Documentation: ReelTime -Android Application

Project Name: ReelTime
App Version: 1.0.0
Author / Developer: Arslan Ahmed
Date: 06 Dec 2025

Contents

1. Project Overview

1.1 Introduction

ReelTime is a feature-rich Android application developed for browsing movies and booking tickets. It demonstrates a full-stack mobile development approach, from a dynamic, user-centric UI to a real-time Firebase backend integrated with Cloudinary for image storage.

1.2 Purpose and Problem Solved

ReelTime provides a seamless and intuitive experience for discovering movies and booking tickets. It solves the problem of complex, slow, or cluttered booking platforms by providing a simple, fast, and visually appealing interface.

1.3 Target Audience

Casual movie-goers who want a quick, visually driven experience for browsing films and managing bookings on their mobile devices.

1.4 Platform

- Operating System: Android
- Minimum SDK: 26 (Android 8.0 Oreo)
- Target SDK: 36 (Android 15)

2. Features

- **Secure User Authentication**: Firebase Email/Password login and registration, secure sessions.
- **Dynamic Home Screen**: Personalized greetings, real-time movie lists fetched from Firebase.
- **Comprehensive Movie Details**: Poster, summary, IMDb rating, cast, genre.
- **Interactive Seat Selection**: Real-time seat availability and price updates.
- **QR Code Ticket Generation**: Unique QR code for each ticket using ZXing library.
- **Saved Movies (Watchlist)**: Persistent user-specific bookmarks.
- **Ticket Management**: View, manage, and delete previously booked tickets.
- **User Profile**: Displays user info and allows logout.
- **Polished UI**: Material Components, smooth transitions, custom UI elements, real-time blur effect.

