

# **SENG2021**

## **MOSH**

**A social media platform focussed  
on connecting people through music**

**BY: team\_aace  
Austin Vuong, Alvin Cheng, Cuthbert Zhong,  
Elliott Yu, Luke Elsabbagh**

## **Purpose / Problem Statement**

In today's era of multimedia entertainment and streaming services, music platforms such as Spotify and YouTube have provided an easily accessible way for users to listen to their favourite tracks and artists. Each of these platforms has a set of associated advantages and disadvantages which we will explore in the following section.

### **Spotify**

#### **Advantages**

- Large range of music and artists to choose from
- Provides recommendations of playlists and tracks to users
- Can follow artists and create personal playlists
- Can listen to the radio
- Free service

#### **Disadvantages**

- Limited social media functionality between users
  - Unable to share posts, messages or music with other users
- Can only view user's other playlists and followed artists

### **Youtube**

#### **Advantages**

- Large range of both official and unofficial music and artists to choose from
- Provides recommendations of playlists and videos to users
- Free service
- Videos on the platform not limited to just music related ones

#### **Disadvantages**

- No functionality of friends or direct interaction between users
- Scattered sense of community through only comments across videos
- Recommendations are not consistently focussed on music even when listening to music videos

As shown through the advantages and disadvantages of each platform, the lack of social networking and interaction is an issue that is prevalent in music streaming platforms. Whilst some platforms such as Spotify provide limited functionality for interaction between users, there is no fully-fledged media platform that implements a holistic range of features expected from a social media platform such as the ability to add friends and interact (message / share posts) with other users. Additionally, due to the intrinsic competition between music streaming platforms, it is often difficult to share playlists and tracks from one platform to another. It is this lack of interactivity between and within music streaming platforms that Mosh attempts to address.

## **Key Features**

With Mosh's unique purpose and the endless possibilities provided by the streaming service industry whether it is through the data they collect or the number of users they attract, Mosh is able to deliver innovative features that have the potential to create a more holistic and interactive music community online.

### **Key Features:**

- A social media experience created through posts, communities and connections.
- A platform that centralises a user's music recommendations enabling users to discover new music and people with similar taste.
- A personalised experience that continues to evolve as a user's music taste changes.
- Communities growing through similar music taste and favourite artists.
- Ability to play and easily share music they discover across streaming service platforms, encouraging shared experiences through music.

## User Stories

**Requirement :** A social media platform that is catered towards music interests.

**Feature:** Sign Up

**As a user,**  
**So that** my data can persist on the platform  
**I want to** be able to sign up.

**GIVEN** I am on the landing page  
**AND** I do not already have an account  
**WHEN** I click "sign up"  
**THEN** I am shown a sign up page where I can enter my account details  
**WHEN** I click "sign up"  
**THEN** an account and profile are created  
**AND** I am directed to my homepage.

**Feature:** Log In

**As a user,**  
**So that** I can access my profile and account  
**I want to** be able to log into the platform.

**GIVEN** I am on the landing page  
**AND** I have an account  
**WHEN** I click "sign in"  
**THEN** I am shown a login page  
**WHEN** I enter my valid account details  
**AND** press enter  
**THEN** I should be able to access the service  
**AND** my homepage is shown.

**Feature:** Log out

**As a user,**  
**So that** my account remains secure from outside access  
**I want to** be able to log out of my account.

**GIVEN** that I am logged into my account  
**WHEN** I am on any page  
**THEN** I should be able to click "log out"  
**AND** log out of my account

**Feature:** Edit Account Settings

**As a user,**

**So that** my account can better fit my needs

**I want to** be able to edit my account settings.

**GIVEN** I am on my profile page

**WHEN** I click “edit profile settings”

**THEN** I am shown a page of settings for my profile

**GIVEN** I am on my profile settings page

**WHEN** I click “update avatar”

**THEN** I can either use a preregistered avatar or upload my own image so that

**WHEN** I click ‘update’

**THEN** my avatar is changed to that image.

**GIVEN** that I am on the “Profile Settings page”

**WHEN** I press the “deactivate account button”

**AND** press “yes” on the are you sure prompt

**THEN** my account should be deleted from the platform

**Feature:** Connect with streaming services

**As a user,**

**So that** I can have a more personalised experience,

**I want to** be able to connect to streaming services that I use

**GIVEN** that I am signing up

**THEN** I am shown a list of possible streaming services to connect to

**WHEN** I choose a streaming service

**THEN** I should be able to login to that streaming service

**AND** my account will become linked to my streaming service account

**AND** my data from the streaming service platform will be loaded into my profile.

**Requirement:** Recommend people, music and other features to me based on my personal preferences.

**Feature:** Discover people

**As** a user,

**So that** I can meet people of similar interests

**I want to** be recommended their profiles

**GIVEN** that I am on my home page

**WHEN** I click the “discover” tab

**THEN** I should be on the discover page

**WHEN** I click the ‘people’ side tab

**THEN** I am shown a list of people recommended to connect with based on our mutual interests.

**Feature:** Discover Music

**As** a user,

**So that** I can broaden my horizons

**I want to** be able to discover new music recommendations

**GIVEN** I have a streaming service account linked to my profile

**AND** I am on the discover tab

**WHEN** I click on the music tab on the side bar

**THEN** I am shown a list of recommended tracks on the discover page

**WHEN** I click a track

**THEN** I am redirected to a media platform that plays the selected media

**Requirement:** Ability to interact with other users of the social media platform

**Feature:** Connect with other users

**As a user,**

**So that** I can build up my social network,

**I want to** be able to connect with other users of the social media platform.

**GIVEN** I am on the the “discover” page

**AND** have selected the “people” sidebar option

**WHEN** I click on a user’s profile

**THEN** I am shown that user’s profile

**WHEN** I click on the connect button

**THEN** I am able to add that user to my connections.

**Feature:** View existing connections

**As a user,**

**So that** I can keep track of my social network,

**I want to** be able to view my existing connections with other users.

**GIVEN** I am on the “discover” page,

**WHEN** I click on the “your connections” sidebar button

**THEN** I am shown a list of existing connections with other users.

**Feature:** Remove connections

**As a user,**

**So that** I can modify my social network

**I want to** be able to disconnect with users that I have connected with in the past.

**GIVEN** I am on another user’s profile page

**AND** I am already connected with them

**WHEN** I click the disconnect button

**THEN** I am no longer connected with them.

**Feature:** Share thoughts

**As a user,**

**So that** others can be updated on thoughts/opinions of music,  
**I want to** be able to share my thoughts on the feed.

**GIVEN** I am on my homepage,

**THEN** I am shown a feed of other users' posts

**WHERE** I can filter relevant posts by tag

**OR** view my own posts.

**GIVEN** that I am on the homepage

**WHEN** I click "post"

**THEN** I should be able to post an update

**AND** attach relevant tags (eg. artists that the post is about)

**GIVEN** that I am on the homepage

**WHEN** I click the comment icon on a post

**THEN** I am shown a list of comments for that post

**AND** should be able to post new comments

**Feature:** Earn Badges

**As a user,**

**So that** I can have a more personalised profile,  
**I want to** be able to earn badges by completing achievements

**GIVEN** I am on my profile page,

**THEN** I can view my current badge progress

**WHEN** I click the badge

**THEN** I can view my progress



**Requirement:** Provide an easy cross-platform music sharing experience not provided by social media and streaming service platforms

**Feature:** Map music tracks

**As a user,**  
**So that** I can more easily listen to music  
**I want to** be able to map music tracks from one platform to another.

**GIVEN** I am logged in  
**AND** have selected a music track  
**WHEN** I click on that track  
**THEN** I am shown links to open that track on various media platforms.

**Feature:** Send music tracks to other users

**As a user,**  
**So that** I can share my interests with other people  
**I want to** be able to send music tracks to other users.

**GIVEN** I am logged in  
**AND** on the “share music” tab  
**WHEN** I type in a song  
**AND** I click the “search” button  
**THEN** I am able to specify a music track  
**AND** send it to a particular user who I have connected with.

**Feature:** Receive music tracks from other users

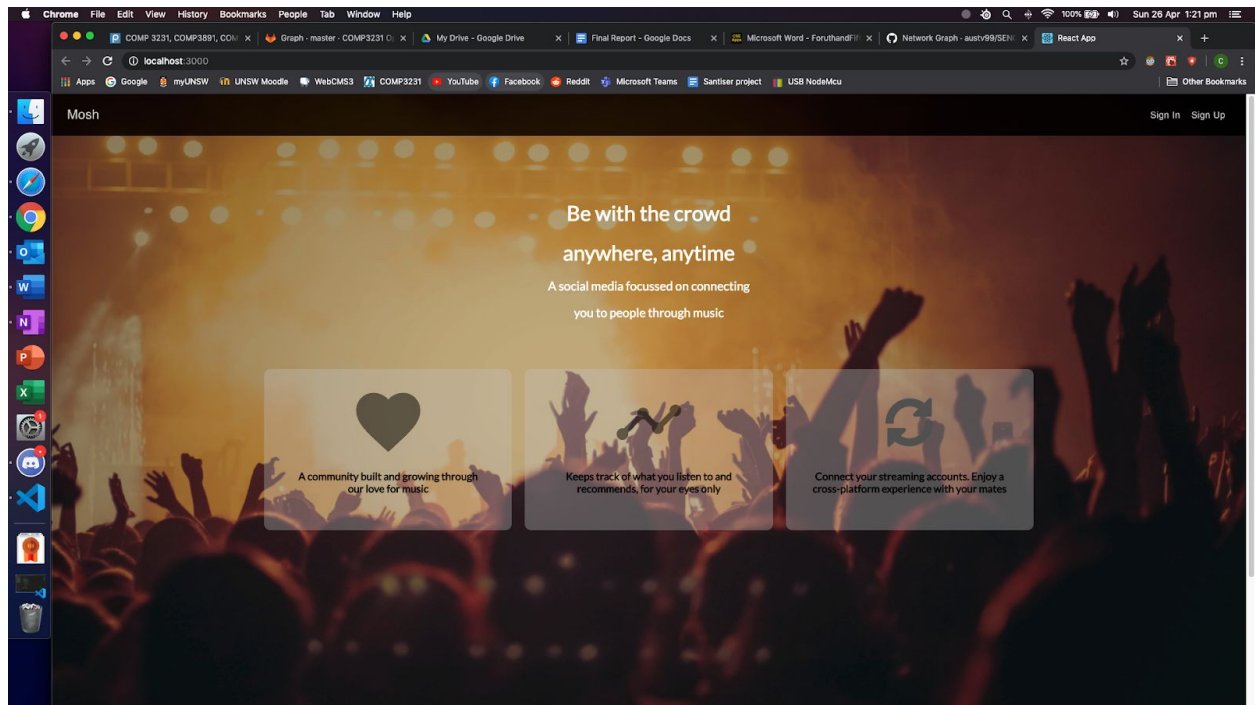
**As a user,**  
**So that** I can experience what others are listening to  
**I want to** be able to receive music tracks sent to me by other users.

