## FitnessTrack Report

**Mohammed Muzzamill Miah**  
**Student ID: P2716217**  
**BSc (Hons) Software Engineering**  
**Supervisor: Bahareh Daneshvar**  
**Submission Date: 23/05/2025**

**Abstract**

FitnessTrack is a mobile fitness and health tracking application designed to provide users with personalised insights into their daily activity, nutrition, and overall wellness. The app offers goal setting, real-time progress tracking, and AI-powered motivational support to help individuals stay engaged and consistent in their health journeys. By integrating modern tools such as Firebase, Kotlin, and a step counter, FitnessTrack delivers a seamless, user-friendly experience that adapts to individual needs and supports long-term fitness goals.

Contents

[FitnessTrack Report 1](#_Toc765977812)

[Acknowledgement 1](#_Toc1566776956)

[Background and Context 2](#_Toc1106199016)

[Aims and Objectives 3](#_Toc235047243)

[Motivation for the Project 3](#_Toc721355858)

[Personal Development Objectives 3](#_Toc1046828384)

[2.1 Literature Review 4](#_Toc755763096)

[Evaluation of Current Fitness Applications and Their Impact on User Engagement 5](#_Toc871418868)

[Technology and Tools Overview 7](#_Toc86957562)

[Summary of Research Findings 8](#_Toc1013696408)

[System Overview 8](#_Toc1672137732)

[3.2 Architecture Diagrams 9](#_Toc2110185114)

[3.3 Major Components 21](#_Toc2033083369)

[4.1 Methodology Used 22](#_Toc137851909)

[4,2 Planning and Requirements Gathering 23](#_Toc7154575)

[4.3 Implementation Process 25](#_Toc446287307)

[4.4 Testing and Validation 26](#_Toc827709720)

[5.1 What Worked Well 27](#_Toc2126507923)

[5.2 Challenges and Issues Faced 28](#_Toc377763489)

[5.3 Lessons Learned 29](#_Toc1902843915)

[5.4 Retrospective: What Would Be Done Differently 30](#_Toc2082330604)

[5.5 Evaluation of Tools and Technologies 31](#_Toc16109716)

[5.6 Technical Appraisal of Final Product 32](#_Toc283190961)

[6. Conclusion 32](#_Toc1411679167)

[6.1 Recap of Aims and Objectives 32](#_Toc1300329647)

[6.2 Summary of Work Done 32](#_Toc1612315294)

[6.3 Key Findings 33](#_Toc527496078)

[6.4 Final Thoughts and Future Work 33](#_Toc1003274061)

[Screenshots 34](#_Toc1861069261)

[References 41](#_Toc395723664)

## Acknowledgement

﷽

"In the name of Allah, the Most Gracious, the Most Merciful"

First and foremost, I would like to praise Allah (S.W.T), the Most Gracious and the Most Merciful, for making it possible for me to complete this dissertation. I am immensely grateful for the strength, knowledge, and opportunities He has bestowed upon me, and for the constant reminder that He is always by my side.

To my beloved father: Thank you, Abba, for always being there for me, for taking me to school in the early mornings, encouraging me to study, and offering me your wisdom and unwavering support. Your presence in my life has been a blessing, and I am forever grateful for the discipline, care, and love you have given me throughout every step of this journey.

To my mother: I owe you the world. You have been my rock since the very beginning. From your late-night calls checking on me at university to your constant prayers and encouragement, you have always kept me going. I hope this dissertation makes you proud and brings a smile to your face.

Thank you to Afa for being like a second mother to me. I have always looked up to you and always will. You’re the first person I turn to for support. From picking me up from school in Year 1 when I was small, to putting me to sleep at night—when I wouldn’t let you sleep until you prayed the First Kalimah with me—to dropping me off at university and offering advice that has helped guide my life, you have always been there. You are my big sister, my right hand, and I’m endlessly grateful for everything you’ve done for me.

I would like to dedicate this dissertation to my late grandfather, Abdul Hamid, and my late grandmother, Anwara Bibi.  
 Nana, I hope you are smiling down on me. I kept the promise I made to you in September 2018—to complete my education and earn my degree. You always used to say, *“Gain an abundance of knowledge, as everything else can be taken away from you except that.”*  
 Nani, I lost you on the 10th of August, 2024. Thank you for always telling me to *“Study as much as your heart desires, dear one.”* Your faith in me and the reassurance you provided gave me strength when I needed it most.  
 May Allah (S.W.T) grant you both the highest ranks in Jannatul Firdous. Ameen.

At last, I want to express my deepest love and gratitude to my beautiful mother.  
 Thank you, Amma, for always pushing me beyond my limits, reminding me never to give up, and giving me the opportunity to strive for a better future. You are my best friend, my mother, my everything. I cannot thank you enough—for without you, I would not be where I am today, completing this dissertation.

Thank you, Amma and Abba.  
 *“Study hard, little Shubon—you’ll go far in life.”*

**قُمِ اللَّيْلَ إِلَّا قَلِيلًا**  
 *Qumi Al-Layla Illa Qaleelan*  
 *"Stand [for prayer] the night, except a little."*  
 — *Surah Al-Muzzammil [73:2]*

## Background and Context

In today’s rapidly advancing technological era, the integration of health and fitness into digital applications has become increasingly prevalent. With the growing reliance on smartphones across all age groups, the demand for a user-friendly fitness application is more evident than ever. Despite the wide variety of fitness apps available on the market, many users, particularly beginners, continue to encounter difficulties in tracking their goals, monitoring nutritional intake, and navigating overly complex interfaces. This issue has been highlighted in numerous online blogs and forums, where users have expressed frustration with existing solutions. Many noted the challenge of tracking specific fitness metrics, such as calorie intake, workout progress, and nutrient consumption, within a single, cohesive platform. These overly complicated applications often lead to decreased motivation and consistency, ultimately hindering users' progress.

## Aims and Objectives

The idea for FitnessTrack emerges as a response to common frustrations faced by users of existing fitness applications. The goal of the project is to develop a comprehensive and intuitive fitness app that helps users achieve their health and fitness goals through accurate tracking and personalized guidance. Core objectives include enabling users to log and monitor daily calorie intake to promote nutritional awareness, and providing a workout tracker that allows users to set fitness goals, record exercise routines, and view progress over time. The application also offers personalized meal recommendations based on dietary preferences and goals, encouraging balanced nutrition. Additional features include customizable user profiles, a detailed settings panel, and essential authentication tools such as sign-up and login to ensure secure and personalized access. A central home screen serves as the main dashboard, offering quick and easy access to all key features and updates in one convenient location.

## Motivation for the Project

The motivation behind this project, stems from my personal experiences and challenges I faced when starting my own fitness journey. Like many beginners, I was driven by a desire to improve my health and overall wellbeing however I quickly encountered many obstacles, relating to consistency, information overload and progress tracking. I found it difficult to determine whether I was making meaningful progress and often felt overwhelmed by conflicting advice and fragmented tools. These difficulties inspired me to create a solution that just doesn’t help my struggles but also offers a supportive and structured approach to fitness for others facing similar challenges.

FitnessTrack is designed to simplify the process, deliver reliable and personalised guidance, and empowers users to take control of their health with clarity and confidence.

## Personal Development Objectives

In addition to solving a real world problem, this project has offered me a meaningful opportunity for both professional and personal growth. One of me key development goals was to gain hands on experience in building a fully functional fitness application, which aligns with my long term ambition a skilled to explore the connection between software development and user centred health tools, giving me valuable insight into how digital solutions can genuinely improve people’s well being through thoughtful design and practical functionality.

From a technical perspective, the project has helped me strengthen my skills in both Java and Kotlin, two foundational languages for Android development. While Kotlin offers a more modern, concise, and safer coding, having a strong grasp of Java remains important, especially when working with existing Android libraries or legacy systems. By using both within Android Studio, I’ve been able to create scalable, efficient code that is easier to read, maintain, and extend.

A core feature of the app is secure user authentication, which I implemented using Firebase Authentication. This handles critical functions such as sign-up, login, password reset, and session control. Using Firebase not only simplified backend management but also freed up time to focus on crafting a better front-end experience. It mirrors the type of infrastructure used in real-world applications, where secure and smooth onboarding is essential for user trust and retention.

The project also pushed me to improve my UI/UX design abilities. I focused on creating responsive layouts, simple navigation, and clear interactions. This was especially important given the app’s goal: to help users stay engaged and motivated in their fitness routines. I worked to ensure the design was approachable for beginners while still meeting the expectations of more experienced users.

Ultimately, this project has been more than just a coding challenge. It reflects my passion for fitness and my commitment to building meaningful, user-driven technology. Working with tools like Kotlin, Java, Android Studio, and Firebase has not only prepared me for a career in tech but also allowed me to create something that can genuinely help others take control of their health and fitness in a practical, accessible way.

## 2.1 Literature Review

This literature review explores the current landscape of mobile fitness tracking applications, focusing on the design considerations and technologies that inform the development of FitnessTrack, a health and wellness Android applications designed to help users monitor workouts, track nutrition, and achieve their fitness goals. This review evaluates leading applications such as Nike Training Club, Fitbit and Strava to identify the limitations and reoccurring user feedback that FitnessTrack aims to address. These include complex interfaces, lack of personalisation and limited guidance for beginners (Stassen, Garcia and Mohan, 2022).

In addition to solving a real-world problem, this project offers a meaningful opportunity for both professional and personal growth. One of my key development goals is to gain hands-on experience in building a fully functional fitness application, which aligns with my long-term ambition to explore the connection between software development and user-centered health tools. This project provides valuable insight into how digital solutions can genuinely improve people's well-being through thoughtful design and practical functionality.

From a technical perspective, the project strengthens my skills in both Java and Kotlin—two foundational languages for Android development. While Kotlin offers a more modern, concise, and safer coding experience, maintaining a strong grasp of Java remains important, especially when working with existing Android libraries or legacy systems. By using both languages within Android Studio, I can write scalable, efficient code that is easier to read, maintain, and extend.

A core feature of the app is secure user authentication, which I implement using Firebase Authentication. This handles critical functions such as sign-up, login, password reset, and session control. Using Firebase not only simplifies backend management but also allows me to focus more on enhancing the front-end experience. It reflects the kind of infrastructure used in real-world applications, where secure and seamless onboarding is essential for user trust and retention.