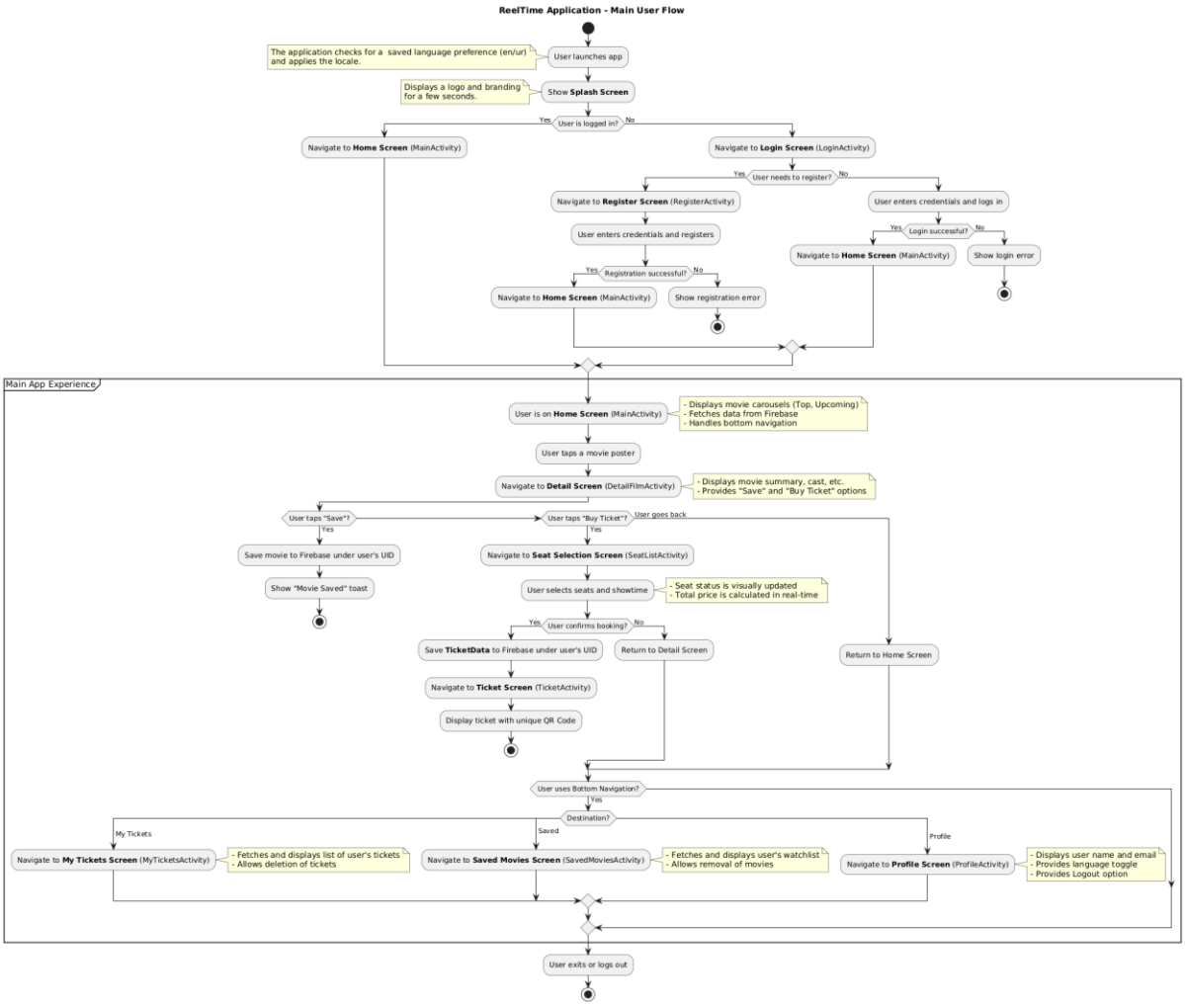
3. Architecture and Design

3.1 App Architecture

The app follows **MVVM** principles:

- **Model**: Data classes (Film, TicketData) and Firebase Realtime Database.
- **View**: XML layouts and Activities.
- **ViewModel (Conceptual)**: Business logic handled in Activities; future scalability would move this to androidx.lifecycle.ViewModel.

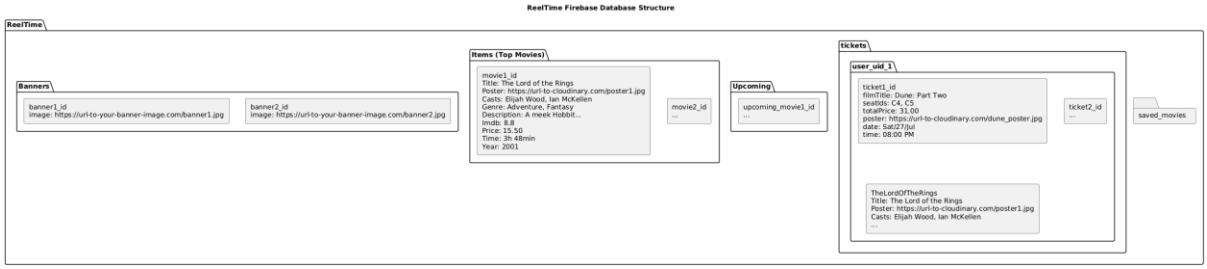
3.2 Application Flow



3.3 UI/UX Design

- Dark cinematic theme.
- Visual hierarchy with bold typography and accent colors.
- Real-time feedback on interactions.
- Component-based UI for consistency (CardView, RecyclerView).

3.4 Database Structure



3.5 External APIs & Services

- **Firebase**: Authentication and real-time data storage.
- **Cloudinary**: Cloud-based image hosting for all movie posters and cast images. Firebase stores the image URLs, and the app fetches them in real-time.

4. Technical Details

- **Programming Language**: Kotlin
- **Frameworks & Libraries**: Android SDK, AndroidX, Material Components, View Binding, Firebase (Auth + Realtime Database), Glide, Chip Navigation Bar, BlurView, ZXing
- **SDK Versions**: minSdk 26, targetSdk 36, compileSdk 36
- **Permissions**: INTERNET for network operations
- **Development Tools**: Android Studio, Gradle, Git