**GIVEN** I am on the “discover” page  
**WHEN** I click on the “Shared with me” page  
**THEN** I am shown a list of music tracks sent to me by other users.

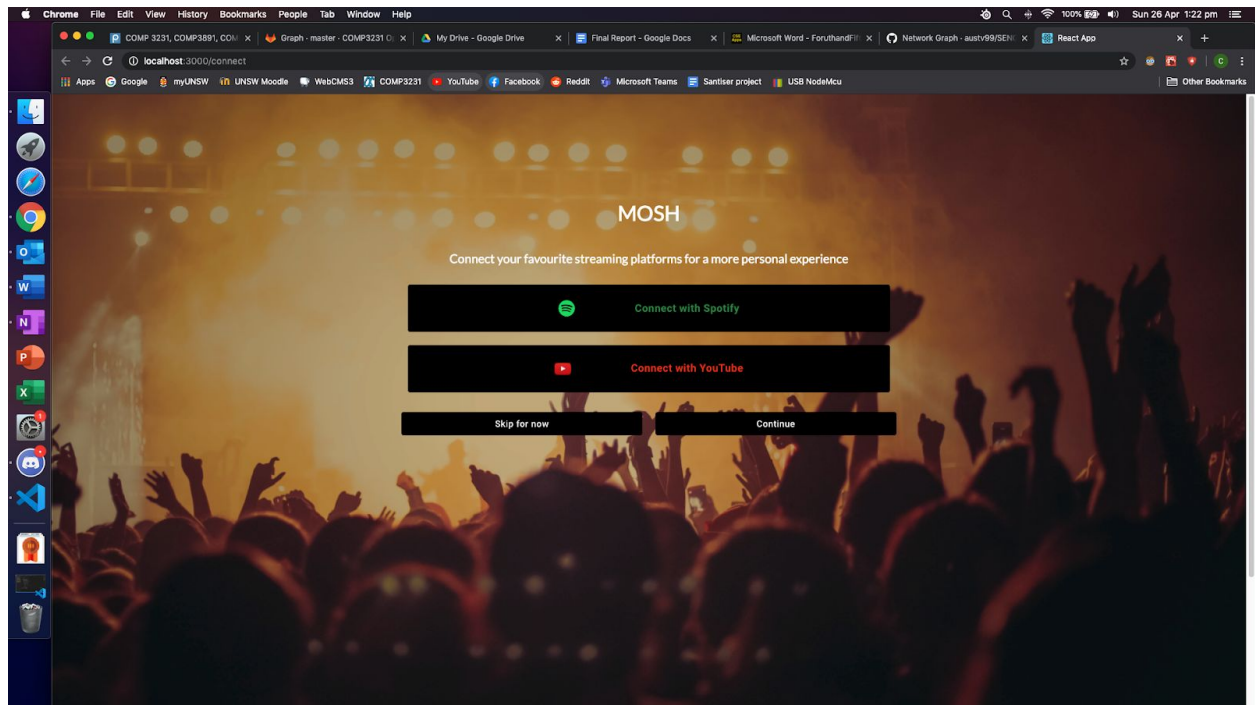
**GIVEN** I have tracks that have been shared with me  
**WHEN** I click on a track  
**THEN** I am able to play the track on any platform that I desire.

