ANDROID APPLICATION FOR PODCAST PLAYER: AUDIOPODS

TEAM MEMBERS:

SYED JAWAAD LATEEF - 20BCT0034
SUDHANSHU CHAUHAN - 20BCI0035
SAMRUDDHI DHAVALE - 20BCT0074

INDEX:

Project Report Titles

1. INTRODUCTION

1.1.Overview

This report presents a comprehensive analysis and summary of an Android application developed using Jetpack Compose. The application serves as a feature-rich podcast player, offering a wide range of functionalities such as sign-up and sign-in pages, a home page with podcast grids, and seamless podcast playback. Authentication is implemented using RoomDB, ensuring secure storage and management of user data, while the integration with the Listen Notes API enables the retrieval and playback of podcast episodes.

The primary goal of this project is to create an immersive and user-friendly podcast listening experience for Android users. By leveraging the power of Jetpack Compose, the application provides a visually appealing and responsive user interface. Users can register and log in securely, explore a diverse collection of podcasts, and enjoy their favorite episodes seamlessly within the application.

1.2.Purpose

The purpose of this project is to address the need for a modern and efficient podcast player on the Android platform. Existing approaches often relied on traditional UI frameworks that were less flexible and required extensive coding efforts. By utilizing Jetpack Compose, a modern UI toolkit, this project aims to overcome these limitations and provide a cutting-edge podcast application.

The application's core purpose is to offer a personalized and convenient podcast listening experience to users. Through the sign-up and sign-in pages, users can create and manage their accounts securely. The home page, with its visually appealing podcast grids, allows users to easily discover and explore a wide variety of podcasts. By seamlessly integrating with the Listen Notes API, users can select and play their preferred episodes with ease.

Overall, this project aims to provide an intuitive and immersive podcast player that enhances the way users consume audio content on the Android platform.

2. LITERATURE SURVEY

2.1.Existing problem

In the realm of Android podcast applications, there have been several challenges and limitations that developers have encountered. Traditional UI frameworks often posed difficulties in terms of designing visually appealing and responsive interfaces. These frameworks required extensive coding efforts, leading to complex and time-consuming development processes. Furthermore, managing user authentication and securely storing user data presented additional challenges, especially in scenarios where multiple users were involved.By combining Jetpack Compose, RoomDB, and the Listen Notes API, this project aims to address the existing challenges in Android podcast applications. The proposed solution offers a modern and efficient approach to UI development, secure authentication and data management, and a vast selection of podcasts for users to explore and enjoy.

2.2.Proposed solution

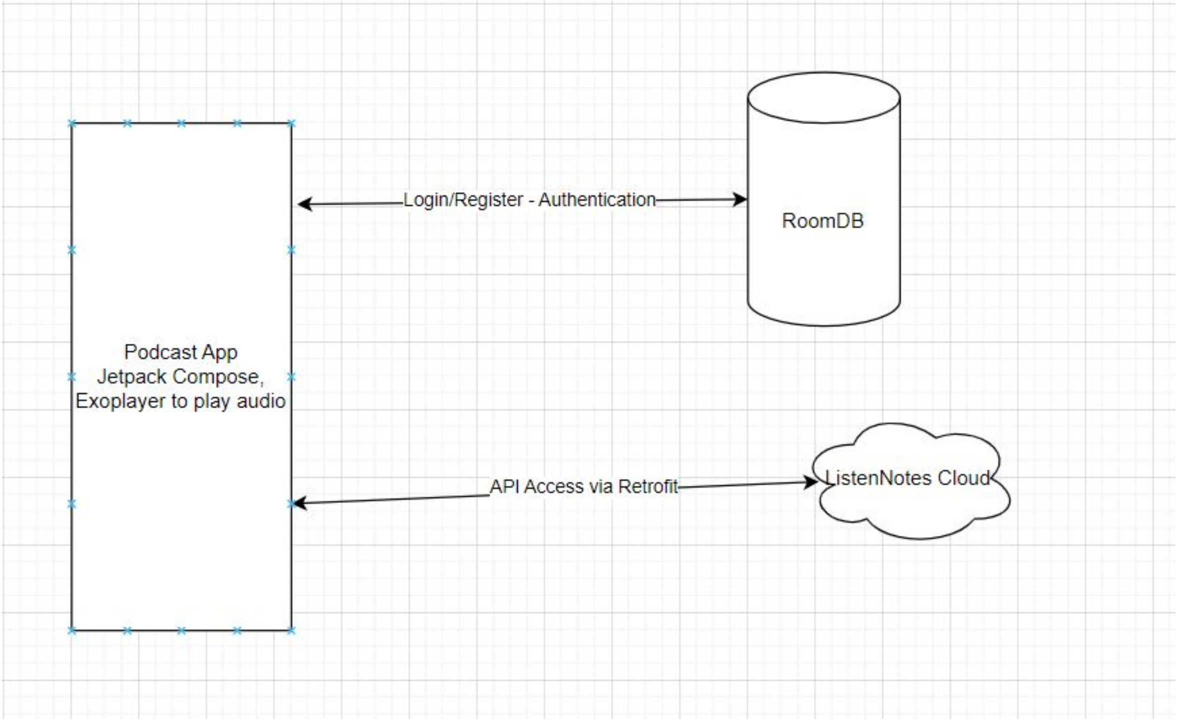
To address the limitations of existing approaches, this project proposes the use of Jetpack Compose, a modern UI toolkit, as a solution. Jetpack Compose leverages a declarative and reactive programming model, enabling developers to create visually stunning and responsive user interfaces with ease. By utilising Compose's built-in components and powerful layout system, the application's UI can be efficiently designed and developed.

