "danceability," and "energy." In the "punk rock" genre, for example, think about the types of songs you'd hear with that seed combination: maybe acoustic covers of high energy, mosh-pit-worthy, acoustic covers of pop songs. This combination could easily be achieved with a setting from the user that essentially communicated something like, "I like punk rock and I want to dance!"

- A general visualization of the preliminary database schema is below:



Users	
spotify_user_id	int
username	string

Songs	
spotify_song_id	int
song_name	string

Playlists	
playlist_id	int
playlist_name	string
spotify_user_id	int
spotify_song_id	int

Seeds	
seed_id	int
input_1	string
input_2	string
input_3	string
input_4	string
input_5	string
spotify_user_id	int