FitAI

Software Requirements Specification

CSE227
ADVANCED ANDROID APP DEVELOPMENT

Srikant Kumar Aich
1211192

Prepared for
Continuous Assessment 3
Spring 2025

Table of Contents

1. Introduction

This section provides an overview of the FitAI mobile application, outlining the purpose, scope, and structure of this Software Requirements Specification (SRS). The goal is to present all necessary information to understand what the software does, whom it serves, and how it should behave.

1.1 Purpose

The purpose of this document is to detail the functional and non-functional requirements of the ***FitAI Android application****. It is intended for use by the development team, project managers, testers, and stakeholders to understand the design, behavior, and scope of the application.*
The primary audience includes:

- *Android developers implementing the application.*
- *Quality assurance/test engineers verifying functionality.*
- *UI/UX designers designing user flows and layout.*
- *Project supervisors reviewing deliverables.*
- *End-user documentation writers*

1.2 Scope

FitAI *is an AI-powered Android fitness application designed to provide users with personalized workout suggestions based on selected body parts or goals (e.g., cardio) and preferred duration.*
The application features:

- *An interactive interface using modern Android design (Jetpack Compose)*
- *Real-time step tracking for physical activity monitoring*
- *AI-generated exercise plans using the Gemini API*
- *A Gemini-powered search assistant*
- *login/signup functionality using Firebase.*
- *A single-page user interface with structured exercise recommendations*

the application does ***not*** *include:*

- *Diet planning or nutrition tracking.*
- *Third-party fitness device integration (e.g., Fitbit, Apple Watch)*
- *Cloud backup or multi-device sync*
- *Social networking or progress sharing*

This app aims to make intelligent fitness planning accessible without requiring complex equipment or human coaching.

1.3 Definitions, Acronyms, and Abbreviations

Term / Acronym	*Description*
SRS	*Software Requirements Specification*
AI	*Artificial Intelligence*
UI	*User Interface*
UX	*User Experience*
API	*Application Programming Interface*
JSON	*JavaScript Object Notation*
MVVM	*Model-View-ViewModel (Android architecture pattern)*
Jetpack Compose	*Android's modern UI toolkit*
Firebase	*Android storage mechanism for key-value data*
Gemini	*Google's AI platform/API used in the app*
Step Tracker	*Module that counts steps using phone sensors*

1.4 References

- *Android Developer Documentation - https://developer.android.com*
- *Jetpack Compose - https://developer.android.com/jetpack/compose*
- *Google Gemini API - Official documentation (internal/external as per availability)*
- *IEEE 830-1998 - Recommended Practice for Software Requirements Specifications*
- *Kotlin Language Guide - https://kotlinlang.org/docs*

2. General Description

This section provides a high-level overview of the FitAI application, focusing on the product perspective, user characteristics, and operational environment.

2.1 Product Perspective

***FitAI** is an independent Android application developed from scratch using Kotlin and Jetpack Compose. It integrates Google's Gemini AI API for generating personalized workout routines. The app follows the MVVM architecture to ensure separation of concerns and scalability.*
Key components include:

- ***Frontend UI (Jetpack Compose):** For seamless and reactive user experience.*
- ***Gemini Integration Module:** Handles AI-powered workout generation based on user input.*
- ***Step Tracker Module:** Uses the device's sensors to count steps.*
- ***Firebase:** For storing user credentials.*
- ***AI Workout Planner:** Generates a structured plan with headings and subheadings based on body part and duration.*

The app does not rely on external databases or cloud services, making it lightweight and privacy focused.

2.2 Product Functions

The primary functions of the FitAI app are:

- *Allow users to **sign up** and **log in** with data storage.*
- *Let users choose a **body part or workout type** (e.g., chest, cardio) and **duration** (e.g., 10 minutes).*
- *Generate **custom workout plans** using Gemini API based on input.*
- *Display workout plans with a clean and structured layout (heading + subheading).*
- *Count user **steps in real time** using phone sensors.*
- *Allow users to search for workouts or queries using **Gemini-powered assistant**.*
- *Show all features in a modern single-screen layout.*

2.3 User Characteristics

- ***Target Audience:** Fitness enthusiasts, beginners, and casual users aged 16–50.*
- ***Technical Skill Level:** Minimal; the app is designed to be user-friendly and intuitive.*
- ***Accessibility:** Colorful visuals, clear fonts, and easy navigation.*
- ***Expected Behavior:** Users interact via dropdowns, buttons, and search prompts. They are expected to provide basic input like workout goals and time.*

2.4 General Constraints

- *The app is **limited to Android OS (API 26 and above)**.*
- *Requires **internet access** for Gemini API features.*
- *Step tracking accuracy depends on **device hardware sensors**.*
- *No **backend server** or **multi-user sync** functionality is implemented.*
- *Gemini API usage is limited based on **quota and rate limits**.*

2.5 Assumptions and Dependencies

- *Users have **working internet access** for AI features.*
- *Gemini API key is valid and properly configured.*
- *Users give **permission for activity recognition** for step tracking.*
- *The device supports basic motion sensors.*
- *The Android system supports **Jetpack Compose** (newer versions).*

3. Specific Requirements

*This section outlines all functional and non-functional requirements needed to build and validate the FitAI application. These requirements are written to be **clear**, **testable**, **traceable**, and **prioritized**.*

3.1.1 User Interfaces

*The app uses a **Jetpack Compose UI** with a single-screen layout containing:*

- *Sign-up and login forms*
- *Dropdowns for body part and duration*
- *Gemini-powered AI prompt box*
- *Step tracker card*
- *AI-generated workout display (heading + subheading)*

3.1.2 Hardware Interfaces

- *Use **an accelerometer** and/or **step detector sensor** for step counting.*
- *Requires devices with **Activity Recognition hardware support**.*

3.1.3 Software Interfaces

- *Gemini API integration via HTTP requests using a valid API key.*
- *Use **Firebase** for secure storage of user credentials.*
- *Requires Android 8.0 (API level 26) or higher.*