Moreover, to ensure secure authentication and data storage, RoomDB, an Android library built on top of SQLite, is employed. RoomDB offers an abstraction layer that simplifies database operations and provides robust data management capabilities. This enables the application to securely store user information, manage user authentication, and retrieve relevant data when needed.

Overall, this project's proposed solution provides a significant improvement over existing approaches, offering a visually appealing and responsive UI, secure authentication and data storage, and seamless podcast playback capabilities Hardware requirements: Android device or emulator.

3. THEORETICAL ANALYSIS

3.1.Block diagram



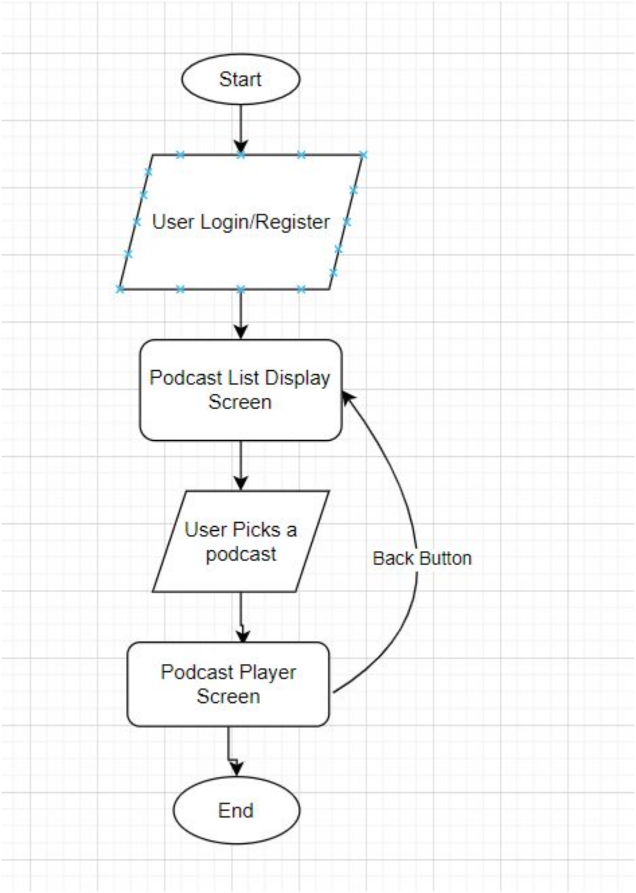
3.2.Hardware / Software designing

- Hardware requirements: Android device or emulator
- Software requirements: Android Studio, Jetpack Compose, RoomDB, Listen Notes API, RetroFit (API Access), Coil (Image display), ExoPlayer (Media player)