## Final Interface Screenshots

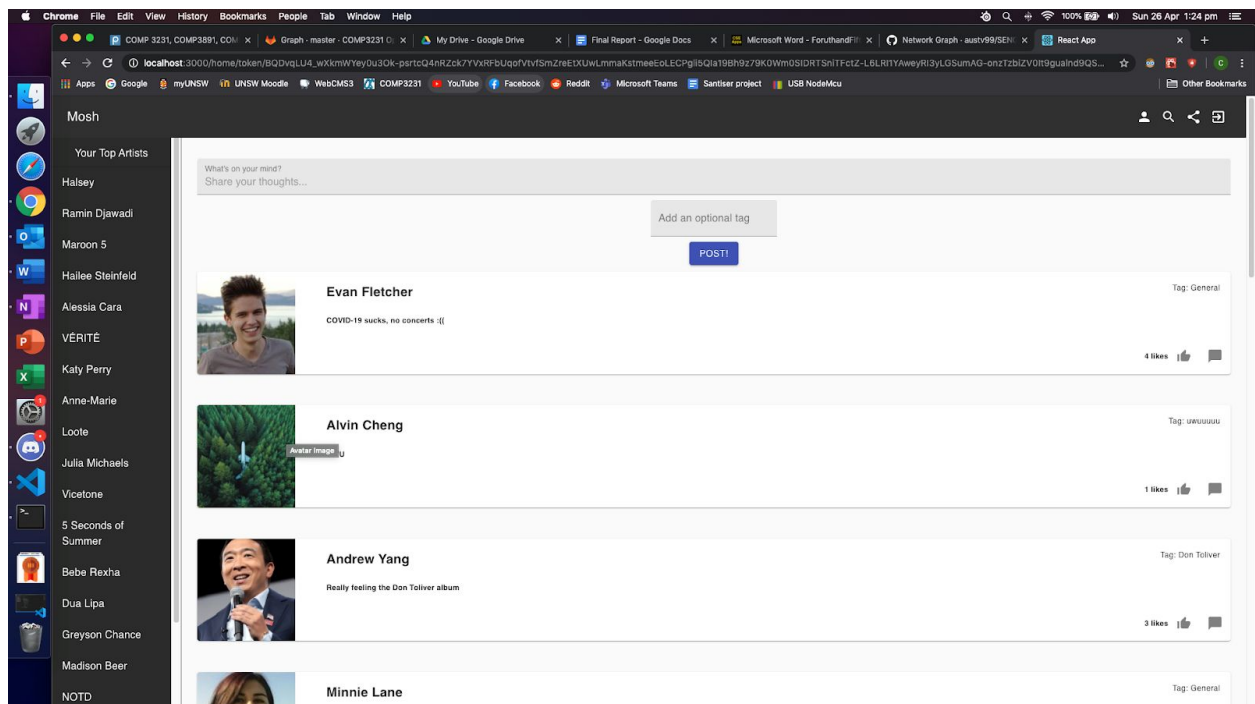
### Landing Page



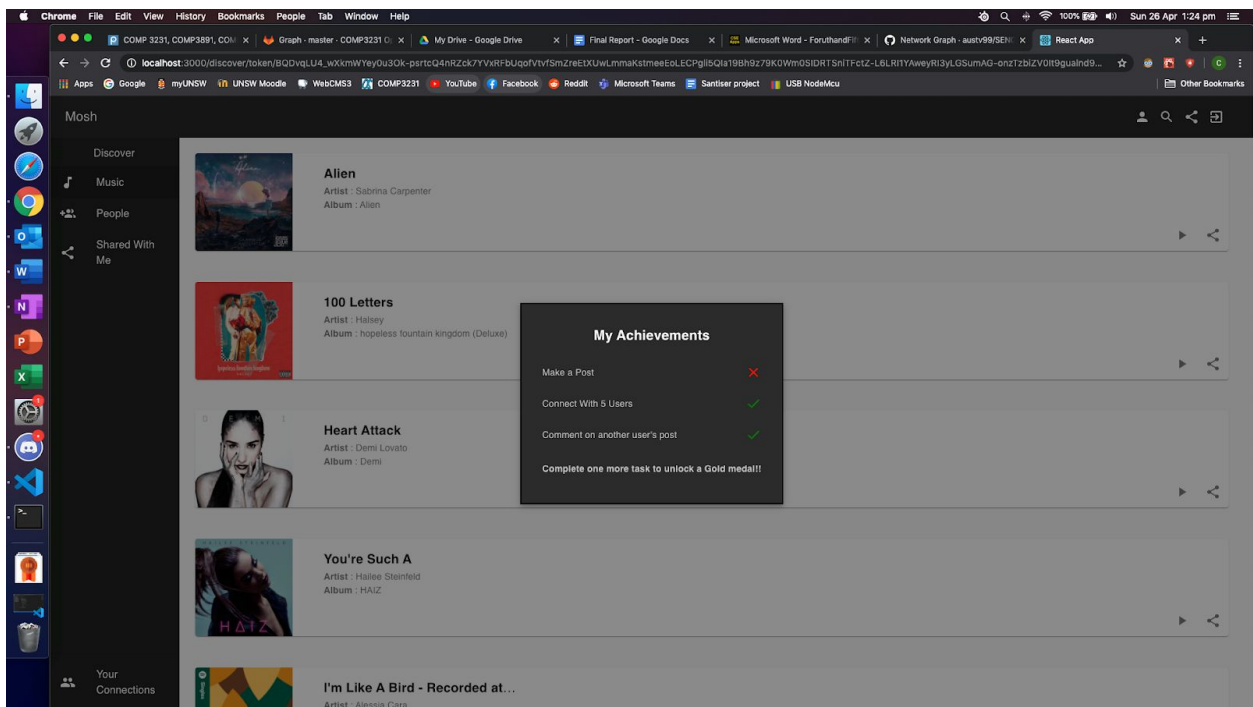
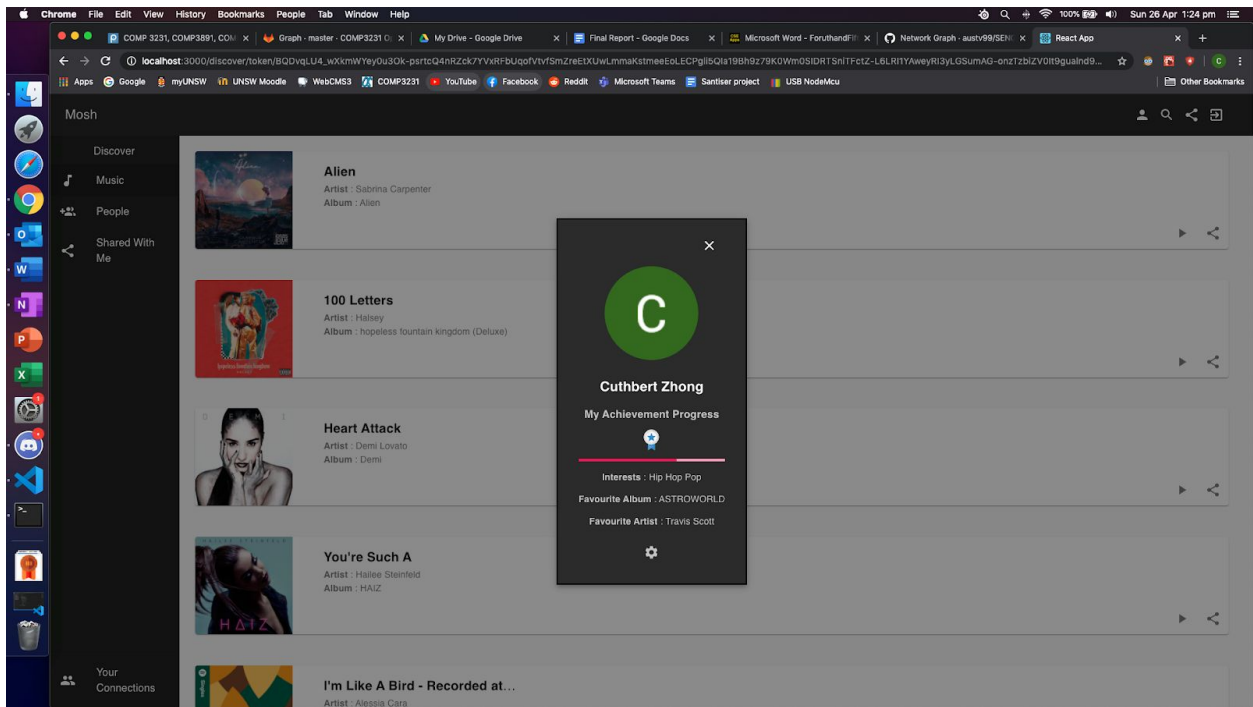
## Connect Page



