Submitted To:  
 Hammad Mujtaba   
Lecturer, Victoria University Sydney

NIT3213 Mobile Application Development

Assignment 1: Mobile App UI Design and Implementation

Manu Manvi (s8088576)

2025

Table of Contents

[1. OBJECTIVE 3](#_Toc211033902)

[2. FIGMA DESIGN PROCESS 3](#_Toc211033903)

[A. Prompts Used 3](#_Toc211033904)

[i. Login 3](#_Toc211033905)

[ii. Home 3](#_Toc211033906)

[iii. Profile 3](#_Toc211033907)

[B. AI Design Copilot Usage 4](#_Toc211033908)

[C. Figma Outputs 4](#_Toc211033909)

[3. IMPLEMENTATION IN ANDROID STUDIO 8](#_Toc211033910)

[A. Project Setup 8](#_Toc211033911)

[B. Fragments Created 8](#_Toc211033912)

[4.NAVIGATION IMPLEMENTATION 8](#_Toc211033913)

[A. Overview 8](#_Toc211033914)

[B. Step-by-Step Implementation 8](#_Toc211033915)

[5.MATERIAL DESIGN CONCEPTS USED 13](#_Toc211033916)

[6.VECTOR DRAWABLES AND RESOURCES 15](#_Toc211033917)

[7.FINAL APP SCREENS 17](#_Toc211033918)

[8.CHALLENGES AND LEARNINGS 19](#_Toc211033919)

[9.CONCLUSION 19](#_Toc211033920)

[10.APPENDIX 20](#_Toc211033921)

Table of Figures

[Figure 1: Login Screen Figma Output 4](#_Toc211033853)

[Figure 2: Home Screen Figma Output 5](#_Toc211033854)

[Figure 3: Profile Screen Figma Output 6](#_Toc211033855)

[Figure 4: Setting Screen Figma Output 7](#_Toc211033856)

[Figure 5: Navigation graph xml code 9](#_Toc211033857)

[Figure 6:Navigation graph (nav\_graph.xml) illustrating the flow between fragments: Login → Home → Settings → Home. 9](#_Toc211033858)

[Figure 7: activity\_main.xml setup showing the NavHostFragment, which hosts all three fragments and controls navigation transitions. 10](#_Toc211033859)

[Figure 8: Navigating from Login to Home, implemented using the Android navigation Component. 11](#_Toc211033860)

[Figure 9: Navigating with the Bottom Navigation Bar, implemented using the Android Navigation Component 12](#_Toc211033861)

[Figure 10: Screenshot of Material 3 text input and button components in the Login fragment, applying Luno’s dark neon theme. 14](#_Toc211033862)

[Figure 11: Screenshot of Material 3 text input and button components in the Home fragment, applying Luno’s dark neon theme and drawables pictures. 14](#_Toc211033863)

[Figure 12: Example from fragment\_profile.xml 15](#_Toc211033864)

[Figure 13: res/drawable folder from the Android Studio project view showing files like ic\_arrow\_back.xml, placeholder\_album\_art.png, gradient\_button\_background.xml, etc. 16](#_Toc211033865)

[Figure 14: Android Studio Emulator - Login Screen 17](#_Toc211033866)

[Figure 15: Android Studio Emulator - Home Screen 18](#_Toc211033867)

[Figure 16: Android Studio Emulator - Profile Screen 18](#_Toc211033868)

[Figure 17: Gradient Button Background 20](#_Toc211033869)

[Figure 18: Custom Outlined Text Field background 20](#_Toc211033870)

# OBJECTIVE

The objective of this project is to design and implement a mobile app UI using AI-generated Figma designs and Android Studio layouts. The app is named Luno, a next-generation music streaming app with a dark neon aesthetic and Gen Z–inspired interface.

# FIGMA DESIGN PROCESS

## Prompts Used

### Login

1. Design a sleek, dark-themed login screen for a music streaming app. Use a vibrant accent color scheme with fields for email and password, and an enticing 'Start Listening' button.
2. Redesign it with neon boundaries with changing colours and more creative icons with options and have more options than just basic requirements, something which is different than other apps, more gen z eye catching thing other than "welcome back" and make it of a bit dark colour just that the white is too bright.
3. Change the app name to luno and instead of " ready to vibe?" write "Beyond streaming. This is Luno." and keep everything else same, dim the lights of "lets catch this wave".

### Home

1. Develop a home screen using Spotify's intuitive design. Include a search bar for songs and artists, and showcase trending playlists or new releases using a grid of colorful album covers.
2. Customize: Top: Search bar → placeholder “Search your vibe 🎧”. Make bottom navigation bar with 3 icons: Home Profile Settings
3. Add sections like: “Trending Now” “Recently Played” “New Drops" Use album artwork placeholders (use rectangles and fill with cover images from Unsplash or gradient colors) Background stays dark (#000 or #111)
4. Add vibrant gradient accent at top (deep blue to deep green) and include real albums from artists like future, drake, playboi carti, taylor swift, sabrina carpenter and add a song track already playing/ at the pause the song should be highest in the room by travis scott.

### Profile

1. Create a user profile screen inspired by Spotify's clean, modern layout. Display the user's favorite genres and recently played tracks using album art and minimalist graphics.
2. Use artists like lil uzi vert, TV girl, billie eilish and some new trending songs.
3. Customize: Header: Profile pic (round) + “Hey Manu!” Add section “Your Vibe Stats” → show listening hours or favorite genre badges (Pop, HipPop, Rap)
4. Below that: “Recently Played” → album covers in horizontal scroll Keep layout clean, minimal, but glowing accents around icons not too shiny but cool colour slight neon
5. Use neon-style small borders (outer glow effect) Add Luno logo watermark faintly in background for branding.
6. Make the navigation smooth. and a back button to go back on home, the bar at the top right in little icons with user profile in round shape like it was before.

## AI Design Copilot Usage

* Used *AI Design Copilot – Text to UI Generator* plugin in Figma.
* Entered the three prompts individually to generate base designs for Login, Home, and Profile screens.
* Customized the design manually to match a **Gen Z, futuristic “Luno” vibe**, including:
* Tagline: *“Beyond streaming. This is Luno.”*
* Neon glow buttons and borders (purple & cyan tones).
* Gradient backgrounds and subtle shadows.
* Added Luno logo and icons for Google, Spotify, and Voice logins.
* Exported screenshots (due to AI credit limits) as login\_luno.png, home\_luno.png, and profile\_luno.png.