4. EXPERIMENTAL INVESTIGATIONS

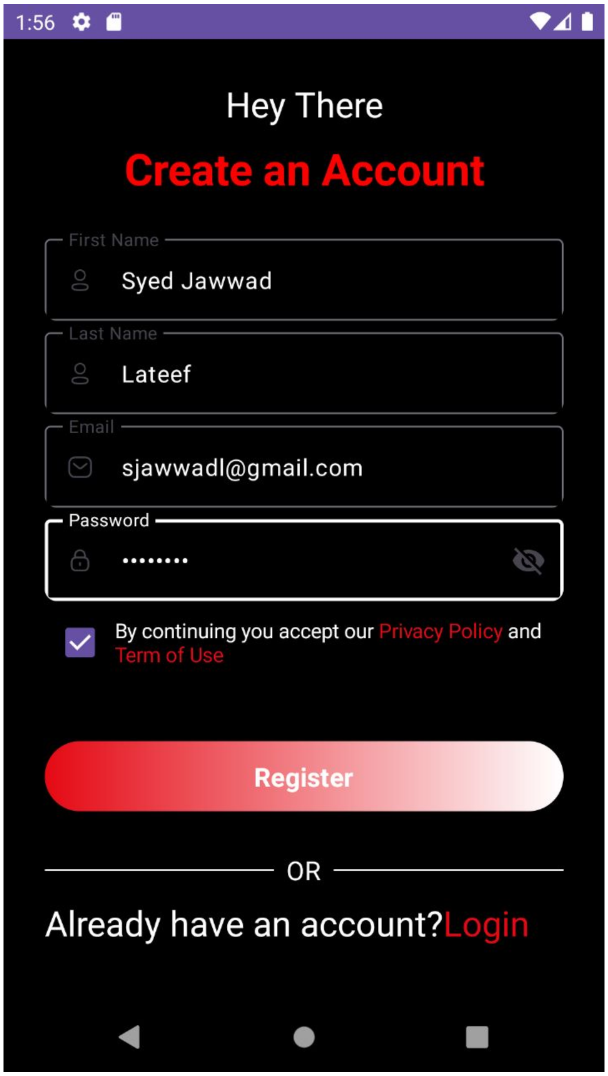
During the development of this project, several experiments and investigations were conducted to ensure seamless integration of the various components. This included testing the sign-up and sign-in functionalities, verifying the retrieval of podcast data from the Listen Notes API, and testing the playback of podcast episodes within the application.

5. FLOWCHART



6. RESULT

The final output of the project is an Android podcast application that incorporates sign-up and sign-in pages, a home page with podcast grids, and a playback feature. Users can successfully create accounts, log in securely, browse and select podcasts from the grid, and play their preferred episodes. The application provides a user-friendly interface and seamless playback functionality, enhancing the overall podcast listening experience.



