Theresa Pham

CSC 415: Software Engineering

Assignment 3 Proposal

Spotify WorldView Proposal

There currently exists an open source project called "Album Availability" which is a web application that allows a user to either input the URI of an album or drag an album straight from the Spotify application to get a visualization of the countries in which that album is available. I would like to modify and add to this project to take it to the next level.

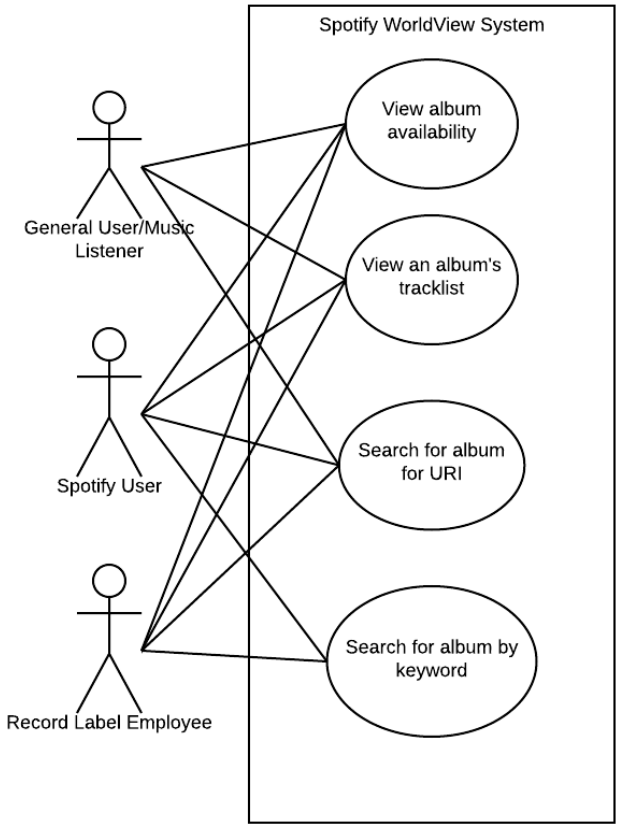
The project will be entitled "Spotify WorldView", and since I will be contributing to an existing open source project, I am working on Option 1 of the two project options. By adding greater functionalities such as being able to do the reverse of what the app currently does to be able to click on a country to view albums available in the country, as well as give the option to generate a playlist of available songs that are most popular in a given country, I will be making significant contributions to the project that will increase its usability. Many people enjoy using Spotify to stream music and discover new music to enjoy. My application will give Spotify users another facet of discovering new music via country and also gives a visualization with the map, which Spotify does not have in its interface. Because the current "Album Availability" app is web-based, I will be keeping the app web-based. The "Album Availability" application also seems to have been done entirely through use of HTML, CSS, and JavaScript, but I take that code and also utilize Ruby, and more specifically Ruby on Rails, which caters more to web application development. HTML, CSS, and JavaScript can also be merged into a Rails app, so utilizing this language for the bulk of the code would be ideal.

The algorithm that will be used will most likely combine a merge sort with another way of prioritizing music and songs so it may cater to either the searches that were made previously or the music that a user listens to if there is time to implement the functionality of user login on the application. Merge sorts are quickest when data does not need to be stored and when dealing with a large set of data, and the Spotify database includes thousands upon thousands of songs that

would need to be sorted through when generating these playlists, and because the playlist generation function of the app will be where a data structure is implemented, the data structure that would be best suited for this would be a priority queue implemented using a heap structure, as the nodes of a heap structure would allow for comparison of songs based on popularity or other factors which would then help generate the playlists themselves in a priority queue. It makes sense for playlists to be in some kind of queue since it is natural for a playlist to play songs in order unless the user shuffled the playlist or chooses to play a specific song.

To be able to work with this open source project, I will need to learn how to use the tools that brought together the "Album Availability" application, which includes Spotify API (RSpotify was found and used, a Ruby Wrapper for the Spotify API). I will also then need to learn Ruby on Rails to be able to code the application itself and merge it with the HTML, CSS, and JavaScript that currently exists. My plan for learning these is to first spend time for each looking through documentation for these APIs and tools, and then to actually look at code that can be implemented and attempt to run some of that code myself so I can learn to work with each language and tool individually. I will then look into other projects where similar things may have been accomplished as what I have planned for this project and where similar languages and tools where used to learn how to use these tools in conjunction with each other. My plan for learning is along the lines of how people do unit testing and integration testing for applications, except I will be "unit learning" and "integration learning".

Use Case Diagram:



License Comparative Analysis

3 different licenses that are commonly used for open source projects include the MIT license, Apache License 2.0, and the GNU GPLv3 license. The MIT license is a very permissive license that allows anyone to take the existing and code to use it and make any changes they desire as long as they provide attribution back to the original creator and do not hold the creator liable. The Apache License 2.0 is a license that is also fairly permissive, but requires preservation of license and copyright notices. With this license, contributors provide an express grant of patent rights. The GNU GPLv3 license is a copyleft license that requires anyone that distributes the original code or modified code to make the source available under

the same terms and the same license so that improvements can be shared and the project can continue to be built upon. Like the Apache License 2.0, copyright and license notices must be preserved. Because the original application used the MIT license to allow for open access to the code for use, that license will also be used for the Spotify WorldView application to respect the original content owner's decision to allow others to make changes to develop and modify the application if they so desire, and to continue to foster open source development.

Github Repository: itstheresa/spotifyWorldView
URL: https://github.com/itstheresa/spotifyWorldView