A Project Report

On

"BAND-U"-A SOCIAL NETWORKING APPLICATION BASED ON MUSIC TASTE

BY

KRISHNA CHARAN GUDA

19XJ1A0226

SRUTHIJHA PAGOLU

19XJ1A0245

GAGAN CHANDRA JAINI

19XJ1A0219

Under the supervision of

Dr. Jayanth Lal Bhattacharya

HOD, Electronics and Electrical Engineering Department

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF

PR 402: PROJECT-TYPE COURSE

ECOLE CENTRALE SCHOOL OF ENGINEERING

HYDERABAD

(December 2022)

ACKNOWLEDGMENTS

We would like to express our gratitude and appreciation to all those who gave us the possibility to complete this project. Special thanks are due to our supervisor Dr. Jayanth Lal Bhattacharya whose help, stimulating suggestions, and encouragement helped us throughout the project and in preparing this report.

We would also like to acknowledge with much appreciation the crucial role of the staff in school laboratories, who gave us permission to use the lab equipment and software.

Last but not least, we thank our parents for financing our studies in this college and for constantly encouraging us to learn to engineer. Their personal sacrifice in providing this opportunity to learn engineering is gratefully acknowledged.

KRISHNA CHARAN GUDA

SRUTHIJHA PAGOLU

GAGAN CHANDRA JAINI

Ecole Centrale School of Engineering

Hyderabad

Certificate

This is to certify that the project report entitled "**SOCIAL NETWORKING APPLICATION BASED ON MUSIC TASTE**" submitted by Mr. KRISHNA CHARAN GUDA (HT No. 19XJ1A0226), Ms. SRUTHIJHA PAGOLU (HT No. 19XJ1A0245), Mr. GAGAN CHANDRA JAINI (HT No. 19XJ1A0219) in partial fulfillment of the requirements of the course PR 402, Project Course, embodies the work done by him under my supervision and guidance.

(Dr. Jayanth Lal Bhattacharya & Signature)

Ecole Centrale School of Engineering, Hyderabad.

Date: 30 December 2022

ABSTRACT

Music-based social networking applications are a growing trend in the digital world, providing a platform for individuals to connect with others who share a passion for music. These types of apps allow users to create profiles, follow other users, and share music-related content such as songs, playlists, and concert information. They can also offer features such as the ability to message other users, attend virtual concerts or events, and discover new music through personalized recommendations.

One major benefit of music-based social networking apps is the ability for users to discover and connect with new artists and music. These apps often include features such as personalized recommendations and the ability to follow other users who have similar musical tastes, which can expose users to new genres and artists that they may not have encountered otherwise. Additionally, music-based social networking apps can be a great way for users to connect with like-minded individuals and join online communities centered around music.

However, there are also potential risks and drawbacks to using music-based social networking apps. One concern is the issue of privacy, as these apps often require users to share personal information such as their name, location, and musical preferences. This information can be used to personalize the user experience, but it can also be accessed and potentially misused by third parties. It is important for users to be aware of the privacy policies of these apps and to take steps to protect their personal information.

In addition to privacy concerns, there are also potential issues with the content that is shared on music-based social networking apps. These apps can be a source of pirated music, as users may share copyrighted content without permission. This can lead to legal consequences for both the app and its users. There is also the potential for users to encounter inappropriate or offensive content, as there is often a lack of moderation on these platforms.

Overall, music-based social networking apps offer a unique way for users to connect and engage with music, but it is important for users to be aware of the risks and take steps to protect their privacy. These apps can be a great resource for discovering new music and connecting with like-minded individuals, but it is important for users to be cautious and to use these apps responsibly.

CONTENTS

1. INTRODUCTION

1.1 Motivation

There are many potential motivations for creating a music-based social networking platform. One of the main motivations is the desire to bring people together through a shared love of music. Music has the ability to bring people together and create a sense of community, and a music-based social networking platform could provide a space for people to connect with each other and share their passion for music.

Another motivation for creating a music-based social networking platform could be to provide a space for musicians and music fans to connect and collaborate. Many musicians struggle to find opportunities to connect with other musicians and industry professionals, and a social networking platform specifically focused on music could provide a place for these connections to be made. Additionally, such a platform could provide musicians with a way to share their music and promote their work to a larger audience.

A music-based social networking platform could also be a way to support and promote the music industry as a whole. With the rise of streaming platforms, many musicians have struggled to monetize their work and make a living from their music. A social networking platform that focuses on music could provide a way for musicians to monetize their work through things like merchandise sales, ticket sales, and other revenue streams.

Finally, a music-based social networking platform could be a way to support music education and promote the importance of music in society. Many people believe that music is an important part of human culture and can have a positive impact on people's lives. A social networking platform focused on music could provide a space for people to learn about different types of music, share their own musical experiences, and engage in discussions about the role of music in society.

1.1.1 Existing System

The existing system of dating apps is a way for people to meet and connect with potential romantic partners through the use of technology. Social networking applications are available as mobile applications or on the web, and allow users to create a profile, browse through potential matches, and message one another in order to arrange a date.

There are a wide variety of dating apps available, catering to different demographics, interests, and relationship goals. Some popular dating apps include Tinder, Bumble, Hinge, and OkCupid. These apps often use algorithms to match users based on common interests or preferences, such as location, age, and personal information provided in the user's profile.

In addition to traditional dating apps, there are also specialized apps for specific demographics or interests. For example, there are dating apps for LGBTQ+ individuals, as well as apps for people looking for relationships within their own religion or culture.

One benefit of the existing system of dating apps is that they provide a convenient and efficient way for people to meet and connect with potential partners. These platforms make it easy for users to browse through profiles and message one another, allowing them to get to know each other before meeting in person. This can be especially useful for people who may not have the opportunity to meet potential partners through traditional means, such as through work or social events.

Another benefit is that dating apps can provide users with access to a larger pool of potential partners than they may have access to through traditional means. This can increase the chances of finding a compatible partner and can allow users to connect with people who may not have otherwise crossed their path.

1.1.2 Drawbacks of Existing System

One potential drawback is that online dating can be time-consuming and overwhelming, with so many options available. It can be difficult to sort through all the profiles and messages to find someone who is a good match, and some users may feel pressure to constantly be swiping and messaging in order to increase their chances of finding a compatible partner. This can be tiring and stressful.

Another potential drawback is that relying on dating apps can lead to unrealistic expectations. Many people create carefully curated profiles and put their best foot forward when using these platforms, which can lead to disappointment when the reality of the person does not match the online persona. Additionally, it can be difficult to get a true sense of someone's personality and compatibility through online communication alone, leading to mismatched expectations and misunderstandings.

There is also the risk of encountering scammers or individuals with ulterior motives when using dating apps. These individuals may use fake profiles or try to extract personal

information or money from users. It's important to be cautious and protect your personal information when using these platforms, and to be aware of the signs of potential scams.

Another potential drawback is that using dating apps can lead to a lack of face-to-face interaction and genuine connection. While these platforms can be a convenient way to meet people, they may not provide the same level of intimacy and genuine connection that can be found through more traditional forms of dating and meeting people.

Finally, some users may feel pressure to present a certain image or conform to certain expectations when using dating apps. This can lead to users feeling like they have to constantly be "on" and putting on a show in order to attract potential partners, which can be exhausting and lead to a lack of authenticity.

1.1.3 Proposed System

The solution we are proposing is a mobile app called "Band-U" that is designed to address the limitations mentioned by focusing solely on music preferences. This app is available on both the iOS and Android platforms and requires an internet connection to function. The purpose of "Band-U" is to provide a set of software that can be used on various mobile devices, such as smartphones and tablets.

(1) User profiles: Users would create profiles by providing information about themselves, including their music preferences and other interests. Users could also upload photos and other media to their profiles. To create a sense of community and encourage users to be authentic, the app could include a feature that allows users to write a short bio about themselves and their interests.