## Figma Outputs

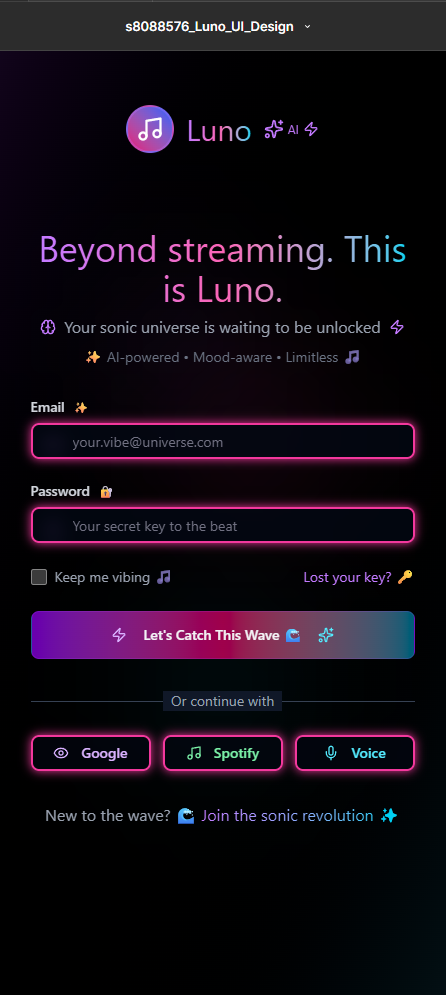


Figure 1: Login Screen Figma Output

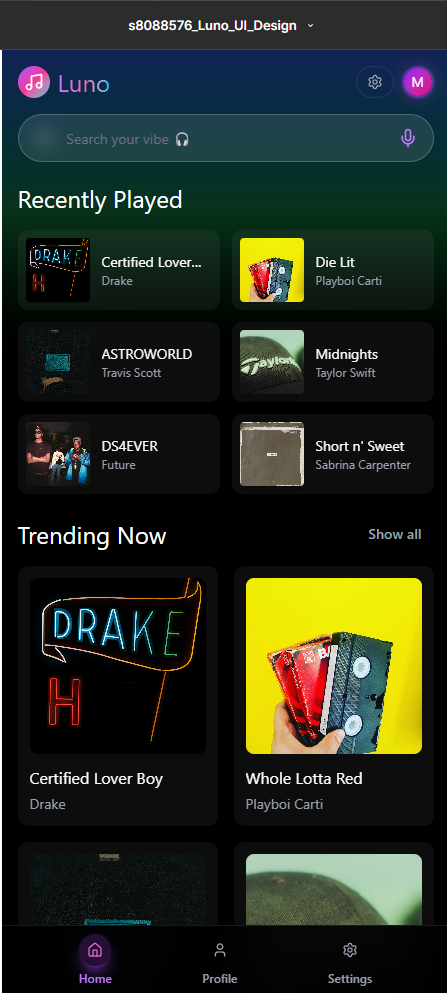


Figure 2: Home Screen Figma Output

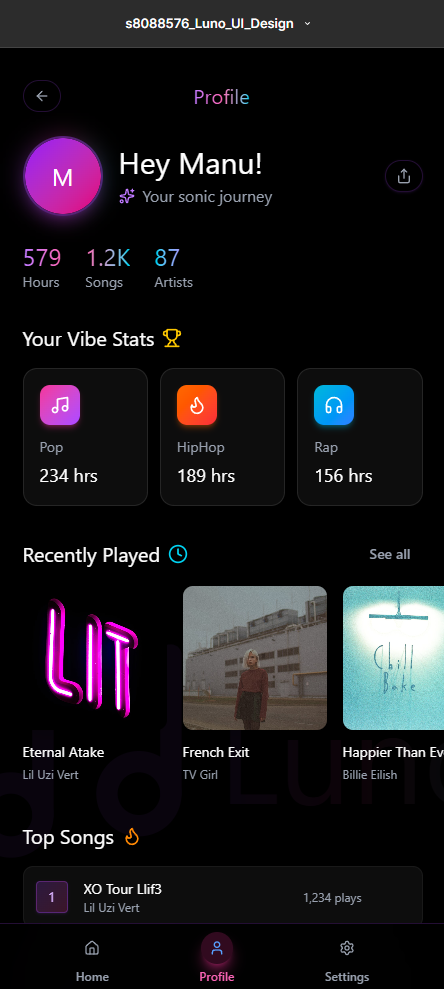


Figure 3: Profile Screen Figma Output

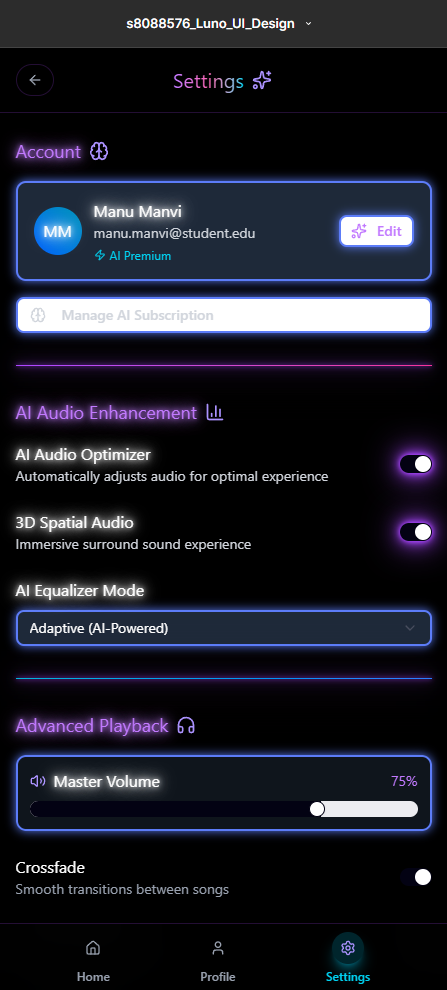


Figure 4: Setting Screen Figma Output

# 3. IMPLEMENTATION IN ANDROID STUDIO

## A. Project Setup

* Project name: s8088576\_assignment1
* Language: Kotlin
* Minimum SDK: API 21
* Used **Material 3** theme with dark background (#000000) and accent colors (#C77DFF, #00FFFF).

## B. Fragments Created

| **Fragment Name** | **Purpose** |
| --- | --- |
| LoginFragment | Login screen with email/password fields and “Let’s Catch This Wave 🌊” button |
| HomeFragment | Home screen showing main dashboard with a navigation button to Profile |
| ProfileFragment | Profile screen showing user data and a button to go back to Home |
| SettingsFragment | Settings page for customization — includes options for theme, notifications, audio quality, and privacy preferences. |

# 4.NAVIGATION IMPLEMENTATION

## A. Overview

Navigation was implemented using the **Navigation Component**, Google’s recommended method for fragment transitions.  
The navigation flow is:  
**Login → Home → Profile → Home/Settings**

## B. Step-by-Step Implementation

1. Added Navigation dependencies to build.gradle.kts

// In the dependencies { ... } block

implementation(libs.androidx.navigation.fragment.ktx)

implementation(libs.androidx.navigation.ui.ktx)