Hey There

Create an Account

First Name: Syed Jawwad

Last Name: Lateef

Email: sjawwadl@gmail.com

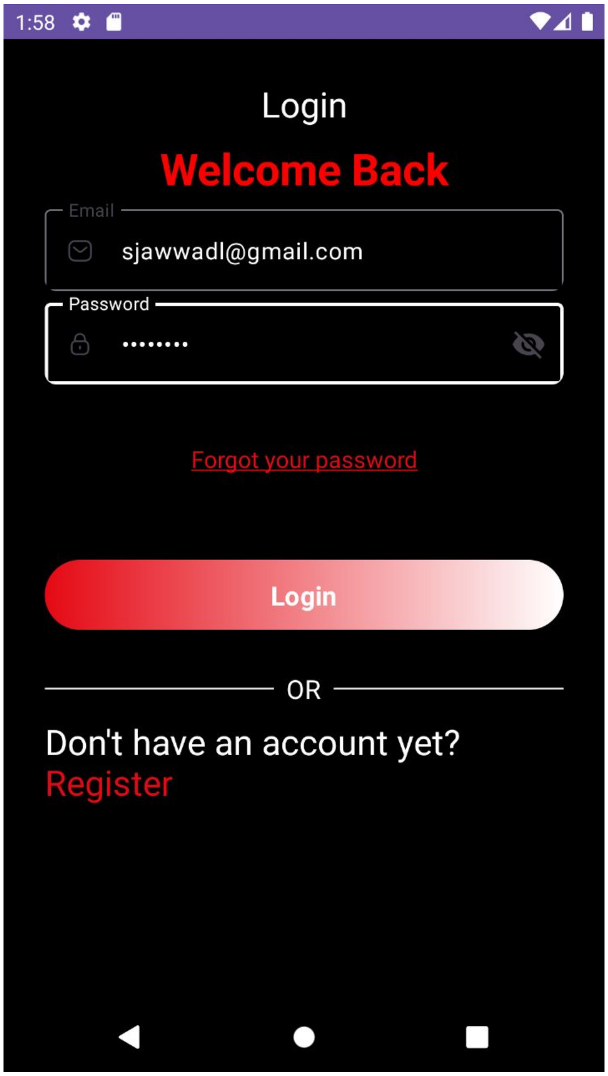
Password: ••••••••

☑ By continuing you accept our Privacy Policy and Term of Use

OR

Already have an account?Login

Registration Page



Login

Welcome Back

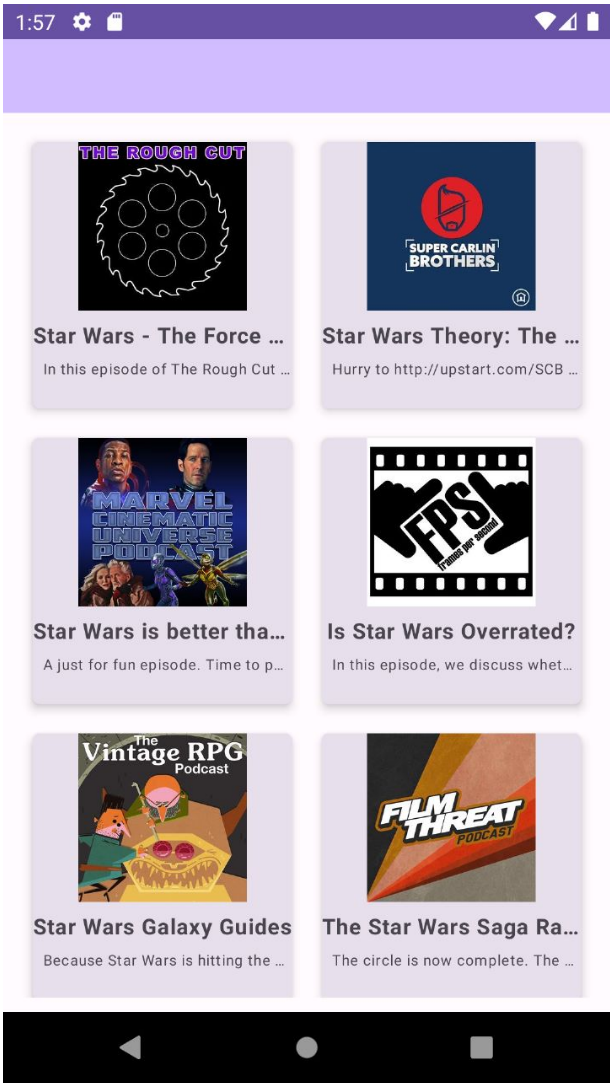
Email: sjawwadl@gmail.com

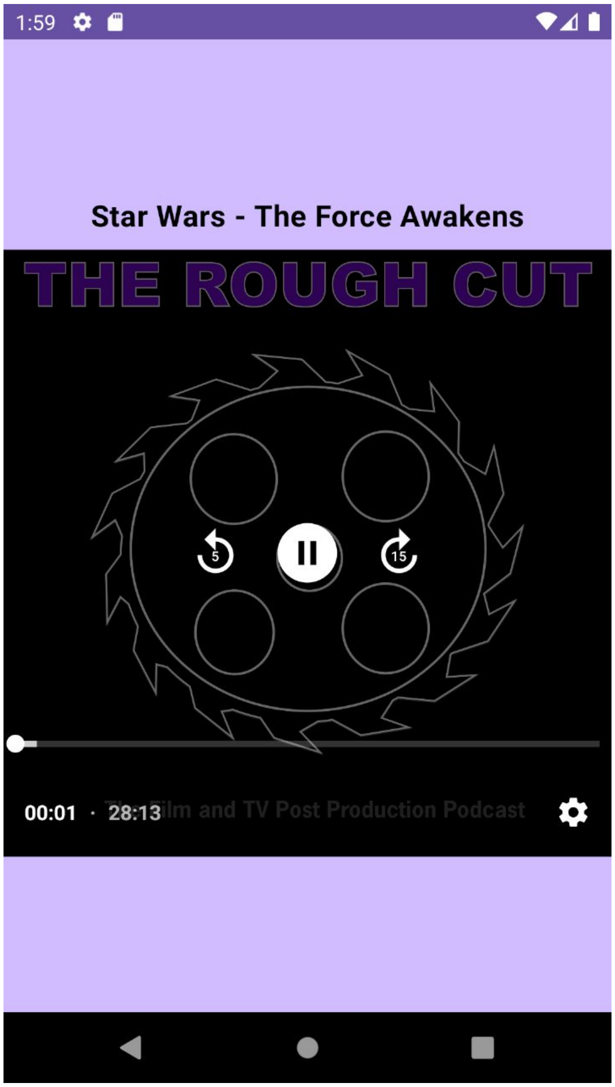
Password: ••••••••

OR

Don't have an account yet? Register

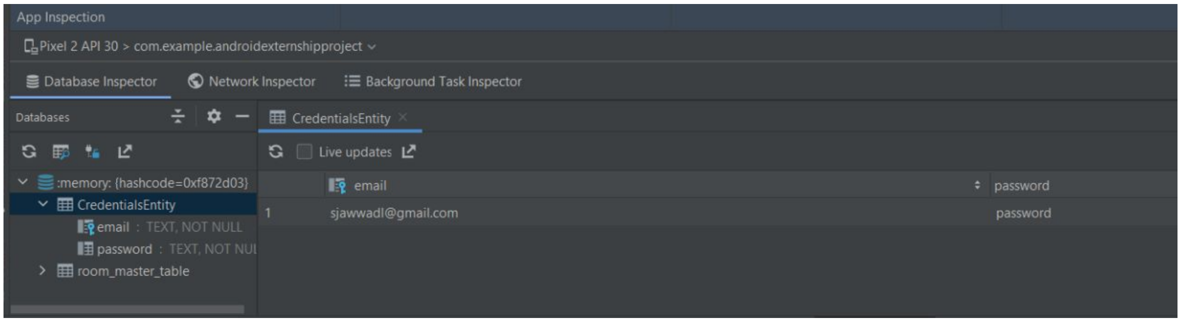
Login Page


1:57
THE ROUGH CUT
Star Wars - The Force ...
In this episode of The Rough Cut ...
SUPER CARLIN BROTHERS
Star Wars Theory: The ...
Hurry to http://upstart.com/SCB ...
MARVEL CINEMATIC UNIVERSE PODCAST
Star Wars is better tha...
A just for fun episode. Time to p...
FPS frames per second
Is Star Wars Overrated?
In this episode, we discuss whet...
The Vintage RPG Podcast
Star Wars Galaxy Guides
Because Star Wars is hitting the ...
FILM THREAT PODCAST
The Star Wars Saga Ra...
The circle is now complete. The ...


1:59
Star Wars - The Force Awakens
THE ROUGH CUT
5
15
00:01 · 28:13
The Film and TV Post Production Podcast

Home Page

Podcast Player



App Inspection

Pixel 2 API 30 > com.example.androidexternshipproject

Database Inspector · Network Inspector · Background Task Inspector

Databases

- :memory: {hashcode=0xf872d03}
  - CredentialsEntity
    - email : TEXT, NOT NULL
    - password : TEXT, NOT NUL
  - room_master_table

CredentialsEntity

□ Live updates

	email	password
1	sjawwadl@gmail.com	password

Credentials of user for authentication using RoomDB

7. ADVANTAGES & DISADVANTAGES

Advantages:

- Modern and responsive user interface using Jetpack Compose.
- Secure authentication and data storage with RoomDB.
- Wide range of podcast options available through the Listen Notes API.
- Intuitive and convenient playback functionality

Disadvantages:

- Dependency on external APIs for podcast data and playback.
- Limited customization options for the playback interface

8. APPLICATIONS

The proposed solution can be applied in various scenarios, including:

Personal podcast listening: Users can enjoy their favorite podcasts conveniently and personalize their listening experience.

