

MAP711S Group Assignment Assignment

Group Members

Keenan Husselmann (214076784) - Lead Developer

Lance Cloete (223000132) - Team Lead

Edmund Jansen (223118796) - Frontend Developer

Sander Santana (223034738) - Backend Developer

Table of Contents

[Project Description 1](#_Toc198500978)

[1. Introduction 2](#_Toc198500979)

[2. Client Requirements and MoSCoW Prioritization 2](#_Toc198500980)

[3. Development Stack and Tools 2](#_Toc198500981)

[4. Architecture and Software Engineering Principles 3](#_Toc198500982)

[5. App Navigation System 3](#_Toc198500983)

[6. Testing Techniques 3](#_Toc198500984)

[7. Challenges Faced and Resolutions 3](#_Toc198500985)

[8. Reflection and Conclusion 3](#_Toc198500986)

[9. UI Pages and Screens 4](#_Toc198500987)

[10. Kotlin Code Screenshots 9](#_Toc198500988)

[11. GitHub Repository 15](#_Toc198500989)

[12. Project Management 15](#_Toc198500990)

[13. Log Sheet 16](#_Toc198500991)

[14. Timeline Analysis 16](#_Toc198500992)

## Project Description

In this task, students are expected to develop a mobile application for the Namibia

Hockey Union (https://namibiahockey.org) in groups of four(4) or five (5) students.

The minimum functional requirements of the application are, but not limited to, the

following:

• team registration;

• event entries;

• player registration & management;

• real-time information sharing.

## 1. Introduction

The Namibia Hockey Union (NHU) requires a mobile application to support and streamline various administrative and event-related functions for its stakeholders, including players, teams, and officials. Our team was tasked with designing and building this mobile application to meet the Union’s needs through modern software engineering practices. Our core goal is to create a user-friendly, real-time application that facilitated team and player registrations. This project simulated a real-world development experience, from gathering client requirements to deployment and testing.

## 2. Client Requirements and MoSCoW Prioritization

We held several brainstorming sessions to gather and interpret the client’s vision and functional expectations. To ensure focus and feasibility within the project timeline, we adopted the MoSCoW prioritization method:

* Must Have:
* User Authentication (Sign-Up, Login, Logout)
* Team Registration and Management
* Player Registration
* Real-Time Information Sharing
* Should Have:
* Event Entry and Participation Tracking
* Role-Based Access Control
* Could Have:
* Push Notifications for announcements
* In-app calendar and event reminders
* Historical data (past match results, player stats)
* Won’t Have (for this release):
* Social Media Integration
* E-Commerce functionality (e.g., ticket purchases or merchandise)

## 3. Development Stack and Tools

We made use of the following tools and technologies:  
- Android Studio: Our primary IDE.  
- Kotlin & Jetpack Compose: For modern, declarative UI.  
- Firebase Authentication: For managing user sessions.  
- Firebase Firestore: For real-time data synchronization and data storage.  
- Real Device Debugging: For testing the app on physical Android devices.

## 4. Architecture and Software Engineering Principles

We adopted MVVM (Model-View-ViewModel) architecture and clean code principles:  
- View: Composables for UI.  
- ViewModel: Manages logic and data interaction.  
- Model: Data structures (e.g., Player, Team).  
- FirebaseService: Abstraction of database operations.

## 5. App Navigation System

Navigation is handled using Jetpack Navigation Compose. Screens like 'auth', 'login', 'signup', etc., are registered with routes, and parameters were passed as arguments where needed.

## 6. Testing Techniques

To ensure app reliability we implemented:  
- Manual Testing: On real devices.  
- Unit Testing: With JUnit for ViewModel and logic.  
- Peer Reviews: For better code quality and fewer bugs.

## 7. Challenges Faced and Resolutions

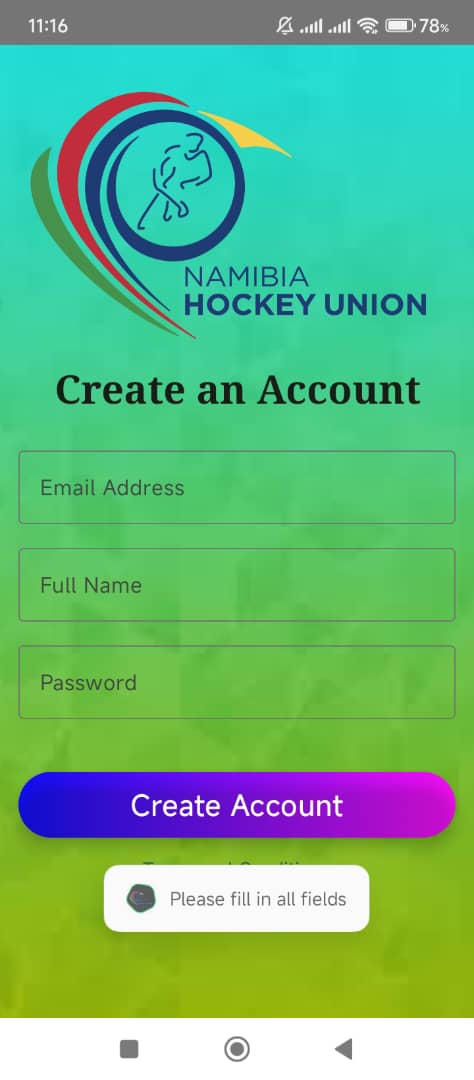
Some of the key challenges were:  
- Implementing Firebase security rules.  
- Designing adaptive UIs for multiple devices.

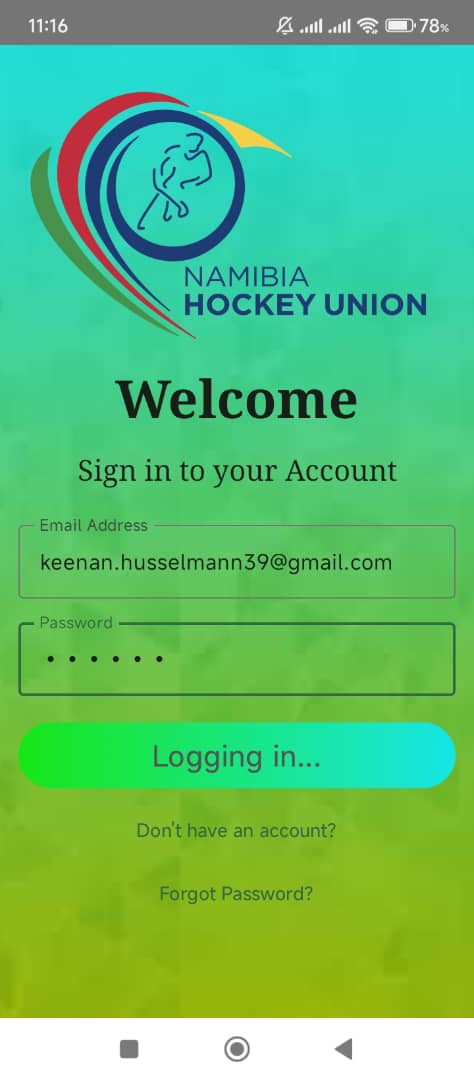
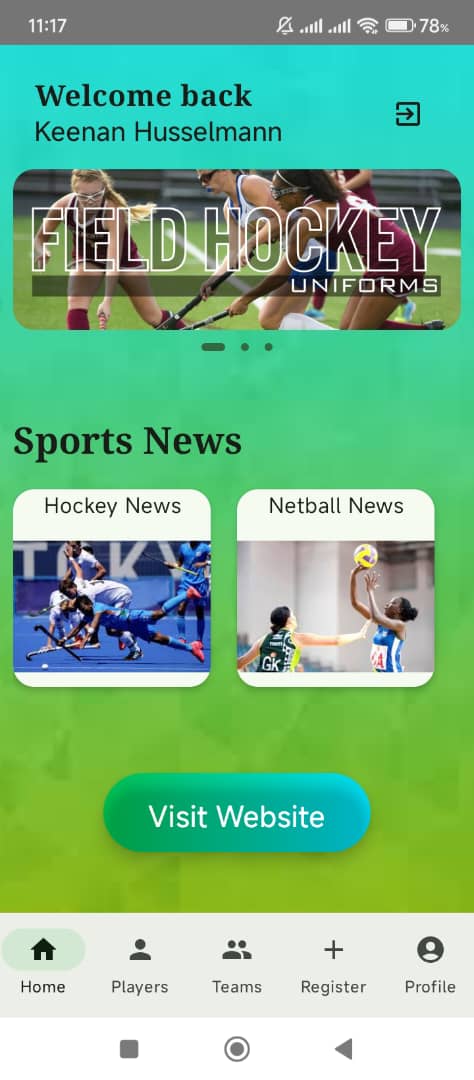
-Limited resources, slow PC’s, little time