(2) Music preferences: Users would be able to select their favorite music genres and artists, and the app could use this information to create a "music profile" for each user. This profile could be used to match users with other users who have similar music tastes, as well as to suggest music-themed activities and events in the user's area.

(3) Matching algorithm: The app would use an algorithm to match users based on their music preferences, as well as other factors such as location and age. Users could specify their preferred age range and location, and the app could use this information to suggest potential matches.

(4) Messaging: Users would be able to message one another within the app to get to know each other and potentially arrange an interaction. To encourage genuine communication and connection, the app could include prompts or icebreakers to help users start conversations.

(5) Verification: In order to increase safety and trust, the app could include a verification process for users. This could involve verifying users' identities through social media accounts or other means. Verified users could be marked with a special badge, which could help to build trust and encourage users to be more authentic.

(6) Moderation: To ensure that the app is a safe and welcoming environment for all users, there should be a moderation team in place to monitor activity and remove any inappropriate or malicious content. The app could also include a reporting feature that allows users to flag any inappropriate or suspicious activity.

(7) Premium features: The app could offer premium features for users who want to enhance their experience on the platform. These features could include the ability to see who has liked their profile, access to more advanced matching algorithms, or the ability to message other users without a mutual match. Users could purchase premium features through an in-app purchase or subscribe to a premium membership.

Figure 1: Band-U Logo

1.2 Theory and Tools

A social networking mobile application (app) is a software application designed to facilitate communication and interaction between individuals who are seeking romantic relationships. These apps often utilize swiping mechanics similar to the popular dating app, Tinder, where users can view profiles of potential matches and "swipe" right if they are interested or "swipe"

left if they are not. If two users "swipe" right on each other's profiles, it is known as a "match" and the app will facilitate communication between the two users.

1.2.1 Application

A mobile application is a software application that is meant to run on a mobile device, such as a smartphone or tablet. Mobile apps are often designed specifically for the platform on which they are intended to function, such as Android for devices running Google's Android operating system or iOS for Apple devices.

Mobile apps can serve a wide range of purposes, from providing useful tools and information to entertainment and socialization.

1.2.2 Android

Android is a mobile operating system (OS) developed by Google for use on smartphones, tablets, and other devices. Android was first released in 2007 and has since become one of the most popular mobile OSes in the world.

One of the main features of Android is its open-source nature, which allows developers to freely modify and distribute the OS. This has led to the creation of a large and diverse ecosystem of apps and devices that run on Android.

Android is designed to be user-friendly and customizable, with a variety of features that allow users to personalize their devices to suit their needs. These features include a customizable home screen, the ability to install and run third-party apps, and support for different types of media and communication.

Android devices come in a wide range of shapes and sizes, including smartphones, tablets, and other types of devices. These devices are made by a variety of manufacturers, including Samsung, Google, Huawei, and OnePlus, among others.

1.2.3 iOS

iOS is a mobile operating system (OS) developed by Apple Inc. for use on its iPhone, iPad, and iPod touch devices. iOS was first released in 2007 and has since become one of the most popular mobile OSes in the world.

One of the main features of iOS is its focus on user experience and design. Apple has put a lot of effort into making iOS easy to use and visually appealing, with a user interface that is intuitive and responsive. iOS also includes a variety of built-in apps and features, such as a messaging app, email client, and web browser, as well as support for third-party apps that can be downloaded from the App Store.

iOS devices are known for their high-quality hardware and attention to detail. Apple's iPhone and iPad devices are among the most popular in the world and are known for their performance, camera quality, and sleek design.

Unlike Android, which is open-source and can be modified and distributed freely by developers, iOS is a closed platform that is only available on devices manufactured by Apple. This means that iOS users are limited to the hardware and software options that Apple provides, but it also means that iOS devices are generally more secure and receive regular updates and support from Apple.

1.2.4 Software Development Kit

A software development kit (SDK) is a set of tools and resources that developers can use to create applications for a specific software platform or operating system. An SDK typically includes a set of APIs (application programming interfaces), documentation, and code samples that developers can use to build applications.

SDKs are commonly used by developers to create applications for mobile operating systems, such as Android and iOS, as well as desktop operating systems, such as Windows and macOS. They can also be used to create applications for gaming platforms, such as PlayStation and Xbox, and other types of software platforms.

In addition to APIs and documentation, an SDK may also include libraries, sample code, and other resources that developers can use to build their applications. Some SDKs may also include tools for testing and debugging applications, as well as tools for deploying and distributing applications to users.

1.2.5 Application Programming Interface

An application programming interface (API) is a set of programming instructions and standards for accessing a web-based software application or web tool. A software company releases its API to the public so that other software developers can design products that are

powered by its service. Using an API, a developer can access the features or data of an application or service and use them in their own application.

APIs consist of a set of functions and protocols that developers can use to interact with an application or service. They allow developers to build applications that can communicate with and make use of the functionality provided by other applications or services. For example, a developer might use an API to access data and use it to add music preferences to their own application.

APIs can be implemented using various technologies and protocols, such as REST (Representational State Transfer), SOAP (Simple Object Access Protocol), and JSON (JavaScript Object Notation). These technologies and protocols define the way that different applications and services can communicate with each other and exchange data.

1.2.6 Android Studio

One of the main tools used for Android development is Android Studio, which is a comprehensive development environment provided by Google. Android Studio includes a range of features and tools that make it easier for developers to create, test and debug Android applications. It includes a code editor, debugger, emulator, and other tools that are commonly used in the development process.

1.2.7 Figma

Figma is a cloud-based design and prototyping tool that allows teams to create and collaborate on design projects in real-time. It is commonly used by web and app designers, as well as product managers, marketers, and other professionals who need to create and share visual designs.

One of the main features of Figma is its collaborative nature. Figma allows multiple users to work on the same design project at the same time, making it easy for teams to collaborate and share ideas. It also includes a range of tools and features for creating and editing design elements, such as shapes, text, and images.

1.2.8 Visual-Studio Code

Visual Studio Code (VS Code) is a popular source code editor developed by Microsoft. It is available for Windows, macOS, and Linux and is commonly used by developers to write and edit code for a variety of programming languages.

VS Code includes a range of features that make it a powerful tool for developers, including a code editor, debugger, and integrated terminal. It also includes support for version control systems such as Git, as well as integration with a variety of other tools and services, including cloud-based development platforms like Dart.

1.2.9 Flutter (Dart)

Flutter is an open-source mobile application development framework developed by Google. It is designed to allow developers to build high-quality, cross-platform mobile applications for Android and iOS using a single codebase.

One of the main features of Flutter is its fast development cycle. It includes a hot reload feature that allows developers to make changes to their code and see the results in real-time, without having to rebuild and redeploy the app. This makes it easier for developers to iterate on their designs and test new ideas.

Flutter is written in the Dart programming language, which is easy to learn and includes features like static type checking and a rich set of libraries. It is also fully integrated with popular IDEs (Integrated Development Environments) such as Android Studio and Visual Studio Code, making it easy for developers to get started with Flutter.

2. REQUIREMENTS AND SYSTEM SPECIFICATIONS

2.1 Data Requirements

The set of data required for this project is user credentials for registering and logging into the application. Music application credentials for extracting data to the application. Without this information, the application cannot process profile suggestions as the data is not extracted from the cloud database.

2.2 Functional Requirements

Functional requirements are characteristics that must be present in the final system. In the case of a mobile application, the user must be able to download the app from a store like the Play Store. The app may be free or require payment depending on the source. To access the app, the user must register and log in with their login information. Once logged in, the user can use all the features of the app.