The project also challenges me to improve my UI/UX design abilities. I focus on creating responsive layouts, intuitive navigation, and clear interactions. This is especially important given the app’s goal: to keep users engaged and motivated in their fitness routines. I work to ensure the design is approachable for beginners while still meeting the needs of more experienced users.

Ultimately, this project is more than just a coding challenge. It reflects my passion for fitness and my commitment to building meaningful, user-driven technology. Working with tools like Kotlin, Java, Android Studio, and Firebase not only prepares me for a career in tech but also allows me to create something that can truly help others take control of their health and fitness in a practical, accessible way.

## Evaluation of Current Fitness Applications and Their Impact on User Engagement

A wide range of mobile fitness applications are currently available, offering diverse features to support user health. Among the most popular are Strava, Nike Training Club, and MyFitnessPal. Each of these applications brings unique functionalities to the fitness technology space. Common features include workout tracking, personalized workout plans (often behind a paywall), and social connectivity. Many of these apps also support integration with wearables and health data platforms, enhancing usability by syncing data across multiple devices.

**Strava: Fitness Tracking Through Community Engagement**

Strava is a leading platform, particularly favoured by runners and cyclists. It differentiates itself through strong social networking components, allowing users to share activities, follow one another, and participate in challenges. Core features of Strava include:

• Detailed tracking of distance, pace, elevation, and route mapping  
• Gamified elements such as leaderboards and achievement badges  
• A community-focused feed that encourages user interaction

Research has shown the platform’s effectiveness in boosting motivation and long-term engagement through its social infrastructure. For example, Franken, R., Bekhuis, H., & Tolsma, J. (2023) found that Strava’s community-driven approach significantly influences user consistency and goal achievement, owing to the natural competitive drive-in humans.

Strava's community-driven features significantly influence user consistency and goal achievement by fostering accountability and shared progress. A study by Franken, Bekhuis, and Tolsma (2023) analysed data from Strava clubs and found that social interactions—such as receiving 'kudos' and observing peers' activities—encourage users to increase their running frequency. These social feedback mechanisms create a sense of accountability and motivation, leading to more consistent exercise habits among users.

**Nike Training Club: Personalized Workout Guidance**

Nike Training Club (NTC) provides a more structured and guided fitness experience through professionally designed workouts and adaptive plans. The app tailors its offerings to users’ individual fitness levels and goals, including:

• Customizable training programs with integrated progress tracking  
• On-demand workouts with video and audio coaching  
• Nutritional tips and holistic wellness content

NTC’s personalization capabilities have been linked to improved user satisfaction and program effectiveness. Its use of machine learning to recommend workouts enhances relevance and engagement by tailoring sessions to user data and preferences. This aligns with findings from Evangelist Apps (2024), which highlight how machine learning in fitness apps improves personalization and keeps users motivated through more targeted content and feedback.

**Comparative Analysis of Existing Applications**

**Strengths**  
• Strava’s social features create a community-driven environment that motivates users through competition, peer engagement, and friendly competition.  
• NTC’s tailored fitness plans and guided workouts provide structure and expertise, which cater to users with varying levels of fitness.  
• Most applications support integration with wearables and other platforms, allowing for centralised tracking of health metrics.

**Weaknesses**  
• Limited cross-functionality among many apps; for example, a user may need separate apps for workout tracking, nutrition, and goal setting.  
• Inconsistent personalisation, with some apps offering generic workout recommendations that do not adapt to user progress or feedback.  
• User experience challenges, including complex navigation and cluttered interfaces.

**Identifies Gaps in the Market**  
Despite their popularity and functionality, current fitness applications often lack full integration of features on a single, user-friendly platform. Most fail to offer a seamless blend of fitness tracking, personalised workout planning/goal setting, and nutritional guidance within one app. This fragmentation can lead to reduced engagement and a disjointed user experience, as users may find it inconvenient and time-consuming to switch between different applications for different fitness needs.

Moreover, while apps like NTC offer guided plans, more data-driven and adaptive feedback mechanisms are needed to maintain user motivation and provide real-time suggestions. Without personalised, evolving guidance, users may lose interest or fail to see consistent progress, which affects their overall fitness journey.

These gaps highlight the opportunity for applications like FitnessTrack to unify essential features and deliver a comprehensive solution for fitness beginners and enthusiasts, while being free. FitnessTrack is designed to bring together core functionalities such as workout tracking, nutrition logging, and personalised goal setting into one intuitive platform. Its user-friendly interface and integrated approach address the limitations of current apps, offering a more cohesive and motivating experience for users across all fitness levels.

Technology and Tools Overview  
The development of mobile fitness applications such as FitnessTrack relies heavily on a robust combination of front-end and back-end technologies. These tools and frameworks are chosen to ensure a smooth user experience, real-time data interactions, secure authentication, and reliable performance on Android platforms.

Android development for FitnessTrack is carried out using Android Studio, the official integrated development environment (IDE) supported by Google. Android Studio offers a comprehensive suite of tools for building, testing, and debugging mobile applications. It supports both Java and Kotlin, with Kotlin being the preferred language due to its concise syntax, null safety features, and full compatibility with Java-based libraries (Obregon, 2021). Kotlin’s modern programming approach not only enhances developer productivity but also contributes to more readable and maintainable codebases, which is crucial for long-term application sustainability.

For backend services, Firebase is used due to its seamless integration with Android Studio and Kotlin. Firebase Authentication provides a secure and straightforward method for handling user login and registration, allowing developers to implement email-password or third-party authentication (e.g., Google Sign-In) without managing custom backends (Firebase, 2023a). In parallel, Firebase Cloud Firestore serves as the primary NoSQL cloud database, offering real-time syncing, scalable data storage, and offline support—essential features for a fitness app where users often interact with the app during workouts without guaranteed internet access (Firebase, 2023a).

Additionally, Firebase Realtime Database may be used in scenarios requiring synchronised updates across users or devices, though Firestore is often preferred for more complex data structures and scalability. These technologies form the foundation for FitnessTrack’s user data management, including tracking nutrition, exercise routines, and goal progress.

In terms of wearable device integration, APIs such as Google Fit are used to collect and analyse fitness-related data from various sources like smartwatches and fitness bands. This allows the app to display comprehensive user insights, including steps, heart rate, and calories burned. Bluetooth Low Energy (BLE) protocols may also be implemented to sync real-time health metrics from external fitness hardware, further enhancing the app’s data ecosystem (Müller et al., 2022).

To promote user engagement and support long-term motivation, FitnessTrack delivers daily tips and motivational quotes. These lightweight features are integrated directly into the user interface to inspire and guide users without introducing unnecessary complexity. Instead of relying on complex automation, the application focuses on providing meaningful, practical content through a clean and accessible design.

## Summary of Research Findings

Research highlights the importance of user engagement strategies in the success of mobile fitness applications. Features such as social connectivity and gamification, particularly in apps like Strava, have proven effective in encouraging long-term use and commitment. Leaderboards, achievement badges, community challenges, and social motivation have been shown to significantly boost user consistency and accountability. Similarly, the availability of personalised experiences—such as those offered by Nike Training Club (NTC) through custom workout plans—has been found to enhance user satisfaction and improve adherence to fitness routines over time.

The impact of AI and personalisation is another key area of advancement in fitness app development. AI-driven recommendations allow applications to dynamically adapt to users’ goals, fitness levels, and progress, resulting in more relevant and engaging content. For instance, NTC’s machine learning algorithms recommend tailored workouts, increasing the app’s usefulness and effectiveness. Furthermore, the incorporation of virtual coaching and real-time feedback supports users in maintaining correct form during exercise and provides motivational prompts, both of which contribute to better performance and reduced risk of injury. In place of complex automation, apps like FitnessTrack deliver motivational speeches and daily tips to inspire and engage users consistently.

Integration of data across platforms and devices also plays a crucial role in enhancing the user experience. Fitness apps that connect with wearable technology (such as Apple Watch, Fitbit, or Garmin) and data platforms like Apple Health and Google Fit provide users with a comprehensive view of their health and activity metrics. Cross-platform functionality, as seen in MyFitnessPal, enables users to monitor workouts, nutrition, and goals in one location, streamlining the fitness tracking process and offering a more cohesive ecosystem.

Despite these advancements, several challenges remain in the current market. Many fitness applications lack full integration of their core features—workout tracking, nutrition monitoring, and goal setting—resulting in a suboptimal user experience. Additionally, while some level of personalisation is common, many apps still offer generic recommendations without allowing users to edit them. User experience design also remains a significant hurdle, with users struggling to navigate complex interfaces or access relevant features efficiently. These challenges underscore the need for more unified, user-centric solutions like FitnessTrack, which provide an all-in-one management experience.

## System Overview

FitnessTrack is a mobile health and wellness application developed for the Android platform, aimed at helping users effectively manage their workouts, track nutrition, and achieve their own fitness goals within a single, user-friendly interface. Designed with accessibility and functionality in mind, the application combines core features such as workout logging, nutrition tracking, and goal setting, all of which are manually controlled and customizable by the user.

The application is built using Kotlin within Android Studio, which provides a stable and modern development environment. Kotlin is chosen for its concise syntax, null safety, and full compatibility with Java, contributing to clean, maintainable code and a smoother development process. The user interface is designed to be minimalistic and intuitive, offering a streamlined experience suitable for both fitness beginners and regular users. Clear navigation menus, input forms, and progress screens allow users to easily enter and update their fitness data without cognitive overload.

FitnessTrack’s backend infrastructure is supported by Firebase, which provides both secure user authentication and real-time cloud-based data storage. Firebase Authentication handles user login and registration processes with support for common sign-in methods, such as email/password and Google sign-in. The application stores user-generated data using Firebase Cloud Firestore, which offers real-time synchronization and offline capabilities. This allows users to log workouts and nutrition entries even when not connected to the internet, with automatic syncing once reconnected.

A key feature of FitnessTrack is its goal-setting functionality, which enables users to define, track, and manually update fitness objectives over time. These goals can be related to workout frequency, calorie intake, or other fitness metrics. Unlike some fitness applications, FitnessTrack gives users full control over their routines and allows them to adjust their goals as needed, providing flexibility and personalization without added complexity.

The app also includes built-in daily tips and motivational quotes designed to inspire users and provide helpful fitness advice. While the application does not use advanced AI or natural language processing, these motivational elements contribute to a positive and engaging user experience by offering encouragement and practical guidance regularly.

