

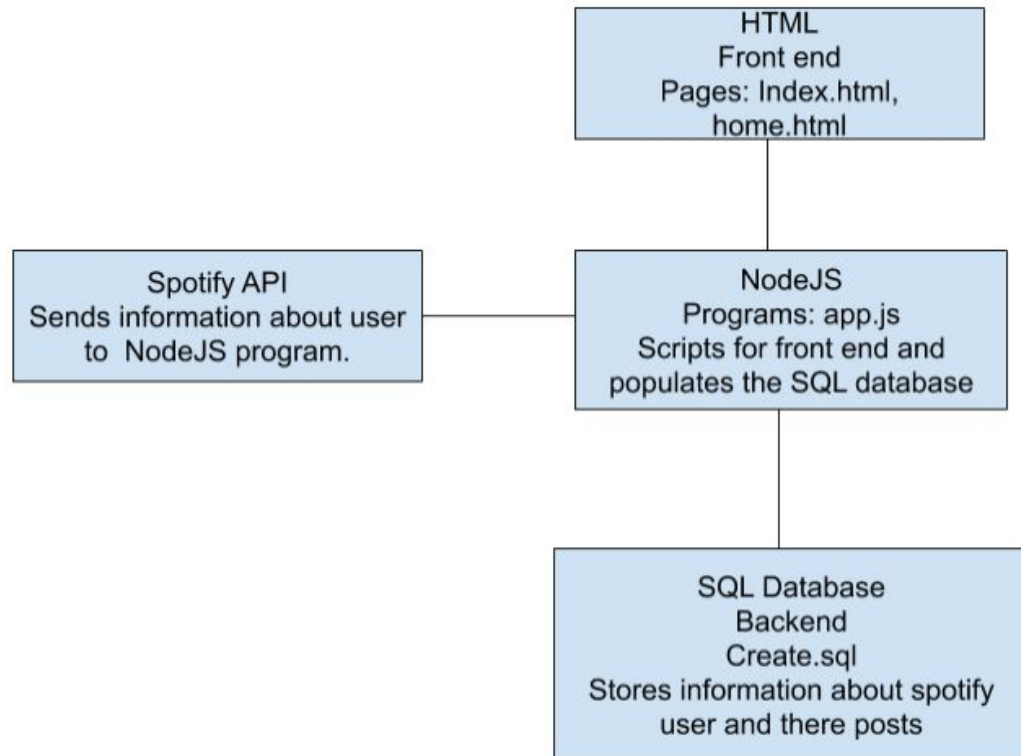
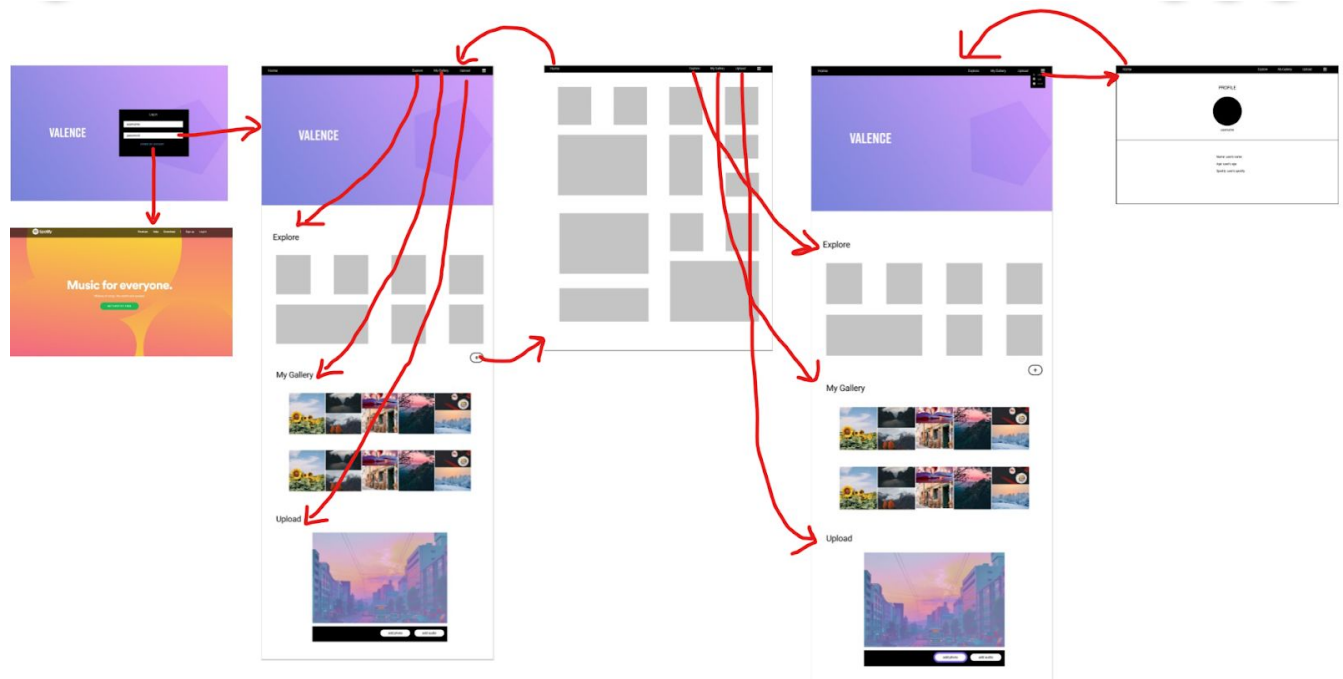
## **VALENCE**

