Team Number:
#7
Team Name:
https://cuboulder.zoom.us/j/93709434501

Comp Sci Hotties (Group 07)
Team Members:
Logan Chayet - Logan-Chayet - logan.chayet@colorado.edul

Clay Fricke - WillyFri - william.fricke@colorado.edu

Jack Higgins - higginsjack - jack.higgins@colorado.edu

Jordan Wood - jwoody02 - Jordan.Wood-1@colorado.edu

Preston Dotsey - Pdotsey - preston.dotsey@colorado.edu

Project Features List

1. **Login to Apple Music or Spotify**
    a. After the app has launched, users are presented with the option to sign to either Apple music or Spotify (this step is skipped if they have signed in before).
    b. The API available for both platforms return an authentication key which would be saved on device to be used in the future (for future transfers)
2. **Search for Playlists**
    a. Search for existing playlists on both Apple Music and Spotify & view the songs in them
    b. Screen presented after logging in -- search bar at the top where the user can either search for playlists (and they are presented in a table) or paste the link to a playlist to get more information, view songs in playlist, and start transfer.
3. **Search for Songs**
    a. Find each song in playlist selected and find the equivalent on the other platform
    b. Since there is no direct transfer method already existing, backend will search for each song (using public API) & find the best match (by comparing name and artist) and add to the queue for transferring to playlist
4. **Create Playlist**
    a. Create playlist using user's (logged in) account and transfer playlist cover, name, and description

b. Get metadata & other info using public API and upload info to other platform when creating a playlist.

5. **Add all songs**
    a. Add all songs found in Feature 3 to playlist made in Feature 4 and report to user which songs were/were not found & transferred successfully
    b. Once all URLs for equivalent songs are found, use asynchronous requests to add all songs all at once.
    c. Also give users the option to keep the same order of songs which will slow down the transfer as asynchronous requests cant be used.
        i. I.e if The pina Colada song is the first song in a playlist it'll keep its position as first when transferred.
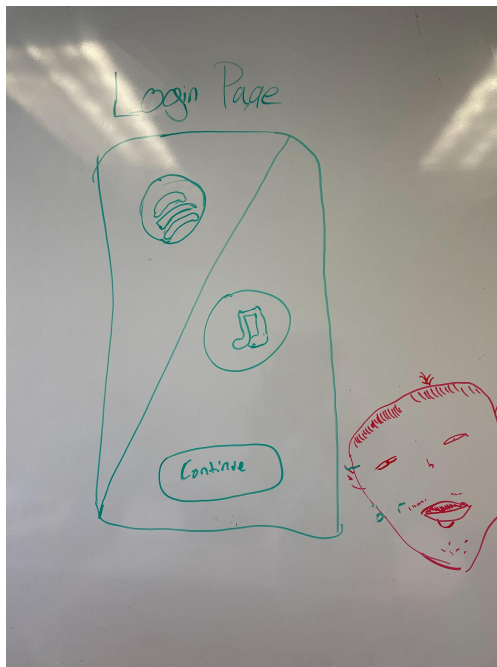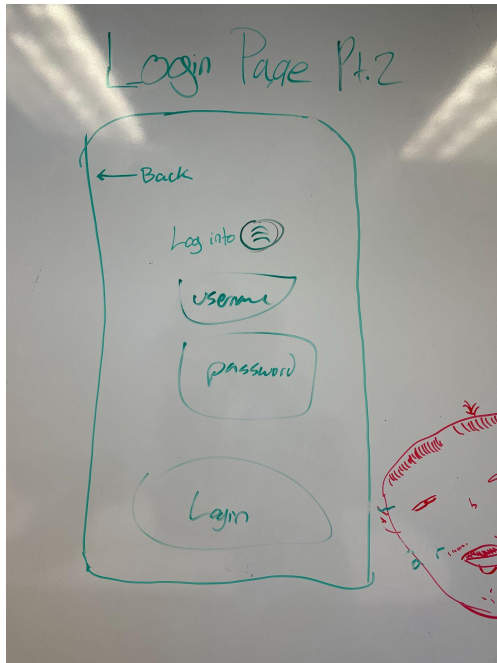
Project Plan
https://trello.com/invite/b/e4EE4ypT/c2cc88d12986aa343ef4a92fbdf4020d/kanban-template

Wireframes & Design
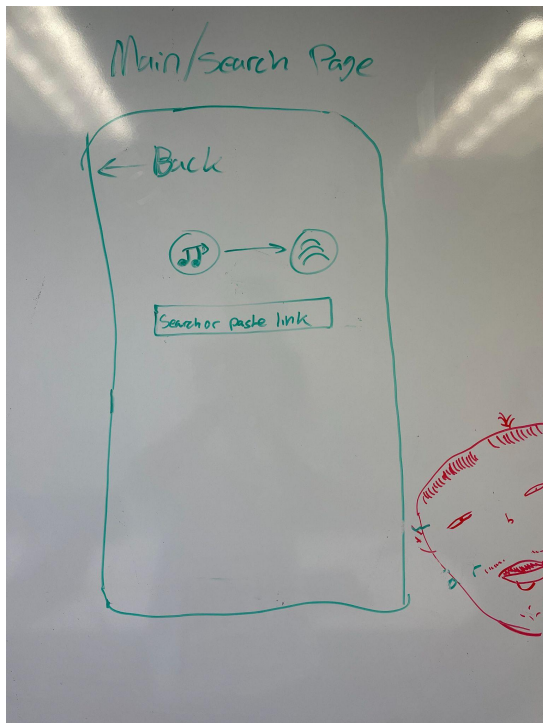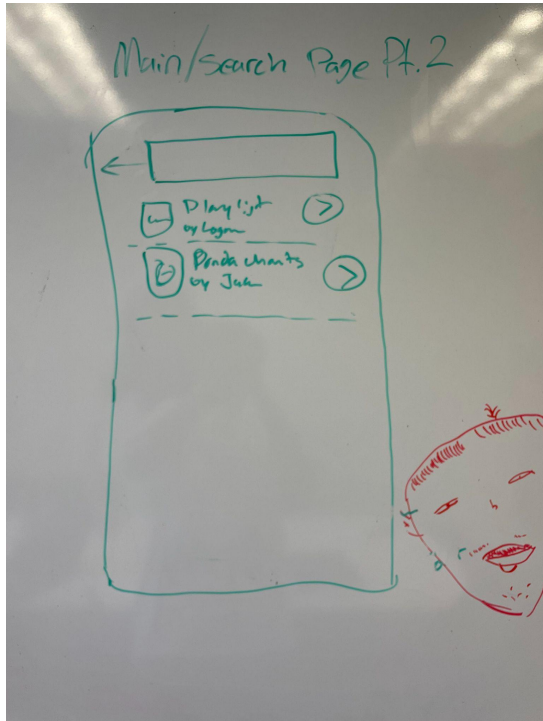# (UIKit will be used for design)