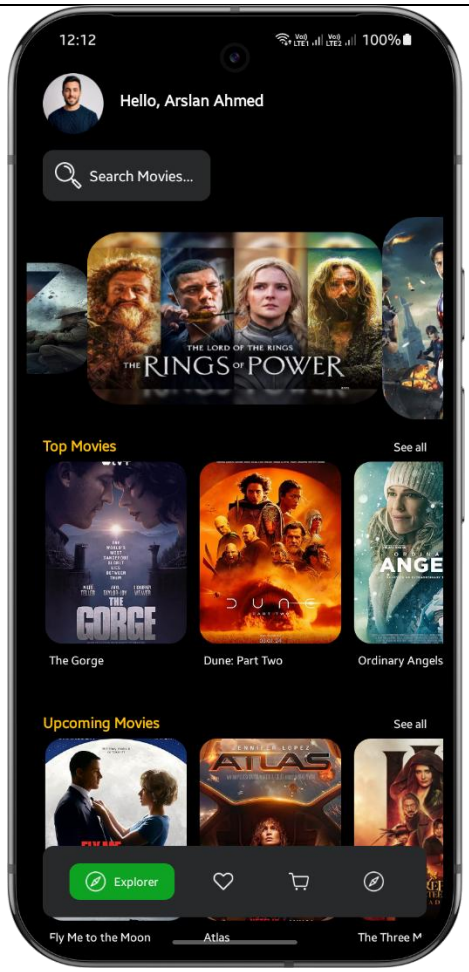
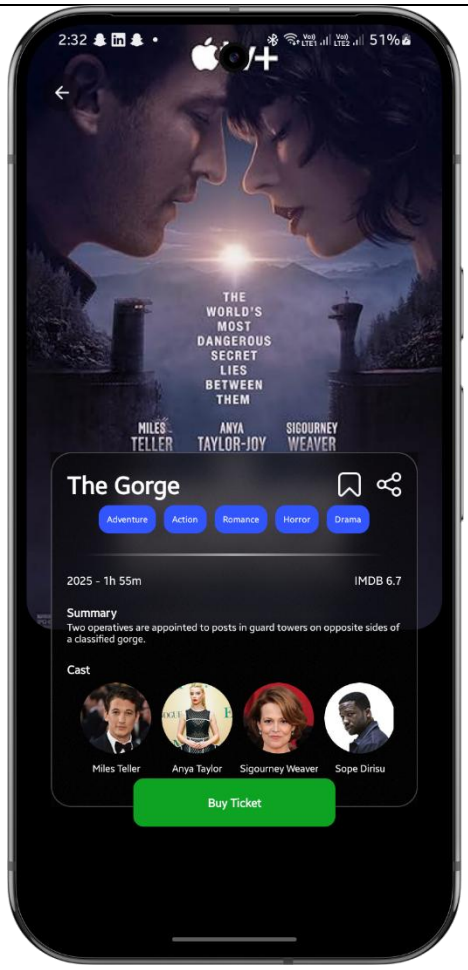
5. Installation / Setup

