



NANYANG TECHNOLOGICAL UNIVERSITY

SINGAPORE

IM3080 Design and Innovation Project

(AY2023/24 Semester 2)

Project Report

Title: Moodify

Github: <https://github.com/jojojoelelel/IEMDIP2023S2-Moodify>

Submitted by: Group 1

Supervisor: Chua Hock Chuan

Table of Content

| | |
|---|-----------|
| 1. Background and Motivation | 3 |
| 1.1 Background Information | 3 |
| 1.2 Motivation | 7 |
| 2. Objective | 8 |
| 2.1 Chatbot to Improve Song Recommendation | 8 |
| 2.2 Music Diary to Save Song Recommendations | 8 |
| 2.3 Virtual Reality Concert for Synchronous Music Sharing | 9 |
| 3. Review of Literature/Technology | 9 |
| 4. Design and Implementation | 9 |
| 4.1 Design and UI/UX Considerations | 9 |
| 4.1.2 Thematic Choices - Color Scheme | 10 |
| 4.1.3 Thematic Choices - Font and Typography | 11 |
| 4.1.4 Thematic Choices - Logo Design | 11 |
| 4.1.5 Thematic Choices - Icons | 11 |
| 4.2 Final Design | 12 |
| 4.2.1 Site map | 12 |
| 4.2.2. Use-case Diagram | 13 |
| 4.2.3. ER Diagram | 13 |
| 4.2.4. Figma Design | 14 |
| 4.3 Implementation | 14 |
| 4.3.1 Frontend | 14 |
| 4.3.2 Backend | 15 |
| 4.4 Development | 16 |
| 4.5 Discussion | 16 |
| 5. Conclusion and Recommendation | 17 |
| 5.1 Conclusion | 17 |
| 5.1 Recommendation for Future Work | 17 |
| 6. Appendices | 19 |
| 6.1 Appendix A — Jira Diagram | 19 |
| 6.2 Appendix B — Design Diagrams | 19 |
| 6.3 Appendix C — User Guide | 19 |
| 6.4 Appendix C — Final Design Guide for Light Theme | 20 |
| 6.5 Appendix D — Source and Figma URL | 32 |
| 6.6 References | 32 |

1. Background and Motivation

1.1 Background Information

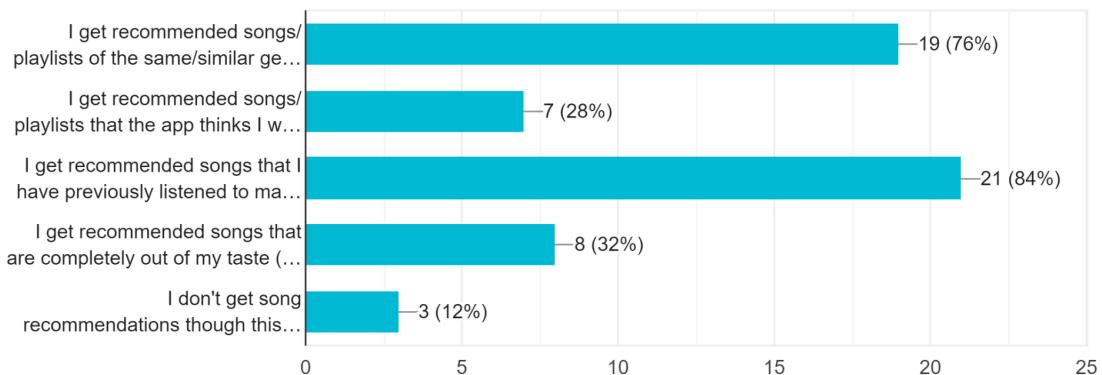
Music streaming makes up 84% of the entire music industry's revenue, with over 600 million people subscribed to a music streaming platform¹. Music streaming services such as Spotify and Soundcloud exploded in popularity as they provide a one-stop source for an extensive song library at a cheap price without the need to digitally download songs along with many other quality-of-life features, giving users great convenience and value for money. These music streaming services also have algorithms to recommend songs as well as entire playlists based on what the user has previously listened to.

However, such song recommendation algorithms upon further analysis of their capability, have been questioned by its users. They can only recommend songs based on the user's history, causing the algorithm only to recommend songs of similar genre and type to what the user has already listened to. Furthermore, songs the user has listened to many times before are recommended repeatedly to the user, which will further increase the amount of times the user listens to the song, causing the algorithm to undergo an indefinite loop of reinforcement learning, which is not desirable.

¹ <https://explodingtopics.com/blog/music-streaming-stats>

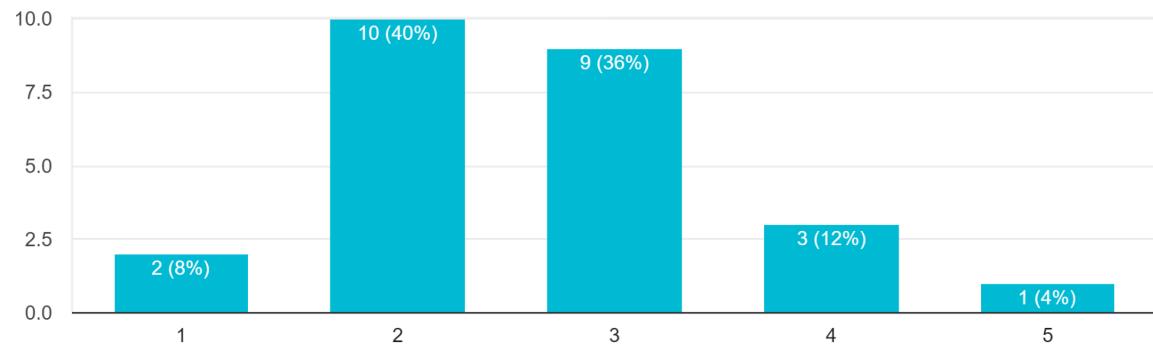
What kind of songs does your go-to music streaming platform (Spotify, Apple Music, SoundCloud, etc) recommend to you based on your listening?

25 responses



How would you rate the current quality and usefulness of the song recommendation algorithms on your go-to music streaming platform? (With 5 being the highest.)

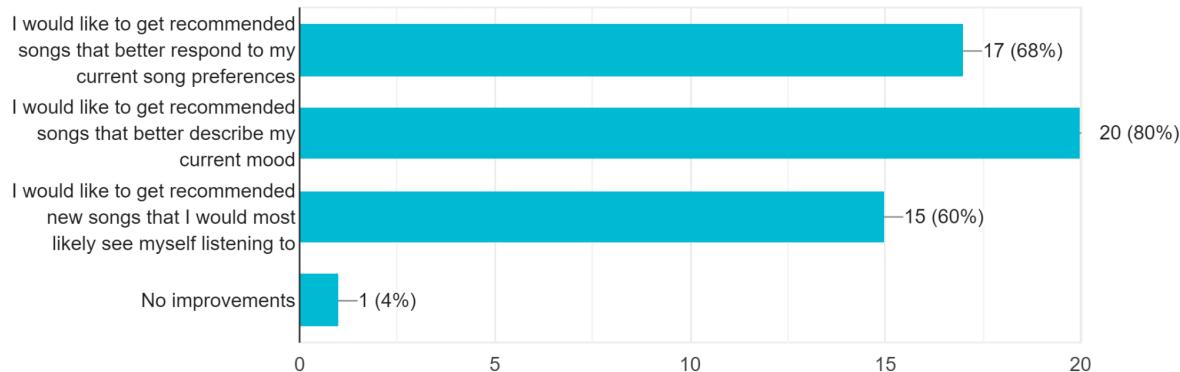
25 responses



Our survey results reflect this as a majority of respondents also noted that the recommender algorithms tended to give similar genres, or songs that the user has already listened to multiple times before, with little consideration as to what the user would want to hear next, this sentiment is reflected by our respondents noting that the algorithms still has room to be improved.

What improvements can you identify from the current song recommendation algorithms in these music streaming platforms?

25 responses



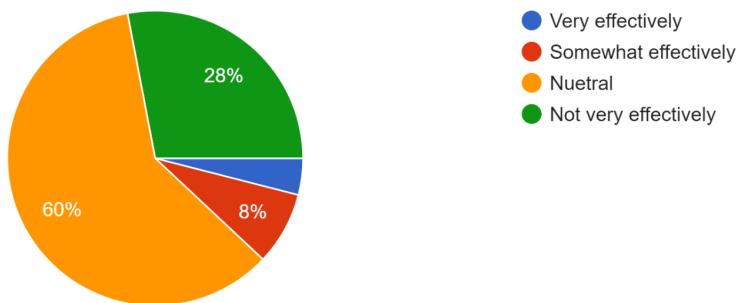
While these algorithms can be useful, due to their limitations, there is a lack of sensitivity towards a user's changing wants and emotions, which may lead to mismatches between what the algorithm recommends and what the user wants to listen to, this finding is supported by our survey noting that many people would like a more sensitive algorithm.

