

Submitted for the Degree of B.Sc. in Computer Science 2023/2024

## **“Full-Stack Fitness App in React Native Expo”**

Athanasios Papadopoulos

202010138

Except where explicitly stated all the work in this report, including appendices, is my own and was carried out during my final year. It has not been submitted for assessment in any other context.

I agree to this material being made available in whole or in part to benefit the education of future students.

A handwritten signature in black ink, consisting of a stylized, cursive letter 'A' followed by a long, sweeping horizontal stroke that extends to the right.

## Abstract

This dissertation aims to explore the thought process, the design, the development process and efficacy of the full stack React Native Expo fitness app. This project will address the gap in existing mobile fitness apps by integrating a combination of wide range of functionalities and creating a more fun user experience by fusing a reward-based system and gamifying fitness.

The full stack application was made in React Native which serves as the frontend, a Firebase backend for handling server-side operations such as user authentication and Firestore Database for user data storage and real-time synchronisation.

Through a user-centred design approach, the application was adapted to meet the expectations of a broad user base with a heavy shift towards user engagement backed-up by research.

## Acknowledgements

I would like to thank my supervisor Martin Goodfellow for his support and help throughout my project's duration.

I would also like to thank my second marker Emma Nicol for her valuable feedback during my presentation.

## Table of Contents

Abstract .....	ii
Acknowledgements .....	iii
Chapter 1 - Introduction .....	1
1.1 Problem Statement.....	1
1.2 Aim and Objectives .....	1
Chapter 2 - Background Research and Literature Review .....	2
2.1 Literature Review.....	2
2.1.1 Step Counters and Physical Activity.....	2
2.1.2 Leaderboards and Social Comparison.....	2
2.1.3 Achievements and Gamification .....	3
2.2 Related Work .....	3
2.2.1 MyFitnessPal: Calorie Counter .....	3
2.2.2 Strava .....	4
2.3 Conclusion .....	4
Chapter 3 - Specification.....	5
3.1 Arriving at Specification .....	5
3.2 Target Group.....	5
3.3 User stories and scenarios .....	5

3.3.1 Functional Requirements .....	6
3.3.2 Non-functional Requirements .....	6
Chapter 4 - System Design.....	8
4.1 High-Level Architecture.....	8
4.2 Design Methodology .....	9
4.2.1 Design Process and Prototype .....	9
4.2.2 User Interface Design .....	10
4.2.3 Colour Scheme and Visual Aesthetics .....	11
4.2.4 Typography and Text Entry Fields.....	11
4.2.5 Navigability and User Flow.....	11
Chapter 5 - Detailed Design and Implementation .....	13
5.1 Technologies Used.....	13
5.1.1 React Native and Implementation .....	13
5.1.2 Firebase .....	15
5.1.3 CSS .....	17
Chapter 6 - Verification and Validation .....	18
6.1 Testing Methods .....	18
6.2 Unit Testing .....	18
6.3 Black Box Testing .....	19
6.4 User Testing.....	20
Chapter 7 - Results and Evaluation .....	21
7.1 Final Product.....	21

7.1.1 Core Functionality .....	21
7.1.2 Evaluation of the process .....	26
Chapter 8 - Summary and Conclusions.....	27
8.1 Overall Reflection .....	27
8.2 Future Development.....	27
8.3 Conclusion .....	28
References.....	28
Appendix A - Detailed Specification and Design .....	30
Appendix A.1 Project Plan .....	30
Appendix B - Detailed Test Strategy and Test Cases.....	31
Appendix B.1 - Black Box Testing .....	31
Appendix C - User Guide .....	32
Appendix C.1 - How to run .....	32



## Chapter 1 - Introduction

Smartphones nowadays have become important, if not a necessity, in our daily life not just as communication tools but entertainment, photography, travel and health & fitness. The introduction of mobile health applications has changed how people track their health and exercise. Such applications take advantage of the new technology and more so the rapid technological advances to provide users with simple and tailored health management solutions at their disposal. However, the success of most of the available fitness apps is low due to their poor ability to keep users engaged.

### 1.1 Problem Statement

Studies show that fitness apps were downloaded a total of 400 million times in 2021 alone [1]. Despite the high number of downloads and potential benefits, user engagement and retention rate remain significantly low at 37% on the first day and dropping to 9% by day 28 [2]. This problem could be due to various factors such as: poor user experience, lack of content or insufficient integration into daily life and the abundance of premium features and ads.

### 1.2 Aim and Objectives

To solve this issue this project aims to develop a mobile health application that includes a variety of features: simple user interface, integration of a global leaderboard to create a compelling and fun competition for fitness that includes achievements and workout advice based on different custom user profiles.

The application will be developed using React Native with Firebase as a backend for user authentication and log in, and Firestore database for user and app storage. This app was developed and tested only for android devices. The specific objectives of the project are to:

1. Design and implement a user-friendly interface that encourages regular interaction.
2. Develop tailored features that respond to users' preferences resulting in long-term app usage.
3. Implement Firebase to build a strong backend service and Firestore database for real-time data storage and ensure seamless app performance and scalability.



## Chapter 2 - Background Research and Literature Review

This chapter explores the body of literature that already exists on mobile health applications, their functionalities, user engagement tactics and the technology used. This is needed to understand the existing marketplace of fitness apps, to identify the gaps and solve the proposed problem of user engagement. The top two fitness and sports apps worldwide in January 2024 by revenue were MyFitnessPal and Strava [7].

### 2.1 Literature Review

The literature review explores existing research in the field of mobile health applications that include the main features of the app that was developed. Three major areas of research will be examined: the effectiveness and impact of step counters on physical activity, the role of leaderboards and social comparison in motivating exercise and the impact of achievements and gamification on user engagement and behaviour change. The integration of gamification elements, such as rewards and challenges, has been identified as a key factor in enhancing user motivation and adherence to physical activity routines.

#### 2.1.1 Step Counters and Physical Activity

Step counters are one of the most integrated features on a fitness application or wearable device. It is an instrument for recording the number of steps taken by an individual. Studies have shown that using step counters can result in considerable increase in physical activity [5]. A study in 2020 discovered that adopting step-based goals can significantly boost short and long-term physical activity levels [6].

#### 2.1.2 Leaderboards and Social Comparison

Leaderboards in fitness apps provide a competitive element by ranking users based on their achievements or steps taken. Tapping into a user's natural competitiveness is an excellent way to inspire and engage the user in more physical activities. Zuckerman and Gal-Oz (2014) discovered that social comparison via leaderboards could encourage users to improve their performance to meet or exceed that of their peers. However, the usefulness of leaderboards

varies depending on individual characteristics [8]. Some of those users found inspiration in competition but others were demotivated if the challenge appeared unattainable [9].