3.1.4 Communications Interfaces

***Internet connection** required for:*

- *Fetching workout plans from Gemini AI*
- *Searching fitness-related queries via AI*
- *No server-client model; app is standalone.*

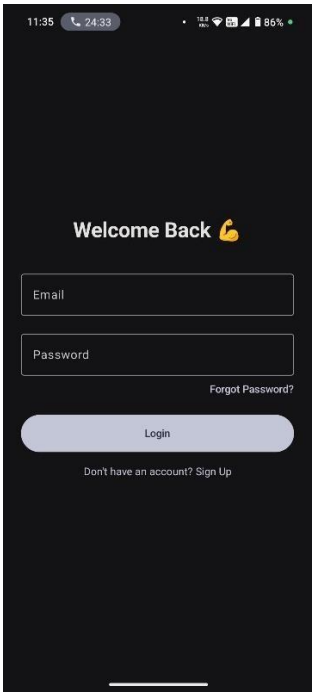

Figure 1

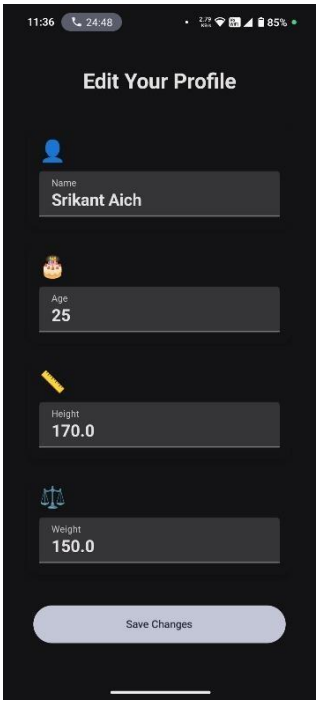

Figure 2

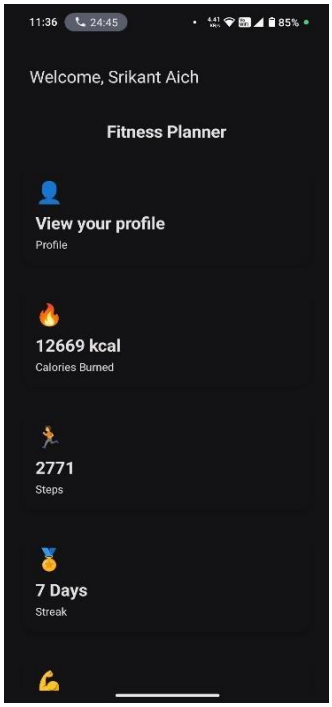

Figure 3

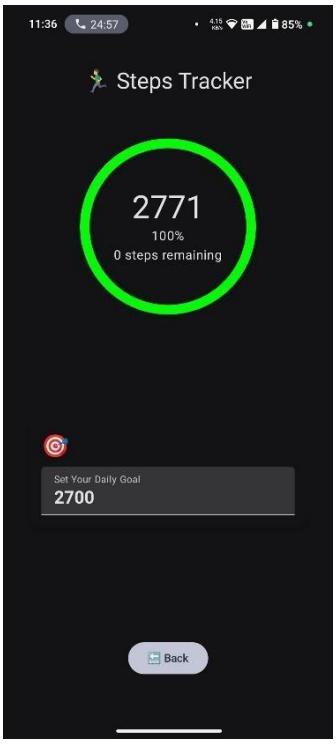

Figure 4

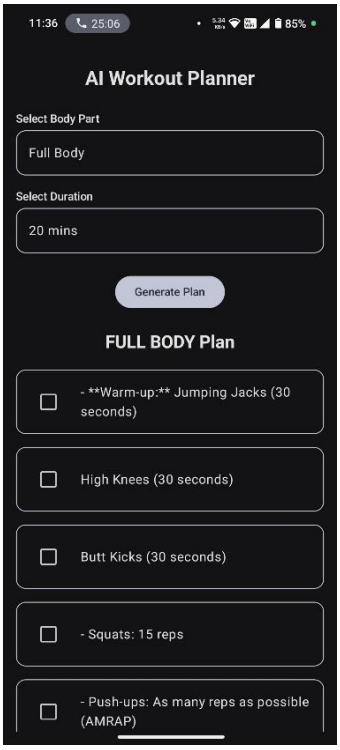

Figure 5

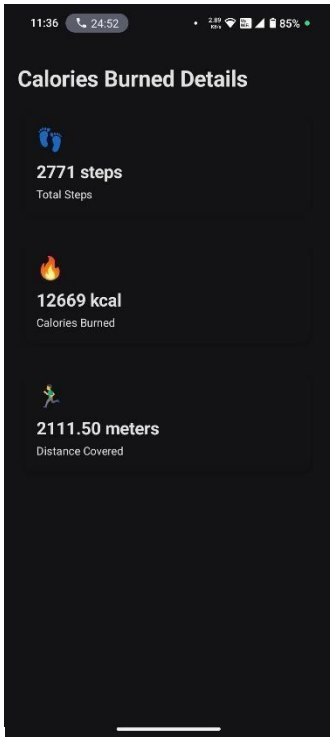

Figure 6

3.2 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

3.2.1 User Authentication

3.2.1.1 Introduction

Allow users to sign up, store credentials, and log in securely.

3.2.1.2 Input

- *Username*
- *Password*

3.2.1.3 Processing

- *Check for existing users during signup.*
- *Match credentials on login using Firebase.*

3.2.1.4 Output

- *Login success/failure messages*
- *Navigate to dashboard upon success.*

3.2.1.5 Error Handling

- *Show "Invalid credentials" toast if login fails.*
- *Show "User already exists" if duplicate signup.*

3.2.2 AI Workout Generator

3.2.2.1 Introduction

Generates personalized workouts using Gemini AI.

