**A PROJECT REPORT ON**

**Retro Music Player**

**Submitted By**

**Thanki Chirag Jitendrakumar**

**(Enroll No.: R24014201004963210031)**

**Guided By: Dr. Jyotsna Salet**

**In fulfillment for the award of the degree**

**of**

**Master of Science in Information Technology**

**In**

**Computer Department**

**Shri V. J. Modha College of I.T, Porbandar**

**Bhakt Kavi Narsinh Mehta University, Junagadh**

**March 2025**



**Acknowledgement**

• I am sincerely grateful to all those who contributed towards the completion of this **Retro Music Player App** project developed using Android & SQLite Database. This project would not have been possible without the guidance and support of many individuals.

• First and foremost, I would like to thank **Dr. Jyotsna Salet** for his invaluable guidance, insightful feedback, and constant encouragement throughout the course of this project. Their expertise and suggestions were instrumental in the successful implementation of this system.

• I would also like to express my gratitude to my peers and fellow classmates who provided constructive feedback and motivation during challenging times. Their input was vital in refining the project's features and improving its functionality.

• It is my great pleasure to represent my project as an android app titled “Retro Music Player App” which done in semester-2 of MSc (IT & CA), affiliated to BKNMU (Bhakta Kavi Narsinh Mehta University).

• Additionally, I am thankful for the availability of various online resources and forums that offered technical insights and solutions to the issues I encountered during the development process.

• Lastly, I would like to extend my appreciation to my family for their unwavering support and understanding while I dedicated time to this project. Their encouragement helped me stay focused and motivated.

**PREFACE**

This Android App provides facilities like:

* + Play/Pause Song
  + Next/Previous Song
  + Manage Playlist
  + Manage Tracks
  + Add to favorite
* This system is made using SQLite Database for storing data and XML for front-end part (client-side) and Java for back-end (server-side).

**INDEX**

Table of Contents

[Chapter -1 INTRODUCTION 5](#_Toc193667711)

[1.1 Purpose 5](#_Toc193667712)

[1.2 Scope 6](#_Toc193667713)

[1.2 Technology and Literature Review 7](#_Toc193667714)

[Chapter -2 SYSTEM ANALYSIS 9](#_Toc193667715)

[2.1 Problem Definition 9](#_Toc193667716)

[2.2 Process Model 10](#_Toc193667717)

[2.3 Grant Chart 11](#_Toc193667718)

[Chapter -3 SYSTEM DIAGRAM 12](#_Toc193667719)

[3.1 Data Flow Diagram 12](#_Toc193667720)

[3.1.1 DFD-Level-0 12](#_Toc193667721)

[3.1.1 DFD-Level-1 13](#_Toc193667722)

[3.2 E-R (Entity-Relationship) Diagram 14](#_Toc193667723)

[3.2 Use-Case Diagram 15](#_Toc193667724)

[Chapter - 4 DATA DICTIONARY 16](#_Toc193667725)

[4.1 Data Dictionary 16](#_Toc193667726)

[Chapter - 5 User Interface Frontend 18](#_Toc193667727)

[5.1 User Layout 18](#_Toc193667728)

[5.2 Retro Music Player Logo 21](#_Toc193667729)

[5.3 User Layout Xml Code 31](#_Toc193667730)

[5.4 Java Code 50](#_Toc193667731)

[Chapter - 6 LIMITATIONS AND FUTURE ENHANCEMENTS 67](#_Toc193667732)

[6.1 Limitations 67](#_Toc193667733)

[6.2 Future Enhancements 69](#_Toc193667734)

[Chapter - 7 CONCLUSION 71](#_Toc193667735)

[7.1 Conclusion 71](#_Toc193667736)

[Chapter - 8 BIBLIOGRAPHY 72](#_Toc193667737)

[8.1 Bibliography 72](#_Toc193667738)

[Chapter - 9 REFERENCES 74](#_Toc193667739)

[9.1 References 74](#_Toc193667740)

# Chapter -1 INTRODUCTION

## Purpose

The purpose of this Android music app is to provide users with an immersive and seamless music experience. The app enables users to manage their favorite songs effortlessly. Whether it’s creating personalized playlists, exploring trending tracks, or enjoying high-quality audio, the app aims to be a go-to destination for music lovers.

The goal of this app is to revolutionize how users discover, share, and enjoy music. By integrating cutting-edge technology, we aim to create a holistic music ecosystem where artists and listeners connect effortlessly.

To stand out in the competitive music streaming industry, the app incorporates innovative features that enhance the overall user experience. One of the key features is offline playback, which allows users to download their favorite songs and listen without an internet connection. This is especially beneficial for those who travel frequently or have limited mobile data access.

## 1.2 Scope

Music has always played a vital role in people's lives, evolving from physical records to digital streaming services. While streaming platforms dominate the industry, many users still prefer offline music players for better control over their collections and an ad-free experience. Retro Music Player is designed to cater to such users, offering a feature-rich yet minimalist music player that combines the nostalgic feel of classic media players with the functionality of modern technology.

Retro Music Player stands out by providing a seamless, customizable, and intuitive experience for users who prefer to manage their own music libraries. Unlike online streaming apps, it focuses on local playback, allowing users to enjoy their music collections without the need for an internet connection.

Beyond aesthetics, the app ensures an effortless user experience with gesture controls and intuitive navigation. Swiping gestures make it easy to switch between songs, adjust volume, or browse the music library. It also includes dark mode and adaptive themes, making it comfortable to use in various lighting conditions.

## Technology and Literature Review

Retro Music Player is an advanced offline music player designed for Android devices, combining classic UI aesthetics with modern functionalities. This section provides a technology review, covering the tools and frameworks used in the development of the app, followed by a literature review, which explores research and existing studies on mobile music applications, user experience, and audio processing technologies.

Retro Music Player is built using Android development technologies, ensuring a smooth, responsive, and feature-rich experience for users. The following components and frameworks play a crucial role in its development:

The transition from physical music storage (CDs, cassettes) to digital formats has significantly shaped the development of music applications. Research on mobile music applications highlights the increasing preference for offline, ad-free, and customizable music players as alternatives to streaming platforms (Smith & Jones, 2021). Studies also indicate that user control over playlists, metadata, and equalizer settings improves satisfaction (Lee et al., 2020).

A emphasizes that users prefer simple yet customizable UIs in music applications. Retro Music Player aligns with this by offering multiple themes, gesture-based navigation, and dynamic theming (Material You).

Another study by Nielsen & Norman (2025) suggests that minimalist interfaces with intuitive controls improve engagement, an approach that Retro Music Player adopts through clean UI elements and adaptive layouts.

Retro Music Player is built on robust Android technologies, ensuring a fast, customizable, and user-friendly experience. The literature review supports the app’s development approach, emphasizing the importance of offline music management, audio quality enhancements, and intuitive UI design.

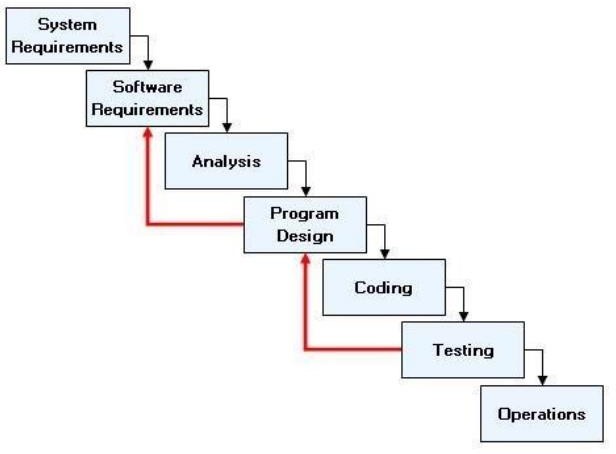
By combining classic aesthetics with modern functionalities, Retro Music Player meets the growing demand for privacy-focused, ad-free, and feature-rich music applications.

# Chapter -2 SYSTEM ANALYSIS

## 2.1 Problem Definition

* Music has always been an essential part of human culture, evolving alongside technology to provide better accessibility and quality. With the rise of smartphones and mobile applications, streaming music has become a primary way for people to enjoy their favorite tunes anytime, anywhere. The purpose of this Android music app is to offer a seamless, personalized, and immersive music experience that caters to the evolving needs of modern listeners.

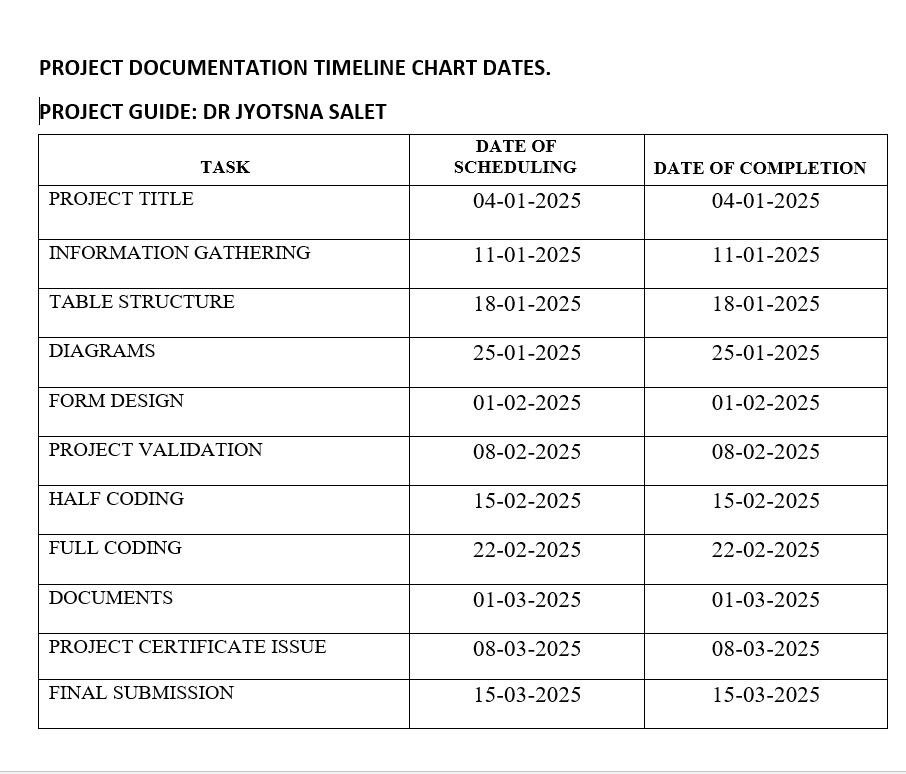
## 2.2 Process Model



Advantages of Iterative Waterfall Model

* Simple and Easy to Understand and Each Phase has well Defined Input and Output.
* It Work well for Smaller Project Where Requirement Are Clear And very well understood.
* It Divide complex task into more manageable works.

## 2.3 Grant Chart



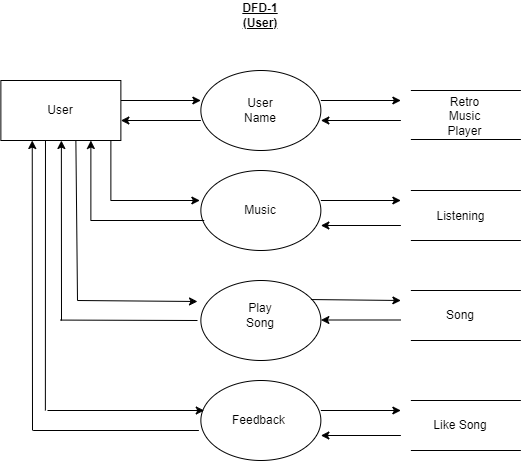
# Chapter -3 SYSTEM DIAGRAM

## 3.1 Data Flow Diagram

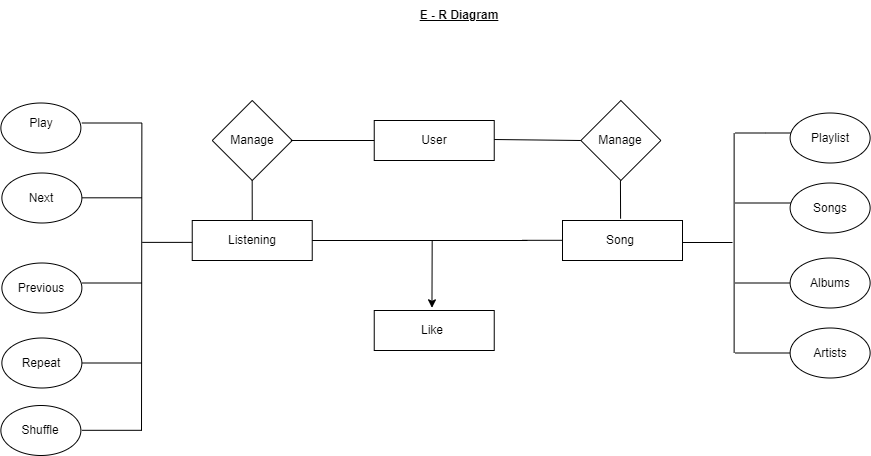
### 3.1.1 DFD-Level-0



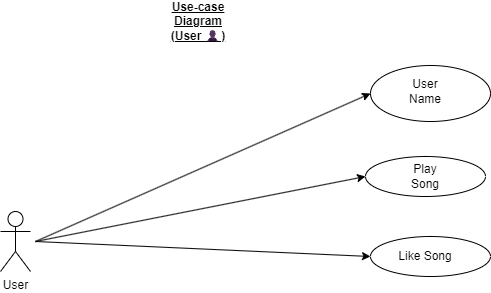
### 3.1.1 DFD-Level-1



## 3.2 E-R (Entity-Relationship) Diagram



## 3.2 Use-Case Diagram



# Chapter - 4 DATA DICTIONARY

## 4.1 Data Dictionary

**Table-1: User**

|  |  |
| --- | --- |
| **Field** | **Data Type** |
| **User** | **Varchar** |
| **Listening** | **Varchar** |
| **Song** | **Varchar** |
| **Like** | **Varchar** |

