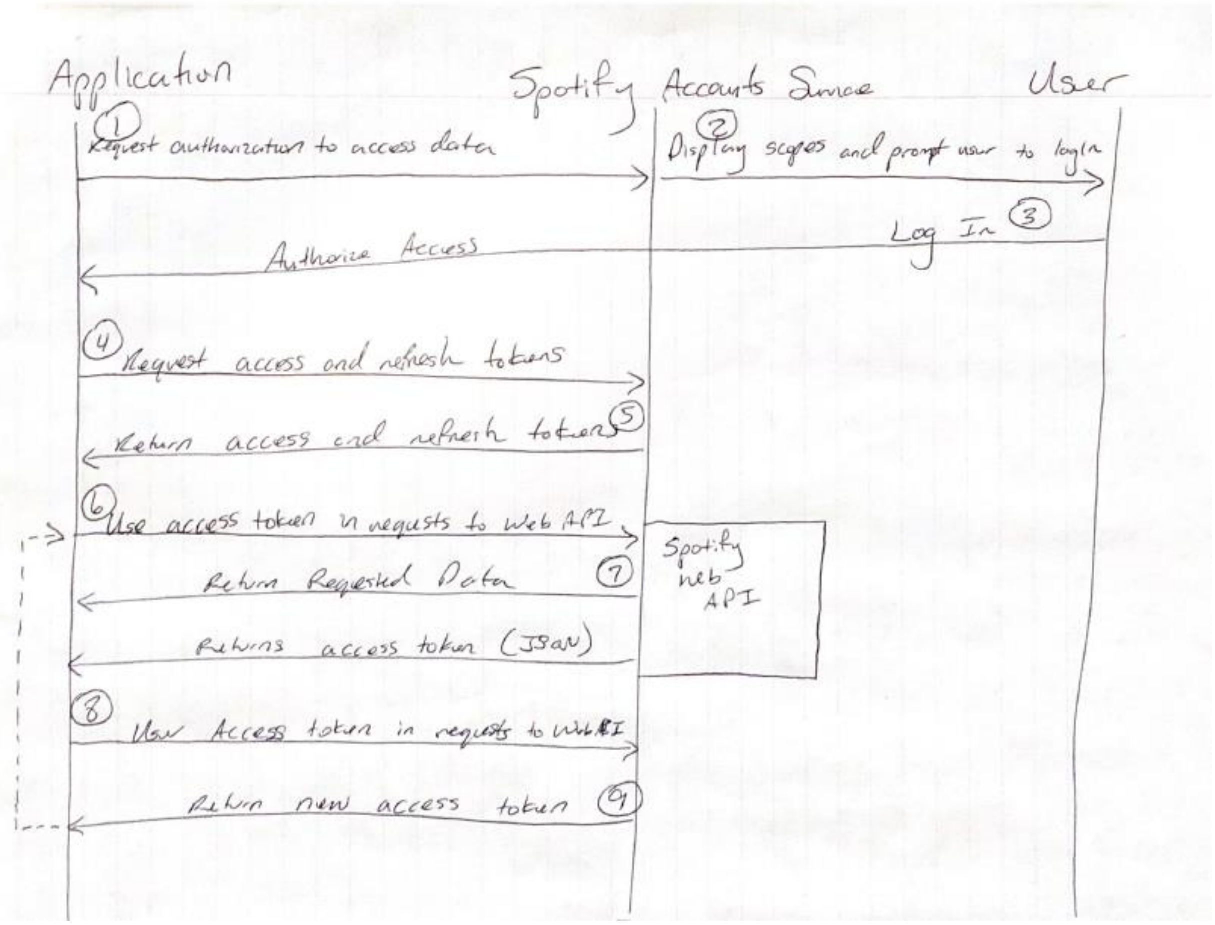
CAS411 Team Assignment 3: Analysis and Architecture Decision

1. The user story we used to create a diagram was the process of authorization code flow among the application, Spotify's Accounts Service, and the User. The diagram is found below:


Application
Spotify Accounts Service
User
1
Request authorization to access data
2
Display scopes and prompt user to login
Log In
3
Authorize Access
4
Request access and refresh tokens
Return access and refresh tokens
5
6
Use access token in requests to Web API
Spotify Web API
Return Requested Data
7
Returns access token (JSON)
8
Use Access token in requests to Web API
Return new access token
9

This diagram can be found in PDF format in the same directory as this report.

3. Listing of the steps of the happy path of the authentication user story:
    1. When the user first visits our application, the user will be redirected to the Login page, which will ask the user to provide his or her Spotify credentials.
        a. If the user provides incorrect credentials, the page will refresh and the user will be asked to submit valid credentials to log in to the application.
    2. Once the user has been authorized access by obtaining access and refresh tokens, the user can then use the access token for requests to the Web API.
        a. If the user somehow loses the access or refresh token, the application will log the user out and redirect him or her to the login page.
    3. With the requested data, or in our case the playlist name and contents, the application will then be ready to proceed to the next user story of converting Spotify playlists to YouTube playlists. The Web UI will then proceed to return a new access token.

a. If the requested data returned is missing, then that could mean that the user’s Spotify account does not contain any personal playlists, in which case the application will return “No available playlists to convert”

4. The architecture and platforms we will use to develop our platform are:
    a. Python Flask for the backend/application code: This is due to previous experience of David, who is responsible for the backend as he had previous projects working with Python Flask incorporating OAuth and APIs. Also, Python Flask is a powerful but simple framework that the other team members would like to learn as well.
    b. React JS: Andrew is in charge of working with the front end of the project, and he is currently working on another project using React as his front end. Other team members have only had experience in HTML so this would be a new learning experience for the majority of the team.
    c. MongoDB: Drew is working on the data storage and we have chosen to use MongoDB as our database. As per recommendation from Professor Donham and from our own opinions, we believed that a non-relational database would work best due to its ability to store and process large amounts of unstructured data.
    d. Spotify API: Saroj is in charge of working with the Spotify API, which includes the SSO process of logging into a Spotify account and accessing the playlists.
    e. YouTube API: Phumin is in charge of working with the YouTube API; understanding how to create new playlists and investigating how the recommendation of videos work in YouTube.