3.2.2.2 Input

- *Dropdown selection of body part or workout type*
- *Dropdown selection of workout duration*

3.2.2.3 Processing

- *Prompt Gemini with structured query like:*
  "Generate a 15-minute back workout with warm-up and main exercises."

3.2.2.4 Output

- *Structured workout with a **heading** and **subheading***
- *The display result is a styled card.*

3.2.2.5 Error Handling

- *Display fallback error if Gemini API fails or times out.*

3.2.3 Step Tracker

3.2.3.1 Introduction

Track and display the user's steps in real time.

3.2.3.2 Input

- *Device's motion sensors*

3.2.3.3 Processing

- *Listen to step events and update the count.*

3.2.3.4 Output

- *Realtime step count shown in a colorful card.*

3.2.3.5 Error Handling

- *Show "Step tracking not available" if sensors aren't found.*

4. Analysis Models

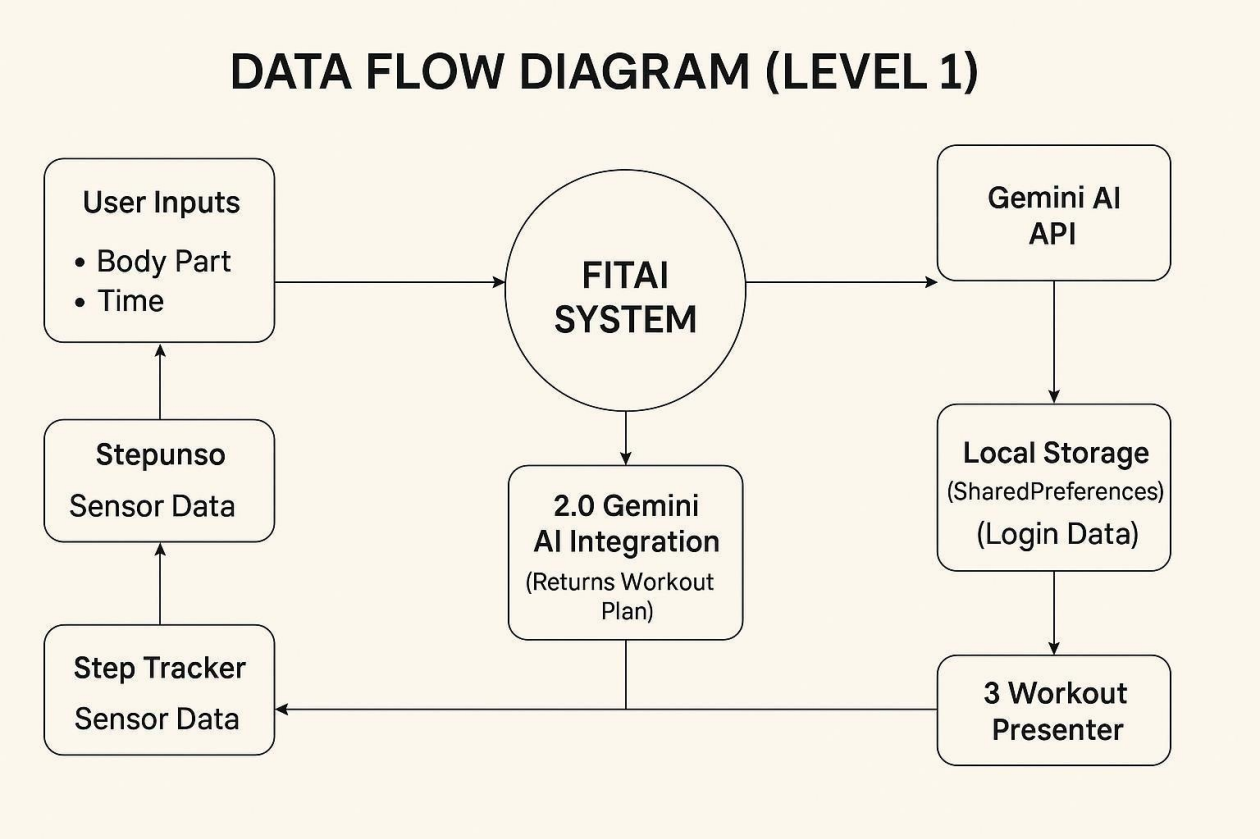

Figure 7

*The user interface is designed using **Jetpack Compose**, with a clean and modern layout that is intuitive and responsive. Key UI elements include:*

- ***Login/Sign-up Screen:***
  - *Text fields for username and password.*
  - *Buttons to submit and switch between login/sign-up modes.*
  - *Error messages for invalid or duplicate credentials.*
- ***Main Dashboard:***
- *Step tracker card showing current step count.*
- *Gemini AI workout generator section:*
    - *Dropdowns to select body part and duration.*
    - *Prompt and response box styled for clarity.*
  - *Output area displays structured workout with:*
    - ***Heading** (e.g., "Back Workout")*
    - ***Subheading** (e.g., warm-up + exercises)*
- ***Colors and Themes:***
  - *Colorful cards inspired by fitness dashboards.*
  - *Smooth animations using Jetpack Compose's built-in transitions.*

Hardware Interfaces

- ***Sensors Used:***
  - ***Step Counter*** *and/or* ***Step Detector***
  - *Device must support* ***Activity Recognition*** *for accurate step tracking.*

4.3 Software Interfaces

- ***Android Platform:***
  - *Minimum SDK version:* ***API 26 (Android 8.0)***
  - *Target SDK:* ***API 34 (Android 14)***
- ***APIs & Libraries:***
  - ***Gemini AI API*** *(Google Generative AI client)*
  - ***Firebase*** *for credential storage*
  - ***Jetpack Compose*** *for UI development.*
  - ***Kotlin*** *as the programming language*

4.4 Communication Interfaces

- ***Internet Connectivity:***
  - *Required to interact with Gemini AI and fetch workout suggestions.*
  - *Apps should gracefully handle lack of connection (show error prompts).*
- ***No backend/server dependency:***
  - *All logic and storage are client-side (offline-first except AI).*

