Spinterest

Kayla Delk, Colter Purcell, Gale Dela Cruz, Pragati Toppo
Created on 11/19
Last updated 11/21
Repo: https://github.com/kadelk04/CSC308_TeamProject

Product Vision and Scope

The Problem

As it stands, there are few options for avid music enjoyers to share their personal collections of playlists and compilations with the world. Whether it be building out a wide library of genres spanning Polish Classic Piano to Latin Tech House, or just trying to nail that perfect “Sweater-Weather-vibe” playlist, we wanted to provide a tool and platform for Spotify users to discover and share their meticulously crafted playlists with a broader audience than their immediate following on Spotify.

Glossary of Terminology

- Spotify API: The Spotify API allows developers to programmatically retrieve and interact with Spotify’s music streaming platform.

Updates From Previous Version (3/11/25)

- Data Schema Changes
    - User model
        - Added following and followers so that users can add friends.
    - Playlist model
        - Added likes, creator information so users can interact with other user’s playlists.
    - Notification model creation
        - Created for a notification system so that users can facilitate real time interaction with other users, such as liking playlists and sending follow requests.
- User Stories Completion
    - We were unable to complete all of our planned 309 user stories, but completed the remaining 308 stories.
- Goals

    - Due to not receiving approval for unlimited Spotify API usage, we were forced to spend a majority of our time re-evaluating our approach to both querying the API and storing its data.
- Activity Diagram
    - Dashboard Page
        - Dashboard now works as an interactive explore page, where you can view all Spinterest users playlists.
        - If you were unable to successfully login or create an account, you will not be permitted access to the dashboard.

User Personas

Simon

Age: 19
Occupation: College Student/Intern
Relationship Status: Single
Gender: Male
Activities and Clubs: Cal Poly Bike Builders, Cycling Club, Triathlon Club, Unicycle Club

"Did you know that Tame Impala is just one guy?"

Special Interests
- Avid music listener
- Endurance athlete
- Concert-goer
- Very into maple syrup

Work Experience
- FUJIFILM Design Engineer (CAD, 1 year)

Technical Information
- Top apps:
    - Strava, Spotify, and Instagram
- Uses smart phone to watch reels and update his growing athlete fan base on his adventures
- Spends 3 hours on phone each day

User Motivation
- Motivated by "kudos" and vibes
- Looking for ways to share his passion for bands and genres
- Likes to share niche artists with friends
- Makes lots of playlists, categorizes none of them
- Needs a way of sharing his vast catalogue of bands that are on the verge of making it big
- Would like to discover new music for long bike rides

Simon is a college student seeking opportunities to broaden his music listening horizons and share his love for indie artists. He loves to talk about obscure artists and is delighted when one of his friends listens to one of his niche artists. He wants to find other users that share similar music tastes to chase that feeling outside of the scope of his existing Spotify friend network. Given he is an endurance athlete as well, he is always looking for new tunes to listen to to ensure he never has a dull moment on his adventures. Simon would benefit from using Spinterest for its playlist discovery features as well as the social networking to add people on Spotify who curate playlists with artists that he has interest in.

COMFORT WITH TECHNOLOGY
INTERNET
SOFTWARE
MOBILE APPS
SOCIAL NETWORK

AIMEE
AGE: 35
GENDER: FEMALE
RESIDENCE: HOBOKEN, NJ
EDUCATION: CULINARY INSTITUTE OF AMERICA MASTERS IN FOOD BUSINESS
WORK EXPERIENCE:
- BAKERY ASST
- CAFE OWNER (CURRENT)

RELATIONSHIP: MARRIED W/O KIDS

GOALS
- HAVE A GOOD CAFE PLAYLIST
- MAKING SURE THE CAFE HAS A VISUALLY APPEALING AESTHETIC
- ENSURING THE FOOD AND DRINKS ARE APPETIZING AND TRENDY

MOTIVATIONS
- MOTIVATED BY THEIR LOVE FOR SWEET TREATS AND JAZZ/CLASSICAL MUSIC
- SHE WANTS TO SHARE HER VISION OF THE CAFE'S VIBE WITH OTHERS

INTERESTS
- CULINARY ARTS
- PRODUCT MANAGEMENT
- PLAYING THE PIANO
- PAINTING
- WATCHING MOVIES

TECHNICAL
TOP APPS: INSTAGRAM, PINTEREST, SPOTIFY, TIKTOK
- INSTAGRAM AND TIKTOK ARE USED THE MOST TO PROMOTE THEIR BUSINESS
- SPENDS THE MAJORITY OF HER TIME FILMING BAKING/LATTE-MAKING VIDEOS

Aimee is a cafe owner who is looking to elevate the aura of her cafe. She believes that if she can find music that helps to showcase her cafe as tranquil and laid-back, but also trendy, more customers will choose her cafe as a place of lounging. She hopes that the music she finds via Spinterest can help grow the cafe's playlist, leading to less songs being replayed.