### 2.1.3 Achievements and Gamification

The aspects of achievements and gamification aim to increase user engagement and retention rate by introducing game-like elements into the physical and activity monitoring experience. Gamification approaches such as achievements, awards and badges have shown to improve participation in physical activities (Hamari and Koivisto, 2015) [10]. Achievements can promote a sense of progress and accomplishment when a user defines a specific goal and gets rewarded for its completion [11]. Mekler et al. (2017) study show that gamification integration such as achievements contribute to users' motivation by meeting basic psychological requirements such as competence and autonomy [12].

## 2.2 Related Work

### 2.2.1 MyFitnessPal: Calorie Counter

MyFitnessPal is a complete health and fitness app that helps people achieve their nutritional and weight management goals. One of its most notable features is its extensive food database which offers nutritional information for millions of food items. This allows users to easily log everything they eat and drink throughout the day. The user can also search for an item by scanning its barcode. The software behind the app then calculates the total calorie consumption such as macronutrients, vitamins and minerals.

MyFitnessPal also allows users to track their physical activity or workout. The app's database has a range of workouts allowing users to balance their calorie intake and expenditure. The app allows for a customisable profile page based on the user's information. The software then monitors user's progress towards their selected goals.

Community features are another important component. The app includes a friendly environment which users can connect with friends and share their progress.

Overall, this app includes everything a user would want from a fitness app and its extensive features give it the rank 1 spot in the current literature review. However, there are many user reviews which have ranked this fitness app poorly due to its premium subscription and ‘annoying’ email service [3].

### 2.2.2 Strava

Strava is a versatile app created for athletes who enjoy running, hiking or cycling. The app allows users to track their activities using GPS and enables them to record their actions and analyse their performance. The app provides thorough post activity analysis such as previous performance comparisons and relative effort. Strava is distinguished for its community driven approach where anyone can share their achievements, connect with friends and participate in challenges. Strava is also compatible with numerous GPS devices and can sync with fitness apps.

Many users are complaining that some of Strava’s fan favourite features have been moved behind a paywall. Users have also noted issues with customer support, unauthorised charges and data being lost but these are only a few cases [4].

### 2.3 Conclusion

To conclude, based on the current available research the addition of step counters, leaderboards and achievements to fitness apps can have a major positive impact on user interest, engagement, retention rate, motivation and physical activity.

## Chapter 3 - Specification

### 3.1 Arriving at Specification

The specification was created using a combination of market research and literature review and study. Key user demands were found such as the need for simple approach to measure exercise, a reward/competition-based system and a user-friendly interface.

### 3.2 Target Group

The app is primarily designed for younger individuals who enjoy the competitive features of video games. While everyone is welcome, this typically includes millennials and Generation Z who are noted for their technological savvy and have a strong preference for interacting in the digital world.

### 3.3 User stories and scenarios

User stories were created to represent the typical app users and how they might interact with the app. Scenarios were mapped out to visualise and construct more precise functional and non-functional requirements.

User story 1: As a new user I want to easily register and set up my profile fast so I can start tracking my activities quickly.

User story 2: As an active user I want to accurately track my daily steps, view my progress and unlock achievements where possible so I can stay motivated to reach my fitness goals.

User story 3: As a competitive user I want to view other users progress so I can stay ahead of the competition.

User story 4: As a concerned user I want my data to be stored securely and not worry about unauthorised access.

### 3.3.1 Functional Requirements

#### User Registration and Authentication:

- Users should be able to register/log in using their email address
- The app must provide a secure authentication process
- The app must include password recovery to the designated email address

#### Step Tracking:

- The app must track the user's steps using the smartphone's sensors
- Users should be able to set a daily step goal
- Users should be able to view a summary of their activities including calories burned and distance travelled
- The app should provide progress tracking towards the step goal

#### Achievements and Leaderboard:

- The app should include a reward/achievement system that allocates points based on user's activity and steps taken
- The app should provide a leaderboard feature allowing all users to track each other's progress

#### Data Synchronisation and Storage:

- The app must ensure real-time synchronisation of user data across devices
- The app must securely store user data using cloud storage

### 3.3.2 Non-functional Requirements

#### Performance:

- The app should be responsive and capable of handling multiple user actions or any button spamming
- Step counting should be accurate and updated in real-time

Usability:

- The app must offer a simple, intuitive and user-friendly interface
- Navigation must be simple, allowing all users to access its features easily

Scalability:

- The app architecture should support scaling to accommodate an increasing number of users without throttling the performance

Security:

- User data must be securely stored

## Chapter 4 - System Design

The purpose of this chapter is to explain how the project specification's goals were designed prior to the implementation. Making sure that appearance and functionality were properly designed is crucial for development.

### 4.1 High-Level Architecture

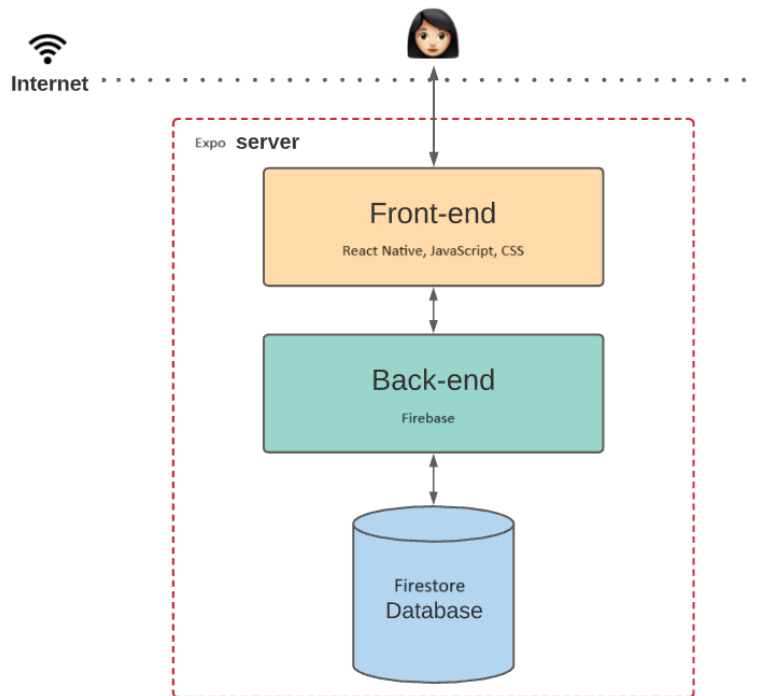


Figure 1 - High-Level Architecture of Directories

Figure 1 shows the high-level architecture of the full stack app. React Native, Javascript and CSS are used as the frontend of the app. Firebase is used as the backend and is connected to the Firestore database to store any necessary user data. A more detailed description of the design and implementation of what each part does and how will be provided in Chapter 5.

## 4.2 Design Methodology

The software development process for the app took an incremental development approach. Incremental development is distinguished by splitting the project into small and manageable increments or modules. Each of those modules can provide a part of the desired functionality. This approach can enable partial or full implementation of features and testing of other fully working modules.

### 4.2.1 Design Process and Prototype

The first step in the design process was to create a basic wireframe and a user flow chart for the whole app.

