IS52018C Software Project: Final Report

Done By:

Michal Nowak, Simran Kaur Grewal, Thomas Mason, Ivan Chamukov, George Jones & Amin Ahmed (Group 5)

Supervisor: Louis McCallum

LIST OF FIGURES PAGE

LIST OF TABLES PAGE

TABLE OF CONTENTS PAGE

PAGE

1. Introduction

This chapter is a brief explanation of our applications motivation and aims. It also gives a brief description of each chapter in this report.

1.1 Motivation

Nowadays there are many different health problems around the world regardless of your age or gender. To combat the rise in health issues, the Government is recommending all age groups to live a healthy lifestyle as "increasing physical activity has the potential to improve the physical and mental health and wellbeing of individuals, families, communities and the nation. Public Health England (PHE) wants to see more people being physically active" [1] because the NHS says, "exercise can reduce your risk of major illnesses, such as heart disease, stroke, type 2 diabetes and cancer by up to 50% and lower your risk of early death by up to 30%". [2] Our motivation was driven by these reasons as we believe that keeping fit and living a healthy lifestyle is crucial and this was a way of playing our part in helping our society and communities.

1.2 Aim

Our aim is to encourage and motivate users to keep a balance of both, fitness and a healthy lifestyle by providing them with an aesthetically pleasing environment. 'EasyFit' is an all in one fitness mobile phone application developed for iOS. Its goal is to encourage users to maintain a healthy lifestyle by combining fitness and healthy living into one application. We already know that users are working hard to keep themselves fit and healthy thus they should not have to spend their time and effort switching through different applications to track their fitness and to check the progress of their goals. Therefore, we believe they deserve an all in one fitness application that does it all; cardio workouts, strength workouts and nutrition tracking.

Our application is responsible for a host of functions with the following requirements:

- The ability to sign in to make an account is key as users can only use the application if they have an account.
- Implementing user authentication when logging in or signing up or when connecting to Facebook because we are dealing with sensitive and confidential data.
- The profile page will contain important information about the user such as their height, weight, age, gender and goals. The ability to store this data to keep track of the user's fitness is vital for our application.
- Having a levelling system that displays the user's current level and 'EXP' that is obtained from the database is important. 'EXP' represents how close the user it to levelling up depending on the amount they have gained because there

is a certain amount of EXP the user needs to level up and this increases every time they complete an activity. This should be on the home page so that the user can see their current progress daily.

- Implementing a strength workout feature that allows the user to add a workout they have done and to encourage them to exercise more. This data should be saved to the database as it is necessary to keep track of the user's current fitness level.
- Implementing the 'route creation' feature is necessary because the best route needs to be calculated for the user if they are to use the cardio feature and it should be able to keep track of their calories.
- Having a cardio feature that displays the user's distance, pace, time and run date when they finish a run or walk. These final run stats should be saved to the database so that the user can view their previous fitness levels in future.
- To allow the user to add food into their diary either manually, searching through the database or through scanning a barcode. We were not able to complete this feature because it was the most challenging and due to its complexity and shortage of time, we intend to implement it in the future.

As a group, we preferred using a client-server model as the mobile phone is acting as the client, where the user data will be displayed, and the server is the mobile backend that will be used to verify users and to store their data that will be used during and after their workout. If the user wants to use their previous workout routes, the server will act as the mobile backend that will allow the user to access their workout history.

1.3 Coverage

This report will assume prior knowledge from the project proposal report presented in December 2018. The report contains 5 chapters, excluding the Introduction:

Chapter 2 - Literature Review will cover the theoretical background and research carried out when building the application using the iOS framework.

Chapter 3 - Development Record will cover the methodology used and the structure used by the group through development.

Chapter 4- Design and Implementation will show the development of the technical architecture and include the choices and justifications for the decisions made regarding the final application design.

Chapter 5- Formative Evaluation and Quality Assurance will include the user testing done while developing the application, the testing flow implemented with the results included and the evaluation after the testing phase.

Chapter 6 - Summative Evaluation will describe the evaluation methodology used which includes the methods, results, and conclusions for our application.

2. Literature Overview

This chapter reviews the foundations of our project and it describes the theoretical background and research carried out when building the application using the iOS framework.

2.1 Swift and Xcode

During the development, we used Swift 5 to create our iOS application. "Swift is a general-purpose, multi-paradigm, compiled programming language developed by Apple Inc". [3] The foremost reason we decided to use Swift as the programming language is because we chose to only focus on creating an iOS application. This was because there is only a limited amount of iOS devices thus we can assure customers that our application and user experience delivers the excellent experience and functionality we intend it to. This was very important to us because we believe first impressions of an application can make or break a product like this, if the application is not user friendly, customers are more likely to stop using the application.

During development, it started off being a challenge to use Swift because none of the group were familiar or had never programmed with Swift prior to this project. This meant we had to learn the language while we were programming the application. To help with the transition to using a new programming language, we decided to use a user friendly and helpful programme to design the application. We decided to use Xcode as the integrated development environment (IDE) this offered a suite of software development tools that helped to guide the team throughout the development. [4] Xcode makes it simpler to build an application this is because the IDE contains a storyboard feature which allows us to design the user interface by just dragging and dropping objects, text boxes etc. (see Figure 1). This was very helpful as it allowed the team to see exactly what the application will look like throughout the development process and not just when the application has been finished. Another great feature is the simulator, this allowed the team to choose an iOS device such as an iPhone X, and then view a simulation of how the application would run on such a device. This allowed us to test the application thoroughly to make sure the application worked as intended on all iOS devices. Lastly, Xcode has git implemented within it (see Figure 2) this made it very easy for our group to work on the same project simultaneously without any issues. In addition, if there were ever any merge conflicts, Xcode highlights exactly where the merge conflict is and shows the user the two different files side by side and line by line which ensured that our work was never accidently lost.

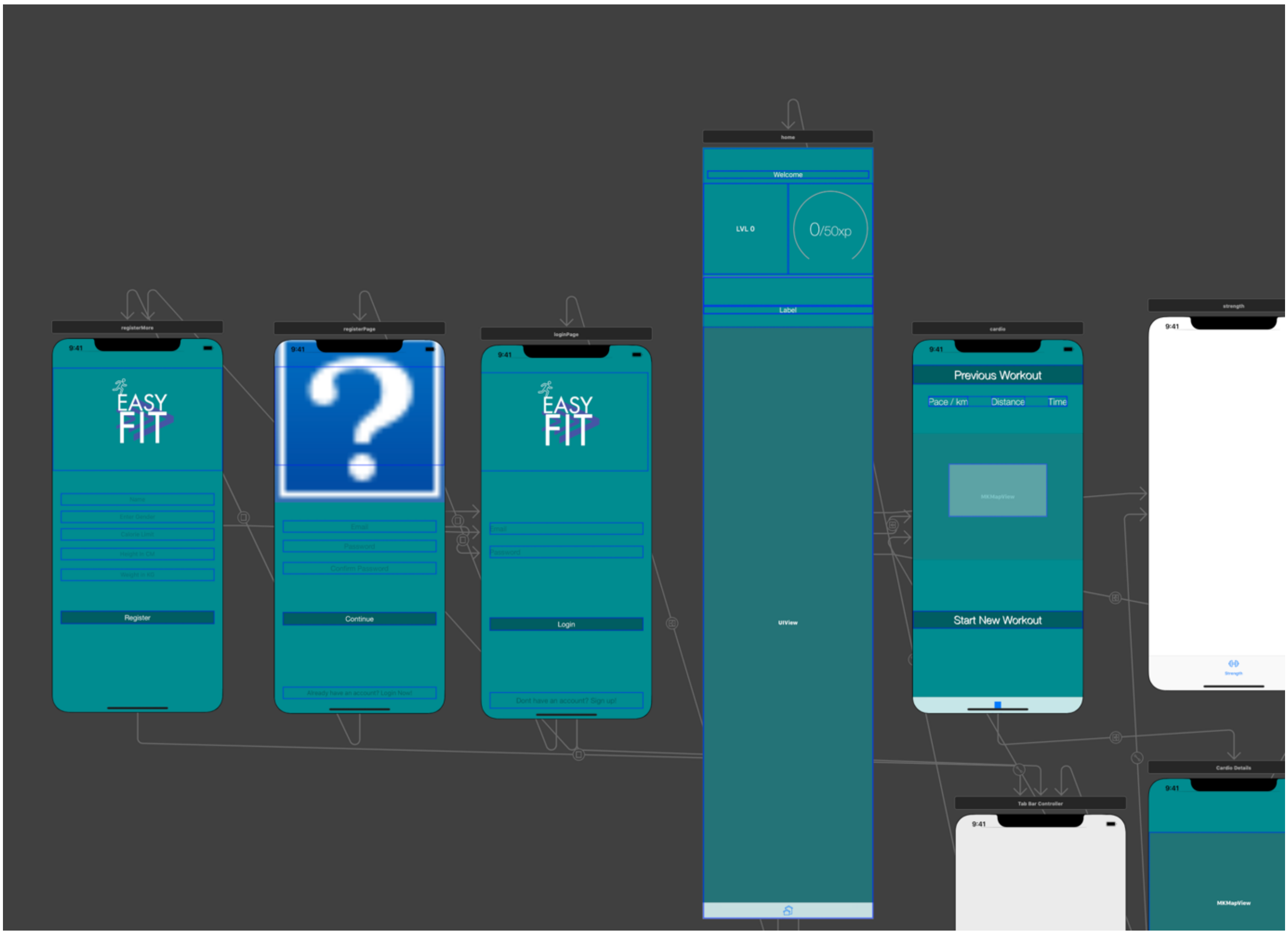


Figure 1: The figure depicts the storyboard feature within Xcode

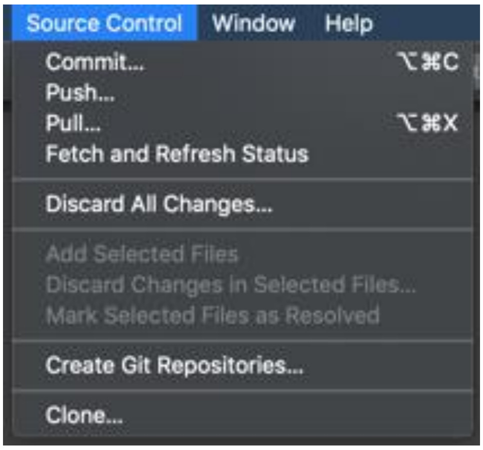


Figure 2: The figure depicts git implementation within Xcode

2.1.1 CocoaPods

CocoaPods is an application level dependency manager which we used to connect the database, Firebase, to our application. This also allowed us to implement the Facebook

Software Development Kit (SDK) which allows users to connect, login and register using their Facebook account. [5]