## 8. Reflection and Conclusion

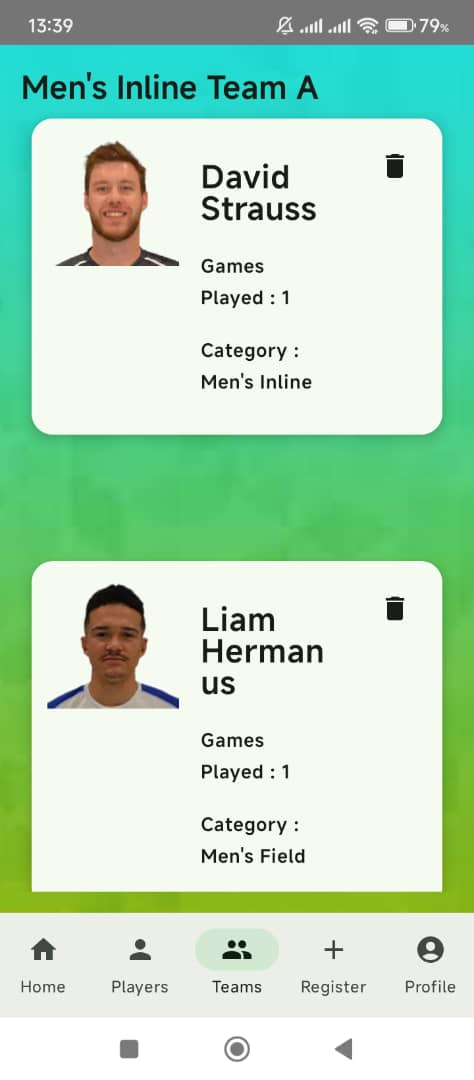
This project helped us enhance our skills in Jetpack Compose, Firebase, and collaborative development. We delivered an MVP that met core NHU requirements and laid a foundation for future improvements such as notifications, analytics, and advanced admin tools.

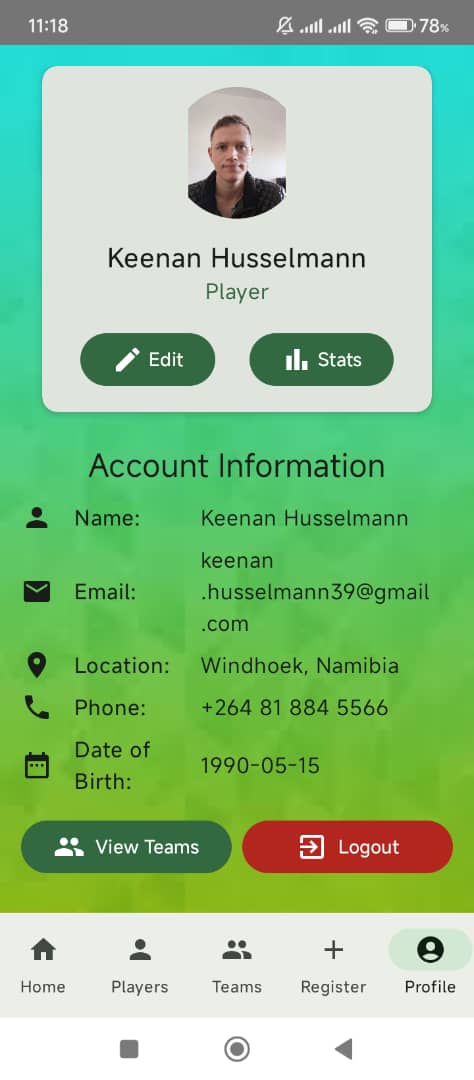
## 9. UI Pages and Screens

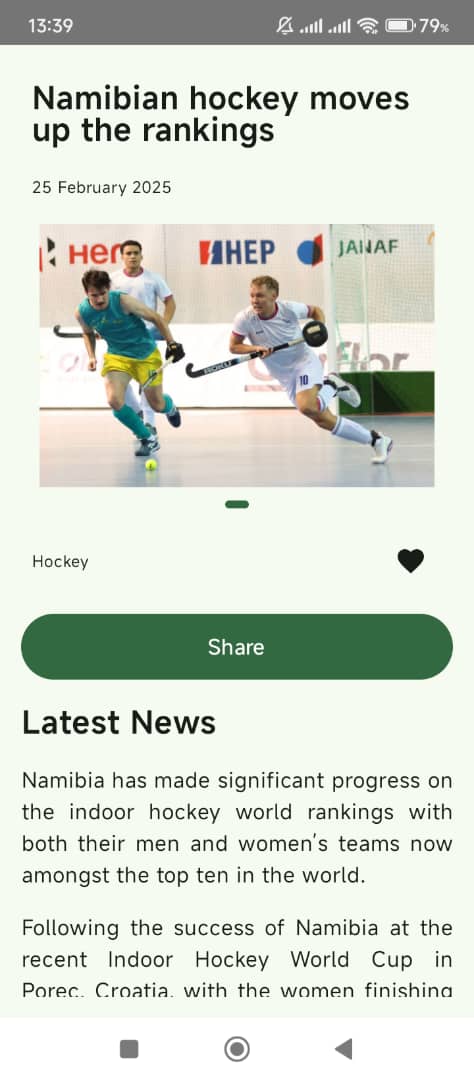
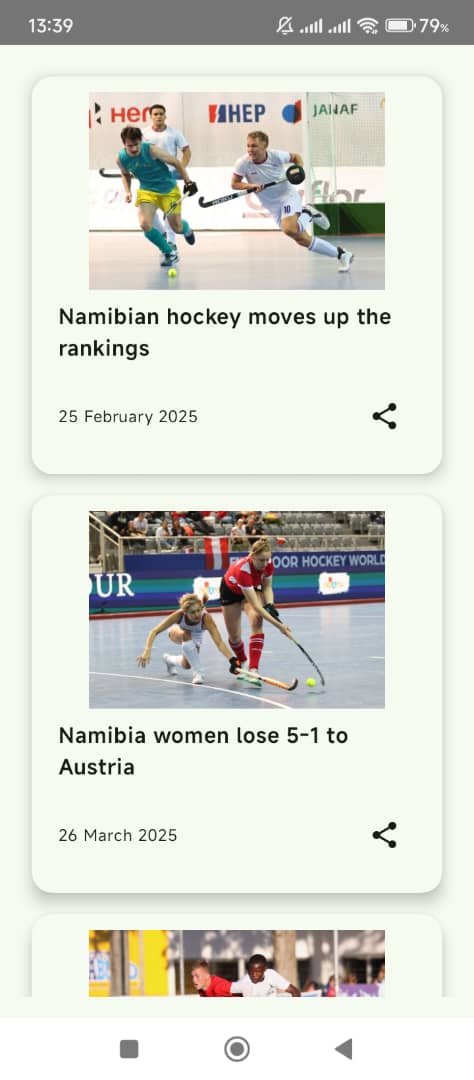
*Image Description: Landing Page Image Description: Sign Up Page*

