TuneMe

Project Members and Advisor

Member(s)

- Zaina Qasim
- Program: Computer Science

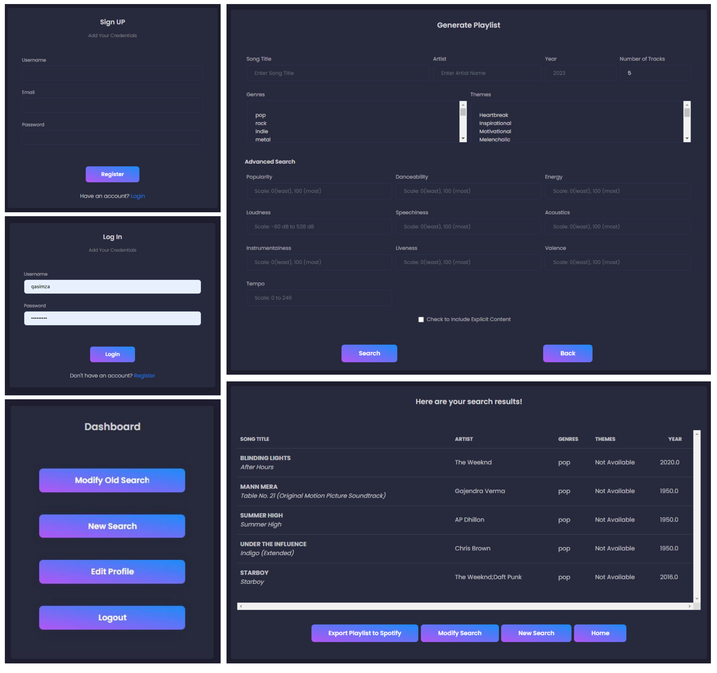
Faculty Advisor

- Fred Annexstein

Project Abstract

The most popular music listening, and recommendation apps are Apple Music and Spotify. The former uses listening history and similarity to other users' tastes to recommend music, and the latter analyzes a song's rhythm and structure. Both do not ask users to specify their definition of "similarity" - by artists, years, and/or genres, etc. This project aims to provide better suggestions by doing so.

User Interface

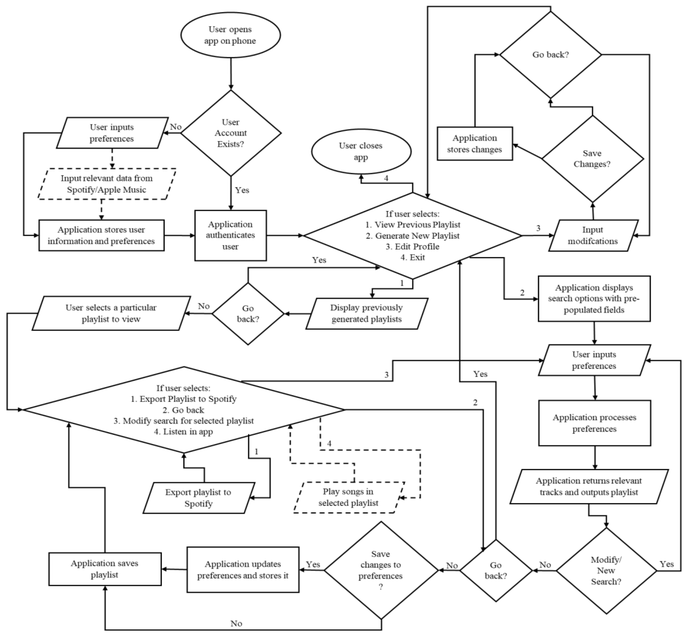


Technologies Used

Programming Languages – Python, HTML/CSS/JavaScript, SQLite

APIs – Flask, NumPy, Pandas, Matplotlib, scikit-learn, Bootstrap 4, jQuery.

DESIGN DIAGRAM



System Overview

INTENDED AUDIENCE/USER: Anyone who wants to find new music.

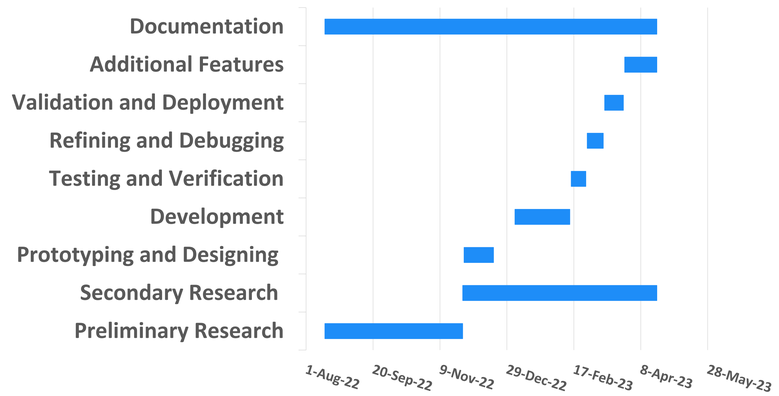
KEY FEATURES AND FUNCTIONALITY:

- **Signup/Login:** Easy Signup/Login process to access user dashboard.
- **Dashboard:** Dashboard links to new search page, old searches and a page to edit profile.
- **Search Page:**
  - Allows users to find music by one or more of the following - Song Title, Artist, Genre, Year or Theme.
  - User may specify the number of suggestions (default =5).
  - 10 extra advanced parameters for fine tuning search results.
  - Allows user to exclude(default) or include explicit content.
- **Results Page:**
  - Results page displays songs matching user's query.
  - Allows user to export playlist to Spotify, Modify the current search or perform a new one.

SYSTEM'S CONSTRAINTS:

- Inaccurate/Dissatisfactory results on some occasions.
- No results for highly complex queries.
- Results with no themes.

Timeline of Project Phases



Total Days 248

Total Hours 281

Results

- Successful implementation of a **Flask application.**
- Effectively find music based on a variety of parameters, including artist, year, and song title.
- At least **10 songs/query** on average.
- **70 % median accuracy.**
- **0.092 seconds/query.**

Challenges

Data Cleaning – Kaggle datasets posed data cleaning challenges in a project despite providing valuable information, requiring special attention to ensure data quality and consistency.

Technical Knowledge and Time - The app's frontend was built with Ionic for Android and its backend with Flask, but complications led to a switch to a lighter Flask app, while the challenge of mastering musical terms arose for accurate music categorization and personalized recommendations.

Diversity and Cultural Challenges- Music, with its varied cultural influences such as language and instrumentation, encompasses diverse genres unique to geographical, ethnic, religious, or linguistic groups, making it unfeasible to develop a comprehensive solution applicable to all types of music.