5. VS CODE SCREENSHOTS

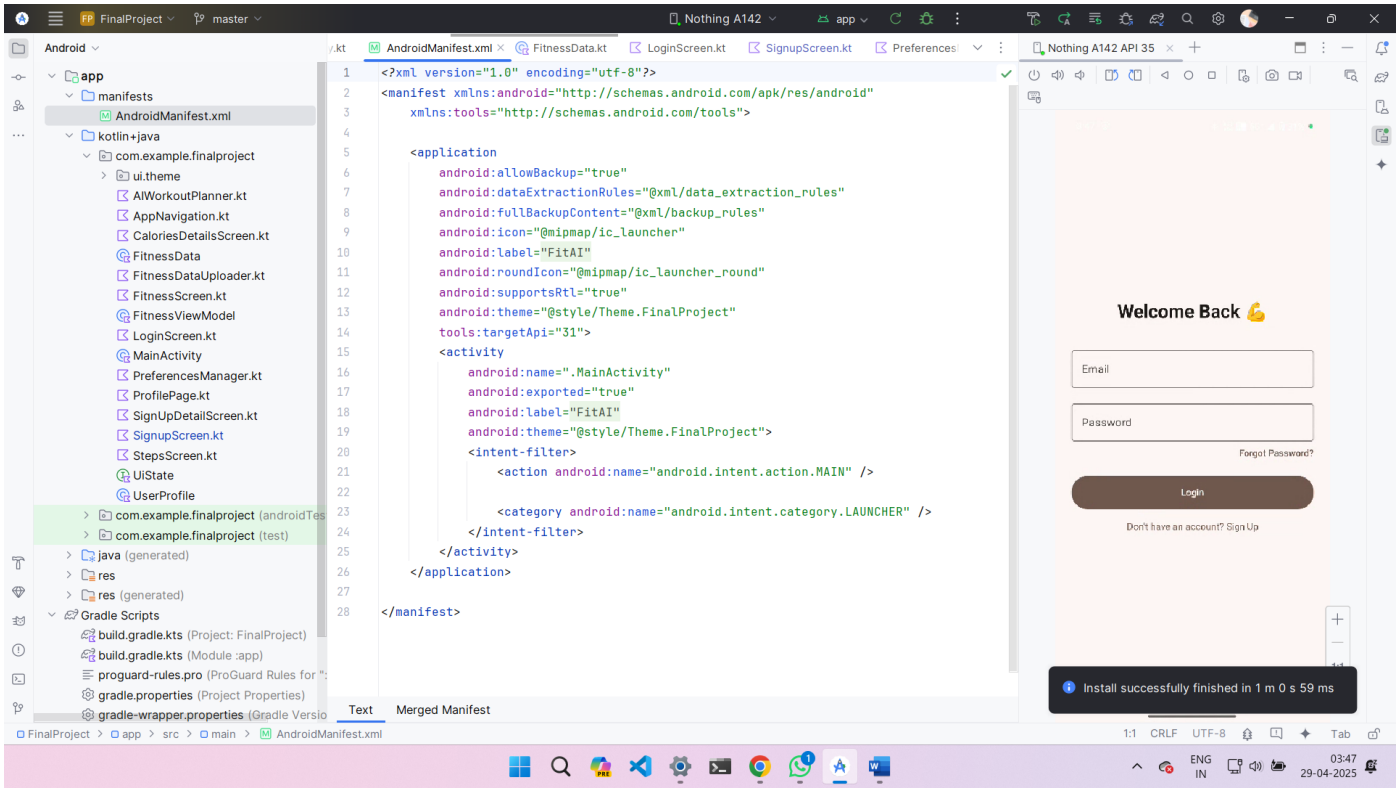

Figure 8 AndroidManifest.xml

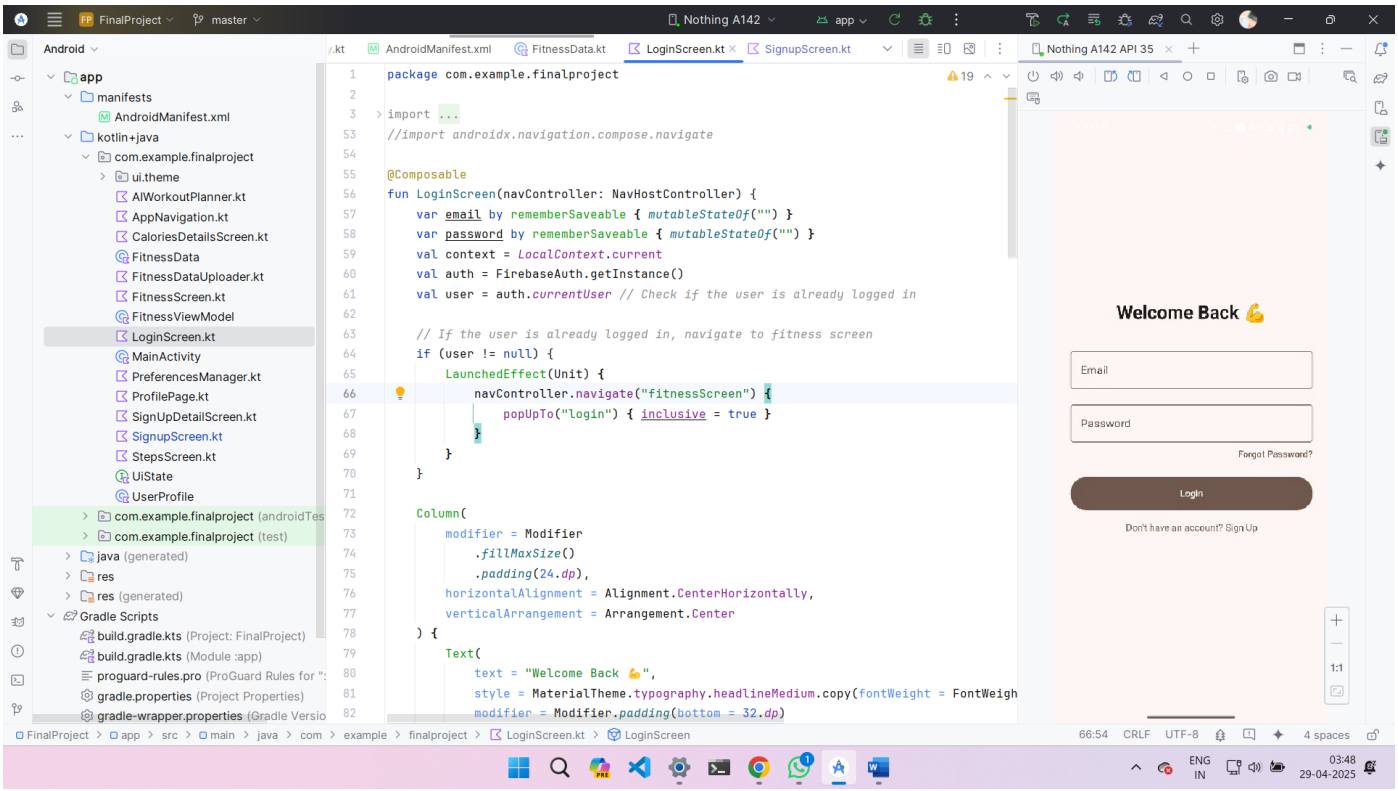