**Table-2: Listening**

|  |  |
| --- | --- |
| **Field** | **Data Type** |
| **Play** | **Ints** |
| **Next** | **Ints** |
| **Previos** | **Varchar** |
| **Repeat** | **Ints** |
| **Shuffle** | **Varchar** |

**Table-3: Song**

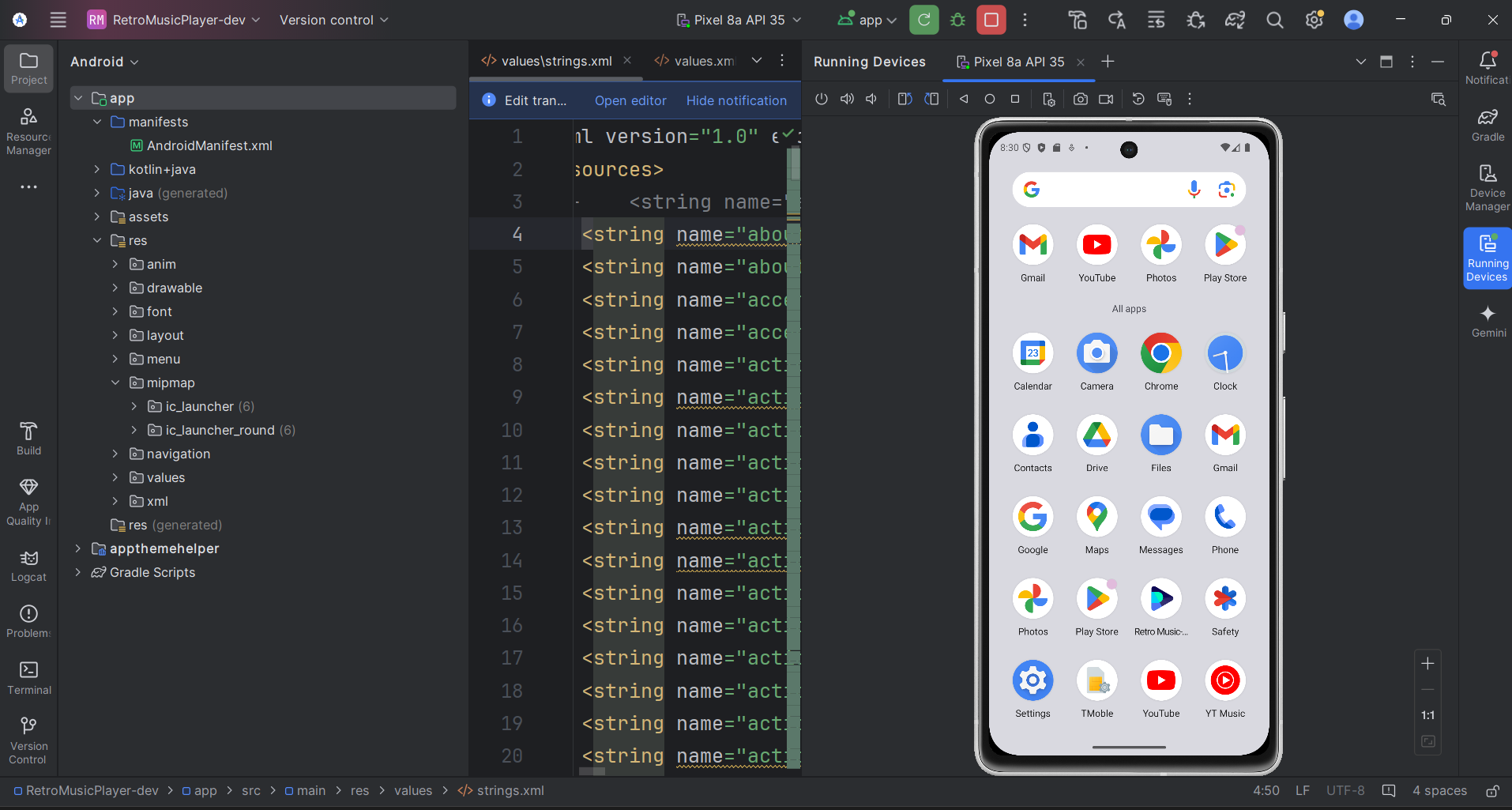
|  |  |
| --- | --- |
| **Field** | **Data Type** |
| **Playlist** | **Varchar** |
| **Songs** | **Ints** |
| **Albums** | **Ints** |
| **Artists** | **Varchar** |

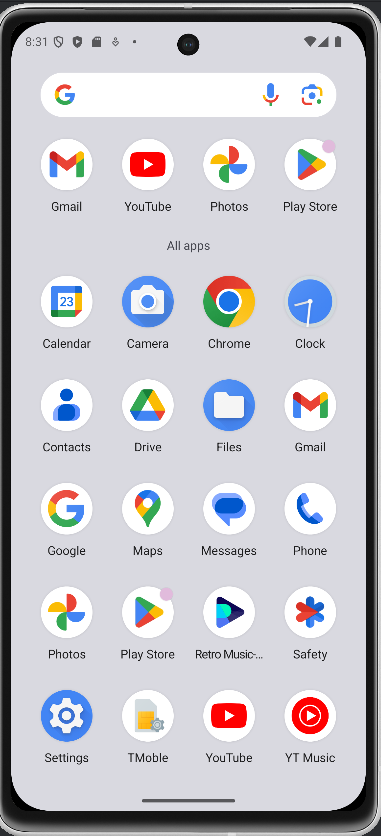
**Table-4: Like**

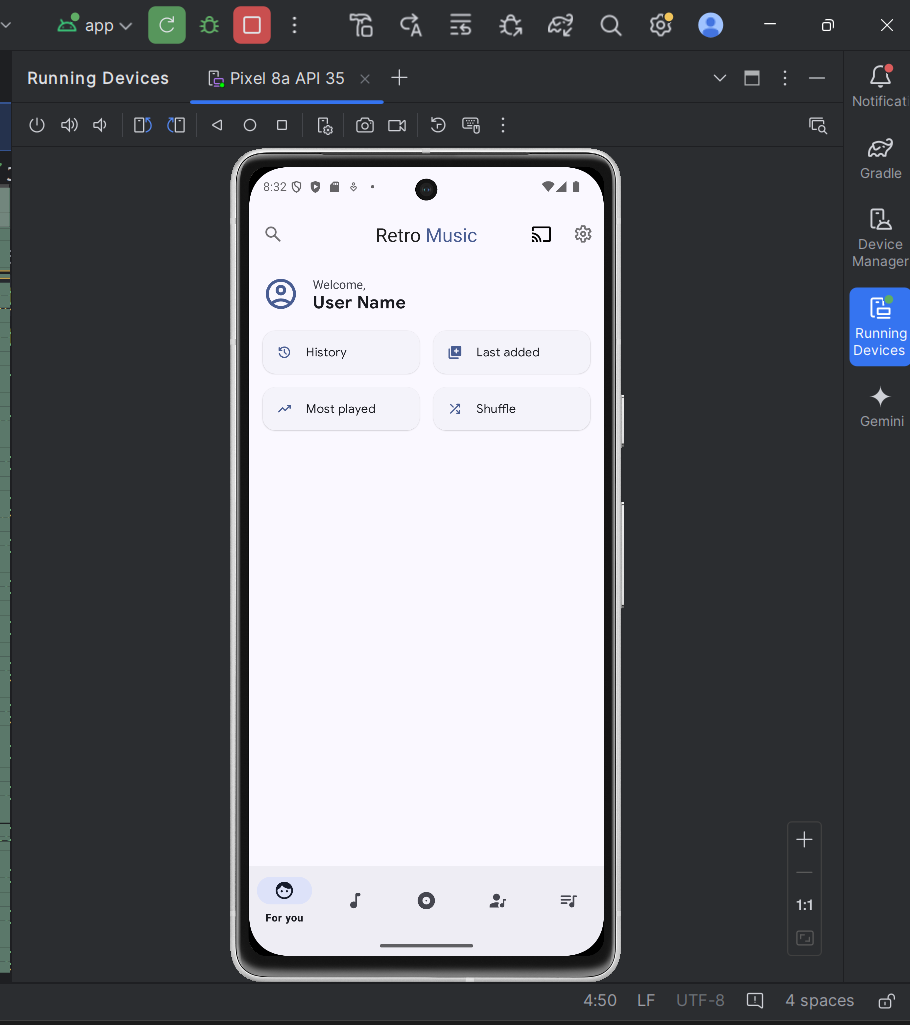
|  |  |
| --- | --- |
| **Field** | **Data Type** |
| **Like** | **Varchar** |