Additionally, FitnessTrack is designed to be compatible with common fitness platforms through Google Fit API integration. This allows the app to retrieve basic health data such as steps and calories burned from connected devices like smartphones and smartwatches. While not essential to the app’s core functionality, this integration enhances the comprehensiveness of user data and provides a more complete overview of health progress when available.

Overall, FitnessTrack delivers a focused, practical solution for users seeking a reliable fitness companion app. It avoids unnecessary complexity by forgoing AI and heavy analytics, instead prioritizing user autonomy, simplicity, and reliable performance through its core features and technology.

## 3.2 Architecture Diagrams

**Fitness Track System Architecture**

A screenshot of a computer

AI-generated content may be incorrect.

The architecture of FitnessTrack follows a modular and layered design, separating the user interface, application logic, and backend services to enhance maintainability, scalability, and testing. Key features such as workout logging, nutrition tracking, and goal setting are developed as independent modules, allowing for clear code organization and easier future enhancements.

Firebase Authentication is used to securely manage user accounts, eliminating the need to store sensitive data locally. Data is handled through both Firebase Cloud Firestore and the Realtime Database: Firestore supports complex, structured data with advanced querying, while the Realtime Database is used for fast, lightweight syncing, such as real-time workout logs or step tracking. This dual approach ensures high performance and offline availability.

Firebase Cloud Functions are included to support background processes such as chatbot responses or notification triggers, offering a pathway for future scalability.

Overall, this cloud-backed architecture provides a secure, responsive, and adaptable foundation. It supports the project’s goal of delivering a seamless and engaging fitness tracking experience while addressing long-term maintainability and user expectations for performance and reliability.

A screenshot of a computer

AI-generated content may be incorrect.

**Entity Relationship Diagram**

A screenshot of a computer

AI-generated content may be incorrect.

The Entity Relationship Diagram (ERD) was developed to represent a scalable, modular data structure appropriate for mobile fitness applications. It establishes one to many relationships between the User entity and the core data entities: Workout, Meal, and Goal, ensuring that each user’s information is logically grouped while preserving separation of concerns. This relational structure supports efficient data retrieval and real-time synchronization, which are essential for delivering a smooth and responsive user experience.

Each entity aligns with a distinct functional component within the application. The Workout entity captures data from the workout logging feature, the Meal entity supports nutritional tracking, and the Goal entity stores user-defined fitness objectives. By isolating these entities, the system promotes maintainability, simplifies the development process, and facilitates future feature integration.

The ERD has been structured to be fully compatible with Firebase Firestore, using a collection and subcollection model such as users/{userId}/workouts/{workoutId} in accordance with NoSQL best practices. This approach enables scalable data storage, optimized querying, and robust offline support, all of which are critical for reliable and user-focused mobile application performance.

A screenshot of a computer

AI-generated content may be incorrect.

**Use Case Diagram**

A screenshot of a black screen

AI-generated content may be incorrect.

The Use Case Diagram for FitnessTrack represents the core system functionalities as experienced by its primary actors: the User, Admin, Firebase, and Wearable Device. The User is the central actor, interacting with essential features such as account registration and login, setting fitness goals, logging workouts and meals, viewing progress, and synchronising health data from external devices. These actions reflect the app’s focus on usability, engagement, and personalisation.

Firebase is represented as an external system that handles secure authentication and cloud-based data storage. It ensures user credentials, fitness logs, and goal-related data are processed reliably and securely, in line with mobile application best practices. The Wearable Device serves as another external actor, supplying health metrics such as step count and heart rate. These metrics are integrated into the app via APIs like Google Fit, enhancing the accuracy and richness of user data without requiring manual entry.

The admin role is included to represent backend oversight, including the potential for managing user activity and maintaining system integrity. Although this role is limited in the current implementation, it provides a pathway for future expansion involving administrative analytics or support features.

Overall, the diagram offers a clear, high-level visualisation of system interactions. It supports both system validation and stakeholder communication by illustrating how the application meets user requirements while incorporating security, scalability, and external data integration.

A screenshot of a black screen

AI-generated content may be incorrect.

A black screen with white squares

AI-generated content may be incorrect.**Zero Trust Architecture**

FitnessTrack adopts a Zero Trust Architecture (ZTA), which assumes that no user or system component is inherently trusted. Every action must be verified, and access is only granted after strict authentication. Users must log in via Firebase Authentication before accessing any features, and a valid token is required for all data interactions.

Access to data follows the principle of least privilege. Firebase Firestore security rules ensure that users can only view or modify their own data. These rules validate each request in real time, preventing unauthorised access even within the app.

All communication with Firebase occurs over secure HTTPS, and internal requests must include valid authentication headers. By enforcing strong identity checks, limiting access, and avoiding implicit trust, FitnessTrack maintains a secure, privacy-focused system built on modern mobile security practices.

In the future, enhancements such as two-factor authentication and end-to-end encryption will be considered to further strengthen user data protection.

A black screen with white squares

AI-generated content may be incorrect.

**Data Flow Diagram with Trust Boundaries**

A black screen with white squares

AI-generated content may be incorrect.

The Data Flow Diagram for FitnessTrack helps illustrate how information moves securely through the system and where key security checks take place. It starts with the user interacting with the Kotlin based Android app, where sensitive features such as workout tracking and goal setting are protected behind a secure login. When the user signs in, their credentials are sent over HTTPS to Firebase Authentication, which returns a secure token, known as a JSON Web Token (JWT), used to confirm their identity.

This token is attached to every request the app makes, whether it is sending data to Firebase Firestore or pulling health statistics from the Google Fit API. Each time data moves from one part of the system to another, such as from the user to the app, or from the app to Firebase, it crosses a trust boundary. These boundaries act as security checkpoints, where authentication is required and all communication is encrypted.

The system does not process any request unless the user’s permissions are first checked through Firebase’s security rules. These rules are applied in real time and ensure that users can only access their own data. By clearly separating trusted and untrusted areas, the app reduces risk and protects user information. This structure follows modern Zero Trust security principles and supports a secure, reliable mobile experience.

A black screen with white squares

AI-generated content may be incorrect.

**Firebase Authentication & Authorization Flow**

A black screen with white squares

AI-generated content may be incorrect.

FitnessTrack uses Firebase Authentication and Firestore Security Rules to ensure secure user access and data protection. When a user signs in using email, password, or Google Sign-In, Firebase verifies their credentials and issues a JSON Web Token (JWT). This token is securely stored on the device and included in all future requests to the backend.

Each time a request is made, Firestore checks the token and applies security rules that restrict access to user-specific data. For example, users can only read or write documents where the document ID matches their own unique user ID. These checks happen on every request, not just at login, ensuring ongoing protection of sensitive information.

Token expiration, encrypted HTTPS communication, and real-time rule enforcement help prevent common threats like session hijacking or unauthorized data access. Together, these measures form a layered and reliable access control system that aligns with best practices in mobile and cloud security.

A black screen with white squares

AI-generated content may be incorrect.

## 3.3 Major Components

**3.3.1 Frontend**  
For the frontend, Kotlin was used alongside XML layout files within Android Studio to build a responsive and intuitive user interface. Kotlin’s modern language features and strong support in the Android ecosystem enabled efficient development and seamless integration with backend services, enhancing overall app performance and user experience.

**3.3.2 Backend**  
The backend architecture relied on Firebase, utilizing its Cloud Functions and Firestore services to manage serverless operations and real-time data synchronization. Firebase’s cloud-based infrastructure provided scalability, security, and simplified backend management, making it well-suited for mobile health application needs.

**3.3.3 Database**  
Firebase Realtime Database was chosen as the database solution due to its ability to handle real-time data updates across multiple devices efficiently. Its NoSQL cloud-hosted design allowed flexible data storage and quick retrieval, which are essential for managing dynamic user fitness and nutrition information.

**3.3.4 APIs / Integrations**  
Several APIs and integrations were incorporated, including TheMealDB API for meal data, OpenAI API for delivering motivational quotes and daily tips, Firebase Authentication for user security, and Firebase Cloud Services for backend operations. Additionally, hardware integrations like footstep counters, Bluetooth, and body sensors were utilized to enhance fitness tracking, while Google Snackbar was used for user interface notifications.

**3.3.5 Security**  
Security measures included implementing Firebase Authentication for secure user sign-in and access control, alongside privacy-focused design principles such as encrypted data transmission and strict access rules within Firebase. These precautions ensured protection of sensitive health data and compliance with best practices for mobile health applications.

## 4.1 Methodology Used

The development of FitnessTrack followed an Agile development methodology, which was selected due to its iterative, user-focused, and flexible nature. Agile allows for continuous refinement through short development cycles (sprints), regular testing, and responsive adaptation to user feedback (Beck et al., 2001). This approach was particularly advantageous for FitnessTrack, where user engagement, interface responsiveness, and usability are critical.

The development process was structured into multiple sprints, with each sprint dedicated to a specific set of features such as UI design, authentication system, workout goal settings, calorie intake tracking, and real-time data syncing. At the end of each sprint, the completed features were reviewed and tested before progressing, in line with Agile principles of incremental progress and feedback incorporation (Highsmith & Cockburn, 2001). Throughout the project, Agile practices such as daily stand-up meetings and sprint retrospectives were conducted to ensure team alignment and continuous improvement.

A key element of the methodology was user-centred design, focusing on incorporating user needs and preferences at all stages of development. Early in the development cycle, the application was tested by final-year software engineering students Isaaq Mahmood and Mark Alan Henry, who provided valuable informal user feedback. Although they were not directly involved in the development process, their role as early testers helped validate design decisions and highlighted areas for improvement. For example, their feedback led to refinements in the layout of the manual goal-setting feature, simplification of the chatbot interaction flow, and optimization of the workout logging screen. These testing sessions were conducted across multiple iterations, with informal usability metrics such as ease of navigation and feature clarity being evaluated. This early user testing aligns with modern Agile development practices, where feedback loops are leveraged to drive continuous product enhancements (Gothelf & Seiden, 2013).

While the application does not rely on advanced natural language processing (NLP) or machine learning for personalized recommendations, the motivational quotes feature was thoughtfully built using the OpenAI API. This enables it to provide smart yet predictable responses, guided by predefined conversational flows. The interaction design focuses on clarity and ease of use, particularly for beginners, by using structured prompts and offering limited response options. This reduces cognitive load and makes the experience more approachable. Rather than aiming for complexity, the chatbot prioritizes helpfulness and simplicity, in line with user-centred design principles that value clear, intuitive interfaces—especially in health and fitness apps (Allous, 2023).