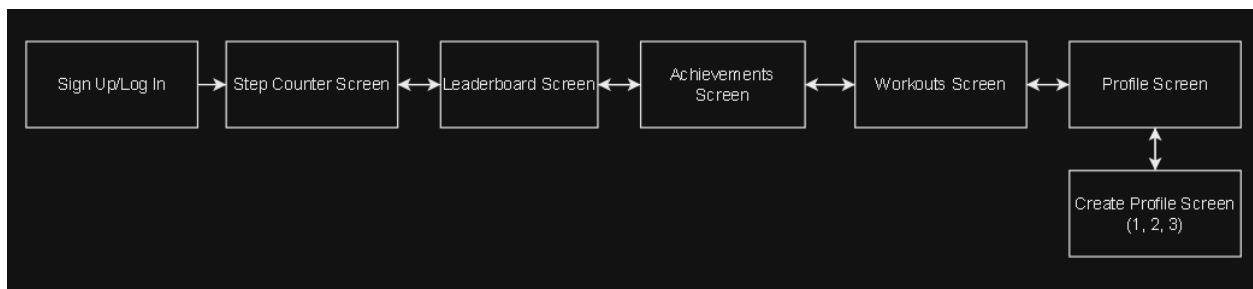
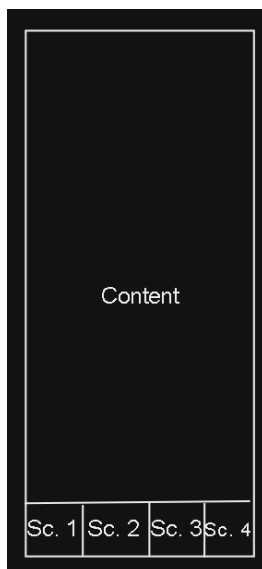


Figure 2 - User flow chart of the app



A tab view at the bottom of the app was chosen as a simple UI and navigation tool. The contents in each screen would then be displayed above the tab view. The theme remains the same throughout the app excluding the Sign Up/Log In screen.

This is a simple starting design of the wireframe. The user can navigate each screen with the tab view at the bottom. Content is displayed above the tab view. Without considering the different screens this was the initial idea and design of the app. Adding to this would be replacing the screen names, labeled as Sc. 1 - 4, with custom icons matching the description of each screen. For example, having a trophy cup icon in the tab view of the

Figure 3 - App Wireframe



achievement screen. Having icons as a part of the UI enhances user experience. If done correctly it can attract the attention of the user and increase aesthetic appeal [13].

#### 4.2.2 User Interface Design

The user interface design is critical in defining the app usability, engagement level and overall user experience.

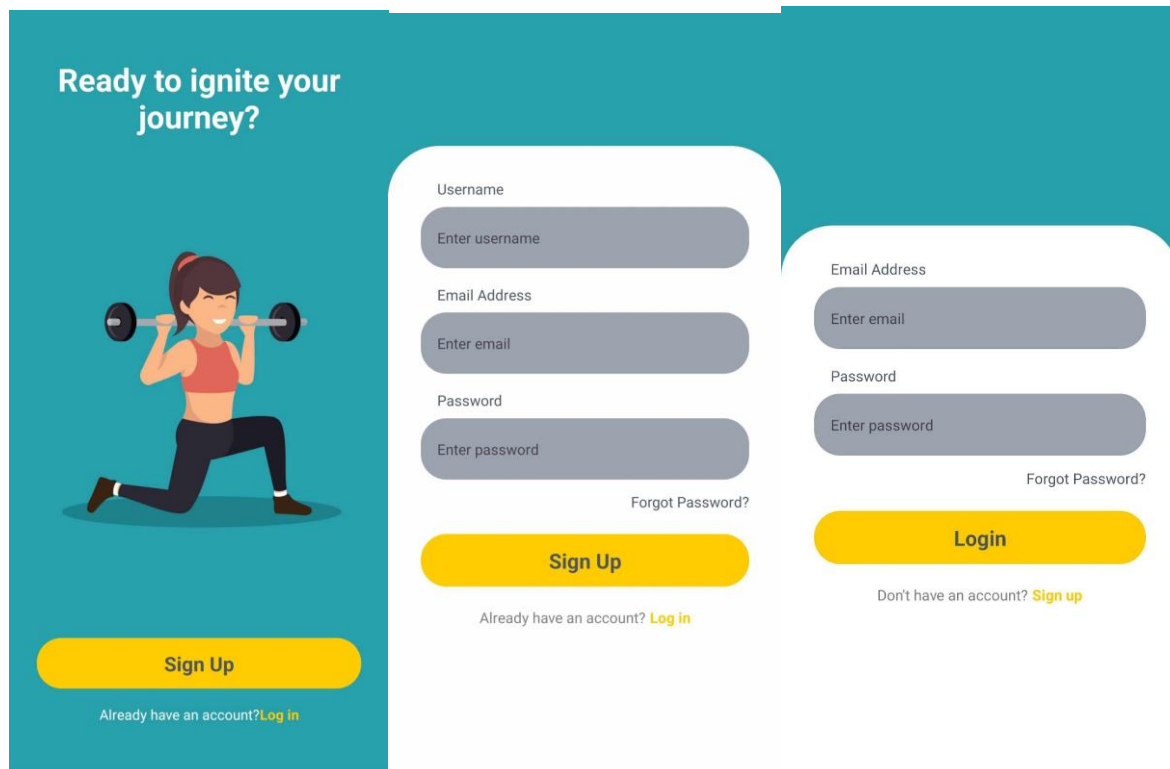


Figure 3 - Welcome Screen    Figure 4 - Sign Up Screen    Figure 5 - Log in Screen

The app's user design exhibits a user-centric approach that prioritises simplicity and aesthetics. Figures 3, 4 and 5 depict the welcome screen, sign up screen and log in screen which is the first experience a user will have with the app.

### 4.2.3 Colour Scheme and Visual Aesthetics

The turquoise backdrop used in the design matches with the yellow buttons. This minimal colour scheme reduces visual distractions and directs the user's attention to the most important features.

### 4.2.4 Typography and Text Entry Fields

The rounded corners used to separate the text fields for "Email Address", "Password" and "Username" soften the interface and give it a more personable approach. The text field placeholders offer simple instructions.

### 4.2.5 Navigability and User Flow

With all the important buttons labelled and styled yellow with CSS, the UI design efficiently leads the user through their first interactions with the app. The simplicity of the design puts an emphasis on fast access either for account creation or log in without any other distracting buttons.



Figure 6 - Tab View Navigation Inside the App

The app's tab view navigation is a popular example in design patterns that makes it simple for users to move between its many sections. The simple layout allows for an approachable and intuitive design.