# Chapter - 5 User Interface Frontend

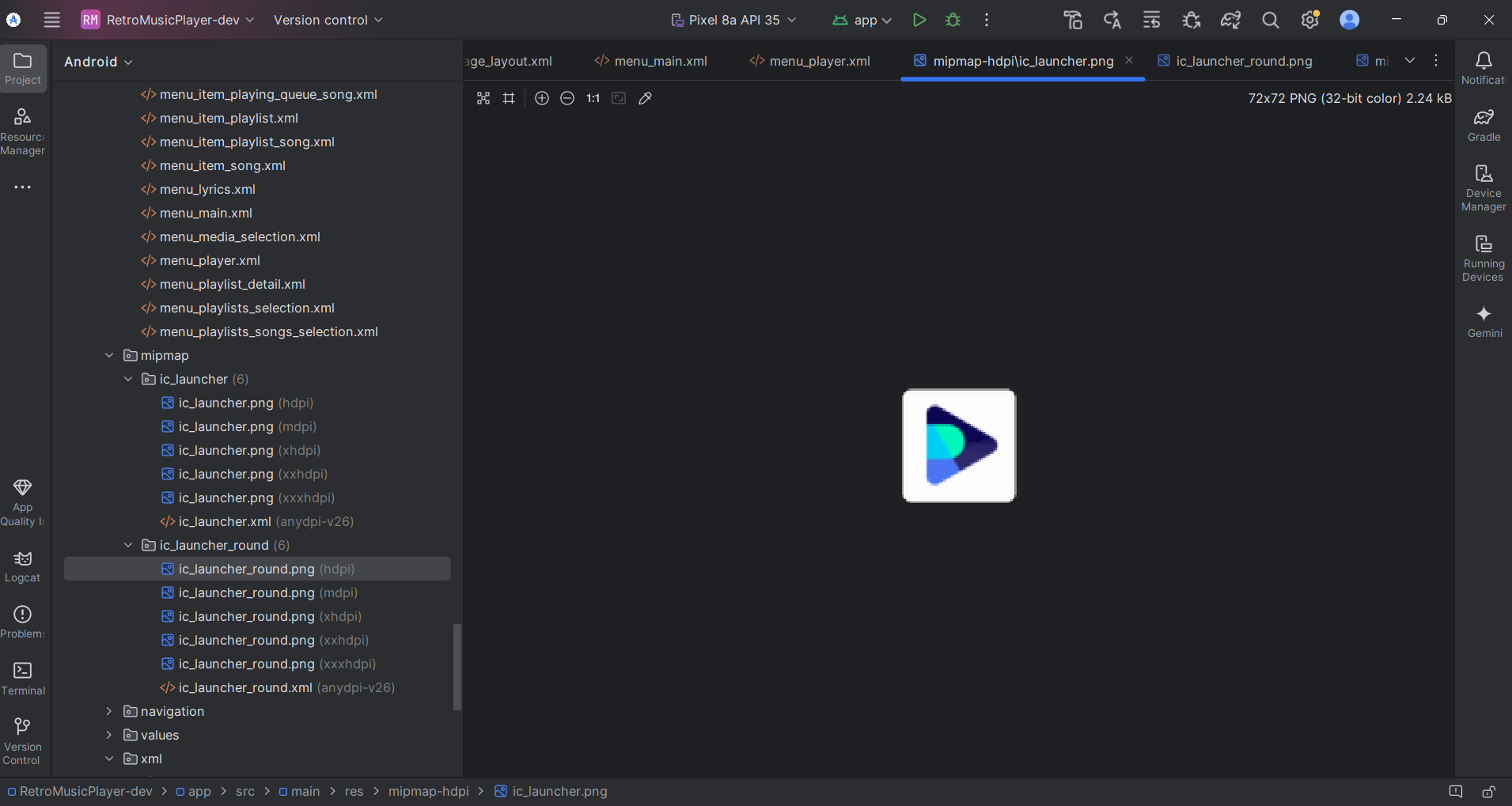
## 5.1 User Layout

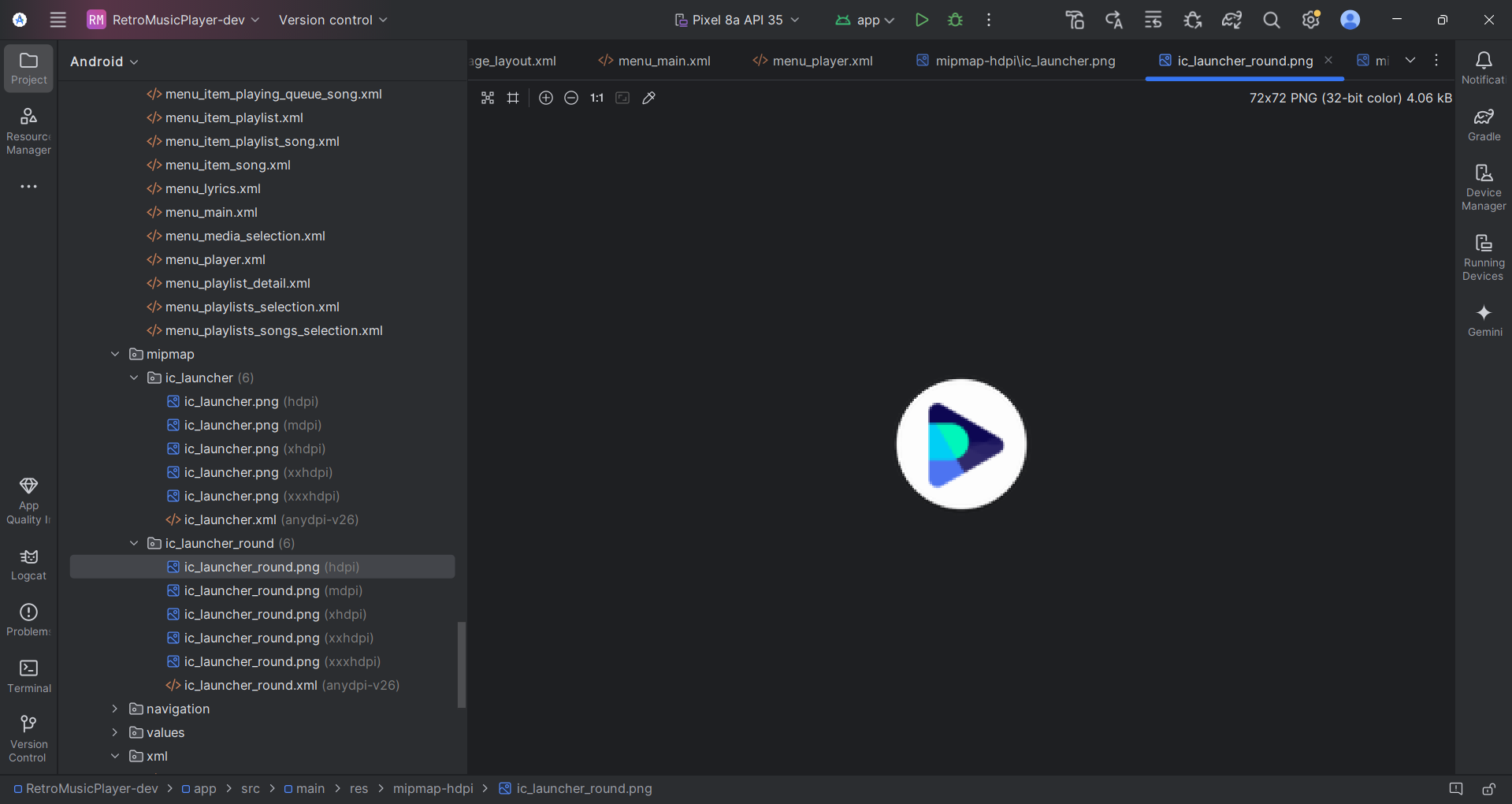






## 5.2 Retro Music Player Logo





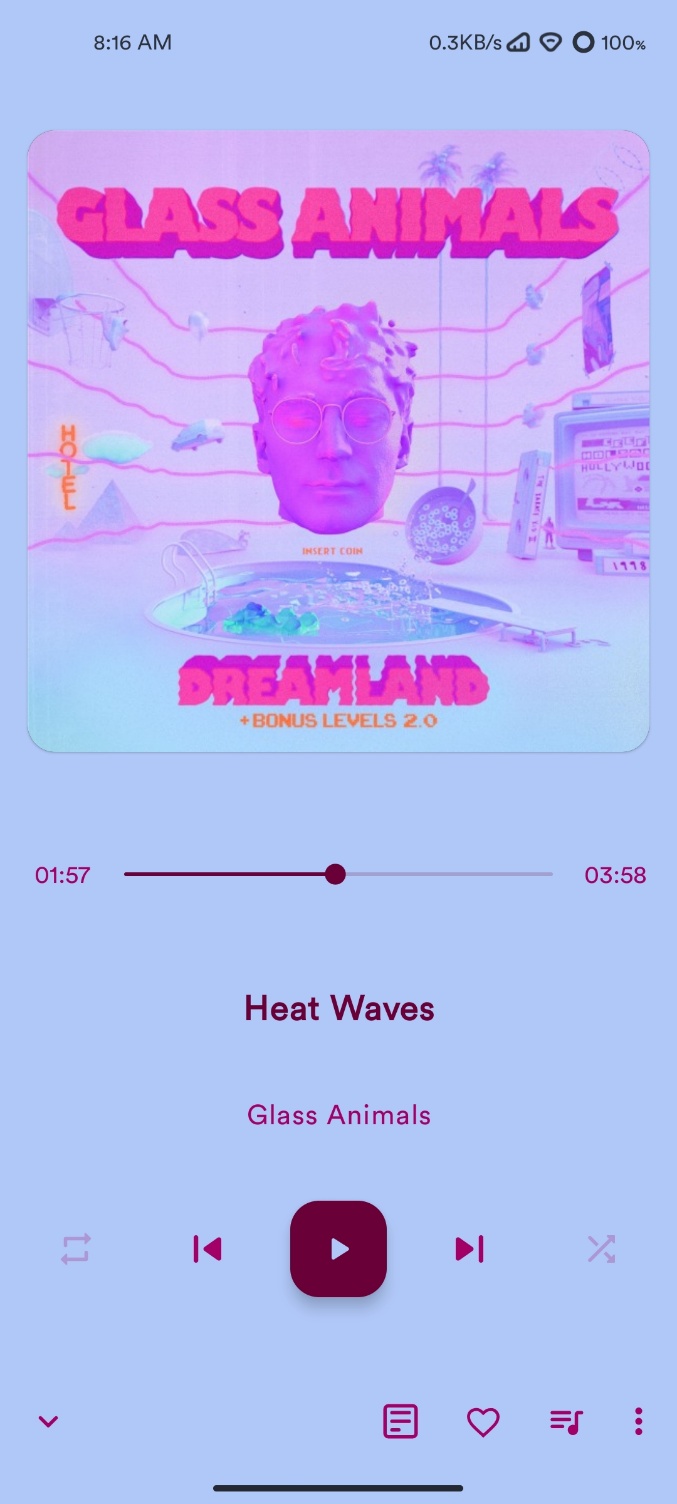
<resources xmlns:tools="http://schemas.android.com/tools">  
  
 <style name="Theme.RetroMusic" parent="Theme.RetroMusic.Base" />  
  
 <style name="Theme.RetroMusic.Light" parent="Theme.RetroMusic.Base.Light" />  
  
 <style name="Theme.RetroMusic.Black" parent="Theme.RetroMusic.Base.Black" />  
  
 <style name="Theme.RetroMusic.FollowSystem" parent="Theme.RetroMusic.Base.Adaptive">  
 <item name="android:windowBackground">?colorSurface</item>  
 </style>  
  
 <style name="Theme.RetroMusic.MD3" parent="@style/Theme.Material3.DayNight.NoActionBar">  
 <item name="roundSelector">@drawable/round\_selector</item>  
 <item name="rectSelector">@drawable/rect\_selector</item>  
 <item name="materialCardViewStyle">@style/Widget.Material3.CardView.Elevated</item>  
 <item name="materialButtonStyle">@style/MaterialButtonTheme</item>  
 </style>  
  
 <style name="Theme.RetroMusic.Notification" parent="@android:style/TextAppearance.StatusBar.EventContent" />  
  
 <style name="Theme.RetroMusic.Notification.Title" parent="@android:style/TextAppearance.StatusBar.EventContent.Title" />  
  
 <style name="OverFlowButton">  
 <item name="srcCompat">@drawable/ic\_more\_vert</item>  
 <item name="android:layout\_width">48dp</item>  
 <item name="android:layout\_height">48dp</item>  
 <item name="android:scaleType">center</item>  
 <item name="android:background">?attr/roundSelector</item>  
 <item name="android:focusableInTouchMode">false</item>  
 <item name="android:focusable">false</item>  
 </style>  
  
 <style name="MusicProgressSlider" parent="MusicProgressSliderParent">  
 <item name="android:padding">0dp</item>  
 </style>  
  
 <style name="Toolbar" parent="Widget.Material3.Toolbar">  
 <item name="popupTheme">?toolbarPopupTheme</item>  
 <item name="android:transitionName" tools:ignore="NewApi">toolbar</item>  
 <item name="layout\_collapseMode">pin</item>  
 <item name="contentInsetStartWithNavigation">0dp</item>  
 <item name="contentInsetStart">0dp</item>  
 <item name="titleMarginStart">16dp</item>  
 </style>  
  
 <style name="mcab\_overflow\_style" parent="Widget.AppCompat.ActionButton.Overflow">  
 <item name="srcCompat">@drawable/ic\_more\_vert</item>  
 <item name="android:tint">?colorSurface</item>  
 </style>  
  
 <style name="ToolbarTextAppearanceNormal">  
 <item name="android:textStyle">bold</item>  
 <item name="android:textAllCaps">false</item>  
 <item name="android:textSize">20sp</item>  
 <item name="android:letterSpacing">0.0125</item>  
 <item name="android:textColor">?android:attr/textColorPrimary</item>  
 </style>  
  
 <style name="SubTitleTextAppearance">  
 <item name="android:textAppearance">@style/TextViewOverline</item>  
 <item name="android:layout\_gravity">start|center\_vertical</item>  
 <item name="android:padding">12dp</item>  
 <item name="android:layout\_width">wrap\_content</item>  
 <item name="android:layout\_height">wrap\_content</item>  
 </style>  
  
 <style name="BottomSheetItemTextAppearance" parent="Widget.MaterialComponents.BottomNavigationView.Colored">  
 <item name="android:textSize">13sp</item>  
 </style>  
  
 <style name="Fab" parent="FabParent" />  
  
 <style name="TextViewNormal" parent="">  
 <item name="android:textSize">14sp</item>  
 </style>  
  
 <style name="TextViewNormalCompress" parent="TextAppearance.MaterialComponents.Caption">  
 <item name="android:textSize">14sp</item>  
 </style>  
  
 <style name="TextViewHeadline4.Compress" parent="TextAppearance.MaterialComponents.Headline4">  
 <item name="android:textSize">32sp</item>  
 </style>  
  
 <style name="TextViewHeadline4" parent="TextAppearance.MaterialComponents.Headline4" />  
  
 <style name="TextViewHeadline5" parent="TextAppearance.MaterialComponents.Headline5" />  
  
 <style name="TextViewCaption" parent="TextAppearance.MaterialComponents.Caption" />  
  
 <style name="TextViewHeadline6" parent="TextAppearance.MaterialComponents.Headline6" />  
  
 <style name="TextViewHeadline3" parent="TextAppearance.MaterialComponents.Headline3" />  
  
 <style name="TextViewHeadline2" parent="TextAppearance.MaterialComponents.Headline2" />  
  
 <style name="TextViewSubtitle1" parent="TextAppearance.MaterialComponents.Subtitle1" />  
  
 <style name="TextViewSubtitle2" parent="TextAppearance.MaterialComponents.Subtitle2" />  
  
 <style name="TextViewBody1" parent="TextAppearance.MaterialComponents.Body1" />  
  
 <style name="TextViewButton" parent="TextAppearance.MaterialComponents.Button" />  
  
 <style name="TextViewBody2" parent="TextAppearance.MaterialComponents.Body2" />  
  
 <style name="TextViewOverline" parent="TextAppearance.MaterialComponents.Overline" />  
  
 <style name="TopCornerCardView">  
 <item name="cornerFamilyTopLeft">rounded</item>  
 <item name="cornerFamilyTopRight">rounded</item>  
 <item name="cornerSizeTopLeft">16dp</item>  
 <item name="cornerSizeTopRight">16dp</item>  
 </style>  
  
 <style name="ClassicThemeOverLay">  
 <item name="cornerFamily">cut</item>  
 <item name="cornerSize">0dp</item>  
 </style>  
  
 <style name="Theme.RetroMusic.SplashScreen" parent="Theme.AppCompat.DayNight.NoActionBar">  
 <item name="android:windowBackground">@drawable/splash</item>  
 <item name="android:statusBarColor">@android:color/transparent</item>  
 </style>  
  
 <style name="MaterialPopupMenuStyle" parent="Widget.Material3.PopupMenu">  
 <item name="android:popupBackground">@drawable/popup\_background</item>  
 </style>  
  
 <style name="CenteredCheckBoxTheme">  
 <item name="checkboxStyle">@style/CheckBoxStyle</item>  
 </style>  
  
 <style name="CheckBoxStyle" parent="@style/Widget.AppCompat.CompoundButton.CheckBox">  
 <item name="android:gravity">center\_vertical|end</item>  
 </style>  
  
 <style name="MaterialAlertDialogTheme" parent="ThemeOverlay.Material3.MaterialAlertDialog" />  
  
 <style name="AppTextAppearance.MaterialAlertDialog.Button" parent="Widget.MaterialComponents.Button.TextButton">  
 <item name="android:textSize">16sp</item>  
 <item name="android:textStyle">bold</item>  
 <item name="android:padding">0dp</item>  
 </style>  
  
 <style name="AppTextAppearance.MaterialAlertDialog.Body" parent="MaterialAlertDialog.Material3.Body.Text">  
 <item name="android:textAppearance">@style/TextViewBody1</item>  
 <item name="android:paddingTop">16dp</item>  
 </style>  
  
 <style name="AppTextAppearance.MaterialAlertDialog.Title" parent="MaterialAlertDialog.Material3.Title.Text">  
 <item name="android:textAppearance">@style/TextViewHeadline6</item>  
 <item name="android:textStyle">bold</item>  
 </style>  
  
 <style name="MaterialButtonTheme" parent="Widget.Material3.Button">  
 <item name="cornerRadius">6dp</item>  
 <item name="iconGravity">textStart</item>  
 <item name="iconTint">?attr/colorControlNormal</item>  
 <item name="android:textColor">?android:attr/textColorPrimary</item>  
 <item name="android:textAppearance">@style/TextViewNormal</item>  
 <item name="android:textAllCaps">false</item>  
 <item name="android:paddingTop">@dimen/button\_padding\_vertical</item>  
 <item name="android:paddingBottom">@dimen/button\_padding\_vertical</item>  
 </style>  
  