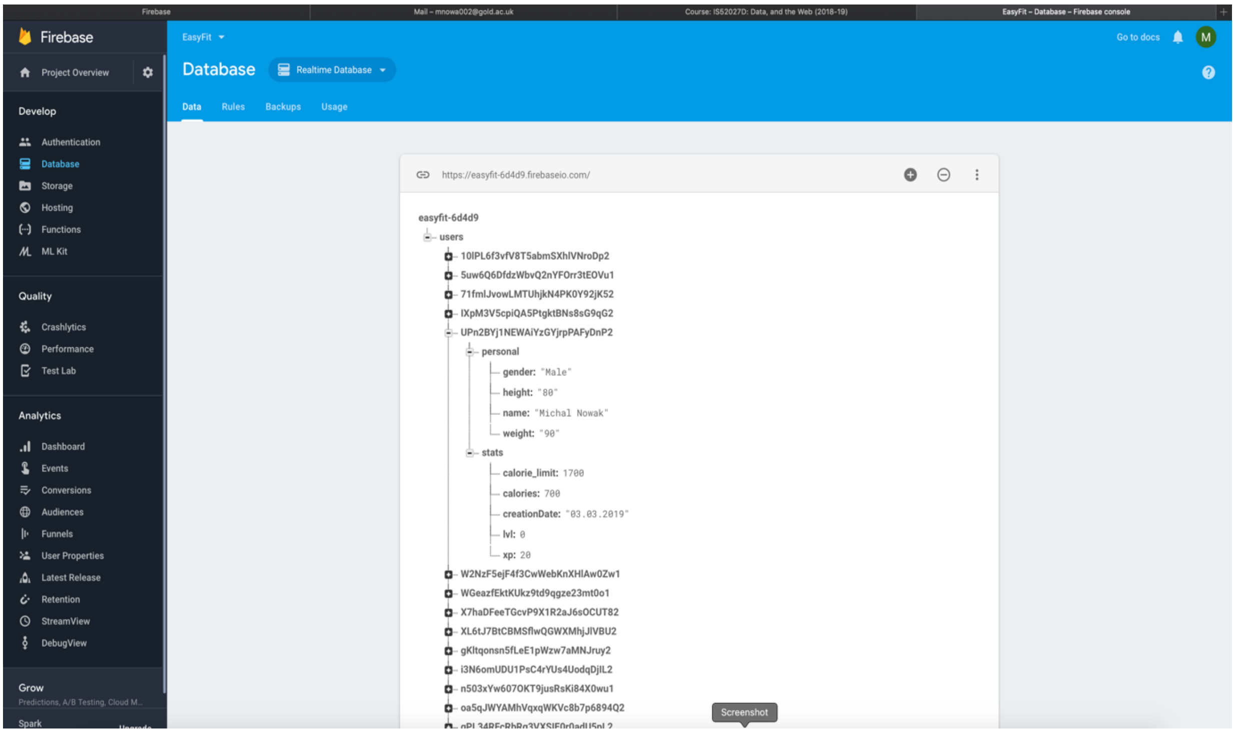
2.2 Firebase Realtime database and backend

For the backend and database, we used Firebase. Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011and was later acquired by Google in 2014. [6] Using Firebase simplified the backend process dramatically because the backend was all designed and created ready to use which allowed us to put more focus and time into the frontend. Firebase has multiple extremely useful features which normally, if we had to create our own backend we would not have had time to create. One of the services it offers is called ‘Firebase Auth’ which authenticates users using only client-side code. Firebase handles the authentication, so we do not have to be concerned about privacy and data issues such as hashing the passwords and then storing them. This is because all the safety measures are handled by Firebase themselves.

2.2.1 Realtime Database

The Realtime database provides application developers an Application Programming Interface (API) that allows data to be synchronized across clients and stored on Firebase's cloud. [6] This feature is beneficial as it allowed us to save a lot of time and meant we did not have to create the database. In addition, it allows us to save money by not being forced to rent a server to run and store the database in. Another benefit of using Firebase is that expansion in the future is very simple and intuitive, if we need more storage we just upgrade the payment plan. Furthermore, Firebase provides multiple useful analytics such as application usage and user engagement.

Owing to Google being the owners of Firebase, they are also able to provide a feature called ‘Crashlytics’. Crashlytics is a Google-owned Massachusetts-based software company. Crashlytic’s main product is a software development kit for crash reporting. [7] This software development kit is used to create detailed reports of errors in an application, any errors that occur are grouped into clusters of similar stack traces and ordered by the severity of impact on the application users. This allowed our group to identify bugs and problems much easier and faster resulting in faster bug fixes thus providing an enhanced user experience.



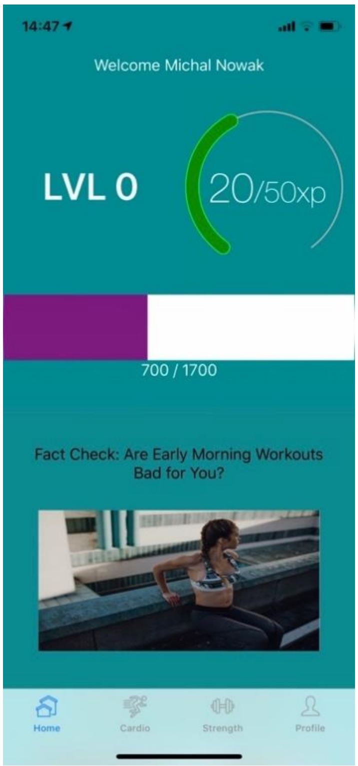
https://easyfit-6d4d9.firebaseio.com/

```
easyfit-6d4d9
 users
   10IPL6f3vfV8T5abmSXhIVNroDp2
   5uw6Q6DfdzWbvQ2nYFOrr3tEOVu1
   71fmlJvowLMTUhjkN4PK0Y92jK52
   IXpM3V5cpiQA5PtgktBNs8sG9qG2
   UPn2BYj1NEWAiYzGYjrpPAFyDnP2
     personal
       gender: "Male"
       height: "80"
       name: "Michal Nowak"
       weight: "90"
     stats
       calorie_limit: 1700
       calories: 700
       creationDate: "03.03.2019"
       lvl: 0
       xp: 20
   W2NzF5ejF4f3CwWebKnXHlAw0Zw1
   WGeazfEktKUkz9td9qgze23mt0o1
   X7haDFeeTGcvP9X1R2aJ6sOCUT82
   XL6tJ7BtCBMSflwQGWXMhjJlVBU2
   gKltqonsn5fLeE1pWzw7aMNJruy2
   i3N6omUDU1PsC4rYUs4UodqDjIL2
   n503xYw607OKT9jusRsKi84X0wu1
   oa5qJWYAMhVqxqWKVc8b7p6894Q2
```

Figure 3: The figure depicts the firebase interface and the Realtime database infrastructure

2.3 Gamification

In the initial design stage of our project we decided to gamify our application. Gamification is the application of game-design elements and game principles in non-game context. [8] By gamifying the application, we hope to make the application more enjoyable for our users. We gamified ‘EasyFit’ by adding a levelling system. The way this worked was, each user has a level and can gain experience points by completing a variety of tasks. These tasks can range from simply logging in daily to complete a workout and the more difficult the task, the more experience points gained. As soon as the user accumulates the correct amount of experience points needed for the level up, they will level up and their progress and achievement will be displayed on a leader board. Gamifying will motivate existing users to continue using the application by providing a progressive competitive experience in addition to encouraging new users to try the application and in hand improve their general well-being. Out of 19 different papers which reported empirical evidence on the effect of gamification on health and well-being, 59% reported gamification had a positive impact on health-related behaviours. [9]



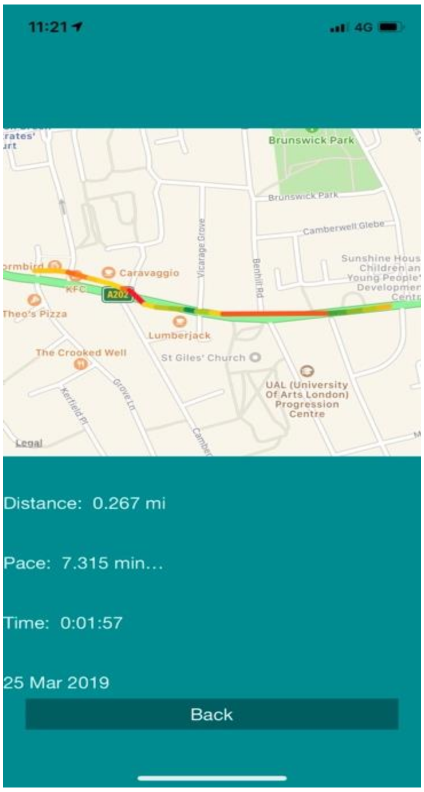
Welcome Michal Nowak

Fact Check: Are Early Morning Workouts Bad for You?

Figure 4: This figure depicts the use of gamification in our application

2.4 Fitness tracking using technology

There are many ways of tracking fitness with the use of technology whether it is using wearable technology such as a watch or through a mobile device itself. The current version of our application makes use of the users iPhone's GPS to track the location of the user when a cardio workout has been started. Using the data from the GPS, our application can calculate the pace, distance and map the route of the workout. This data is then saved to the device and displayed to the user, allowing the user to view their progress and potentially get a slight but still useful indication on their general health. For example, if the user is unable to keep a steady pace for longer than 5 minutes this may be an indication that they are unfit.



Distance: 0.267 mi

Pace: 7.315 min...

Time: 0:01:57

25 Mar 2019

Figure 5: This figure depicts the use of GPS to track a run in our application

3. Development Record

This chapter describes the way we divided the tasks amongst ourselves to work collaboratively throughout the development of our application and the development methods used by our group.

3.1 Team Structure

In the first week of term 2, as a group, we decided the structure of communication and collaborative work to ensure any possible issues were minimised. Then, each group member was assigned a task to start the development of the application. This included learning Swift which is a powerful and intuitive programming language for macOS, iOS applications [10], researching for a database for the application and creating a logo for the application. The reason we divided the tasks amongst ourselves was to ensure that everyone worked in a different area throughout the development so that there were fewer chances of collisions.

When developing through the frontend and backend areas, there was slight collisions, and these were resolved through constant communication and updates between the members working in that area. Secondly, when a group member had an issue we ensured that the other group members were available to help, and this was a good way to make sure the development continued without any interruptions.

3.2 Methodology

The development process included several stages that we considered as the application's evolutionary lifecycle [11] to tackle any issues that could have possibly occurred. We considered the diagram (see Figure 6) to analyse our planning procedure.

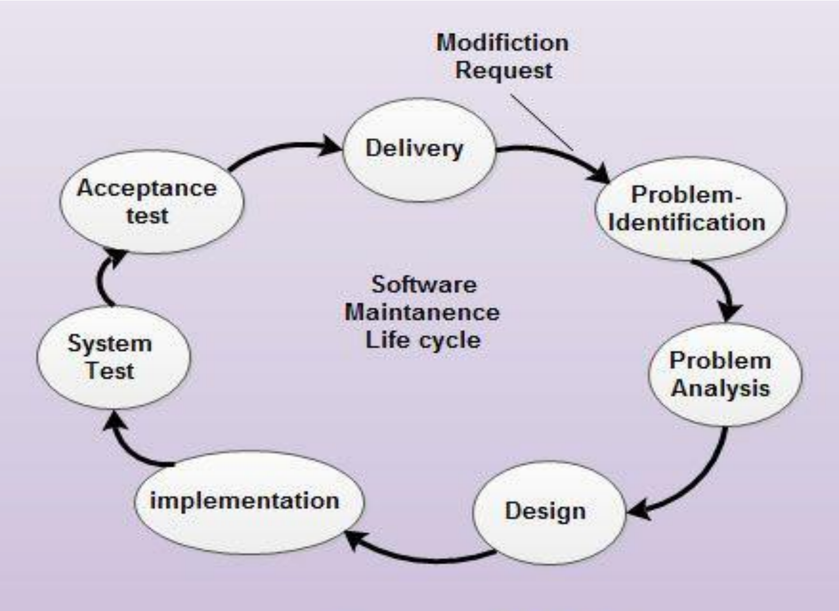


Figure 6: Software Evolution Cycle [12]

Following the software evolution cycle (see Figure 6) we defined the features of our application and we used our prototypes to sketch our application. Then, for the implementation phase, the group members that were confident in programming would implement the features for our application. For the evaluation and testing phase, we decided to have iterations for our application and consider any bugs to ensure the functionalities of the application met our expectations.

