Final Review Report: EquitySpin Mobile Application

Names (Group 5):
Carlos Casanova Romero
Ricardo Vazquez Alvarez
Mario Coronado Fernandez

Introduction

EquitySpin has been successfully developed as a comprehensive mobile application for portfolio management, integrating real-time market data, community insights, and personalized portfolio tracking. This report outlines the implemented functionalities, technical architecture, and user experience design. This makes EquitySpin a valuable tool for investors and users interested in managing an investment portfolio and staying up to date with the latest news and stock values.

Work Distribution

All team members contributed to multiple areas of the project to ensure balanced workload and comprehensive understanding of the codebase. Weekly scrum meetings ensured coordination, progress tracking and knowledge sharing across the team.
All tasks were logged and tracked using an Excel-based planner, allowing us to prioritize and assign tasks based on our scrum meetings.

Team Member	Role	Main Tasks
Ricardo Vazquez Alvarez	Frontend Lead	UI/UX design, Jetpack Compose implementation, Material Theme, Layouts.
Mario Coronado Fernandez	Backend Lead	Firebase integration, database architecture, Authentication system, Asynchronous data flow management.
Carlos Casanova Romero	API integration	Yahoo Finance & Reddit API integrations. Chart Visualization.

Main Features

1. API Integration

A core feature of EquitySpin is its integration with external data sources to provide users with timely, relevant, and accurate financial information. This is achieved through two main APIs:

- Yahoo Finance API, used to fetch real-time stock prices and historical data for portfolio tracking.
- Reddit API, specifically from r/wallstreetbets, used to fetch trending market discussions and community sentiment.

These integrations follow a modular network architecture involving three main layers: Instance, ApiService, and Response classes.

1. **Instance** (e.g., YahooFinanceInstance, RedditInstance). Creates a Retrofit client configured with the respective base URL, and then, uses GsonConverterFactory to parse JSON responses into Kotlin data classes.

2. **ApiService** (e.g., YahooFinanceApiService, RedditApiService). Defines HTTP methods (@GET) and query parameters to fetch specific data. Uses suspend functions to leverage Kotlin Coroutines for asynchronous, non-blocking calls. One key difference between the Reddit and Yahoo APIs in this layer is that the Reddit API does not require headers, while Yahoo APIs require custom headers (API key and host).

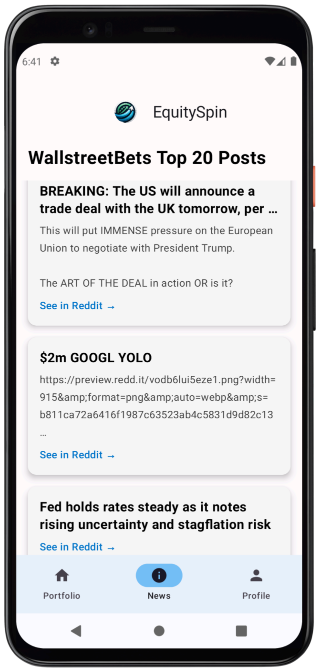
3. **Response** (e.g., YahooFinanceResponse, RedditResponse). Contains data classes that model the API response JSON structure. In them, fields are often nested and require multiple data classes (e.g., children, data, body), to properly parse the JSON. There's a particular case in YahooHistoricalResponse, that uses @SerializedName for JSON fields with non-standard naming.

1.1. Reddit API Market News Integration

EquitySpin provides users with timely, relevant financial information through its robust news integration feature. The application seamlessly connects to the Reddit API, specifically targeting the highly active r/wallstreetbets community, to deliver current market trends and community sentiment directly to users. With the top 20 hottest posts of the moment.

Implementation Highlights:

- Successfully integrated the Reddit API to fetch the top 20 trending posts from r/wallstreetbets.
- Designed an intuitive card-based news interface that displays post titles and descriptions.
- Implemented direct linking functionality that redirects users to the original Reddit posts for deeper engagement.
- Created a modular architecture for API integration that ensures scalability and maintainability.


6:41

EquitySpin

WallstreetBets Top 20 Posts

BREAKING: The US will announce a trade deal with the UK tomorrow, per …

This will put IMMENSE pressure on the European Union to negotiate with President Trump.

The ART OF THE DEAL in action OR is it?