2.3 Performance Requirements

When designing a system, response time, scalability, platform dependencies, and tolerance should be considered as performance requirements. The system should respond quickly to user input and be able to handle new features as the complexity of the application increases. It should be compatible with the specified hardware and software requirements and have a high tolerance for faults such as network or connectivity issues, or crashes. The system should also be able to notify the user if it is unable to provide results when requested due to these issues.

2.4 System Requirements

The application should be installed on a device or system in a way that meets the basic requirements of the device, such as compatible hardware and software, access to built-in software like the camera on a mobile device, internet permissions, and protection against potential security threats like viruses or malware.

2.5 Testing and Maintainability Requirements

The application should be able to pass all potential test cases, both positive and negative, in a test environment. It should be developed to be stable and not crash when in use, and be able to accommodate new functions or code expansion without issue.

3. SYSTEM DESIGN

3.1 Design Approach

The functional design approach is the foundation of this project. It allows for a clearer understanding of the project's design by breaking it down into manageable parts, such as flow, use cases, and implementation. The project is composed of multiple modules, each with its own specific function and possibly other sub-functionalities or modules.

All the modules are implemented and integrated together to make a flawless working application.

3.2 Detailed Design

The detailed design including modules and sub-modules of the application is as follows:

3.2.1 User Registration

If the user wants to use the Band-U application, then they must download from the app store, install and register it by providing login information. Once, they register the registered information is stored on the cloud server and can be validated, by clicking the valid credentials the next time they logins in with the application.

3.2.2 Music Application Connection

When the user logins with the credential, they need to register the music application to extract data about the user. The Music application can be Spotify, Apple Music, Amazon Music, Wynk, and Youtube Music. Once they connect the third-party application to Band-U, they cannot use it for another account of Band-U. De-registration can be done with the help of the support team.

3.2.3 Filtering

Profiles can be searched by filtering the genre, artists, albums, or composers. The effective profiles are displayed on a priority basis to connect. These filters can be changed anytime.

3.2.4 Messaging

Messaging option is available when the profile is swiped right. Texting, sending stickers and GIFs, and sharing music are possible. The chats are stored and encrypted in the cloud database.

3.3 Application Design

The main aim of the system design is to match profiles. The profiles can be made when a third-party music application is connected and data is extracted using the API. The matches can be monitored by the user at any time.

3.4 User Interface Design

The user interface of the application is kept simple. few clicks can navigate to deeper parts of the applications without any hassle. Every navigation pointer also explains where it navigates which helps the user to navigate freely.

The UI Design is done on Figma where all the navigation can be done easily. The first registration and login screens are kept very simple without any extra unwanted requirements. Basic colors like black and green are used. Next is the home page where other profiles can be seen. Settings can be navigated from the bottom, where they need to connect to the music application for data extraction. The music applications and colors for those applications resemble each other. Messages which come after matching is kept very simple to choose to chat. Every option that needs while chatting such as textbox, send button, stickers, and documents sending is mentioned clearly.

3.5 Database Design

The database design should be easy where data should be easily accessed and manipulated. Database manipulation includes adding, deleting, and updating the values of the user. Firebase Firestore is the database offered by Google and is used in this application for better security. The database server for the client can be installed by signing up with a username and password.

4. IMPLEMENTATION

4.1 Technologies Used for Implementation

As the project is developing an iOS-Android Application, the best and most effective language used is Dart. The VS-Code is used to write the program for the application. Android Studio is used simultaneously to watch the progress of the application. The code is written according to the Figma design created for the application. The front end is dependent on flutter. The user data is stored in the Firebase Firestore cloud services. The Music application data is retrieved using the corresponding APIs. That music data is stored in the same server as the credentials of the user are stored.

1. Figma
2. Flutter
3. Firestore
4. Canva
5. SDK
6. API
7. HTML
8. Canva
9. JSON
10. Build Gradle

5. SCREENS AND FEATURES

5.1 Screens

In this project, there are mainly of four screens named login screen, home screen, messaging screen, and settings screen.

5.1.1 Login Screen

This screen is used to log into the application. If the user is new to the application, then the registration can be done from the same screen. Registration spots can be navigated with the help of buttons on the screen. When credentials are forgotten, they can be retrieved by clicking on the “forgot” option.

5.1.2 Home Screen

The other user’s music preferences and data can be viewed. The matches can be done according to the liking of the profile. Filtering of the profiles and preferences can be set here. This home screen acts as the landing screen and the home screen itself.

5.1.3 Message Screen

The matched profiles can be viewed here. Texting with the profiles is enabled on this screen. The messages are stored on the cloud. The track of the swiped profiles can be viewed here. The interest of that user can be changed after going into the “swiped profile” section.

5.1.4 Settings Screen

Own profile can be seen. The connection of the third-party music application is done on this screen. When clicking on connect of that specific music application, a pop-up screen appears where you need to use your credentials of that music application to make the connection with this Band-U. Only one music application can be connected at a time. Logging out of the application can be done on this screen. De-registering of music application to Band-U is processed here.

5.2 Features

Each screen has its unique features and its own work. One screen is connected to another using navigation bars. Every feature on every screen is explained below.

5.2.1 User Login/ Registration

If the user wants to use the Band-U, they must download the application, and install and register it by providing login information. As shown in Figure 2, the login information includes the user name and password. For the new user, the user must register by providing an Email Id and password as in Figure 3. Once, the registration is done, the information is stored on the server and can be validated, checking for valid credentials for the next time he logins with the application. After using the application, log out can be processed. When the user forgets the credentials, the password can be changed by using the “Forgot Password” widget.

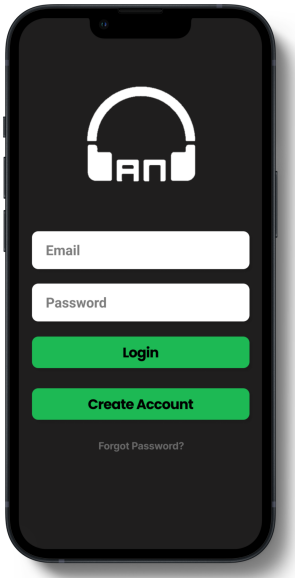


Figure 2: Login Screen

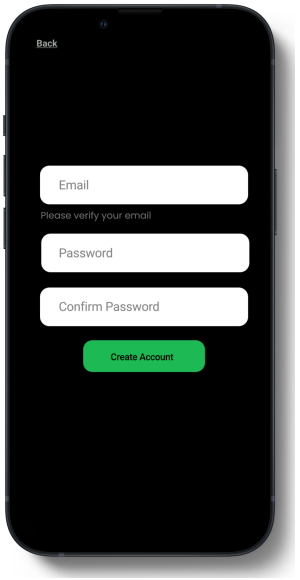


Figure 3: Registration Screen

5.2.2 Landing Screen/ Home Screen

As soon as logging in, the profiles of the other users can be viewed. But to swipe on those profiles, the current user must be logged into a music account from the settings page. If swiped right, that implies that you are interested in that profile, and when swiped left it indicated you are not impressed or interested in that profile. Other user profile picture and name can be viewed on the top. While scrolling down, three widgets named “4 Weeks”, “6 Weeks”, and

“All Time” as shown in Figure 4 display the corresponding songs and artists who listened to them in those periods. There is a search column where the user can select up to one artist or song or composer to make a priority while viewing others' profiles as shown in Figure 5. When all profiles are swiped and left with no profile, “No more Users :)” is displayed as in Figure 6. When a match is done then you get a message “Yayy! You have a match” as in Figure 7. Swiping can also be done by clicking the heart shape to swipe right and cross mark to swipe left as in figure 8.