Initially, to manage the development, we used the agile, Scrum technique. In the 'Trello' project management system, we created a work area where a Kanban board was used to keep track of the features to be added, with the appropriate priorities defined (see Appendix A). Our Scrum followed the 'Plan-Do-Measure-Adapt' cycle, which had the following stages: the Kanban board was divided into 4 development sections; Backlog, Development, Testing and Complete. Each section had 'Sprints' that each member chose to do, and each sprint had an estimated time of completion so that it was completed before the next one was added. Some tasks required more time, so they suffered a pushback. Even so, every task was added to the Sprint Backlog and checked against the Product Backlog. After completing a sprint, as a group, we reviewed the project's progress and assessed the outcome every week.

The benefits of using the Scrum technique was that we were able to divide the tasks into manageable sprints and each member was a leader in their development area which meant that they could add tasks and update their own task. This helped in preventing situations where the group leader would have had to assign tasks and time limit to each member expecting them to be done in the given time. Thus, increasing the efficiency of the project's development. In addition, the transparency and visibility in Scrum gave each member the opportunity to know the current state of the development which helped us accurately identify issues and predict the project’s progress. Distributing the workload evenly amongst the group was a decent way of meeting the deadlines (which was at the end of every sprint) without causing any pressure amongst the group members. As a group, we did need a specific approach to communication because having regular meetings was one of the principles of the agile development. This did prove to be an issue for our group because meeting in person was not always possible as it depended on everyone’s availability. To prevent decreasing our project’s productivity, we would have the meetings via Discord, a popular VoIP software or Skype for discussing the progress of each group member, reporting any issues and asking any questions for the next stages of development. This ensured that the whole group understood the project status and it was an effective way of working on any problem that was affecting the efficiency of our project’s development.

In addition to using the Scrum technique, to manage our application’s source code, we chose to use ‘Git’ because it acted as a repository manager that allowed us to access, manage, keep track of any changes and share the source code throughout the implementation of our application amongst the group. [13] The group leader created a ‘TODO list’ that showed the tasks of development for each member. As a group, we could see changes made to the source code and work on the recently saved changes and the repository was regularly managed by the group leader. This was a productive way of working because each member was able to update their tasks and record their activities weekly.

4. Design and Implementation

This chapter describes the design, structure and models used when developing our application, providing an overview of the final design. It also shows the process of using the Cocoa Touch Framework in Xcode, the development and design methodologies discussed in Chapter 3: Development Record, for the implementation process of our application. In addition, it addresses any changes from our initial application ideas and the significant decisions made as a group.

4.1 Design overview & models

In the following service model diagram, we represent how the user and the server (our application) communicate with each other in certain use case scenarios. The login procedure shows how we deal with any errors or failures. Similarly, the other use case scenarios show the main and important features of our application and we show how the server reacts to the user's choice or input.

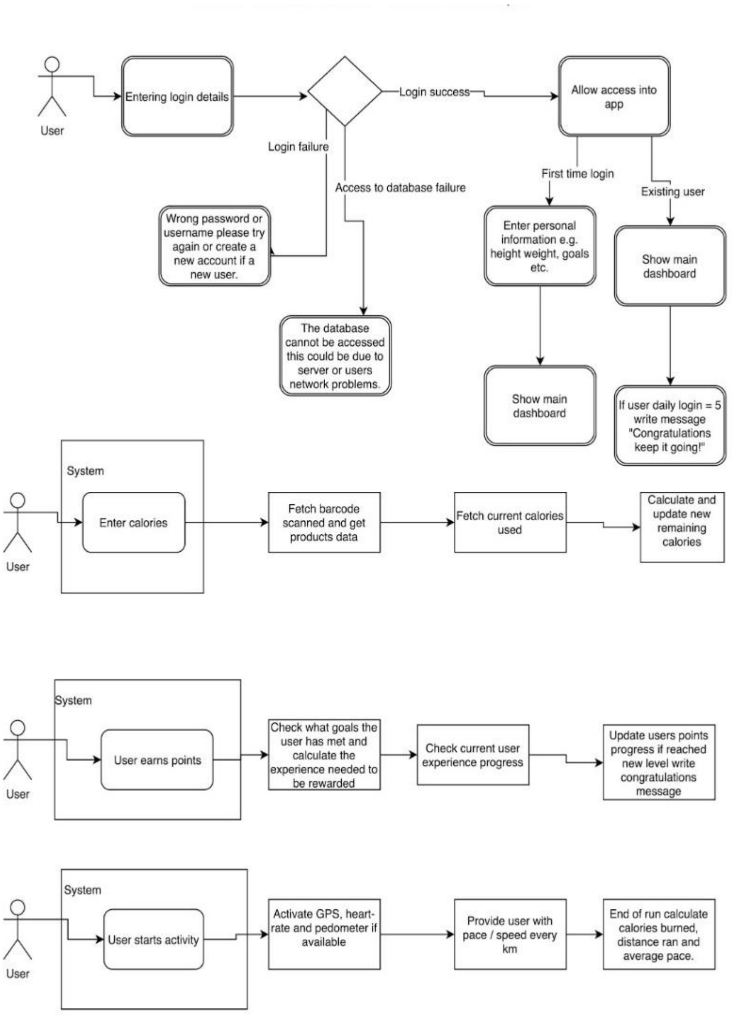

User
Entering login details
Login success
Allow access into app
Login failure
Access to database failure
First time login
Existing user
Wrong password or username please try again or create a new account if a new user.
Enter personal information e.g. height weight, goals etc.
Show main dashboard
The database cannot be accessed this could be due to server or users network problems.
Show main dashboard
If user daily login = 5 write message "Congratulations keep it going!"
System
User
Enter calories
Fetch barcode scanned and get products data
Fetch current calories used
Calculate and update new remaining calories
System
User
User earns points
Check what goals the user has met and calculate the experience needed to be rewarded
Check current user experience progress
Update users points progress if reached new level write congratulations message
System
User
User starts activity
Activate GPS, heart-rate and pedometer if available
Provide user with pace / speed every km
End of run calculate calories burned, distance ran and average pace.

Figure 7: Service model showing the interaction of a user with our application

The overall design of our application has not had many changes since the proposal report. When we finished the prototype, we decided to make a questionnaire and get feedback from our stakeholders (Appendix B). 83% of stakeholders said they liked the design and 81% of stakeholders said the features in the prototype were better than the current fitness application they are using. This gave us the confidence to stick to our design and begin implementing the features. However, we were not satisfied with the colour scheme and as a group, after experimenting with colours we concluded to use the colour teal instead of red which was favoured in the questionnaire. In addition, our initial design used an image from the internet on our login/signup page. As a group, we decided not to use online images to avoid copyright issues and therefore, we decided to create our own logo and use that because this would represent our application, 'EasyFit'.

Initially, we decided to create our own articles for the RSS feed but due to a limited time frame we opted to use runtastic's articles [14] instead. In the future we plan to create our own articles, as this would be a great way to generate ad revenue and take sponsored articles which would allow us to keep the application free for our users.

4.2 Database Design

The Entity Relationship diagram below, represents the structure of the database. The *users* table has a primary key *UserID* which uniquely identifies each user. It has 2 foreign keys *PersonalID* and *statsID* which are connected to the *Personal* and *stats* tables.

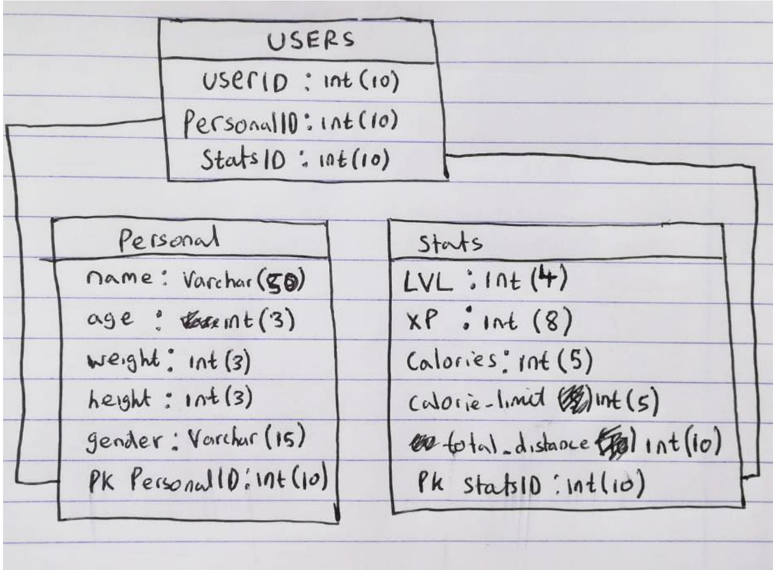


Figure 8: ER diagram

4.3 Implementation

4.3.1 Frontend implementation

Chapter 2: Section 2.1, describes the use of Xcode and Swift in details during the development stage.

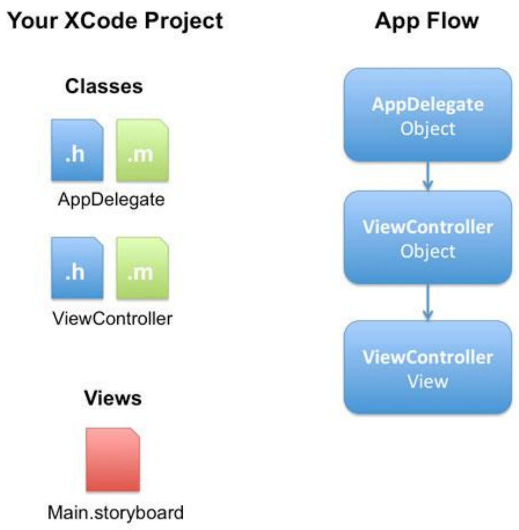


Figure 9: Application flow [15]

We decided to use Xcode as our integrated development environment when implementing our frontend. It uses the Cocoa Touch Framework which is compatible with Swift. We decided to choose Swift over Objective-C, due to its easier learning curve and it being the modern alternative. Swift also tends to be faster, requires less code and has stricter syntax which prevents errors and improves readability.

