KHAN INSTITUTE OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY, IST CAMPUS KAHUTA

COMPUTER SCIENCE DEPARTMENT

PROJECT PROPOSAL

HealHive Mobile Application

Submitted by:

Sareen Fatima

232201061

Submitted to:

Sir Uzair Hassan

Department of Computer Science

December 7, 2025

Contents

1 Introduction

HealthHive is a mobile application designed to simplify the management of user health data. The app provides features such as personalized dashboards, health monitoring, secure authentication, real-time notifications, and cloud-based storage using Firebase. This proposal outlines the project architecture, key modules, technology stack, and workflows involved in building a scalable, maintainable, and user-friendly health tracking application.

2 Project Objectives

The primary objectives of HealthHive are as follows:

- Provide an intuitive and interactive interface for users to track daily health metrics including steps, heart rate, and activity.
- Ensure secure storage and retrieval of health data using Firebase Authentication and Real-time Database.
- Deliver timely notifications and reminders to enhance user engagement and adherence to health routines.
- Follow the MVVM architecture with Jetpack Compose UI for clean separation of concerns, modularity, and maintainability.
- Facilitate future scalability to add new features like analytics dashboards, AI recommendations, and health reports.

3 Technology Stack

The project employs modern technologies to ensure efficiency, scalability, and maintainability:

- **Frontend:** Jetpack Compose and Material 3 components for modern Android UI development.

- **Backend:** Firebase services including Authentication, Realtime Database, and Cloud Messaging for secure data handling.
- **Architecture:** MVVM (Model-View-ViewModel) to separate UI and business logic.
- **Programming Language:** Kotlin for native Android development.
- **Version Control:** Git and GitHub for source code management and collaboration.
- **Design Tools:** Figma for UI/UX design, wireframing, and prototyping.

4 System Architecture

HealthHive follows a modular layered architecture to ensure clarity, maintainability, and scalability. The key layers include the UI layer, ViewModel layer, Firebase API handlers, utility classes, and theme configuration.

4.1 Architecture Diagram

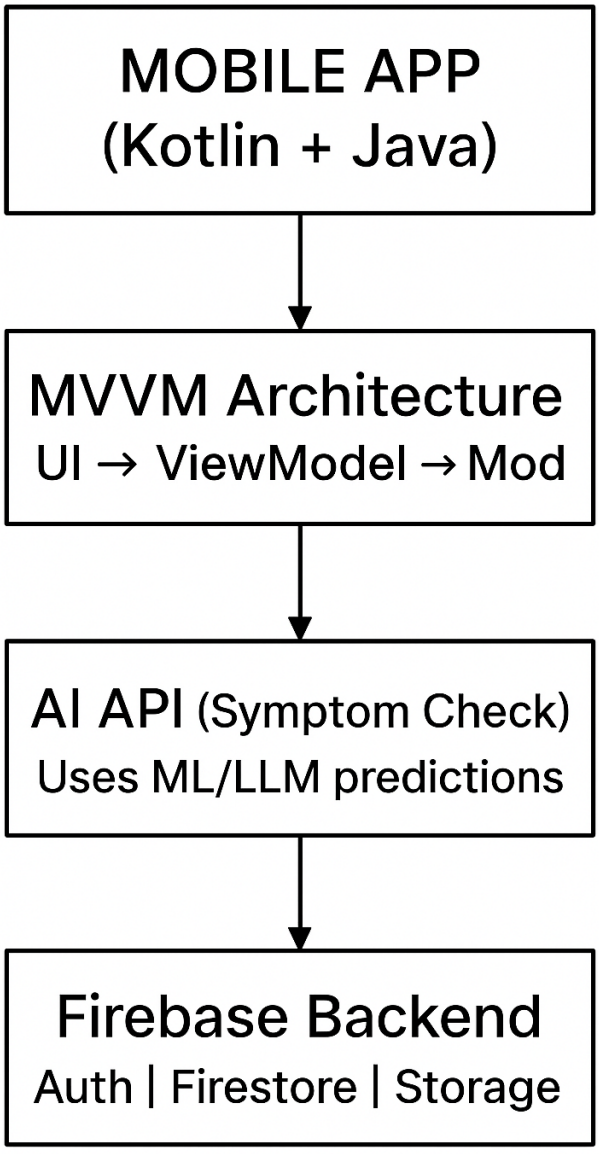

Figure 1: System Architecture: Illustrates data flow between Firebase backend, ViewModel, and Jetpack Compose UI components. Each layer is independent, allowing easy debugging, testing, and future feature integration.