Listening to music can also be a cathartic and deeply emotional experience by itself. A lack of a dedicated feature to take note of the kind of music that one listens to over time to provide a place for introspection and reflection would mean not placing value to a key aspect of music.

There is also a lack of overall communal experiences in current services beyond sharing playlists and songs.

How effectively do you feel you are able to bond with your friends through shared interests with your current use of music streaming platforms?

25 responses



The asynchronous nature of current sharing features means that music's ability as a communal experience to connect and bond individuals is not leveraged, which is also supported by our respondents who believe that current music sharing features on existing platforms are not outstanding, showing that there is still potential growth in this area.

Physical music festivals and concerts have shown to be a successful communal musical experience where many people go to enjoy a shared atmosphere and music. Recreating such an atmosphere digitally thus shows promise in creating a synchronous and communal music experience within the streaming service app itself.

Thus, it is evident that improvements can be made to the existing system. We have thus codified key takeaways and solutions as follows

1. Unwanted Reinforcement Learning

Address unwanted reinforcement learning by recommending unique songs.

2. Insensitivity to the User's Current Mood

Implement the ability for the app to react and respond to the user's current preferences and state of mind.

3. Introspection and Reflection

Design a feature to record the type of music one listens to to promote introspection

4. Asynchronous Communal Experience

Create the ability for users to have a synchronous communal experience to facilitate connection and bonding between users.

1.2 Motivation

To create a music streaming app that addresses the found limitations and can recommend unique songs that are sensitive to the user's current preferences. The app also intends to improve upon current music-sharing features by leveraging synchronous music-sharing concepts to allow users to connect.

2. Objective

2.1 Chatbot to Improve Song Recommendation

Our app is designed to give unique song recommendations that cater to the user's current emotions and state of mind in order to solve the unintentional reinforcement learning and the insensitivity problem. We achieve this via a chatbot powered by GPT 3.5 that is finetuned to be a song recommender based on the user's current emotional state in order to streamline the process of searching for music that resonates with the user, as well as improving while also simplifying the recommendation system used. We use GPT 3.5 as it is one of the Large Language Models with the most amount of training data available at the moment, which allows it to have a large knowledge base of songs within its training data that can be recommended to the user. Whenever the user tells the chatbot about their current emotional state, the chatbot will then curate and recommend a unique song that is relevant to the user's mood.

2.2 Music Diary to Save Song Recommendations

One of our application's killer features is our music diary. This feature allows users to record their song preferences day by day just like how one would typically share about their feelings in a diary. As a supporting and integrative function to our AI chatbot, this feature saves the one recommended song given by the chatbot into a calendar screen. This saved information includes song name, artist name, and the song's URL to play the track. When the user clicks on one of the dates on the calendar, the song data from Firebase that matches with the clicked timestamp will be retrieved. As a user navigates from the calendar screen to the RecomScreen which shows the saved song information, the previously retrieved data will be displayed and the saved song's URL will be fed into the musicPlayer function so that it can be played in the embedded spotify music player. This added feature offers users a chance to track their mood and changing music preferences. From a business point of view, this new feature may also help to build stronger relationships between the user and the application which is essential to maintain the user activity of our application over time.

2.3 Virtual Reality Concert for Synchronous Music Sharing

Our app also features a Virtual Reality (VR) Concert feature that brings the experience of a music festival into the digital world within the app. The VR Concert provides an immersive, synchronous and communal music-sharing experience that will allow users to experience their favourite music in a virtual reality landscape using avatars. The aim for this feature is to provide the user a deeper musical experience as they transport themselves into our dynamically crafted virtual environment. Users will also have the ability to share the immersive experience with other users on the application.

3. Review of Literature/Technology

We reviewed existing music streaming services and analysed their strengths and limitations which would be taken into consideration in the design and implementation of our app. We decided on Spotify to build upon as Spotify is the largest music streaming service at the moment, with over 30% market share. The Spotify app has a robust suite of features, and a well-documented API that can be used to streamline the development process of our app.

4. Design and Implementation

4.1 Design and UI/UX Considerations

Our aim was to provide a clean and user-friendly interface that is intuitive for users to pick up if they already have experience using another music streaming app. We also provide a Light Mode and Dark Mode to cater to a broader variety of users who may have different preferences for different colour schemes.

Our navigation structure is clear and predictable, using standard music app conventions.

Key features are easily accessible via a persistent bottom navigation bar or clear menus.

4.1.2 Thematic Choices - Color Scheme

We have developed two distinct colour schemes within the app to enhance user experience in both Light Mode and Dark Mode, each designed to evoke specific themes and emotional responses. Our core palette of purple, pink, and teal, with accents of yellow, evokes a sense of futurism and fun.

In Dark Mode, We use a similar colour scheme to Spotify in order to pay homage to the base app. The colours used are contrasting, allowing text and images to be easily readable and

improving user-friendliness. The dark background in contrast with neon green gives a sense of futurism and energy, perfectly complementing the novelty of the interactive features and simulated live music experiences.

In Light Mode, we choose light, pastel colours, transitioning to a pink background to evoke calmness and relaxation, contrasting with purple and teal accents to evoke a sense of playfulness. The pastel colours are more mellow and thus aligns with our goal of wanting users to be more in tune with their emotional side when interacting with our chatbot and using the VR concert in order to better connect with other users.

4.1.3 Thematic Choices - Font and Typography

We utilised a clean, san-serif font family for optimal readability across all screens. Sans-serif fonts offer a modern and sleek aesthetic, aligning with the app's futuristic theme.

4.1.4 Thematic Choices - Logo Design

Contrasting the clean lines of our primary font, our app's logo features a unique, custom design. We incorporated hand-drawn elements into the logo to highlight the personalised recommendations and interactions offered by our AI-powered features. Additionally, we chose playful bubble lettering, a popular trend among younger generations, to emphasise the youthful, fun, and forward-thinking nature of our app. The logo prominently features our core colour palette.