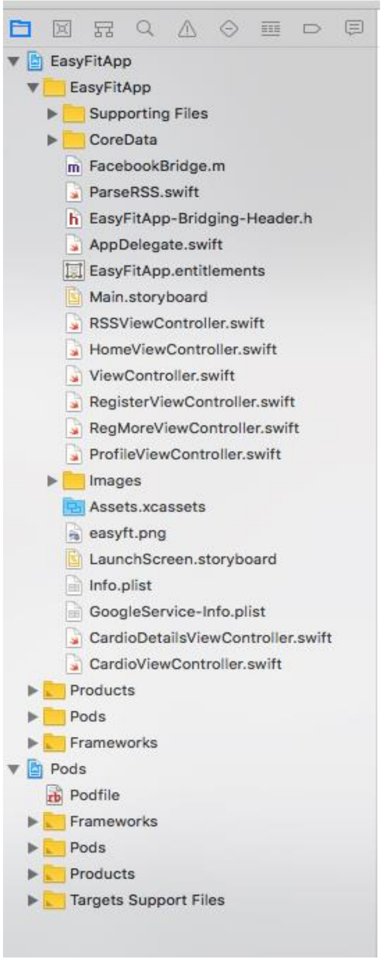
The 'EasyFitApp' Xcode project contains '**EasyFitApp.xcodeproj'**, which is at the root of the project and this contains the settings and the build path required to run the application. As shown in Figure 9, the Xcode Project contains an 'AppDelegate' and a 'ViewController' class. The 'AppDelegate' class connects iOS to the application, which means it acts as a main entry point of the application. One way we used this class in our application was requesting permission from the user to share their location when the application starts. The 'ViewController' class allows us to create different views/pages for our application using Swift and this is essentially the user interface.

The storyboard is a visual editor, which allowed us to lay out and design the user interface of our application for each 'ViewController'. It let us drag and drop objects such as buttons, textboxes and progress views and position them. Each item on the storyboard had properties which could be changed in the storyboard or view controllers. In addition, we could use constraints which allowed us to make the page dynamic, so it fit all mobile phone screen sizes. Segues allowed us to switch to another ‘ViewController’, which was used primarily in the navigation.

Throughout the implementation, we needed to use third party libraries. To be able to import these libraries, we needed to use Cocoa pods, which is a dependency manager for Swift projects. After installing Cocoa pods, to add a library to the project we simply had to add the pod name to the ‘PodFile’ and then execute **pod install** in the command line. Finally, we imported the library to all view controllers that required it, and this let our application have access to the library.

4.3.1.1 EasyFit’s Project Structure

Xcode’s file system uses the following directory hierarchy. *Figure 10* shows the structure of our project.



```
▼ EasyFitApp
  ▼ EasyFitApp
    ▶ Supporting Files
    ▶ CoreData
      FacebookBridge.m
      ParseRSS.swift
      EasyFitApp-Bridging-Header.h
      AppDelegate.swift
      EasyFitApp.entitlements
      Main.storyboard
      RSSViewController.swift
      HomeViewController.swift
      ViewController.swift
      RegisterViewController.swift
      RegMoreViewController.swift
      ProfileViewController.swift
    ▶ Images
      Assets.xcassets
      easyft.png
      LaunchScreen.storyboard
      Info.plist
      GoogleService-Info.plist
      CardioDetailsViewController.swift
      CardioViewController.swift
  ▶ Products
  ▶ Pods
  ▶ Frameworks
▼ Pods
    Podfile
  ▶ Frameworks
  ▶ Pods
  ▶ Products
  ▶ Targets Support Files
```

Figure 10: EasyFit’s project structure

Resources:

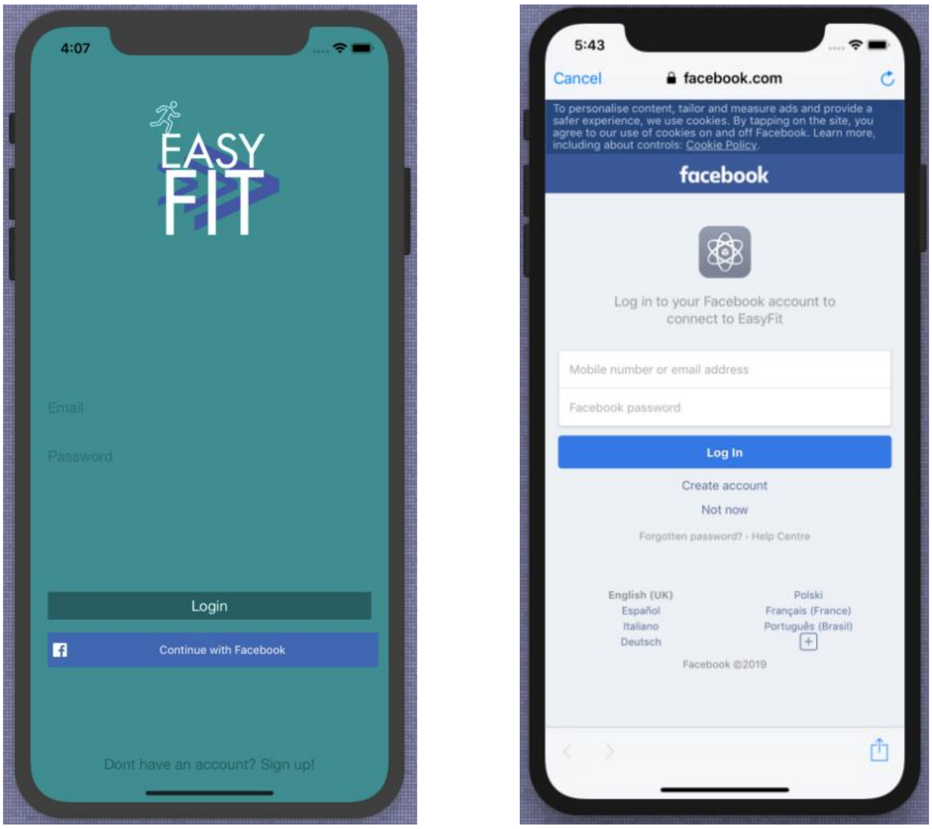
- EasyFitApp\EasyFitApp\CoreData – This contains all the view controller Swift files and the storyboard.
- EasyFitApp\EasyFitApp\Images – This contains all the images used for our project.

- Pods\PodFile – Describes the dependencies of the project.
- Pods\Pods – This contains the installed libraries/frameworks

4.3.1.2 Application Features

1. Login page

The login page was the initial view that was created. It allows the user to login with an account by using a registered email and password, and these details are checked using the Firebase authentication. The user also has the option to connect to the application using Facebook because we used Facebook’s login SDK (see Figure 11).



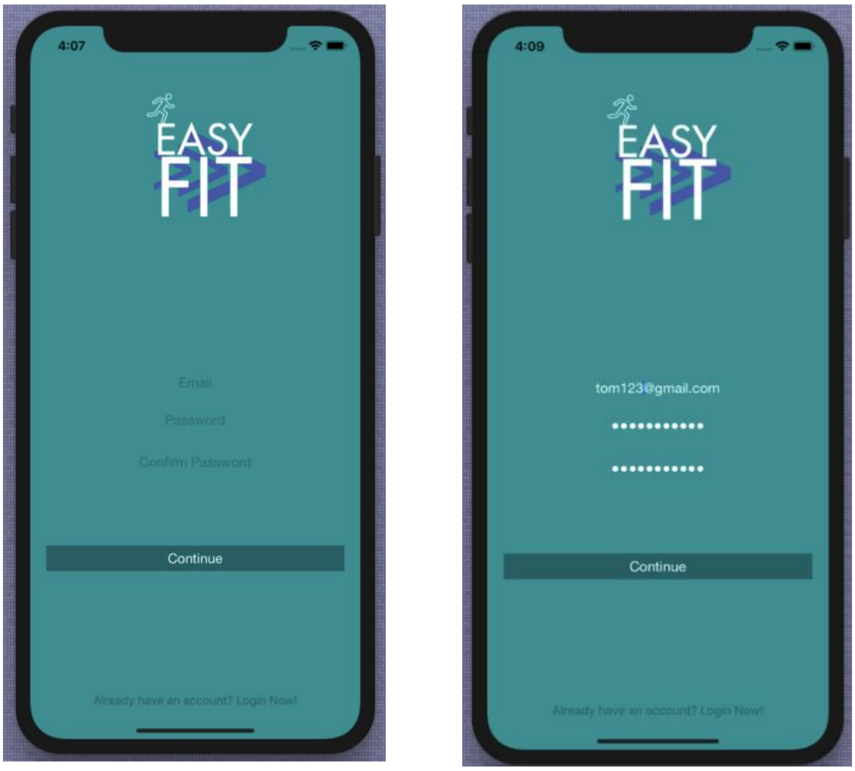
To personalise content, tailor and measure ads and provide a safer experience, we use cookies. By tapping on the site, you agree to our use of cookies on and off Facebook. Learn more, including about controls: Cookie Policy.

Log in to your Facebook account to connect to EasyFit

Figure 11: The login page (email and password) and the second image shows the option to connect to the application with a Facebook account

2. Signup page

The signup page consists of two views, the first contains the email and password entries required by the user. The user must confirm the password to ensure the passwords match as a word might have been mistyped (see Figure 12). The second view contains entry fields, this allows the user to enter their personal information (see Figure 13).

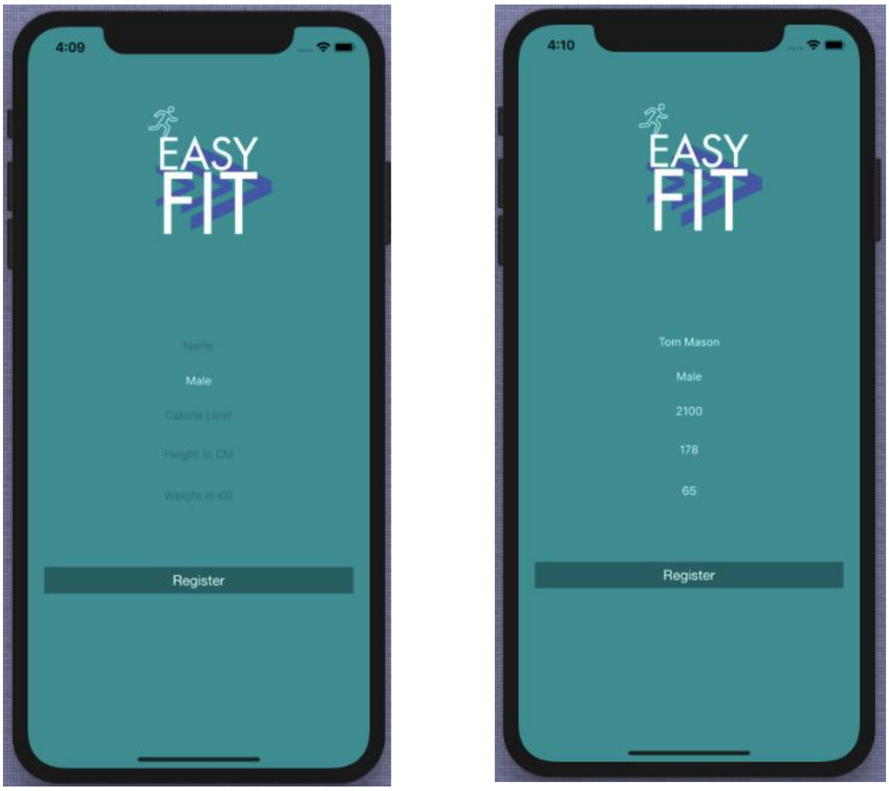

4:07
EASY FIT
Email
Password
Confirm Password
Continue
Already have an account? Login Now!
4:09
EASY FIT
Continue
Already have an account? Login Now!

tom123@gmail.com

•••••••••••

•••••••••••

Figure 12: Signup page showing the Email/Password



Male

Tom Mason

Male

2100

178

65

Figure 13: Signup page showing the user’s information

3. *Home page*

The home page was the next view implemented after the signup page. It includes a simple and tidy layout, with a levelling system which displays the user's level and current 'EXP' progress obtained from the database. It also includes a calories progress bar, which shows how many calories the user has used for the current day which is retrieved from the database (see Figure 14). Using a sub view, the home page also includes a scrollable RSS feed which shows the most recent articles with media, which when clicked, redirects the user to the articles page (see Figure 15).