*Image Description: Login Page* *Image Description: Home Page*

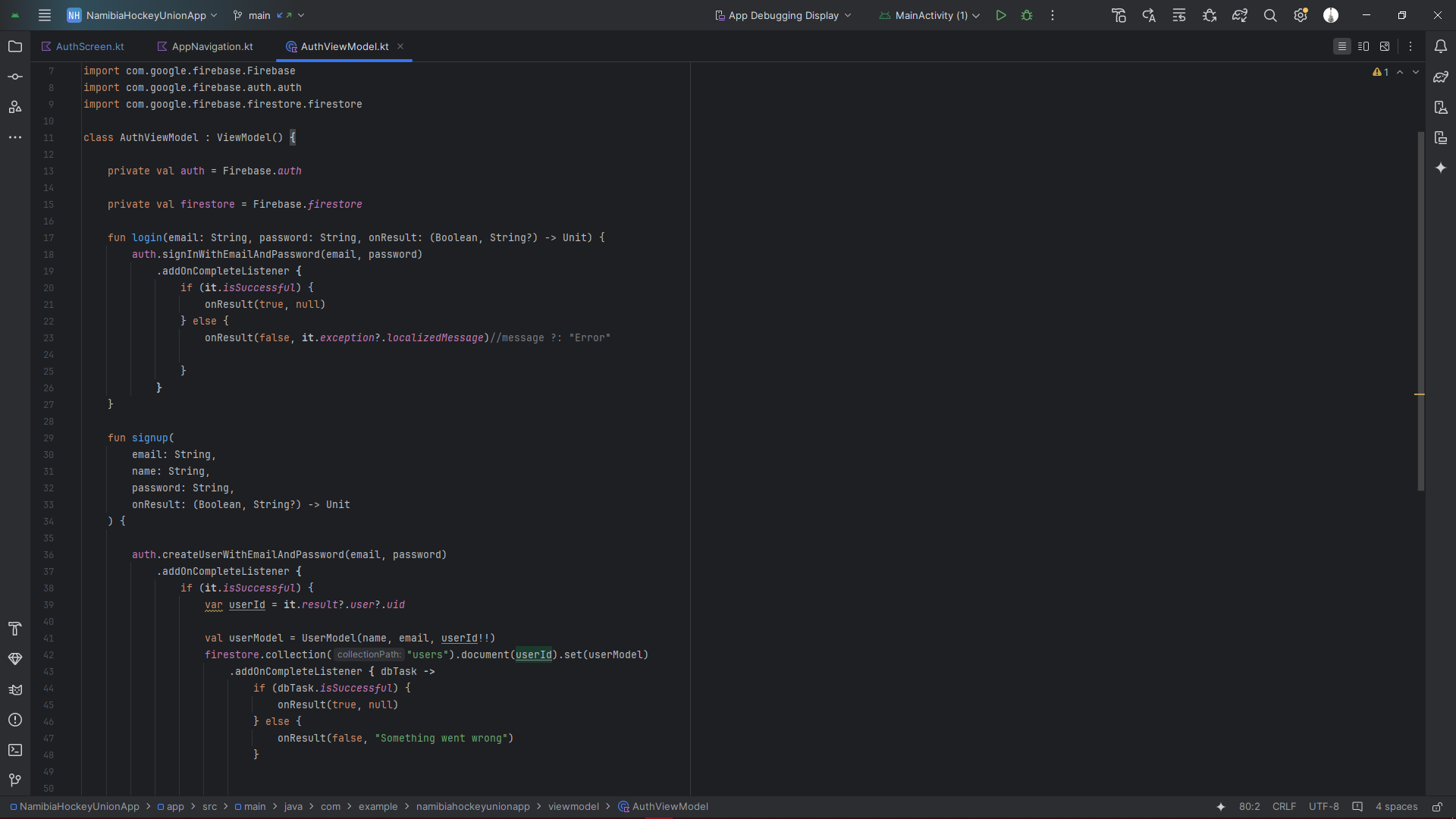
*Image Description: Team Registration Page* *Image Description: Teams Page*