1. Created nav\_graph.xml under res/navigation/

File Location: app/src/main/res/navigation/nav\_graph.xml

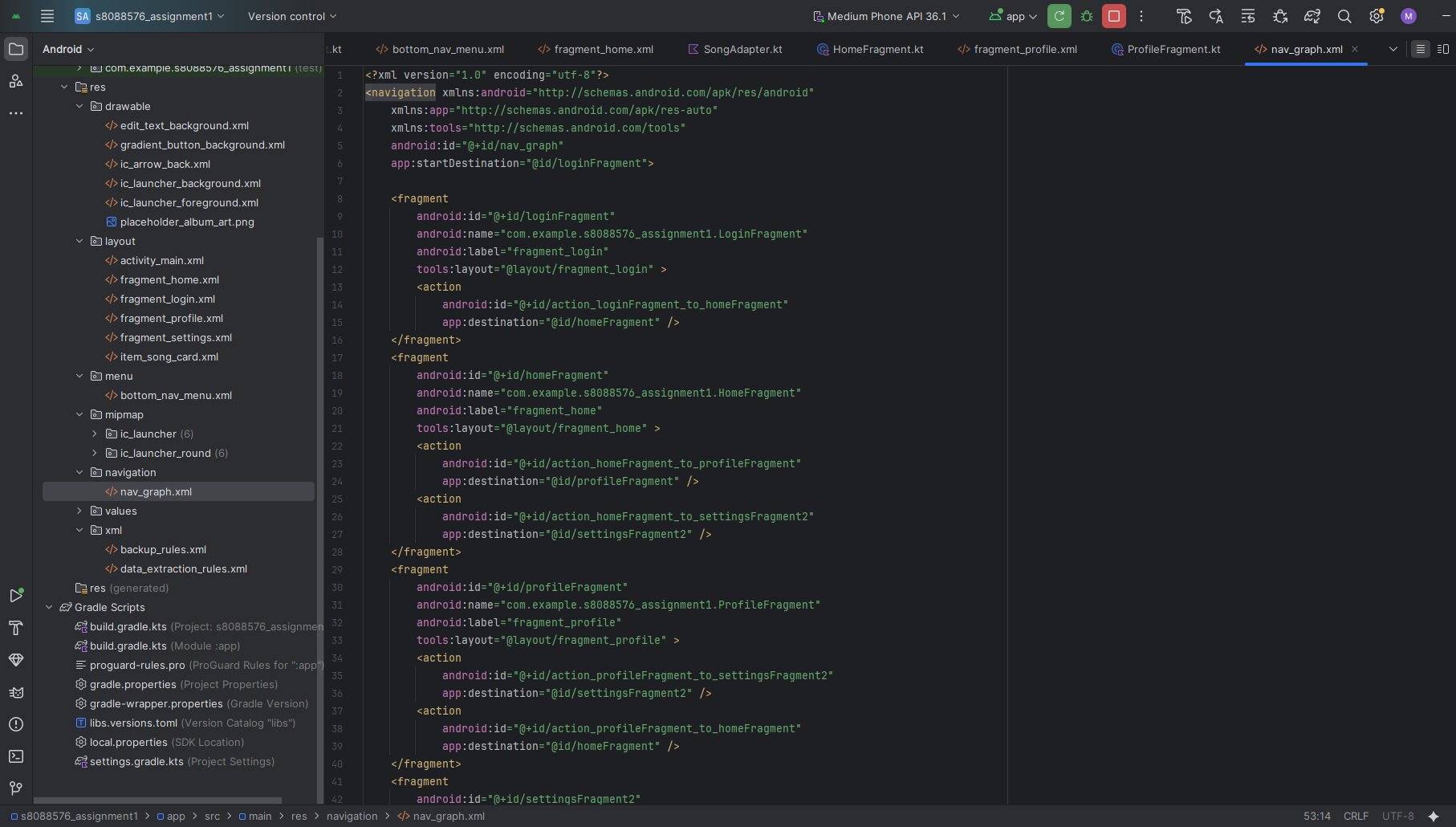


Figure 5: Navigation graph xml code

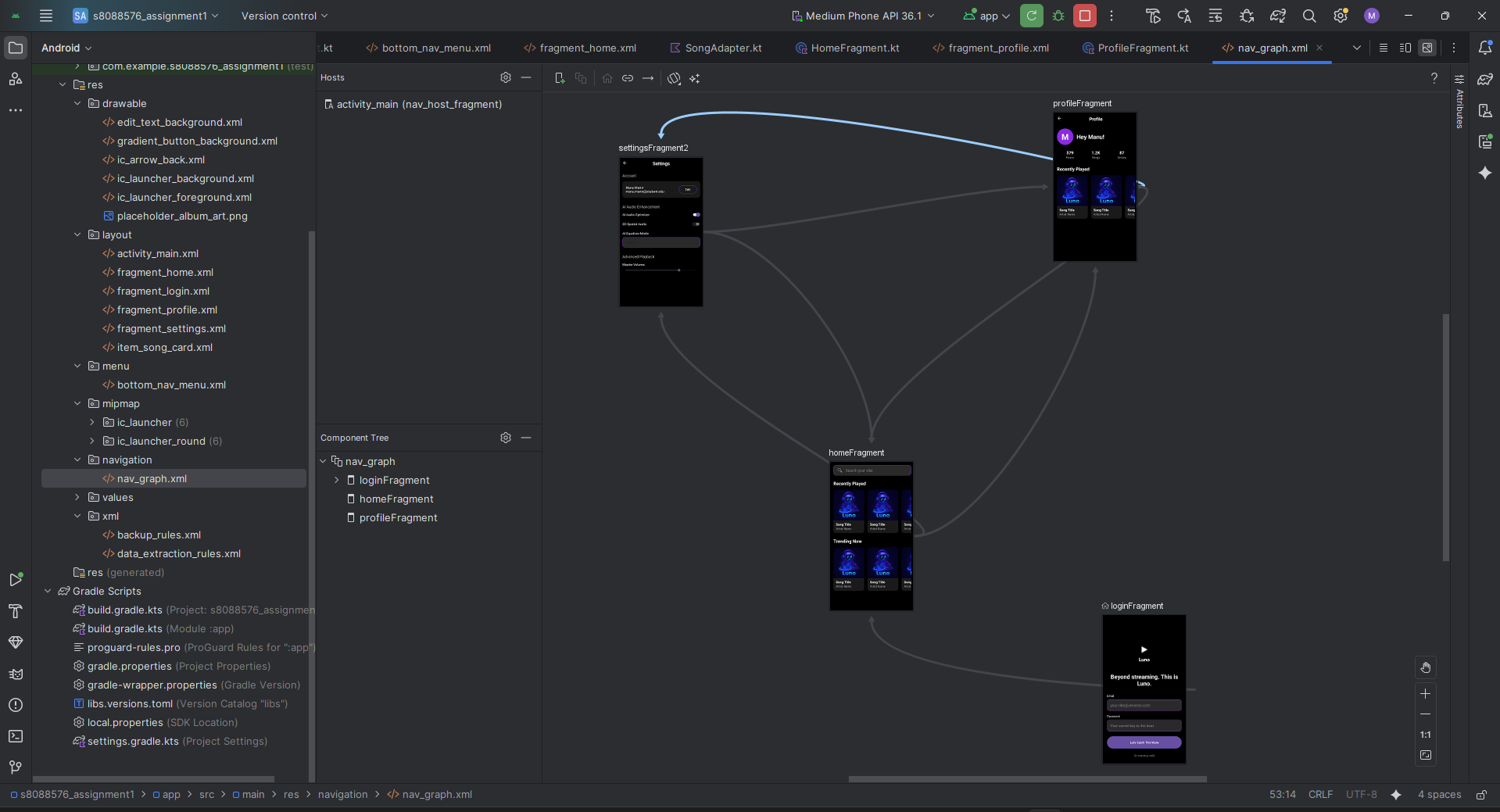


Figure 6:Navigation graph (nav\_graph.xml) illustrating the flow between fragments: Login → Home → Settings → Home.

1. Added NavHostFragment in activity\_main.xml

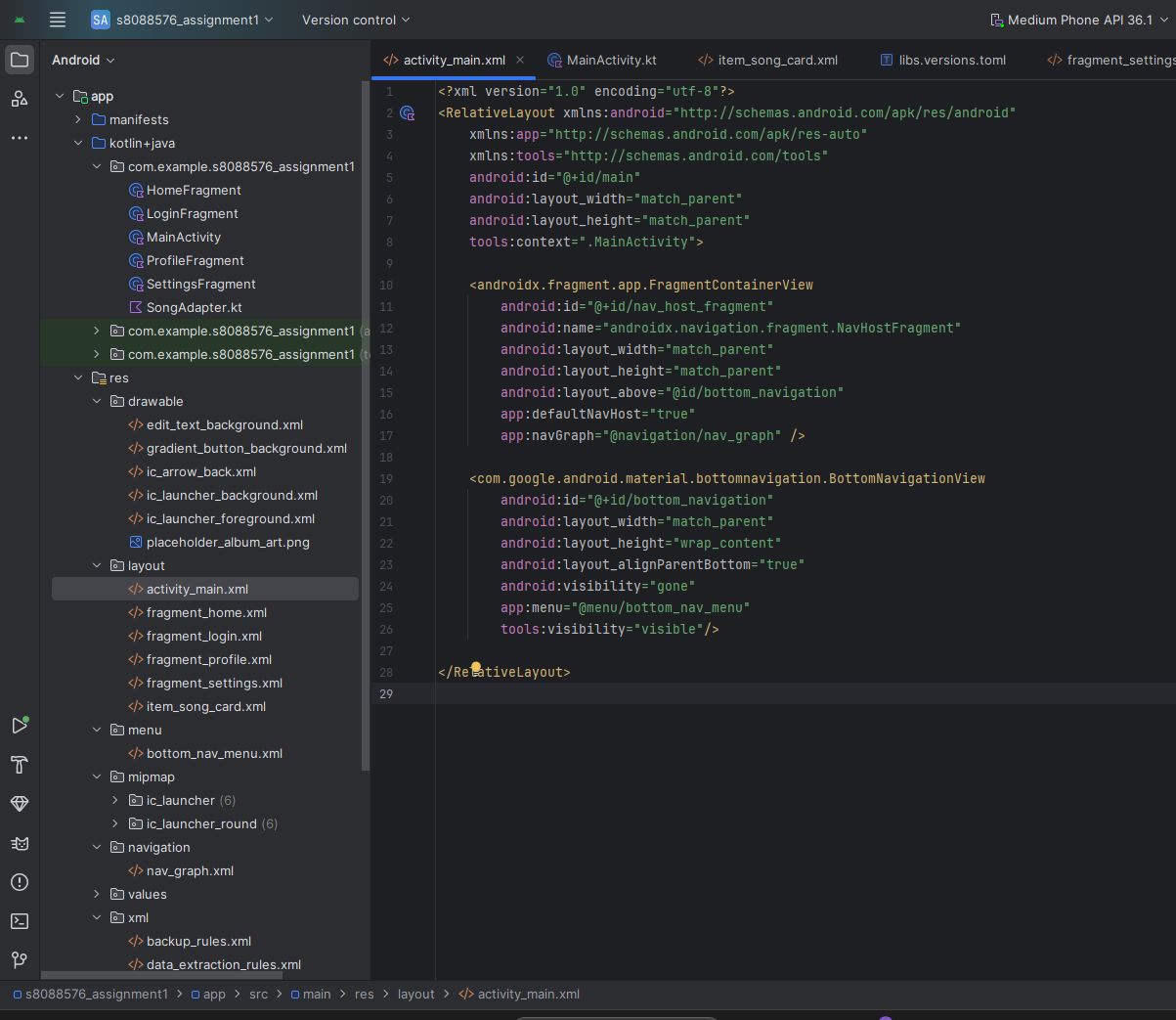


Figure 7: activity\_main.xml setup showing the NavHostFragment, which hosts all three fragments and controls navigation transitions.