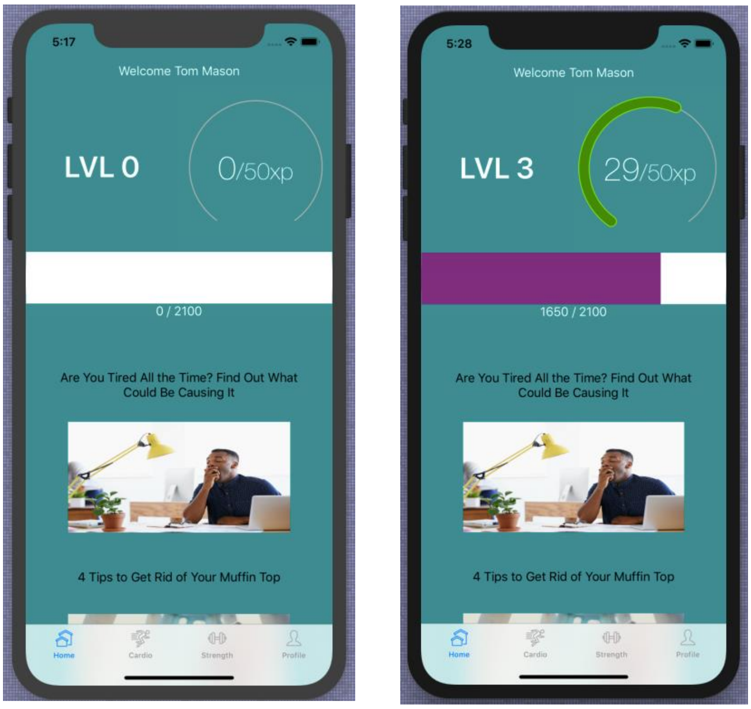
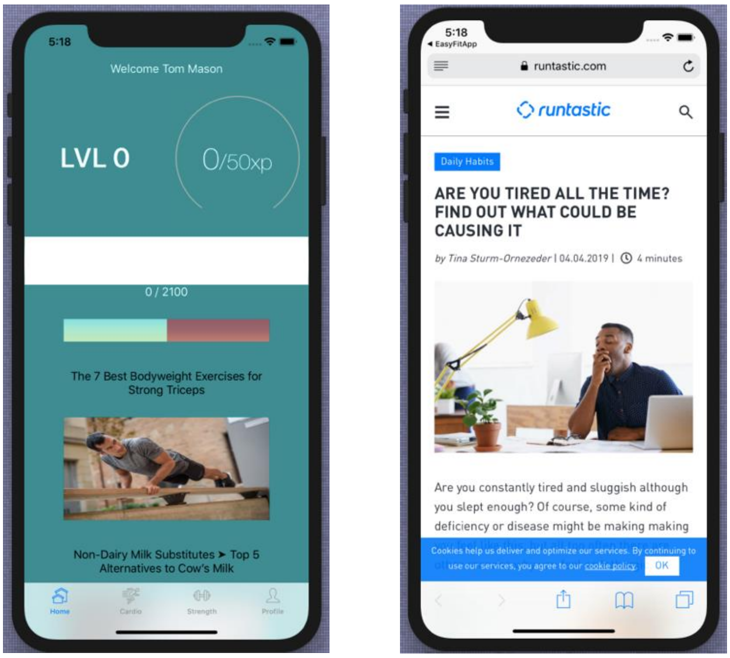

5:17
Welcome Tom Mason
LVL 0
0/50xp
0 / 2100
Are You Tired All the Time? Find Out What Could Be Causing It
4 Tips to Get Rid of Your Muffin Top
Home
Cardio
Strength
Profile
5:28
Welcome Tom Mason
LVL 3
29/50xp
1650 / 2100
Are You Tired All the Time? Find Out What Could Be Causing It
4 Tips to Get Rid of Your Muffin Top
Home
Cardio
Strength
Profile

Figure 14: Home page demonstrating the user's current level, 'EXP' and calories progress



ARE YOU TIRED ALL THE TIME? FIND OUT WHAT COULD BE CAUSING IT

by Tina Sturm-Ornezeder | 04.04.2019 | 4 minutes

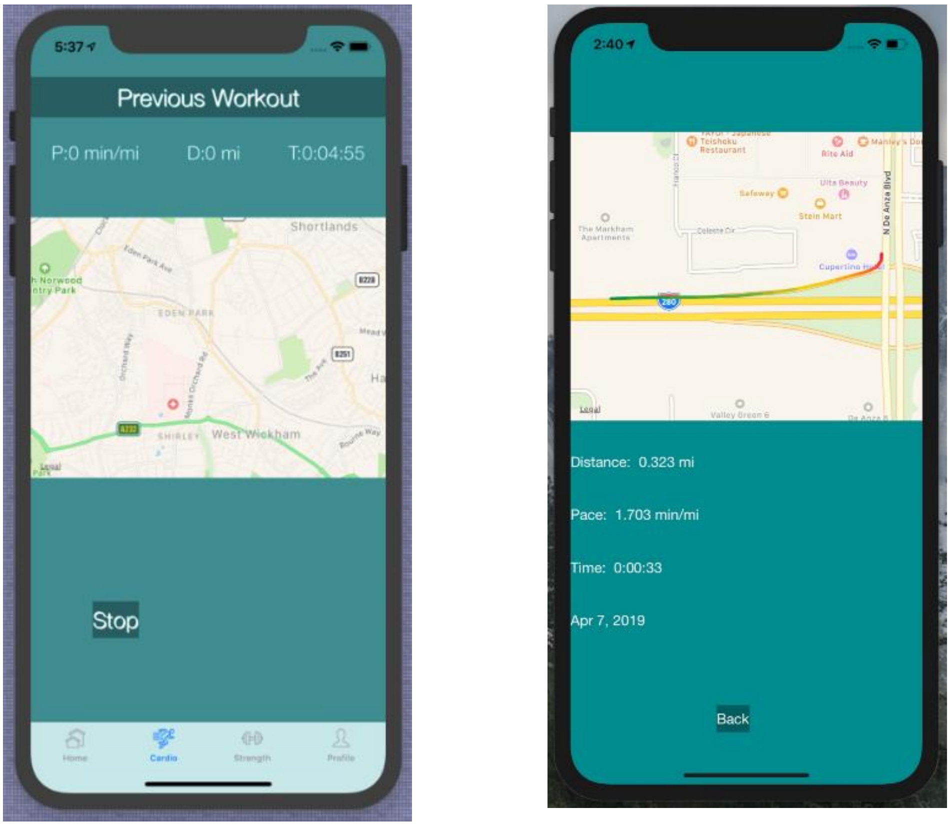
Are you constantly tired and sluggish although you slept enough? Of course, some kind of deficiency or disease might be making making

Cookies help us deliver and optimize our services. By continuing to use our services, you agree to our cookie policy. OK

Figure 15: The scrollable RSS feed and the second image shows the page redirection

4. *Cardio feature*

The cardio page lets the user record their run, by using their mobile phones GPS. When the user starts the run, the application records their whole journey. When the user decides to stop the run, the application displays the user's distance, pace, time and run date. In addition, it displays a map which shows the whole route. To make it a unique and fun experience for the user, we added a colour scheme where there is a colour code for each section of the run, depending on the pace of the user (see Figure 16).
(Colour code: **red** represents **slow pace**, **yellow** represents **average pace**, **green** represents **fast pace**).



Previous Workout

P:0 min/mi D:0 mi T:0:04:55

Stop

Distance: 0.323 mi

Pace: 1.703 min/mi

Time: 0:00:33

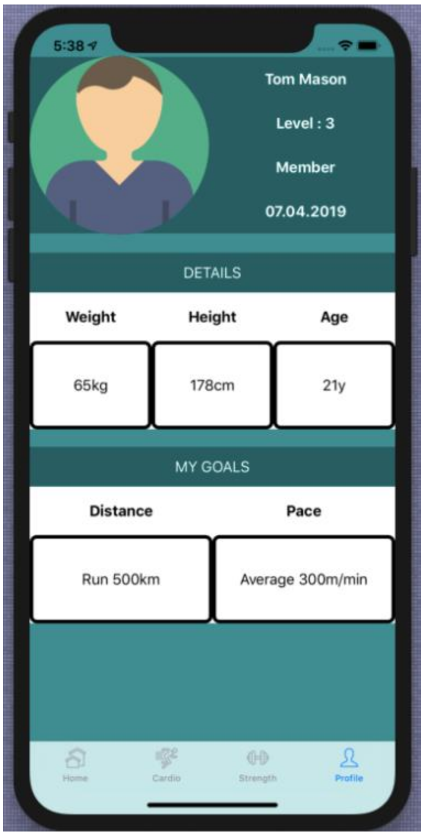
Apr 7, 2019

Back

Figure 16: The cardio page showing the user's route and the second image shows their final run stats with the colour coded route

5. *Profile page*

The profile page displays all the information related to the user. It includes their personal and statistics data retrieved from the database (see Figure 17).



Tom Mason

Level : 3

Member

07.04.2019

DETAILS

Weight	Height	Age
65kg	178cm	21y

MY GOALS

Distance	Pace
Run 500km	Average 300m/min

Figure 17: The user's profile page

6. Strength feature

Currently, the strength page is not fully functional. However, we designed the layout of the strength page and we intend to implement these features to work as intended in the future (see Figure 18).