The various components of the navigation bar are represented by a combination of icons and text. Each icon reflects the roles they play and is instantly recognisable. The bar includes consistent styling and text alignment. The selected tab's icon is highlighted indicating which tab is active. Another subtle but important design feature is the placement of the navigation bar. The tab bar's ergonomic placement near the bottom of the screen is beneficial for the user since it is

within the natural thumb reach when using a handheld device. Lastly, by prioritising the most important features such as 'Step Counter' and 'Leaderboards', this reflects a user-centred design focusing on the user's primary goals being first.

## Chapter 5 - Detailed Design and Implementation

Following the extensive design of the app, this chapter describes in detail the developmental phase of the project once the design phase was over.

### 5.1 Technologies Used

This project used numerous languages and technologies. The upcoming section will describe each technology used, how and where they were used and what role they serve.

#### 5.1.1 React Native and Implementation

React Native was used for frontend development. React Native was selected due to its support for iOS and Android even though this project was only focused on Android, good performance, simplified UI and large and active community.

React has many production grade-frameworks for different types of projects. The Expo framework was used for this app. “Expo is a React framework that lets you create universal Android, iOS, and web apps with truly native UIs. It provides an SDK for React Native that makes the native parts easier to use.” (React Native Website) [14] This framework offers a managed workflow which reduces setup time by abstracting many configuration stages.

Over-The-Air Updates: Expo makes it possible for updates to be sent straight to users’ smartphones without going through the app store submission process. This allows for a faster release of features and bug fixes.

Expo offers a large library of pre-built APIs for accessing device functions like the location, the camera and many more.

```
import React, { useState, useEffect, useRef, useCall
import { View, Text, TouchableOpacity, StyleSheet, I
import { Pedometer } from 'expo-sensors';
import { useTheme } from './ThemeContext';
import { MaterialCommunityIcons } from '@expo/vector
import * as Progress from 'react-native-progress';
```

Figure 7 - Library import for the Pedometer Expo sensor API..

React Native as the frontend takes care of a wide range of UI components.

**View:** The view component is the basic building block for the UI. The 'view' is used in a place of containers like divs are in web development.

**Text:** The text component is for displaying any form of text or labels in the app.

**Button:** The button component is the basic component for a touchable and customisable button to execute a line of code when pressed. Similar components are 'TouchableOpacity' and 'TouchableHighlight' which are used in this project instead of button.

**Image:** The image component is used to display images either from the network, static or saved.

```
<View style={{ flex: 1, justifyContent: 'space-around', marginVertical: 16 }}>
  <Text style={{ color: 'white', fontWeight: 'bold', fontSize: 32, textAlign: 'center' }}>
    Ready to ignite your journey?
  </Text>
  <View style={{ flexDirection: 'row', justifyContent: 'center' }}>
    <Image
      source={require("../assets/welcome.jpg")}
      style={{ width: 350, height: 350 }}
    />
  </View>
  <View style={{ paddingHorizontal: 28 }}>
    <TouchableOpacity onPress={()=> navigation.navigate('SignUp')}>
```

Figure 9 - Part of the welcome screen using the basic React UI components..

## 5.1.2 Firebase

Firebase was used for the backend development of the app. Firebase provides a wide range of tools that are essential to the functionality of the app and serves as the foundation for backend services. Cloud functions carry out backend functionality, Firestore offers real-time data storage for accessing and displaying user information and Firebase Authentication manages safe user sign-ins and registrations. Firebase was selected because of its scalability and easy integration with React Native.

```
import { initializeApp } from "firebase/app";
import { getAuth } from "firebase/auth";
import { initializeAuth, getReactNativePersistence } from 'firebase/auth';
import ReactNativeAsyncStorage from '@react-native-async-storage/async-storage';
import { getFirestore } from 'firebase/firestore';
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyD9ElHKcaYRvx1cruRacCfBSinGQ_fP38",
  authDomain: "fitnessappweb.firebaseio.com",
  projectId: "fitnessappweb",
  storageBucket: "fitnessappweb.appspot.com",
  messagingSenderId: "637823728576",
  appId: "1:637823728576:web:288b7b318e1d45220c6625",
  measurementId: "G-GNRK0E7N5G"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);

export const auth = initializeAuth(app, {
  persistence: getReactNativePersistence(ReactNativeAsyncStorage)
});

export const db = getFirestore(app); //Firestore database initilisation
```

Figure 10 - The configuration file provided by Firebase along with Firestore database initialisation.

```

import React, { useState, useEffect } from 'react';
import { View, Text, StyleSheet, FlatList } from 'react-native';
import { db, auth } from '../config/firebase';
import { collection, query, orderBy, onSnapshot } from 'firebase/firestore';

const LeaderboardsScreen = () => {
  const [userSteps, setUserSteps] = useState([]);
  const currentUserID = auth.currentUser ? auth.currentUser.uid : null;

  useEffect(() => {
    const q = query(collection(db, 'users'), orderBy('steps', 'desc'));

    const unsubscribe = onSnapshot(q, (querySnapshot) => {
      const steps = [];
      querySnapshot.forEach((doc) => {
        steps.push({ id: doc.id, ...doc.data() });
      });
      setUserSteps(steps);
    });
  });
};

```

Figure 11 - Real-time (onSnapshot) update of Leaderboards using Firestore Database by searching for the 'steps' data in the 'users' table stored in the database.

```

import { signInWithEmailAndPassword } from 'firebase/auth';
import { auth } from '../config/firebase';

tabnine: test | explain | document | ask
export default function LoginVideo() {
  const navigation = useNavigation();
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const [errorMessage, setErrorMessage] = useState('');
  const handleSubmit = async () => {
    if(email && password){
      try{
        await signInWithEmailAndPassword(auth, email, password);
      }catch(err){
        console.log('got error', err);
        setErrorMessage(err.message);
      }
    } else {
      setErrorMessage('Please enter both email and password.');
```

Figure 12 - Firebase authentication when logging in, checking the email and password using the firebase auth API.

### 5.1.3 CSS

CSS was used throughout the app for styling, guaranteeing a unified and aesthetically pleasing user experience. CSS is adaptable, simple to use and its capacity to produce designs on different device displays was why it was chosen.

```
const getStyles = (theme) => StyleSheet.create({
  buttonActive: {
    padding: 10,
    margin: 5,
    backgroundColor: '#4CAF50', // Active button color
    borderRadius: 5,
    alignItems: 'center',
  },
  buttonInactive: {
    padding: 10,
    margin: 5,
    backgroundColor: '#9E9E9E', // Inactive button color
    borderRadius: 5,
    alignItems: 'center',
  },
});
```

Figure 13 - CSS being used for button 'highlighting' when active.