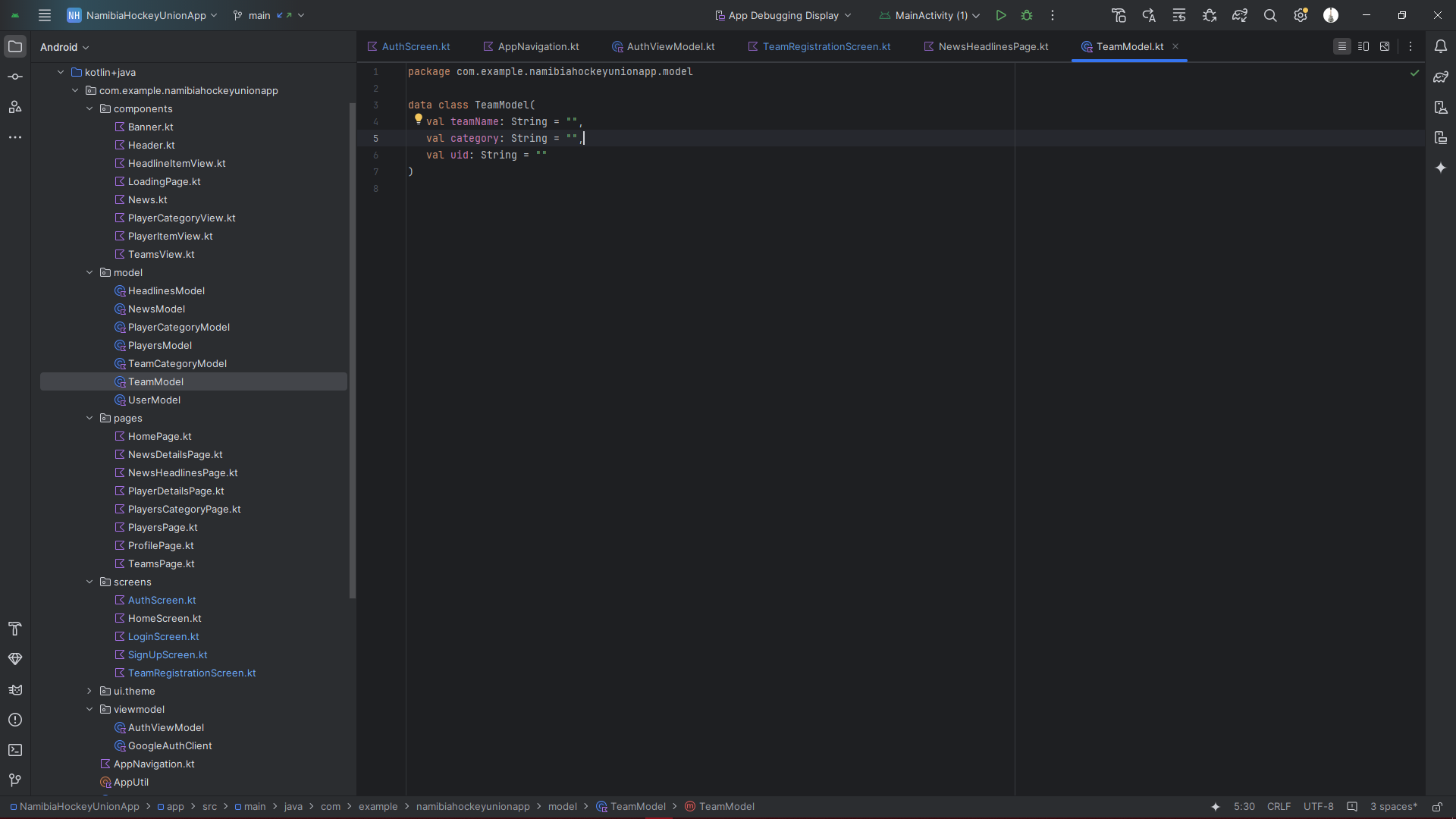


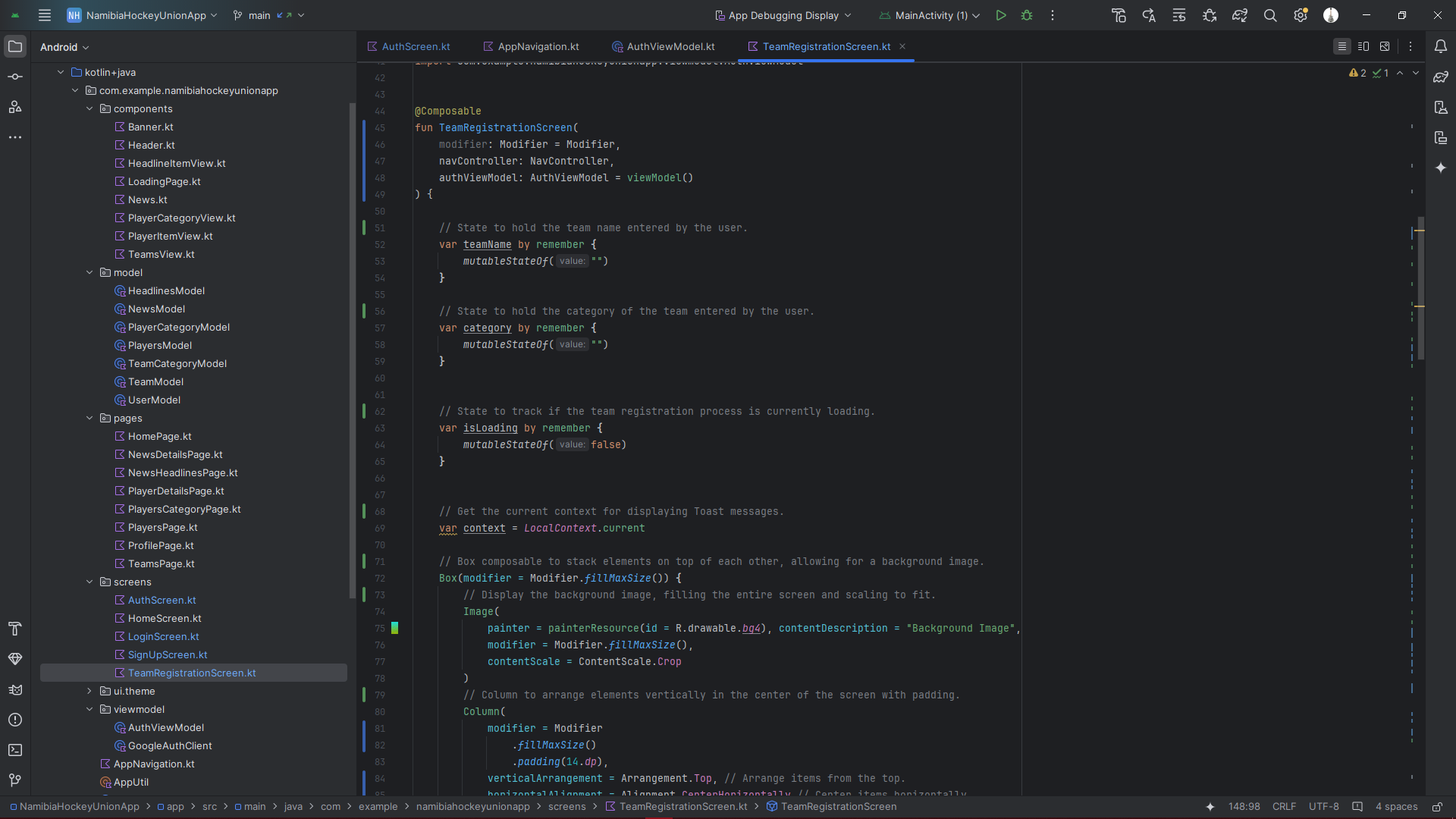
*Image Description: Player Category Page*  *Image Description: Profile Page*

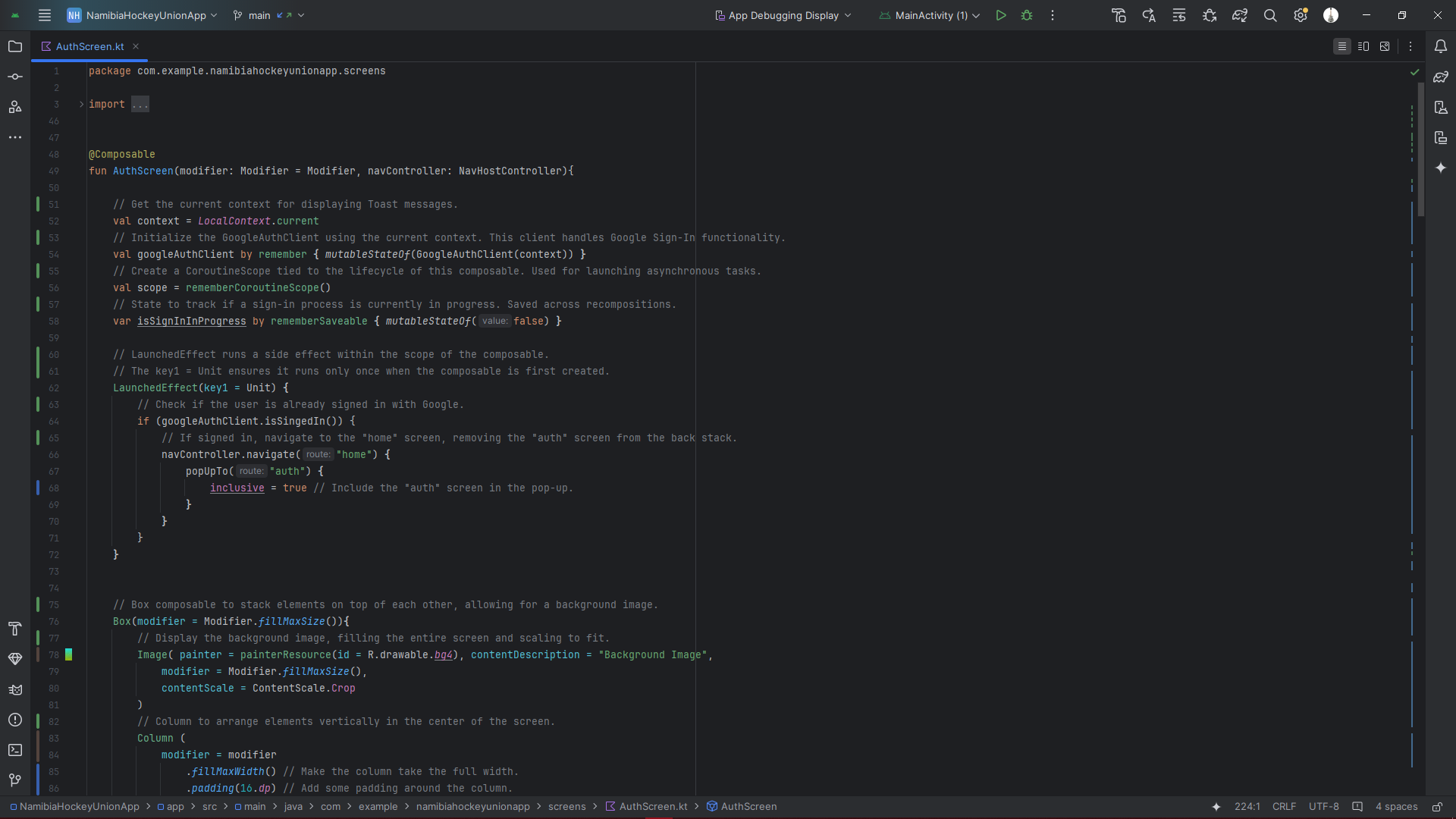
*Image Description: News Headlines Page*  *Image Description: Headline Details Page*