Tools such as Git for version control and Trello for sprint planning were employed to support the Agile workflow. These tools enabled streamlined organization and maintained code integrity throughout the development lifecycle. Visibility into sprint goals and progress was maintained, while continuous testing and debugging were conducted after each sprint to ensure system reliability and smooth feature integration (Rubin, 2013).

In conclusion, the Agile methodology provided a responsive and iterative approach to building FitnessTrack, ensuring user feedback directly influenced design and functionality. The structured involvement of early testers, combined with the use of sprint cycles and Agile ceremonies, enabled a well-rounded and user-validated mobile fitness solution.

## 4,2 Planning and Requirements Gathering

**Stakeholders Identification**

For this project, the primary stakeholders were the users, fitness enthusiasts, and developers. Users are the key drivers of the application, as they provide the necessary feedback and requirements that guide the app’s features and overall direction. Fitness enthusiasts, who actively use fitness apps for tracking workouts, nutrition, and progress, are the target audience, ensuring that the features align with their needs. As the developer of the app, I played a crucial role in ensuring that the app's design, functionality, and user experience met the needs of both fitness enthusiasts and general users. Secondary stakeholders included trainers, nutritionists, and specialists who provided input on fitness and dietary needs. Trainers and nutritionists could offer insights on fitness and dietary needs, potentially influencing the app's functionality in the future, while specialists advised on integrating advanced features to ensure a seamless experience. However, the focus was primarily on users, and feedback from early testers like Isaaq Mahmood and Mark Allan Henry, who were not directly involved in development but served as testers, helped ensure that the app met the intended user experience goals.

|  |  |  |
| --- | --- | --- |
| Stakeholders | Priority Level | Role |
| Users | Primary | Provide feedback and requirements guiding app features. |
| Fitness Enthusiasts | Primary | Target audience ensuring features meet tracking and progress needs |
| Developer (Self) | Primary | Designs and implements app functionality and user experience |
| Nutritionists | Secondary | |  | | --- | |  |  |  | | --- | | Advise on dietary needs and nutrition tracking | |
| Specialists | Secondary | Offer guidance on advanced features and integration aspects |
| Early Testers (Isaaq Mahmood, Mark Allan Henry) | Secondary | |  | | --- | |  |  |  | | --- | | Offer informal user feedback validating design decisions | |

**Technical Requirements**

The technical requirements for the app were defined to ensure smooth development and integration with key tools. The app would be built for Android devices using Kotlin, which offers a modern, concise, and highly compatible programming language for mobile app development. The backend and database of the app would rely on Firebase, which provides cloud-based services such as authentication, real-time database, and cloud storage. This would ensure a seamless user experience and efficient data management. Integration with wearables, such as Galaxy Watch and Garmin, was identified as a key technical requirement to enhance the app’s data tracking capabilities, allowing users to sync their fitness data and track progress across platforms.

**Feasibility Study**

A feasibility study was conducted early in the development process to assess the budget, timeline, and technology stack for the project, ensuring the app could be developed within the set timeframe and that chosen technologies could support planned features. Given the project scope and available resources, Firebase for the backend and Kotlin for the Android app were deemed both feasible and practical. The estimated development timeline was approximately 12 weeks, with core features like user authentication, workout logging, and nutrition tracking prioritized in the initial sprints, while wearable integration was allocated additional time due to potential API complexities. Key technical risks included real-time data synchronization during offline use and device compatibility, which were mitigated by leveraging Firebase’s offline support and early prototyping of Google Fit API integration. Ultimately, the feasibility study confirmed the project’s viability within constraints, supporting delivery of a fully functional, engaging fitness app.

**Sprint Plan**

* **Sprint 1: Project Kickoff & Setup (2 weeks)**  
  Finalized the project scope and goals, set up the development environment (Android Studio, Firebase), created initial UI wireframes, and implemented Firebase Authentication for user sign-in.
* **Sprint 2: UI & User Profile (2 weeks)**  
  Developed the basic UI using Material Design principles. Implemented the user profile feature with Firebase Firestore for storing and retrieving user data.
* **Sprint 3: Fitness Tracking (2 weeks)**  
  Added workout logging, step tracking, and calorie tracking. Designed new UI screens for exercise input and integrated Firebase Realtime Database for data syncing.
* **Sprint 4: Nutrition Tracking (2 weeks)**  
  Enabled meal logging, calorie tracking, and nutritional insights. Implemented a simple meal suggestion feature based on user goals.
* **Sprint 5: Motivation Integration (2 weeks)**  
  Integrated a motivational quotes and daily tips via Firebase Cloud Functions. Designed a smooth chat interface for user interaction and advice delivery.
* **Sprint 6: UI/UX Testing & Refinement (2 weeks)**  
  Conducted usability testing, fixed bugs, and refined the interface based on user feedback. Improved overall app performance and user experience. Some UI adjustments were made following tester feedback, demonstrating the adaptive nature of the development.
* **Sprint 7: Wearables Integration (2 weeks)**  
  Integrated APIs for syncing data (heart rate, steps, etc.) from devices like Samsung and Garmin for seamless fitness tracking. Encountered minor integration challenges that were addressed through iterative testing.
* **Sprint 8: Final Testing & Release Prep (2 weeks)**  
  Performed thorough testing (unit, integration, UI), resolved remaining bugs, and optimized the app for stability. Prepared the app for final submission.

## 4.3 Implementation Process

The development of the FitnessTrack application followed a structured, multi-phase implementation process that emphasizes modularity, security, and responsiveness. This process was divided into frontend, backend, and feature-specific development phases, ensuring a clear and maintainable architecture while prioritizing user experience and functionality.

**Frontend Development**  
The frontend of the application was built using Kotlin in Android Studio, adhering closely to Google’s Material Design guidelines. This ensured a clean, intuitive, and responsive user interface suitable for both fitness enthusiasts and casual users. Special attention was paid to user journey design, incorporating logical navigation patterns and accessible input forms. Kotlin was selected as the primary frontend language due to its concise syntax, null safety, and full Java interoperability, all of which enhanced development speed, reliability, and application stability.

**Backend** **Development**  
The backend development was implemented using Firebase, providing a secure, robust, and scalable cloud service fully integrated within Android Studio. Firebase Realtime Database was utilized to store and synchronize user data in real-time, supporting workout logs, nutrition entries, and goal tracking. Data consistency across devices was maintained through its low-latency syncing capabilities. Firebase Authentication handled secure sign-up, login, logout, and session management, ensuring all data access was protected via token-based authorization. Additionally, Firebase Cloud Storage was used for media uploads, such as profile pictures or workout images, ensuring fast and reliable content delivery. Firebase Cloud Functions were employed to run custom server-side logic securely in response to specific events.

**Feature Implementation**  
A key engagement feature in FitnessTrack is the daily motivational quotes and fitness tips that are delivered to users. These messages are designed to encourage consistent effort and foster a positive mindset towards health and fitness. The motivational content is managed on the backend and delivered through Firebase Cloud Functions, allowing for dynamic updates and personalized tips over time. The profile management feature allows users to create and customize their profiles, track fitness goals, and link wearable devices via APIs such as Google Fit. User profile data, including goal progress and health metrics, is stored securely in Firebase Firestore, which offers a scalable NoSQL structure with real-time update capabilities.

## 4.4 Testing and Validation

The testing and validation phase of FitnessTrack development process was comprehensive iterative, and aligned with best practices in modern application development. Multiple testing types were employed to ensure the reliability, usability and robustness of both individual components and the overall system. This rigorous approach helped verify that the application not only functioned correctly but also provided a smooth and the secure experience for users across different devices and usage scenarios.

**Type of Testing**

To begin, unit testing was performed using JUNIT, a standard testing framework for Kotlin, to verify the correctness of individual functions. This included testing critical components such as the Motivational Quotes functionality, API integration methods, login handler and user authentication handlers. By isolating each function and validating its output under various conditions, unit testing ensured a solid foundation for all applications modules.

Next, integration testing was conducted to assess the interaction between the frontend (developed in Kotlin) and the backend services powered by Firebase. This testing ensured that user actions – such as logging a workout or updating a meal entry, which trigger the appropriate backend processes and that data was correctly stored and retrieved in real-time from the Firebase Realtime Database Integration testing also covered asynchronous interactions, including those involving Firebase Cloud Functions for random motivational quotes.

A key focus of testing was UI/UX evaluation, which involved usability testing with real users. Early-stage testers interacted with the application and provided feedback on layout clarity, navigation flow, and interface intuitiveness. Their insights led to several refinements, including improved input validation, better feedback prompts, and more accessible goal-setting screens. This user-centric testing phase played a critical role in ensuring that the interface was not only functional but also welcoming and intuitive, especially for beginners.

Finally, Regression testing was also systematically applied throughout the development lifecycle to ensure that new feature additions or code changes did not introduce bugs or break existing functionality. Both automated scripts and manual test cases were used to repeatedly test core functions such as login, data sync, and fitness tracking after every sprint. This practice-maintained code integrity and minimized the risk of overlooked issues.

**Bug Fixes & Iteration**

Throughout the beta testing period, user feedback played a key role in shaping the app’s development. Ongoing feedback loops supported multiple rounds of bug fixing and feature refinement. One major area of improvement was the motivational quotes and daily tips. Users noted issues with the response time and the relevance of the daily tips suggestions, which led to targeted updates in the firebase database logic handled by Firebase Cloud Functions. Likewise, bugs affecting wearable integration, particularly inconsistencies in syncing data from Google Fit—were identified and addressed through several iterative development cycles.

Performance concerns also emerged during testing, especially around data sync speed and app responsiveness. In response, backend optimisations were made to improve overall performance. Firebase queries were refined, database paths were restructured for quicker access, and Cloud Function triggers were streamlined to reduce processing delays. These changes led to faster feedback during activity logging and a smoother experience when syncing data across devices, resulting in a more reliable and user-friendly system.

## 5.1 What Worked Well

Several aspects of the FitnessTrack project contributed significantly to its success, particularly in terms of methodology, technology stack, and user-centred design. Each of these factors supported a streamlined development experience and delivered meaningful value to end users.