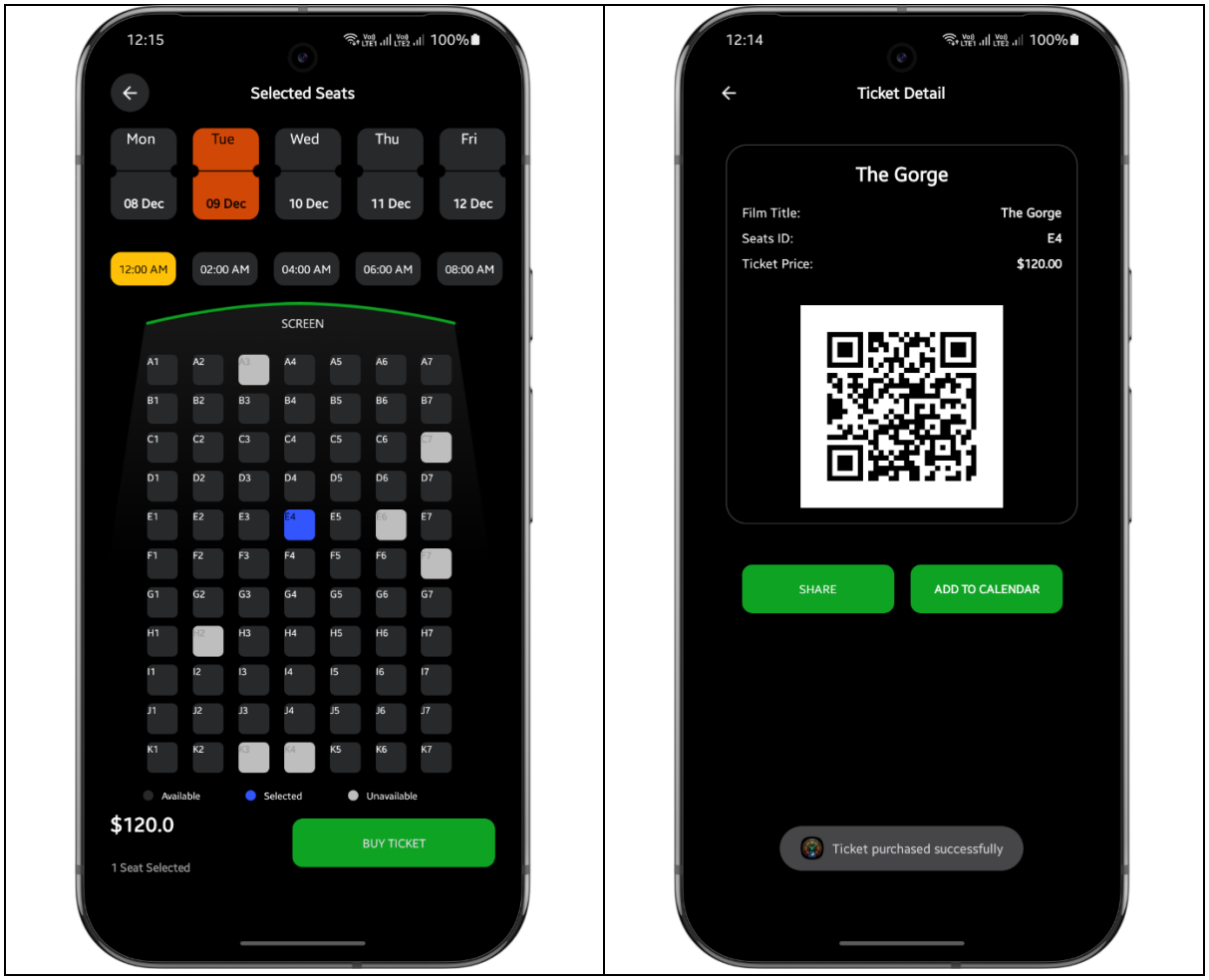
1. Clone repository:
   git clone https://github.com/your-username/ReelTime.git
2. Firebase Setup:
   - Create Firebase project.
   - Add Android app with package com.arslan.reeltime.
   - Place google-services.json in app/.
   - Enable Email/Password authentication and Realtime Database.
3. Cloudinary Setup (used for images):
   - Movie and cast images are hosted on Cloudinary. Firebase stores their URLs. Ensure app has network access to fetch images.
4. Build: Open in Android Studio, sync Gradle, click "Build Project".

5. Run: Run on emulator or device.

6. Usage Instructions

1. Launch the app and register or log in.
2. Browse "Top Movies" and "Upcoming Movies".
3. Tap a movie poster to view details.
4. Save to watchlist using bookmark icon.
5. Tap "Buy Ticket" → select seats → observe price updates → finalize booking.
6. View tickets in "My Tickets" or saved movies in "Saved".

Home Screen	Movie Details
	
Seat Selection Screen	**Ticket Detail**



7. Testing

- **Manual Testing**: Verified all flows on emulators and devices.
- **Test Cases**: Registration/login, home screen data, navigation, seat selection, ticket generation, watchlist, language toggle.
- **Known Limitations**: No real payment gateway, no search functionality, no unit tests implemented yet.

8. Deployment / Release

1. Build signed APK: Go to Build tab then Generate Signed Bundle / APK then Select APK then Next.
2. Create or select key store for signing.
3. Choose release variant and press Finish. APK saved in app/release.

9. Code Documentation

- Packages:

- activity: UI controllers
- adapter: RecyclerView adapters
- model: Data classes

10. Future Work

- Add real-time search.
- Add movie trailers via YouTube API.
- Integrate payment gateway (Stripe / Google Pay).
- Implement unit and UI tests (JUnit, Espresso).

11. References

- Firebase Documentation
- Glide Documentation
- Chip Navigation Bar
- BlurView
- ZXing (Zebra Crossing)
- Cloudinary