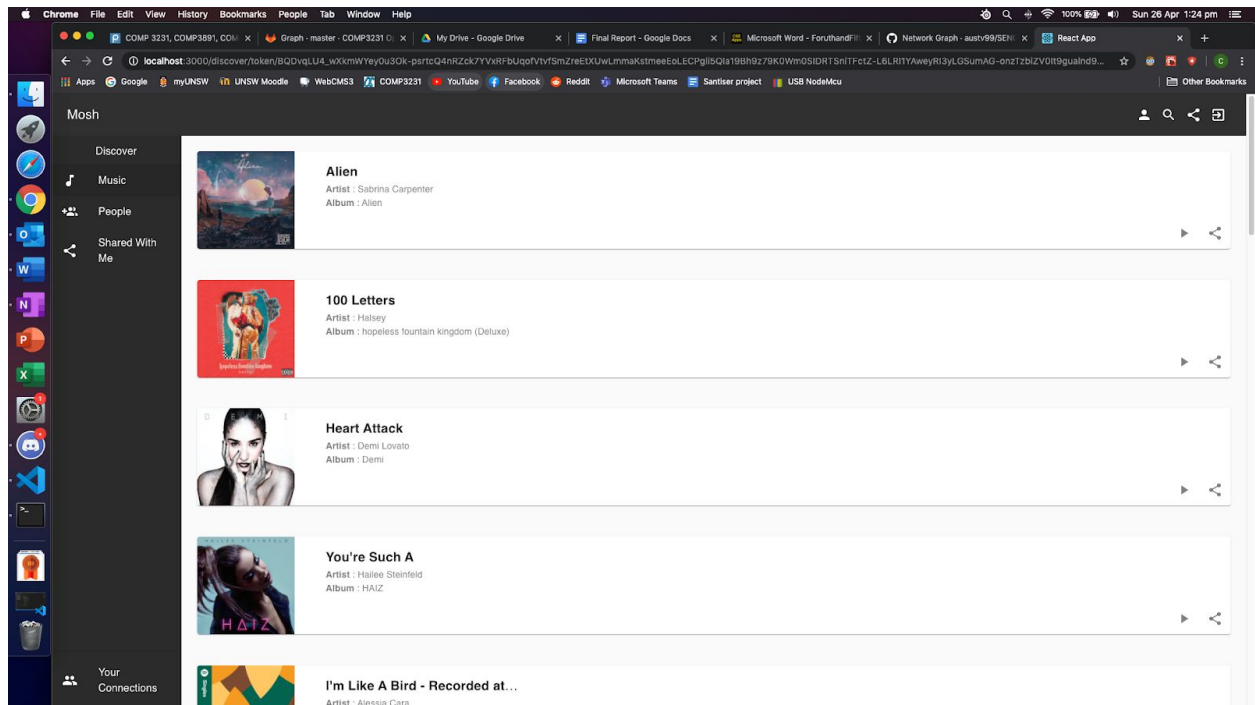
## User Feed (Home) Page



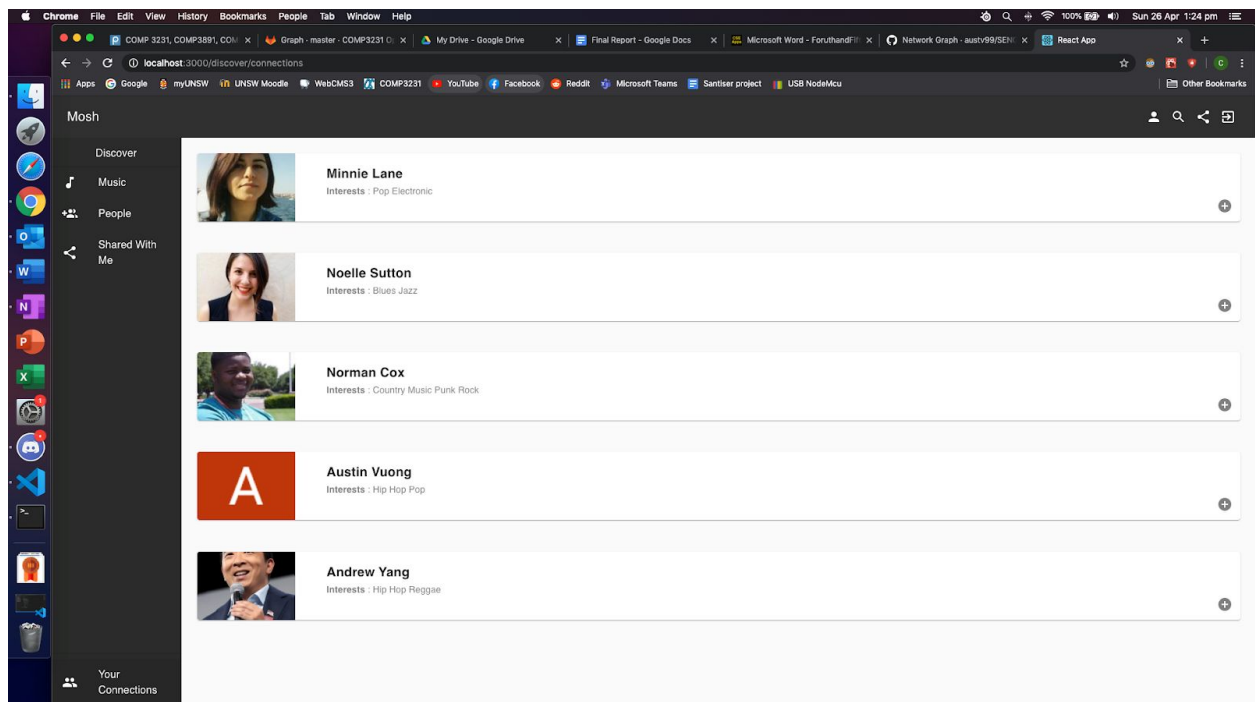
# User Profile



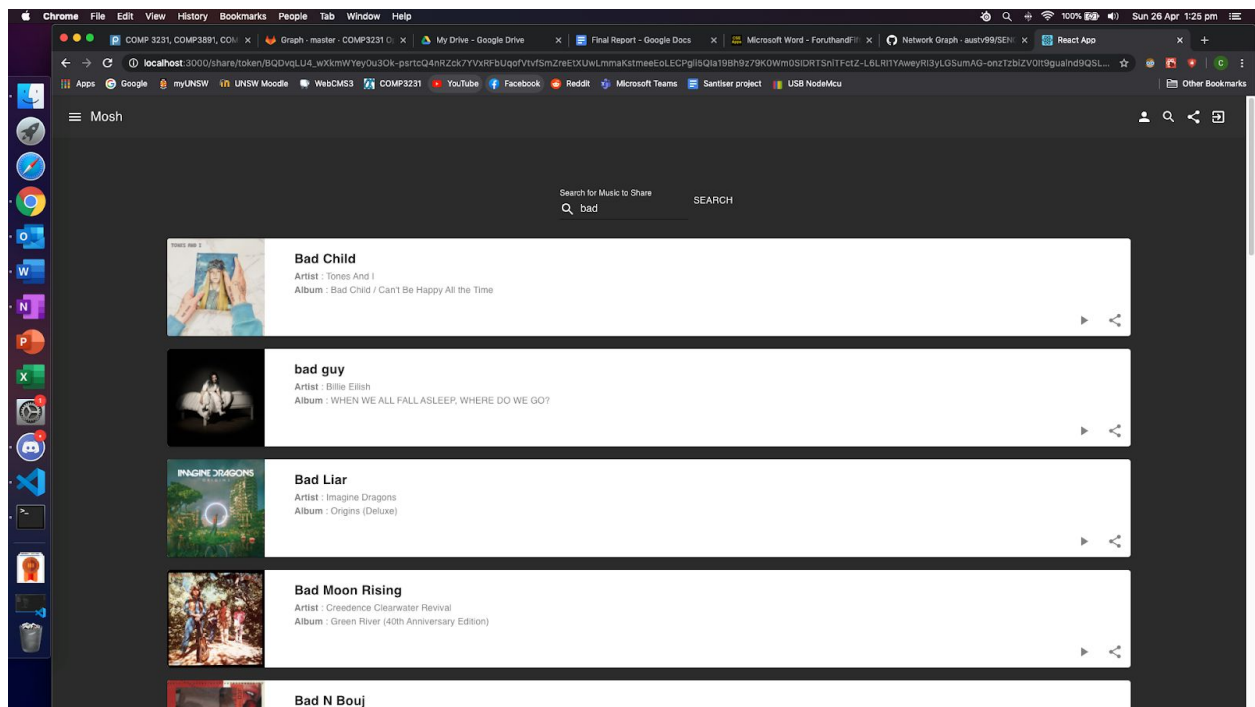
## Discover Music Page



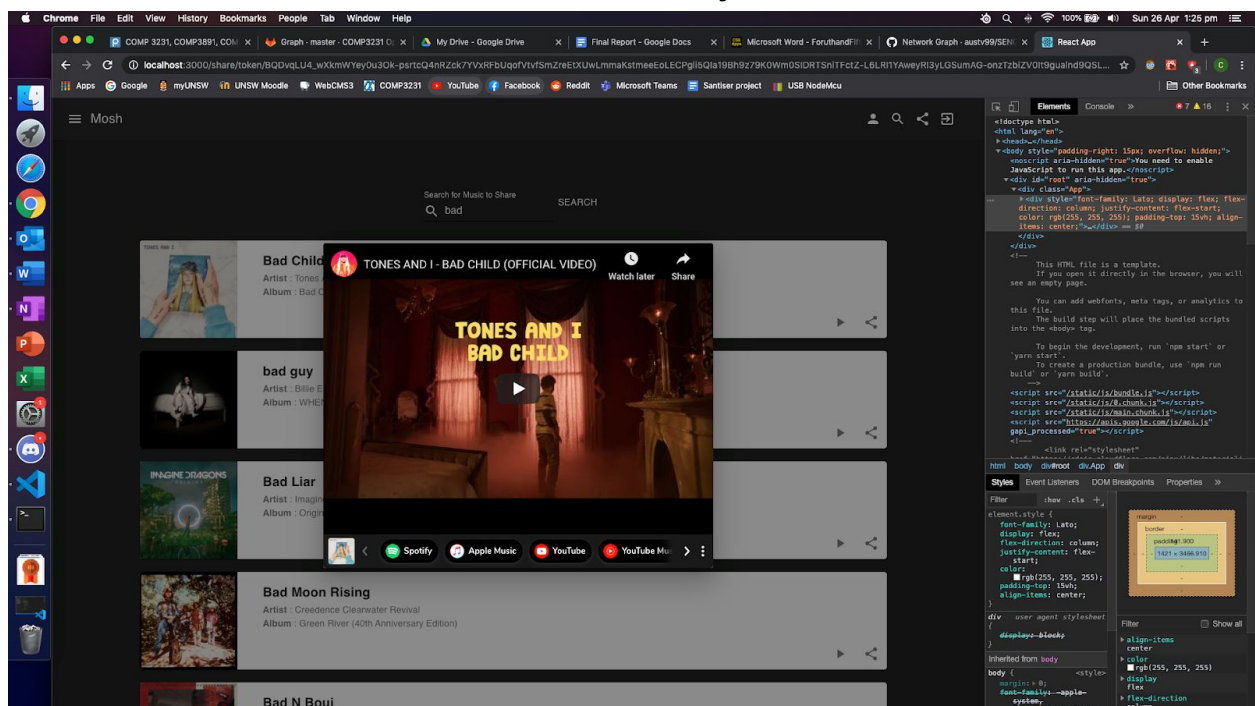
## Discover People/ Your Connections Page



## Share Page

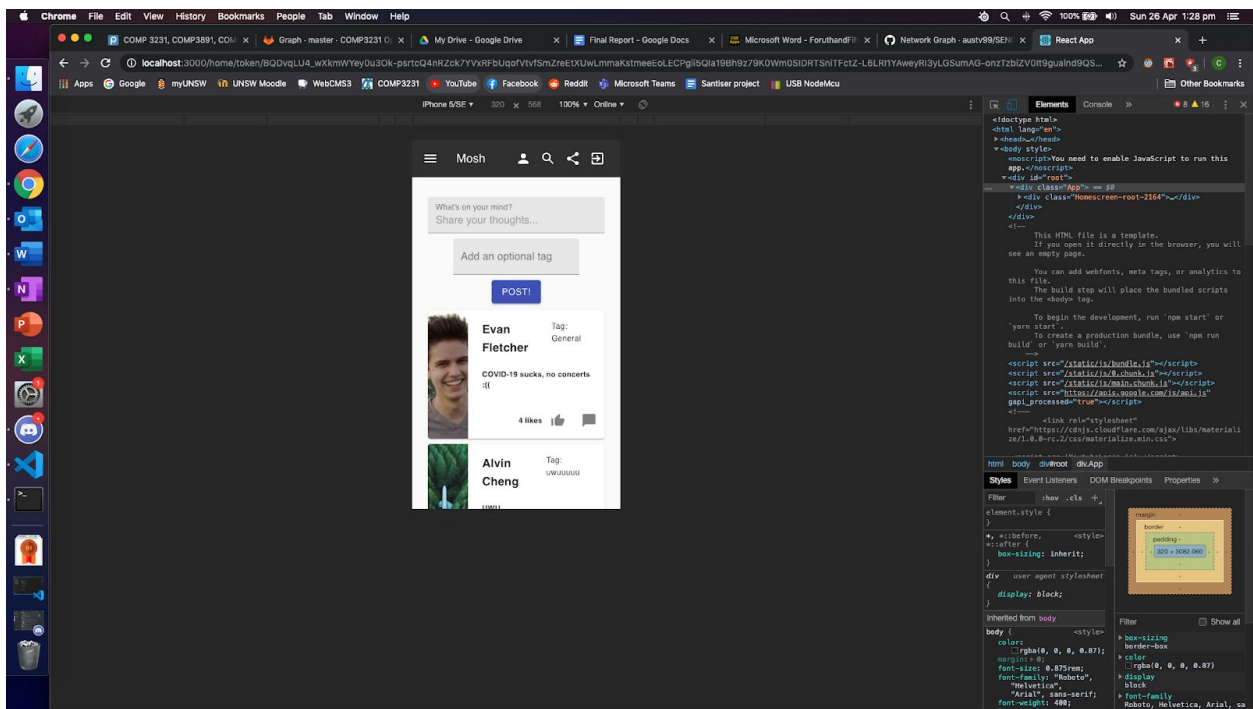
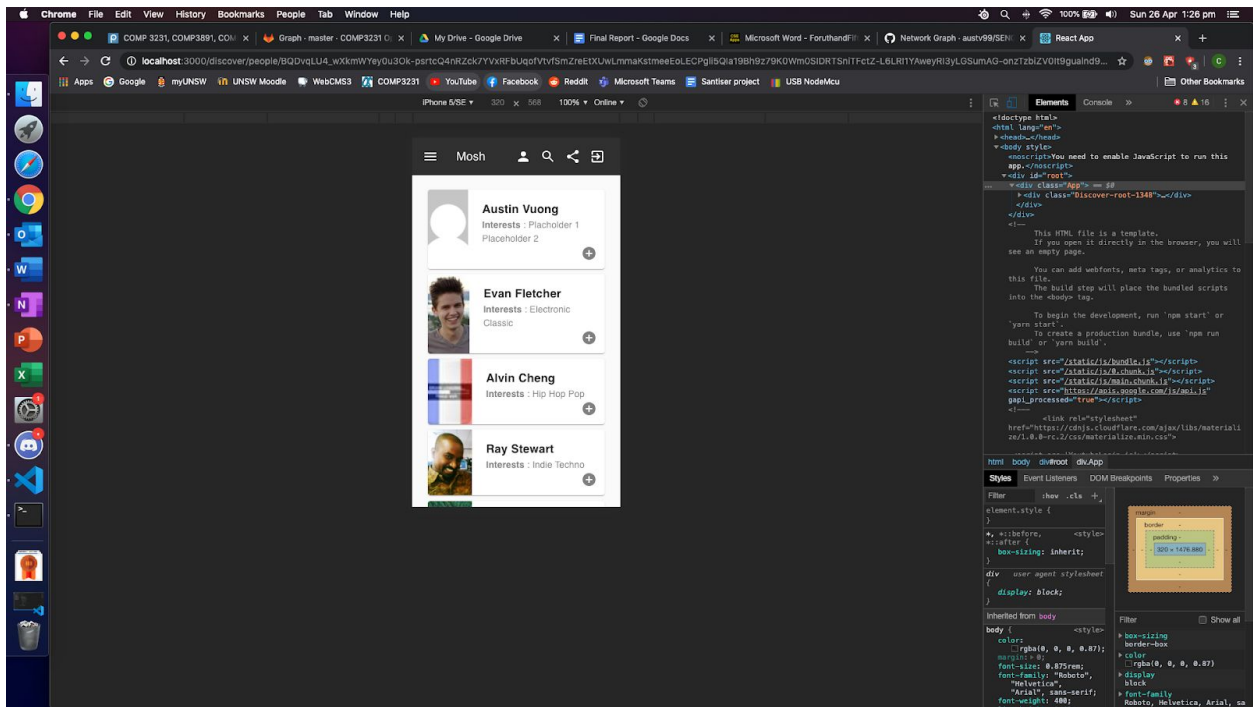


