Project Description

My project is called SpotiLast. It is a program that loads a user’s scrobble data from last.fm and allows the user to do various things with this data. For reference, last.fm is a service that tracks every single you listen to. Each listen is called a “scrobble”. My program downloads all of a user’s scrobble data and allows them to filter through it in various ways. There are also graphs available to look at to see various statistics, such as peak listen times, favorite artists, songs, genres, and more. In the app, there are also various queries that you can ask, such as “What song am I most likely to listen to at a certain time” or “What song am I most likely to listen to if I am listening to a different song”. The last functionality of this program is that it can take any set of songs the user selects in the app and convert the set of songs to a Spotify playlist.

Competitive Analysis

Soundiiz is a service that is similar to my app. It is able to take last.fm data and convert it to a Spotify playlist. However, you need to pay to transfer more than 200 songs. Furthermore, they don’t have graphs or the extra search queries that my app has. Another service that is relatively similar is Spotlistr. Spotlistr lets a user transfer their most scrobbled tracks to a Spotify playlist as well. Like Soundiiz, there are no graphs or extra queries that you can ask. The app also does not seem to work with a large volume of songs, as it hangs every time I try.

Structural Plan

The main part of my program will be structured with different screens. The summary is listed below.

* splash/login
  + will prompt user for last.fm + spotify api key, api secret, and username
  + directions on where to find said info will be on this screen
* Loading screen
  + Loads in last.fm all tracks and top tracks
* Main search screen
  + After all data has been loaded, a screen with a search bar + a filtering system will be at the top, with all the songs arranged in a row format with a small picture of the album art, song name, album name, artist, release date, and length
    - Filters - Release date, genre, length, artist, album, times scrobbled,
  + There will be check boxes next to each song that will allow you to select it
* Graphs/Analytics Screen
  + Graphs for listening data that you can mess with
    - Will try to do these manually with tkinter, might be hard
    - Peak listening times
    - Peak artists
    - Peak songs
* Query Screen
  + Various queries that you can ask
    - “What song am i most likely to play at this time”
    - “What song am i most likely to listen to after I listen to a certain song”
* Export Screen
  + Takes all selected songs and exports them to a Spotify playlist

Algorithmic Plan

The trickiest part of the project is the algorithm used to recommend on the query screen. Originally, I planned on using recommendation matrices for these algorithms. These algorithms seem super complex and I am not sure if they will work well for the format of data that I have, so I may do probability based recommendations instead.

Time Plan

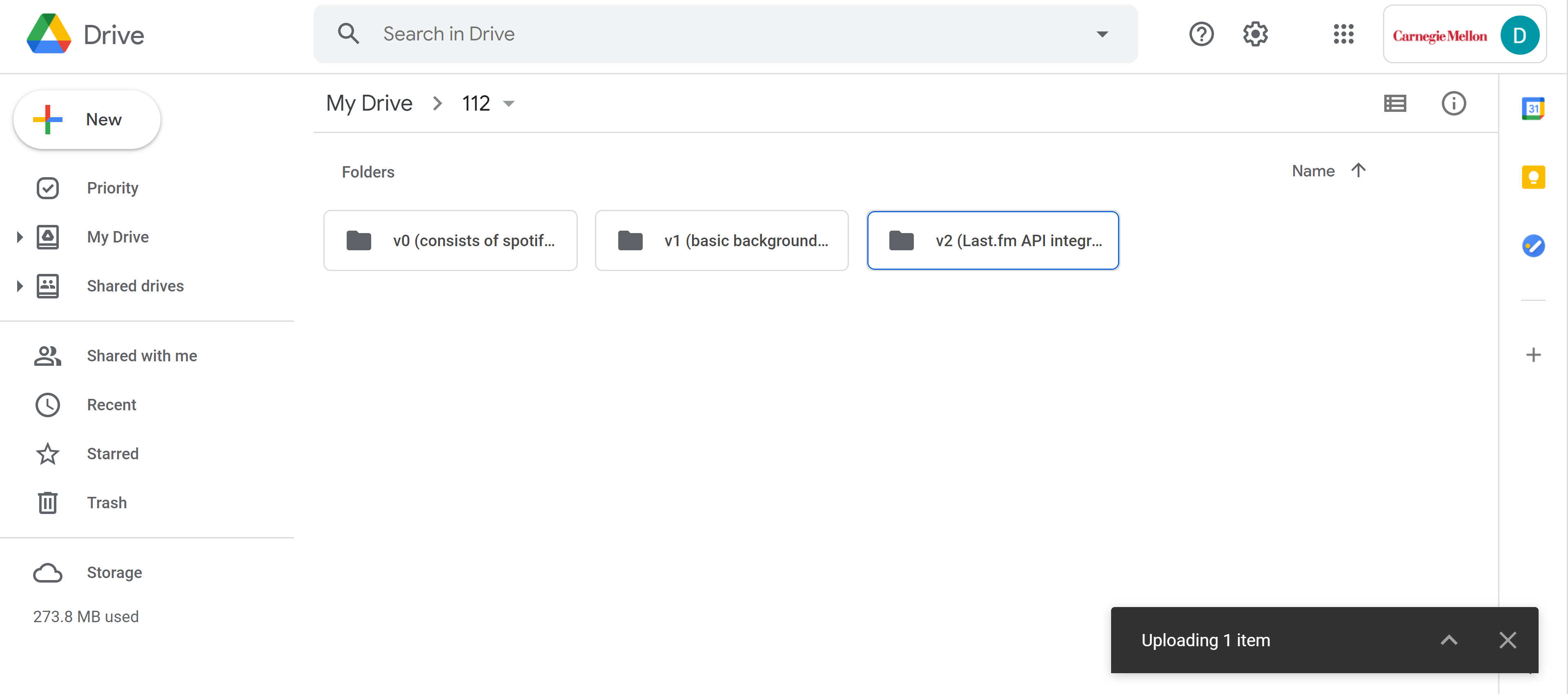
TP1 – Have last.fm loading, exporting, UI for searching and exporting

TP2 – Finish graphs, algorithms, and the UI for the query screen

TP3 – Touch up on any bugs and create the tech demo

Version control plans

Currently uploading backups of my code to Google Drive



Module List

Spotipy, last.fm API

TP2 Update:

* Removed query feature (for now)
* Removed date filter for searching
* Getting last.fm data, selecting/sorting through songs, searching, spotify export all complete
  + Need to do graphs

TP3 Update:

* Changed splashScreen to have textbox inputs rather than message windows
* Added three different types of graphs that allow for changing of filters
  + Pie chart, Bar graph, line graph
* Added query: “What songs am I most likely to listen to at a certain time?”
* Added finalScreen page to show songs that had errors and allow for restarting of app
* Added ability to select all songs and deselect songs on search tab