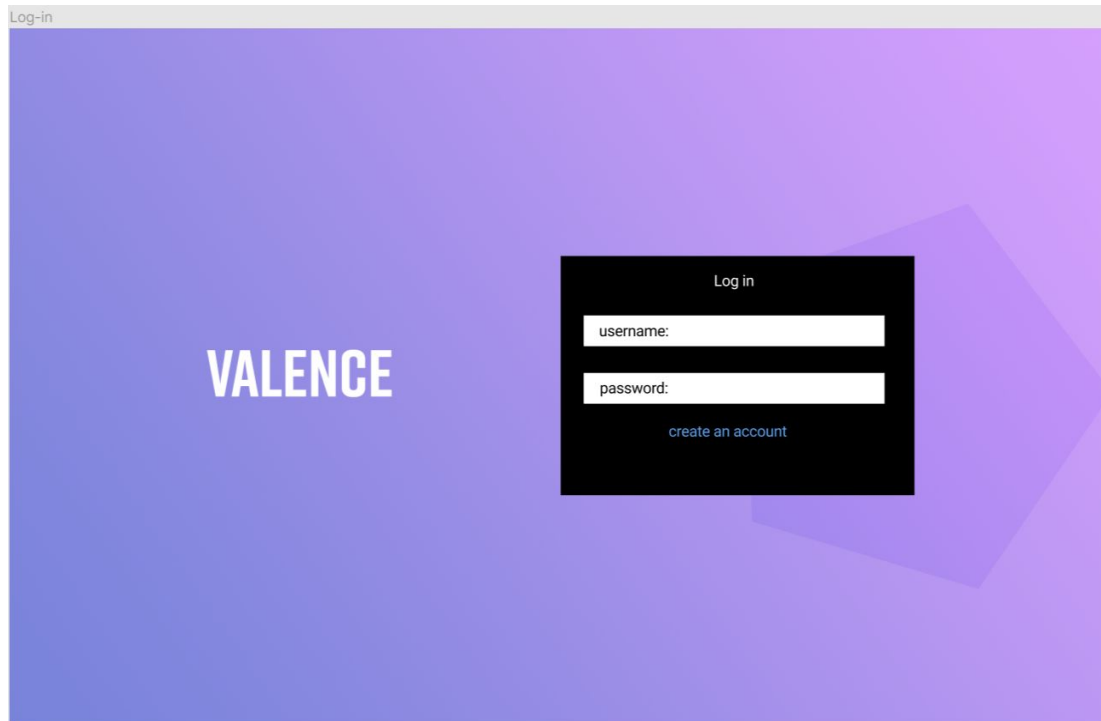
### **REVISED LIST OF FEATURES:**

- User Account/Profile
  - Software should be able to connect user accounts on spotify with ease.
  - The software should be secure and protected from unauthorized access. The data will be collected from the Spotify API.
  - There will also be a user account profile page on our website that shows their profile picture, name, etc.
- Upload a picture
  - This feature should allow the user to upload a picture from their system to put into their gallery.
  - This feature uses HTML upload and should store it in our database to be used for the gallery.
- Search
  - The search feature should be able to look for songs the user specifies.
  - This feature will use the database to search for songs using spotify's web service.
- Commenting
  - This feature should allow users to comment on other's photos.
  - This feature will fetch data from the database to store or delete comments.
- Add audio to picture
  - This feature should allow the user to select a song from spotify and have it linked to the photo they upload.
  - This feature will use the database to get the specified song from Spotify.
- Playlists
  - This feature should allow users to look at each other's playlists gathered from Spotify.