 <style name="BottomSheetStyle" parent="Widget.Material3.BottomSheet">  
 <item name="paddingBottomSystemWindowInsets">false</item>  
 <item name="android:maxWidth">@empty</item>  
 <item name="shapeAppearance">?attr/shapeAppearanceCornerMedium</item>  
 </style>  
  
 <style name="MaterialCardViewStroke">  
 <item name="cardUseCompatPadding">true</item>  
 <item name="strokeWidth">1dp</item>  
 <item name="strokeColor">?android:attr/colorButtonNormal</item>  
 <item name="cardElevation">0dp</item>  
 </style>  
  
 <!-- This will set the fade in animation on all your activities by default -->  
 <style name="WindowAnimationTransition">  
 <item name="android:windowEnterAnimation">@android:anim/fade\_in</item>  
 <item name="android:windowExitAnimation">@android:anim/fade\_out</item>  
 </style>  
  
 <style name="circleImageView" parent="ShapeAppearance.MaterialComponents">  
 <item name="cornerSize">40dp</item>  
 </style>  
  
 <style name="SearchChipStyle" parent="Widget.Material3.Chip.Filter">  
 <item name="android:checked">false</item>  
 <item name="android:textSize">16sp</item>  
 <item name="checkedIconVisible">false</item>  
 <item name="chipEndPadding">10dp</item>  
 <item name="chipMinHeight">40dp</item>  
 <item name="chipStartPadding">10dp</item>  
 </style>  
 <!--Bottom Sheet Dialog Style-->  
 <style name="BottomSheetDialogStyle" parent="Theme.Design.BottomSheetDialog">  
 <item name="android:windowIsFloating">false</item>  
 <item name="android:statusBarColor">@android:color/transparent</item>  
 <item name="android:windowSoftInputMode">adjustResize|stateVisible</item>  
 </style>  
  
 <style name="HomeActionButton" parent="Widget.Material3.Button.ElevatedButton.Icon">  
 <item name="android:paddingTop">16dp</item>  
 <item name="android:paddingBottom">16dp</item>  
 <item name="android:textColor">?attr/colorOnSurface</item>  
 <item name="shapeAppearanceOverlay">  
 @style/ShapeAppearanceOverlay.Material3.FloatingActionButton  
 </item>  
 <item name="iconPadding">16dp</item>  
 <item name="iconGravity">start</item>  
 <item name="android:gravity">start|center\_vertical</item>  
 <item name="android:textAppearance">@style/TextViewBody2</item>  
 </style>  
  
 <style name="CarTheme" parent="Theme.AppCompat.Light.NoActionBar">  
 <item name="colorPrimaryDark">@color/md\_deep\_purple\_700</item>  
 <item name="colorAccent">@color/md\_deep\_purple\_A400</item>  
 <item name="colorPrimary">@color/md\_deep\_purple\_500</item>  
 </style>  
  
 <style name="Theme.RetroMusic.Dialog" parent="Theme.Material3.DayNight.Dialog">  
 <item name="windowNoTitle">true</item>  
 <item name="android:windowIsFloating">true</item>  
 <item name="android:windowIsTranslucent">true</item>  
 <item name="android:windowBackground">@drawable/rounded\_drawable</item>  
 <item name="android:windowFrame">@null</item>  
 <item name="background">@color/transparent</item>  
 </style>  
  
 <style name="FontThemeOverlay">  
 <item name="fontFamily">@font/manrope</item>  
 <item name="android:fontFamily">@font/manrope</item>  
 </style>  
  
 <style name="CircleFABOverlay">  
 <item name="floatingActionButtonStyle">  
 @style/Widget.MaterialComponents.FloatingActionButton  
 </item>  
 </style>  
  
 <style name="RoundedFABOverlay">  
 <item name="floatingActionButtonStyle">  
 @style/Widget.Material3.FloatingActionButton.Primary  
 </item>  
 </style>  
  
 <style name="Theme.AppWidget" parent="">  
 <item name="colorSurface">@color/md\_black\_1000</item>  
 </style>  
  
 <style name="ShapeAppearance.Material3.Circle" parent="">  
 <item name="cornerSize">50%</item>  
 </style>  
  
 <style name="Widget.Retro.Slider" parent="Widget.Material3.Slider.Legacy">  
 <item name="labelBehavior">gone</item>  
 <item name="thumbRadius">6dp</item>  
 <item name="haloRadius">18dp</item>  
 </style>  
  
 <style name="Widget.Retro.MD3.Custom.Slider" parent="Widget.Material3.Slider">  
 <item name="labelBehavior">gone</item>  
 <item name="thumbHeight">26dp</item>  
 <item name="thumbWidth">4dp</item>  
 <item name="thumbTrackGapSize">4dp</item>  
 <item name="trackHeight">14dp</item>  
 <item name="thumbElevation">0dp</item>  
 <item name="background">@android:color/transparent</item>  
 <item name="haloRadius">0dp</item>  
 </style>  
</resources>

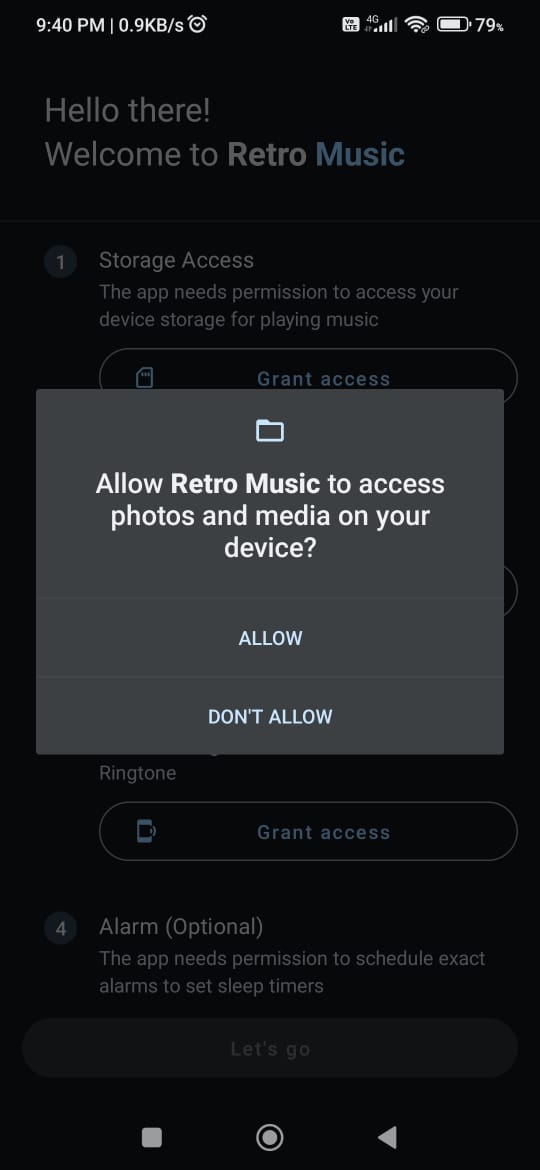
## 5.3 User Layout Xml Code

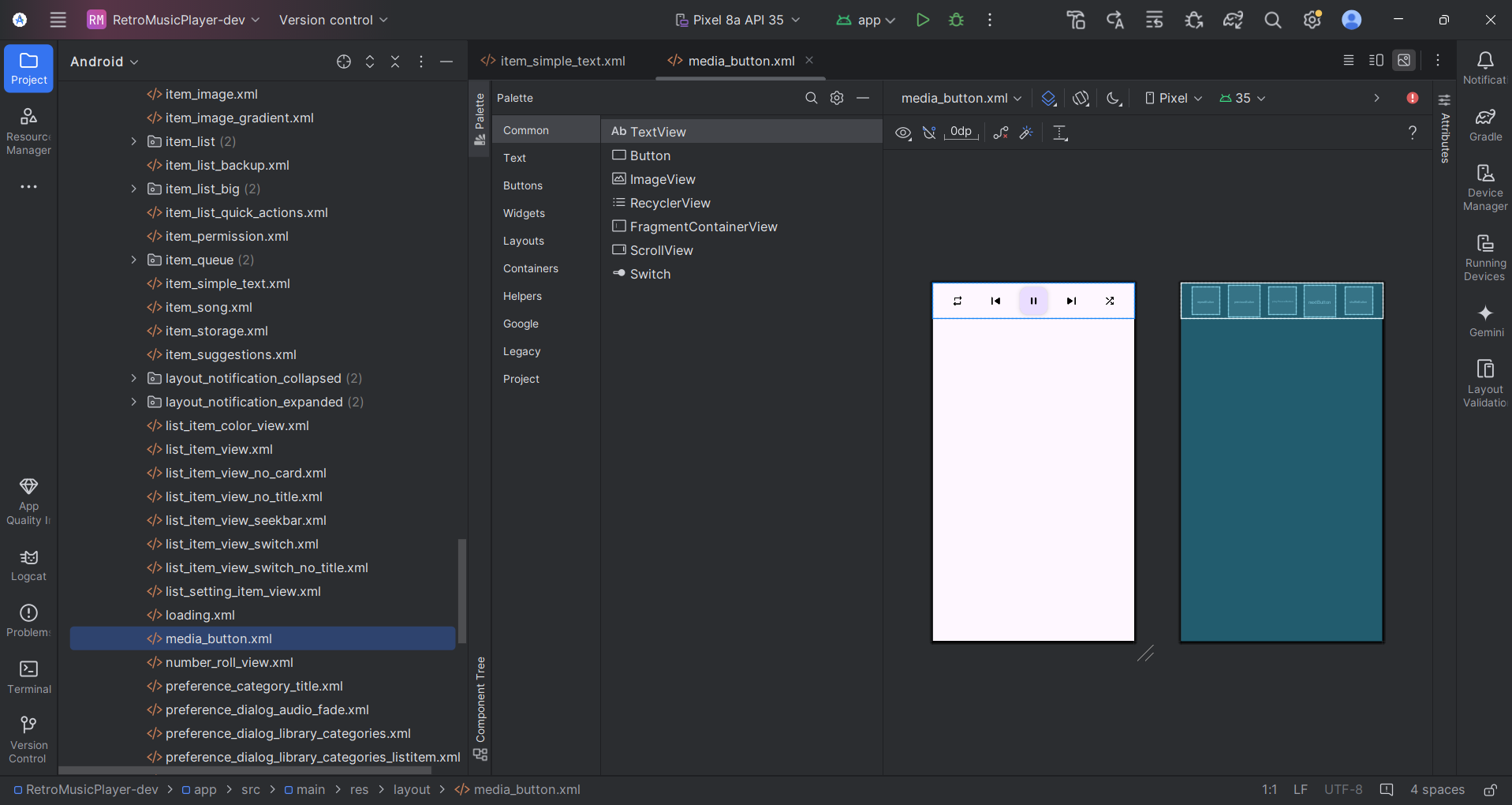
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
  
 <uses-permission  
 android:name="android.permission.BLUETOOTH"  
 android:maxSdkVersion="30" />  
 <uses-permission android:name="android.permission.DISABLE\_KEYGUARD" />  
 <uses-permission  
 android:name="android.permission.SCHEDULE\_EXACT\_ALARM"  
 tools:ignore="ProtectedPermissions" />  
 <uses-permission android:name="android.permission.VIBRATE" />  
 <uses-permission android:name="android.permission.WAKE\_LOCK" />  
 <uses-permission  
 android:name="android.permission.READ\_EXTERNAL\_STORAGE"  
 android:maxSdkVersion="32" />  
 <uses-permission android:name="android.permission.READ\_MEDIA\_AUDIO" />  
 <uses-permission android:name="android.permission.POST\_NOTIFICATIONS" />  
 <uses-permission  
 android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"  
 android:maxSdkVersion="29" />  
 <uses-permission android:name="android.permission.INTERNET" />  
 <uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />  
 <uses-permission android:name="android.permission.BROADCAST\_STICKY" />  
 <uses-permission android:name="android.permission.FOREGROUND\_SERVICE" />  
 <uses-permission android:name="android.permission.FOREGROUND\_SERVICE\_MEDIA\_PLAYBACK" />  
 <uses-permission  
 android:name="android.permission.WRITE\_SETTINGS"  
 tools:ignore="ProtectedPermissions" />  
 <uses-permission android:name="android.permission.USE\_FULL\_SCREEN\_INTENT" />  
 <uses-permission  
 android:name="android.permission.BLUETOOTH\_CONNECT"  
 android:usesPermissionFlags="neverForLocation"  
 tools:targetApi="s" />  
  
 <application  
 android:name=".App"  
 android:allowBackup="@bool/allowBackup"  
 android:appCategory="audio"  
 android:configChanges="locale|layoutDirection"  
 android:enableOnBackInvokedCallback="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:localeConfig="@xml/locales\_config"  
 android:requestLegacyExternalStorage="true"  
 android:restoreAnyVersion="true"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.RetroMusic.FollowSystem"  
 android:usesCleartextTraffic="true"  
 tools:ignore="UnusedAttribute">  
 <activity  
 android:name=".activities.MainActivity"  
 android:exported="true"  
 android:launchMode="singleTop"  
 android:theme="@style/Theme.RetroMusic.SplashScreen">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
 <action android:name="android.intent.action.MUSIC\_PLAYER" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 <category android:name="android.intent.category.APP\_MUSIC" />  
 <category android:name="android.intent.category.DEFAULT" />  
 </intent-filter>  
 <intent-filter>  
 <action android:name="android.media.action.MEDIA\_PLAY\_FROM\_SEARCH" />  
 <category android:name="android.intent.category.DEFAULT" />  
 </intent-filter>  
 <intent-filter>  
 <action android:name="android.intent.action.VIEW" />  
  
 <category android:name="android.intent.category.DEFAULT" />  
  
 <data android:scheme="content" />  
 <data android:mimeType="audio/\*" />  
 <data android:mimeType="application/ogg" />  
 <data android:mimeType="application/x-ogg" />  
 <data android:mimeType="application/itunes" />  
 </intent-filter>  
 <intent-filter>  
 <action android:name="android.intent.action.VIEW" />  
  