1. Set up Navigation Actions in Kotlin Files

This shows how you programmatically trigger navigation from one screen to another.

**Example 1**: Navigating from Login to HomeFile

Location: app/src/main/java/com/example/s8088576\_assignment1/LoginFragment.kt

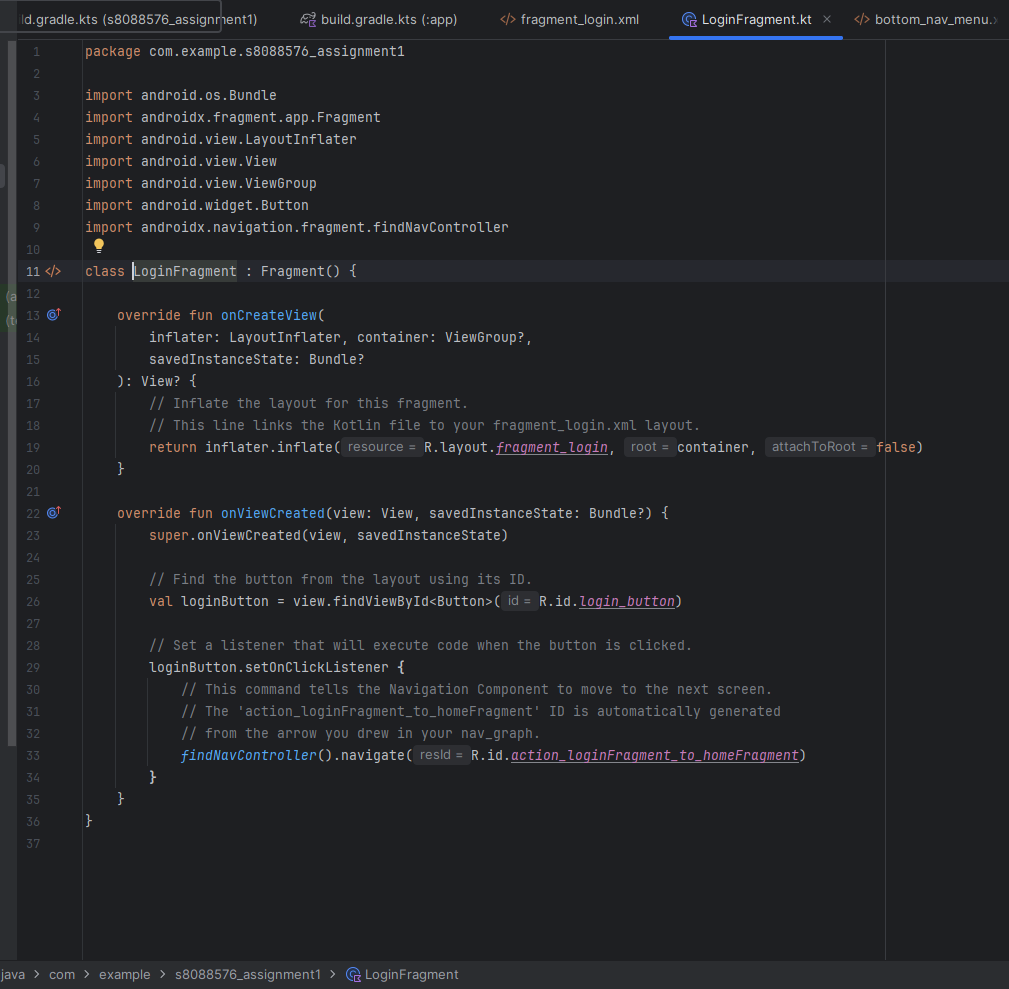


Figure 8: Navigating from Login to Home, implemented using the Android navigation Component.

**Example 2**: Navigating with the Bottom Navigation Bar

This shows the code that powers the bottom bar. It's the most important piece for your main navigation.

File Location: app/src/main/java/com/example/s8088576\_assignment1/MainActivity.kt

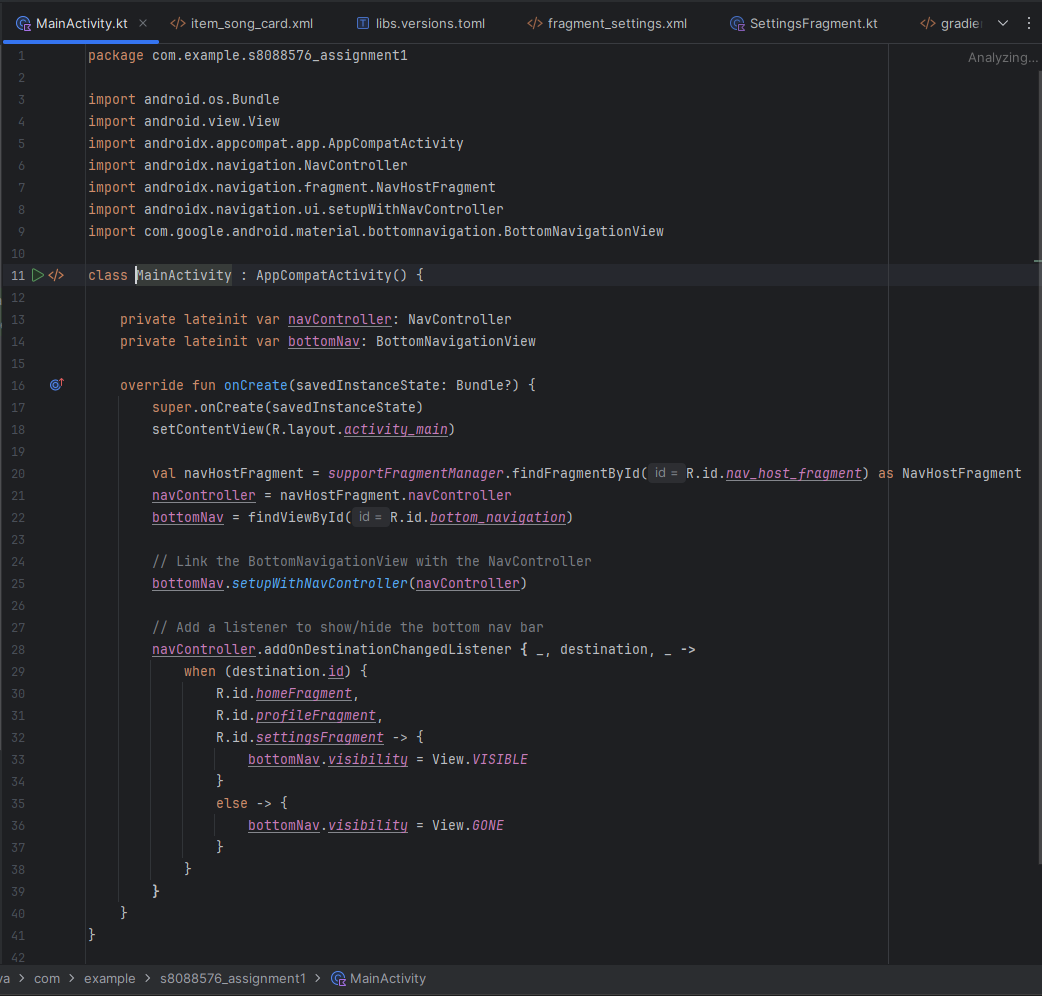


Figure 9: Navigating with the Bottom Navigation Bar, implemented using the Android Navigation Component

1. Verified transitions between fragments using Emulator.

# 5.MATERIAL DESIGN CONCEPTS USED

Our "Luno" app was developed using modern Material 3 design principles to ensure a consistent, intuitive, and visually appealing user experience.

•Component-Based UI:

The app is built with standard Material 3 components, including MaterialButton for calls-to-action, com.google.android.material.switchmaterial.SwitchMaterial for toggles, and BottomNavigationView for main screen navigation. This ensures predictable behavior and a consistent look.