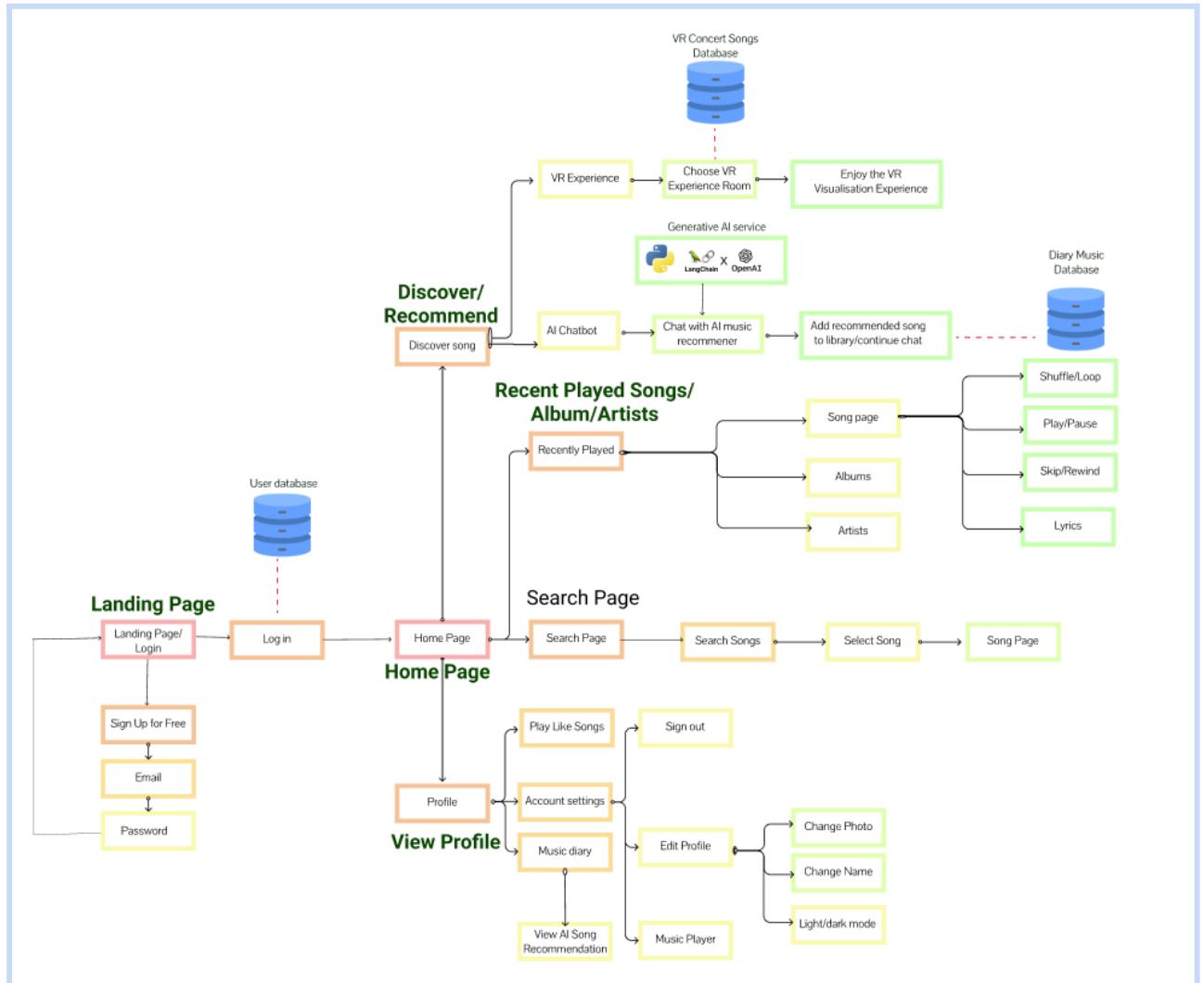
4.1.5 Thematic Choices - Icons

To provide a clear and intuitive visual language throughout our app, we opted to use the Ionicons library. Ionicons ensures that our app's icons look consistent and clean across all major platforms, including iOS and Android. The minimalistic style of Ionicons promotes immediate recognition and understanding of their associated functions. This reduces

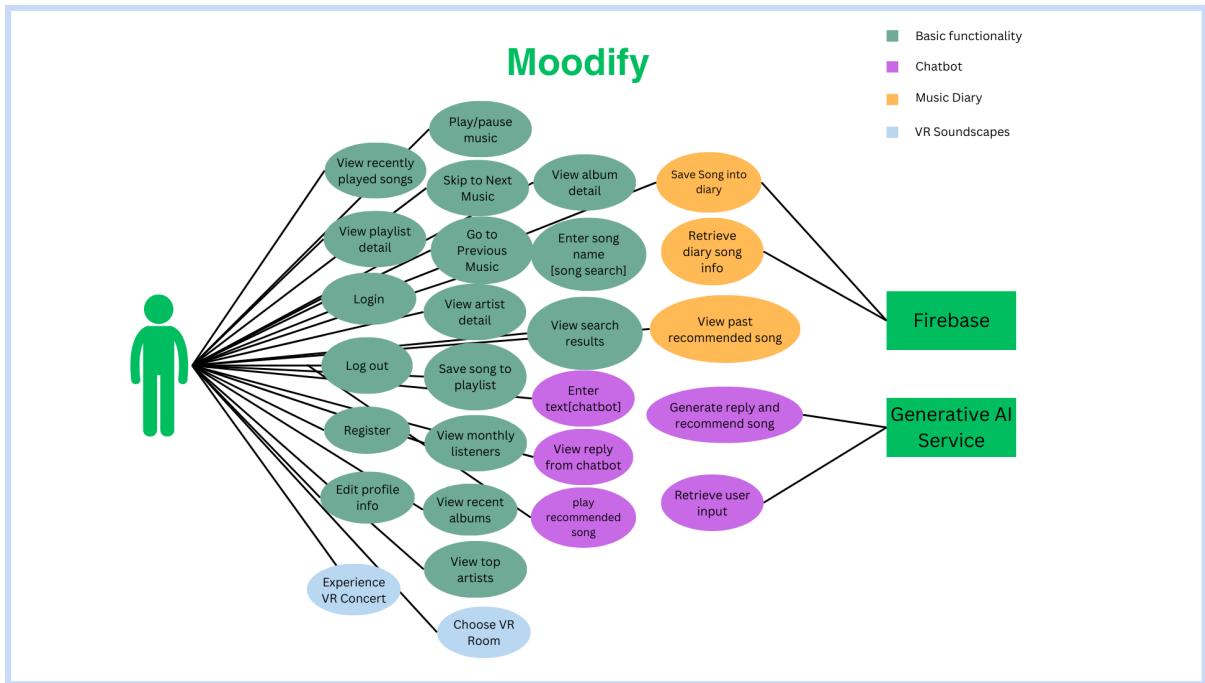
confusion and enhances intuitive navigation. Additionally, as vector-based icons, they maintain their clarity at any size, adapting perfectly to different screen resolutions.