 <category android:name="android.intent.category.DEFAULT" />  
  
 <data android:scheme="file" />  
 <data android:mimeType="audio/\*" />  
 <data android:mimeType="application/ogg" />  
 <data android:mimeType="application/x-ogg" />  
 <data android:mimeType="application/itunes" />  
 </intent-filter>  
 <intent-filter>  
 <action android:name="android.intent.action.VIEW" />  
  
 <category android:name="android.intent.category.DEFAULT" />  
 <category android:name="android.intent.category.BROWSABLE" />  
  
 <data android:scheme="http" />  
 <data android:mimeType="audio/\*" />  
 <data android:mimeType="application/ogg" />  
 <data android:mimeType="application/x-ogg" />  
 <data android:mimeType="application/itunes" />  
 </intent-filter>  
 <intent-filter tools:ignore="AppLinkUrlError">  
 <action android:name="android.intent.action.VIEW" />  
  
 <category android:name="android.intent.category.DEFAULT" />  
  
 <data android:mimeType="vnd.android.cursor.dir/playlist" />  
 <data android:mimeType="vnd.android.cursor.dir/albums" />  
 <data android:mimeType="vnd.android.cursor.dir/artists" />  
 </intent-filter>  
 <intent-filter>  
 <action android:name="com.cyanogenmod.eleven.AUDIO\_PLAYER" />  
  
 <category android:name="android.intent.category.DEFAULT" />  
 </intent-filter>  
 <intent-filter>  
 <action android:name="android.intent.action.PICK" />  
  
 <category android:name="android.intent.category.DEFAULT" />  
 <category android:name="android.intent.category.OPENABLE" />  
  
 <data android:mimeType="vnd.android.cursor.dir/audio" />  
 </intent-filter>  
 </activity>  
 <activity android:name=".activities.tageditor.AlbumTagEditorActivity" />  
 <activity android:name=".activities.tageditor.SongTagEditorActivity" />  
 <activity android:name=".activities.SupportDevelopmentActivity" />  
 <activity android:name=".activities.LicenseActivity" />  
 <activity android:name=".activities.bugreport.BugReportActivity" />  
 <activity android:name=".activities.ShareInstagramStory" />  
 <activity android:name=".activities.DriveModeActivity" />  
 <activity android:name=".activities.PermissionActivity" />  
 <activity  
 android:name=".activities.LockScreenActivity"  
 android:excludeFromRecents="true"  
 android:launchMode="singleTask"  
 android:showOnLockScreen="true" />  
 <activity  
 android:name=".fragments.backup.RestoreActivity"  
 android:excludeFromRecents="false"  
 android:exported="true"  
 android:label="@string/restore"  
 android:theme="@style/Theme.RetroMusic.Dialog">

<?xml version="1.0" encoding="UTF-8" standalone="no"?>  
<resources>  
 <string name="app\_name">Melodify</string>  
 <string name="about\_album\_label">About %s</string>  
 <string name="about\_settings\_summary">Team, social links</string>  
 <string name="accent\_color">Accent color</string>  
 <string name="accent\_color\_desc">The theme accent color, defaults to purple</string>  
 <string name="action\_about">About</string>  
 <string name="action\_add\_to\_blacklist">Add to Blacklist</string>  
 <string name="action\_add\_to\_favorites">Add to favorites</string>  
 <string name="action\_add\_to\_playing\_queue">Add to playing queue</string>  
 <string name="action\_add\_to\_playlist">Add to playlist</string>  
 <string name="action\_cancel">Cancel</string>  
 <string name="action\_cast">Cast</string>  
 <string name="action\_clear\_playing\_queue">Clear playing queue</string>  
 <string name="action\_cycle\_repeat">Cycle repeat mode</string>  
 <string name="action\_delete">Delete</string>  
 <string name="action\_delete\_from\_device">Delete from device</string>  
 <string name="action\_details">Details</string>  
 <string name="action\_edit">Edit</string>  
 <string name="action\_go\_to\_album">Go to album</string>  
 <string name="action\_go\_to\_artist">Go to artist</string>  
 <string name="action\_go\_to\_genre">Go to genre</string>  
 <string name="action\_go\_to\_lyrics">Go to Lyrics</string>  
 <string name="action\_go\_to\_start\_directory">Go to start directory</string>  
 <string name="action\_grant">Grant</string>  
 <string name="action\_grid\_size">Grid size</string>  
 <string name="action\_grid\_size\_land">Grid size (land)</string>  
 <string name="action\_new\_playlist">New playlist</string>  
 <string name="action\_next">Next</string>  
 <string name="action\_play">Play</string>  
 <string name="action\_play\_all">Play all</string>  
 <string name="action\_play\_next">Play next</string>  
 <string name="action\_play\_pause">Play/Pause</string>  
 <string name="action\_previous">Previous</string>  
 <string name="action\_remove\_from\_favorites">Remove from favorites</string>  
 <string name="action\_remove\_from\_playing\_queue">Remove from playing queue</string>  
 <string name="action\_remove\_from\_playlist">Remove from playlist</string>  
 <string name="action\_rename">Rename</string>  
 <string name="action\_save\_playing\_queue">Save playing queue</string>  
 <string name="action\_scan">Scan</string>  
 <string name="action\_search">Search</string>  
 <string name="action\_set">Start</string>  
 <string name="action\_set\_as\_ringtone">Set as ringtone</string>  
 <string name="action\_set\_as\_start\_directory">Set as start directory</string>  
 <string name="action\_settings">"Settings"</string>  
 <string name="action\_share">Share</string>  
 <string name="action\_shuffle\_all">Shuffle all</string>  
 <string name="action\_shuffle\_playlist">Shuffle playlist</string>  
 <string name="action\_sleep\_timer">Sleep timer</string>  
 <string name="action\_sort\_order">Sort order</string>  
 <string name="action\_tag\_editor">Tag editor</string>  
 <string name="action\_toggle\_favorite">Toggle favorite</string>  
 <string name="action\_toggle\_shuffle">Toggle shuffle mode</string>  
 <string name="adaptive">Adaptive</string>  
 <string name="add\_action">Add</string>  
 <string name="add\_playlist\_title">"Add to playlist"</string>  
 <string name="add\_time\_framed\_lryics">Add Time Framed Lyrics</string>  
 <string name="added\_song\_count\_to\_playlist">Added %1$d song(s) to %2$s</string>  
 <string name="added\_title\_to\_playing\_queue">"Added 1 title to the playing queue."</string>  
 <string name="added\_x\_titles\_to\_playing\_queue">Added %1$d titles to the playing queue.</string>  
 <string name="album">Album</string>

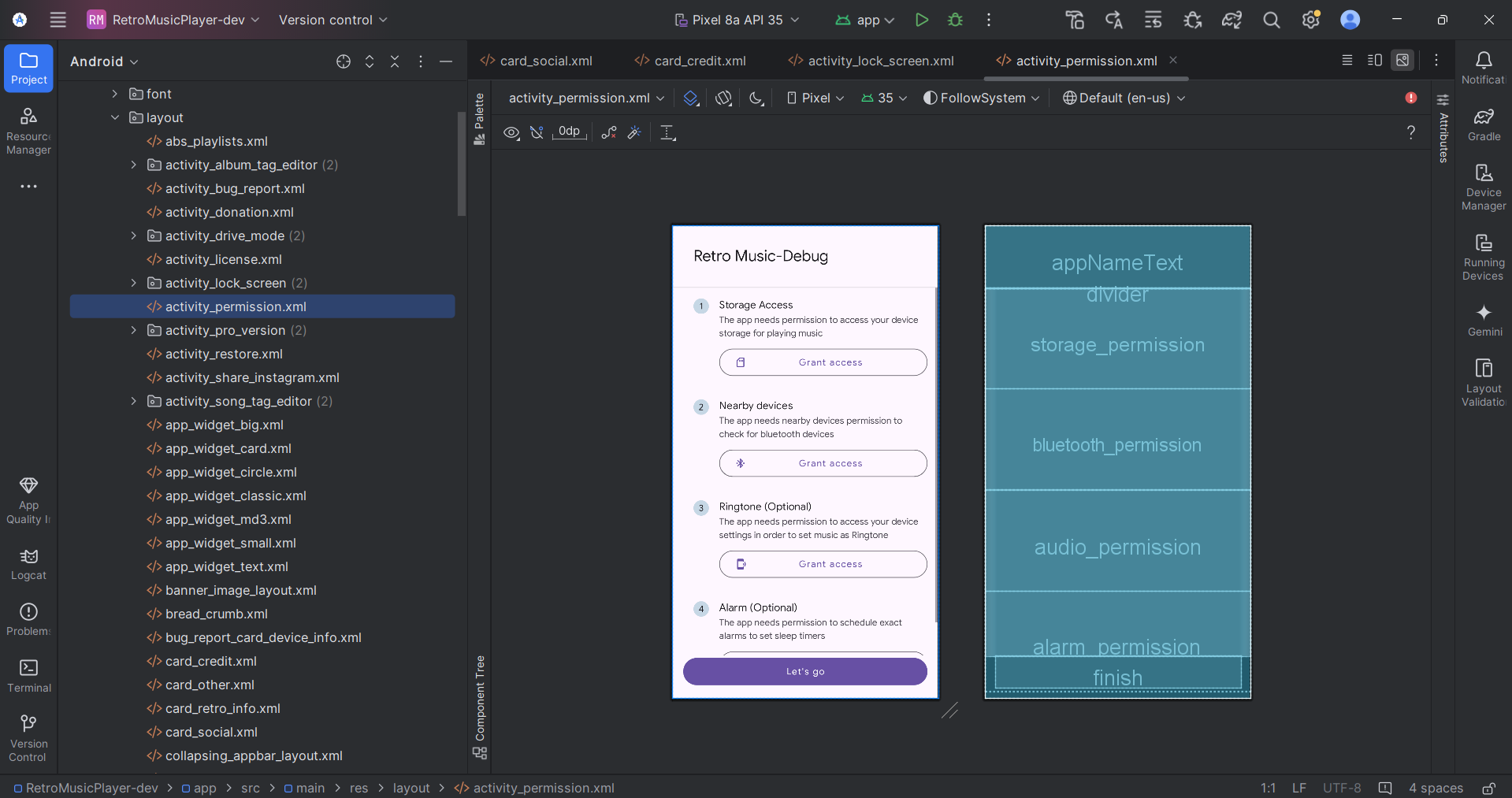
<?xml version="1.0" encoding="utf-8"?>  
<resources xmlns:tools="http://schemas.android.com/tools">  
 <string name="transition\_album\_art" translatable="false">album\_art\_transition</string>  
 <string name="transition\_user\_image" translatable="false">user\_image\_transition</string>  
 <string name="transition\_album\_name" translatable="false" tools:keep="@string/transition\_album\_name">album\_art\_transition</string>  
 <string name="transition\_album\_text" translatable="false" tools:keep="@string/transition\_album\_text">album\_art\_transition</string>  
 <string name="transition\_artist\_image" translatable="false">artist\_image\_transition</string>  
 <string name="transition\_artist\_name" translatable="false" tools:keep="@string/transition\_artist\_name">artist\_image\_transition</string>  
 <string name="transition\_artist\_text" translatable="false" tools:keep="@string/transition\_artist\_text">artist\_image\_transition</string>  
 <string name="transition\_lyrics" translatable="false">toolbar\_transition</string>  
</resources>



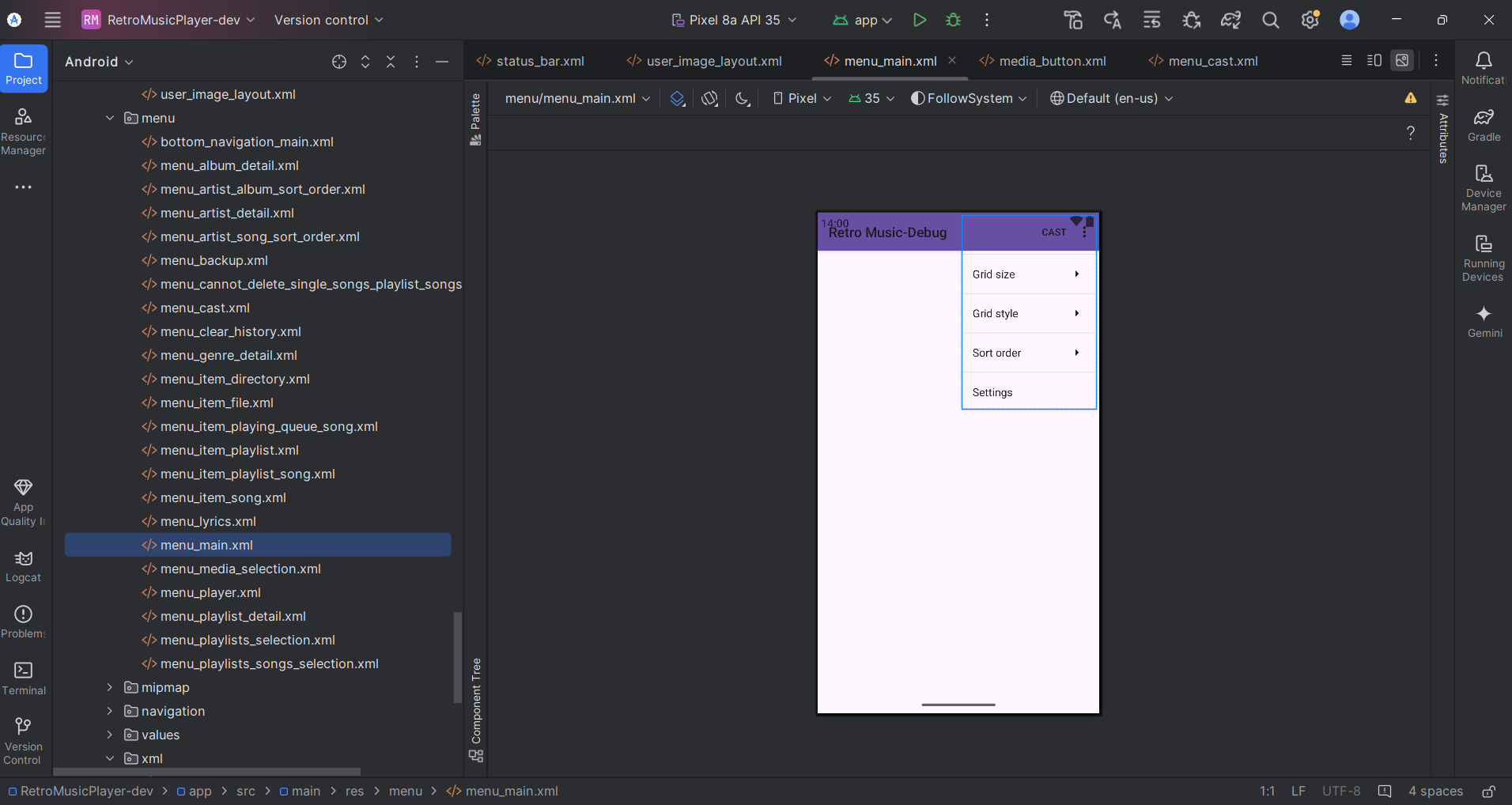




<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:layoutDirection="ltr"  
 android:paddingStart="4dp"  
 android:paddingEnd="4dp"  
 android:clipToPadding="false">  
  