## Music Media Player



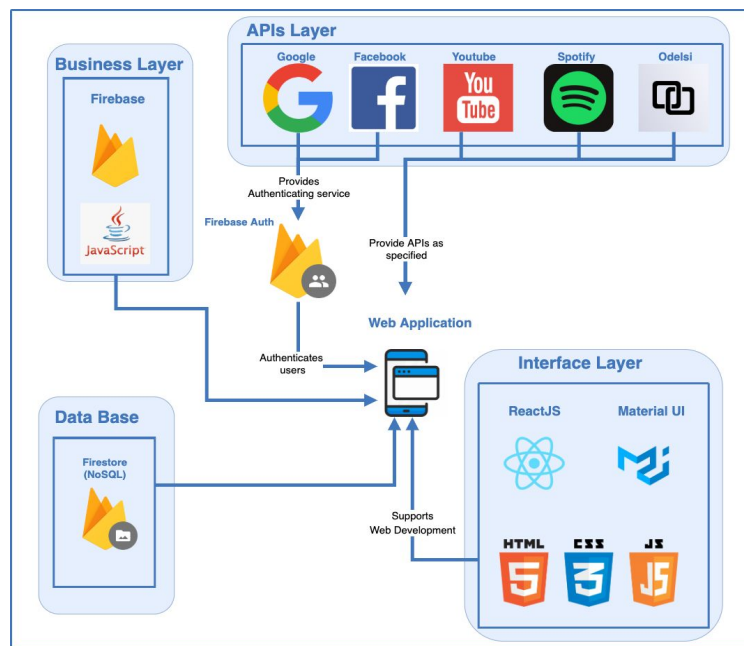


# Mobile Responsivity



# Design and Implementation

## Software Architecture



### Interface Layer

HTML is the standard markup language in web browsers. Web browsers can receive HTML documents from a web server and render such documents. It is supported by technologies such as CSS for styling as well as JavaScript for scripting and the creation of interactive web pages.

ReactJS is a modern JavaScript framework ideal for web application development, enabling for efficient and effective design and development. Its use of class based components allows for ease of reusability between different components. Additionally, ReactJS allows for effective interaction with the web application through the maintaining of state and dynamic updates to the DOM.

Material-UI is a user interface library built with ReactJS, which provides a variety of pre-rendered components such as lists, drawers, application bars and grids. This allows for efficient web development through popular user interface designs provided through the library. Additionally, Material-UI provides CSS and Javascript support for responsive web pages, allowing for our application to be easily accessible across a range of devices.

Due to the popularity of JavaScript and HTML in web development, our technology stack allows for Mosh to run on a variety of web browsers such as Google Chrome and Safari on virtually any operating system. Through the use of web development frameworks such as Material-UI, our application will support a variety of devices from desktop to mobile browsers.



### Business Layer

Firebase, a technology developed by Google, largely provides the basis for our business layer. Through the use of various features provided by Firebase, such as Firestore, Authentication and integration of single sign ons, Firebase provides a secure, scalable and effective basis for web application development.

Firestore provides a NoSQL database infrastructure. This allows for efficient storing and retrieving of data through the use of JSON documents. This allows for more efficient parsing and processing of data. Further, NoSQL databases have the flexibility to change over time. This is useful in the cases where system changes occur, allowing for changes to be made without a lot of refactoring.

Specifically, Firestore was chosen due to its ease of use and quick learning curve. Predominantly accessed/configured through a GUI and used through calls from the front-end, it allows for the database infrastructure to be developed and running quickly in order to support other functionalities of the project at an early stage.

Also, Firestore provides automatic and robust scaling for queries, performance, storage and synchronisation. This is ideal in this project as it allows for greater focus on functionality and user experience rather than focus on the intricacies of the database infrastructure throughout development.

Authentication and single sign on support provided by Firebase allows for the development of a secure user account management system. This allows for users to sign up and sign in into their accounts using email/password, Facebook and Google single sign ons. The effective user account management system also supports database management through the attribution of posts, comments, likes and music sharing to respective users. The support for single sign ons is significant in providing more ways for users to sign up to the platform whilst also contributing to the personalised experience of the platform.

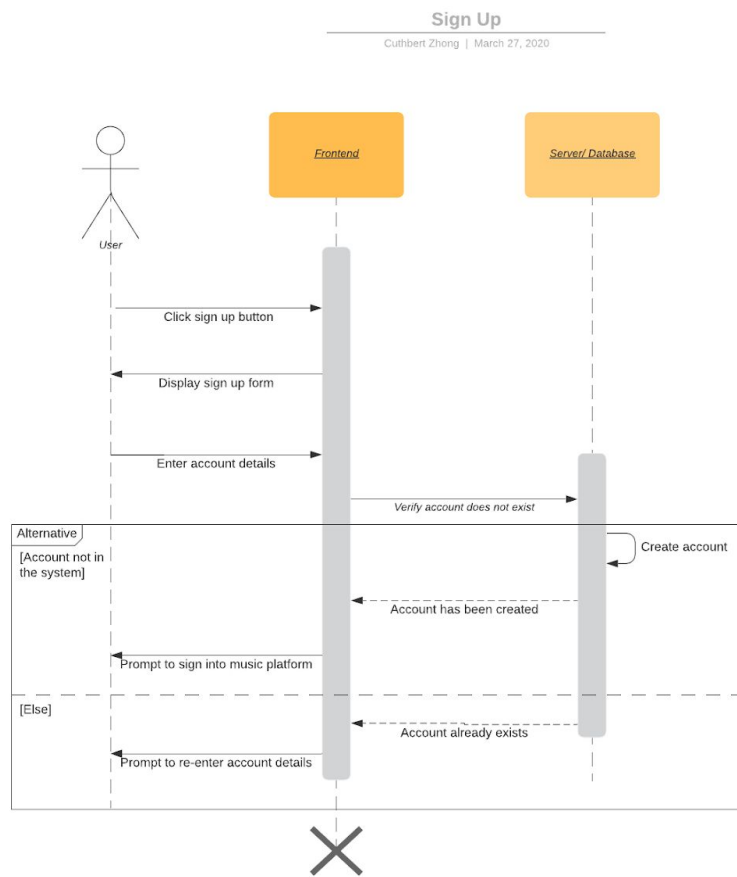
## API Layer

- Google/Facebook
  - The single sign on process supported by our Google and Facebook options provided on the platform streamlines the authentication process.
  - The ease of convenience of “one-click” sign up/in will bring more traffic to the platform as well as a more personalised experience from integration with their existing accounts.
- Spotify
  - Spotify’s expansive API allows for access to a user’s personal Spotify data in order to create a more enriching personal experience on the platform.
  - Data such as top artists, listening history, track searching and recommendations are deeply integrated into the experience of the platform, supporting features that allow users to discover communities through similar music taste and cross-platform music sharing.
- Youtube
  - Similar to Spotify’s API, Youtube allows for access to personal Youtube data to contextualise a user’s experience within the platform.
  - Data used from their liked videos will allow the platform to understand their music listening habits and consequently recommend music to discover.
- Odelsi
  - Odelsi’s API supports the effort to create a cross-platform music sharing experience.
  - Odelsi’s mapping of popular music tracks between various streaming platforms will allow users to share the music they discover with their connections regardless of whether they use the same streaming service or not.

## Sequence/Interaction Diagrams

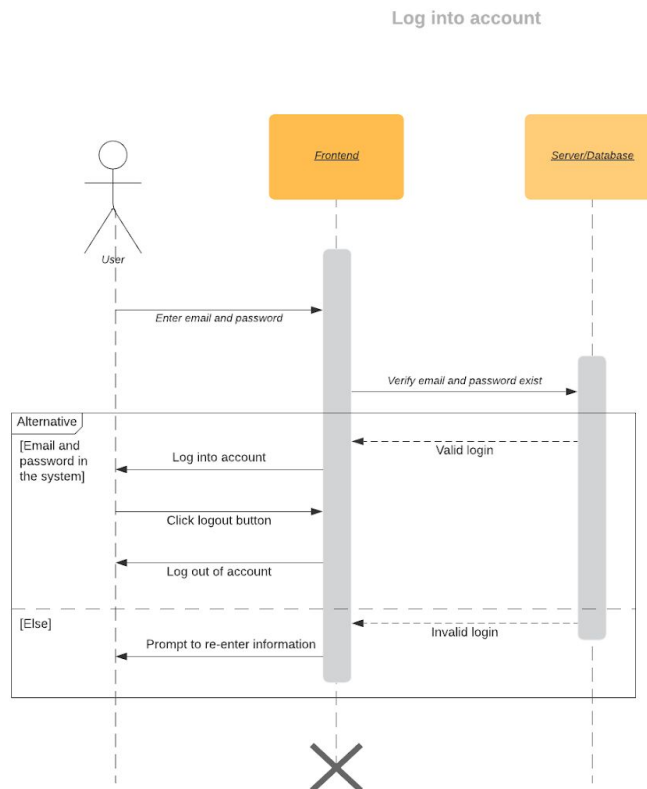
### Sign Up

- A user without an account clicks sign up
- They enter the email and password they want to use for their account
- The server will then check that the details the user is trying to make an account with is valid and an account doesn't already exist with that information