4.2 Final Design

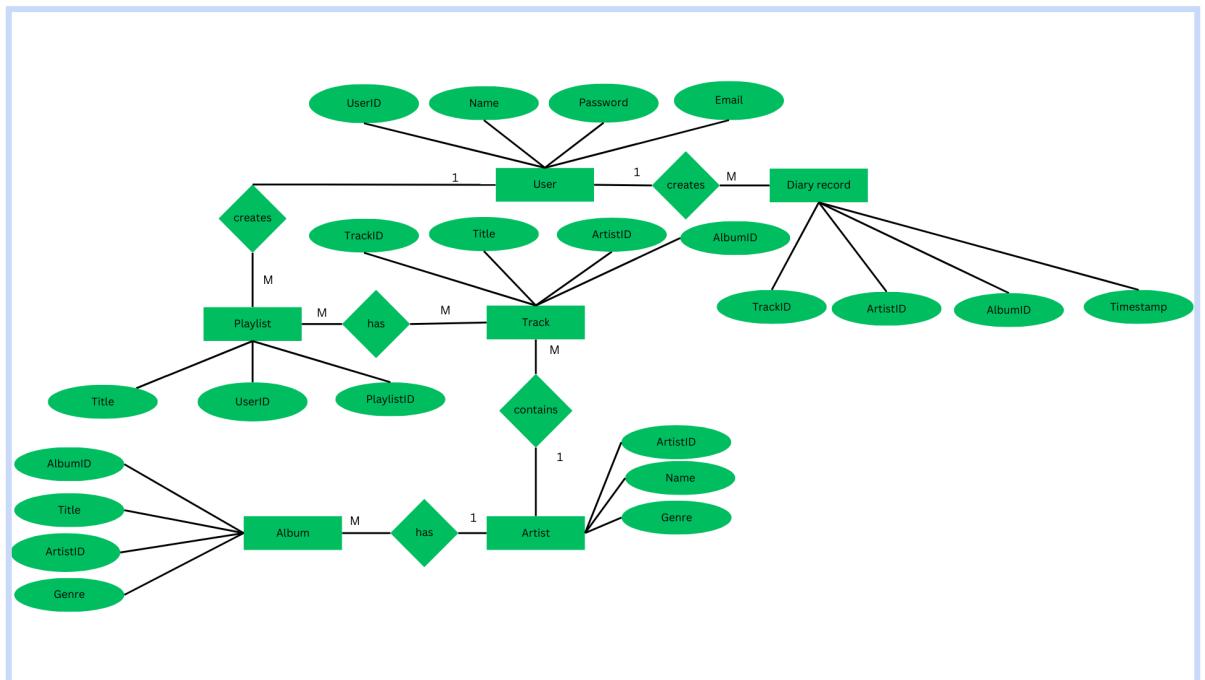
4.2.1 Site map



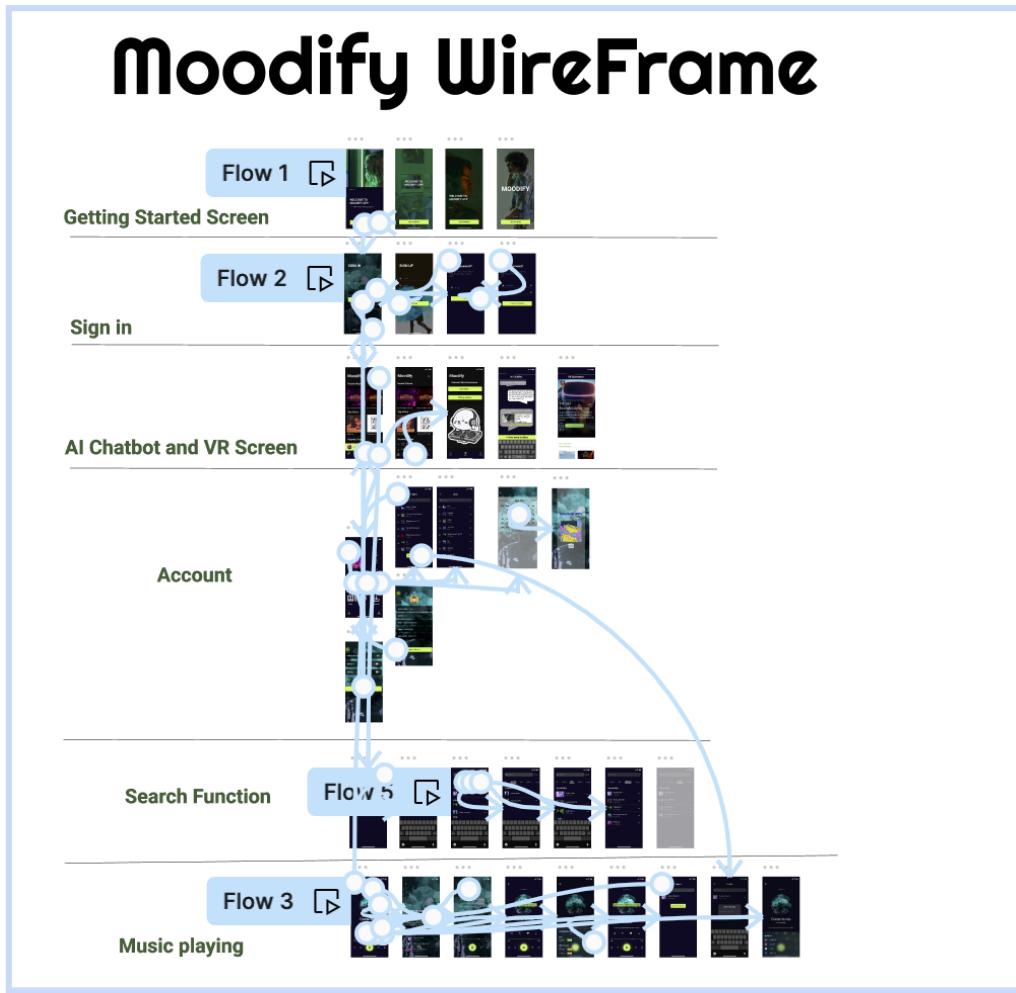
4.2.2. Use-case Diagram



4.2.3. ER Diagram



4.2.4. Figma Design



4.3 Implementation

4.3.1 Frontend

We've selected React Native as our frontend framework of choice. Developed by Facebook (now Meta), React Native is an open-source framework that allows developers to build mobile applications using JavaScript. Its key advantage lies in enabling a single codebase for both iOS and Android platforms, significantly cutting down on development time and resources. React Native blends native development elements with React, a top-tier JavaScript library for crafting user interfaces. This combination delivers almost-native performance, a smooth user experience, and access to an extensive range of libraries and tools, making it a highly effective and favored option among developers.

In addition to adopting React Native for frontend development, our team has also chosen WebVR to develop the VR component of our virtual concert. Integrating WebVR with React Native presents unique challenges and opportunities. While React Native is optimized for mobile app development across different operating systems, WebVR introduces added complexity, primarily targeting web platforms. This integration can be streamlined using libraries such as react-native-webview, which enable React Native apps to display WebVR content within a native app framework, effectively bridging the gap between native mobile development and web-based VR experiences.

4.3.2 Backend

We chose a combination of Firebase as well as Flask to expose Python code as an API for the backend. Firebase includes a host of features that simplify and accelerate the development process. Its real-time capabilities, serverless architecture, and built-in authentication, coupled with ease of use and integration with Google services, make Firebase a compelling choice over traditional server-side technologies for many projects. Python was used for developing the chatbot and its relevant functions as Python hosts an extensive list of libraries which are used for data science.

The chatbot itself is powered by OpenAI's ChatGPT 3.5, a cutting-edge large language model known for its generative capabilities. This advanced model enables our chatbot to deliver highly interactive and contextually relevant responses, making it a core component of our user engagement strategy. The integration of ChatGPT 3.5 through our Flask-powered API not only enhances the functionality of the chatbot but also exemplifies our commitment to leveraging the latest advancements in AI and machine learning to provide a superior user experience.

4.4 Development

Throughout the project, the team was split into two sub-groups, frontend and backend. The first few weeks were spent brainstorming and refining the ideas that we had into realizable goals, before wireframing and designing a prototype using Figma. We finished this initial stage by Week 3.

The group kept in touch, updated progress on deliverables and communicated feedback and new ideas via weekly meetings in addition to class time. We also used Jira as a project management tool in order to track tasks and facilitate collaboration between members by providing a central platform for anything related to the project. Jira allowed us to create and assign tasks and deadlines and monitor progress, which improved our efficiency as we could focus more on developing rather than managing.

While Jira was used to handle project management, Github was used to handle code-sharing management with features such as branches and version control.

4.5 Discussion

The project was a long process that involved a collaboration between design, frontend and backend. Because of the requirement, we required a minimum of 6 hours a week in order to integrate all 3 facets and build a working app.

We leveraged each individual in the group's unique skill set in order to overcome the challenges presented during development of the app using mostly unfamiliar frameworks. By continuous communication amongst ourselves and collaboratively researching on our desired outcomes, we managed to find answers to our questions and share our findings to help one another overcome difficulties.

Ultimately, this project has helped to improve teamwork, technical skills in the languages and frameworks involved in the development, as well as allowed us to gain familiarity in industry-standard tools like Github and Jira. All this gives us an insight into the industry and an understanding into the development process which would definitely be a valuable addition to our knowledge and resume.

5. Conclusion and Recommendation

5.1 Conclusion

Our app aims to improve upon current music streaming services by implementing features that aim to solve limitations of current recommendation algorithms as well as promote communal experiences through the experience of listening to music together. We achieved this via a chatbot designed to recommend unique songs based on the user's mood. We also provide a way to encourage introspection and reflection by implementing a music diary feature which records what songs the chatbot recommends to the user. We also include a VR concert feature to allow users to enter an immersive concert setting to listen to their favourite music. By combining cutting-edge technology with a focus on user experience and emotional engagement, our app stands at the forefront of the evolving landscape of music streaming services, offering users a truly personalized and enriching musical experience.

5.1 Recommendation for Future Work

Our recommendations for future updates to the app include the following.

- Due to time constraints and the complexity of the features that were implemented, we were required to budget our time and manpower to build the core features of the app, and thus were unable to meaningfully implement a multi-user session feature for the VR concert given the timeline of the entire project, which was one of the goals of the

feature. Future development work would include the implementation of such a feature for a truly communal listening experience.

- Currently, the model used to power the chatbot is GPT 3.5 which while contains a large amount of data and can be fine tuned to an extent for our purposes, is not a specialised model that we are currently using it for. A future update that changes the model that is specifically trained for the use case would see far better results.
- A major goal for future iterations of our VR concert feature is to enhance its scalability by automating the generation of the virtual environments for each song. Our current implementation, while engaging, presents a constraint in the number of songs we can support due to the manual designing of the environments. To overcome this, we envision the following approach:
 - Build an NLP model trained to analyse song lyrics and the music genre. This model would generate descriptive prompts to guide the environment generation process.
 - These prompts can then be integrated with an AI environment generator like SkyboxAI's API to dynamically produce and update the VR environments to reflect the song's tone and themes

6. Appendices

6.1 Appendix A — Jira Diagram

The Jira board displays the following tasks:

- TO DO:**
 - Implement Homescreen API (M666-96)
 - Figure out the access token and state management (M666-97)
 - Document backend setup and API integration (Backend) (M666-98)
 - Develop basic app navigation (Frontend) (M666-99)
 - Figure out music player bar and track player functionality (global) (M666-100)
 - Logo design (M666-101)
- IN PROGRESS:**
 - Initialise Firestore database (DEVELOPMENT OF BASE APP: Backend) (TASK-18)
 - Define scope and features (PREPARE TO PRESENT INITIAL DESIGN IN ...) (TASK-10)
- IN PROGRESS 1:**
 - Develop basic app navigation (DEVELOPMENT OF BASE APP: Frontend) (TASK-19)
 - Create user registration and login screens (DEVELOPMENT OF BASE APP: Frontend) (TASK-20)
 - Document backend setup and API integration (DEVELOPMENT OF BASE APP: Backend) (TASK-21)
 - Setup backend API endpoints (DEVELOPMENT OF BASE APP) (TASK-22)
- DONE:**
 - Wireframes and User Flows on Figma (PREPARE TO PRESENT INITIAL DESIGN IN ...) (TASK-11)
 - Finalise Tech Stack (PREPARE TO PRESENT INITIAL DESIGN IN ...) (TASK-9)
 - Create Github Repository (TASK-8)
 - Group meeting on 29/01 (PREPARE TO PRESENT INITIAL DESIGN IN ...) (TASK-15)
 - Prepare to Present Initial Design in Week 3 (TASK-7)
 - Brainstorm some ideas for projects with as much detail as possible (FINALISE OUR APP IDEA) (TASK-3)
 - Group meeting on Monday 22/01 (FINALISE OUR APP IDEA) (TASK-4)

6.2 Appendix B — Design Diagrams

Please refer to “4.2 Final Design” to see our Sitemap, Use-Case Diagram, ER Diagram, and Figma Interface Design.