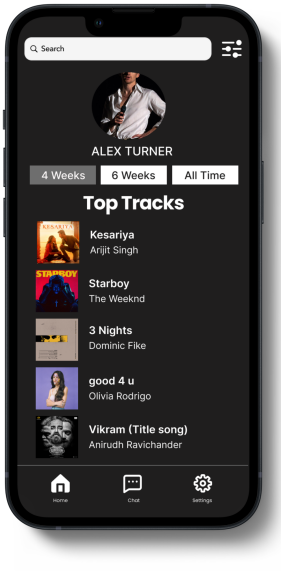


Figure 4: Home Screen

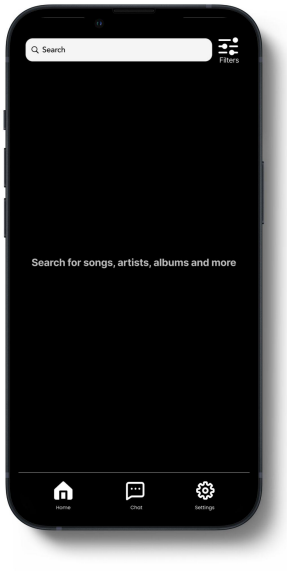


Figure 5: Search

Figure 6: No Users Left

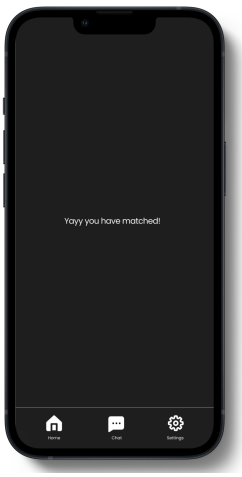


Figure 7: Matched

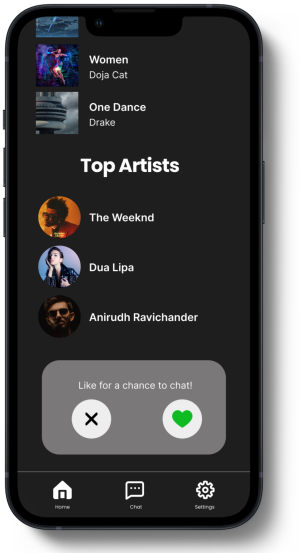


Figure 8: Swipe Alternate

5.2.3 Filters

A widget named “Filters” is located on the right top of the home screen which helps to select multiple artists or albums or genres or composers or compilations as shown in Figure 9 and Figure 10. This is used as a priority while showing other users' profiles. These filters can be altered or removed at any time.

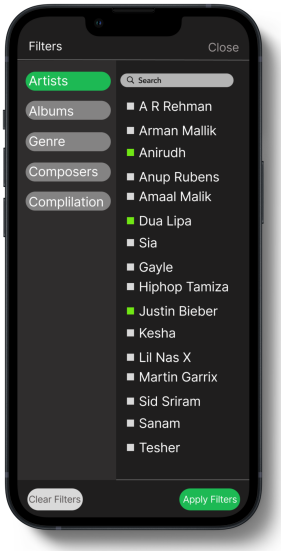


Figure 9: Filters

Figure 10: Filter Options

5.2.4 Message Box

In the message section, i.e., the Chat box, the matched profiles can be seen as shown in Figure 11. No profiles can be seen when there are no matches and “Get to Swiping” is displayed as shown in Figure 12. In the profiles message box, the user can be able to text, add emojis, capture and upload photos, send stickers, and share music as in Figure 13. When a match is done then you get a message “Yayy! You have a match” as in Figure 14. The messages cannot be undone or deleted. All these messages are stored in the cloud, such that your data will be accessed when you log in through any device.

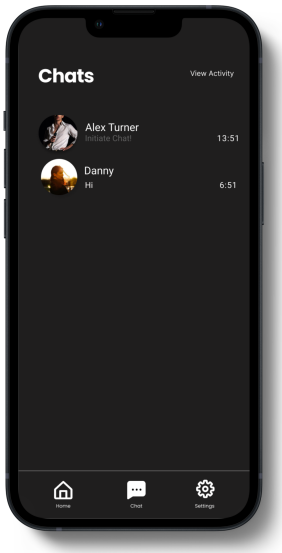


Figure 11: Profiles

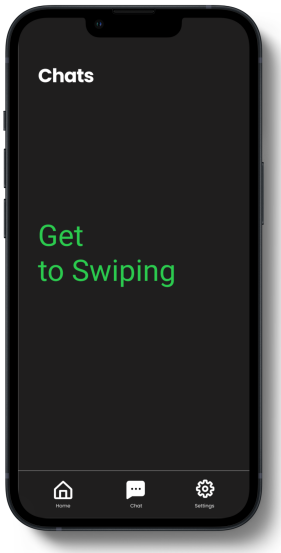


Figure 12: No Matches

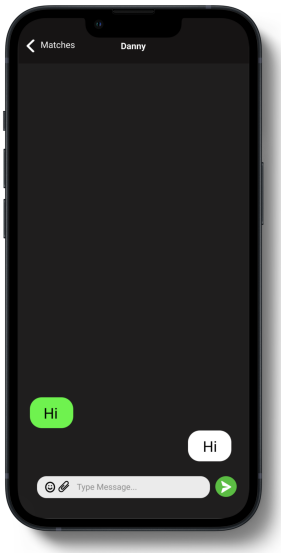


Figure 13: Chat Box

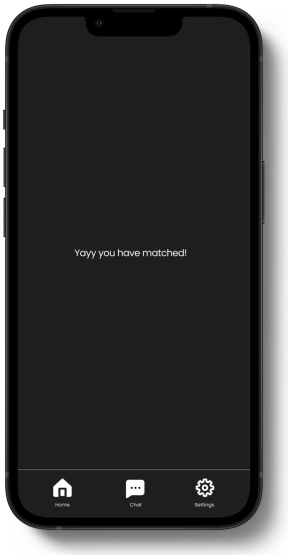


Figure 14: Matched

5.2.5 Activity Section

Users can have full access to their own activity. The swiped profiles can be seen in activity as in Figure 15. That swiped result can be altered anytime, but only twice. A confirmation page is shown to confirm the decision. The process is shown in Figure 16.

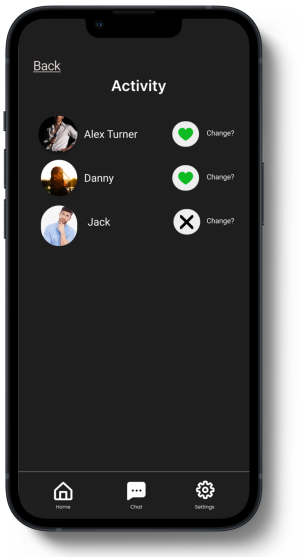


Figure 15: Activity

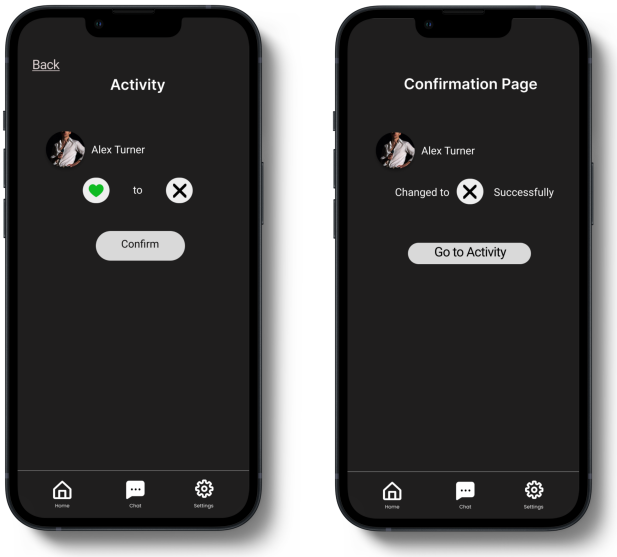


Figure 16: Activity change confirmation

5.2.6 Settings Page

The user needs to log into the music application from this screen. Multiple music logins are given and the user needs to choose only one as in Figure 17. He can access data from only one application at once. When clicking on the music connectivity widget, for example, Spotify, Spotify's login screen pops up as redirected URI as in Figure 18. When logged in by entering that application's credentials, the data is stored in the cloud server of that user and refreshes at every instant. Users can view their own profile from the "View Profile" widget. The user is free to log out at any time or disconnect from the music account for reconnecting again or reconnecting with another application as shown in Figure 19.