5 Project Structure

```
HealthHive/
|
├── app/src/main/java/com/example/healthhive/
|   ├── adapters/          # Recycler/List adapters
|   ├── api/               # Firebase/API interfaces
|   ├── firebase/          # Auth & DB helpers
|   ├── model/             # Data classes
|   ├── notifications/     # Push notifications
|   ├── ui/
|   |   ├── screens/       # Compose screens (Home, Login, Notifications)
|   |   └── theme/         # Color, Typography, Theme
|   ├── viewmodel/         # MVVM ViewModels
|   └── OnboardingActivity.kt
|
├── res/                   # Resources (images, layouts, strings)
├── AndroidManifest.xml
├── build.gradle
└── settings.gradle
```

Figure 2: Project Structure: Organized into modules for adapters, API handlers, Firebase helpers, data models, notifications, UI theme, and ViewModels. This modular approach enhances readability, maintainability, and team collaboration.

6 User Interface Design

The UI is designed using Jetpack Compose and Material 3 to provide a consistent, modern, and responsive experience. It emphasizes clarity, interactivity, and user engagement.

6.1 Home Screen

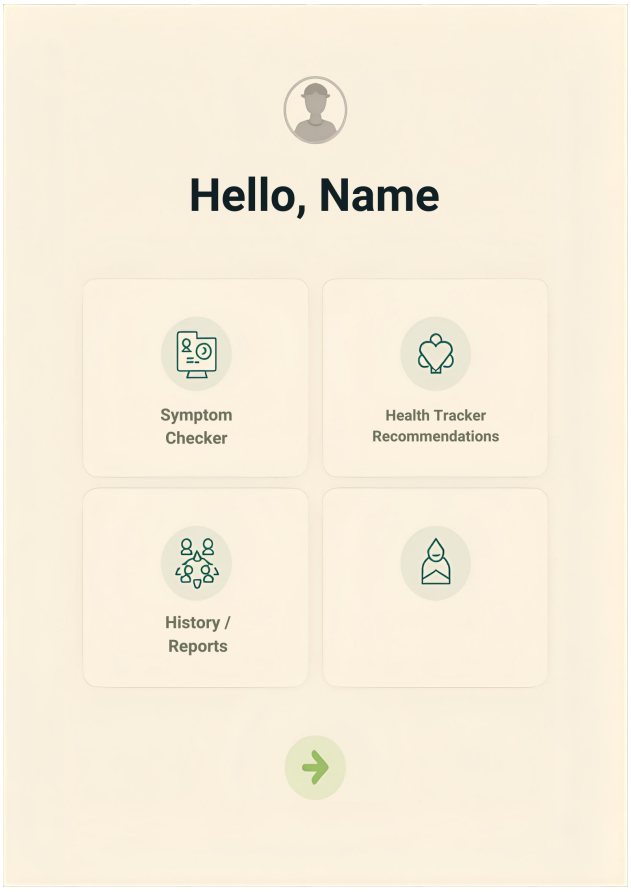

Figure 3: Home Screen: Displays key health metrics including steps, heart rate, and daily goals. The design uses cards and Compose layouts for clarity and ease of navigation.