6.3 Appendix C — User Guide

User guide:

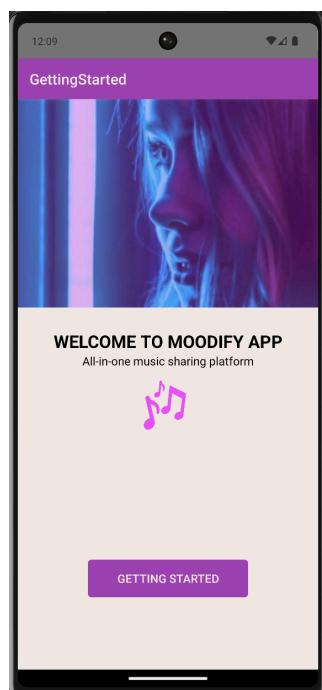
- Creating a Spotify Account:
 - Download the Spotify app from your device's app store or visit the Spotify website.
 - Sign up for a new account by providing your email address, creating a password, and filling out any other necessary information.
- Logging in to the App with Spotify:
 - Open Moodify app and navigate to the login screen.

- Select the option to log in with Spotify.
 - After logging in successfully, you'll be redirected back to the app.
- Enjoying Music:
 - Once logged in, you'll have access to recently played playlists, albums, and artists you follow directly from your Spotify account.
- Discovering Recommended Songs with the AI Chatbot:
 - Navigate to the Discover tab within the app.
 - Enter the AI chatbot feature.
 - Ask the chatbot for recommended songs tailored to your music profile.
 - Choose to save any recommended songs to your music diary if you enjoy them.
- Viewing Music Diary in the Account Tab:
 - Go to the Account tab within the app.
 - Access the Music Diary feature.
 - Here, you can view a record of the songs you've listened to each day displayed in a calendar format.
- Experiencing VR Soundscapes:
 - In the Discover tab, enter the VR Soundscape feature.
 - Experience immersive VR environments with music tailored to your mood and preferences.

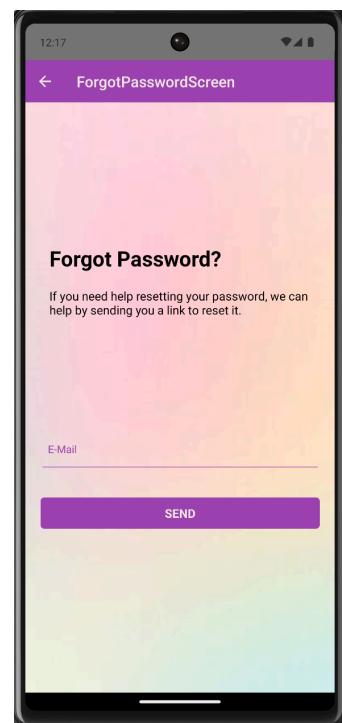
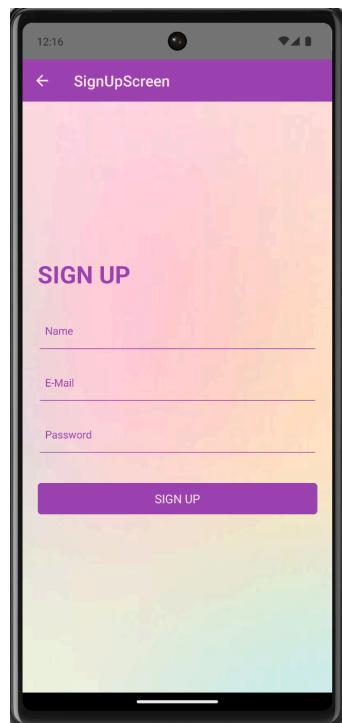
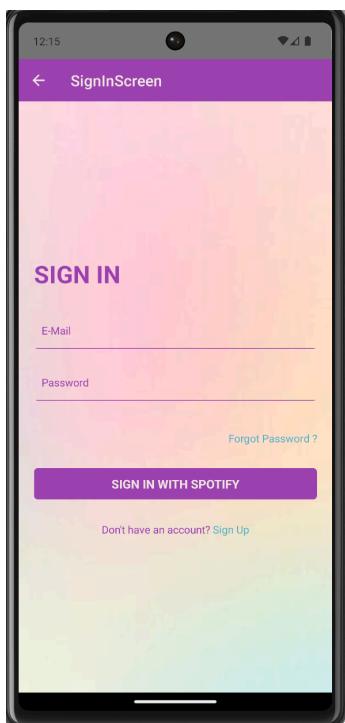
6.4 Appendix C — Final Design Guide for Light Theme