## Chapter 6 - Verification and Validation

### 6.1 Testing Methods

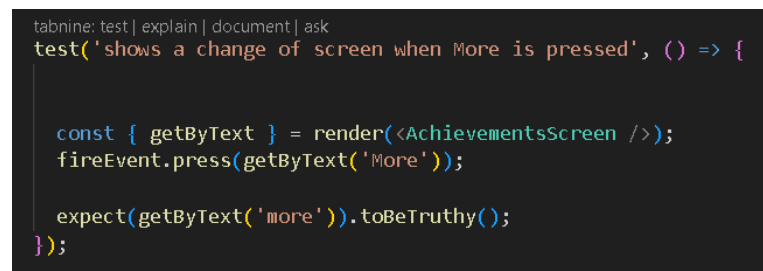
To guarantee that the finished product met its requirements and provided a smooth and enjoyable user experience three types of testing were conducted: Unit, Black Box and user.

### 6.2 Unit Testing

Unit testing was planned from the beginning to be implemented but there were configuration and setup issues with React Native that hindered the process.

Jest is a JavaScript testing framework and is the standard testing framework for React. Tests and test cases were created for the app, its UI components and functions but while implementing and running the tests configuration errors occurred.

The next two prominent testing frameworks for React are Mocha and Chai. Those as well had configuration errors due to the fact that the project imports a lot of external modules that are being run with the addition of Firebase.



```
tabnine: test | explain | document | ask
test('shows a change of screen when More is pressed', () => {

  const { getByText } = render(<AchievementsScreen />);
  fireEvent.press(getByText('More'));

  expect(getByText('more')).toBeTruthy();
});
```

Figure 14 - Proposed test for AchievementsScreen

Figure 14 shows part of the code for testing the AchievementsScreen. In theory by pressing the 'More' button, when the current screen navigates to another screen, the response expected should match a certain text in that screen, in this case 'more' and return True which will pass the test.

```
• shows a change of screen when More is pressed

Unable to find an element with text: more

<RCTScrollView>
  <View>
    <View>
      <Text>
        Level
      </Text>
      <View>
        accessible={true}
      >
        <Text>
          More1
        </Text>
      </View>
    </View>
    <View>
      <Image />
    </View>
    <Text>
      199000
      more steps to reach Level
    </Text>
    <View>
      <View />
    </View>
    <View>
      <Text>
        Distance walked
      </Text>
      <View>
        accessible={true}
      >
        <Text>
          More
        </Text>
      </View>
    </View>
    <Image />
  </View>
  <Text>
    NaN
    more metres to reach Level
  </Text>
</View>
</RCTScrollView>

48 | fireEvent.press(getByText('More'));
49 |
> 50 | expect(getByText(' more ')).toBeTruthy();
    |                               ^
51 |   });
52 |
at Object.getByText (screens/AchievementsScreen.test.js:50:10)

Test Suites: 1 failed, 1 total
Tests:       1 failed, 1 total
Snapshots:  0 total
```

Figure 15 - Outcome of the test for Figure 14

In this case, the screen did not navigate and therefore the test failed.

To combat the missing element of unit testing in testing and evaluation for UI components and other functions, Black Box Testing and User Testing were thoroughly explored.

### 6.3 Black Box Testing

Black box testing helped to validate the app's functional and non-functional requirements without diving into the internal workings of the software because it would be very difficult to create unit tests for they require input from a user which is hard to simulate in a unit test.

This test consists of four sections which are the activity tested, expected output, purpose of the test and result of the test.

Activity	Expected Output	Purpose of test	Result
Logging in	Logs in the user to their specific account	Check to see if the correct user is logged in	Pass
'Start', 'Stop' Button Spamming	Buttons work when pressed, no crash happens	To make sure the app doesn't crash	Pass
Switching tab when step counter is on	Steps are not increasing	Ensure intended working functionality	Pass

Figure 16 - Examples of Black box testing

A detailed image of all the Black Box tests is available in Appendix B.

## 6.4 User Testing

Throughout the whole developmental process of the app, user testing sessions after a new feature were performed to improve experience and make sure intended functionality was implemented.

Every feature, button or component in the app was thoroughly user tested to make sure no bugs and no unwanted actions were created while developing.

## Chapter 7 - Results and Evaluation

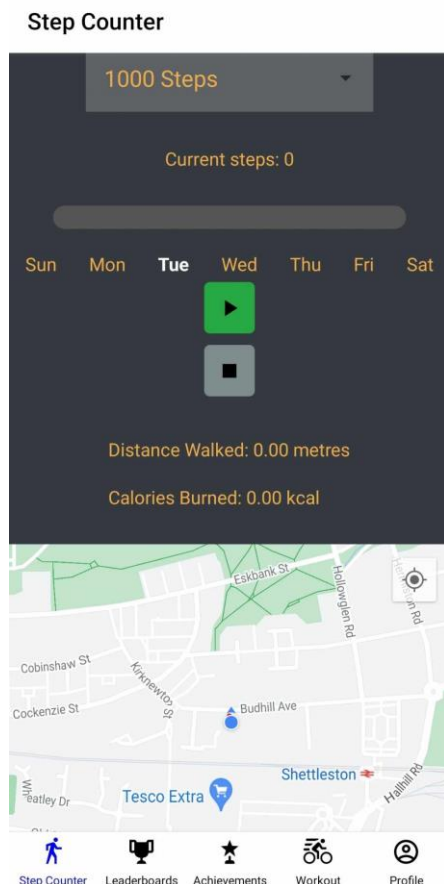
### 7.1 Final Product

This section aims to present and describe the core functionality of the project and its features.

#### 7.1.1 Core Functionality

The final product is an android app allowing users to sign up and compete with their a global leaderboard to reach their fitness goals. The application's starting page is the main step counting feature along with a live tracking map. The user can start or stop the step counter and will update their steps whenever they start walking. This in turn will update the leaderboard and achievements in real-time and show other users your progress.

As the user opens the app, they are greeted with a welcome page accompanied by the options to Sign Up or Log in. Please refer to Figure 3, 4 and 5, in section 4.2.2 for the images of those features.

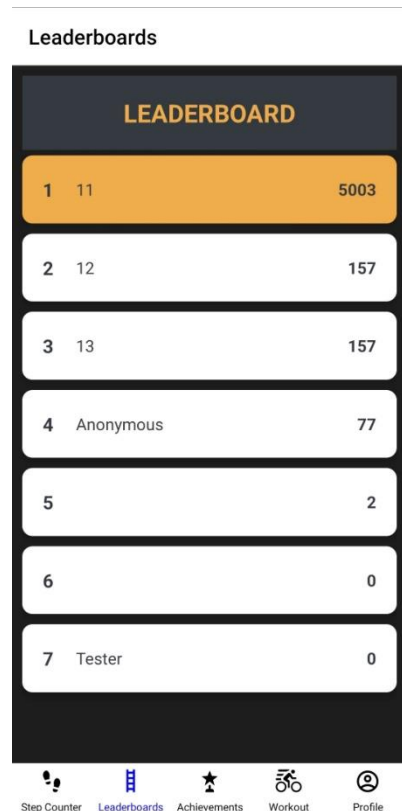


The app's first page is the Step Counter. On the top there is a UI component called the Picker. The user can choose the number of steps he wants to set as his goal. The progress bar will adjust accordingly and will start filling up once progress is being made.