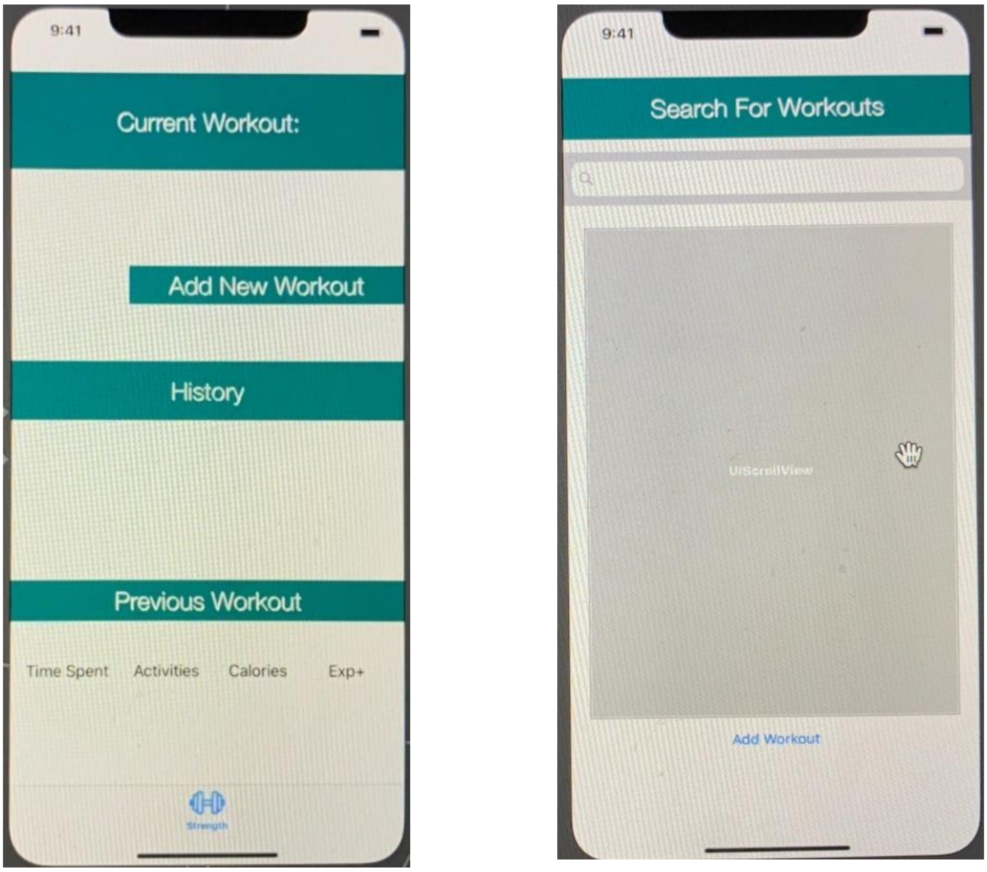


Figure 18: The first image shows the design of the strength page and the second image shows that the user can search for previous workouts when they click the 'Previous Workout' button

4.3.2 Backend implementation

4.3.2.1 Changes from the initial choice during development

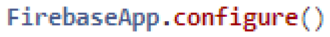
When we began implementation, we were still unsure about what to use for our database. Early in implementation, we were considering using MySQL as most of the group had general understanding of it, however, when we researched about Firebase, as a group we decided it would be a better alternative. This is because it was easy to connect the Firebase package to our project. The main benefits of using Firebase was, it allowed us to authenticate users locally (client side) and it handled the authentication, so we did not have to worry about privacy and security issues. Creating and reading information from the database was also incredibly simple with Firebase (see Chapter 2: section 2.2).

4.3.2.2 Firebase Realtime database

Setting up the firebase database with our application required us to have Cocoa pods installed as described in Chapter 2: section 2.1.1. Once the firebase pod was installed, we needed to configure the 'EasyFit' application to work with firebase by using '**FirebaseApp.configure()**'

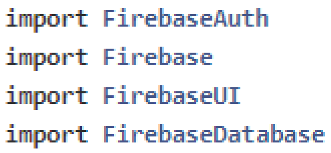
in the ‘**AppDelegate class**’. To be able to use the firebase class, the packages needed to be imported in each view controller which required data from the database or user authentication (see Figure 19 & 19.1). Once everything was setup correctly, the firebase authentication could be used to sign in a user. The sign in authentication takes two parameters, the email and the password which checks if the user details are valid. If they are, then the application takes the user to the home page. However, if the details are invalid then the user is prompted with an error message (see Figure 20). To retrieve data from the database, a database reference is required. We achieved this by using ‘**Database.database().reference()**’. We also required a unique ID (primary key) of the user currently logged in, which we could obtain by using ‘**Auth.auth().currentUser**’ (see Figure 21).

Now that we had both the database reference and the current users ID, we could retrieve data from the database using snapshots as seen in (see Figure 22, 22.1 & 22.2)



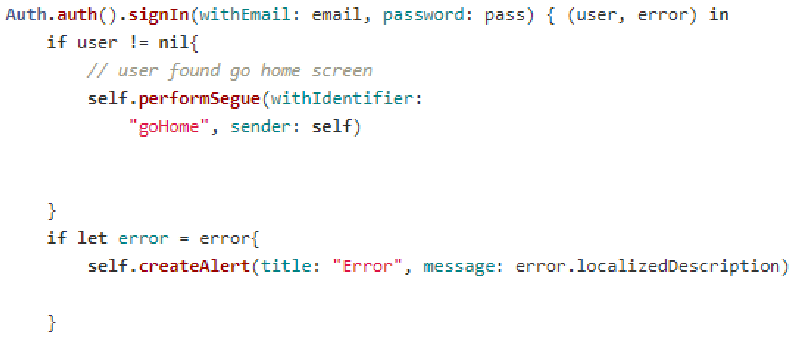
```
FirebaseApp.configure()
```

Figure 19: Configure firebase to work with the ‘EasyFit’ application



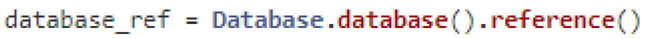
```
import FirebaseAuth
import Firebase
import FirebaseUI
import FirebaseDatabase
```

Figure 19.1: Firebase packages



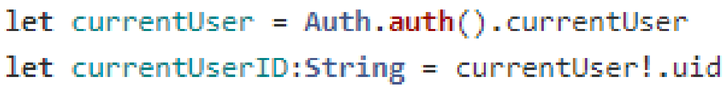
```
Auth.auth().signIn(withEmail: email, password: pass) { (user, error) in
    if user != nil{
        // user found go home screen
        self.performSegue(withIdentifier:
            "goHome", sender: self)


    }
    if let error = error{
        self.createAlert(title: "Error", message: error.localizedDescription)

    }
```

Figure 20: Firebase sign in authentication



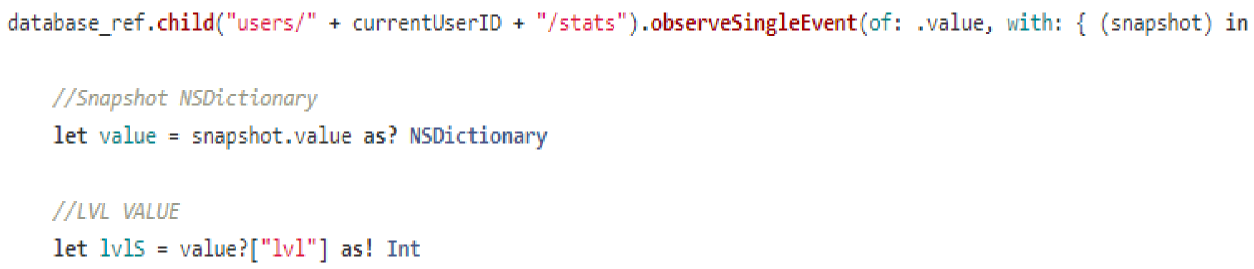
```
database_ref = Database.database().reference()
```

Figure 21: Firebase reference



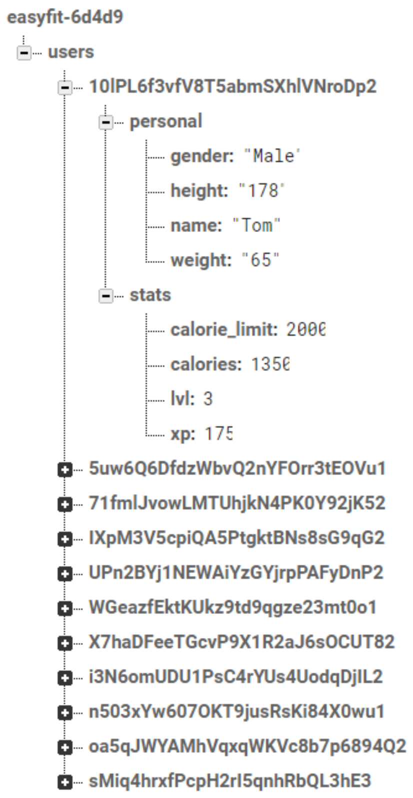
```
let currentUser = Auth.auth().currentUser
let currentUserID:String = currentUser!.uid
```

Figure 22: ‘currentUser’ stores the unique ID of the current user logged in



```
database_ref.child("users/" + currentUserID + "/stats").observeSingleEvent(of: .value, with: { (snapshot) in

    //Snapshot NSDictionary
    let value = snapshot.value as? NSDictionary

    //LVL VALUE
    let lvlS = value?["lvl"] as! Int
```

Figure 22.1: Retrieving data from firebase



```
easyfit-6d4d9
  users
    10lPL6f3vfV8T5abmSXhlVNroDp2
      personal
        gender: "Male'
        height: "178'
        name: "Tom"
        weight: "65"
      stats
        calorie_limit: 2000
        calories: 1350
        lvl: 3
        xp: 175
    5uw6Q6DfdzWbvQ2nYFOrr3tEOVu1
    71fmlJvowLMTUhjkN4PK0Y92jK52
    IXpM3V5cpiQA5PtgktBNs8sG9qG2
    UPn2BYj1NEWAiYzGYjrpPAFyDnP2
    WGeazfEktKUkz9td9qgze23mt0o1
    X7haDFeeTGcvP9X1R2aJ6sOCUT82
    i3N6omUDU1PsC4rYUs4UodqDjIL2
    n503xYw607OKT9jusRsKi84X0wu1
    oa5qJWYAMhVqxqWKVc8b7p6894Q2
    sMiq4hrxfPcpH2rI5qnhRbQL3hE3
```

Figure 22.2: Firebase database structure

5. Formative Evaluation & Quality Assurance

This chapter describes the evaluation with users that was done during the development of our application. It also provides details of our approach to quality assurance and the testing carried out, including the results during the development of our application.

5.1 Initial User Feedback

During development, we used various quality assurance methods to test our application to receive useful feedback. Initially, the non - functional testing we did was creating a questionnaire to help us gain information which ranged from understanding what the market needed, to receiving opinions on different functionalities which we were planning to implement and the general aesthetic of the application. We visited a local gym and we provided students at our University with the three different prototypes we had created, each featuring a different colour scheme and each page having a different layout, asking for their feedback (see Appendix B). The feedback for the colour schemes was quite inconclusive, as all the colours received very similar feedback, and so we decided to go with blue, but a more neutral colour as we felt this would provide the most professional look for our application. The results were from a staggering 83% of users who filled out the questionnaire, providing feedback that they liked the design of the application which were used in our prototyping when producing our final application, so we decided to use these initial designs. The users also gave feedback on the features and we learnt that some of them didn’t have enough functionality or sometimes, too much. For instance, the strength page only had the current workout information and some users did not find that enough and useful. We then decided to add an option for searching for a certain workout and the ability to view previous workouts. This also, increased the chances of users to use this application as other applications such as MyFitnessPal, does not offer such features. In addition, after receiving feedback about the profile page, we decided to further implement a points system, as this would not only be an aesthetic feature, but also motivate the user to come back and use the application more often. The feedback regarding the cardio page said that having a feature that records and shows your journey when walking or running was unique and this encouraged us to implement it in our application.