Figure 9 LoginScreen.kt

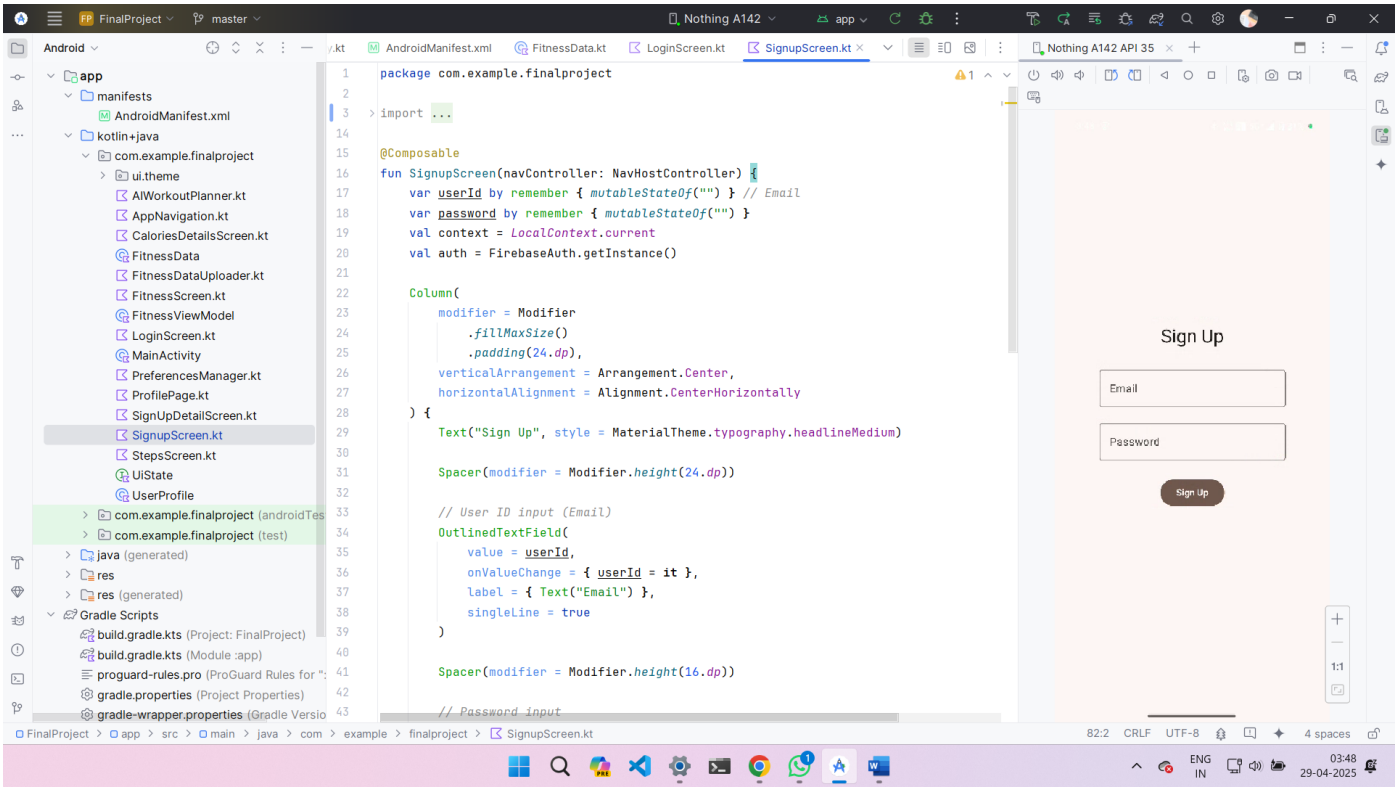

Figure 10 SignUpScreen.kt