On the bottom, there is a live map so the user can track where they are going and have a better understanding of distance covered. The current day is also highlighted as an extra basic feature.

Figure 17 - Step Counter Screen

The next page is the leaderboard section. There, all users' step count is displayed for everyone to compete. It updates in real time, and everyone can track each other's progress.



The first rank of the leaderboard screen is highlighted and there are rankings showing where everyone is currently standing at.

Figure 18 - Leaderboard Screen

The achievement screen tracks the current users progress and displays their achievements. Achievements are based on steps taken, which is called levels, and distance walked. There is an option to view each achievement and its unlocks separately, which takes you to a different screen. There is one screen for levels and one for distance walked as labelled in Figure (figure names). By clicking on the badge, the user can view their reward in detail.

Achievements

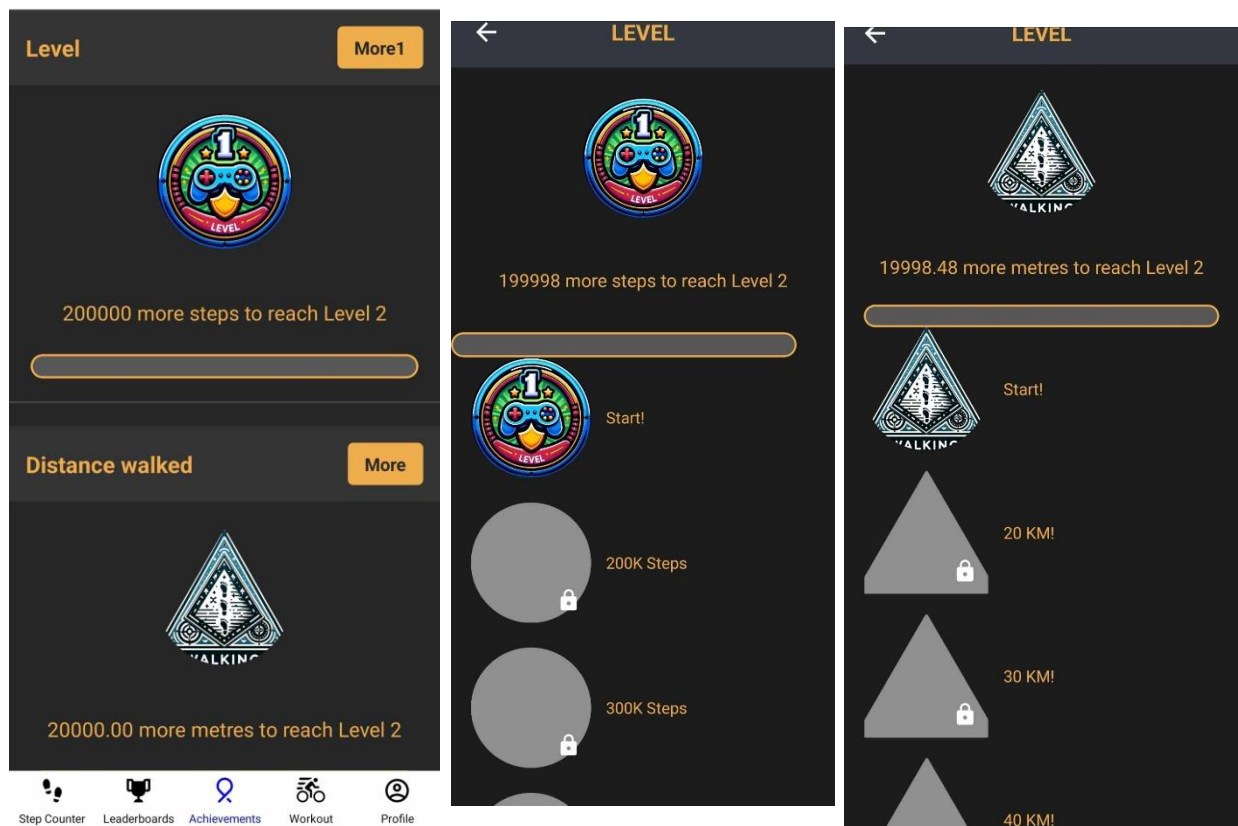


Figure 19 - Achievements Screen Figure 20 - Levels Screen Figure 21 - Distance Levels Screen

The next screen is the workout screen, and it acts as a timer for a quick run or cycle. The timer is 5 minutes, and the user can choose between running for their workout or cycling.

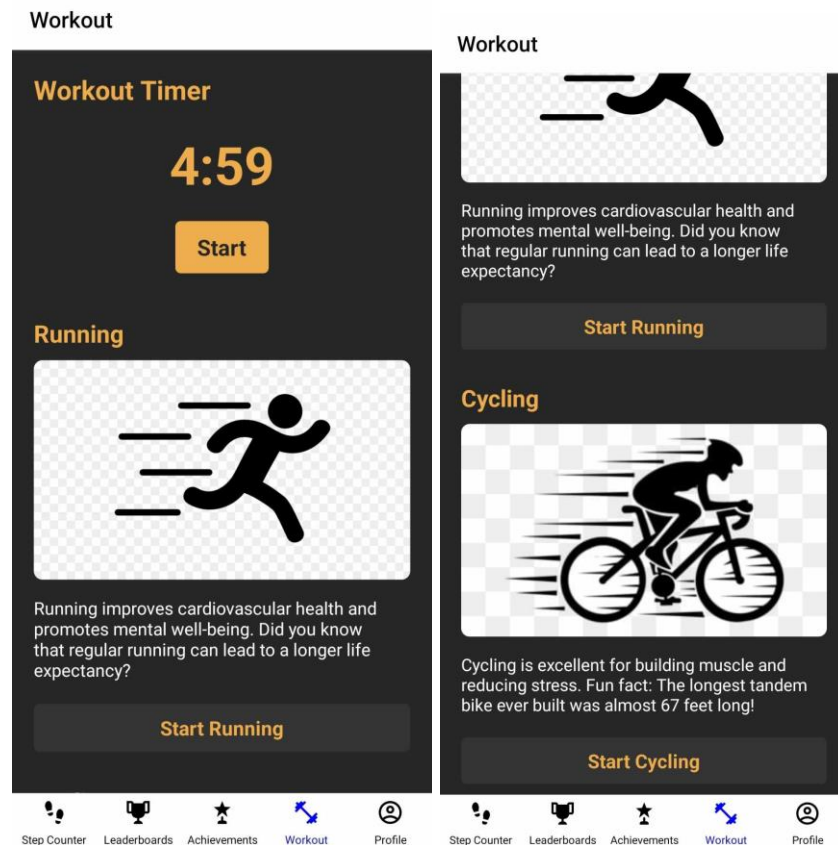
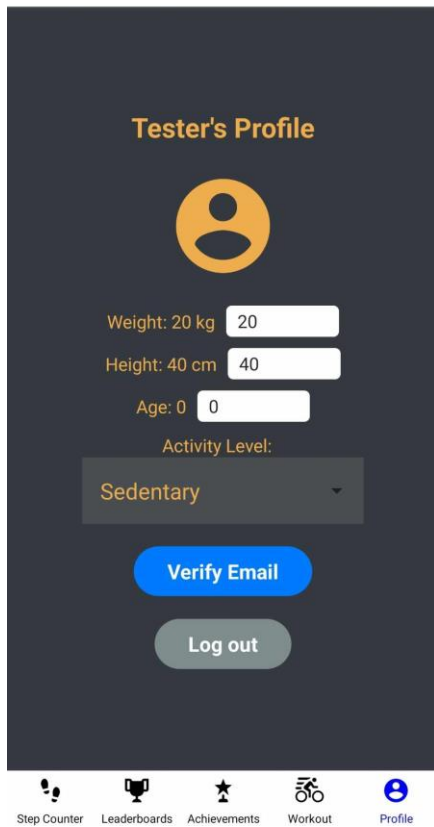


