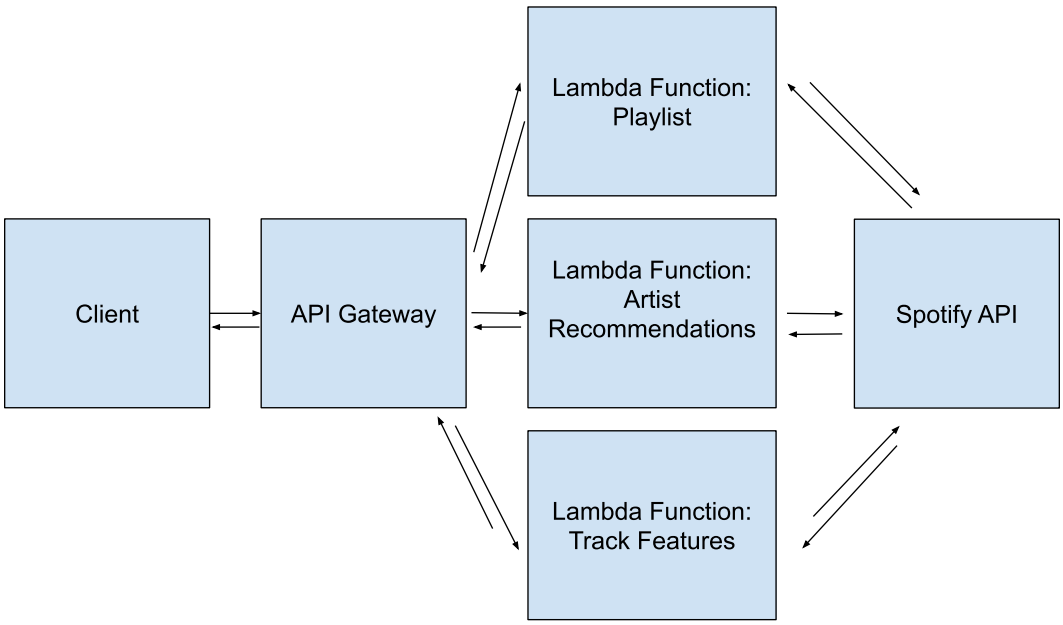
CS310 Final Project

Project Document

Angela Zheng

For the final project, I decided to create an app which allowed users to make generated playlists, receive artist recommendations, as well as receive song recommendations. The generated playlists feature takes in a genre from the user and returns a generated playlist from the top artists in that genre. Users can also receive artist recommendations by inputting an artist's name and getting back a list of artists which they would enjoy based on their search. The song recommendation feature allows users to pick between three track features: danceability, energy, and acousticness, specify a genre, and get a list of songs that are rated high in the track feature. The data used in this project was retrieved from the Spotify API.

Diagram of my Spotify App architecture



Description of each computation

**Generated Playlist**: On the client side, the user enters a genre from the list of genres that are on Spotify. By calling the API Gateway endpoint /playlist with a get method, the client is able to send the user input to the lambda function which the server side code can use to call the Spotify API endpoint. Generated Playlist uses Spotify's /search endpoint as well as the /artists/{id}/top-tracks endpoint to be able to generate a playlist given a genre. It does this by first calling the /search endpoint and passing in the search parameters of the genre, type: artist, and limit of 10. It stores the ids of the top 10 artists from the genre, and uses the ids to call the /artists/{id}/top-tracks to get one of their top songs. From there it compiles it into a playlist of the 10 songs and returns it back to the client which prints it out for the user.

**Artist Recommendation**: The client receives user input of an artist name. It uses the /artist-recommendations endpoint from API gateway through a get method and passes in the user input of the artist name. The lambda function uses that artist name and calls the Spotify api endpoint /search to retrieve the artist id. Using the artist id, it calls Spotify's /artists/{id}/related-artists endpoint and retrieves the artist names of the recommended artists. The lambda function sends back this information to the client which prints out the recommended artists to the user.

**Song Recommendations based on Track Feature**: The client takes in 2 inputs from the user: track feature and genre. Spotify has certain track features which each song has ratings between 0 and 1, such as: danceability, energy, acousticness, etc. For this project, I allowed users to pick between those three track features. From there they have to specify what genre they would want

the songs to be in. The client calls API Gateway endpoint /track-features with a get method and the parameters of the track feature and genre. The lambda function receives these inputs and calls the Spotify API endpoint /recommendations and passes in these parameters along with a high score for target_{track feature} within the search call. For example, if the user selects danceability, the call to /recommendations has a parameter of target_danceability=0.90 which ensures that each song returned has a high danceability rating. It uses the data it gets back from the API to compile a list of 5 songs which have high ratings in that feature. The lambda function returns this information to the client which prints it out to the user.