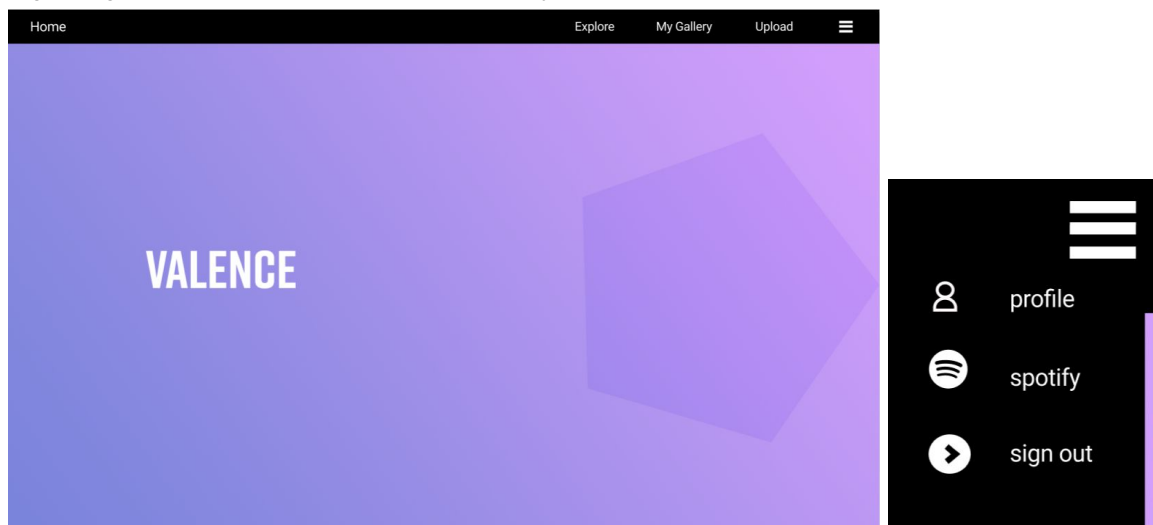
### **ARCHITECTURE DIAGRAM:**

Vananh Le, Amey Erdenebileg, Yosan Russom, Jack Kelly, Drishan Sarkar

**FRONT END DESIGN:**



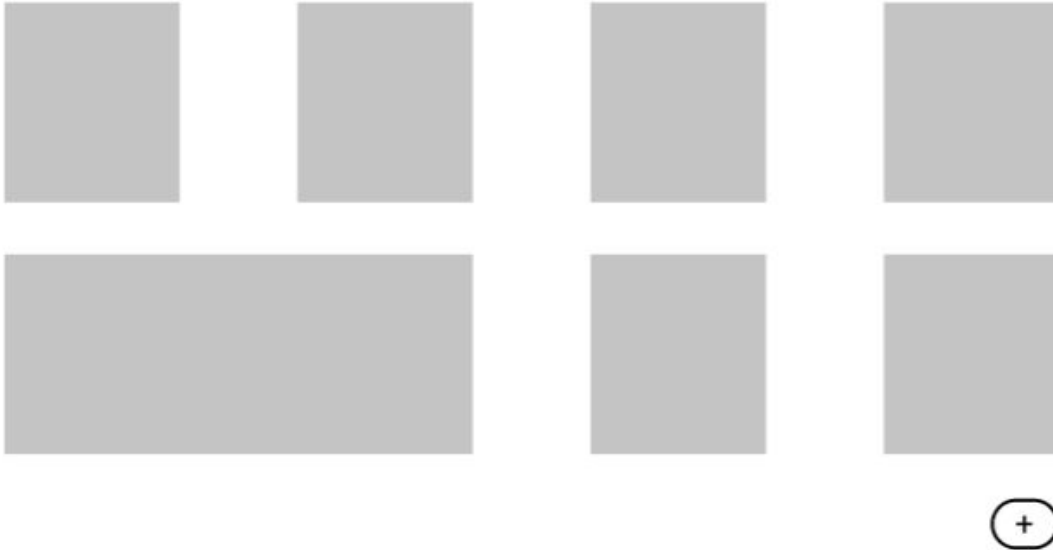
Log-in page. "Create an account" links to the spotify website.