Figure 22 - Workout Screen with a scrollable view extending to the bottom



Figure 23 - Viewing an unlocked badge - Achievement Screen

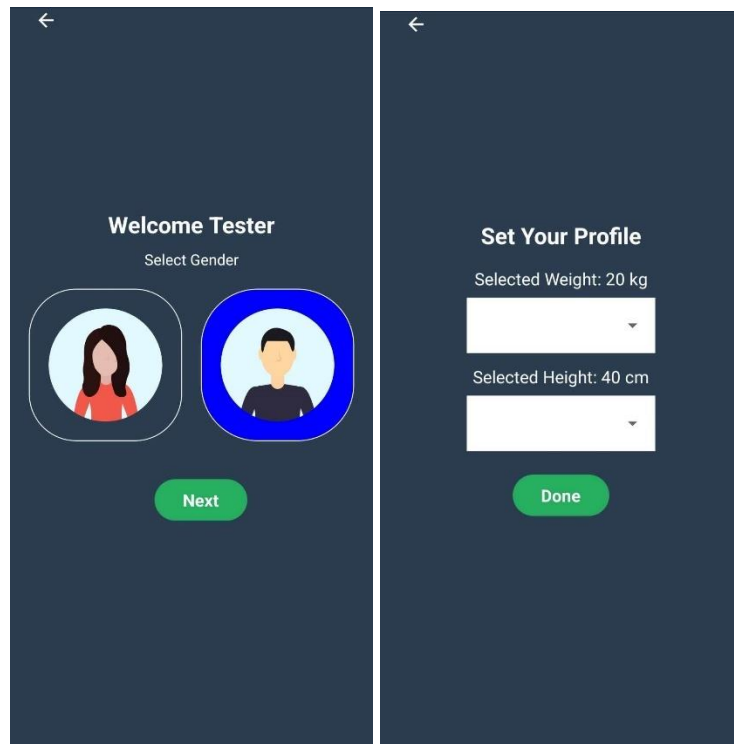
## Profile



The profile screen features a dark blue background. At the top, it says "Tester's Profile" in orange. Below this is a large orange circle representing a profile picture. Underneath the picture are three input fields: "Weight: 20 kg" with a value of "20", "Height: 40 cm" with a value of "40", and "Age: 0" with a value of "0". Below these is a dropdown menu for "Activity Level" currently set to "Sedentary". At the bottom of the form are two buttons: a blue "Verify Email" button and a grey "Log out" button. A bottom navigation bar contains five icons with labels: "Step Counter", "Leaderboards", "Achievements", "Workout", and "Profile".

Figure 24 - Profile Screen

In the profile screen, a new user can setup their profile to get some basic information about their body such as BMI and Basic Calorie Intake based on their measurements and age. A verification button is also embedded, that will send a verification email to email address that was used to sign up, to keep track of verified users and spam accounts. The user can also log out with the log out button. All their information is be stored securely in the Firestore database and will be displayed next time they log back in.



Two side-by-side mockups of the profile setup process. The left screen, titled "Welcome Tester", asks the user to "Select Gender" and shows two circular icons: a female icon and a male icon. A green "Next" button is at the bottom. The right screen, titled "Set Your Profile", shows "Selected Weight: 20 kg" and "Selected Height: 40 cm" with dropdown menus. A green "Done" button is at the bottom.

Figure 25&26 - Screen 1 & 2 of profile setup



### 7.1.2 Evaluation of the process

Overall, the project met its initial objectives but due to limited time there are limited features and content. The app would feel more complete if there was more time to add more achievements and more content and features.

From a functional standpoint the project was a success but another way to tell if the project met its objectives is to see if it covered the non-functional requirements. Everything that was set out to be completed in those requirements are met in the app.

## Chapter 8 - Summary and Conclusions

This chapter will reflect on the project experience which will include design, development and testing. Furthermore, there will be a discussion for potential future development and features.

### 8.1 Overall Reflection

The development of a full-stack fitness app was a fun experience aimed at providing a user-centric platform to motivate and engage users in a healthy lifestyle. Reflecting on the project, it is evident that the initial goals were completed thanks to the incremental development approach and constant iteration. The initial design could have been stronger with more ideas and diagrams which in resulted in the styling needing to be constantly changed.

The original mock-up of the app did not use Firebase as a backend but had implemented sqLite as a local database. Switching to Firebase for authentication and storage proved to be the correct decision but in result took a lot of time adapting to and altering the structure of the app.

A significant time was also spent on trying to implement oAuth for Google but with no success.

The Unit testing configuration errors were not ideal as critical time was spent trying to implement test cases and run the tests successfully. However, by heavily user testing the app and with high certainty, the app has no known bugs concluding that unit testing was not as crucial as originally planned.

Also, it would be a significant improvement to get user feedback from multiple users about usability and overall experience. This feedback could have informed future sprints and given more ideas throughout development.

### 8.2 Future Development

There is a lot of potential for this app and a lot of room for future development. Some ideas would be:

- Adding more content for the Achievements page
- Adding a friend list

- Adding more personalised activities
- Adding Google, Facebook OAuth
- Adding iOS cross platform

### 8.3 Conclusion

Overall, this project has been a satisfying and joyful experience. Reflecting on the journey, this project was more than simply a technical undertaking but an achievement. From the planning stage to the end, the fact that each stage was executed successfully is very satisfactory.