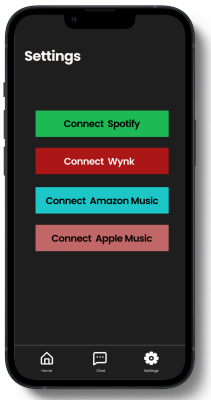


Figure 17: Connectivity

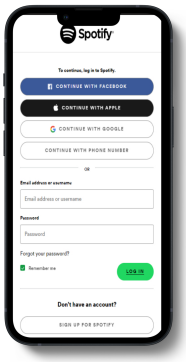


Figure 18: Spotify login

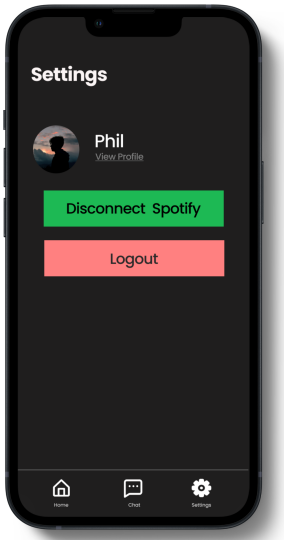


Figure 19: Signing Out

6. DATA MANAGEMENT

6.1 User Data Storage

The user's data is stored in separate servers and has a unique UID. All the texts and media are stored in that same server but encrypted. The user's profile selection and activity are kept confidential and can be accessed by only that user.

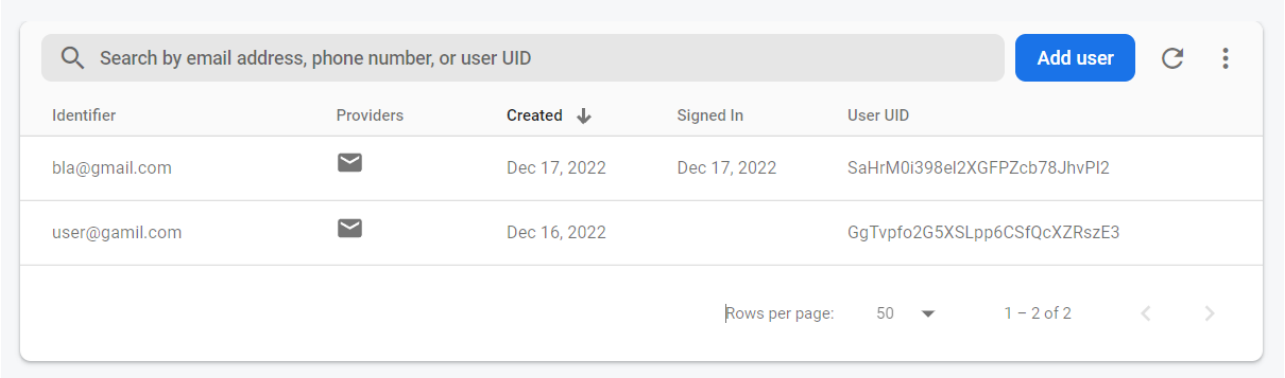


Figure 20: Firebase

6.2 Music Application Data

APIs are used to integrate music applications into Band-U. Each application has its own URI. No credentials for music applications are stored. The URI forwards to authenticate the data of the client. When the login into the music application is done, the data is transferred to the cloud base of Band-U. This data is then used to show your profile to other users.

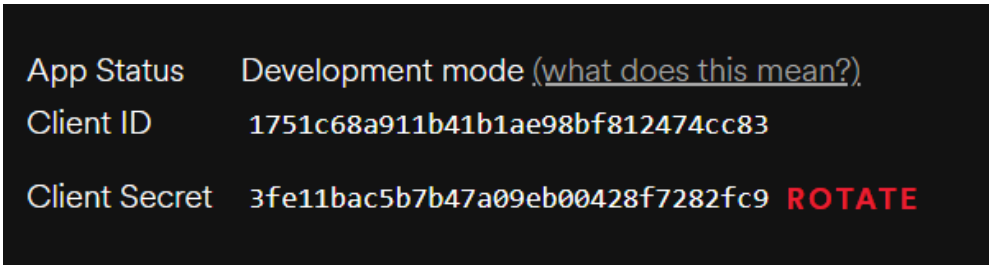


Figure 21: Spotify Developer Mode

6.3 Matching

Swiping is done to decide whether the user has any interest in that profile or no interest. When the swiping is done, the client's decisions are stored in the cloud base. If another client matches that profile, then the server reads the data and confirms a match, and displays it on the screen. If the client changes the decision from the activity section, the server updates and show the matches accordingly.

6.4 Filtering

Filtering is done by selecting the options provided. The data is read and analyzed in the background to show the profiles consisting of that data. This filtering algorithm is used to narrow down to a few users from many. These filtered profiles are shown as a priority and then other profiles come onto the screen.

7. WORKING

7.1 Application Design

The UI of the application is done on Figma. The link to the design of the application is mentioned below.

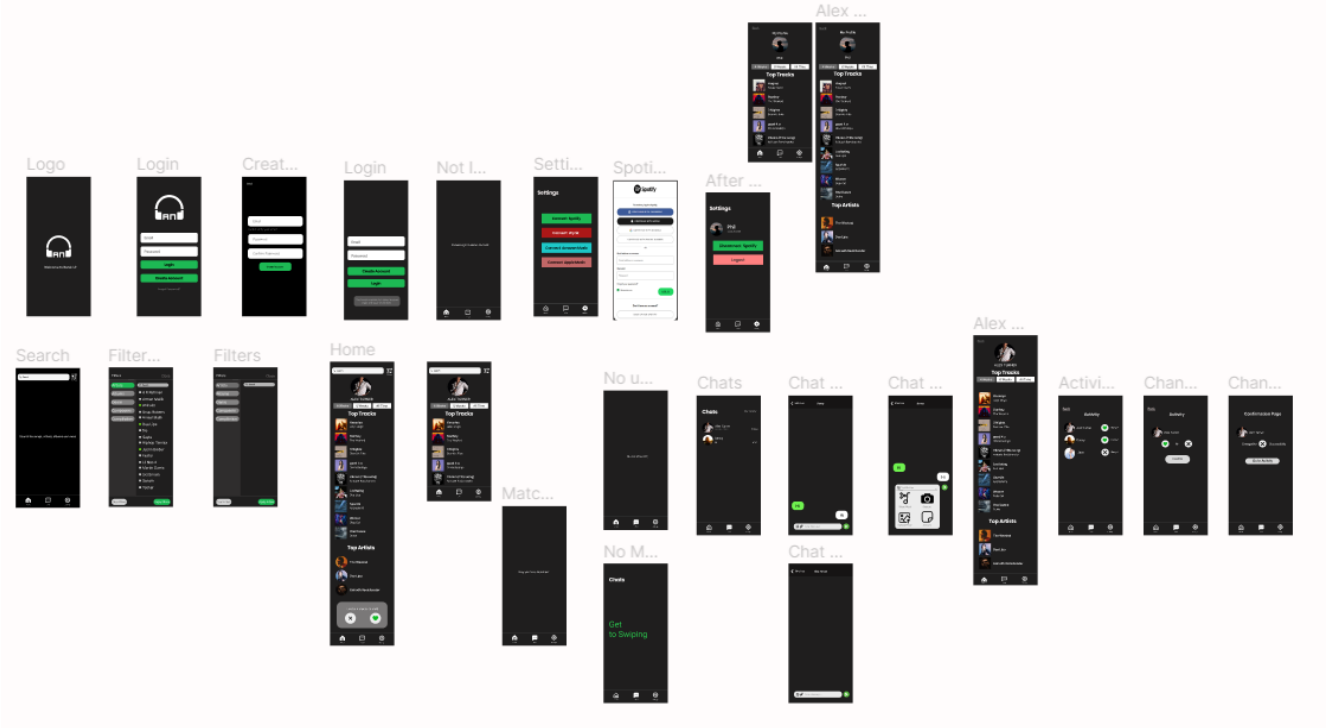


Figure 22: Figma file snippet

https://www.figma.com/file/qTKZmrc0Qu7pEnxxEOreny/BAND?node-id=0%3A1&t=pLcNVEavnS7fKyJf-0