•Dark Theme by Default:

Following modern UX trends and the Figma design's aesthetic, the app exclusively uses a dark theme (android:background="#000000"). Text colors were chosen with high contrast (#FFFFFF for primary text, #B3B3B3 for secondary) to ensure excellent readability against the dark background.

•Shape and Corner Radius:

A key part of the design's soft, modern feel comes from the consistent use of rounded corners. Components like CardView and EditText use a 12dp corner radius, while buttons use a fully rounded 100dp radius to create a pill shape, applying a cohesive shape system across the app.

•Vibrant Accent Palette:

The app's identity is defined by its vibrant accent colors, primarily a neon-style purple (#8A2BE2) and magenta (#FF00FF). These are used for button borders, gradients, and user avatars to create a unique and energetic brand identity that appeals to a Gen Z audience.

•Typography Hierarchy:

A clear visual hierarchy is established using typography. Screen titles use a bold, 22sp font, section headers use 18sp, and body content uses 16sp or 14sp. This guides the user's eye and makes the interface easy to scan.

•Custom Styling for Input Fields:

The EditText fields use a custom drawable (edit\_text\_background.xml) that combines a dark fill color with a vibrant purple outline stroke, following Material Design's guidelines for outlined text fields while adapting them to the app's unique style.

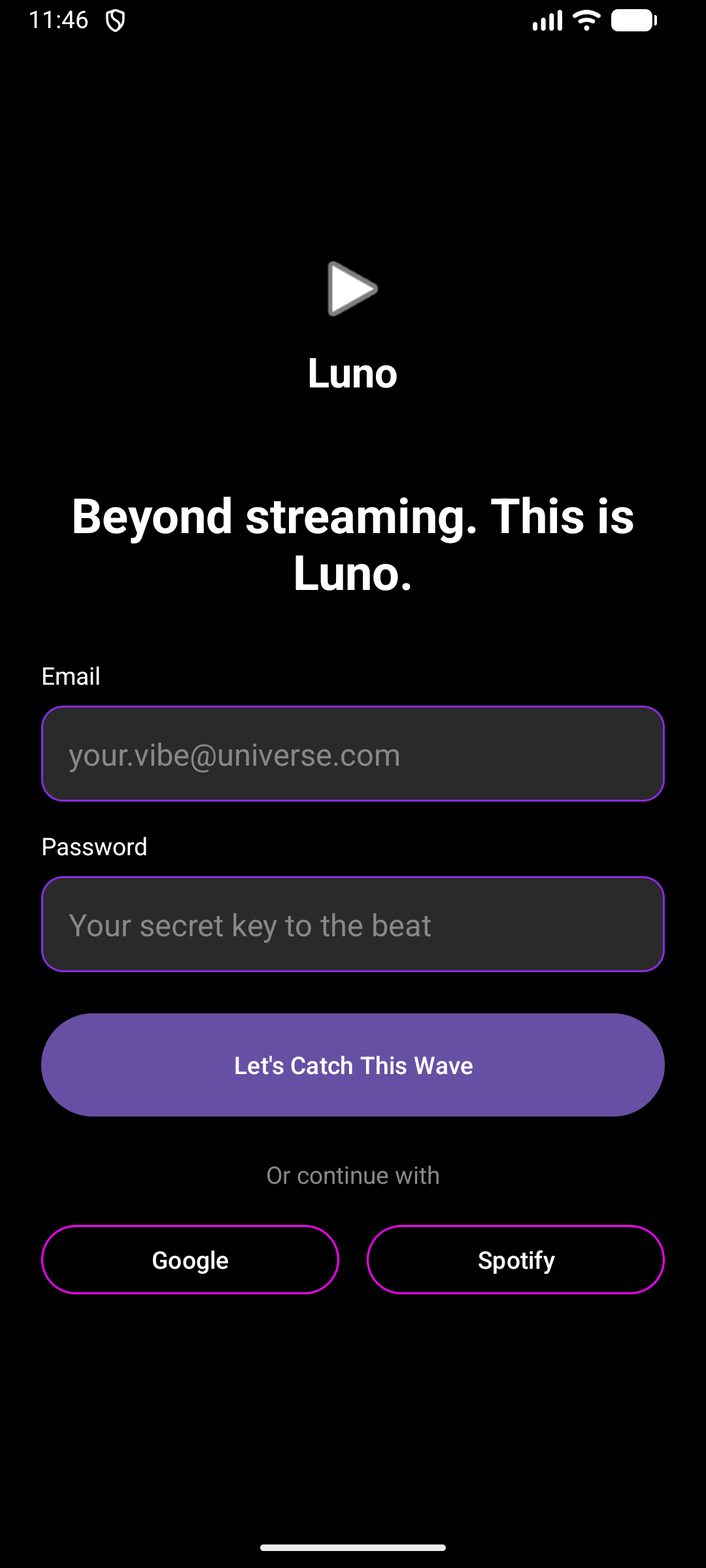


Figure 10: Screenshot of Material 3 text input and button components in the Login fragment, applying Luno’s dark neon theme.

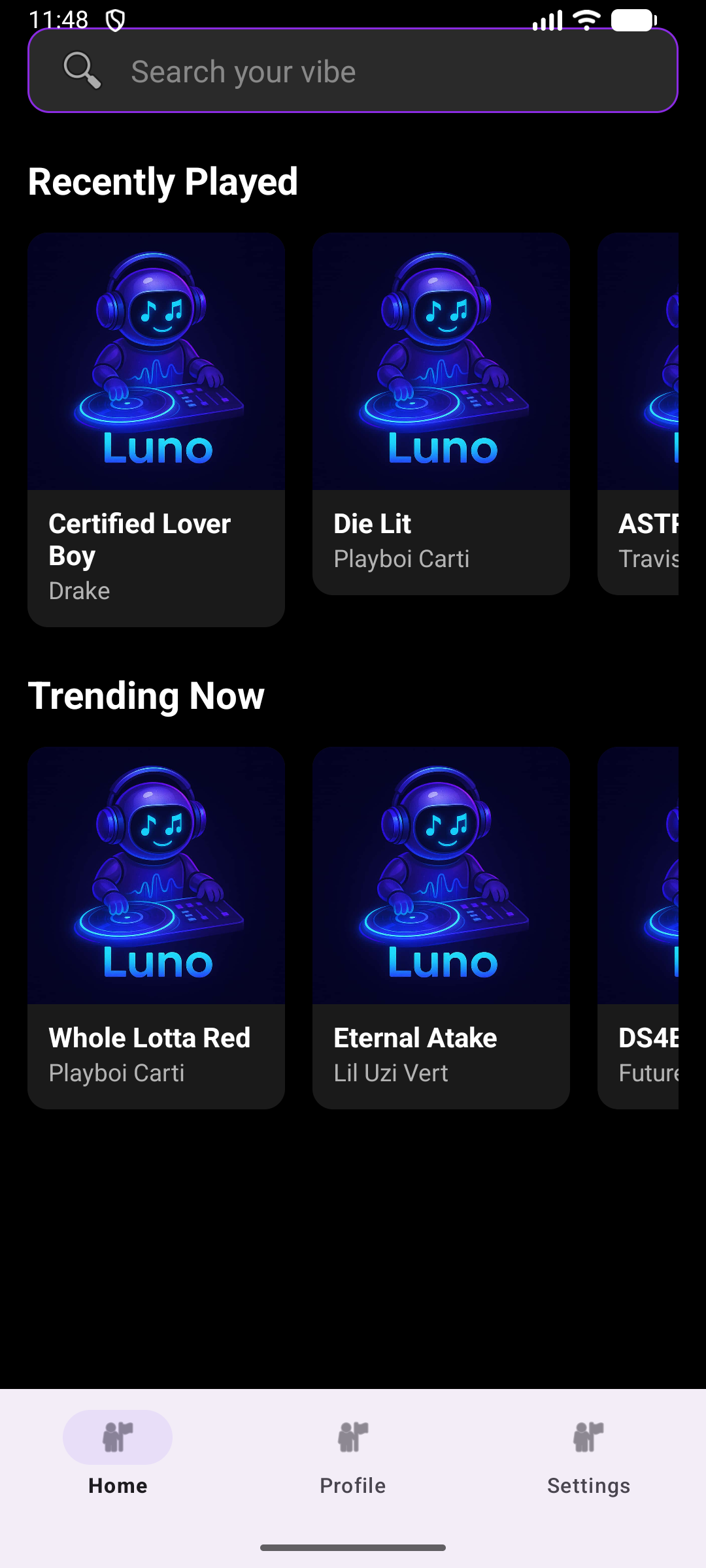


Figure 11: Screenshot of Material 3 text input and button components in the Home fragment, applying Luno’s dark neon theme and drawables pictures.