### References

- [1]. Curry, D. (2021, November 26). *Fitness App Revenue and Usage Statistics (2022)*. Business of Apps. <https://www.businessofapps.com/data/fitness-app-market/>
- [2]. *Health & Fitness App Report 2023*. (n.d.). Business of Apps. <https://www.businessofapps.com/data/health-fitness-app-report/>
- [3]. *MyFitnessPal is rated "Poor" with 1.8 / 5 on Trustpilot*. (2024, March 15). Trustpilot. <https://uk.trustpilot.com/review/www.myfitnesspal.com?stars=1>
- [4]. *Strava is rated "Bad" with 1.6 / 5 on Trustpilot*. (2023, June 5). Trustpilot. <https://uk.trustpilot.com/review/strava.com?page=2&sort=recency&stars=1>
- [5]. Bassett, D. R., Toth, L. P., LaMunion, S. R., & Crouter, S. E. (2016). Step Counting: A Review of Measurement Considerations and Health-Related Applications. *Sports Medicine*, 47(7), 1303-1315. <https://doi.org/10.1007/s40279-016-0663-1>
- [6]. Chaudhry, U. A. R., Wahlich, C., Fortescue, R., Cook, D. G., Knightly, R., & Harris, T. (2020). The effects of step-count monitoring interventions on physical activity: systematic review and meta-analysis of community-based randomised controlled trials in adults. *International*

*Journal of Behavioral Nutrition and Physical Activity*, 17(1). <https://doi.org/10.1186/s12966-020-01020-8>

[7]. Ceci, L. (2023, November 29). *Top fitness and sport apps by revenue 2021*. Statista. <https://www.statista.com/statistics/1239716/top-fitness-and-sport-apps-by-revenue/>

[8]. Zuckerman, O., & Gal-Oz, A. (10 2014). Deconstructing gamification: Evaluating the effectiveness of continuous measurement, virtual rewards, and social comparison for promoting physical activity. *Personal and Ubiquitous Computing*, 18. doi:10.1007/s00779-014-0783-2

[9]. Hamari, J., & Koivisto, J. (06 2013). *Social motivations to use gamification: An empirical study of gamifying exercise*.

[10]. *APA PsycNet*. (n.d.). Psycnet.apa.org. Retrieved March 26, 2024, from <https://psycnet.apa.org/record/2015-25761-034>

[11]. Cugelman, B. (2013). Gamification: What It Is and Why It Matters to Digital Health Behavior Change Developers. *JMIR Serious Games*, 1(1), e3. <https://doi.org/10.2196/games.3139>

[12]. Mekler, E., Brühlmann, F., Tuch, A., & Opwis, K. (06 2017). Towards understanding the effects of individual gamification elements on intrinsic motivation and performance. *Computers in Human Behavior*, 71. doi:10.1016/j.chb.2015.08.048

[13]. <https://www.smashingmagazine.com/author/nickbabich>. (2016). *Icons As Part Of A Great User Experience – Smashing Magazine*. Smashing Magazine. <https://www.smashingmagazine.com/2016/10/icons-as-part-of-a-great-user-experience/>

[14]. *Start a New React Project - React*. (n.d.). React.dev. <https://react.dev/learn/start-a-new-react-project>

## Appendix A - Detailed Specification and Design

### Appendix A.1 Project Plan

#### Semester 1:

##### **1<sup>st</sup> November - 11<sup>th</sup> December**

Research technology options, related apps

Research fitness topics and studies

Choose most suitable technology

Write notes, report draft

##### **11<sup>th</sup> December - 25<sup>th</sup> January**

Start designing the app, creating a mock-up

#### Semester 2:

##### **26<sup>th</sup> January - 1<sup>st</sup> March**

Working on design, implementation of additional features

Writing draft

##### **2<sup>nd</sup> March - 20<sup>th</sup> March**

Unit tests, finishing CSS

Finishing draft

Evaluation process

**21<sup>st</sup> March - 25<sup>rd</sup> March**

Finalizing draft

Fixing code quality

Analysing unit tests

## Appendix B - Detailed Test Strategy and Test Cases

### Appendix B.1 - Black Box Testing

Black Box testing was used due to the high difficulty of integrating some tests with unit testing, like the spamming of buttons or correct configurations with Firebase backend, but mostly due to unit testing being made unavailable due to configuration errors. Black box tests are to ensure that the app behaves in a way that is expected.

Activity	Expected Output	Purpose of test	Result
Logging in	Logs in the user to their specific account	Check to see if the correct user is logged in	Pass
Start, 'Stop' Button Spamming	Buttons work when pressed, no crash happens	To make sure the app doesn't crash	Pass
Switching tab when step counter is on	Steps are not increasing	Ensure intended working functionality	Pass
Tracking steps accuracy	Step counter matches step accuracy	Ensure step counter is accurate	Pass
Data Synchronisation	Data should be synchronised when changed	Ensure proper real-time synchronisation	Pass
More Button for Levels	Should navigate to Levels Screen	Ensure UI component is working	Pass
More Button for Distance	Should navigate to Distance Screen	Ensure UI component is working	Pass
Map is live tracking when moving	Should precisely track movement around the map	Ensure embedded map component is working	Pass
Rewards can be unlocked	Once level is reach, should be unlocked	Ensure firebase and rewards are working as intended for all levels	Pass
Rewards that are locked, don't show	Can't tap/see/zoom in the reward	Ensure firebase and rewards are working as intended for all levels	Pass
User can't log in with wrong details	Error message explaining error	Ensure proper Firebase authentication	Pass
Logout button, logs out the user	Logs out the user, navigates to welcome screen	Ensure proper Firebase functionality	Pass
Verify email button, send a verification email	Sends verification email to signup email	Ensure proper Firebase functionality	Pass
Setup account button	Navigates to setup screen	Ensure UI component is working	Pass
Check leaderboard values	Values should match/update corresponding to the database	Ensure proper Firebase functionality	Pass
Calculate BMI/Calories	Formula coded in the app should match expected outcome	Ensure formula is working as intended/correct coding	Pass
User Data saves	User data should real-time save/display where appropriately	Ensure database communication is onSnapshot(real-time)/working	Pass
Tab Navigator spam test	No performance issues should arise from any spamming	Ensure no crashes/no performances issues happen	Pass
Timer counts down accurately	Accurate measurement of time	Ensure proper timer implementation	Pass
Workout Screen buttons/stop button for timer work	Buttons work when pressed, timer starts	Ensure UI component is working	Pass
Error messages	Errors messages where expected	Ensure error detection is working/notify the user	Pass

Figure 27 - Black Box Test Cases

## Appendix C - User Guide

### Appendix C.1 - How to run

This app is still in development so it can only be accessed by having the source files. Make sure npm and npx are installed on the device that will be running the React Native server.

To start the server, open the terminal and change directory to the source file. Run 'npx expo start' and the metro bundler should start. Once its compiled and if using the developmental build, press "S" to switch to Expo go mode. If you have Expo go already installed on your android device, open the app and scan the QR code from the terminal. If not enable USB debugging from the developer mode settings, connect the USB to the device and it will automatically install the proper Expo go version.

If any errors due to missing modules arise, using npm or npx install them as such:

npm install (module\_name) or npx expo install (module\_name)

```
> Using development build  
> Press s | switch to Expo Go
```

Figure 28 - Switch from Development build to Expo go.



Figure 29 - This screen would signify a correct launch of the expo server for the app.