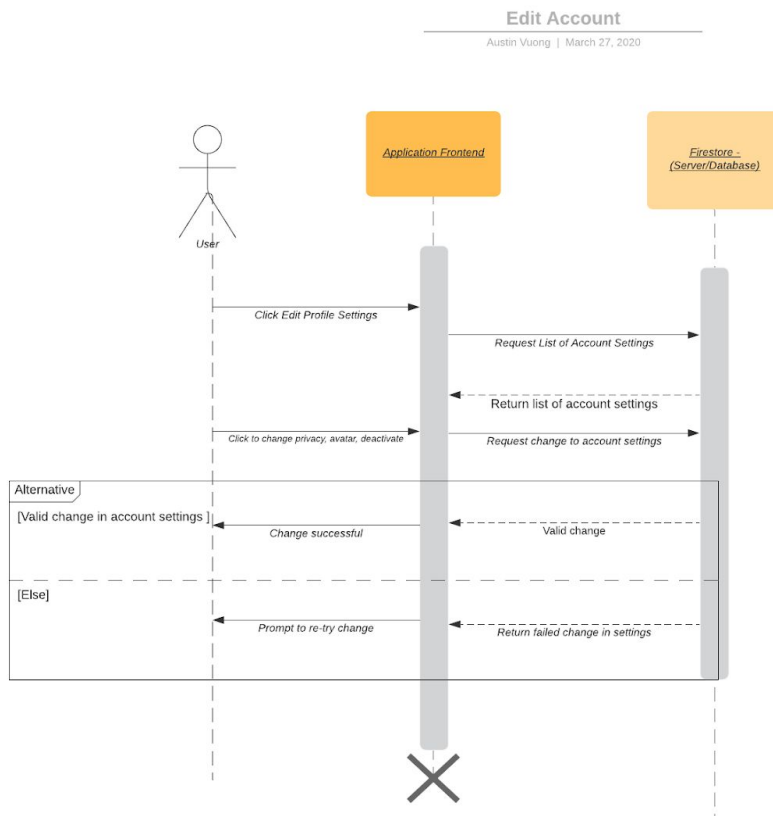
## Log In/Out

- A user on the login screen will enter the details to their account
- The server will then ensure the login is valid for an existing account
  - If the login is valid the user will be logged in
    - The user can then log out of their account
  - If the login is invalid the user will be informed and will be prompted to re-enter their details



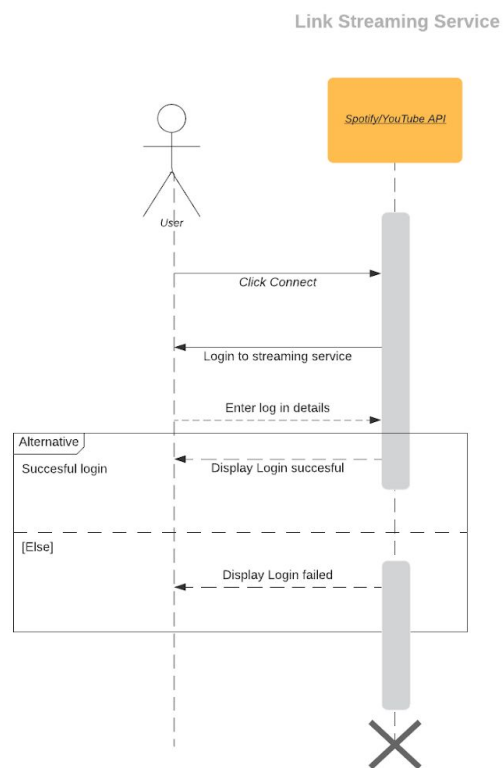
## Edit account settings

- The user will access account settings by clicking on the respective button
- A list of options to change (eg. privacy, avatar and deactivation) will be provided
- The user will then input their change to the desired setting. This request will be passed to the database of users.
- This will involve the necessary error checking to approve or disprove the change.
- If successful, the change will be made and the user notified of the successful change.
- If unsuccessful, the change will not occur and the user notified of the failure in change and asked to retry.



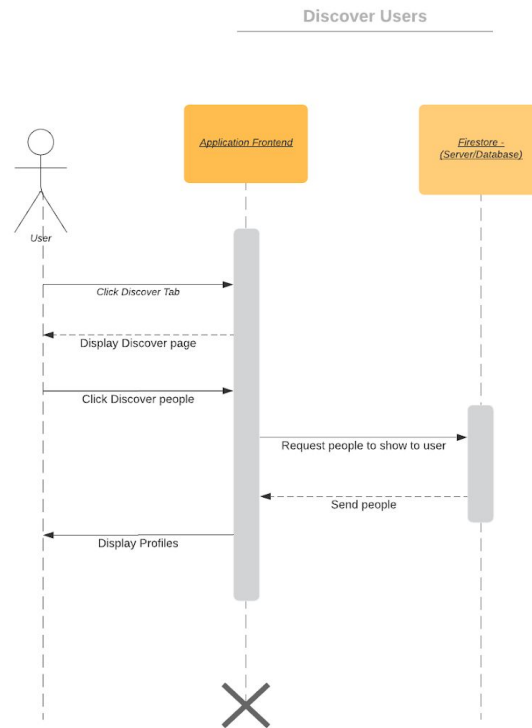
## Connect to Streaming Platforms

- User chooses a streaming platform to connect with and the application sends a request to link to the platform's API
- The platform's API returns a request to login and user is prompted by application to enter login details
- User inputs login details which the application sends to the platform's API
- The platform's API:
  - On success, the streaming platform is connected to the user's account
  - On failure, returns a login failed to the application which is displayed to the user



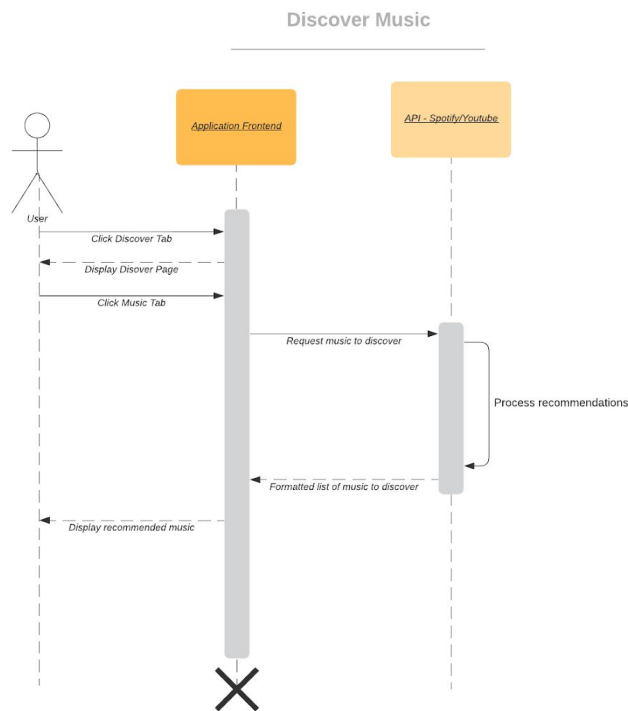
## Discover People

- From anywhere on the site a user can press the discover tab and choose the discover people section
- The server will then fetch users from the database that will be recommended for the user to discover
- These profiles will be displayed and the user can view them and connect with them



## Discover Music

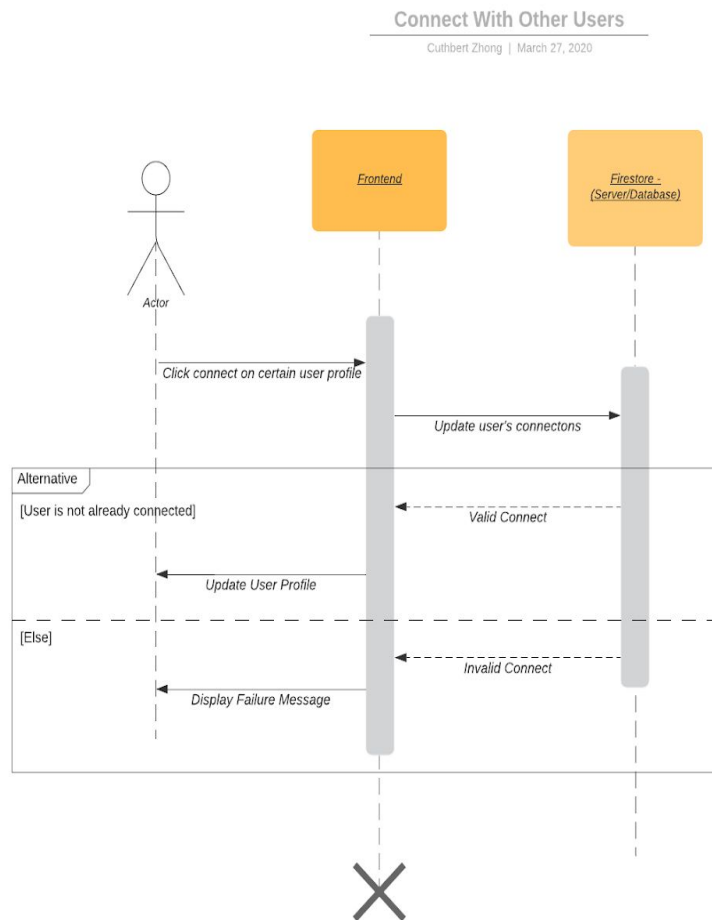
- From anywhere on the site a user can press the discover tab and choose the discover music section
- The application will then request for music to be recommended from their connected streaming service
- The recommended music will be processed and returned by the streaming service API
- These will be displayed to the user





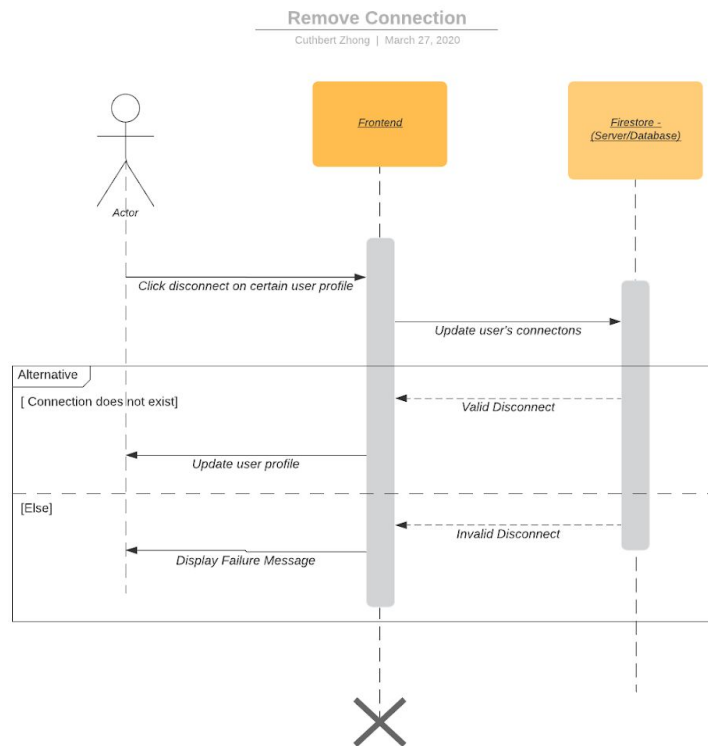
### Connect with other users

- When viewing another user's profile, a user can click a button to connect with that user. When that button is clicked, the frontend sends a query to validate the new connection.
- If that connection is valid, then the user is added to the list of connections and if not valid, an error prompt is displayed.