For Dark Theme, please refer to Figma Design via section 4.2

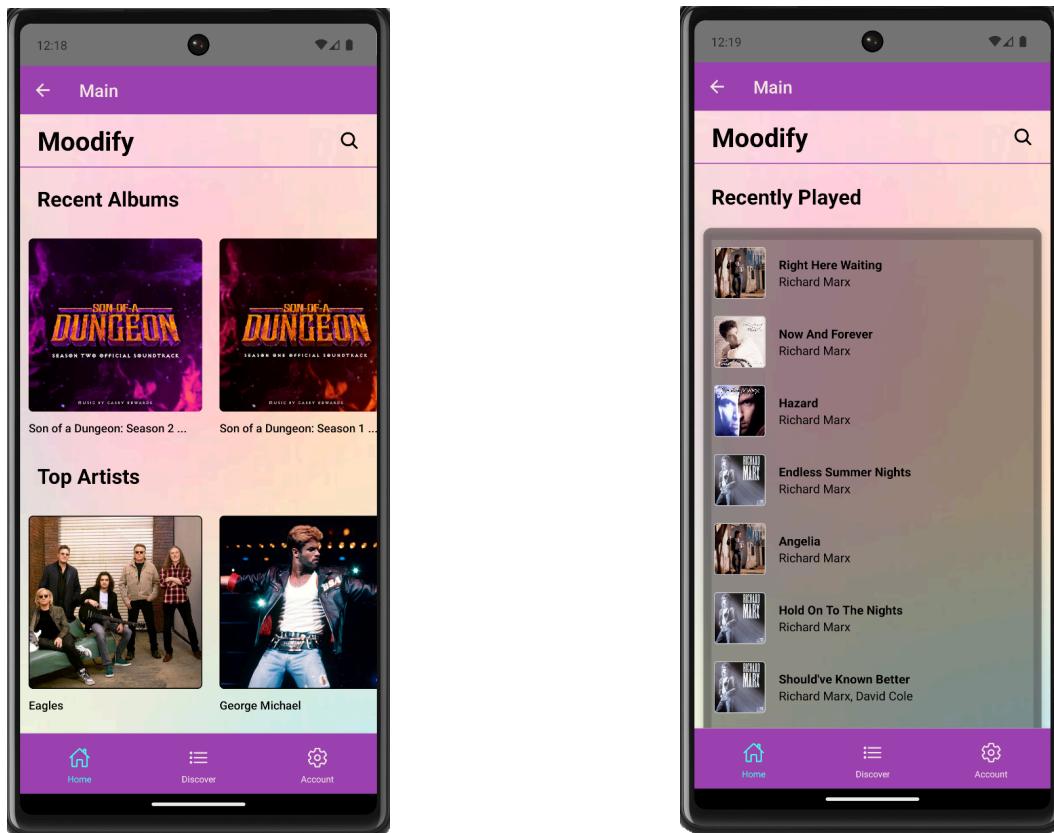
1. Onboarding Screen



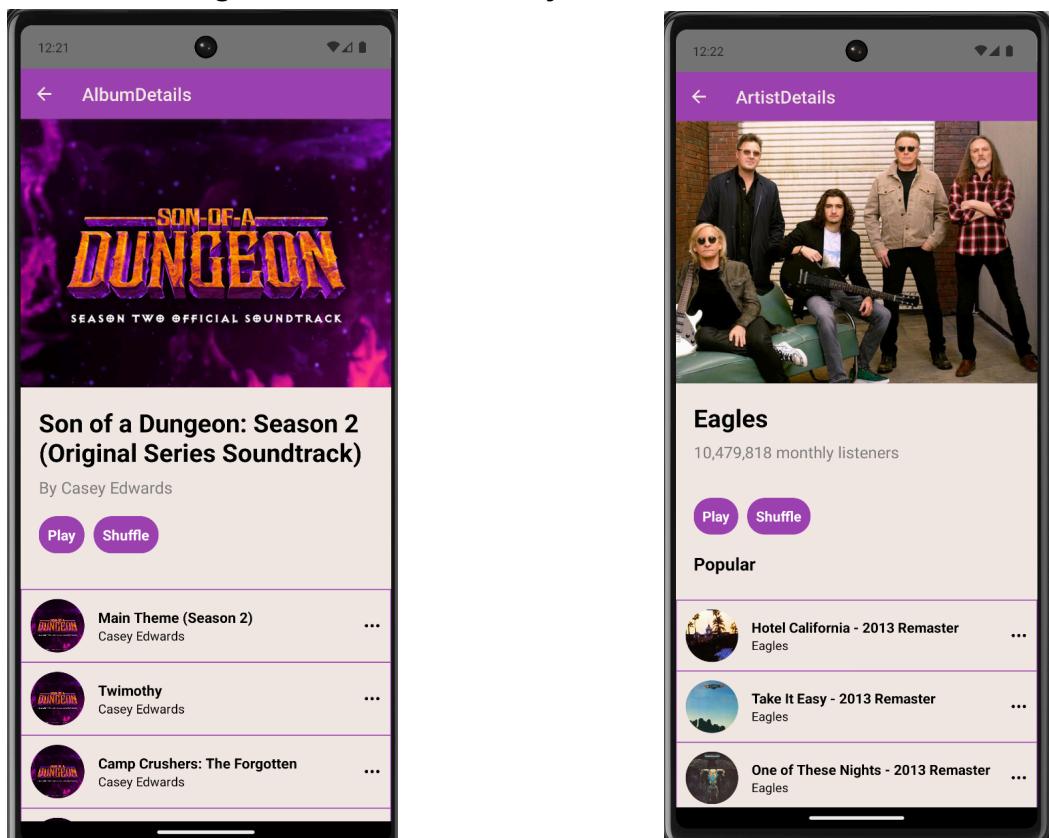
2. Log in and Sign Up



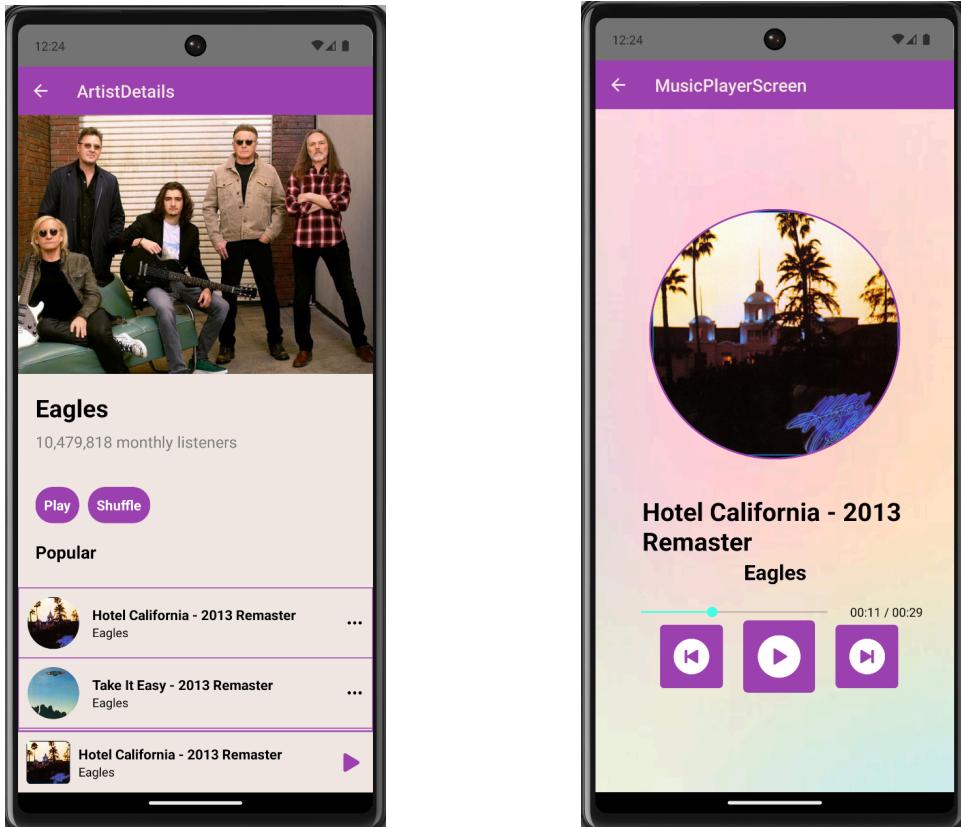
3. Home Screen



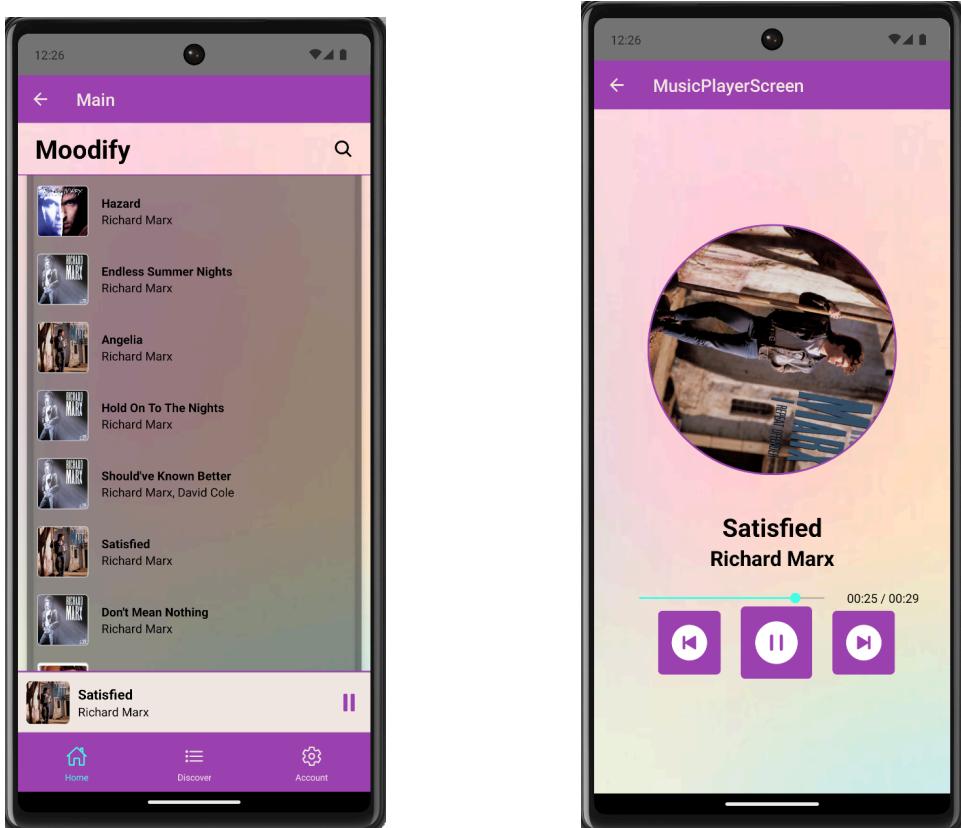
4. When clicking on Album/Artist card, you will see Album/Artist Screen