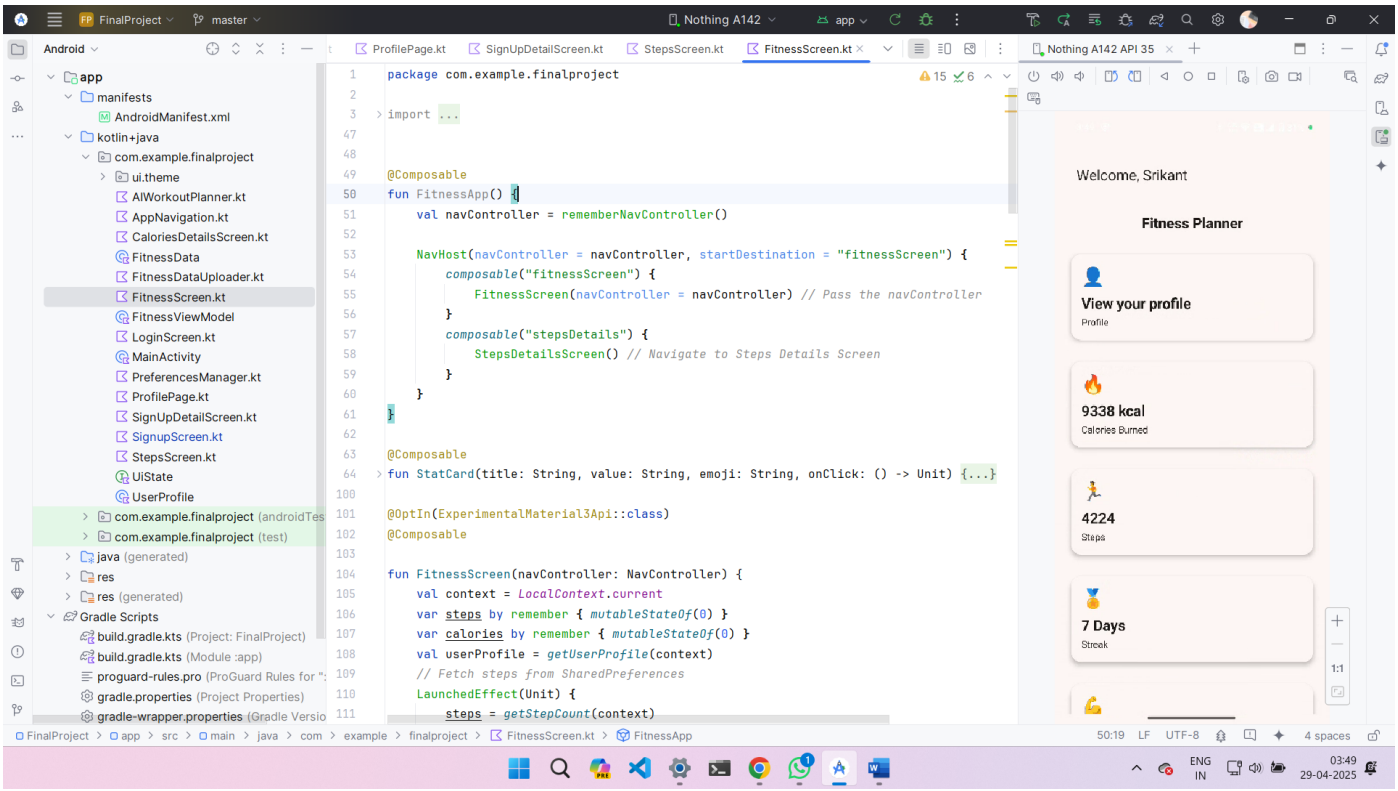

Figure 11 DashBoard/Fitnessscreen.kt

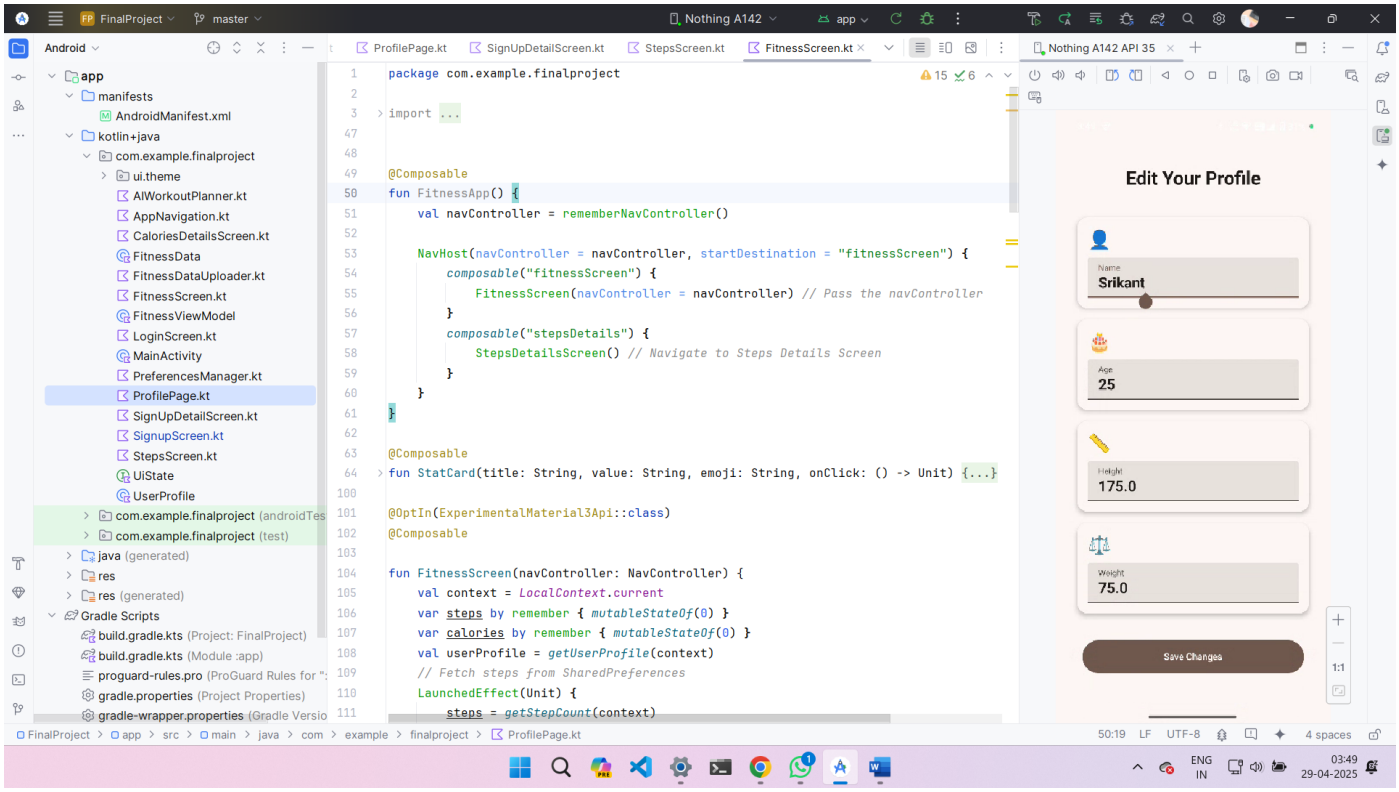