7.2 Application Frontend

The program is written using Dart, which is used to work on both iOS and Android devices. The UI is written with the same program. This front-end is connected to the backend for data storage and retrieval. The link for the front-end of the application is mentioned below.

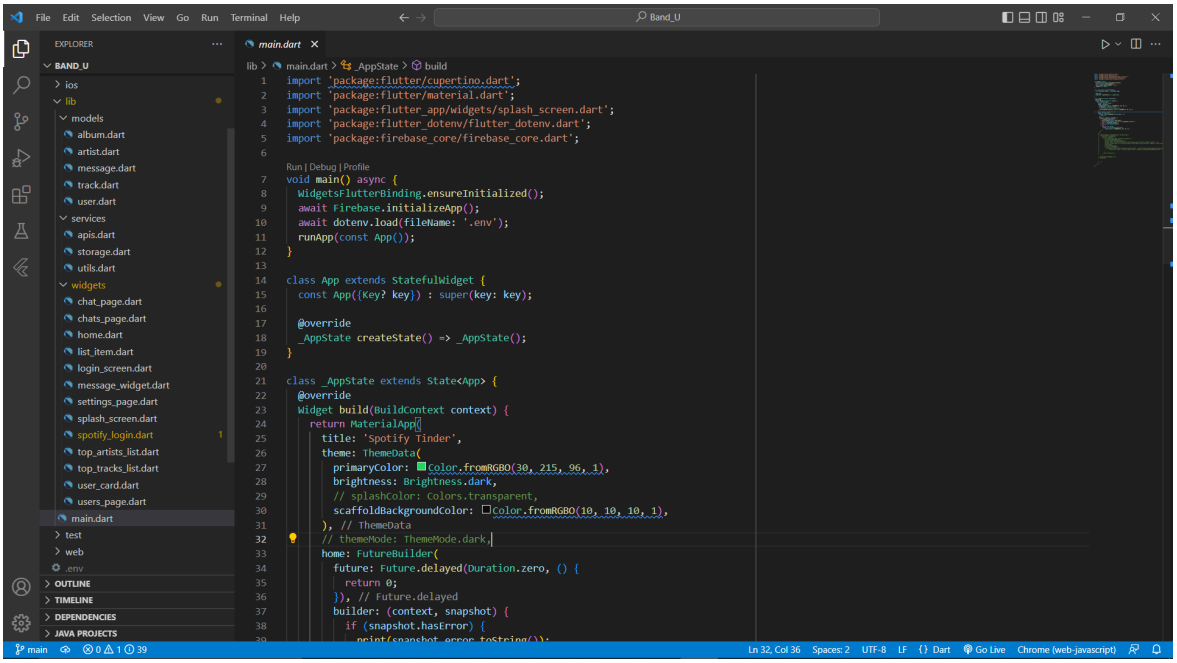


Figure 23: Program code snippet

https://github.com/Kr1shnaG/Band_U.git

7.3 Application Backend

The application backend is Firebase Firestore. Firebase is one of the trusted and safe cloud services available in the market. The link to the backend of the application is mentioned below.

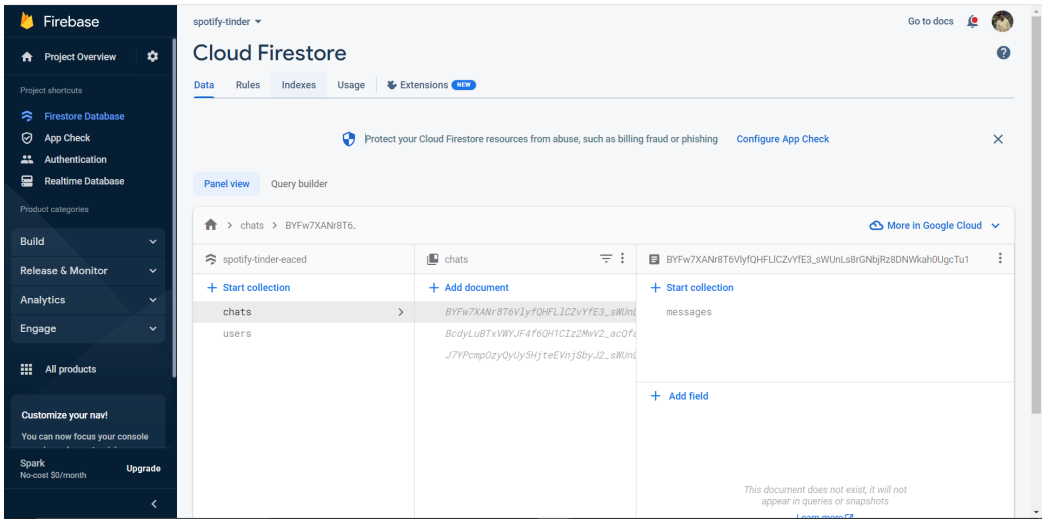


Figure 24: Firebase Firestore snippet

https://console.firebase.google.com/project/spotify-tinder-eaced/firestore/data/~2Fusers

7.4 Spotify API

The Application works on the API of music applications. One of that applications is Spotify. The data from Spotify is stored in the cloud. The link to the API of Spotify is mentioned below.