5.2 In - Person Interviews

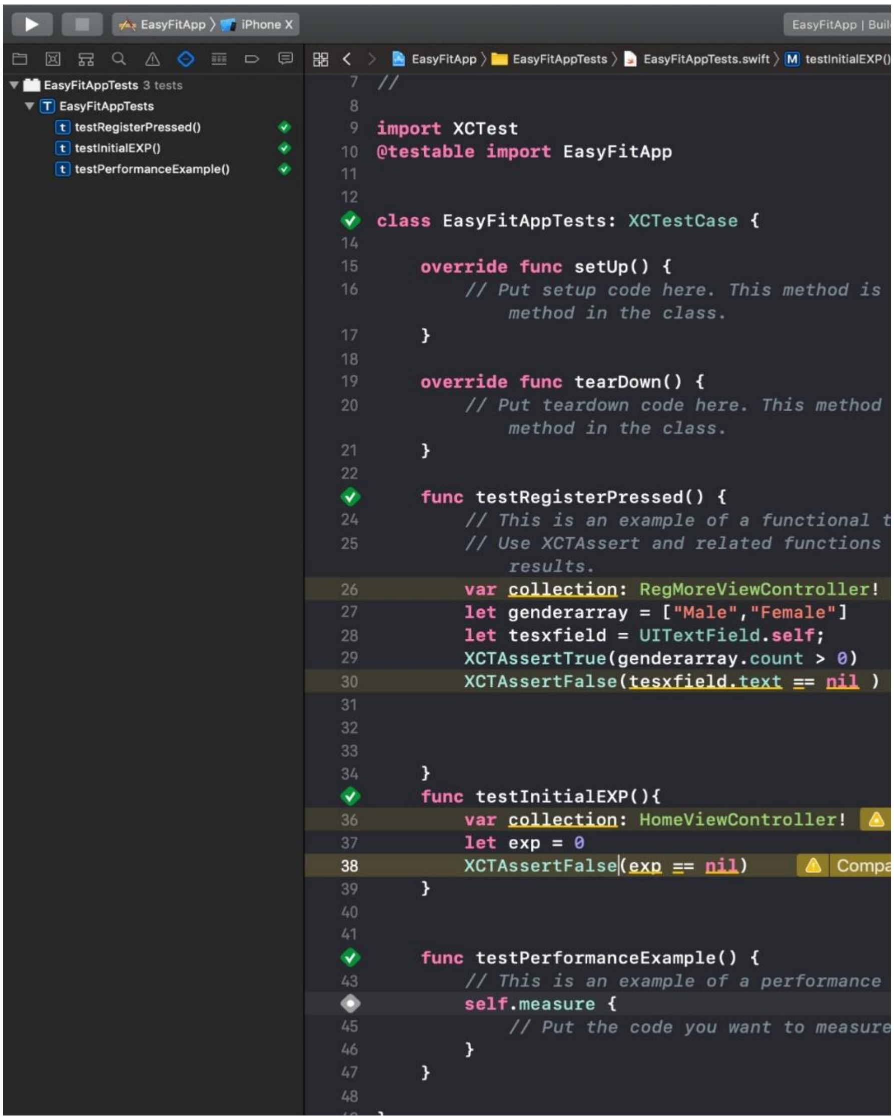
User testing was another way of gaining useful feedback. To begin with we created a testing plan (see Appendix C) to help us in the procedure of user testing. We visited a local gym to gain feedback for our application to make sure that our system met the initial objectives and that it would be the product that our users expected but also to help us introduce new and innovative features to the market. We managed to have 5 members of the gym agree to take part in our user testing. We provided our stakeholders with two documents. One contained a detailed description of the features in our application and the other contained a scoring table that the user had to fill out where each grading criteria was marked out of ten for the first two columns and the third column required them to write the most useful feature(s) they found in the application (see Appendix D: Scoring table using grading criteria). In addition, we observed the user's behaviour and noted it down while they used the application. While testing we had to consider confidentiality and data protection issues, so we ensured that was we did not ask the participants too many questions, this is because we did not want to pressurise them into answering something which they did not truly believe or give false feedback. Therefore, we designed the scoring table (Appendix D: Scoring table using grading criteria) related to the following:

- Overall design - was graded on how the application looked. The questions we asked were "Does the application look interesting?", "Are the colours appealing?" and "Any feedback regarding the design?".
- Ease of use - was graded on how easy the user found to navigate through the different features and use them.
- Useful feature - this was graded by testing all the features in the application and choosing the most useful feature(s). If the users can see themselves using all the features they can mention all of them and on the other hand, if the users found that there are unnecessary features they can mention them as well.

The feedback which we obtained was from a wide range of people, both male and female aged between 20 - 53 who use the gym regularly (see Appendix D: Observation & user feedback). When observing the users using the application we saw that most of the participants were able to use the application with ease. Two users found the RSS feed on the home page confusing, so we explained them that it meant the recent fitness news and to prevent future confusions we have added information about each link that redirects to a website. The overall feedback that we received from the users was positive which convinced us that the application is aesthetically pleasing.

5.3 Unit Testing

The final part of the testing phase was unit testing. This involved using Xcode for writing automated programs which checked the inputs and outputs of our application to make sure it was running correctly. This helped us identify flaws with the link between our application and the database. We wrote a program in Swift which systematically checked features of our application and made sure they are returning the right values. Unit testing was a vital part of the testing process, as it drastically minimised the chances of missing any bugs within the application. For this, we used tutorials from *RayWenderlich.com* [16] as this provided step-by-step instructions to check for any flaws in our application. Xcode's in-built testing within the Xcode test navigator seemed to be the easiest platform to complete our unit testing. While testing, the application worked as expected and this was an effective way fixing errors and bugs in our application. Figure 23 depicts the unit tests carried out on the registration (signup page) and the ‘EXP’ levelling system feature. On the left-hand side of the image under the ‘EasyFitAppTests’, the green ticks show that the tests have been passed.



EasyFitApp 〉 iPhone X
EasyFitApp | Buil
EasyFitApp 〉 EasyFitAppTests 〉 EasyFitAppTests.swift 〉 M testInitialEXP()
EasyFitAppTests 3 tests
T EasyFitAppTests
t testRegisterPressed() ✓
t testInitialEXP() ✓
t testPerformanceExample() ✓
Compa

```
//

import XCTest
@testable import EasyFitApp


class EasyFitAppTests: XCTestCase {

    override func setUp() {
        // Put setup code here. This method is
            method in the class.
    }

    override func tearDown() {
        // Put teardown code here. This method
            method in the class.
    }

    func testRegisterPressed() {
        // This is an example of a functional t
        // Use XCTAssert and related functions
            results.
        var collection: RegMoreViewController!
        let genderarray = ["Male","Female"]
        let tesxfield = UITextField.self;
        XCTAssertTrue(genderarray.count > 0)
        XCTAssertFalse(tesxfield.text == nil )



    }
    func testInitialEXP(){
        var collection: HomeViewController!
        let exp = 0
        XCTAssertFalse(exp == nil)
    }


    func testPerformanceExample() {
        // This is an example of a performance
        self.measure {
            // Put the code you want to measure
        }
    }

```

Figure 23: Unit Testing

5.4 Functional Testing

We tested all the features during and after the implementation. This was crucial because it helped us define the functionality that our application must have to perform successfully as we had to deliver an even User Experience (UX) across all devices and platforms. The following table shows the test cases to evaluate the correct functionality of the application.

Test	Expected Requirement	Core / Optional requirement	Test Pass Criteria
Login screen	Having a page which gives the user the ability to login with either their email address or social media account. Also, the ability to log out.	Core requirement	PASS: This has been achieved because the user is able to log into the application with their email or Facebook account. The database which was used for storing the information is Google Firebase.
EXP Function	Allowing the user to gain EXP (experience points) to level up and compete with their friends.	Core requirement	PASS: The user gains EXP when they have completed a run or workout (achieved a goal).
Cardio tracking	To give the user the ability to record their own routes when running and plan their workout routes.	Core requirement	PARTIAL PASS: The user's workout route and their final run stats shows on the screen. But, it does not let the user plan their routes.
Profile Page	A page which displays the user's profile and statistics of the user. The user should be able to edit their profile.	Core requirement	PASS: The page displays the profile information and statistics. The user can edit their personal information.
Database space	Every user's data should have its own space in the database to keep it secure so that it is not altered by another user's action.	Core requirement	PASS: The database stores every user's data separately which means the user has their own space in the database.
Exercise history	The user must be able to delete their previous workouts.	Optional requirement	FAIL: We were not able to have an option of deleting the user's previous history.
Route creation feature	The user should have access to the internet to allow the route creation feature to work as it requires GPS. This will only apply for the user's that will use this feature.	Optional requirement	PASS: Connecting to the internet enables the GPS and therefore the user can use this feature.

Table 1: Functional Requirements and Tests

5.5 Non - Functional Testing

Non - functional testing was crucial as it helped us to define the criteria that we can use to analyse the operation of our application. This is based on the questionnaire results (see Appendix B). The table below shows the non - functional requirements and the importance of each one.

Description	**Crucial / Non - crucial requirement**	**Checked – meet the requirement**
The user should have a good User Experience when interacting with the application on a mobile device such as navigating easily through the features.	Crucial	Including the icons and clear information on each page allowed the user to have a fun experience when using the application.
The user should find the application visually appealing.	Crucial	The interface of our application was modified after we received feedback and we made the application visually appealing.
The application is secure to use.	Crucial	Using security features such as user authentication, we enabled a solution to hide and secure the users personal and sensitive data.
Feedback from users showed that the external links such as the RSS feeds was a confusing concept. We were determined to make sure that the user understands that the home page allows access to other websites as this was a beneficial way for the user to gain extra information on health and fitness.	Non - crucial	We added a description before every link to show the purpose of the website. This helped the user decide whether they would like to visit the website. For example, we have a link to a website that gives tips on bodyweight exercises and this is shown by an image of a man exercising. To make this clear to the user we added a description saying, “The 7 best bodyweight exercises for strong triceps”. This was an effective way of helping the user decide whether they were interested in visiting the website.

The application must not have bugs but in case there is any, it should inform the user. It should inform the user if they come across any wrong operations.	Crucial	The application is thoroughly tested when a new feature is added, and error messages are implemented to inform the user of any wrong operations.
The application must have the ability to develop new functionalities.	Crucial	The system must be designed in a way that as a group we can add any new functionalities and features to the application in future.

Table 2: Non - functional Requirements

6. Summative Evaluation

This chapter describes the evaluation methodology used which includes the methods, results and conclusions for our application.

6.1 Evaluation of our work

The main aim of our project was to encourage and motivate users to keep a balance of both, fitness and a healthy lifestyle by creating an iOS mobile phone application that is free and intuitive. The current available and work features of our application are; the user is able to register and signup or if they prefer, they can register/login using their Facebook account, the ability to start recording a cardio workout which uses the user's GPS location and pedometer to track the distance and speed of their run, a levelling system which rewards users with experience points ('EXP') to progress their level for completing tasks and using our application, a home page with a Rich Site Summary (RSS) feed with useful fitness and healthy living articles, a profile section where the user can view their current progress as well as their personal information and an intuitive and easy to user overall experience.

