USER REQUIREMENT
TUNESHARE

Team Members

Angel Alvarez - Lead | Bryan Camarillo - Engineer liaison | Rodrigo Esteves - Doc. Manager | Malachi Peel - Programmer | Ryan Moua - Programmer | Mingjie Wang (pineapple) Programmer

Description(main goal)

TuneShare is a social platform where users can connect with others to share the music they love, discover new music, and connect with friends and artists. We aim to create a social platform, where users can create and maintain profiles that display their unique music tastes, and opinions. The platform will allow users to share their music with other users, add friends with similar music tastes, and discover new genres of music from countless users with unique tastes.

Stakeholders want to prioritize a well done UI that will be able to be working for users and non users. Need to be able to post, like and comment on the feed. Need to have admins to have rights more than the users. Different pages that will show the main feed, trending, and for other users.

User Requirements

Features

- Create accounts/ profiles
    - Users will be able to create an account and profile to access the feed and others profiles
- General Usage
    - Share songs, post on their feed, adding other users, profile pictures, be able to like and comment
    - Link Spotify to their profiles
    - Verification to artist accounts
    - Create Playlists
- Dark Mode

Integration Requirements

- Spotify WEB API

    - Chosen for the abundance of music available and the accessibility it provides to this project
    - Free to use
    - In-depth documentation
- Youtube API
    - Chosen for the abundance of music and music videos available
    - Free to use

User Interface

- Login Page, trending feed and searches will be locked for non users
- Main feed will have the trending and personal posts for users
- 2 different searches
    - Main search bar is for music
    - Secondary search is a button that will activate for finding users

Constraints

- This semester, we have a lot of time to try and get all of our must-have features in the prototype.
- Getting rate limited
- Learning the tech stack that we will be using in this project as while we are familiar with some of the tech stack we are not proficient in all of them.
- Resources: Given the Apple Music API does have a significant cost associated with it.

Use Cases

- Case 1:
    - Description: A friend trying to give you music recommendations, and chooses to introduce you to this site to send you some music (i.e. an album or a single song)
    - Actors: A new user, and a Previously logged-in user.
    - Steps: ‘
        - New user lands on the main page of the website, which displays all trending music, as they have no previously logged preferences on the site.
        - A prompt to either sign in(if not already signed in) or sign up appears any time a user attempts to interact with another profile(i.e: attempting to view another account profile, view comments, like a piece of music)
        - Since the user has not created an account on the website they choose to sign up in which case they would fill out some credentials (i.e: email, phone numbers, username, password).
        - Once the new user has created a new account they would now have permission to search for their friend to send them a friend request in order to initiate a direct message.
        - Once the established friend has logged in using their credentials(username and password), they would see your friend request, in which case they

would accept the friend request and then send you a song through direct messages.
        - Once the direct message has been received, you would then be able to interact with the song (play a snippet, add to favorites) and then be able to send a message reacting to the song.
    - Expected Outcome:
        - A user is able to seamlessly sign up to the website and receive music recommendations directly from their friend.
- Case 2:
    - Description: A logged-in user creates a playlist and shares it with friends.
    - Actors: A logged-in user and friends (recipients).
    - Steps:
        - The user logs into their account and navigates to the main page.
        - The user selects several songs by using the search bar or browsing trending music.
        - The user clicks on the "Add to Playlist" option for each selected song.
        - The user is given the option to create a new playlist, which they title.
        - After compiling the playlist, the user clicks the "Share Playlist" button.
        - The user is prompted with sharing options (e.g., share via direct message, post to profile, or generate a link).
        - The user selects "Share with Friends" and picks specific friends from their contact list.
        - The user adds an optional message and clicks "Send."
    - Expected Outcome:
        - The friends receive the shared playlist in their inbox or notifications, with the option to listen to the playlist directly from the message. The playlist is also saved to their library if they choose to follow it.

- Case 3:
    - Description: Displaying, or posting, a song on your page.
    - Actors: A logged-in user
    - Steps:
        - User utilizes search bar to search for desired song
        - User selects desired song, and clicks “share” button
        - The user is presented with the option to share to a friend, or to post
        - User selects post and is directed to page to edit post
        - User can title the post, and share a description/ review
        - User can then click “share” button to post to their feed/profile
    - Expected Outcome: The user’s post will be displayed on their profile with the title visible from the profile view. Friends of the user will be able to see their post,

along with the description, on their own feed; with the ability to like and comment.

Priorities and Milestones

Main priority is to have a functional UI. Users should be able to explore the feed on the main page, search for songs and other users. Users should have their own page and be able to share snippets of songs to their personal page or to their friends.

Milestones

- Backend
    - Database
    - APIs
    -
- Frontend
    - Login page
    - Main feed
    - Profile pages