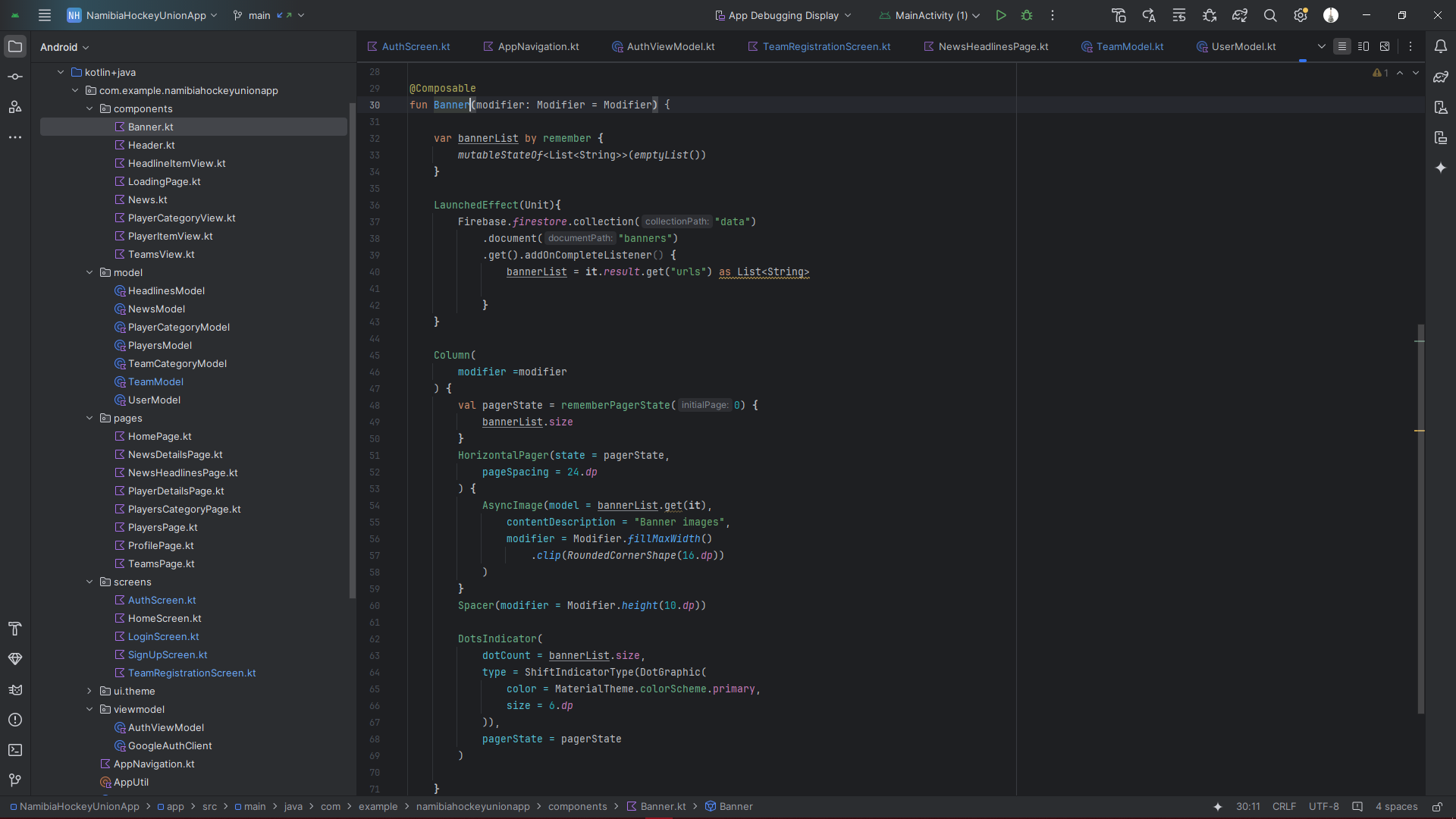
## 10. Kotlin Code Screenshots

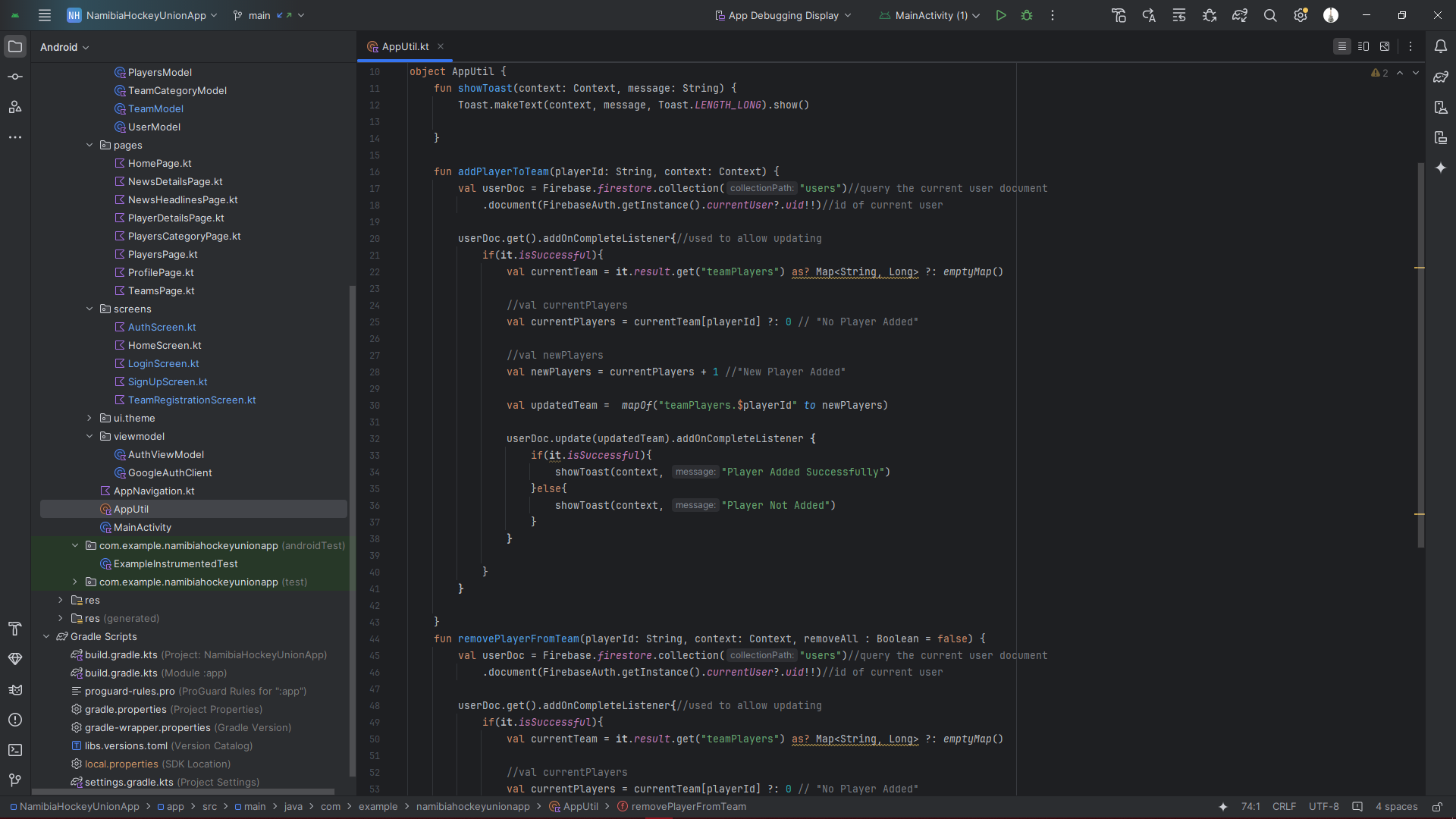
*Image Description: Code for AuthViewModel*