Figure 25: Spotify user track

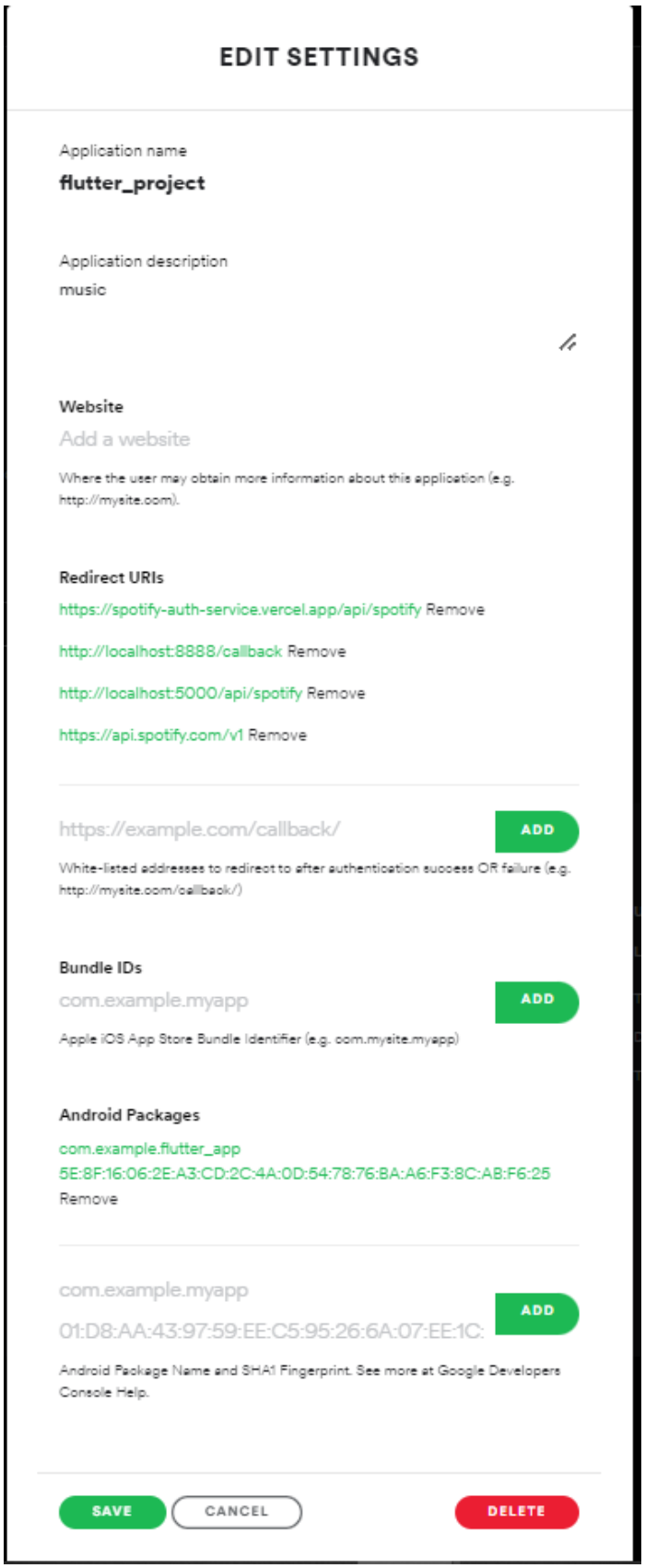
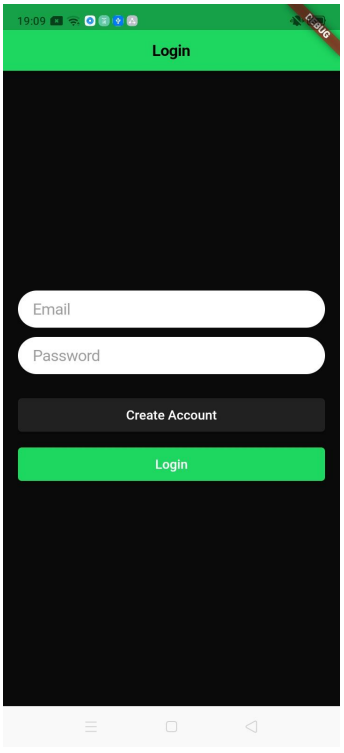


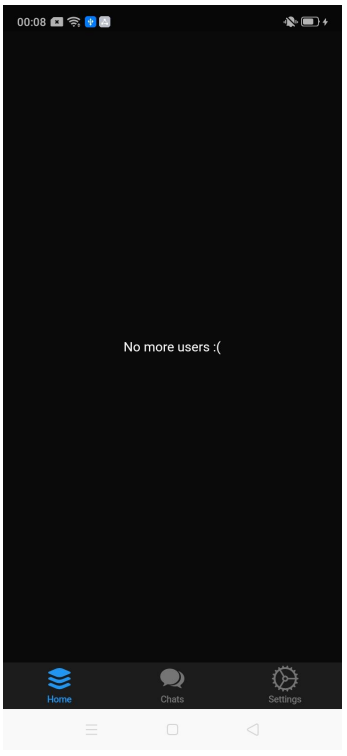
Figure 26: Spotify URI for API

https://developer.spotify.com/dashboard/applications/1751c68a911b41b1ae98bf812474cc83

7.5 Application Working

The Application works on Android and iOS devices. Firstly Login or Registration is done. Forwarding to the connection of the music account. The other user's profiles can be seen. Filtration can be done for narrowing down the profile search. Swipe according to the interest. When matched, the profile appears in the message box. Chatting can be done. Swipe activity can be altered. Can connect with another music account, but need to deregister the existing account. Only one account can be accessed at a time. Sign out when needed. The credentials are safely saved on the cloud. The link to the snippet of the working of the application is mentioned below. The demo video link and APK file is attached below.





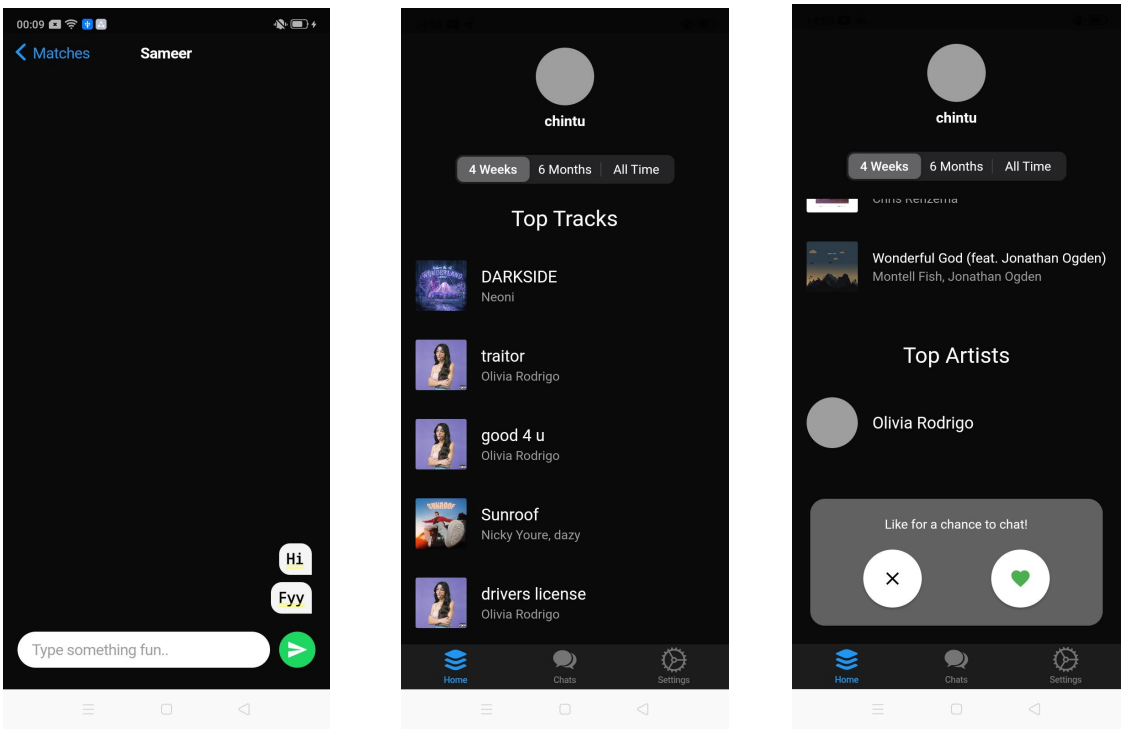


Figure 27: Band-U application snippets

Demo Video links:

https://mahindraecolecentrale-my.sharepoint.com/:v:/g/personal/krishna19226_mechyd_ac_in/ET5-4CkL5_pJqEqRYDghdscBoi4PY1cpAcdehI7xIhAVGA?e=yCNWwh

https://mahindraecolecentrale-my.sharepoint.com/:v:/g/personal/krishna19226_mechyd_ac_in/EdP4EGD_TOdAotEEiuRz8kEBs8HgUP_dOYtW34-NfwmCYw?e=hNxgbK

The demo APK file of the Band-U application can be downloaded from the link below.