# 6.VECTOR DRAWABLES AND RESOURCES

To maintain a sharp and scalable UI, all icons were imported as vector drawables. This ensures they look crisp on any screen density. The process was as follows:

1.Icons from the Figma design, such as the back arrow, were identified.

2.Standard Material icons were imported directly into Android Studio using the Vector Asset tool. For example, arrow\_back was imported and saved as ic\_arrow\_back.xml.

3.These vector drawables were then used in XML layouts for elements like ImageViews and Buttons with a single line of code, ensuring high performance and scalability.

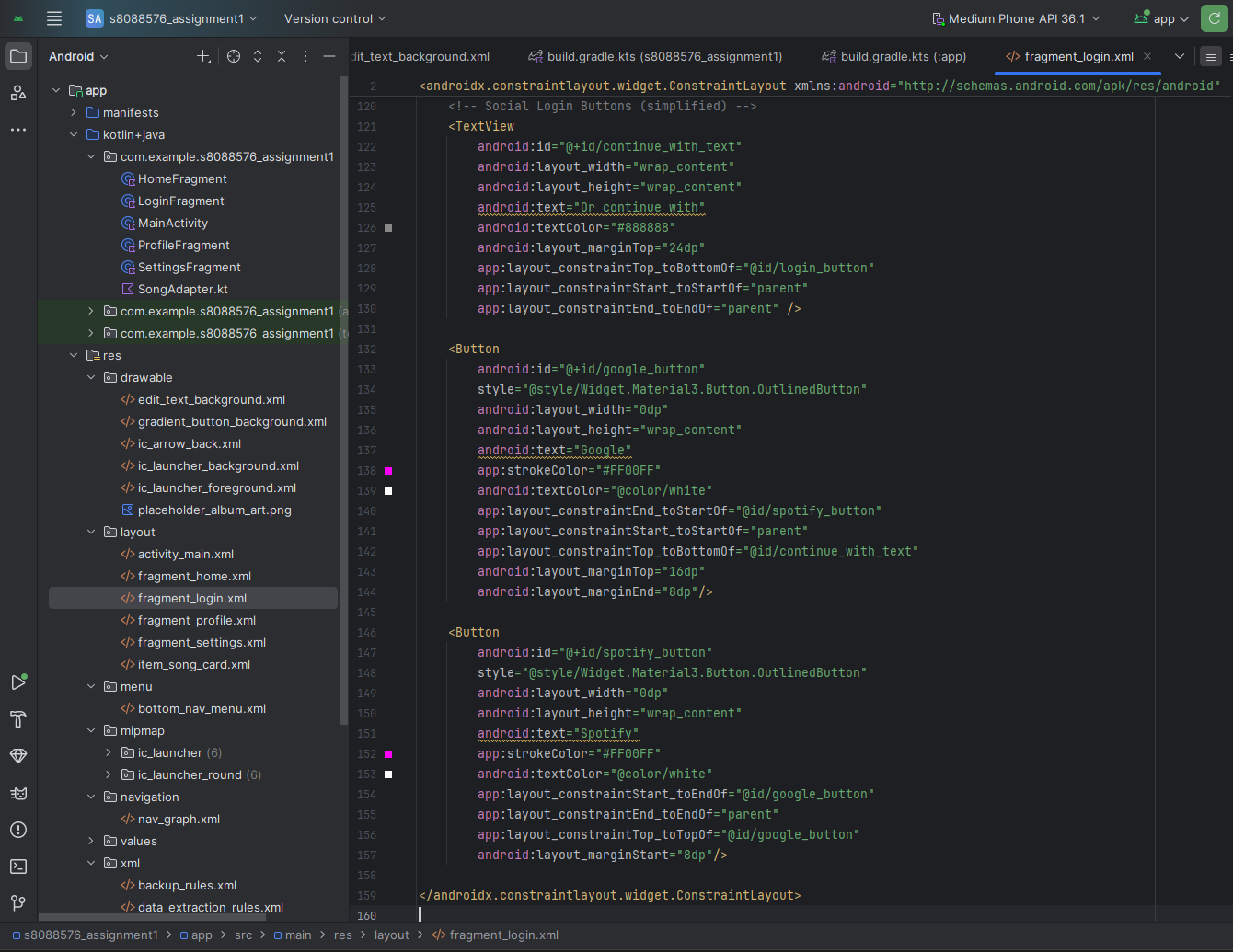


Figure 12: Example from fragment\_profile.xml

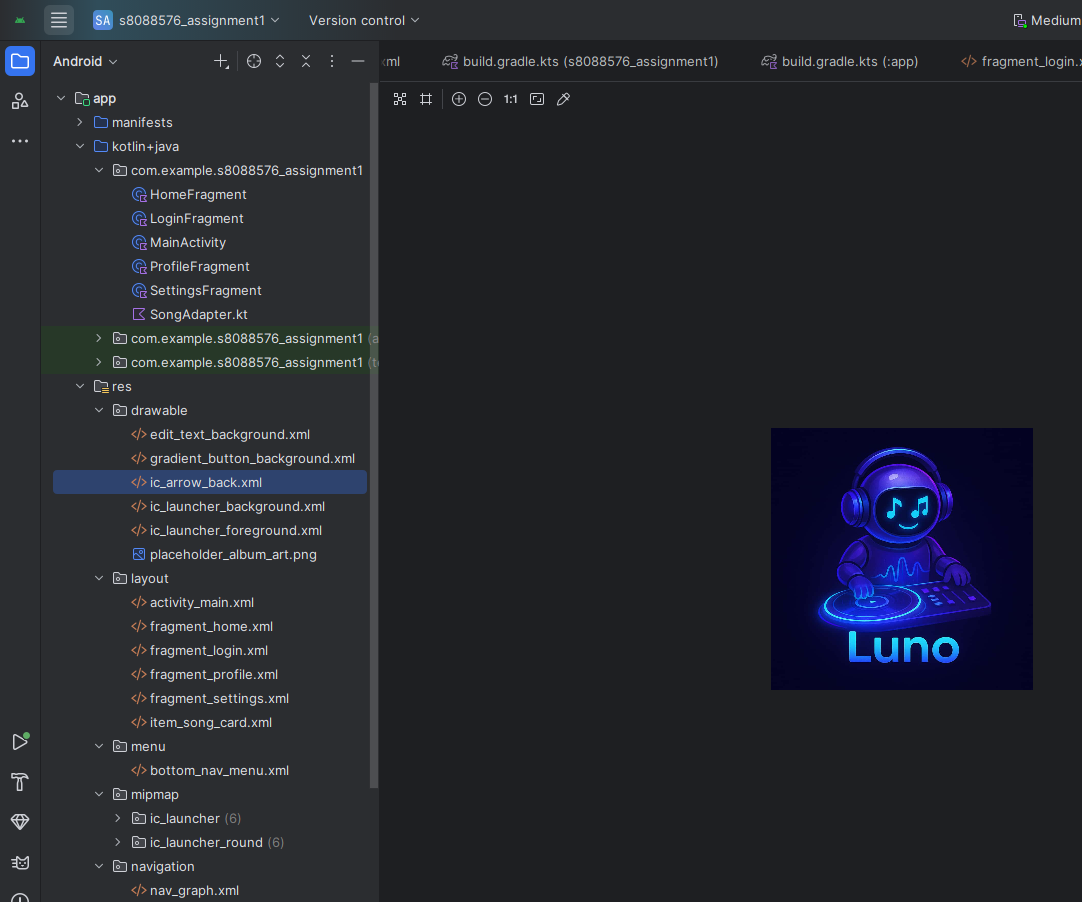


Figure 13: res/drawable folder from the Android Studio project view showing files like ic\_arrow\_back.xml, placeholder\_album\_art.png, gradient\_button\_background.xml, etc.