 <androidx.appcompat.widget.AppCompatImageButton  
 android:id="@+id/repeatButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="?attr/roundSelector"  
 android:padding="16dp"  
 android:scaleType="fitCenter"  
 app:layout\_constraintBottom\_toBottomOf="@+id/previousButton"  
 app:layout\_constraintEnd\_toStartOf="@+id/previousButton"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/previousButton"  
 app:srcCompat="@drawable/ic\_repeat"  
 tools:ignore="MissingPrefix"  
 tools:tint="@color/md\_black\_1000" />  
  
 <androidx.appcompat.widget.AppCompatImageButton  
 android:id="@+id/previousButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="?attr/roundSelector"  
 android:padding="16dp"  
 android:scaleType="fitCenter"  
 app:layout\_constraintBottom\_toBottomOf="@+id/playPauseButton"  
 app:layout\_constraintEnd\_toStartOf="@+id/playPauseButton"  
 app:layout\_constraintStart\_toEndOf="@+id/repeatButton"  
 app:layout\_constraintTop\_toTopOf="@+id/playPauseButton"  
 app:srcCompat="@drawable/ic\_skip\_previous"  
 tools:ignore="MissingPrefix"  
 tools:tint="@color/md\_black\_1000" />  
  
 <com.google.android.material.floatingactionbutton.FloatingActionButton  
 android:id="@+id/playPauseButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginBottom="8dp"  
 android:background="?attr/roundSelector"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toStartOf="@+id/nextButton"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toEndOf="@+id/previousButton"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/ic\_pause\_white\_64dp"  
 tools:tint="@color/md\_black\_1000" />  
  
 <androidx.appcompat.widget.AppCompatImageButton  
 android:id="@+id/nextButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="?attr/roundSelector"  
 android:padding="16dp"  
 android:scaleType="fitCenter"  
 app:layout\_constraintBottom\_toBottomOf="@+id/playPauseButton"  
 app:layout\_constraintEnd\_toStartOf="@+id/shuffleButton"  
 app:layout\_constraintStart\_toEndOf="@+id/playPauseButton"  
 app:layout\_constraintTop\_toTopOf="@+id/playPauseButton"  
 app:srcCompat="@drawable/ic\_skip\_next"  
 tools:ignore="MissingPrefix"  
 tools:tint="@color/md\_black\_1000" />  
  
 <androidx.appcompat.widget.AppCompatImageButton  
 android:id="@+id/shuffleButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="?attr/roundSelector"  
 android:padding="16dp"  
 android:scaleType="fitCenter"  
 app:layout\_constraintBottom\_toBottomOf="@+id/nextButton"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toEndOf="@+id/nextButton"  
 app:layout\_constraintTop\_toTopOf="@+id/nextButton"  
 app:srcCompat="@drawable/ic\_shuffle"  
 tools:ignore="MissingPrefix"  
 tools:tint="@color/md\_black\_1000" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>



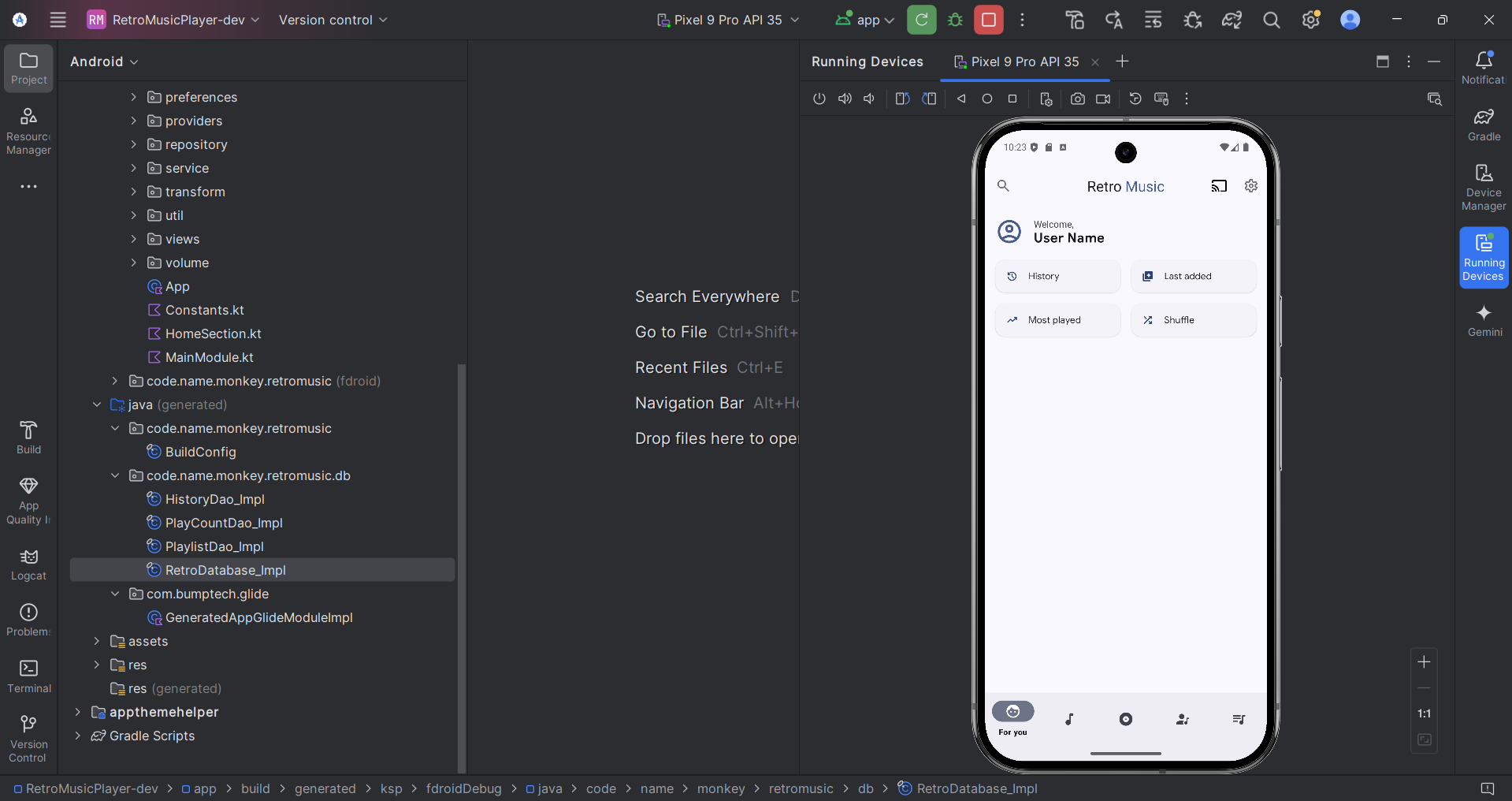
<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true">  
  
 <code.name.monkey.retromusic.views.BaselineGridTextView  
 android:id="@+id/appNameText"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:padding="32dp"  
 android:text="@string/app\_name"  
 android:textAppearance="@style/TextViewHeadline5"  
 android:textColor="?android:attr/textColorPrimary"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:lineHeightHint="32sp" />  
  
 <View  
 android:id="@+id/divider"  
 android:layout\_width="0dp"  
 android:layout\_height="1dp"  
 android:background="?attr/dividerHorizontal"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/appNameText" />  
  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:fillViewport="true"  
 app:layout\_constraintBottom\_toTopOf="@+id/finish"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/divider">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="@integer/permission\_orientation">  
  
 <code.name.monkey.retromusic.views.PermissionItem  
 android:id="@+id/storage\_permission"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="@integer/permission\_layout\_weight"  
 app:permissionButtonTitle="@string/grant\_access"  
 app:permissionIcon="@drawable/ic\_sd\_storage"  
 app:permissionTitle="@string/permission\_title"  
 app:permissionTitleNumber="1"  
 app:permissionTitleSubTitle="@string/permission\_summary" />  
  
 <code.name.monkey.retromusic.views.PermissionItem  
 android:id="@+id/bluetooth\_permission"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="@integer/permission\_layout\_weight"  
 android:visibility="gone"  
 app:permissionButtonTitle="@string/grant\_access"  
 app:permissionIcon="@drawable/ic\_bluetooth\_connect"  
 app:permissionTitle="@string/bluetooth\_title"  
 app:permissionTitleNumber="2"  
 app:permissionTitleSubTitle="@string/bluetooth\_summary"  
 tools:visibility="visible" />  
  
 <code.name.monkey.retromusic.views.PermissionItem  
 android:id="@+id/audio\_permission"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="@integer/permission\_layout\_weight"  
 android:visibility="gone"  
 app:permissionButtonTitle="@string/grant\_access"  
 app:permissionIcon="@drawable/ic\_phonelink\_ring"  
 app:permissionTitle="@string/ringtone\_title"  
 app:permissionTitleNumber="3"  
 app:permissionTitleSubTitle="@string/ringtone\_summary"  
 tools:visibility="visible" />  
  
 <code.name.monkey.retromusic.views.PermissionItem  
 android:id="@+id/alarm\_permission"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="@integer/permission\_layout\_weight"  
 android:visibility="gone"  
 app:permissionButtonTitle="@string/grant\_access"  
 app:permissionIcon="@drawable/ic\_phonelink\_ring"  
 app:permissionTitle="@string/permission\_alarm\_title"  
 app:permissionTitleNumber="4"  
 app:permissionTitleSubTitle="@string/permission\_alarm\_summary"  
 tools:visibility="visible" />  
 </LinearLayout>  
 </ScrollView>  
  
  
 <com.google.android.material.button.MaterialButton  
 android:id="@+id/finish"  
 style="@style/Widget.Material3.Button"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="16dp"  
 android:paddingVertical="12dp"  
 android:text="@string/lets\_go"  
 android:textAppearance="@style/TextViewButton"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent" />  
</androidx.constraintlayout.widget.ConstraintLayout>



<menu xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 tools:context=".DrawerActivity">  
 <item  
 android:id="@+id/action\_grid\_size"  
 android:icon="@drawable/ic\_grid\_size"  
 android:title="@string/action\_grid\_size"  
 app:showAsAction="never">  
 <menu>  
 <group android:checkableBehavior="single">  
 <item  
 android:id="@+id/action\_grid\_size\_1"  
 android:title="@string/grid\_size\_1" />  
 <item  
 android:id="@+id/action\_grid\_size\_2"  
 android:title="@string/grid\_size\_2" />  
 <item  
 android:id="@+id/action\_grid\_size\_3"  
 android:title="@string/grid\_size\_3" />  
 <item  
 android:id="@+id/action\_grid\_size\_4"  
 android:title="@string/grid\_size\_4" />  
 <item  
 android:id="@+id/action\_grid\_size\_5"  
 android:title="@string/grid\_size\_5" />  
 <item  
 android:id="@+id/action\_grid\_size\_6"  
 android:title="@string/grid\_size\_6" />  
 <item  
 android:id="@+id/action\_grid\_size\_7"  
 android:title="@string/grid\_size\_7" />  
 <item  
 android:id="@+id/action\_grid\_size\_8"  
 android:title="@string/grid\_size\_8" />  
 </group>  
 </menu>  
 </item>  
 <item  
 android:id="@+id/action\_layout\_type"  
 android:icon="@drawable/ic\_dashboard"  
 android:title="@string/grid\_style\_label"  
 app:showAsAction="never">  
 <menu>  
 <group android:checkableBehavior="single">  
 <item  
 android:id="@+id/action\_layout\_normal"  
 android:title="@string/normal" />  
 <item  
 android:id="@+id/action\_layout\_card"  
 android:title="@string/card" />  
 <item  
 android:id="@+id/action\_layout\_colored\_card"  
 android:title="@string/card\_color\_style" />  
 <item  
 android:id="@+id/action\_layout\_circular"  
 android:title="@string/circular" />  
 <item  
 android:id="@+id/action\_layout\_image"  
 android:title="@string/image" />  
 <item  
 android:id="@+id/action\_layout\_gradient\_image"  
 android:title="@string/image\_gradient" />  
 </group>  
 </menu>  
 </item>  
 <item  
 android:id="@+id/action\_sort\_order"  
 android:icon="@drawable/ic\_sort"  
 android:title="@string/action\_sort\_order"  
 app:showAsAction="never">  
 <menu></menu>  
 </item>  
 <item  
 android:orderInCategory="10"  
 android:id="@+id/action\_settings"  
 android:icon="@drawable/ic\_settings"  
 android:title="@string/action\_settings"  
 app:showAsAction="never" />  
 <item  
 android:id="@+id/action\_cast"  
 android:title="@string/action\_cast"  
 app:actionProviderClass="androidx.mediarouter.app.MediaRouteActionProvider"  
 app:showAsAction="always" />  
</menu>