Podcast platform integration: The application can serve as a starting point for integrating with other podcast platforms and services, offering a comprehensive podcast listening solution.

9. CONCLUSION

In conclusion, this project successfully developed an Android podcast application using Jetpack Compose, RoomDB, and the Listen Notes API. The application provides a user-friendly interface for sign-up, sign-in, browsing podcasts, and playback functionality. The integration of modern UI tools, secure authentication, and seamless podcast playback enhances the overall podcast listening experience

10. FUTURE SCOPE

Future enhancements for this project could include:

- Podcast recommendation algorithms based on user preferences and listening history.
- Social sharing options to allow users to share their favorite podcasts or episodes with others.
- Integration with additional podcast platforms or services to expand the range of available podcasts.
- Playlist creation and management functionality to offer personalized listening experiences.

11. BIBILOGRAPHY

REFERENCES:

- Android Developers. (n.d.). Jetpack Compose documentation. Retrieved from https://developer.android.com/jetpack/compose
- Android Developers. (n.d.). RoomDB documentation. Retrieved from https://developer.android.com/training/data-storage/room
- Listen Notes API. (n.d.). Official documentation. Retrieved from https://www.listennotes.com/api/
- Smith, J. (2021). Building Android Apps with Jetpack Compose. O'Reilly Media.
- Flutter vs. Jetpack Compose: Which is the Best Choice? (2021). Retrieved from https://www.spaceotechnologies.com/flutter-vs-jetpack-compose-best-choice/
- Mukherjee, S. (2020). Exploring Jetpack Compose for Android. Packt Publishing.
- Android Hive. (2021). Android Jetpack Compose Tutorial: Build a Simple UI with Compose. Retrieved from https://www.androidhive.info/jetpack/android-jetpack-compose-tutorial/
- RoomDB: A New Era of Database on Android. (2021). Retrieved from

- https://developer.android.com/training/data-storage/room
- Vogel, L. (2021). Building an Android App with Jetpack Compose. RayWenderlich.com. Retrieved from https://www.raywenderlich.com/195-android-jetpack-compose-tutorial-building-an-android-app
- Listen Notes. (2023). Listen Notes API: Podcast Search & Directory API. Retrieved from https://www.listennotes.com/api/

APPENDIX

A. Source Code

MainActivity.kt

```
package com.example.androidexternshipproject


import android.os.Bundle
import android.text.Html
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.viewModels
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.ui.Modifier
import androidx.room.Room
import com.example.androidexternshipproject.apiInteractions.APIViewModel
import com.example.androidexternshipproject.app.PodcastApp

import com.example.androidexternshipproject.dbInteractions.RoomDb

import com.example.androidexternshipproject.dbInteractions.RoomDbDao

import com.example.androidexternshipproject.ui.theme.AndroidExternshipProjectTheme


class MainActivity : ComponentActivity() {

    val apiViewModel by viewModels<APIViewModel>()


    override fun onCreate(savedInstanceState: Bundle?) {

        super.onCreate(savedInstanceState)

        setContent {

            AndroidExternshipProjectTheme {

                // A surface container using the 'background' color from the theme

                Surface(

                    modifier = Modifier.fillMaxSize(),

                    color = MaterialTheme.colorScheme.background

                ) {

                    apiViewModel.search("")

                    lateinit var myDao:RoomDbDao

                    val db: RoomDb =

                        Room.inMemoryDatabaseBuilder(applicationContext, RoomDb::class.java).build()
                    myDao = db.roomDbDao()


                    PodcastApp(myDao, apiViewModel.searchResponse)
//for login/register


                }

            }

        }

    }

}

fun stripHtml(html: String?): String {

    return Html.fromHtml(html,
Html.FROM_HTML_MODE_LEGACY).toString()

}
```

For entire code the link of GitHub is added below:

https://github.com/EssJayEl/AndroidExternshipProject