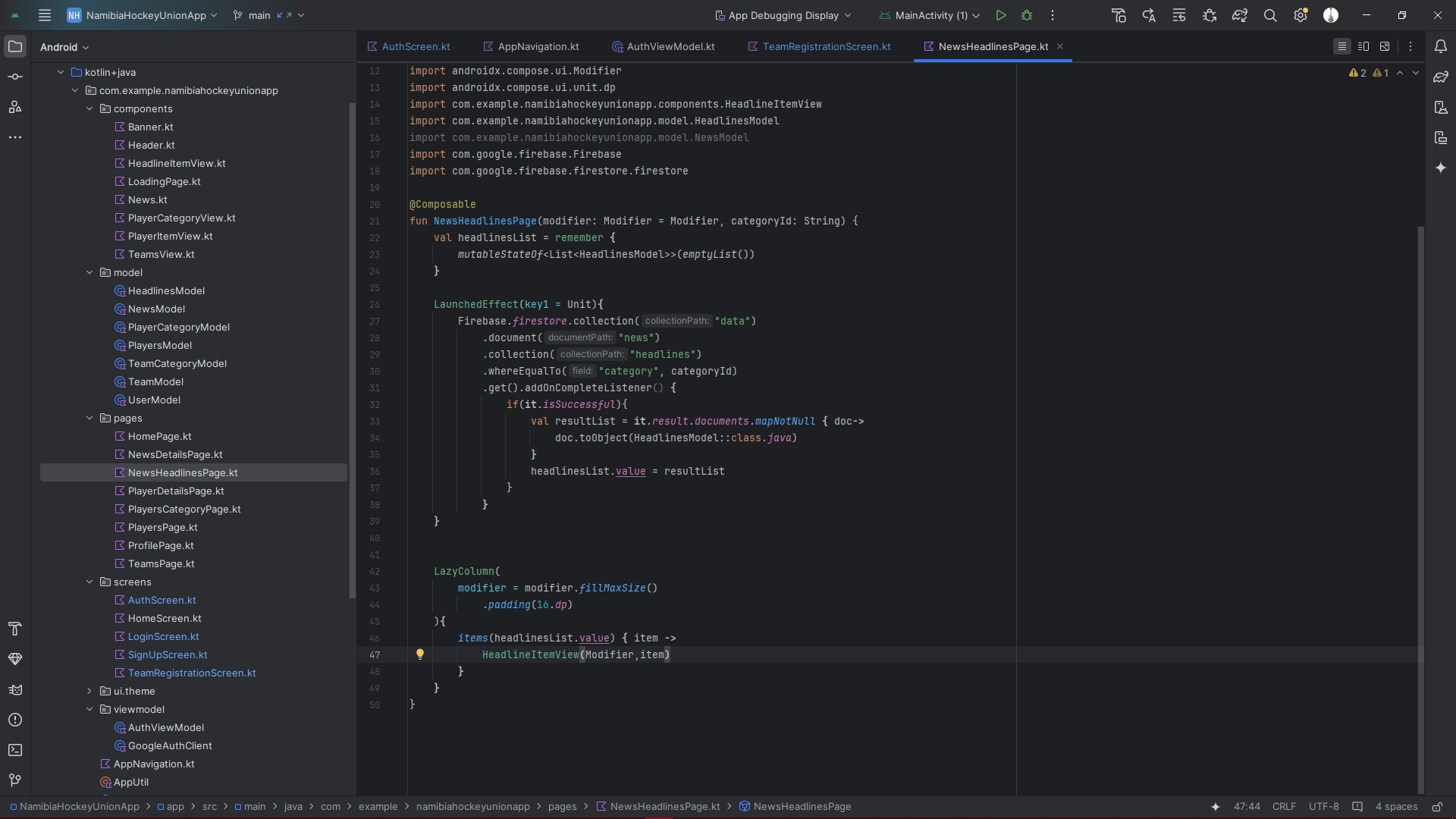
*Image Description: Code for Team Model Class*

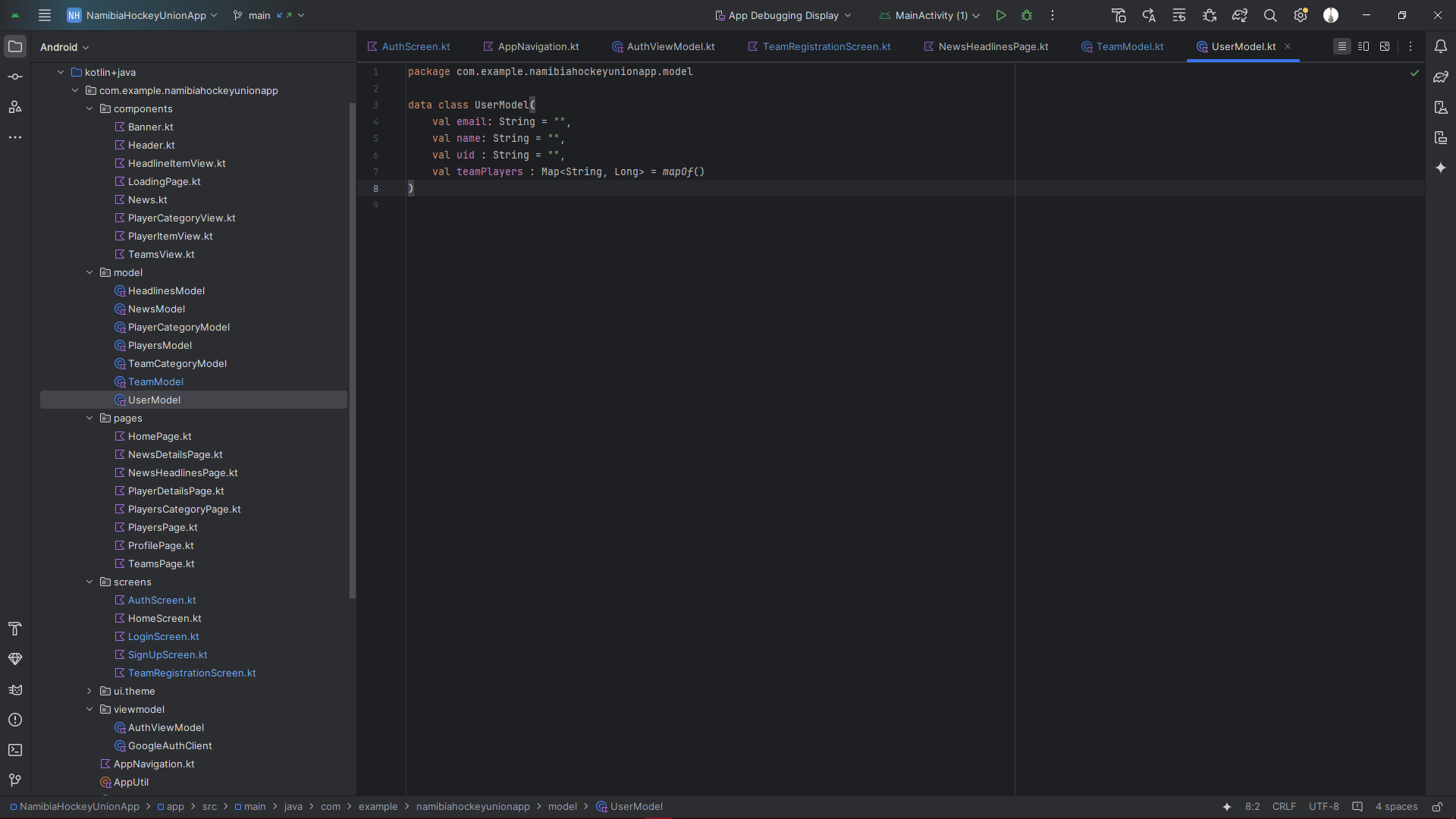
*Image Description: Code for Team Registration*