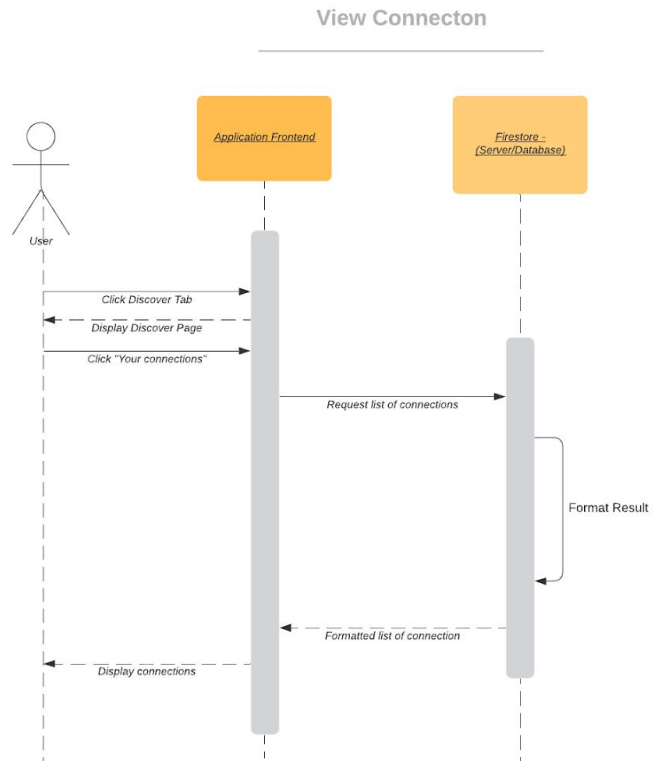
## Remove connections

- When viewing a user's profile that is a connection, a user can click the disconnect button.
- When clicked, the frontend validates the disconnect request with the backend server and if successful, the user is removed from the list of users' connections.



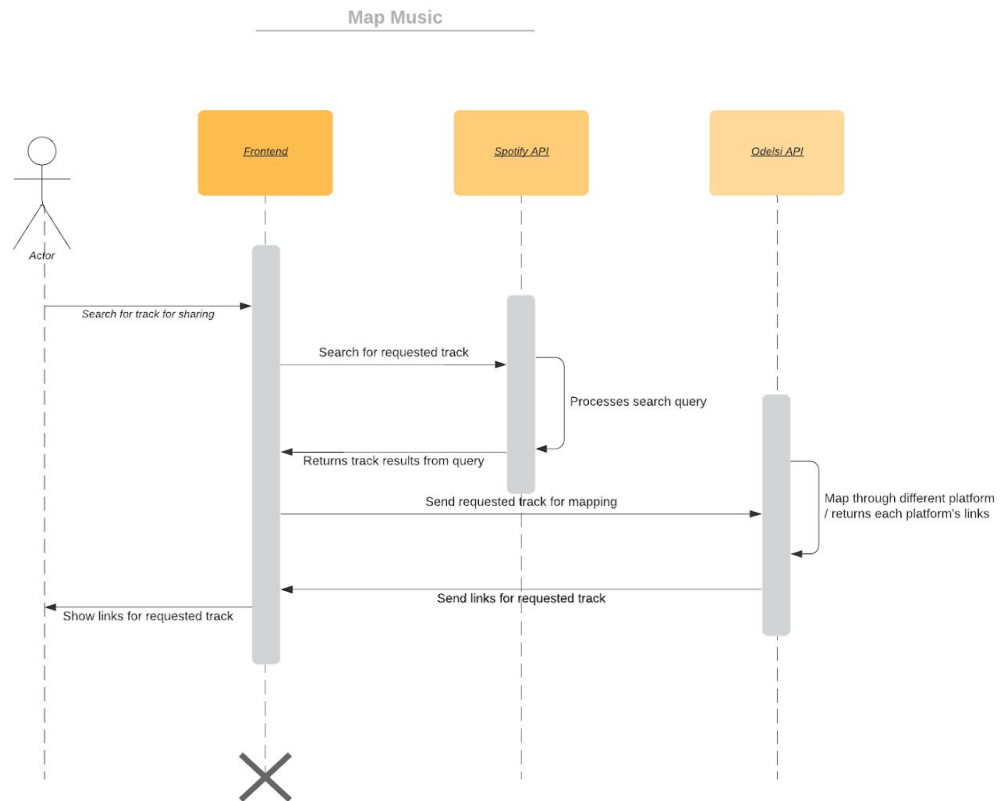
### View existing connections

- A user can access their connections from the Discover tab under "Your Connections"
- The server will then fetch connected users from the database
- These profiles will be displayed and the user can view them



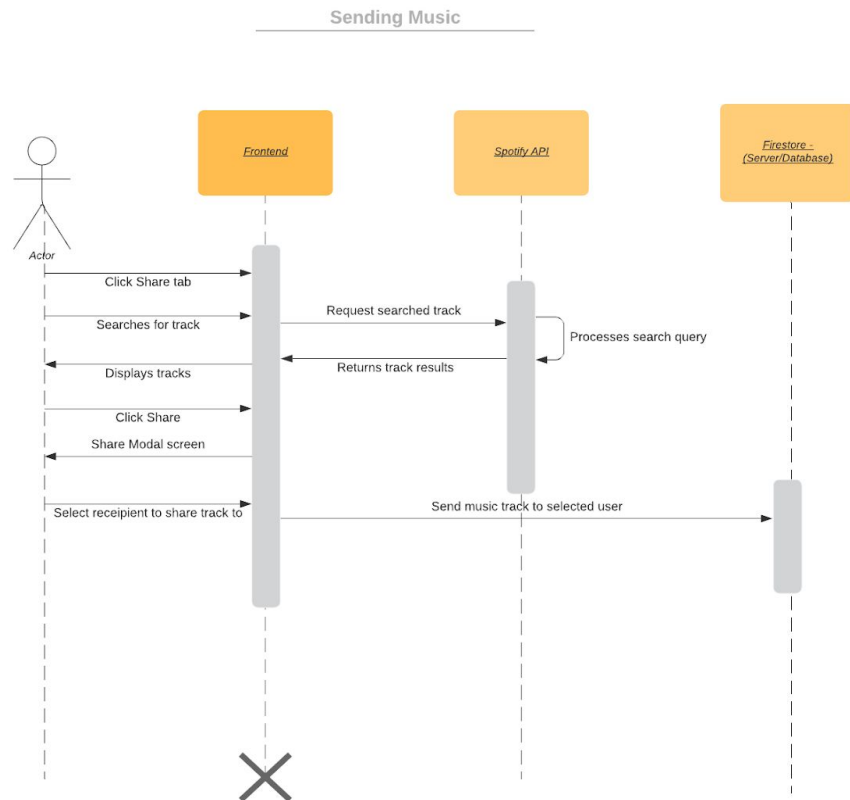
## Map music tracks

- User searches for a track to play or share
- A track provided by Spotify's API through searching will then be a part of a request sent to Odelsi's API
- Odelsi will map the track for other streaming platforms and return their respective links to be viewed by the user



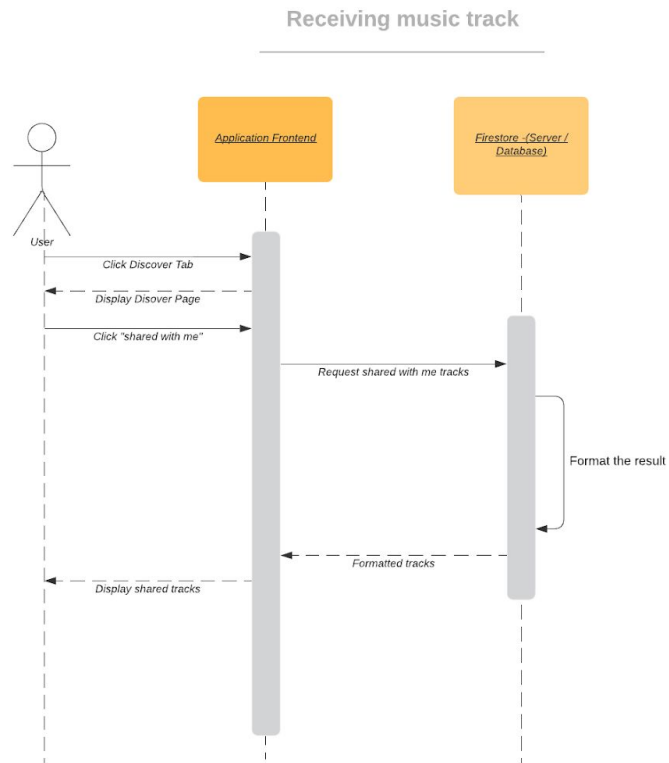
## Sending Music

- The user can select the “share” tab and search for music
- The server will fetch the track results to display to the user
- User can press the share button
- User will then see a modal screen with a list of connected users to share the track to
- Once selected the track will be sent to the user to be found in their “Shared with Me” section