# 7.FINAL APP SCREENS

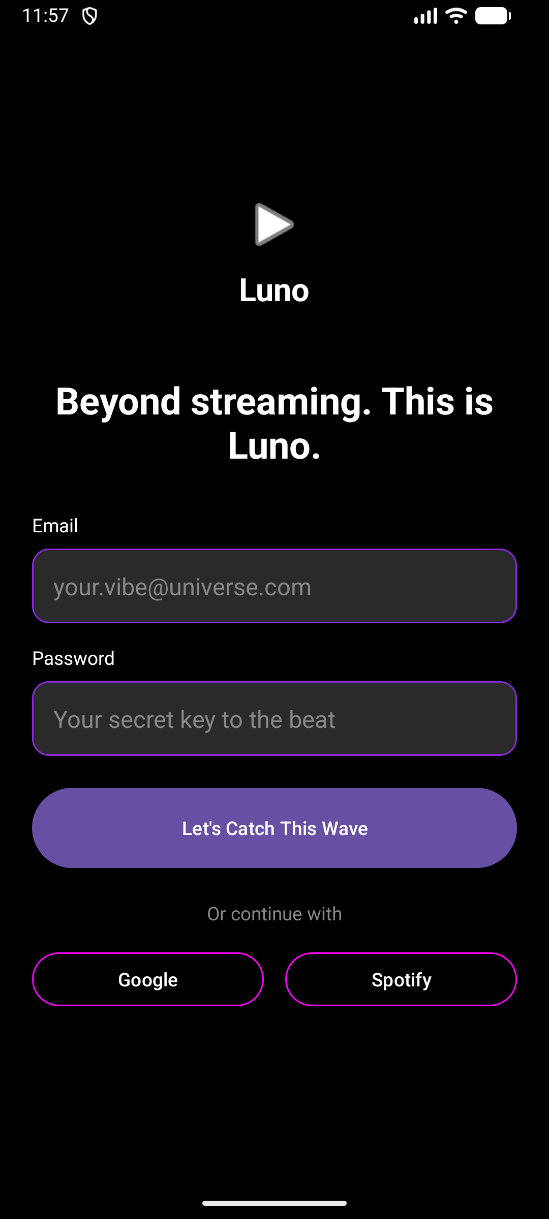


Figure 14: Android Studio Emulator - Login Screen

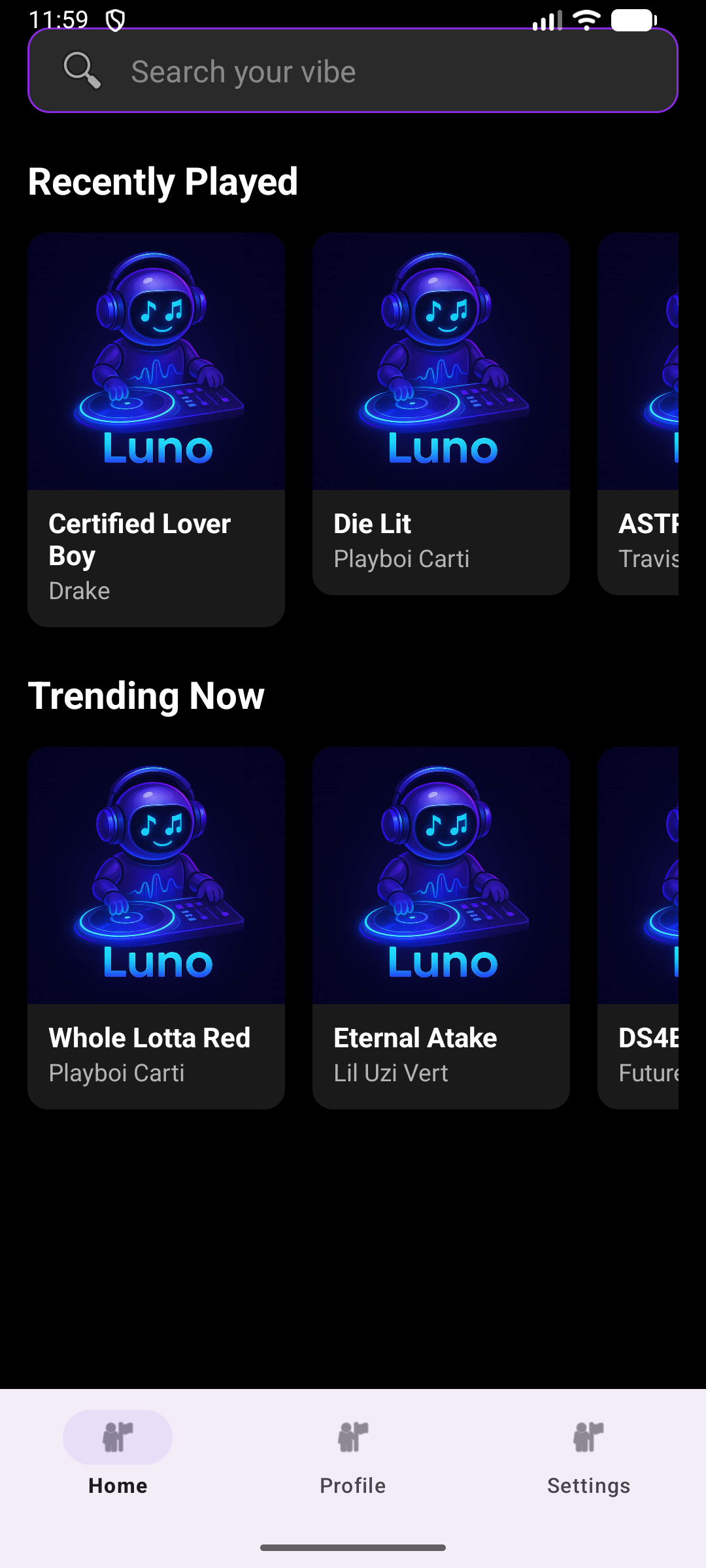


Figure 15: Android Studio Emulator - Home Screen

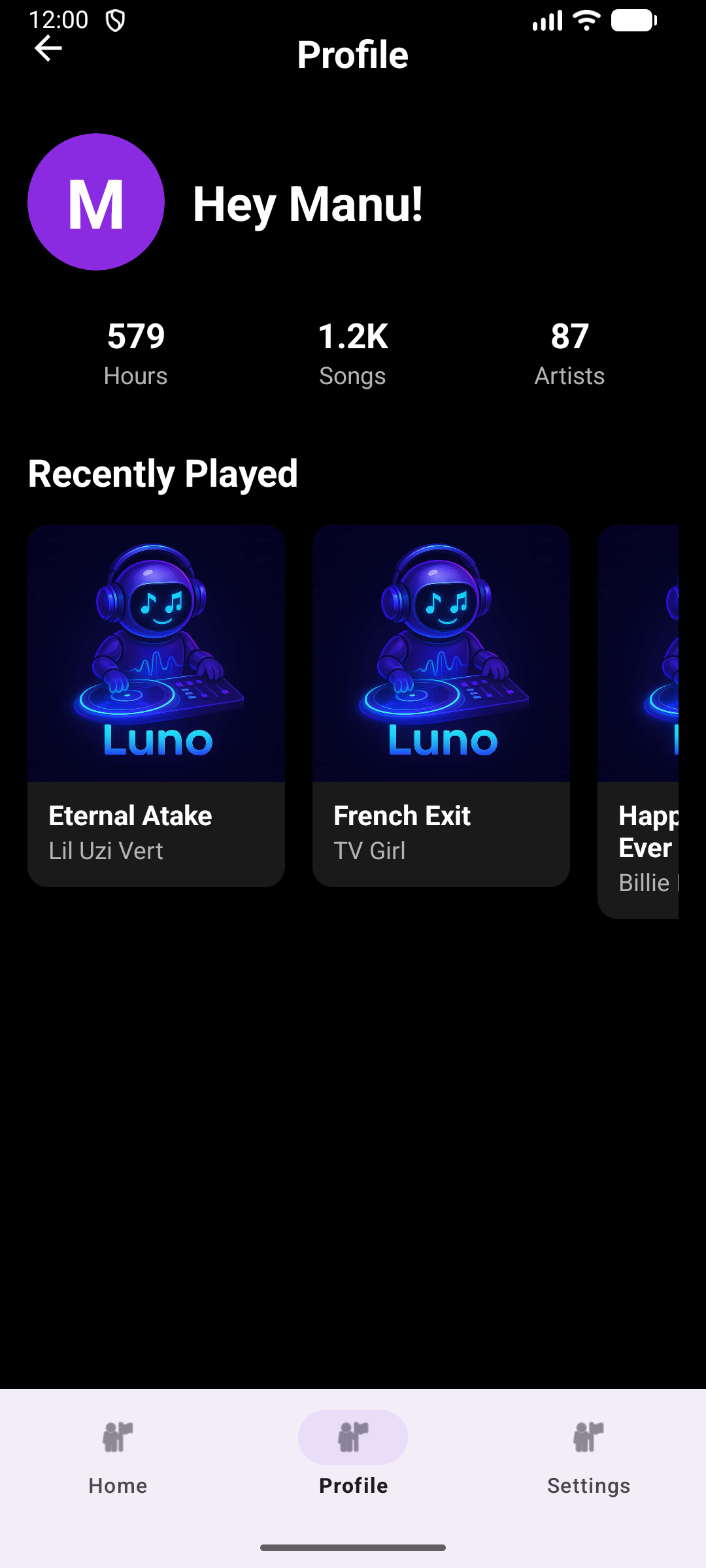


Figure 16: Android Studio Emulator - Profile Screen

# 8.CHALLENGES AND LEARNINGS

•Translating Design to Code: A major challenge was understanding that a Figma design cannot be directly copied. I learned the manual process of translating visual properties from Figma into Android XML, such as recreating gradients and borders using shape drawables.

•Mastering the Navigation Component: Initially, I struggled with linking fragments. I learned how to create a nav\_graph.xml, define navigation actions, and use findNavController().navigate() to programmatically control the user flow. Debugging the non-working Settings tab taught me the importance of matching menu item IDs with fragment IDs.

•Implementing RecyclerView for Dynamic Lists: Building the Home screen was challenging. I learned the RecyclerView, Adapter, ViewHolder, and data class pattern to efficiently display lists of data. I also learned how to update the adapter to show unique images for each list item instead of just a single placeholder.

•Understanding Gradle Dependencies: I encountered Unresolved reference errors with Gradle early on. I learned the difference between the root-level and module-level build.gradle.kts files and understood that dependencies like implementation belong only in the module-level file.