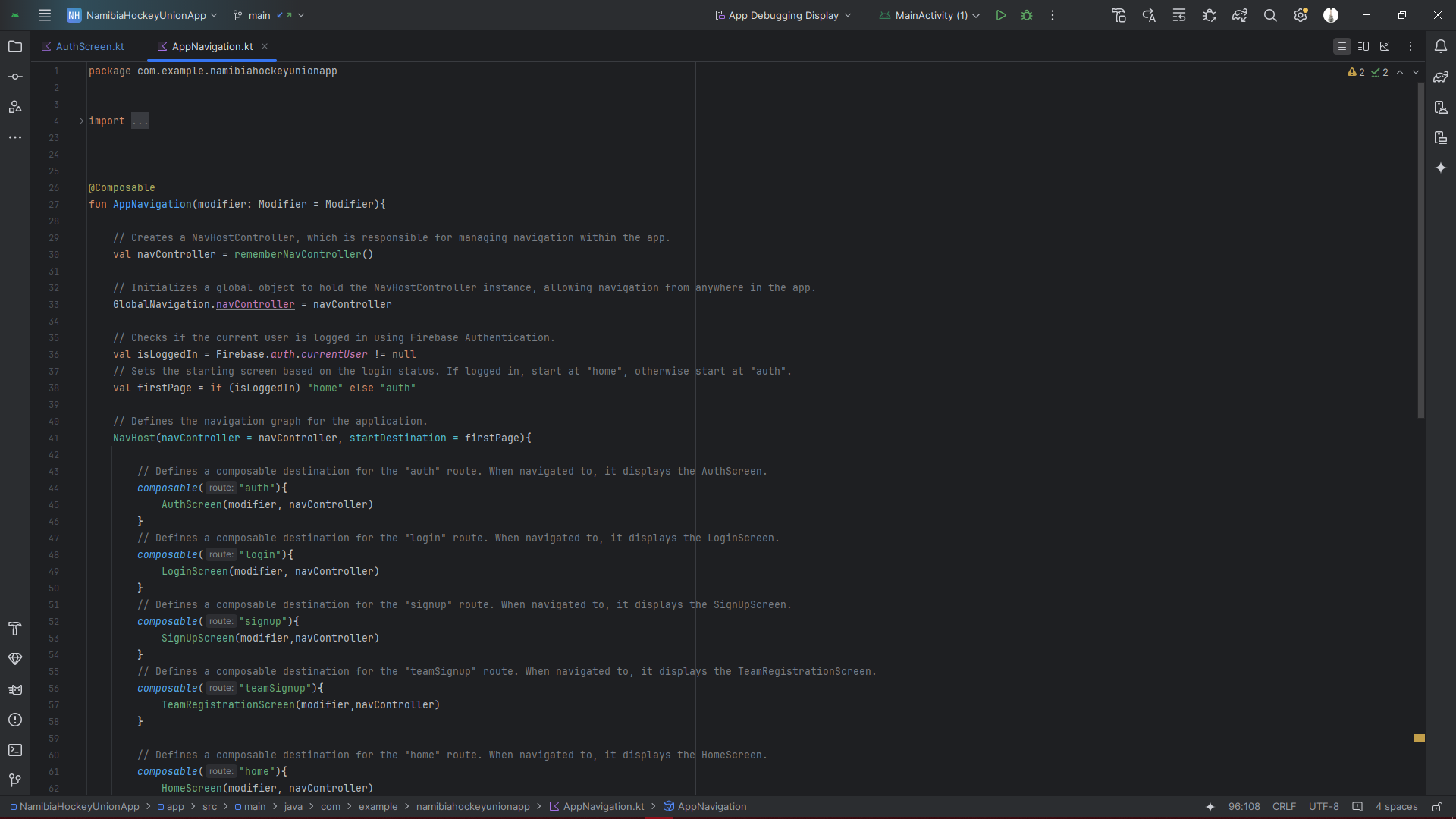
*Image Description: Code for Auth Screen (Landing Page)*

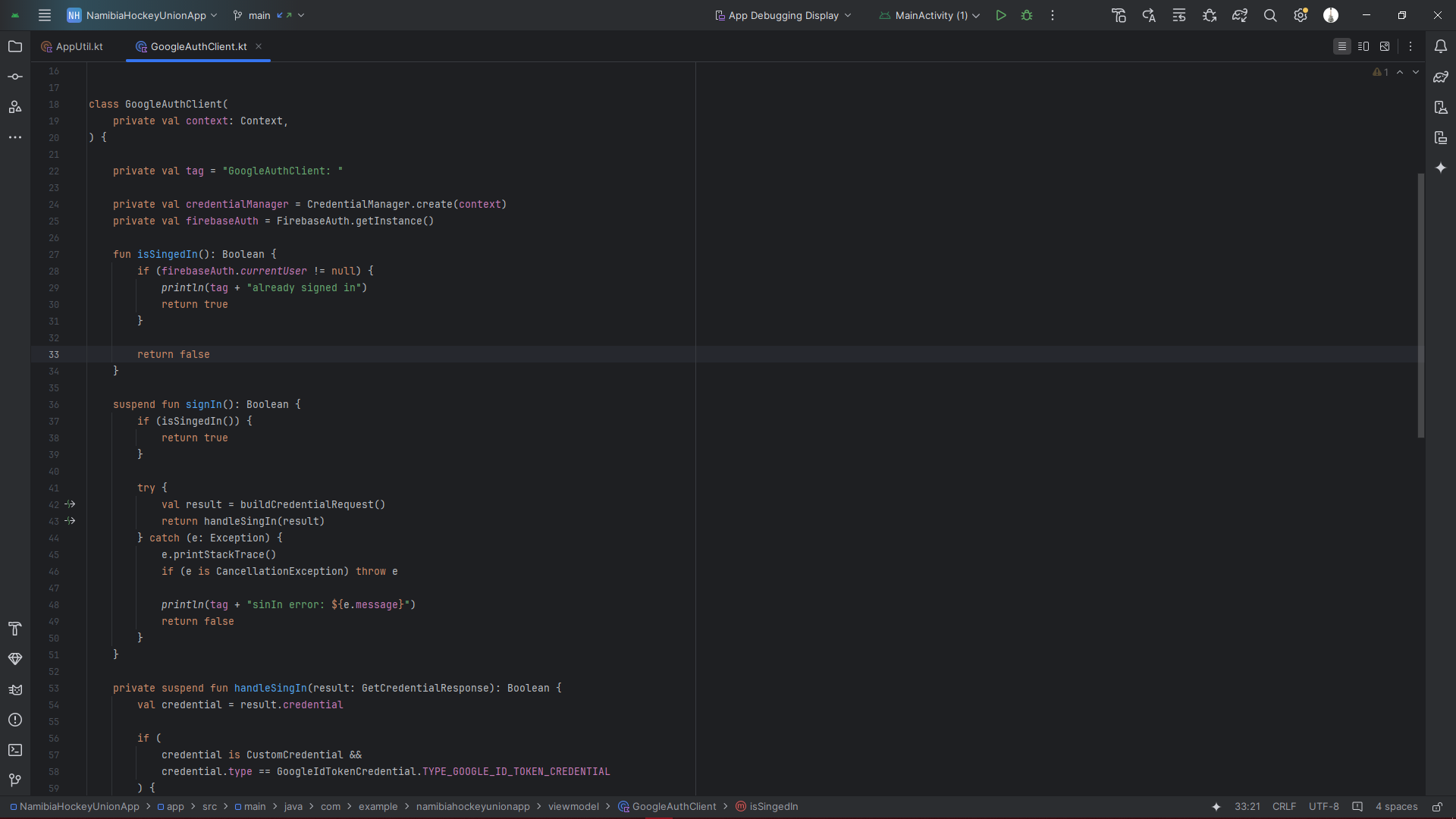
*Image Description: Code for Banner View*