See in Reddit →

$2m GOOGL YOLO

https://preview.redd.it/vodb6lui5eze1.png?width=915&amp;format=png&amp;auto=webp&amp;s=b811ca72a6416f1987c63523ab4c5831d9d82c13 …

See in Reddit →

Fed holds rates steady as it notes rising uncertainty and stagflation risk

See in Reddit →

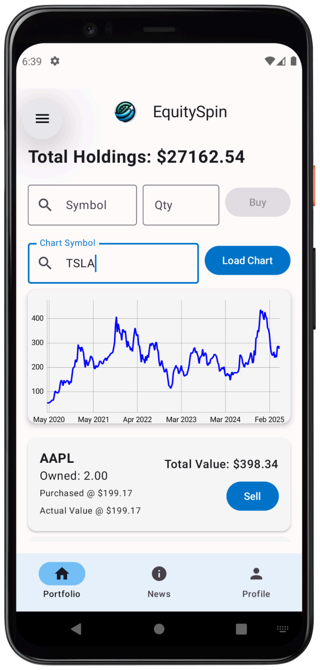
The news screen offers users a quick overview of market sentiments and trending discussions, helping them stay informed about potential investment opportunities or market shifts without having to manually browse multiple sources.

1.2. Portfolio Management with Yahoo Finance Integration

The core of EquitySpin is its comprehensive portfolio management system powered by Yahoo Finance API integration through Rapid API, allowing users to track their investments with real-time market data. Buy and sell stocks.

Implementation Highlights:

- Established connection with Yahoo Finance API to retrieve real-time stock prices and historical data.
- Developed a robust portfolio tracking system that calculates current market value against purchase value.
- Implemented buy and sell functionality with quantity selection for precise portfolio management.
- Created a clean, informative portfolio dashboard that displays owned stocks, quantities, and performance metrics such as overall portfolio value.
- For each owned stock, purchased value and actual value is displayed showing real-time information.


6:39

EquitySpin

Total Holdings: $27162.54

Symbol | Qty | Buy

Chart Symbol: TSLA | Load Chart

AAPL
Owned: 2.00
Purchased @ $199.17
Actual Value @ $199.17
Total Value: $398.34
Sell

The portfolio page serves as the central hub for users to monitor their investments, make informed decisions, and track their overall financial performance in the market.

3. Interactive Chart Visualization

One of the best features implemented in EquitySpin is the interactive chart visualization system using the MPAndroidChart library by PhilJay in GitHub, which provides users with detailed historical performance data for any stock in their portfolio or of interest.

Implementation Highlights:

- Successfully integrated the MPAndroidChart library to create responsive, interactive line charts. Ensuring smooth user experience.
- Developed a custom Composable component that displays 5-year historical performance for any selected stock.
- Implemented dynamic data fetching from Yahoo Finance historical endpoint.

This feature allows users to analyze long-term trends and make more informed investment decisions based on historical performance rather than simply current market prices.

4. Firebase Integration for Authentication and Data Storage

Firebase plays a key role in the architecture of the application. It is used for both user authentication and cloud-based data storage through Firestore. These services allow the application to securely manage user accounts and persistently store portfolio data without the need to maintain a custom backend server.

Firebase Authentication

Firebase Authentication is used to handle all user-related operations, including sign-up, login, and logout. Upon registering or logging in, a user is authenticated via email and password credentials. The FirebaseAuth class provides the necessary interface for interacting with the authentication system. When a user successfully signs up or logs in, they are redirected to the portfolio screen. Logout functionality simply invalidates the current session and returns the user to the login interface.

Cloud Firestore

Firestore is used as the database to manage each user's portfolio information. Upon authentication, each user is assigned a unique document. Within each user's document, a subcollection called holdings stores the individual assets they own. Each asset is represented by a document containing fields such as the quantity and the purchase price.

```
users/
        └── [uid]/
                └── holdings/
                        └── [symbol] → { quantity, purchasePrice }
```

The application interacts with Firestore through several core operations:

- Loading Holdings: Once the user is authenticated, their asset holdings are fetched from Firestore. Each document under the holdings subcollection is parsed and displayed in the Portfolio screen using Jetpack Compose.

- Buying Assets: When a user buys an asset, its current market price is first retrieved from the Yahoo Finance API. Then, the app then checks whether the asset already exists and, if it does, the quantity and average purchase price are updated accordingly. Otherwise, a new document is created.