Login page(s)

- First the user can choose whether they want to transfer a playlist TO Spotify or Apple music
  - Possibly they can skip this step if they just want to browse
- Next, they are brought to the login page for that platform and this is the page they actually sign in on (Login Page Pt. 2 above)
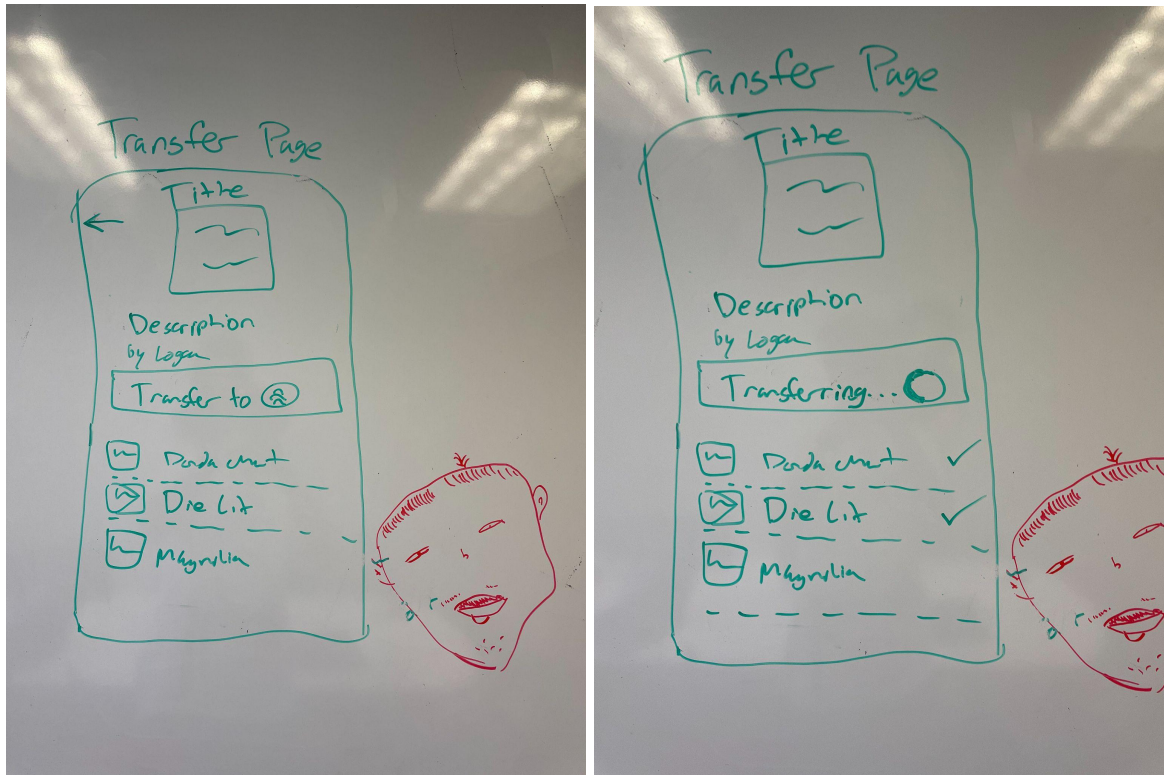
Main page

- On the Main/search page users can either paste the link to a playlist or search for a playlist using the search bar
- The user can also tap a "SWITCH" button to change which platform is being transferred too

Search Page

- In the search page users can start typing in the name to a playlist and results will show up with cover art, author, and title.
- User can select a playlist and they will be brought to the Transfer Page

Transfer Page

- After tapping a playlist in the search results table on the search page, the user is presented with much more info about a playlist, including the songs in it as well as a transfer button which the user can then tap to transfer songs
- One by one the app (or backend server) will add songs to the new playlist on the new platform but may have to be limited as the public API requests may be limited to a certain amount per minute

Individual Contributions:
- **Logan - Contributed to Project Plan and drew wireframe. API documentation (source code) uploaded to Git Hub**
- **Preston - Researched API protocols, contributed to user stories.**
- **Clay - Contributed to project features and assisted with user stories. Will learn Spotify API & Apple API: search for songs & add to existing playlist**
- **Jordan - researched both API, contributed to app features & wrote up/drew wireframe**
- **Jack - Contributed to Project plan and created sprints (Trello board), researched API**