6.2 Login Screen

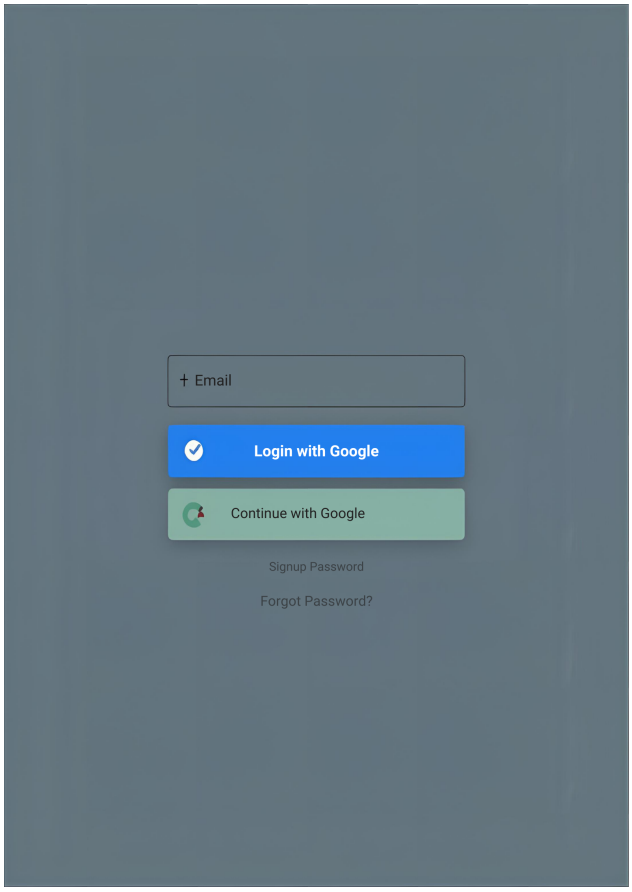

Figure 4: Login Screen: Provides secure authentication using Firebase. Error messages and guidance are displayed using Compose Snackbars for better user experience.

6.3 Notifications Screen

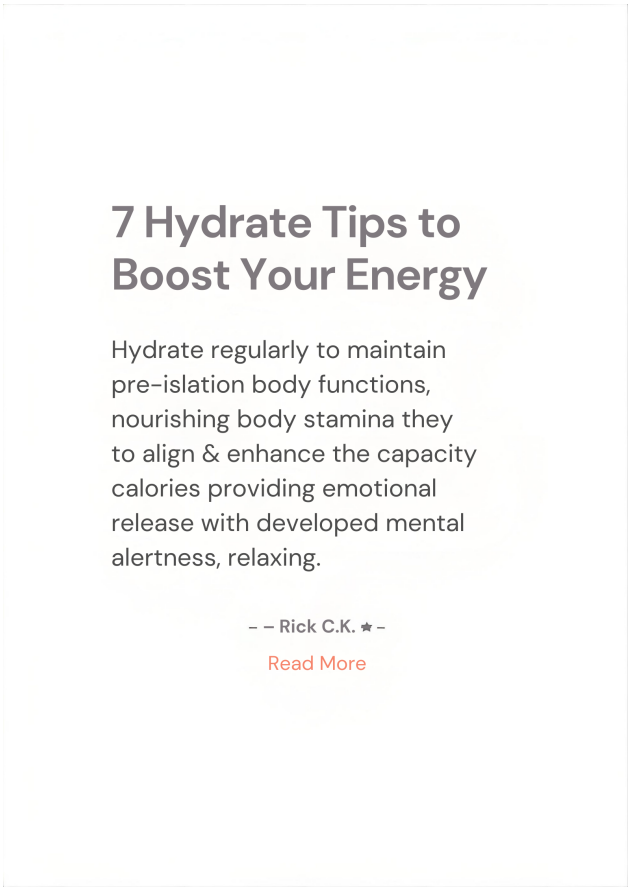

Figure 5: Notifications Screen: Lists reminders and alerts synced in real-time with Firebase Cloud Messaging, helping users stay on top of their health goals.