https://mahindraecolecentrale-my.sharepoint.com/:u:/g/personal/krishna19226_mechyd_ac_in/EdxPHBS9bvtIi6k_06EqHi8BqPHxLnBwX4lF6kdg-VAEoQ?e=8Ih8Oy

CONCLUSION

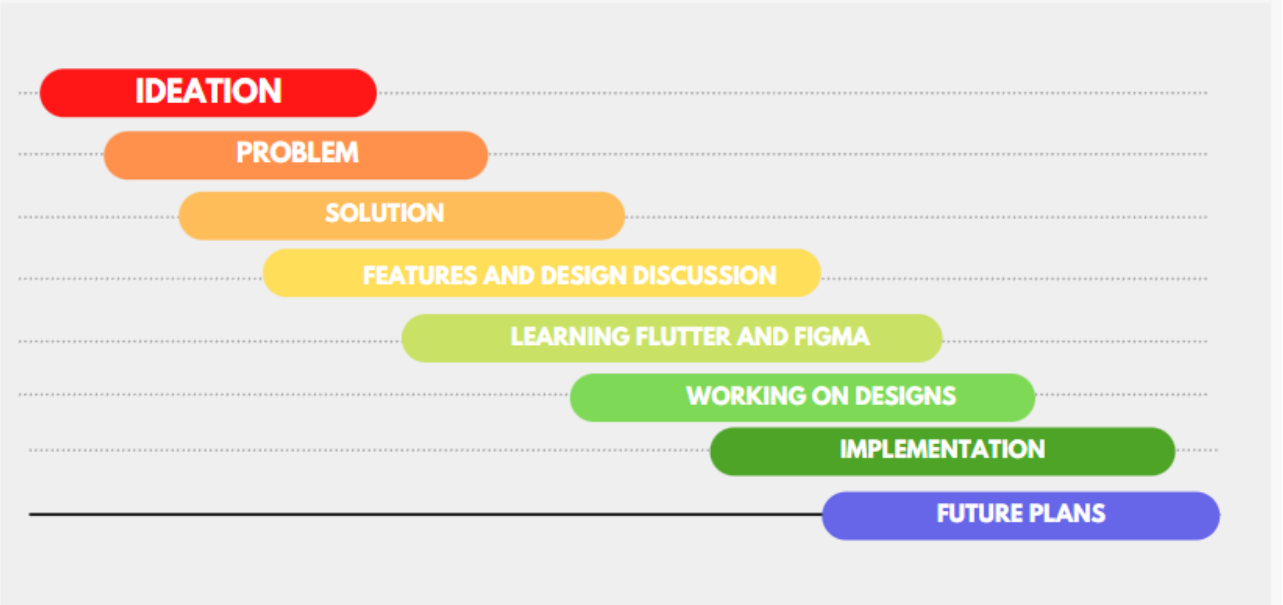
Band-U application has the potential to revolutionize the way people connect with each other through shared interests in music. By allowing users to input their favorite bands and genres, the app can match them with compatible partners who have similar musical tastes. This unique approach to dating has the potential to create stronger, more meaningful connections between individuals.

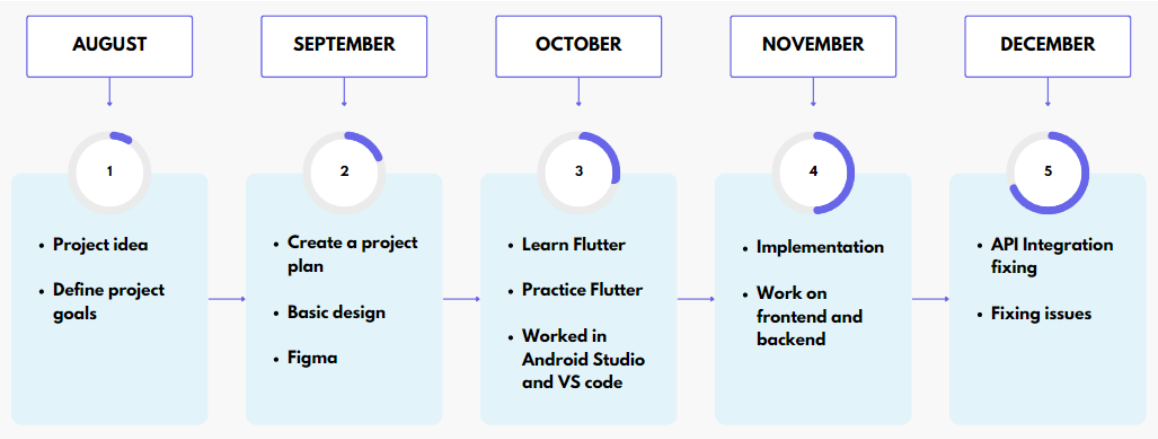
One of the key strengths of the music-based dating app is its ability to bring people together through a shared love of music. In a world where technology often serves as a barrier to personal connections, this app offers a way for people to connect with each other in a more authentic and meaningful way. By focusing on a shared interest in music, the app helps users to bypass the superficial qualities that often dominate traditional dating apps and instead focuses on more meaningful qualities that can lead to long-term relationships.

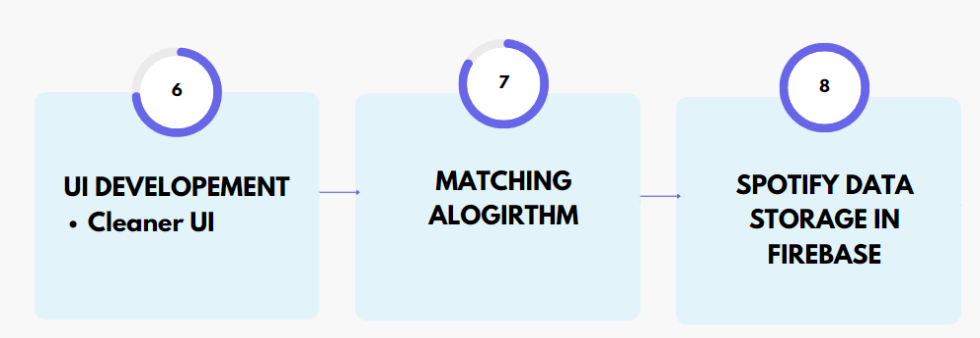
In terms of potential challenges, one of the main concerns with the music-based dating app is the potential for users to misrepresent themselves. While the app requires users to input their favorite bands and genres, it is possible that some users may not be completely truthful about their musical preferences in order to appear more compatible with potential matches. It will be important for the app to have measures in place to ensure that users are accurately representing themselves and their interests.

Overall, the Band-U app has the potential to be a successful and innovative addition to the online dating landscape. By focusing on shared interests in music, the app offers users a unique and authentic way to connect with each other and form meaningful relationships. While there may be challenges to overcome, the app has the potential to bring people together in a more meaningful and authentic way than traditional dating apps.

TIMELINE


IDEATION
PROBLEM
SOLUTION
FEATURES AND DESIGN DISCUSSION
LEARNING FLUTTER AND FIGMA
WORKING ON DESIGNS
IMPLEMENTATION
FUTURE PLANS


AUGUST
1
• Project idea
• Define project goals
SEPTEMBER
2
• Create a project plan
• Basic design
• Figma
OCTOBER
3
• Learn Flutter
• Practice Flutter
• Worked in Android Studio and VS code
NOVEMBER
4
• Implementation
• Work on frontend and backend
DECEMBER
5
• API Integration fixing
• Fixing issues


6
UI DEVELOPEMENT
• Cleaner UI
7
MATCHING ALOGIRTHM
8
SPOTIFY DATA STORAGE IN FIREBASE

REFERENCES

- https://youtu.be/1ukSR1GRtMU
- https://youtu.be/sfA3NWDBPZ4
- https://developer.spotify.com/documentation/web-api/quick-start/
- https://stackoverflow.com/
- https://github.com/