One of the most impactful elements was the adoption of the Agile development methodology, which enabled iterative progress through clearly defined sprints. This approach supported continuous integration of feedback, especially from early testers (Isaaq Mahmood & Mark Allan Henry), leading to refinements that directly addressed real-world user needs. Agile’s core principles—such as flexibility, user collaboration, and responsiveness to change—aligned well with the nature of the mobile application, where usability and user experience are critical success factors (Beck et al., 2001; Gothelf & Seiden, 2013). As a result, the project adapted swiftly to user input, producing a more polished and effective final product.

The integration of Firebase as the backend solution also proved highly effective. Firebase’s suite of tools, including Authentication, Realtime Database, and Cloud Firestore, provided a scalable, secure, and efficient platform for storing user data and managing app state. Its real-time synchronization capabilities enabled seamless data updates across devices, an essential feature for a fitness app where users often interact mid-activity or offline. According to Firebase documentation (Firebase, 2023a), its integration with Android reduces boilerplate code and enhances productivity—both of which were evident in this project’s rapid development cycle. Additionally, Firebase’s built-in security features facilitated straightforward implementation of Zero Trust access rules, improving the overall robustness of the system.

Another major strength was the app’s emphasis on user-centric features. Personalized goal tracking, nutrition logging, and daily motivational quotes and tips successfully addressed common pain points in existing fitness apps, such as lack of personalization and overwhelming interfaces (Stassen, Garcia, and Mohan, 2022). By focusing on simplicity, clarity, and goal alignment, FitnessTrack enhanced user engagement and promoted sustainable use—an achievement supported by recent research suggesting that apps providing tailored feedback and adaptable features are more likely to facilitate long-term behaviour change (Müller et al., 2022).

Finally, the selection of Kotlin as the primary programming language within Android Studio contributed to the stability and maintainability of the codebase. Kotlin’s concise syntax, null safety, and full interoperability with Java vastly improved code quality and developer productivity (Obregon, 2021). Its modern features enabled more expressive code, reducing the risk of common programming errors and enhancing readability. Android Studio’s debugging tools, layout editor, and performance profiler further supported an efficient development workflow, ultimately helping to produce a responsive and high-performing mobile application

## 5.2 Challenges and Issues Faced

Despite the overall success of FitnessTrack, several notable challenges arise during the development and testing. These issues not only tested the technical resilience of the solution but also offered important lessons that will guide future improvements and development cycles.

One of the more complex aspects of the project was managing the delivery of motivational tips and daily guidance to users. While the concept was straightforward—providing regular encouragement and advice to support users’ fitness journeys—the technical execution required adjustment. Initially, these tips were stored and served through the Firebase Realtime Database. However, as the volume of content and user requests increased, the database became overloaded, resulting in slower performance and reduced reliability. To address this, the system was restructured to store the tips as JSON files in Firebase Cloud Storage instead. This approach significantly improved scalability and reduced latency, while still allowing the app to fetch and display content dynamically. Managing file access permissions and ensuring secure delivery remained important considerations throughout the transition (Firebase, 2023a).

Another major hurdle involved integrating wearable fitness devices, particularly from Samsung and Garmin. While syncing data through the Google Fit API was relatively straightforward, support for third-party wearables introduced significant complexity. Many of these platforms lacked clear documentation or required proprietary SDKs that added platform-specific constraints. This not only increased development time but also introduced bugs that were often difficult to identify and resolve. The absence of standardised protocols across devices made it challenging to maintain consistent data flow and functionality across different brands and platforms.

As the app matured, the Firebase Realtime Database began to show limitations. Although it worked well for simple, lightweight operations, the growing volume and complexity of user data led to slower read and write operations, particularly during batch updates. These issues required backend optimisation, including restructuring the database paths, reducing redundant queries, and shifting selected tasks to Firebase Cloud Functions to improve performance. This experience highlighted the importance of planning for scalability from the outset, especially when working with real-time, cloud-based NoSQL databases (Firebase, 2023b).

User testing also had its constraints. Ideally, feedback would be gathered from a broad and diverse user base to reflect a wide range of fitness levels, device types, and accessibility needs. However, due to time and resource limitations, the beta testing group consisted of only two final-year software engineering students. While their technical feedback was helpful, it did not fully capture the perspectives of less tech-savvy users or those with different health and fitness backgrounds. As a result, the scope of UI/UX improvements was narrower than intended, and some assumptions about user preferences could not be fully validated.

Although these challenges introduced additional complexity, they were a valuable part of the learning process. They reinforced the importance of early feasibility checks, scalable architecture design, and inclusive testing strategies. Each obstacle contributed to a deeper understanding of mobile development, cloud integration, and user-focused design—lessons that will shape future iterations of FitnessTrack and future projects alike.

## 5.3 Lessons Learned

The FitnessTrack application offered valuable learning experiences across both academic and technical implementation. Reflecting on the challenges and development process has helped to identify several key lessons that would inform future work, particularly in the areas of third-party API integration, user testing, motivational content delivery, and data security.

One major takeaway was the importance of integrating motivational quotes and daily tips early in the development cycle. Introducing these features sooner would have allowed more time to iterate, test user engagement, and tailor the content for greater user motivation and retention. Early integration also would have provided opportunities to refine the delivery mechanisms and optimize the balance between helpfulness and user cognitive load.

A second lesson emerged from the wearable device integration process. Attempting to synchronize data with devices like Samsung Galaxy Watch and Garmin proved to be very time-consuming due to limited or poorly documented APIs. In retrospect, proactively establishing communication with the manufacturers or their developers could have clarified integration requirements and expedited troubleshooting.

The project also underscored the critical role of iterative user testing throughout development, not just at the end. Limited user diversity in the beta testing phase meant that design flaws, especially those affecting accessibility or user flow, were discovered relatively late. Incorporating ongoing usability testing from the beginning would have allowed the team to adapt the design based on real-world interaction patterns, leading to a more refined and inclusive product. This aligns with Agile principles and user-centred methodologies, which emphasise early and continuous feedback loops (Gothelf & Seiden, 2013).

Finally, the project highlighted the importance of planning for data security and privacy from the outset. Given the sensitive nature of health and fitness data, security should not be treated as a secondary concern or late-stage consideration. While Firebase offered robust authentication and secure data handling capabilities, initial configuration required review and revision to meet best practices. Future projects would benefit from integrating privacy-by-design principles, such as secure data storage rules, encrypted communications, and clear user consent mechanisms right from the initial planning stages (Firebase, 2023a).

Overall, the lessons learned through the development of FitnessTrack will serve as crucial guidelines for future work in both mobile app development and digital health solutions. They reflect the value of foresight, proactive collaboration, and continuous feedback in building effective, secure, and user-centred technologies.

## 5.4 Retrospective: What Would Be Done Differently

Looking back on the development of FitnessTrack, several areas emerged where earlier decisions could have improved outcomes. These lessons will be valuable for future projects, especially in mobile health development.

One key area was wearable integration. While devices like Samsung and Garmin were eventually supported, starting this work late in the cycle led to rushed debugging and limited testing. Integrating wearables earlier—such as allocating dedicated sprints for API research and initial synchronization tests—would have allowed for smoother performance tuning and a more seamless user experience. Given the rising importance of wearable data, this feature should have been a main priority from the outset.

The implementation of motivational quotes and daily tips also presented opportunities for improvement. Introducing these features earlier in the process would have allowed more time to gather user feedback on content relevance and delivery timing. For example, early A/B testing of notification schedules and quote types could have helped tailor the feature to maximize user engagement and motivation.

User testing came late in the project and involved a small, somewhat homogenous group. Incorporating iterative usability testing from early prototypes—with a more diverse pool of users including different ages, fitness levels, and accessibility needs—would have identified design flaws and flow issues sooner. Regular feedback sessions could have been scheduled after each sprint to continuously adapt the app based on real user interactions.

Lastly, scope management could have been more focused. Attempting to implement too many advanced features simultaneously stretched resources thin. Prioritizing a lean MVP centred on core features such as workout logging, goal tracking, and a polished user interface would have allowed for a more stable and refined initial release. Subsequent enhancements could then be rolled out incrementally based on user demand and feedback.

These lessons reinforce the importance of early planning, focused scope, and continuous user feedback—key principles for delivering user-centred and reliable mobile applications.

## 5.5 Evaluation of Tools and Technologies

The development of FitnessTrack was built on a carefully chosen to set of tools designed to support mobile-first development, personalization, and real-time data processing. Each technology contributed significantly to the project’s success while also introducing challenges that influenced both the timeline and architectural decisions. This section evaluates the key tools used based on their performance, efficiency, security, and suitability for the app’s goals.

**Kotlin and Android Studio**  
Kotlin, used as the primary programming language within Android Studio, proved to be a strong choice for mobile development. Its concise syntax, built-in null safety, extension functions, and full compatibility with Java enabled faster development and cleaner, more maintainable code. These features helped reduce common programming errors and enhanced overall productivity (Obregon, 2021). Android Studio provided a robust development environment, with valuable tools such as the layout editor, debugging console, and emulator support that simplified UI testing and design. However, some limitations emerged during the integration of advanced features and external APIs. Certain third-party libraries lacked Kotlin-specific documentation or required Java-based implementation workarounds, occasionally increasing development time and complexity, especially in asynchronous data flows.

Security considerations included Kotlin’s support for modern coding practices that reduce vulnerabilities such as null pointer exceptions and memory leaks. Android Studio also facilitated secure coding by integrating lint tools that detect common security issues during development.

**Firebase**  
Firebase played a central role in the backend architecture of FitnessTrack, providing essential services such as Firestore, Realtime Database, Authentication, and Cloud Functions. Its mobile-native design and seamless integration with Android Studio made it a natural fit. Real-time data syncing, secure user authentication, and cloud-based file storage contributed to a responsive, consistent user experience across devices. Firebase Authentication allowed for secure login and session management with minimal development effort (Firebase, 2023a). Security features such as token-based authentication, role-based access controls, and encrypted data transmission ensured user data was well protected.

However, as the project scaled, Firebase presented challenges in performance and complexity. Increasing workout and nutrition data volumes required careful data structure optimization and query indexing to avoid latency. Cloud Functions introduced a learning curve; debugging and deploying complex asynchronous workflows demanded developer familiarity, which impacted development velocity. These experiences highlighted the importance of early backend planning for data management and security in cloud-based mobile apps.