## 5.4 Java Code

package code.name.retromusic.db;  
  
import androidx.annotation.NonNull;  
import androidx.room.DatabaseConfiguration;  
import androidx.room.InvalidationTracker;  
import androidx.room.RoomDatabase;  
import androidx.room.RoomOpenHelper;  
import androidx.room.migration.AutoMigrationSpec;  
import androidx.room.migration.Migration;  
import androidx.room.util.DBUtil;  
import androidx.room.util.TableInfo;  
import androidx.sqlite.db.SupportSQLiteDatabase;  
import androidx.sqlite.db.SupportSQLiteOpenHelper;  
import java.lang.Class;  
import java.lang.Override;  
import java.lang.String;  
import java.lang.SuppressWarnings;  
import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.HashMap;  
import java.util.HashSet;  
import java.util.List;  
import java.util.Map;  
import java.util.Set;  
import javax.annotation.processing.Generated;  
  
@Generated("androidx.room.RoomProcessor")  
@SuppressWarnings({"unchecked", "deprecation"})  
public final class RetroDatabase\_Impl extends RetroDatabase {  
 private volatile PlaylistDao \_playlistDao;  
  
 private volatile PlayCountDao \_playCountDao;  
  
 private volatile HistoryDao \_historyDao;  
  
 @Override  
 @NonNull  
 protected SupportSQLiteOpenHelper createOpenHelper(@NonNull final DatabaseConfiguration config) {  
 final SupportSQLiteOpenHelper.Callback \_openCallback = new RoomOpenHelper(config, new RoomOpenHelper.Delegate(24) {  
 @Override  
 public void createAllTables(@NonNull final SupportSQLiteDatabase db) {  
 db.execSQL("CREATE TABLE IF NOT EXISTS `PlaylistEntity` (`playlist\_id` INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, `playlist\_name` TEXT NOT NULL)");  
 db.execSQL("CREATE TABLE IF NOT EXISTS `SongEntity` (`song\_key` INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, `playlist\_creator\_id` INTEGER NOT NULL, `id` INTEGER NOT NULL, `title` TEXT NOT NULL, `track\_number` INTEGER NOT NULL, `year` INTEGER NOT NULL, `duration` INTEGER NOT NULL, `data` TEXT NOT NULL, `date\_modified` INTEGER NOT NULL, `album\_id` INTEGER NOT NULL, `album\_name` TEXT NOT NULL, `artist\_id` INTEGER NOT NULL, `artist\_name` TEXT NOT NULL, `composer` TEXT, `album\_artist` TEXT)");  
 db.execSQL("CREATE UNIQUE INDEX IF NOT EXISTS `index\_SongEntity\_playlist\_creator\_id\_id` ON `SongEntity` (`playlist\_creator\_id`, `id`)");  
 db.execSQL("CREATE TABLE IF NOT EXISTS `HistoryEntity` (`id` INTEGER NOT NULL, `title` TEXT NOT NULL, `track\_number` INTEGER NOT NULL, `year` INTEGER NOT NULL, `duration` INTEGER NOT NULL, `data` TEXT NOT NULL, `date\_modified` INTEGER NOT NULL, `album\_id` INTEGER NOT NULL, `album\_name` TEXT NOT NULL, `artist\_id` INTEGER NOT NULL, `artist\_name` TEXT NOT NULL, `composer` TEXT, `album\_artist` TEXT, `time\_played` INTEGER NOT NULL, PRIMARY KEY(`id`))");  
 db.execSQL("CREATE TABLE IF NOT EXISTS `PlayCountEntity` (`id` INTEGER NOT NULL, `title` TEXT NOT NULL, `track\_number` INTEGER NOT NULL, `year` INTEGER NOT NULL, `duration` INTEGER NOT NULL, `data` TEXT NOT NULL, `date\_modified` INTEGER NOT NULL, `album\_id` INTEGER NOT NULL, `album\_name` TEXT NOT NULL, `artist\_id` INTEGER NOT NULL, `artist\_name` TEXT NOT NULL, `composer` TEXT, `album\_artist` TEXT, `time\_played` INTEGER NOT NULL, `play\_count` INTEGER NOT NULL, PRIMARY KEY(`id`))");  
 db.execSQL("CREATE TABLE IF NOT EXISTS room\_master\_table (id INTEGER PRIMARY KEY,identity\_hash TEXT)");  
 db.execSQL("INSERT OR REPLACE INTO room\_master\_table (id,identity\_hash) VALUES(42, '477d6210eb83b418129be8b2c2acb691')");  
 }  
  
 @Override  
 public void dropAllTables(@NonNull final SupportSQLiteDatabase db) {  
 db.execSQL("DROP TABLE IF EXISTS `PlaylistEntity`");  
 db.execSQL("DROP TABLE IF EXISTS `SongEntity`");  
 db.execSQL("DROP TABLE IF EXISTS `HistoryEntity`");  
 db.execSQL("DROP TABLE IF EXISTS `PlayCountEntity`");  
 final List<? extends RoomDatabase.Callback> \_callbacks = mCallbacks;  
 if (\_callbacks != null) {  
 for (RoomDatabase.Callback \_callback : \_callbacks) {  
 \_callback.onDestructiveMigration(db);  
 }  
 }  
 }  
  
 @Override  
 public void onCreate(@NonNull final SupportSQLiteDatabase db) {  
 final List<? extends RoomDatabase.Callback> \_callbacks = mCallbacks;  
 if (\_callbacks != null) {  
 for (RoomDatabase.Callback \_callback : \_callbacks) {  
 \_callback.onCreate(db);  
 }  
 }  
 }  
  
 @Override  
 public void onOpen(@NonNull final SupportSQLiteDatabase db) {  
 mDatabase = db;  
 internalInitInvalidationTracker(db);  
 final List<? extends RoomDatabase.Callback> \_callbacks = mCallbacks;  
 if (\_callbacks != null) {  
 for (RoomDatabase.Callback \_callback : \_callbacks) {  
 \_callback.onOpen(db);  
 }  
 }  
 }  
  
 @Override  
 public void onPreMigrate(@NonNull final SupportSQLiteDatabase db) {  
 DBUtil.*dropFtsSyncTriggers*(db);  
 }  
  
 @Override  
 public void onPostMigrate(@NonNull final SupportSQLiteDatabase db) {  
 }  
  
 @Override  
 @NonNull  
 public RoomOpenHelper.ValidationResult onValidateSchema(  
 @NonNull final SupportSQLiteDatabase db) {  
 final HashMap<String, TableInfo.Column> \_columnsPlaylistEntity = new HashMap<String, TableInfo.Column>(2);  
 \_columnsPlaylistEntity.put("playlist\_id", new TableInfo.Column("playlist\_id", "INTEGER", true, 1, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsPlaylistEntity.put("playlist\_name", new TableInfo.Column("playlist\_name", "TEXT", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 final HashSet<TableInfo.ForeignKey> \_foreignKeysPlaylistEntity = new HashSet<TableInfo.ForeignKey>(0);  
 final HashSet<TableInfo.Index> \_indicesPlaylistEntity = new HashSet<TableInfo.Index>(0);  
 final TableInfo \_infoPlaylistEntity = new TableInfo("PlaylistEntity", \_columnsPlaylistEntity, \_foreignKeysPlaylistEntity, \_indicesPlaylistEntity);  
 final TableInfo \_existingPlaylistEntity = TableInfo.*read*(db, "PlaylistEntity");  
 if (!\_infoPlaylistEntity.equals(\_existingPlaylistEntity)) {  
 return new RoomOpenHelper.ValidationResult(false, "PlaylistEntity(code.name.monkey.retromusic.db.PlaylistEntity).\n"  
 + " Expected:\n" + \_infoPlaylistEntity + "\n"  
 + " Found:\n" + \_existingPlaylistEntity);  
 }  
 final HashMap<String, TableInfo.Column> \_columnsSongEntity = new HashMap<String, TableInfo.Column>(15);  
 \_columnsSongEntity.put("song\_key", new TableInfo.Column("song\_key", "INTEGER", true, 1, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsSongEntity.put("playlist\_creator\_id", new TableInfo.Column("playlist\_creator\_id", "INTEGER", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsSongEntity.put("id", new TableInfo.Column("id", "INTEGER", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsSongEntity.put("title", new TableInfo.Column("title", "TEXT", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsSongEntity.put("track\_number", new TableInfo.Column("track\_number", "INTEGER", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsSongEntity.put("year", new TableInfo.Column("year", "INTEGER", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsSongEntity.put("duration", new TableInfo.Column("duration", "INTEGER", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsSongEntity.put("data", new TableInfo.Column("data", "TEXT", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsSongEntity.put("date\_modified", new TableInfo.Column("date\_modified", "INTEGER", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsSongEntity.put("album\_id", new TableInfo.Column("album\_id", "INTEGER", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsSongEntity.put("album\_name", new TableInfo.Column("album\_name", "TEXT", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsSongEntity.put("artist\_id", new TableInfo.Column("artist\_id", "INTEGER", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsSongEntity.put("artist\_name", new TableInfo.Column("artist\_name", "TEXT", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsSongEntity.put("composer", new TableInfo.Column("composer", "TEXT", false, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsSongEntity.put("album\_artist", new TableInfo.Column("album\_artist", "TEXT", false, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 final HashSet<TableInfo.ForeignKey> \_foreignKeysSongEntity = new HashSet<TableInfo.ForeignKey>(0);  
 final HashSet<TableInfo.Index> \_indicesSongEntity = new HashSet<TableInfo.Index>(1);  
 \_indicesSongEntity.add(new TableInfo.Index("index\_SongEntity\_playlist\_creator\_id\_id", true, Arrays.*asList*("playlist\_creator\_id", "id"), Arrays.*asList*("ASC", "ASC")));  
 final TableInfo \_infoSongEntity = new TableInfo("SongEntity", \_columnsSongEntity, \_foreignKeysSongEntity, \_indicesSongEntity);  
 final TableInfo \_existingSongEntity = TableInfo.*read*(db, "SongEntity");  
 if (!\_infoSongEntity.equals(\_existingSongEntity)) {  
 return new RoomOpenHelper.ValidationResult(false, "SongEntity(code.name.monkey.retromusic.db.SongEntity).\n"  
 + " Expected:\n" + \_infoSongEntity + "\n"  
 + " Found:\n" + \_existingSongEntity);  
 }  
 final HashMap<String, TableInfo.Column> \_columnsHistoryEntity = new HashMap<String, TableInfo.Column>(14);  
 \_columnsHistoryEntity.put("id", new TableInfo.Column("id", "INTEGER", true, 1, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsHistoryEntity.put("title", new TableInfo.Column("title", "TEXT", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsHistoryEntity.put("track\_number", new TableInfo.Column("track\_number", "INTEGER", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsHistoryEntity.put("year", new TableInfo.Column("year", "INTEGER", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsHistoryEntity.put("duration", new TableInfo.Column("duration", "INTEGER", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsHistoryEntity.put("data", new TableInfo.Column("data", "TEXT", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsHistoryEntity.put("date\_modified", new TableInfo.Column("date\_modified", "INTEGER", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsHistoryEntity.put("album\_id", new TableInfo.Column("album\_id", "INTEGER", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsHistoryEntity.put("album\_name", new TableInfo.Column("album\_name", "TEXT", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsHistoryEntity.put("artist\_id", new TableInfo.Column("artist\_id", "INTEGER", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsHistoryEntity.put("artist\_name", new TableInfo.Column("artist\_name", "TEXT", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsHistoryEntity.put("composer", new TableInfo.Column("composer", "TEXT", false, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsHistoryEntity.put("album\_artist", new TableInfo.Column("album\_artist", "TEXT", false, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsHistoryEntity.put("time\_played", new TableInfo.Column("time\_played", "INTEGER", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 final HashSet<TableInfo.ForeignKey> \_foreignKeysHistoryEntity = new HashSet<TableInfo.ForeignKey>(0);  
 final HashSet<TableInfo.Index> \_indicesHistoryEntity = new HashSet<TableInfo.Index>(0);  
 final TableInfo \_infoHistoryEntity = new TableInfo("HistoryEntity", \_columnsHistoryEntity, \_foreignKeysHistoryEntity, \_indicesHistoryEntity);  
 final TableInfo \_existingHistoryEntity = TableInfo.*read*(db, "HistoryEntity");  
 if (!\_infoHistoryEntity.equals(\_existingHistoryEntity)) {  
 return new RoomOpenHelper.ValidationResult(false, "HistoryEntity(code.name.monkey.retromusic.db.HistoryEntity).\n"  
 + " Expected:\n" + \_infoHistoryEntity + "\n"  
 + " Found:\n" + \_existingHistoryEntity);  
 }  
 final HashMap<String, TableInfo.Column> \_columnsPlayCountEntity = new HashMap<String, TableInfo.Column>(15);  
 \_columnsPlayCountEntity.put("id", new TableInfo.Column("id", "INTEGER", true, 1, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsPlayCountEntity.put("title", new TableInfo.Column("title", "TEXT", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsPlayCountEntity.put("track\_number", new TableInfo.Column("track\_number", "INTEGER", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsPlayCountEntity.put("year", new TableInfo.Column("year", "INTEGER", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsPlayCountEntity.put("duration", new TableInfo.Column("duration", "INTEGER", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsPlayCountEntity.put("data", new TableInfo.Column("data", "TEXT", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsPlayCountEntity.put("date\_modified", new TableInfo.Column("date\_modified", "INTEGER", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsPlayCountEntity.put("album\_id", new TableInfo.Column("album\_id", "INTEGER", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsPlayCountEntity.put("album\_name", new TableInfo.Column("album\_name", "TEXT", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsPlayCountEntity.put("artist\_id", new TableInfo.Column("artist\_id", "INTEGER", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsPlayCountEntity.put("artist\_name", new TableInfo.Column("artist\_name", "TEXT", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsPlayCountEntity.put("composer", new TableInfo.Column("composer", "TEXT", false, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsPlayCountEntity.put("album\_artist", new TableInfo.Column("album\_artist", "TEXT", false, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsPlayCountEntity.put("time\_played", new TableInfo.Column("time\_played", "INTEGER", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 \_columnsPlayCountEntity.put("play\_count", new TableInfo.Column("play\_count", "INTEGER", true, 0, null, TableInfo.*CREATED\_FROM\_ENTITY*));  
 final HashSet<TableInfo.ForeignKey> \_foreignKeysPlayCountEntity = new HashSet<TableInfo.ForeignKey>(0);  
 final HashSet<TableInfo.Index> \_indicesPlayCountEntity = new HashSet<TableInfo.Index>(0);  
 final TableInfo \_infoPlayCountEntity = new TableInfo("PlayCountEntity", \_columnsPlayCountEntity, \_foreignKeysPlayCountEntity, \_indicesPlayCountEntity);  
 final TableInfo \_existingPlayCountEntity = TableInfo.*read*(db, "PlayCountEntity");  
 if (!\_infoPlayCountEntity.equals(\_existingPlayCountEntity)) {  
 return new RoomOpenHelper.ValidationResult(false, "PlayCountEntity(code.name.monkey.retromusic.db.PlayCountEntity).\n"  
 + " Expected:\n" + \_infoPlayCountEntity + "\n"  
 + " Found:\n" + \_existingPlayCountEntity);  
 }  
 return new RoomOpenHelper.ValidationResult(true, null);  
 }  
 }, "477d6210eb83b418129be8b2c2acb691", "d09ae233e4e5fe9af703a4480fa9aaf8");  
 final SupportSQLiteOpenHelper.Configuration \_sqliteConfig = SupportSQLiteOpenHelper.Configuration.*builder*(config.context).name(config.name).callback(\_openCallback).build();  
 final SupportSQLiteOpenHelper \_helper = config.sqliteOpenHelperFactory.create(\_sqliteConfig);  
 return \_helper;  
 }  
  