User Stories

308 Stories (Initial Implementation)

- Spotify Data Import: As a user, I want to get my data from Spotify so that my listening history is accurate.
- Spotify Account Sign-In: As a user, I want to sign into my Spotify account on the site so that I can view the details of my profile.
- Site Navigation: As a user, I want to look at the site so that I can navigate through an explore page and a profile page.
- Profile Privacy Options: As a user, I want to make my profile private or public so that I can choose who can view my profile.
- Profile Sharing: As a user, I want to be able to share my profile with my friends/family.
- Music Likes and Saves: As a user, I want this application to be able to like and save my music on Spotify.
- Friend Profiles: As a user, I want to be able to follow friends on the site so that I can view their profiles.
- Custom Profile Page: As a user, I want to have a custom profile page so that my friends and family can identify who I am.
- Bio Editing: As a user, I want to edit a bio so that others know about my thoughts/feelings.

309 Stories (Polish and Social Aspects)

- Combined Playlists: As a user, I want to be able to generate combined playlists so that I can share/listen to music with friends.
- Playlist Compilation: As a user, I want to be able to make/compile playlists of my favorite songs.
- Music Recommendations: As a user, I want to be able to listen/be introduced to new music based on my listening history.
- Vibe Matching: As a user, I want to know my vibe so that I can find others with a similar vibe.
- Pinned Music Display: As a user, I want to view pinned music so that I can change it during the day.
- Profile Comparison: As a user, I want to be able to compare my profile/statistics with my friends so that I can start a conversation about what artists we both enjoy.
- Playlist Generation from Existing Playlist: As a user, I want to be able to generate playlists based on an existing playlist as inspiration.

Technical [non-functional] Requirements

- The system shall be compatible with Chrome and Firefox latest versions.
- The system shall be hosted on the cloud without needing installation on customer locally owned servers.

Non-Goals or Out of Scope

- N/A, we believe all of our user stories are achievable.

Future Goals

- Have Spinterest in a working state early / mid winter quarter so that we can collect user playlists, which are necessary for later user stories.
    i. We want to pull from our database of playlists and use them to generate each user a tailored dashboard, similar to an explore page.
- Get all user stories done
- Message Spotify so that we get unlimited access to their API and don't get ratelimited.

Assumptions

- User has a Spotify account and has playlists saved to their account.

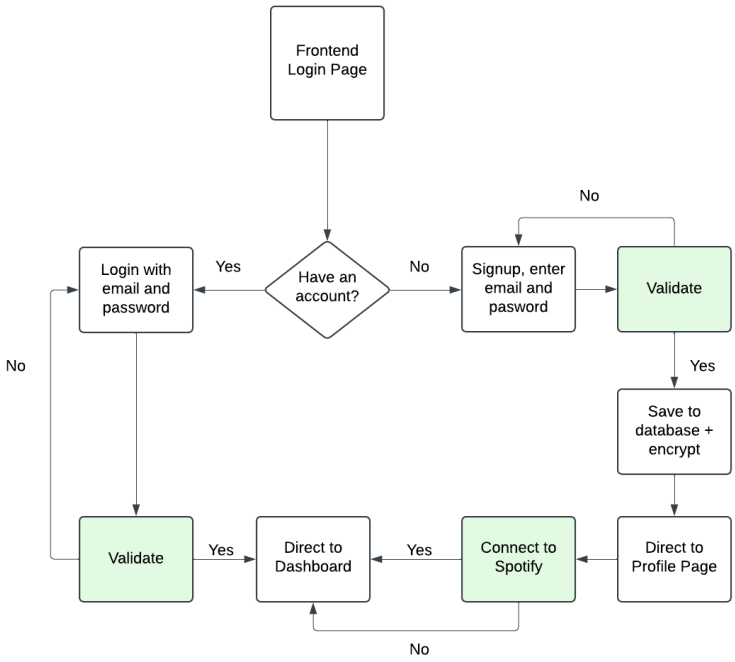
Design (Updated 3/11/25)