**Motivational Quotes and Daily Tips Feature**

FitnessTrack implemented motivational quotes and daily tips delivered through scheduled notifications powered by Firebase Cloud Functions. This approach simplified content management and ensured consistent user engagement without the overhead of real-time natural language processing. The system was reliable, low-latency, and easy to update. Future iterations could explore AI-powered personalization to tailor tips more closely to individual user progress.

Security for this feature was maintained by controlling content through authenticated admin interfaces and ensuring notifications complied with user privacy settings, avoiding intrusive or unsolicited messaging.

**Alternative Technologies Considered**  
Other technologies considered included React Native for cross-platform development, which was ultimately not chosen due to the project’s Android-first focus and desire to leverage Kotlin’s modern features. For the backend, AWS Amplify and Google Cloud Platform were reviewed but Firebase’s integration with Android Studio and its real-time database capabilities made it a better fit for rapid mobile app development. For motivational content delivery, simple scheduled jobs were preferred over complex AI solutions in this version to maintain stability and reduce dependencies.

## 5.6 Technical Appraisal of Final Product

The final version of FitnessTrack performs reliably across a range of Android devices, offering smooth interactions and responsive functionality. Real-time syncing through Firebase worked well, allowing users to see updates to workouts and meals instantly across sessions. While some delays were noted when syncing larger data sets from wearables, these were addressed through backend optimisations, including improved query handling and refined sync intervals.

Usability was a standout strength. The interface, built with Material Design principles, was intuitive and accessible. Users could easily navigate key features like workout logging, meal tracking, and AI-powered suggestions. The consistent layout and low cognitive load made the app particularly approachable for users new to fitness tracking.

From a reliability perspective, the app maintained stable authentication and data sync using Firebase services. Early issues with wearable device integration were resolved in later development stages, resulting in consistent and accurate data syncing across connected platforms.

Functionally, the app met all key requirements. Personalised workouts, nutrition insights, chatbot interactions, and fitness goal tracking were implemented successfully. While improvements could be made to the chatbot’s natural language processing and cross-device syncing, the core system delivered a cohesive, user-focused experience that aligned with project goals.

## 6. Conclusion

## 6.1 Recap of Aims and Objectives

The main goal of FitnessTrack was to create a user-friendly mobile fitness app that combines workout tracking, nutrition monitoring, and goal setting in one cohesive platform. This vision was shaped by early research showing that many existing apps are either too fragmented or overly complex, especially for beginners. The project also served as a personal development opportunity, offering hands-on experience in Android development, Firebase integration, and user-centred mobile design.

## 6.2 Summary of Work Done

Over the course of the project, a full-featured Android application was developed using Kotlin and Android Studio, with Firebase providing the backend infrastructure. The app supports secure login with email or Google accounts, real-time data syncing for workouts and meals, goal setting, and integration with Google Fit for wearable data. Development followed Agile methodology,

## 6.3 Key Findings

Development and testing revealed several important insights. Users clearly preferred an integrated solution over switching between separate apps, confirming the need for a unified design. Firebase proved effective overall, but as data volume grew, performance issues required backend optimisation. Integrating wearables like Samsung and Garmin presented challenges due to inconsistent APIs, but even partial integration added value. The chatbot, though not deeply conversational, was effective in providing structured guidance. Most importantly, early user feedback—even from a small group—was critical in shaping a smoother, more intuitive experience.

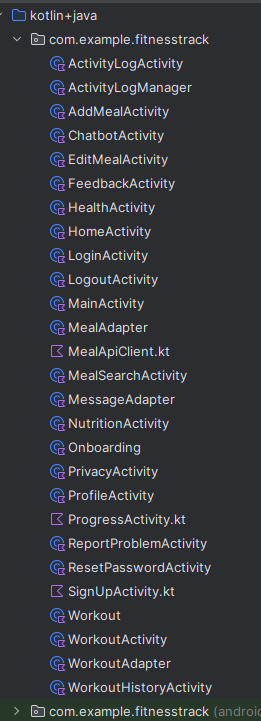
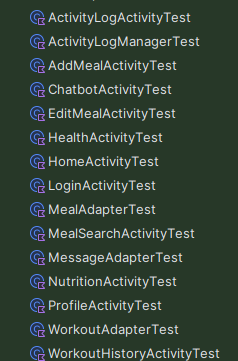
## 6.4 Final Thoughts and Future Work

FitnessTrack successfully delivered a stable, user-centred fitness app that combines multiple health features in one system. The project deepened technical skills and highlighted the value of planning, iteration, and real-world testing in mobile development.

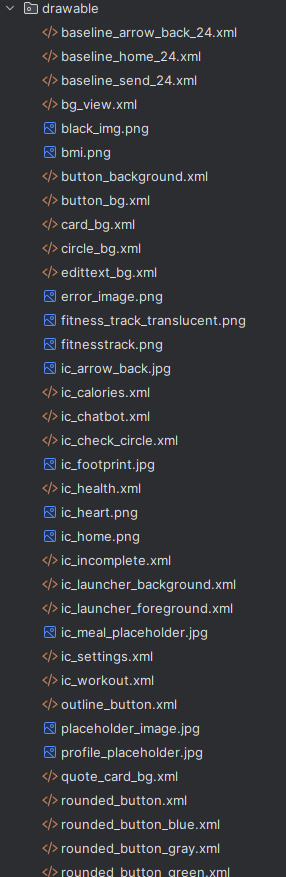
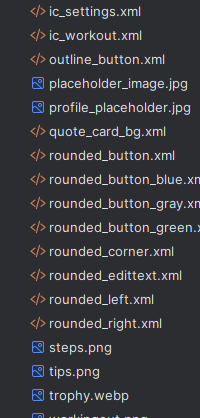
Future improvements will focus on implementing an AI chatbot with advanced natural language processing, expanding wearable support to platforms like Fitbit and Apple Health, and introducing personalised notifications and stronger privacy controls. Broader user testing will also help refine accessibility and ensure the app continues to meet diverse user needs.

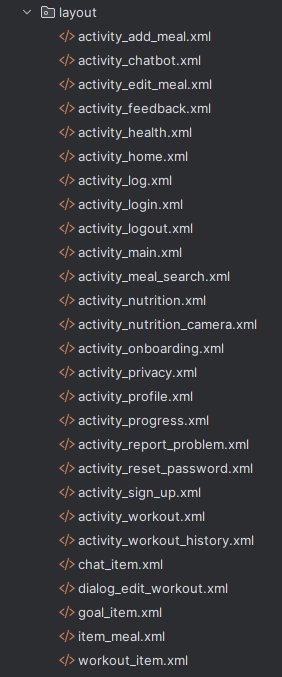
Overall, FitnessTrack lays a strong foundation for ongoing development, offering a practical, well-designed solution in the digital health space with clear potential for future growth and innovation.