 @Override  
 @NonNull  
 protected InvalidationTracker createInvalidationTracker() {  
 final HashMap<String, String> \_shadowTablesMap = new HashMap<String, String>(0);  
 final HashMap<String, Set<String>> \_viewTables = new HashMap<String, Set<String>>(0);  
 return new InvalidationTracker(this, \_shadowTablesMap, \_viewTables, "PlaylistEntity","SongEntity","HistoryEntity","PlayCountEntity");  
 }  
  
 @Override  
 public void clearAllTables() {  
 super.assertNotMainThread();  
 final SupportSQLiteDatabase \_db = super.getOpenHelper().getWritableDatabase();  
 try {  
 super.beginTransaction();  
 \_db.execSQL("DELETE FROM `PlaylistEntity`");  
 \_db.execSQL("DELETE FROM `SongEntity`");  
 \_db.execSQL("DELETE FROM `HistoryEntity`");  
 \_db.execSQL("DELETE FROM `PlayCountEntity`");  
 super.setTransactionSuccessful();  
 } finally {  
 super.endTransaction();  
 \_db.query("PRAGMA wal\_checkpoint(FULL)").close();  
 if (!\_db.inTransaction()) {  
 \_db.execSQL("VACUUM");  
 }  
 }  
 }  
  
 @Override  
 @NonNull  
 protected Map<Class<?>, List<Class<?>>> getRequiredTypeConverters() {  
 final HashMap<Class<?>, List<Class<?>>> \_typeConvertersMap = new HashMap<Class<?>, List<Class<?>>>();  
 \_typeConvertersMap.put(PlaylistDao.class, PlaylistDao\_Impl.*getRequiredConverters*());  
 \_typeConvertersMap.put(PlayCountDao.class, PlayCountDao\_Impl.*getRequiredConverters*());  
 \_typeConvertersMap.put(HistoryDao.class, HistoryDao\_Impl.*getRequiredConverters*());  
 return \_typeConvertersMap;  
 }  
  
 @Override  
 @NonNull  
 public Set<Class<? extends AutoMigrationSpec>> getRequiredAutoMigrationSpecs() {  
 final HashSet<Class<? extends AutoMigrationSpec>> \_autoMigrationSpecsSet = new HashSet<Class<? extends AutoMigrationSpec>>();  
 return \_autoMigrationSpecsSet;  
 }  
  
 @Override  
 @NonNull  
 public List<Migration> getAutoMigrations(  
 @NonNull final Map<Class<? extends AutoMigrationSpec>, AutoMigrationSpec> autoMigrationSpecs) {  
 final List<Migration> \_autoMigrations = new ArrayList<Migration>();  
 return \_autoMigrations;  
 }  
  
 @Override  
 public PlaylistDao playlistDao() {  
 if (\_playlistDao != null) {  
 return \_playlistDao;  
 } else {  
 synchronized(this) {  
 if(\_playlistDao == null) {  
 \_playlistDao = new PlaylistDao\_Impl(this);  
 }  
 return \_playlistDao;  
 }  
 }  
 }  
  
 @Override  
 public PlayCountDao playCountDao() {  
 if (\_playCountDao != null) {  
 return \_playCountDao;  
 } else {  
 synchronized(this) {  
 if(\_playCountDao == null) {  
 \_playCountDao = new PlayCountDao\_Impl(this);  
 }  
 return \_playCountDao;  
 }  
 }  
 }  
  
 @Override  
 public HistoryDao historyDao() {  
 if (\_historyDao != null) {  
 return \_historyDao;  
 } else {  
 synchronized(this) {  
 if(\_historyDao == null) {  
 \_historyDao = new HistoryDao\_Impl(this);  
 }  
 return \_historyDao;  
 }  
 }  
 }  
}



# Chapter - 6 LIMITATIONS AND FUTURE ENHANCEMENTS

## 6.1 Limitations

While a retro music player offers a nostalgic and aesthetically appealing experience, it also comes with certain limitations. These constraints can arise due to technological advancements, user expectations, and compatibility issues. Below are some key limitations of retro music players, particularly in the context of an Android application.

1. Limited Streaming Capabilities

One of the major limitations of a retro music player is the lack of streaming support. Most retro music apps focus on local music playback, meaning users must manually upload or store songs on their device. In contrast, modern music apps like Spotify and Apple Music offer cloud-based streaming, making it easier for users to access vast libraries of songs without consuming local storage.

2. Outdated File Format Support

Many retro music players are designed to play older audio formats like MP3, WAV, and AAC, but they may not support modern high-resolution formats like FLAC, ALAC, or Opus. Additionally, emerging audio formats designed for efficient compression and higher fidelity may not be compatible with a retro-style player.

3. Lack of Smart Features

Modern music applications integrate AI-powered features such as personalized playlists, song recommendations, and voice assistants (e.g., Google Assistant and Alexa). Retro music players, which primarily focus on a classic experience, may not include such advanced features, limiting user engagement and customization.

4. Storage Dependency

Since retro music players rely heavily on local storage, users must download and manually manage their music libraries. This approach can be inconvenient compared to cloud-based solutions that allow users to access their music collection from multiple devices without worrying about storage limitations.

5. Outdated UI/UX Design

A retro music player typically emulates the look and feel of old-school music players, such as cassette decks, CD players, or vintage MP3 interfaces. While this can be appealing for nostalgia, it may not align with modern UI/UX standards, potentially leading to a less intuitive or less efficient user experience.

6. Limited Social Features

Most modern music apps allow users to share playlists, sync across devices, and integrate with social media platforms. However, a retro music player, designed for a classic and offline experience, may lack these sharing capabilities, making it less appealing for users who enjoy social music interactions.

7. No Online Lyrics or Metadata Fetching

Many contemporary music apps provide real-time lyrics, automatic metadata tagging, and album art retrieval from online databases. In contrast, a retro music player may require users to manually add song details and album covers, which can be time-consuming and frustrating.

## 6.2 Future Enhancements

While a retro music player brings nostalgia and simplicity, integrating modern features can improve its functionality without compromising its vintage appeal. Below are some future enhancements that can enhance user experience while maintaining the essence of a classic music player.

1. Cloud and Streaming Integration

Although retro music players traditionally focus on local playback, adding optional streaming integration for services like Spotify, YouTube Music, or SoundCloud can make the app more versatile. Users could switch between local music and online streaming while keeping the retro aesthetic intact.

2. AI-Powered Smart Playlists

A major drawback of classic music players is the lack of personalized recommendations. Implementing AI-based smart playlists that analyze listening habits and suggest songs can modernize the experience while maintaining a retro-styled interface.

3. Improved File Format Support

To attract audiophiles, future versions of the app could support high-resolution audio formats such as FLAC, ALAC, and DSD. This would enhance sound quality and cater to users who prefer lossless music playback.

4. Lyrics and Metadata Fetching

Manually tagging songs can be tedious. The app could automatically fetch lyrics, album art, and metadata from sources like MusicBrainz, Genius, or Last.fm to ensure a richer experience.

5. Enhanced UI with Customizable Themes

A major appeal of a retro music player is its visual design. Future updates could introduce:

* Customizable skins/themes (e.g., Cassette Mode, Vinyl Mode, MP3 Player Mode).
* Animated visualizations that replicate old-school equalizers.
* Dynamic color schemes based on album art.

6. Smart Home and Wearable Device Integration

To increase usability, the app could support:

* Google Assistant / Alexa integration for voice commands.
* Wear OS compatibility to control music from smartwatches.
* Chromecast & Bluetooth speaker support for a seamless audio experience.

7. Social and Sharing Features

Modern users like sharing their music preferences. The app could integrate:

* Playlist sharing with friends.
* Social media integration (e.g., Instagram stories, Facebook, Twitter).
* Collaboration mode, where multiple users can contribute to a playlist.

8. Sleep Timer and Focus Modes

Adding a sleep timer for automatic shutdown or a focus mode with ambient sounds can make the app more appealing for relaxation and productivity.

9. Battery Optimization & Offline Mode Enhancements

To ensure smooth performance, developers can:

* Optimize battery consumption by reducing background activity.
* Improve offline features, such as local backup of playlists and settings.

# Chapter - 7 CONCLUSION

## 7.1 Conclusion

The Retro Music App successfully brings a nostalgic yet modernized music experience to Android users. Throughout the development process, we integrated core functionalities such as media playback, playlist management, equalizer controls, and offline support, ensuring a smooth and immersive user experience.

By leveraging Android’s Media Player API, Jetpack components, and Material Design principles, the app delivers both performance and aesthetics. Additionally, background playback and notification controls enhance usability, while efficient data handling ensures optimized performance across various devices.

Future improvements could include cloud sync, smart recommendations, and enhanced customization features, further enriching user engagement. The project serves as a strong foundation for building advanced music applications, combining retro aesthetics with modern development practices.

This documentation provides a comprehensive guide to the architecture, features, and implementation of the Retro Music App, assisting developers in further enhancements and maintenance.

# Chapter - 8 BIBLIOGRAPHY

## 8.1 Bibliography

Books

1. Android Programming: The Big Nerd Ranch Guide – Bill Phillips, Chris Stewart, and Kristin Marsicano
   * A hands-on guide to Android development with best practices.
2. Kotlin for Android Developers – Antonio Leiva
   * Essential for learning Kotlin, the recommended language for Android apps.
3. Android App Development for Dummies – Michael Burton
   * A beginner-friendly resource for Android development.
4. Pro Android Media: Developing Graphics, Music, Video, and Rich Media Apps for Smartphones and Tablets – Shawn Van Every
   * Focuses on handling media in Android apps.

* Developing a retro music app on the Android platform requires a strong foundation in Android app development, multimedia handling, UI design, and database management. A well-rounded understanding of these topics can be built by referencing books, online documentation, and various libraries that streamline development. This bibliography provides a collection of essential resources that will aid in the successful development of a retro music player for Android.
* One of the most comprehensive guides to Android development is *Android Programming: The Big Nerd Ranch Guide* by Bill Phillips, Chris Stewart, and Kristin Marsicano. This book offers a hands-on approach to learning Android development and introduces best practices for building applications.
* Apart from books, official documentation serves as the primary reference for Android development. The Android Developers Guide is the most authoritative resource, offering documentation on Android architecture, APIs, provides an advanced, customizable solution for streaming and local media playback.
* In addition to official documentation, online tutorials and community resources are valuable for staying updated with the latest developments. The Android Developers YouTube Channel offers video tutorials from Google engineers, providing insights into modern Android development. The RayWenderlich Android Tutorials (https://www.raywenderlich.com/android) are another excellent source of structured learning materials. Furthermore, the Medium Android Community (https://medium.com/androiddevelopers) contains articles from experienced developers discussing best practices, new features, and debugging techniques. For troubleshooting and peer support, Stack Overflow remains an invaluable resource where developers can find answers to common challenges in Android development..

Online Tutorials & Blogs

1. Android Developers YouTube Channel – https://www.youtube.com
   * Official Android development videos.
2. RayWenderlich Android Tutorials – https://www.raywenderlich.com/android
   * High-quality coding tutorials for Android.
3. Medium Android Community – https://medium.com/androiddevelopers
   * Articles from Google developers and experienced Android devs.
4. Stack Overflow – https://stackoverflow.com
   * A great place to find answers to Android development questions.

# 

# Chapter - 9 REFERENCES

## 9.1 References

* I am thankful to my guide **Dr. Jyotsna Salet** for guiding me. I am also thankful to whole staff of the Computer Department for giving me huge support for my project.

**Sites:**

* [www.google.com](http://www.google.com)
* [www.youtube.com](http://www.youtube.com)
* [www.stackoverflow.com](http://www.stackoverflow.com)
* [Download Android Studio & App Tools - Android Developers](https://developer.android.com/studio)

Thank You