Home page

Sidebar can direct users to the following: profile, spotify, or to log out..

## Explore

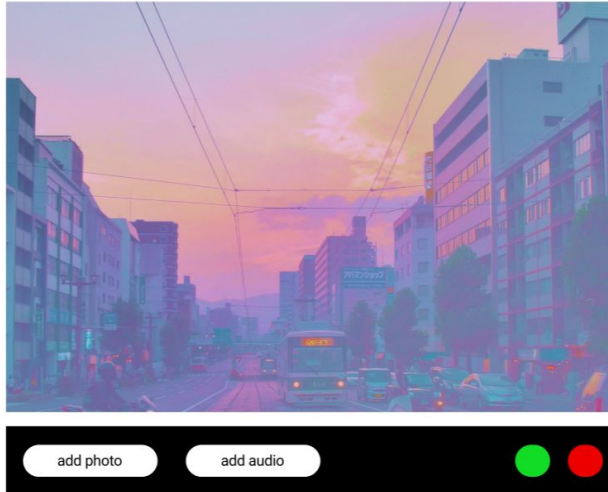


## My Gallery



Note: Every box within the explore and gallery section has an attached audio with it. To play, the user would simply hover over a box, to stop the user can simply click on that specific box that is playing.

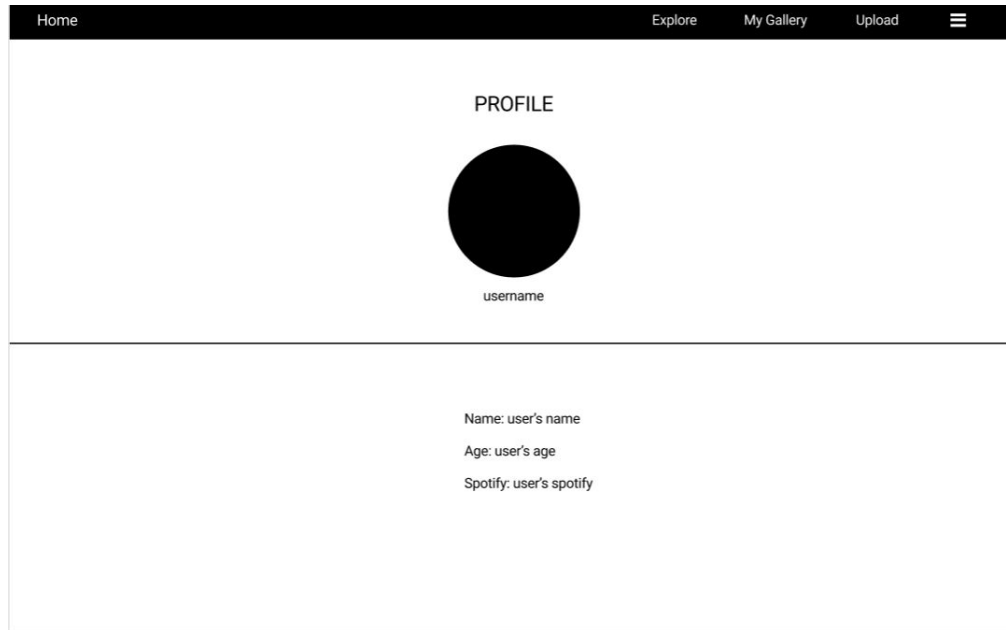
## Upload



Users can upload photos and add audio in their gallery. Green is to upload. Red is to cancel.



For the user to explore more of what other people are uploading.