Figure 12 ProfilePage.kt

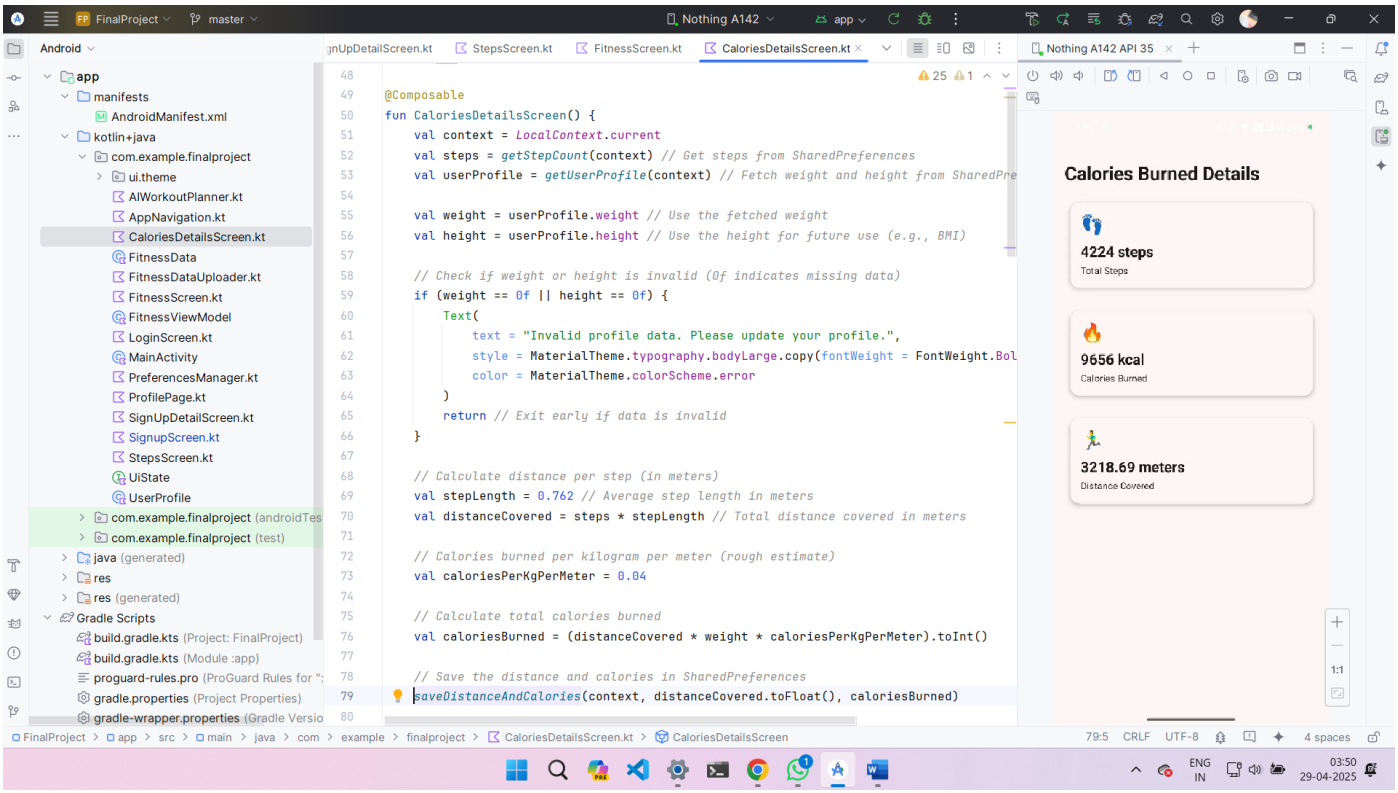

Figure 13 CaloriesDetails.kt

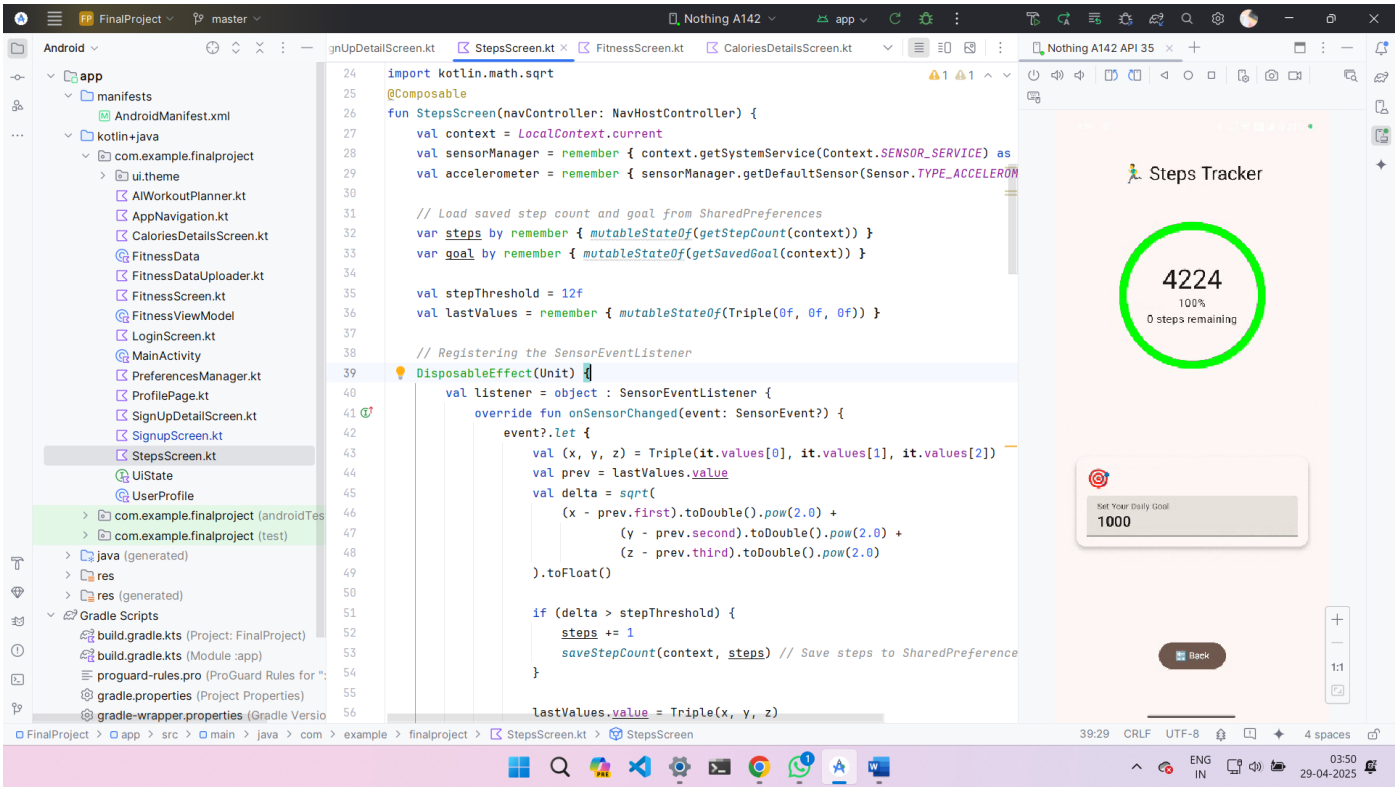