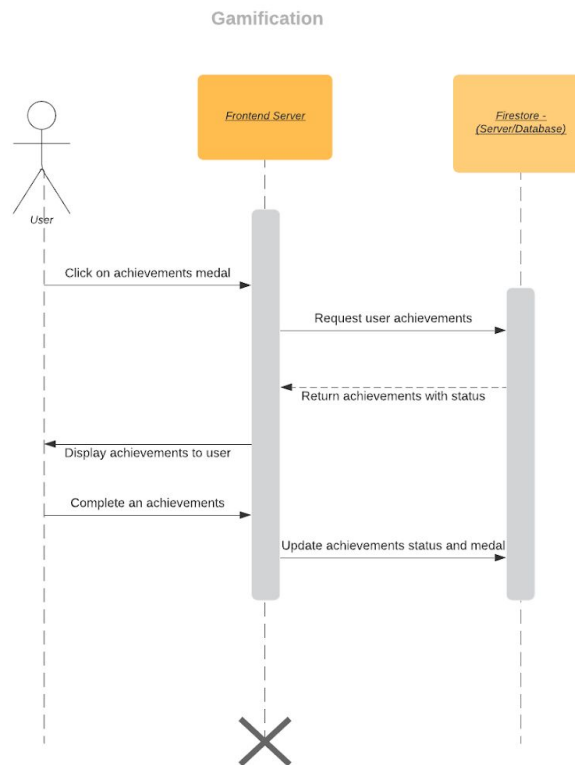
## Receiving music tracks

- In the Discover tab, a user is able to view their “Shared With Me” section
- The server will fetch tracks that have been shared to the user's account from the database
- These tracks will be displayed and the user can discover them through playing or sharing it to other users as well.

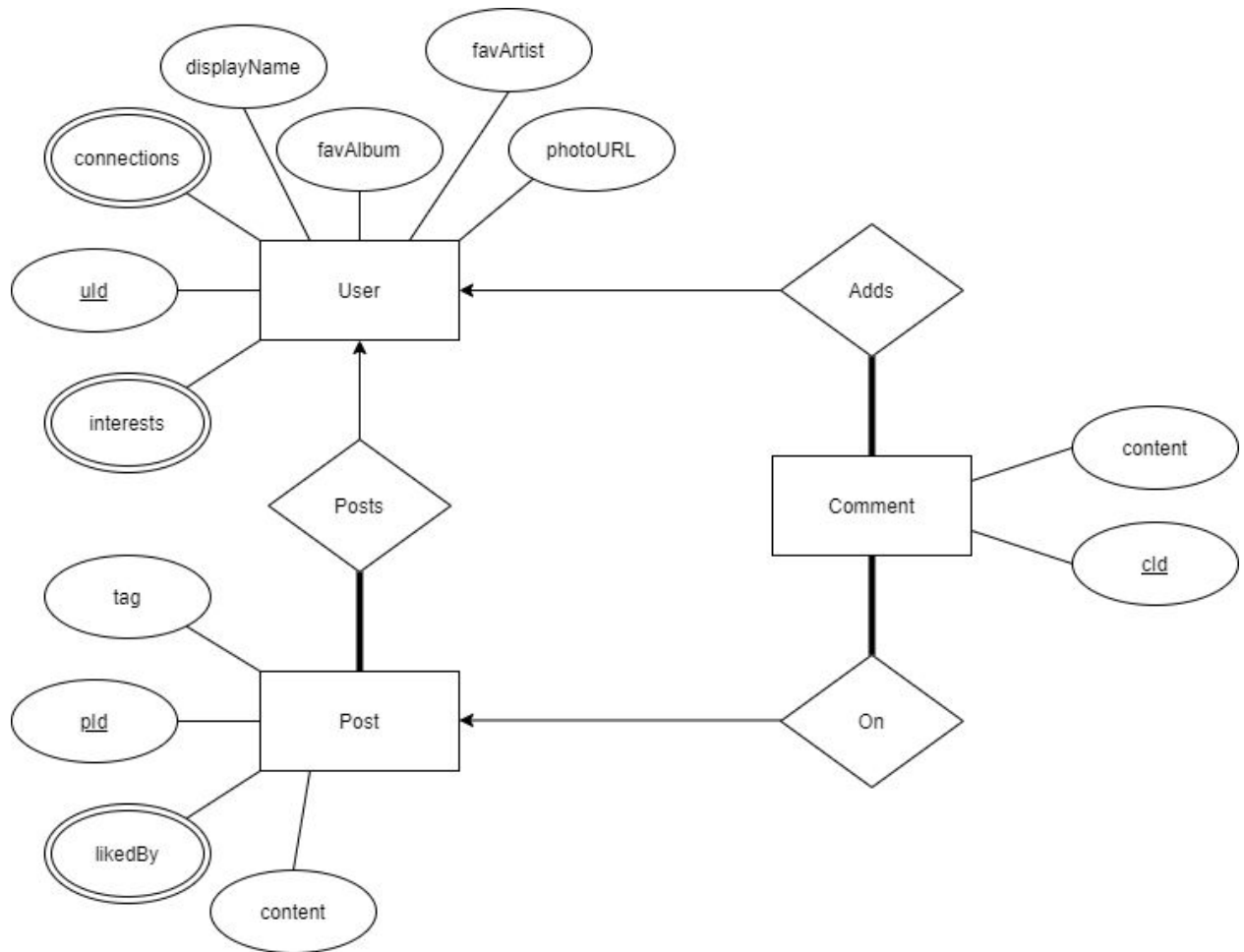


## Gamification

- A user can view their achievements through the profile tab
- Clicking on the achievements medal, they are able to view their current accomplished achievements and status
- Upon completion of an achievement, their status and list of achievements done is updated in the database



## ER Diagram





## **Key Design Considerations**

In the effort to create our intended personalised and shared experience through music on the platform, this involved understanding the ways users wanted new music to be presented to them and how they would want to share music with others in a community that we would create through the platform.

### **Discovering Music**

In order to contribute to the personalised experience of the platform, the “Discover Music” feature needed to be based upon a user’s listening habits. Spotify and Youtube’s APIs were key to achieving this. Instead of creating a manual static way of developing a sense of a user’s music tastes through simply taking input of their favourite artists on sign up, we used these APIs to achieve a dynamic and efficient way of understanding a user’s listening habits. Upon connecting with a streaming service, a user’s top artists are automatically used to personalise their experience. This is used to filter posts, recommend and share music. This is achieved through API endpoints that make the platform aware of their top artists and what tracks to recommend to the user.

### **Community**

Central to any social media platform, creating a sense of community within the platform was an important aspect of design. As with many other social media platforms, Mosh provides users with the ability to discuss through posts, likes and comments. However, due to Mosh’s focus on music, there were greater opportunities to provide this community experience. This was achieved through the filtering of posts/discussions based on music taste in order to allow users to better engage in discussions with people who share similar taste. Through our discover people feature, we provided a way for users to view other users’ favourite artists, albums and interests to create an open community free for the exploration of new people and relationships. In addition, Mosh’s music sharing feature supported by Odelsi’s API creates a unique community experience. As streaming services and social media platforms do not provide their respective users an ability to easily share music to users across platforms (eg. a user who uses Spotify finds it difficult to share music with someone who uses Apple Music), Mosh’s music sharing feature allows for users to share music with their connections as well as allows senders and receivers to play music that they discover on various platforms linked within the web application.

## **Team Organisation and Conclusion**

### Responsibilities/organization of the team

- Cuthbert
  - ◆ Handled frontend development of discover pages and profile modals for Mosh
  - ◆ Implemented firestore functionality to allow for user persistence, dynamic user feed with like, comment and post implementation and an automatic gamification system
- Elliott
  - ◆ Developed the authentication system
  - ◆ Worked on sequence diagrams and software architecture diagrams
- Alvin
  - ◆ Responsible for Google login with Firebase
  - ◆ Responsible for Youtube/Google API
  - ◆ Contributed to integration of Spotify and Youtube
- Luke
  - ◆ Researched useful endpoints and gained access to Spotify API
  - ◆ Created sequence and ER diagrams
- Austin
  - ◆ Developed front-end aspect of API integrations including music cards, sharing modal and Share tab
  - ◆ Implemented Spotify API integration with the web application - setting up access token and integrating it with Discover Music, searching, sharing and personalisation features
  - ◆ Implemented Odelsi API to support cross-platform music sharing feature

### How did the project go in your opinion

The general consensus was that the project went reasonably well. We were able to develop a visually pleasing, functional web application with promising features. This involved taking the time to understand and integrate the APIs we used as well as developing the database infrastructure and the user interface. Through the experience, each team member learnt a lot about API integrations, user interface design and software development in a collaborative environment.

However, due to time constraints, upon reflection, the user interface could have been improved to provide a smoother flow as well as better design. Also, we believe that the features we have implemented could have been fleshed out and expanded upon in order to create more innovative features. This is a matter of improvement with team coordination and early research.

### Any issues/problems encountered

- Apple Music API was not accessible to us due to issues with registering the web application
- Due to the cancellation of events around the COVID-19 pandemic, we could not effectively integrate our intended “Discover Concerts” feature with Ticketmaster’s API
- Soundcloud was also intended to be a part of the platform, however Soundcloud is tailored to a very niche market and we decided that we should focus on our other main APIs due to time constraints

### Would you do it any differently now?

- Earlier research into how our chosen APIs worked and how we can effectively integrate them with our app
- Designing a logo for our platform
- Adding a feature that allows users to choose a colour theme for the platform for better personalisation
- Implementing a chat feature between users
- Adding more streaming service platforms to connect with
- Create a smoother transition between pages
- Optimise speed of loads and renders
- More extensive gamification system