4.1 Here, you can play the songs with/without the music player screen

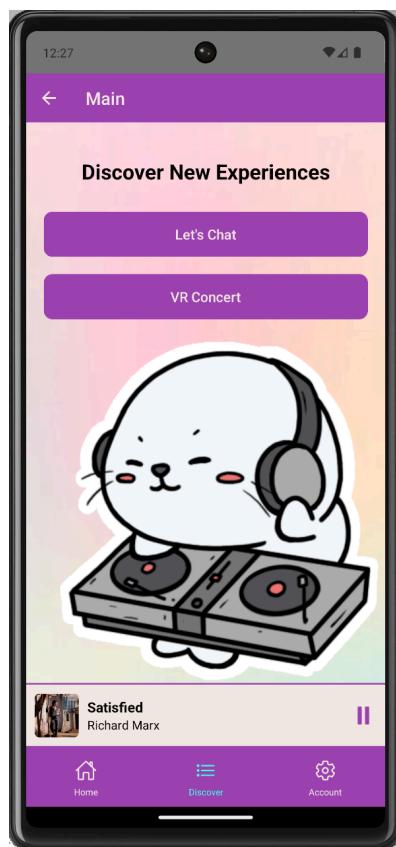


5. On the home screen's "Recently Played" section, you can also opt to use a song as the music player.



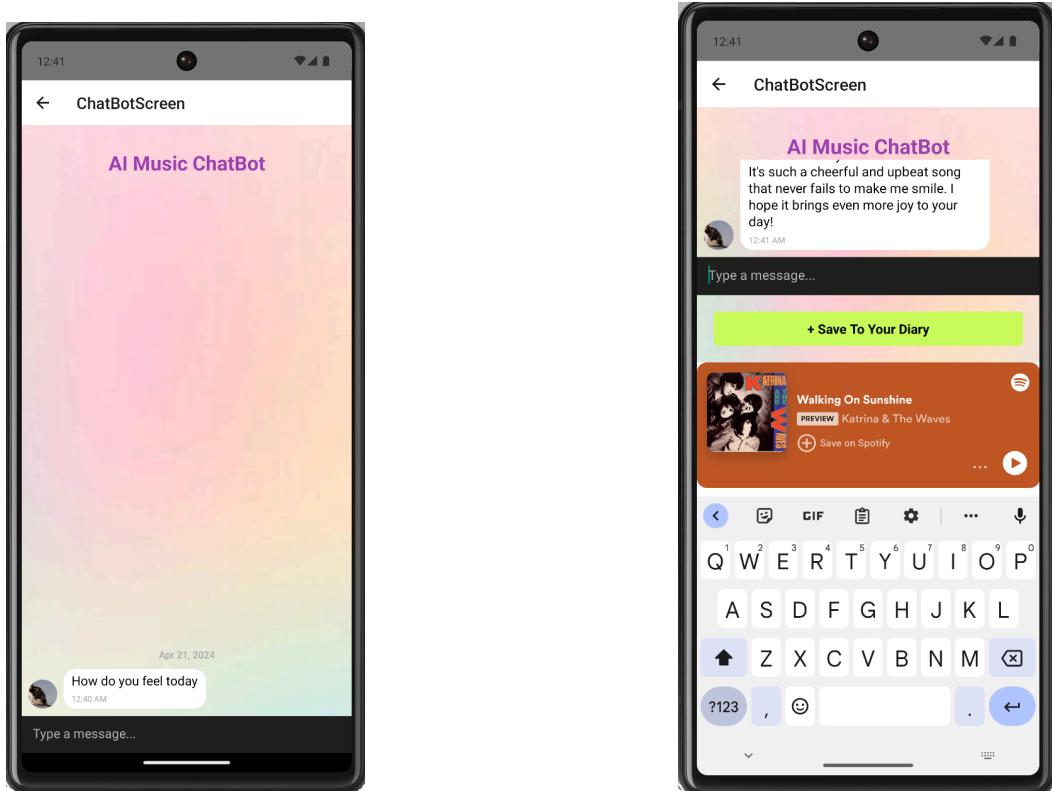
6. Discover Screen

This screen contains the navigation to the AI chatbot and VR concert screens respectively.



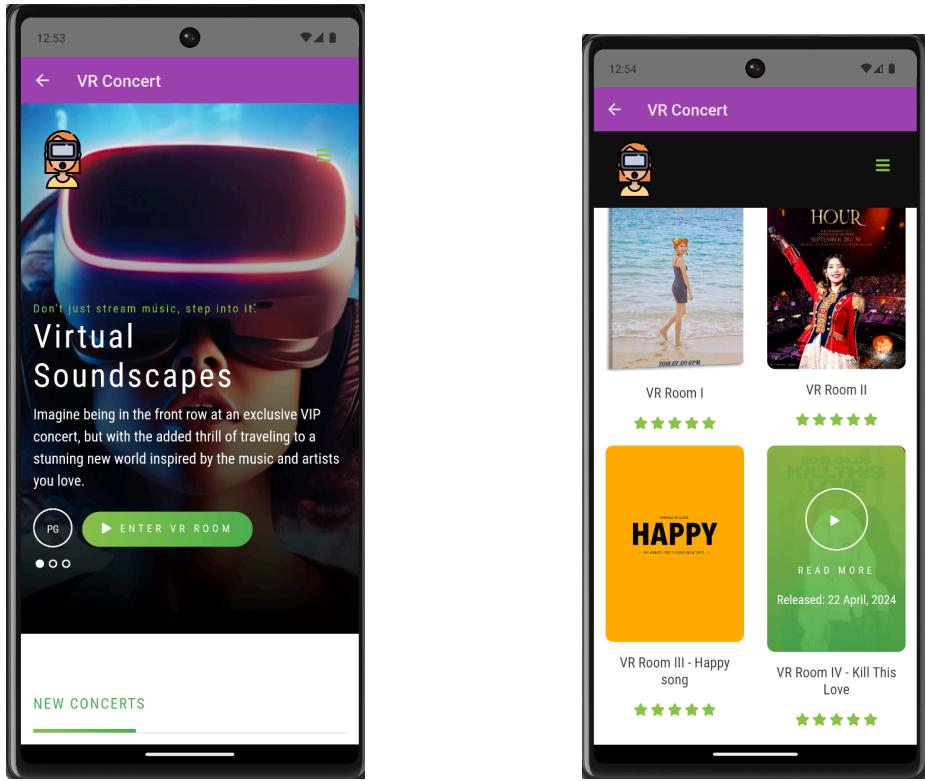
7. Chat screen

Once you input your mood, the AI chatbot will respond and suggest a song accordingly. You have the option to save these songs to your music diary.



8. VR Concert

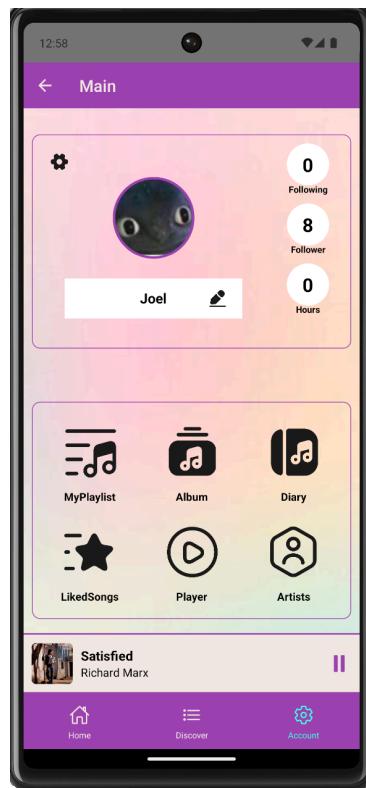
Upon entering the VR Concert section from the Discover tab, you can select to immerse yourself in one of the VR rooms. These rooms are customized to suit your profile.



8.1 VR Experience Screen



9. Account Screen



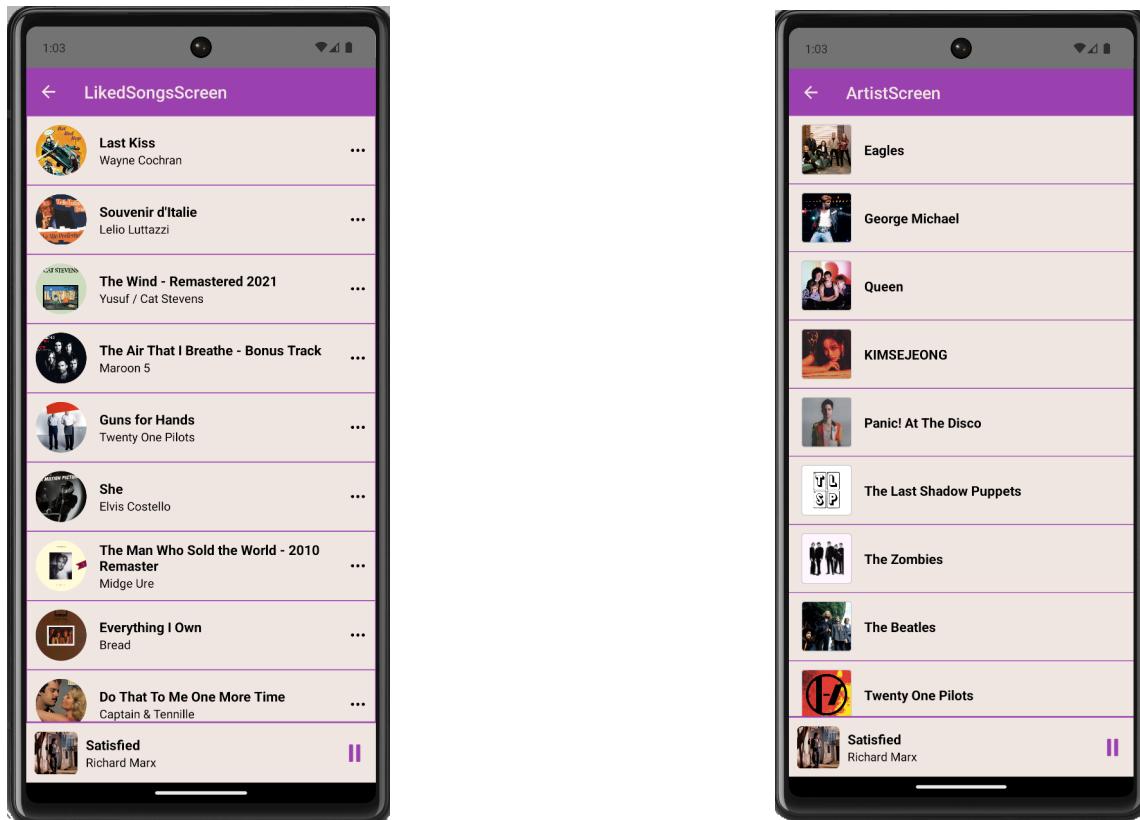
9.1 My Playlist and Album Screen:

Two side-by-side screenshots of the application. The left screenshot shows the "MyPlaylists" screen with a list of playlists: "vibe" by Joel, "Sex Education Official Playlist" by Spotify, "Sarah" by Joel, "Little Miss Sunshine (Original Motion Picture Soundtrack)" by Tori Gimblett, "Sophisti-pop 80s-90s" by Sonny Wisaksono, "00s Rock Anthems" by Spotify, "P" by Joel, "Jazz Classics" by Spotify, and "Robbie Williams | Swing Playlist" by Robbie Williams. The right screenshot shows the "AlbumsScreen" with a list of albums: "Vessel" by Twenty One Pilots, "Overexposed (Deluxe)" by Maroon 5, "Trench" by Twenty One Pilots, "Amnesiac" by Radiohead, "Kid A" by Radiohead, "amo" by Bring Me The Horizon, "OK Computer" by Radiohead, "華納廿三週年紀念精選系列 - Beyond" by Beyond, "Rubber Soul (Remastered)" by The Beatles, and "Satisfied" by Richard Marx.

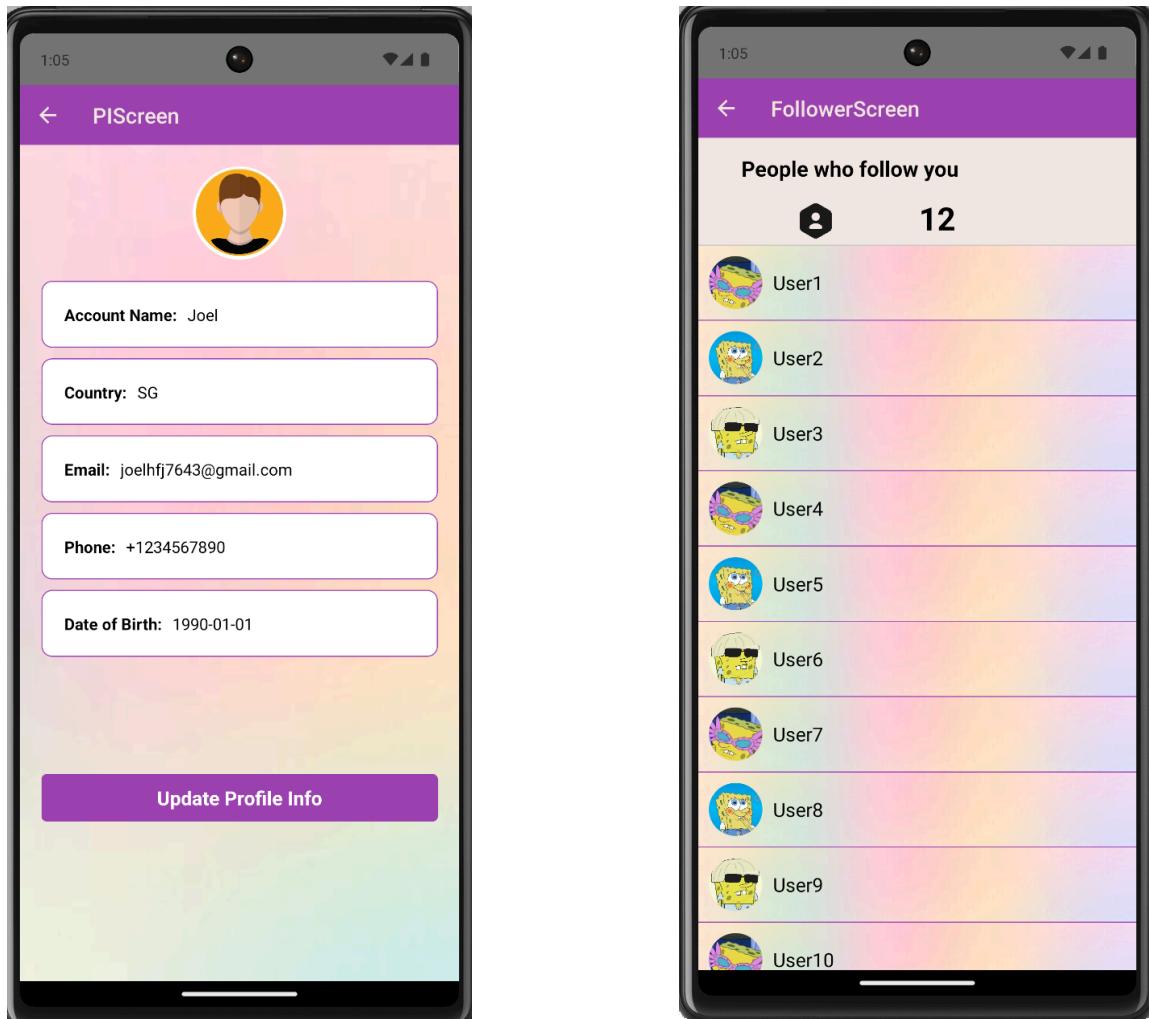
9.2 Music Diary Screen



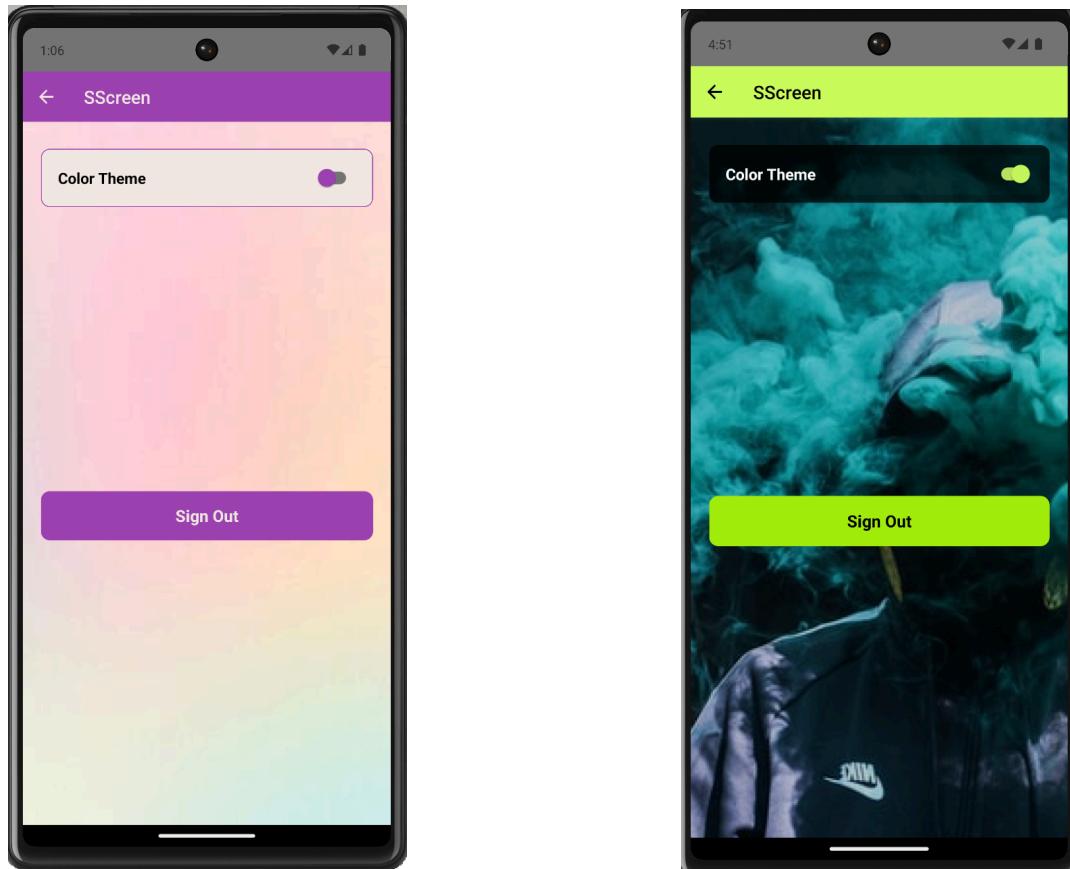
9.3 Liked Song Screen and Artist Screen



9.4 Update your profile info Screen and Follower Screen



9.5. Switch between light and dark theme



6.5 Appendix D — Source and Figma URL

All codes can be accessible via [link](#)

All Figma Design can be accessible via this [link](#)

6.6 References

[1] <https://explodingtopics.com/blog/music-streaming-stats>