Tech Stack

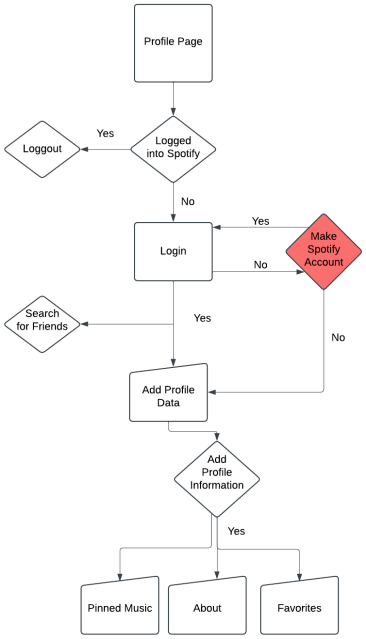
- MERN (MongoDB, Express, React, NodeJS)
- We elected to use this stack because it had the greatest overlap in relation to our skill sets (and familiarities) that we entered this class with

Activity Diagrams

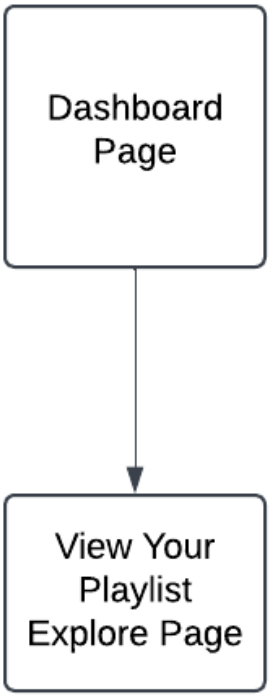
- Login Page Activity Diagram



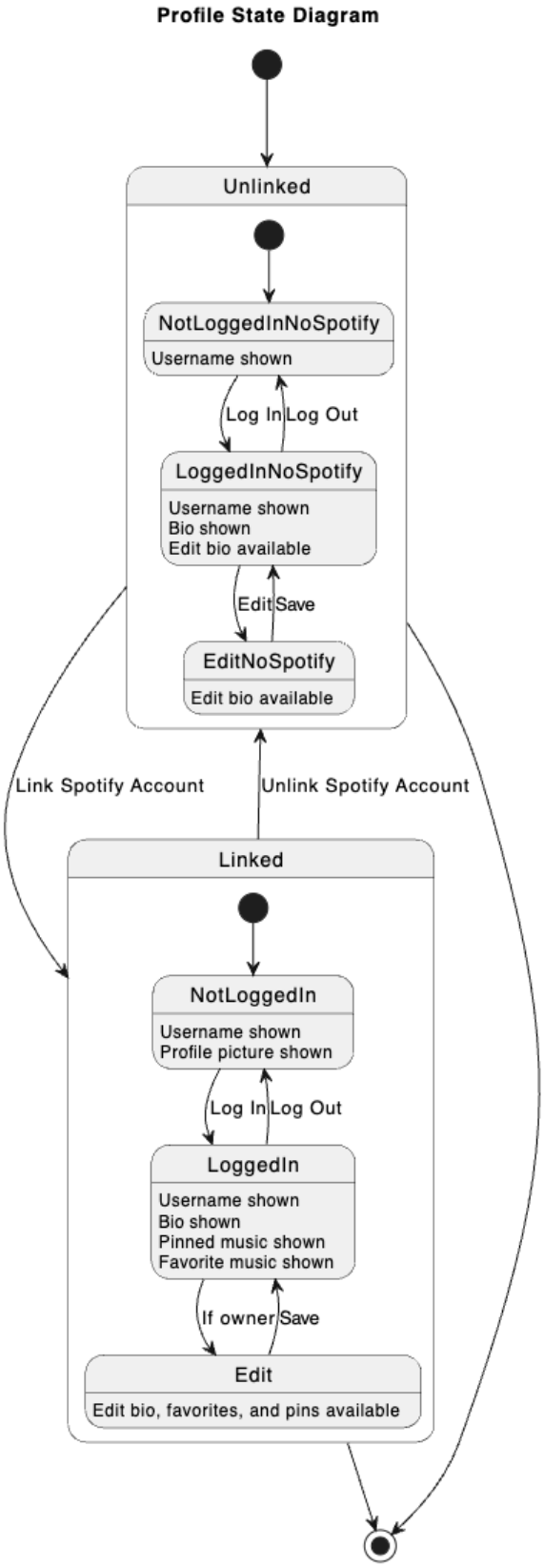
- Profile Page Activity Diagram



- Dashboard Page Activity Diagram (Updated 3/11/25)