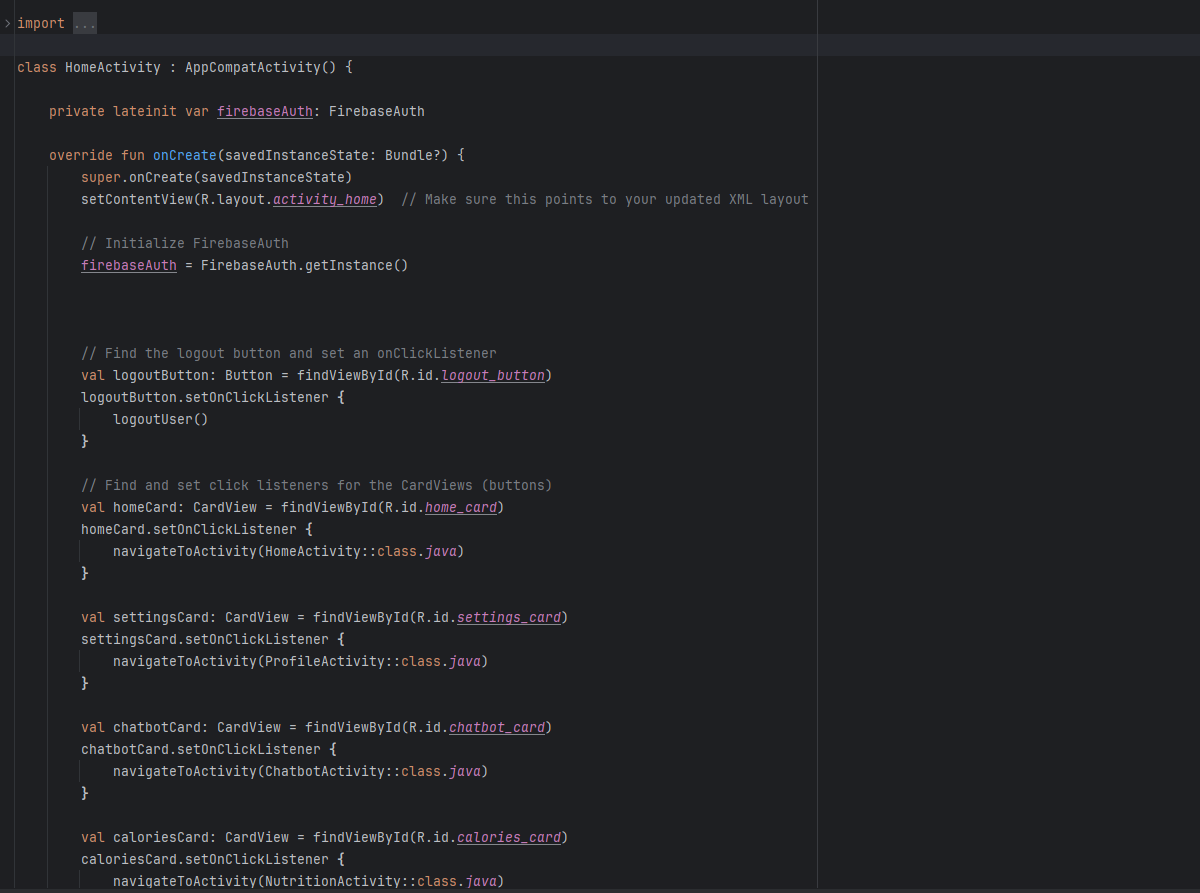
Screenshots

****

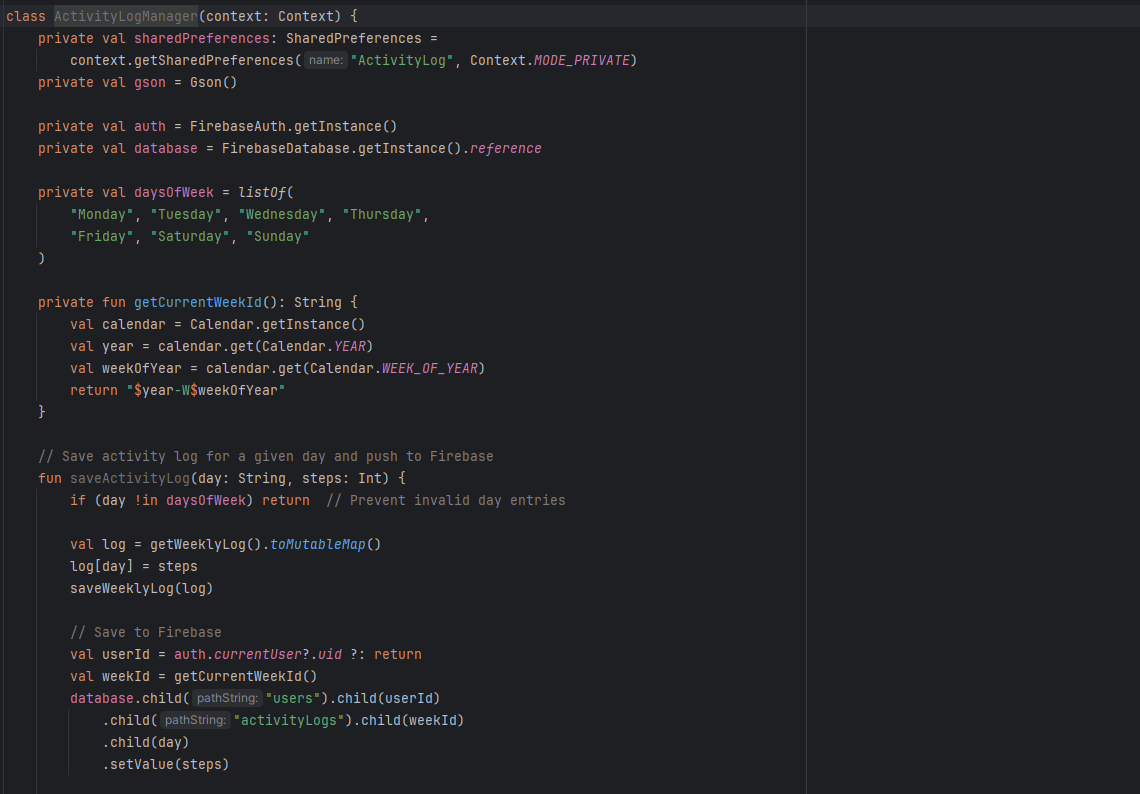
The following section presents the Kotlin classes I have implemented for this project. These classes demonstrate the core functionality and structure used in the application.

****

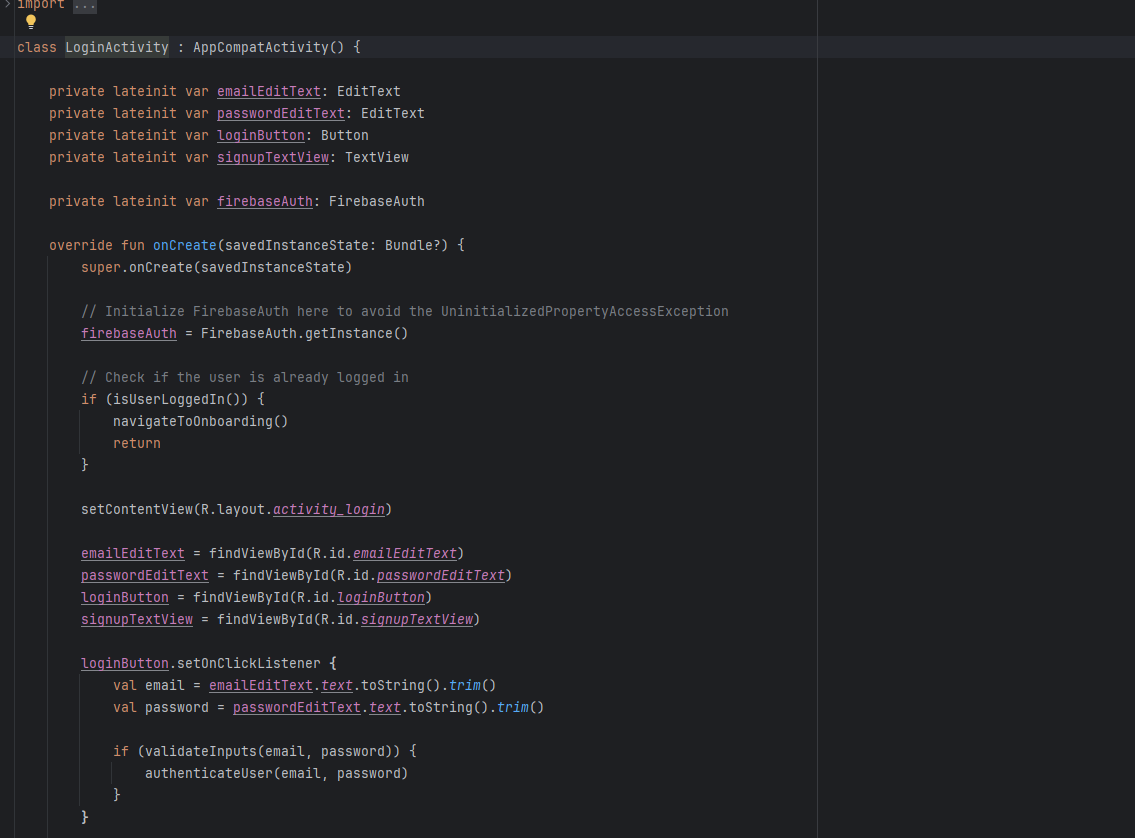
This section presents the layout drawable resources utilized throughout the project. These drawables are essential for defining the visual design and styling of the user interface components. They include shapes, colours, gradients, and other graphic elements that enhance the appearance and usability of the app’s layouts. The careful design of these drawables contributes to a consistent and appealing user experience.



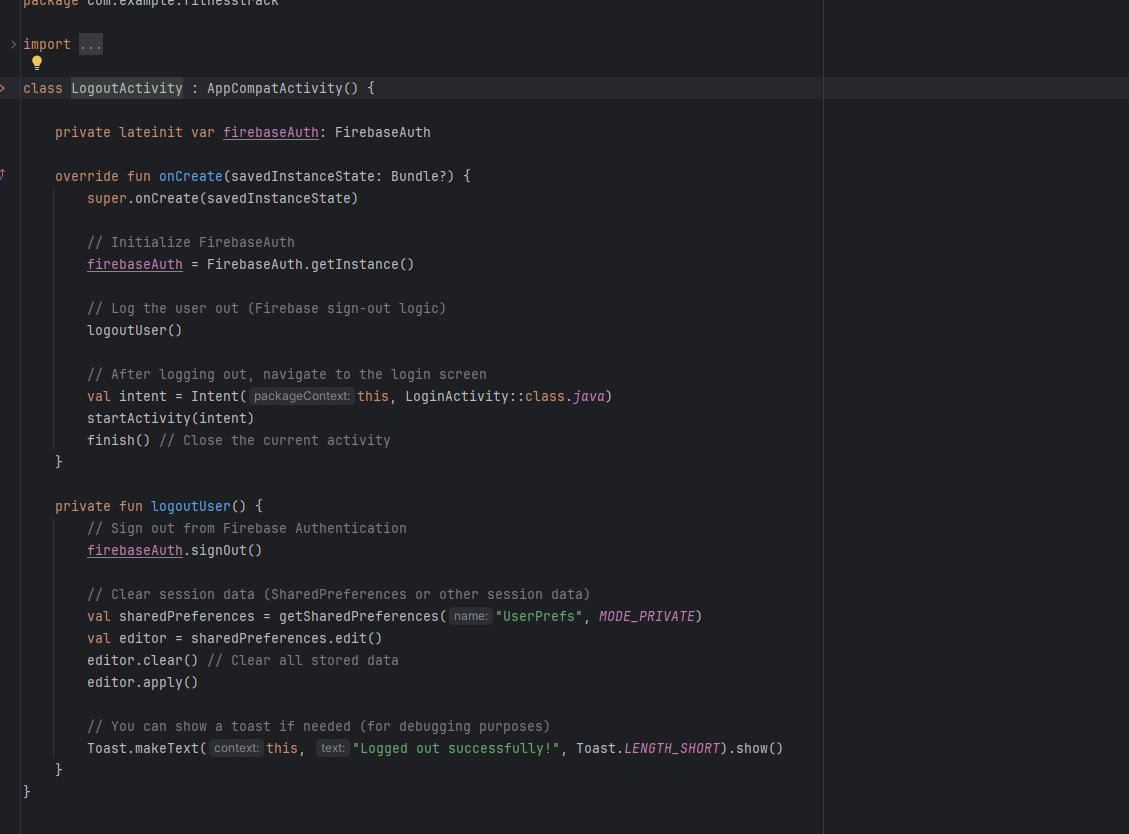
The Home Activity serves as the central navigation hub in your fitness tracking app. It initializes Firebase authentication to manage user sessions and provides a logout function that cleanly signs users out while displaying a confirmation message. The activity features multiple CardViews that act as intuitive buttons, each linking to different aspects of the application, such as settings, the chatbot, nutrition, health, and workout sections. This design creates a seamless and user-friendly interface that allows for easy navigation across all key features of the app.

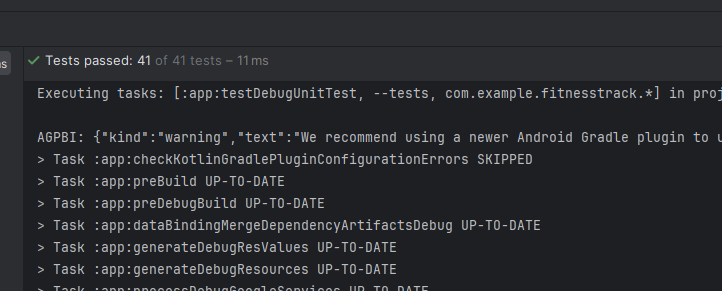


The ActivityLogManager class handles the storage and retrieval of a user's daily and weekly step data in your fitness app. It integrates Firebase Realtime Database for cloud-based persistence and uses SharedPreferences for local caching. The class allows you to save, fetch, and clear step counts for each day of the week and supports weekly summaries through automatic aggregation. By mapping day names to actual calendar days and syncing with Firebase, it ensures accurate tracking and availability of step data even across app sessions.