Figure 14 StepsScreen.kt

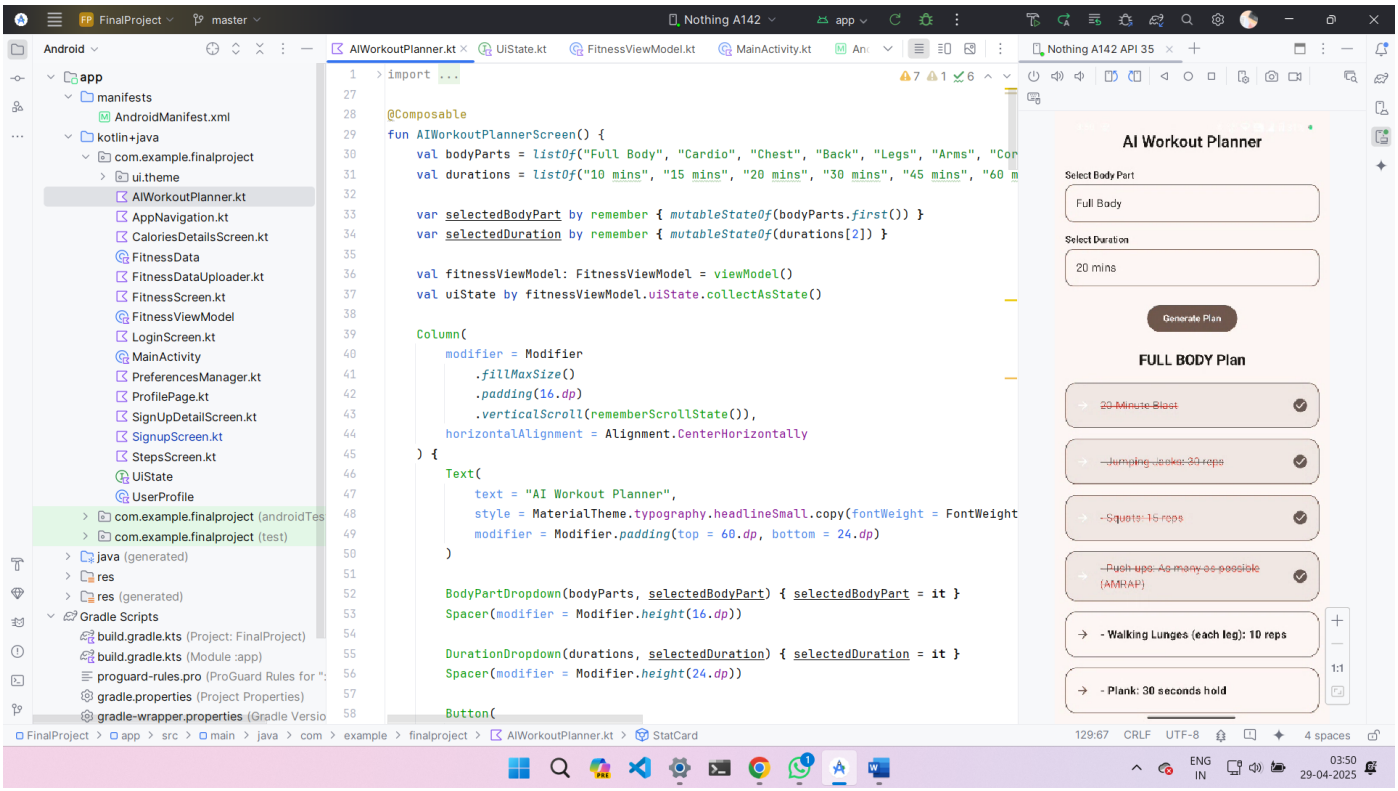

Figure 15 AIWorkoutPlanner.kt