State Diagrams

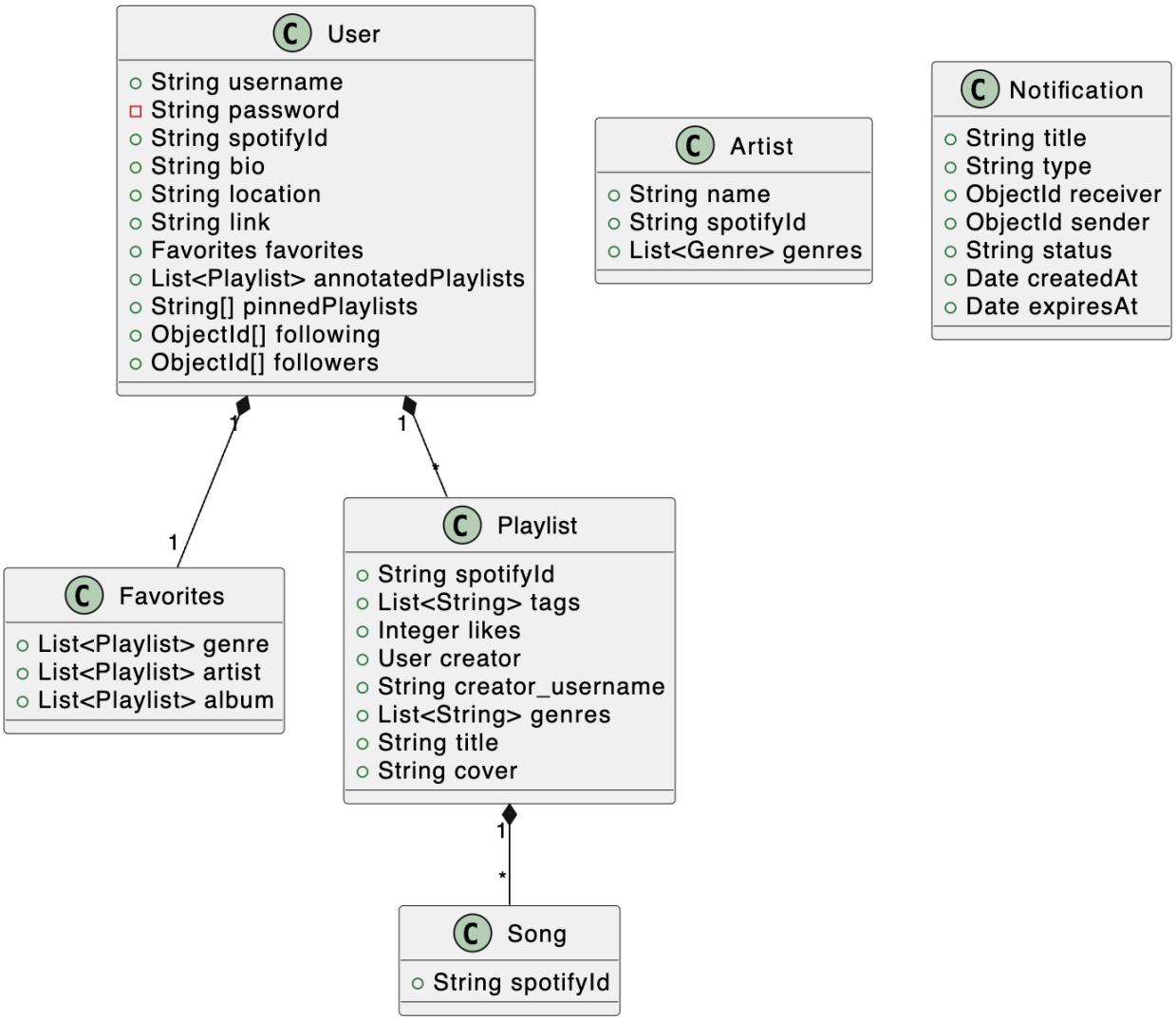


UI Prototypes:
https://www.figma.com/design/8RgzhJh0eKCRSCs1DEv9Fk/UI-Dashboard?node-id=0-1&node-type=canvas&t=PLI9Y2Am1WpW8kkZ-0