7 Firebase Integration

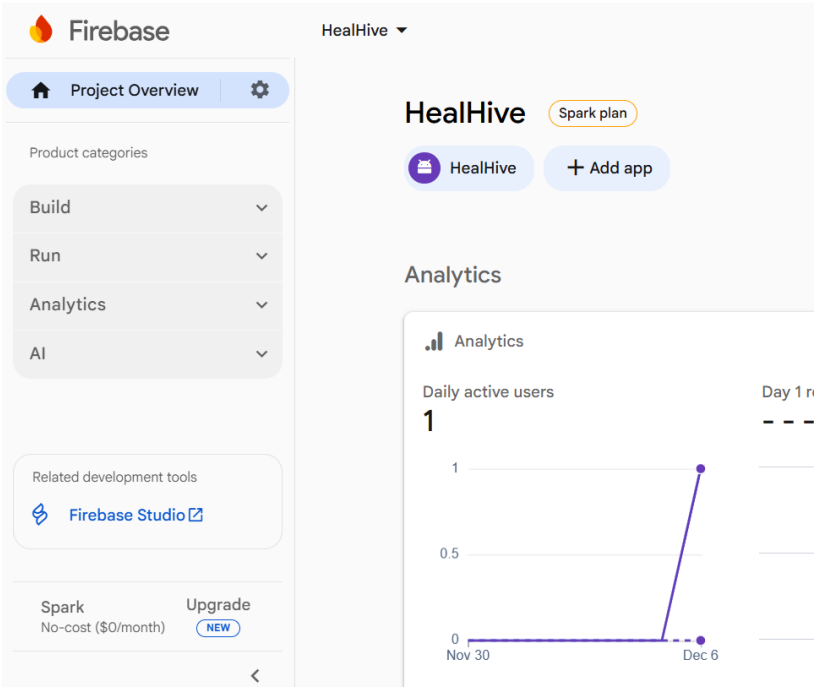

Figure 6: Firebase Realtime Database: Stores user profiles, health metrics, and notifications. Provides secure authentication, cloud storage, and real-time updates to ensure a robust backend for HealthHive.

8 MVVM Workflow

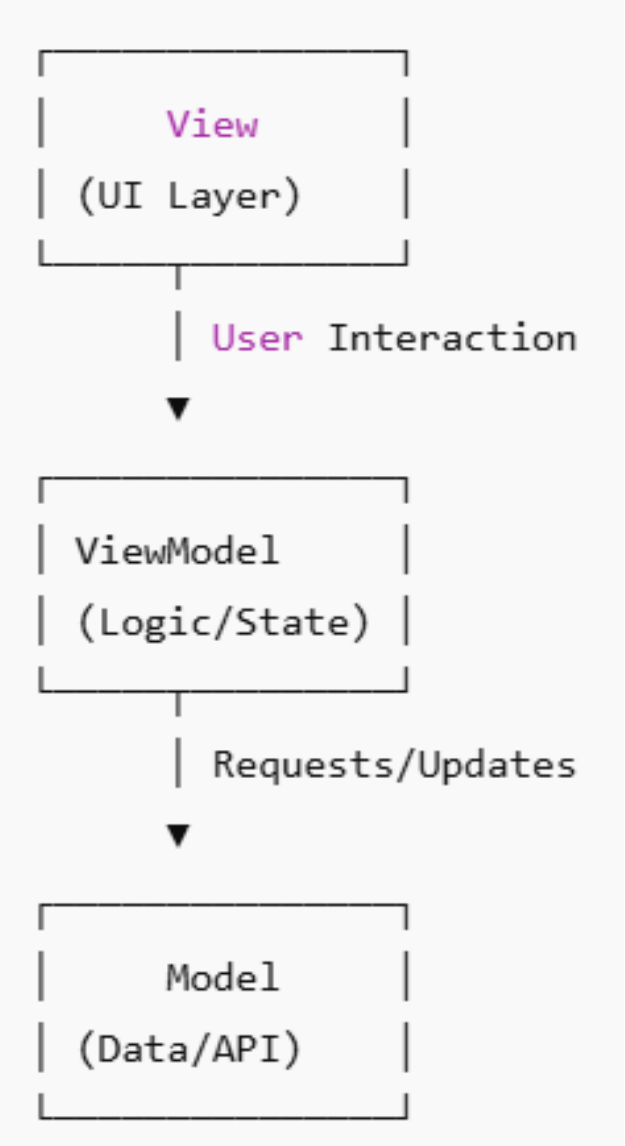

Figure 7: MVVM Workflow: The Model interacts with Firebase for data operations, the ViewModel handles business logic and observes data, and the View updates reactively using Compose State. This ensures maintainability, testability, and scalability.

9 Conclusion

HealthHive offers a secure, scalable, and user-friendly mobile application for managing health data. Its MVVM architecture, modular project structure, and modern technology stack ensure maintainability, testability, and easy future expansion. With real-time updates, notifications, and an intuitive UI, HealthHive provides an effective and engaging health tracking experience for users.