- Selling Assets: When selling, the application retrieves the current quantity of the asset and updates or deletes the document depending on whether the entire quantity is sold or only a portion. Selling operations are also validated to prevent users from selling more than they own.

All of these operations are executed asynchronously using Kotlin Coroutines to ensure that the UI remains responsive and to allow proper state management within the ViewModel architecture.

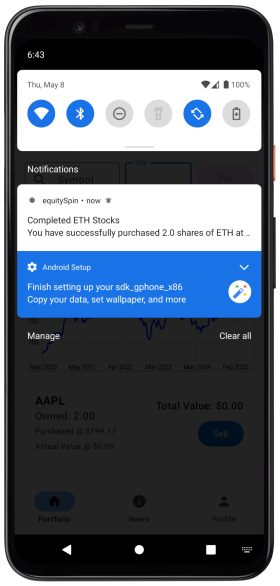
Description of Secondary Features

1. Third-Party Libraries Integration

EquitySpin uses several third-party libraries to enhance functionality and user experience, main ones are:

- **MPAndroidChart**: Implemented for advanced chart visualization capabilities, enabling interactive line charts that display historical stock performance.
- **Retrofit**: Used for efficient network requests to both Reddit and Yahoo Finance APIs, with custom configurations for different endpoints.

2. Notification System


6:43

Thu, May 8 100%

Notifications

equitySpin • now

Completed ETH Stocks
You have successfully purchased 2.0 shares of ETH at ..

Android Setup

Finish setting up your sdk_gphone_x86
Copy your data, set wallpaper, and more

The app features a comprehensive notification system that keeps users informed about their portfolio activities.

Implementation Details:

- Developed a NotificationHelper class that manages notification creation and display.
- Implemented status bar notifications for buy and sell transactions.
- Created user-friendly notification content with transaction details

This feature ensures users remain informed about their portfolio changes even when not actively using the application.

3. Material Design Implementation

EquitySpin follows Material Design principles throughout the application:

- Consistent color scheme and typography.
- Intuitive navigation patterns.
- Responsive layouts that adapt to different screen sizes.
- Smooth animations and transitions between screens.
- Profile screen with personalized user name.

4. Extensive Debug Logging with Logcat

To facilitate smooth development and easier maintenance, EquitySpin includes a robust logging mechanism using Android's Logcat system.

Implementation Details:

- Strategic use of Log.d, Log.e, and Log.i statements across key modules to trace API responses, user interactions, and application state.
- Debug logs are particularly useful during data fetching, error handling, and navigation events.

This feature significantly improved our ability to detect and resolve issues efficiently throughout development.

Considerations to Execute the APP

The EquitySpin app is tested on a **Pixel 4** with **API Level 30 (Android 11)**

Link to Google Drive shared source code Zip File

https://drive.google.com/file/d/1hMKtNX77Te5tLi6iF9mhlkK3mGEenia-/view?usp=sharing

Link to Youtube Video App Demo

https://youtu.be/JCTn1clHrWs

Summary and Outlook

We have successfully achieved our objective of creating a comprehensive investment portfolio management application that combines real-time market data, community insights, and personal portfolio tracking. The application provides users with the tools they need to make informed investment decisions and monitor their performance.

Most important achievements include:

- Implementation of a reactive UI pattern where portfolio data updates propagate automatically through StateFlow collections.
- Efficient integration of multiple external APIs.
- Secure user authentication with Firebase.
- Efficient chart rendering for large historical datasets using MPAndroidChart.

Future plans for the application include:

- Adding customizable price alerts with notifications.
- Expanding news sources beyond Reddit.
- Creating a watchlist feature for tracking potential investments.
- Customizable Profile features such as profile image.
- Unique EquitySpin portfolio design

References

The references used to inspire, help and guide the development of our app are as follows:

- Bonigarcia. (n.d.). *GitHub - bonigarcia/android-examples: Sample Android apps created with Kotlin and Java*. GitHub. https://github.com/bonigarcia/android-examples/tree/main

- OpenAI. (2025). ChatGPT (Version 4) [Large language model]. OpenAI. https://openai.com/chatgpt

- Google. (n.d.). *Jetpack Compose documentation*. Android Developers. https://developer.android.com/develop/ui/compose/documentation?hl=es-419

- PhilJay. (n.d.). *MPAndroidChart* [GitHub repository]. GitHub. https://github.com/PhilJay/MPAndroidChart