External Software Interfaces
Our application relies heavily on the Spotify API. In fact, most of the data that we have is ported from the API and stored for long-term recall within our database. Spotify uses the OAuth 2.0 protocol to authorize the API requests, which we currently (before MongoDB login is implemented) have used to manage user accounts without having to own any user data ourselves. We currently use HTTP GET requests to retrieve our data from the API (we don't need to make PUTs, PATCHes, or POSTs because we don't edit the user's library). In the future we will use the other CRUD operations to modify user's liked song's, playlists, and friends, but those are features for when the platform becomes more social. We have also worked on implementing a backend of our own to increase the amount of data that we can pull from our own resources (and limit our chances of having issues if Spotify's API is unavailable or we are rate limited). We set up this database using a Mongo Cluster hosted by AWS. The collections we have include the Favorites, Users, Playlists, and soon to be Artists. The playlists are updated in our database each time a user logs in to Spotify. The favorites are user-defined (and hosted by us) and are managed by a series of CRUD operations in our backend. We elected not to store songs as the sheer volume of documents that would have to be stored in our backend would potentially bloat our database. Instead we use the playlistId from the playlist object in our database to query the SpotifyAPI for the tracks that a given playlist contains.

Data Schema/Model

- Data Relationship Model - Updated 3/11/25



- Each user will have a local username and password, along with their Spotify ID, enabling retrieval of Spotify data through the API.
- From the user profile, we will store essential details such as the user's bio, location, and personal link. Additionally, we will manage a `Favorites` model that highlights the user's preferred genres, artists, and albums for display.
- For each user, playlists fetched via the Spotify API (annotatedPlaylist model) will be stored and enhanced with "annotations," which include tags describing the playlists. Each playlist will consist of its Spotify ID (for identification), a list of songs (each having its own unique Spotify ID), and a collection of descriptive tag
- For each user they will have a followers and following attribute so they can follow other Spinterest users.
- Entity Sets:
    - User Entities

        - Playlist Entities that belong to a user
        - Favorite Entities that belong to a user
        - Artists
- Relationship Sets:
    - Playlist belong to user
    - Songs belong to playlist

Tests (Updated 3/11/25)

- Code coverage report

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	86.76	89.03	85.07	86.76	
src	0	0	0	0	
backend.ts	0	0	0	0	1-35
src/controllers	86.84	89	92.98	86.84	
UserController.ts	80.91	92.18	82.35	80.91	36-48,61-65,160-189,231-232,267-270,345-346,349-350,360-372
artistController.ts	91.66	90.9	100	91.66	94-98,100-103
favoritesController.ts	100	100	100	100	
notificationController.ts	95.62	90.69	100	95.62	114-116,150-152,180-182,234-236
playlistController.ts	86.87	96	85.71	86.87	38-39,142-160
profileController.ts	73.02	77.08	100	73.02	34-107,109-110,115-116,182-183,199-200,211-213,245-247,374-382,384-392
spotifyController.ts	100	87.09	100	100	34,163,183-184
vibesController.ts	92.02	88.57	100	92.02	64-66,99-101,115-121
src/middleware	96.06	92.3	100	96.06	
auth.ts	96.06	92.3	100	96.06	41-44,53
src/models	98.21	100	0	98.21	
Artist.ts	100	100	100	100	
Favorites.ts	100	100	100	100	
Notification.ts	98.24	100	0	98.24	44
Playlist.ts	100	100	100	100	
User.ts	96.77	100	0	96.77	46,57
src/routes	100	100	100	100	
index.ts	100	100	100	100	
src/routes/api	96.92	100	100	96.92	
artistRoutes.ts	84.37	100	100	84.37	13,17,21,25,29
loginRoutes.ts	90	100	100	90	7
notificationRoutes.ts	100	100	100	100	
playlistRoutes.ts	100	100	100	100	
profileRoute.ts	100	100	100	100	
spotifyRoutes.ts	100	100	100	100	
userRoutes.ts	100	100	100	100	
vibeRoutes.ts	100	100	100	100	
src/utils	34.69	100	0	34.69	
connection.ts	34.69	100	0	34.69	16-42,45-49

-
- Acceptance criteria specs
    - https://docs.google.com/document/d/1hc0EOyTt1-NLrpY4EA5ZGleJ9J7Axmhm2Io-IKNeugQ/edit?usp=sharing

CI/CD (Updated 3/11/25)

- Frontend URI: https://proud-desert-0084e151e.4.azurestaticapps.net
- Backend URI:

Deliberation

Open Questions

- How do we avoid getting rate limited by the Spotify API?
    - Either by storing artist data in our MongoDB database, or by requesting a higher request quota from Spotify.
        - We did end up implementing this approach and even extended into our playlist collection
        - This reduced our loading time on our front page by about 10x

References and Acknowledgements

- Product Idea: Gale Dela Cruz