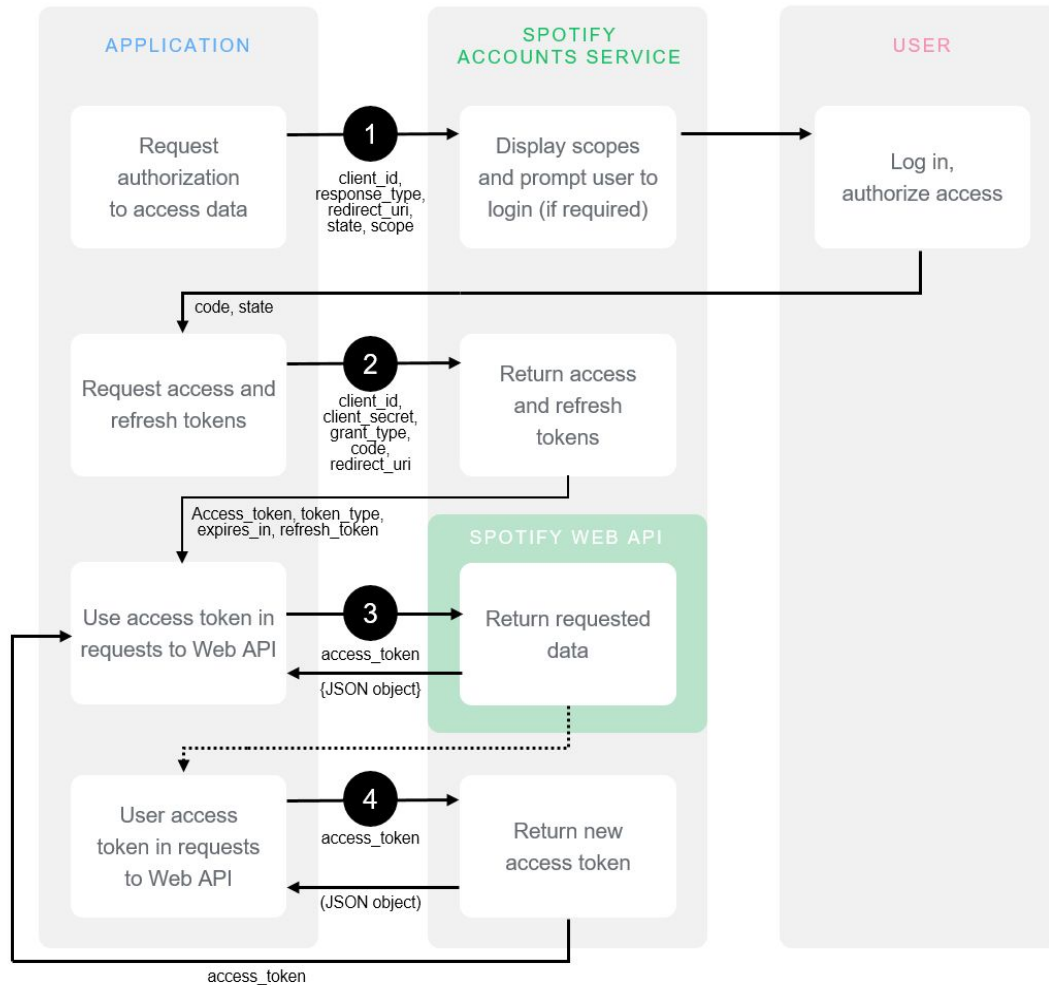
Profile page.

### **WEB SERVICE DESIGN:**

- For the web service, Spotify API has a really well-documented API for the project application we are working on. The idea is, when the program is running, the ID and secret will be passed in calls to the Spotify Account service. The Spotify Account service is the service that handles the authentication of the user and asks the user's permission to access data from their account. For the search feature, within the JavaScript, a call is being made by the search endpoint that passes the artist's name that the user entered and asks for any matching albums. Once the JSON result returned, it is analyzed to remove the links to the albums. A second set of calls is being made after that in order to get the album's main cover art image.
- For authorization, we need to get permission from the user in order to manipulate their data. We would have to use the OAuth in order to get the data we need. We also need to get the access token in order for the program to pass in calls to the Web API. To accomplish that, we registered our application to the Spotify Developer websites and obtained the client ID and client secret key that will be generated and displayed in the application detail page. While the application is running, the ID and secret will be passed in calls to the Spotify Accounts service. This handles the authentication of the user and asks the user's permission for