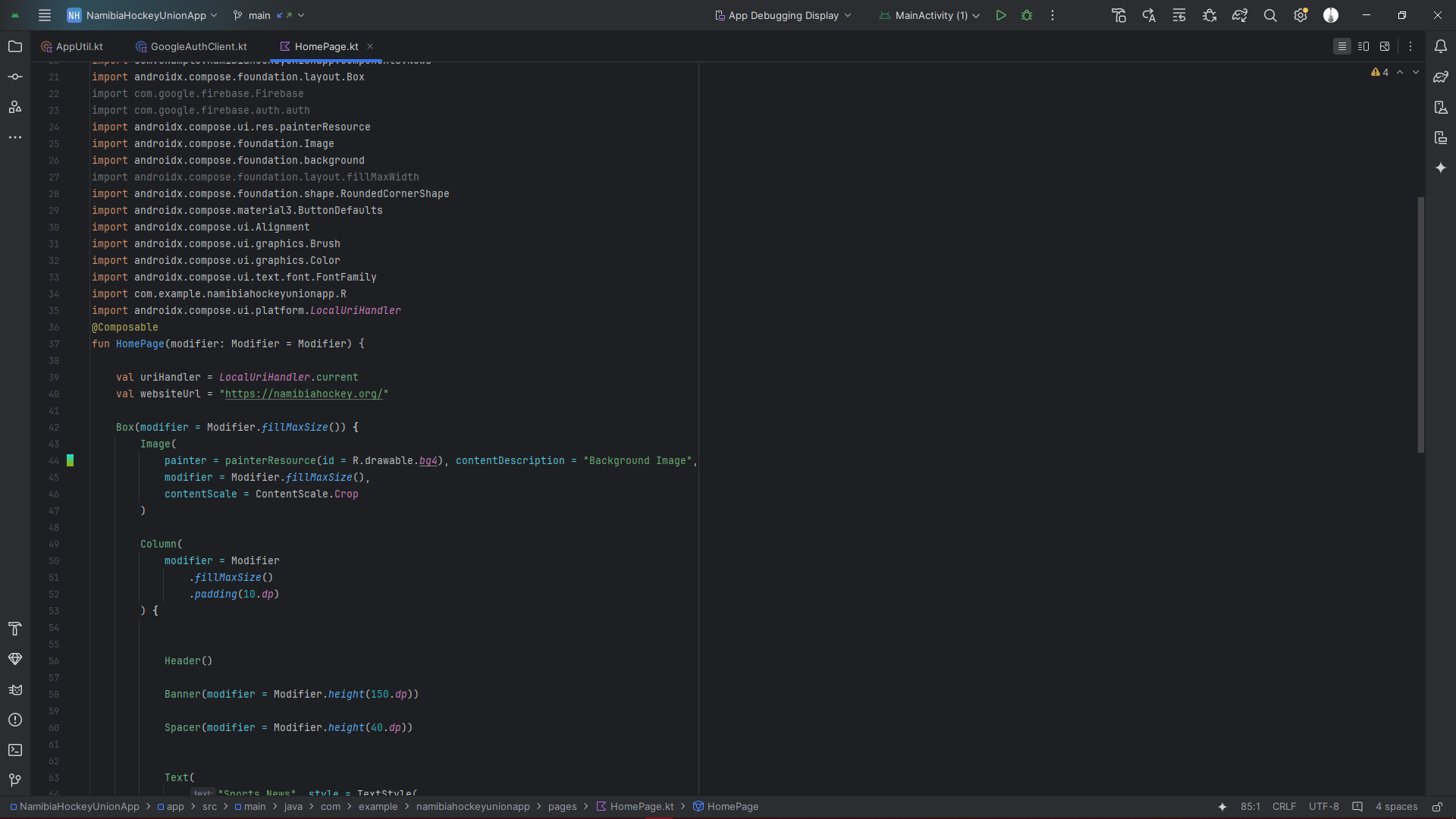
*Image Description: Code for App Util, for implementing backend operations*

*Image Description: Code for displaying News Headlines*

*Image Description: Code for User Model*

*Image Description: Code for App Navigation, using NavController*

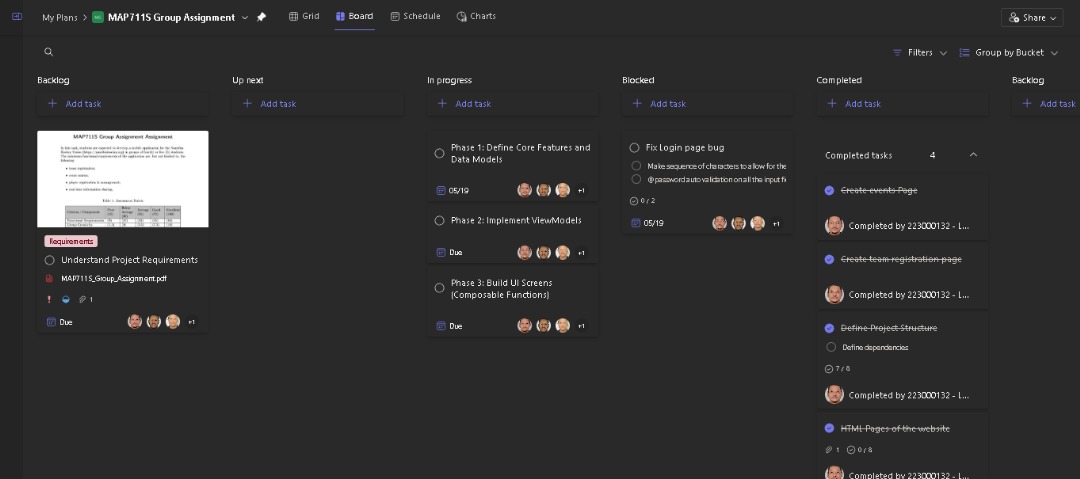
*Image Description: Code for Google Login implementation*

*Image Description: Code for Home Page*

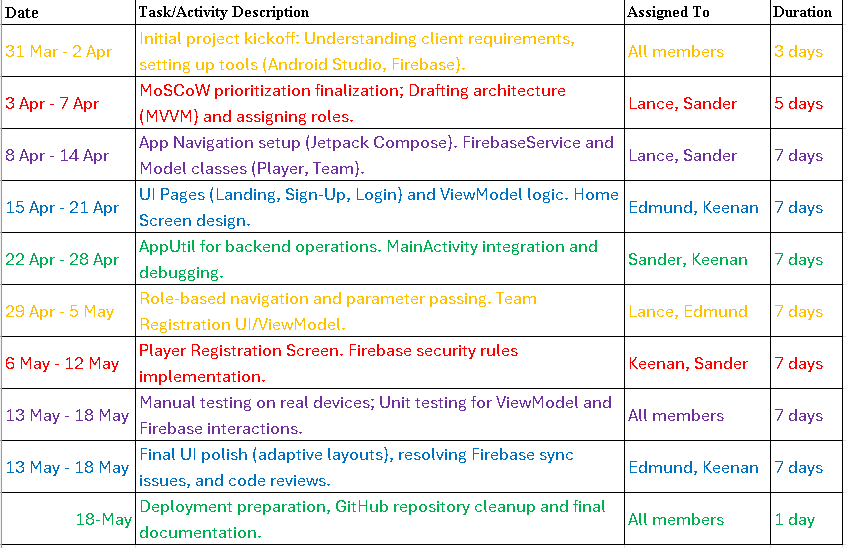
## 11. GitHub Repository

## 12. Project Management

Additionally, we made use of MS Planner to manage our project deliverables in a timely fashion.



## 13. Log Sheet



## 14. Timeline Analysis

A white sheet with colorful arrows

AI-generated content may be incorrect.