•Adopting Material 3 Principles: I practiced implementing Material 3 styling by creating custom backgrounds for buttons and text fields, establishing a color palette in colors.xml (conceptually), and building a consistent UI hierarchy through typography and spacing.

# 9.CONCLUSION

This project successfully demonstrates the complete workflow of designing and implementing a modern, music-themed Android application. By leveraging AI-assisted design with Figma and translating it into a robust native app using Android Studio, the core objectives were met. The final "Luno" application stands out through its distinct dark neon aesthetic, intuitive navigation powered by the Android Navigation Component, and a user-friendly interface built on Material 3 principles. The development process fulfilled both the functional requirements of a multi-screen app and the creative goals of realizing a unique brand identity, showcasing a solid understanding of key Android development practices.

# 10.APPENDIX

Code Snippet 1: Gradient Button Background

File Location: app/src/main/res/drawable/gradient\_button\_background.xml

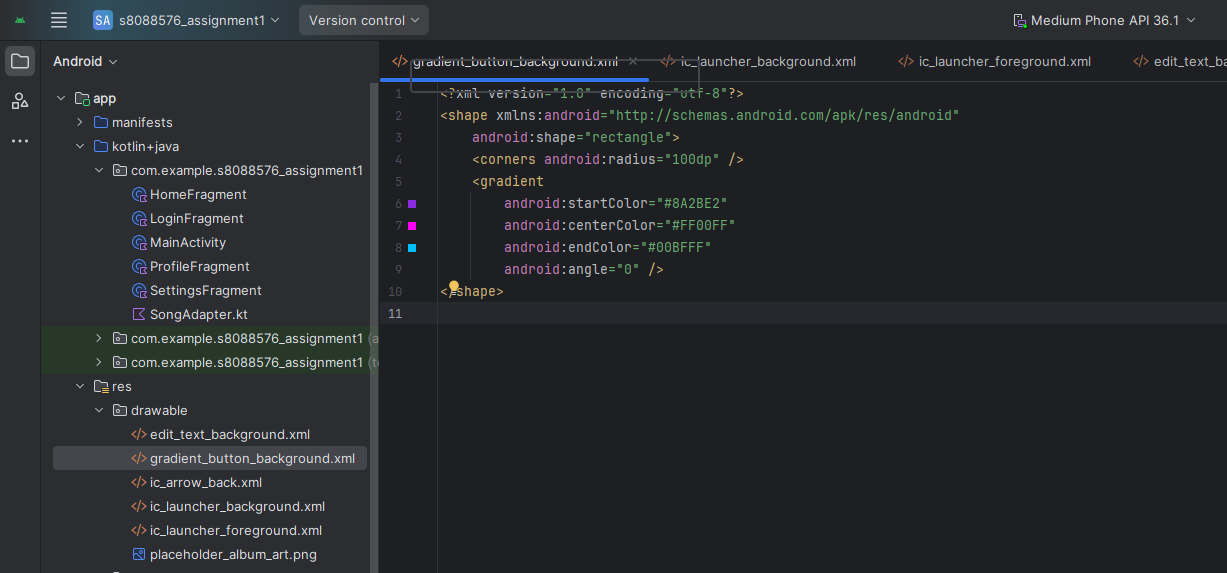


Figure 17: Gradient Button Background

Code Snippet 2: Custom Outlined Text Field Background

File Location: app/src/main/res/drawable/edit\_text\_background.xml

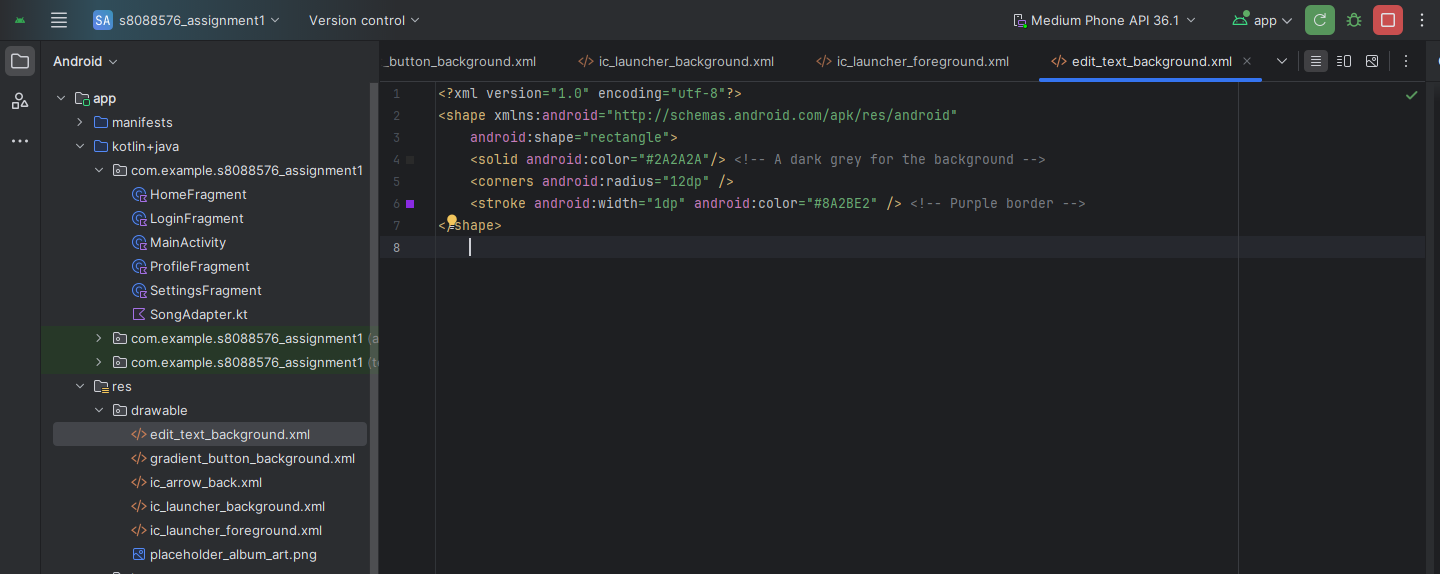


Figure 18: Custom Outlined Text Field background