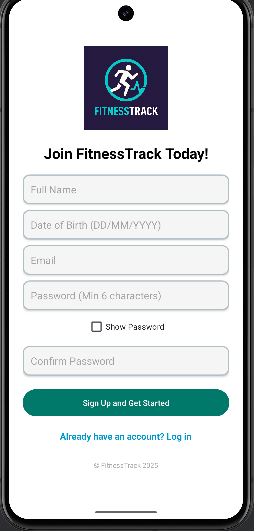


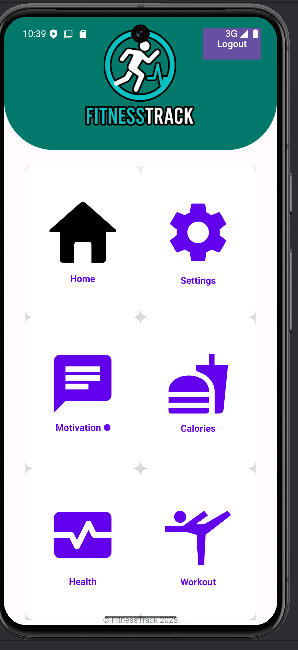
The LoginActivity handles user authentication using Firebase in your fitness tracking app. It checks if a user is already logged in and redirects them to the onboarding screen if so. Users can input their credentials, which are validated before attempting to sign in via Firebase Authentication. Successful logins result in credentials being stored locally using SharedPreferences, ensuring a persistent login state. The activity also provides navigation to the sign-up screen for new users.

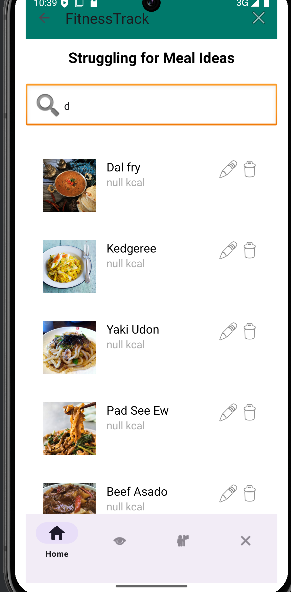
The LogoutActivity is responsible for securely signing users out of the app. It uses Firebase Authentication to end the session and clears any locally stored user data in SharedPreferences. Once the logout process is complete, the user is redirected back to the LoginActivity, ensuring a clean transition and preventing access to restricted areas without reauthentication.

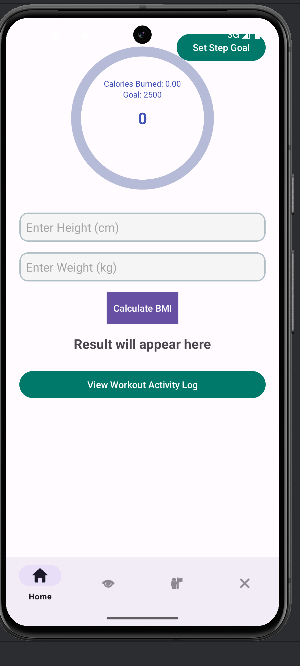


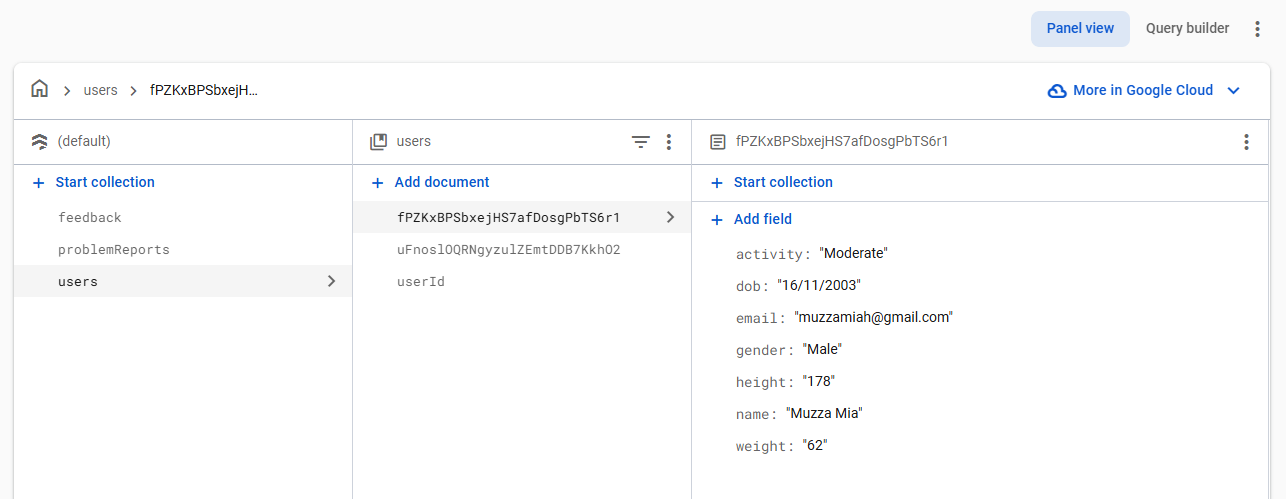
JUnit testing was conducted to verify key functionalities such as input validation, Firebase authentication, and SharedPreferences handling. Tests confirmed that user login, logout, and session management logic worked as expected. All tests passed successfully, ensuring the app performs reliably under typical and edge case scenarios.

The **SignUpActivity** allows new users to register by entering their full name, date of birth, email address, and password. It uses **Firebase Authentication** to securely create user accounts. Upon successful registration, additional user details such as full name and date of birth are stored in Firebase Realtime Database or Firestore, enhancing the user profile beyond just email and password authentication.

This image shows the **Home Page** of the app, where users can quickly access key features like workouts, nutrition, chatbot, and settings through simple card-style buttons. It serves as the central navigation hub after login

This Nutrition page lets users search, view, edit, and delete meals fetched from an external API. It includes smooth real-time search and easy navigation to other app sections via a bottom menu. The activity also handles logout and displays helpful messages for errors or no results.

This Health page tracks user steps using the device's step counter sensor and shows progress toward a customizable daily step goal with a circular progress bar. It calculates calories burned based on steps and includes a BMI calculator. User data is synced with Firebase Realtime Database. The page has bottom navigation for easy access to other app sections and handles permissions gracefully for step tracking.

y Firebase database securely stores user details, feedback, and problem reports. This setup allows users to provide valuable input and report issues directly through the app. Storing this data in Firebase enables real-time updates and easy management, helping improve the app’s functionality and user experience.

## **References**

Allous, A. (2023) *Gamification in Mobile Fitness Apps: Enhancing User Motivation*. University of Twente Student Theses. Available at: <https://essay.utwente.nl/100800/1/Allous_BA_EEMCS.pdf> (Accessed: 16 April 2025).

Beck, K., Beedle, M., van Bennekum, A., et al. (2001) *Manifesto for Agile Software Development*. Available at: <https://agilemanifesto.org/> (Accessed: 16 April 2025).

Eastern Peak (2023) *Fitness App Design Best Practices: Features and UX Tips*. Available at: <https://easternpeak.com/blog/fitness-app-design-best-practices/> (Accessed: 16 April 2025).

Evangelist Apps (2024) ‘How Machine Learning is Shaping Personalized User Experiences in Apps’, *Medium*, 8 October. Available at: <https://medium.com/evangelist-apps/how-machine-learning-is-shaping-personalized-user-experiences-in-apps-2128f839f0b7> (Accessed: 16 April 2025).

Firebase (2023a) *Get Started with Firebase Authentication on Android*. Available at: <https://firebase.google.com/docs/android/setup> (Accessed: 16 April 2025).

Firebase (2023b) *Cloud Firestore for Realtime Database Users*. Available at: <https://firebase.google.com/docs/firestore/firestore-for-rtdb> (Accessed: 16 April 2025).

Franken, R., Bekhuis, H. and Tolsma, J. (2023) ‘Kudos make you run! How runners influence each other on the online social network Strava’, *Social Networks*, 72, pp. 151–164. doi: 10.1016/j.socnet.2022.10.001.

Gothelf, J. and Seiden, J. (2013) *Lean UX: Applying Lean Principles to Improve User Experience*. Sebastopol, CA: O’Reilly Media.

Highsmith, J. and Cockburn, A. (2001) ‘Agile Software Development: The Business of Innovation’, *Computer*, 34(9), pp. 120–127. doi: 10.1109/2.947100.

Liao, S. and Yang, T. (2017) ‘Personalized Workout Recommendations Using Machine Learning: A Case Study of Nike Training Club’, *Journal of Sports Technology and Performance*, 29(3), pp. 42–49. Available at: <https://www.sciencedirect.com/science/article/abs/pii/S0022103117301846> (Accessed: 16 April 2025).

Müller, A.M., Alley, S., Schoeppe, S. and Vandelanotte, C. (2022) ‘The effectiveness of e-& mHealth interventions to promote physical activity and healthy diets in developing countries: A systematic review’, *International Journal of Behavioral Nutrition and Physical Activity*, 19(1), pp. 1–12. Available at: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9309778/> (Accessed: 16 April 2025).

Obregon, A. (2021) ‘Using Kotlin with Firebase: A Guide for Android App Developers’, *Medium*. Available at: <https://medium.com/@AlexanderObregon/using-kotlin-with-firebase-a-guide-for-android-app-developers-28f3fd2b57bc> (Accessed: 16 April 2025).

Rubin, K.S. (2013) *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley.

Stassen, G., Garcia, J. and Mohan, V. (2022) ‘Mobile nutrition tracking applications and user burden: A usability study’, *Journal of Nutrition*, 158(4), pp. 887–896. Available at: <https://www.sciencedirect.com/science/article/abs/pii/S0899900722001034> (Accessed: 16 April 2025).