Vananh Le, Amey Erdenebileg, Yosan Russom, Jack Kelly, Drishan Sarkar

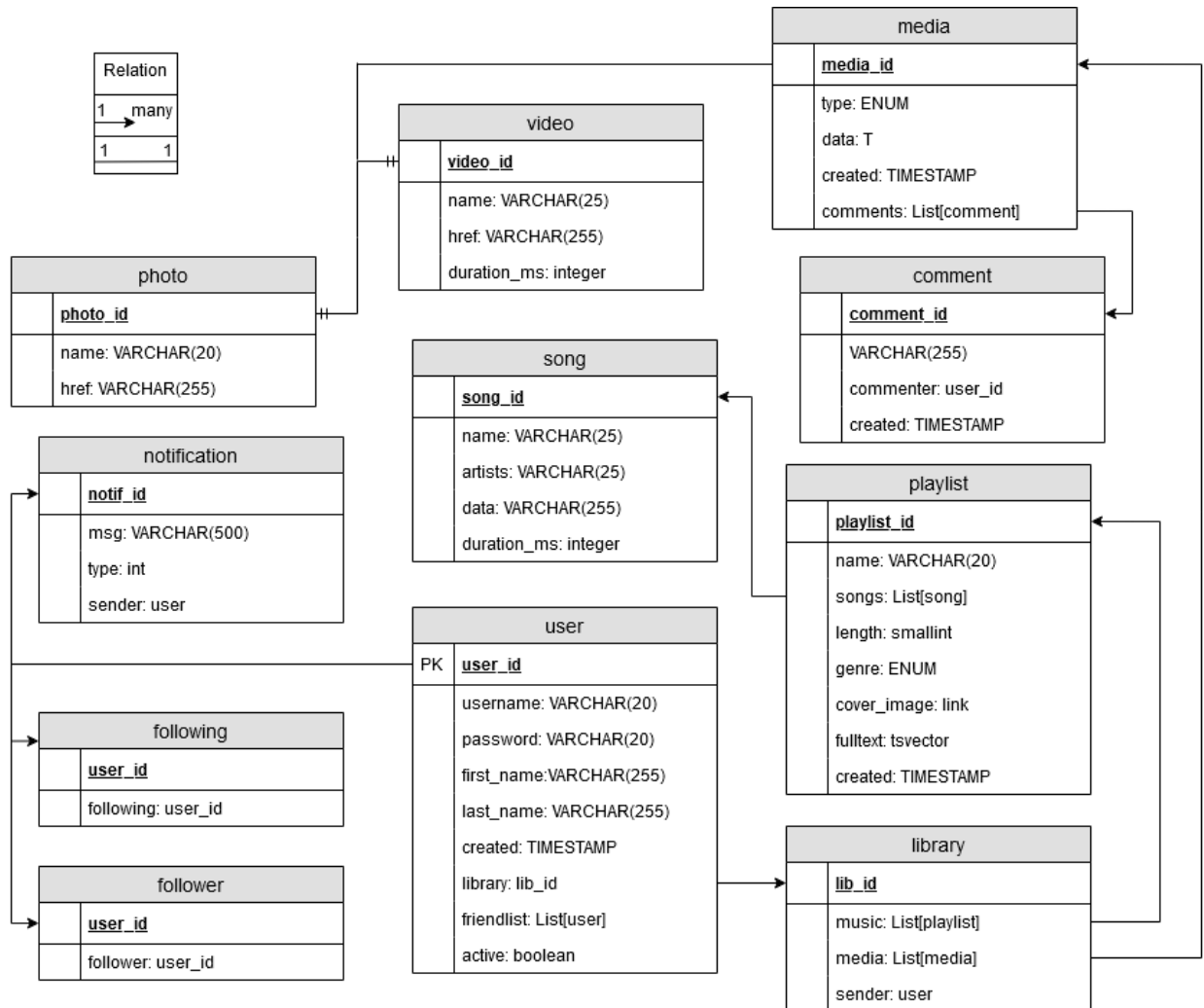
data. For our application, we are planning to use Heroku to store the client ID and secret for security reason. This method just helps the user access the program without manually entering the client ID and secret.



Vananh Le, Amey Erdenebileg, Yosan Russom, Jack Kelly, Drishan Sarkar

## DATABASE DESIGN:

- PostgreSQL is the primary DBMS technology we use for this project
- Entity Relationship Diagram:



- We use GET requests to fetch song and playlist data from spotify's web service
- NoSQL/MongoDB may be used to store the user's friends as a graph