We strongly believe first impressions make or break a service/product thus we concluded to provide all the users the most intuitive and best overall experience, we would have to develop the application exclusively for iOS devices. Due to the limited number of iOS devices we could test and fine tune the application to each one to ensure it meets our high standards. Even though our team had no prior experience with Swift we were very ambitious and determined to create an iOS application. Thus, during development, we had to learn and develop at the same time which impeded our development progress, but we managed to meet all our milestones and even due to the limited time given for development, we made prominent progress and with more time we know we could create the experience we believe our target audience deserves. Furthermore, we wanted to focus on the frontend, so we could provide an intuitive user interface therefore, we decided to user Firebase as our backend and this relieved the stress and time it takes to create a whole backend system.

Throughout the project we learned many invaluable lessons and developed as individuals, learning to work as a team and to project manage efficiently. Due to many successful techniques and methods, we were able to end the project with a sense of pride and accomplishment. The methods and techniques which had a crucial impact on our project were; assigning each group member a task to start developing the application, dividing the tasks amongst every member ensured that everyone worked in a different area throughout the development process, reducing chances of collisions and allowing everyone to experience working on different parts of the development phase.

During the development process, we experienced multiple front and backend collisions, these were solved through constant communication and updates between all members. In addition, during the development we followed the software evolution cycle to tackle any issues that occurred. To manage the development, we used the agile, Scrum technique. Using a Project Management System called ‘Trello’ we created a work area where a Kanban board was used to keep track of the tasks that needed to be completed, with the appropriate priorities, as well as the date that it had to be achieved. Our board was divided in 4 development sections – Backlog, Development. Testing and Complete. Each section had ‘Sprints’ that each member was assigned, and each one had an estimated time of completion so that it was completed before the next one was added. After completing the sprints, as a group, we reviewed the project’s progress and assessed the outcome every week.

6.2 Future Development

At the beginning of the whole project our goal was to go ahead with an ambitious and complex idea which challenged us in a way we have never been challenged before, testing and enhancing a wide range of our skills while also delivering a product which we believe has huge potential. The overall project idea was difficult and complex alongside having many different assignments from other modules certain functionalities of our application were delayed or not achievable. Justifying that we needed more time to complete such a vast number of features.

The main functions of the project were:

- To let the users, create an account and login to the application, using their registered email and password or using their Facebook account. This feature is successfully completed, and it is working.
- To have a levelling system, which gives ‘EXP’ points to every user, when they complete a workout, workout regularly (few days in a row) and meet their calories goal. This feature is successfully completed, and it is working as it is showing the experience and level for every user and storing it to the database for every account.
- To have a profile page, that shows all the details about the user. This feature is successfully completed, and it is working.
- To let the user, track a basic cardio workout. This feature is successfully completed, and it is working as the user can start recording a cardio workout which uses their GPS location and pedometer to track the distance and speed of their run and at the end, it displays a map of the route they ran with a colour code indicating their pace - red for slowest pace, yellow for the average pace and green for the fastest pace.
- To add a strength workout feature where the user can add a workout and it will be saved to the database and the user can see their weight and progress information.

We were not able to complete this feature, so it is not fully functional. However, we designed the layout of the strength page and we intend to implement these features to work as intended in the future (see Figure 18).

- To allow the user to add food into their diary either manually, searching through the database or through scanning a barcode. We were not able to complete this feature because it was the most challenging and due to its complexity and shortage of time, we intend to implement it in the future.

In future given more time, we would potentially complete the following and enhance the features in our application:

- Improve the cardio feature by allowing the user to view previous workouts with all the details.
- Improve and fine tune the whole user interface to make it look more pleasing to the eye.
- Allows users to connect and use our application with other social media platforms such as Instagram.
- Allows users to connect with and add their friends with the ability to see their current level and progress.
- Create a leader board and monthly competitions to encourage users to compete, and potentially, if we could find the right sponsor, we would like to introduce a reward for constant progress. For example, if a user has been completing many exercises, filling in their food diary and gaining experience points which leads them to reach a certain level or amount of experience points they would be rewarded with a 10% discount code to use on that sponsors website.

6.3 Conclusion

To conclude, the group was satisfied with the progress made and strongly believe our idea has great potential, with more time and better organisation, we would be able to finish the product completely. We have all learnt invaluable lessons from this project and even though it has been challenging, at times, the experience has been positive.

7. Bibliography

[1] "Health matters: getting every adult active every day" [WWW Document], GOV.UK. URL: https://www.gov.uk/government/publications/health-matters-getting-every-adult-active-every-day/health-matters-getting-every-adult-active-every-day
(Accessed on: 3.03.2019)

[2]Benefits of exercise [WWW Document], 2018 nhs.uk. URL https://www.nhs.uk/live-well/exercise/exercise-health-benefits/ (Accessed on: 3.03.2019)

[3] En.wikipedia.org. (2019). *Swift (programming language)*. [online] Available at: https://en.wikipedia.org/wiki/Swift_(programming_language)
(Accessed on: 9.04.2019)

[4] En.wikipedia.org. (2019). *Xcode*. [online] Available at: https://en.wikipedia.org/wiki/Xcode (Accessed on: 9.04.2019)

[5] En.wikipedia.org. (2019). *CocoaPods*. [online] Available at: https://en.wikipedia.org/wiki/CocoaPods (Accessed on: 9.04.2019)

[6] En.wikipedia.org. (2019). *Firebase*. [online] Available at: https://en.wikipedia.org/wiki/Firebase (Accessed on: 9.04.2019).

[7] En.wikipedia.org. (2019). *Crashlytics*. [online] Available at: https://en.wikipedia.org/wiki/Crashlytics (Accessed 9 Apr. 2019).

[8] "Gamification" (24.04.19). Available at: https://en.wikipedia.org/wiki/Gamification
(Accessed on: 25.04.2019)

[9] Johnson.D, Deterding.S, Kuhn.K.A, Staneva.A, Stoyanov.S, Hides.L. Volume 6, Pages 89 - 106 "Gamification for health and wellbeing: A systematic review of the literature" (02.11.2016).
Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6096297/
(Accessed on: 25.04.2019)

[10] “Swift.org” [WWW Document]. Swift.org. URL https://swift.org
(Accessed on: 3.03.2019).

[11] Chapter 9 – Software Evolution, n.d. . Software evolution 56. URL: http://csis.pace.edu/~marchese/CS389/L9/Ch9_summary.pdf (Accessed on: 3.03.2019)

[12] Thakur D, “Software Maintenance Life Cycle” URL: http://ecomputernotes.com/software-engineering/software-maintenance-life-cycle
(Accessed on: 3.03.2019)

[13] Git - Recording Changes to the Repository [WWW Document], n.d. URL: https://git-scm.com/book/en/v2/Git-Basics-Recording-Changes-to-the-Repository (Accessed on: 3.10.2019)

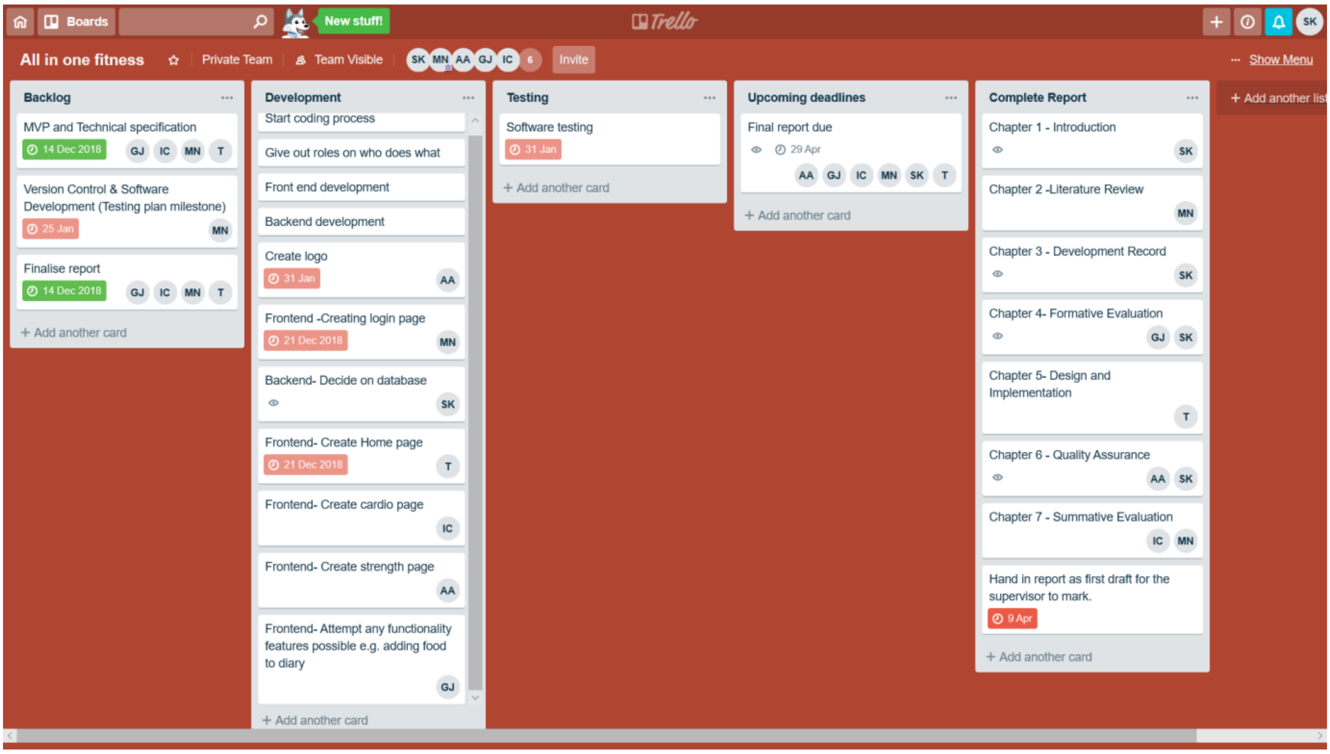
[14] Runtastic articles Available at: https://www.runtastic.com/blog/en/feed/ (Accessed on: 25.04.2019)

[15] "Your XCode Project Files Explained (and object oriented programming concepts)". 03/10/2013. URL: https://codewithchris.com/xcode-project-files-explained/ (Accessed on: 7.04.2019)

[16] "iOS Unit Testing and UI Testing Tutorial". (13.03.2017). Available at: https://www.raywenderlich.com/709-ios-unit-testing-and-ui-testing-tutorial (Accessed on: 23.04.2019)

8. Appendices

Appendix A: Trello Board



Backlog

- MVP and Technical specification — 14 Dec 2018 — GJ IC MN T
- Version Control & Software Development (Testing plan milestone) — 25 Jan — MN
- Finalise report — 14 Dec 2018 — GJ IC MN T
- + Add another card

Development

- Start coding process
- Give out roles on who does what
- Front end development
- Backend development
- Create logo — 31 Jan — AA
- Frontend -Creating login page — 21 Dec 2018 — MN
- Backend- Decide on database — SK
- Frontend- Create Home page — 21 Dec 2018 — T
- Frontend- Create cardio page — IC
- Frontend- Create strength page — AA
- Frontend- Attempt any functionality features possible e.g. adding food to diary — GJ
- + Add another card

Testing

- Software testing — 31 Jan
- + Add another card

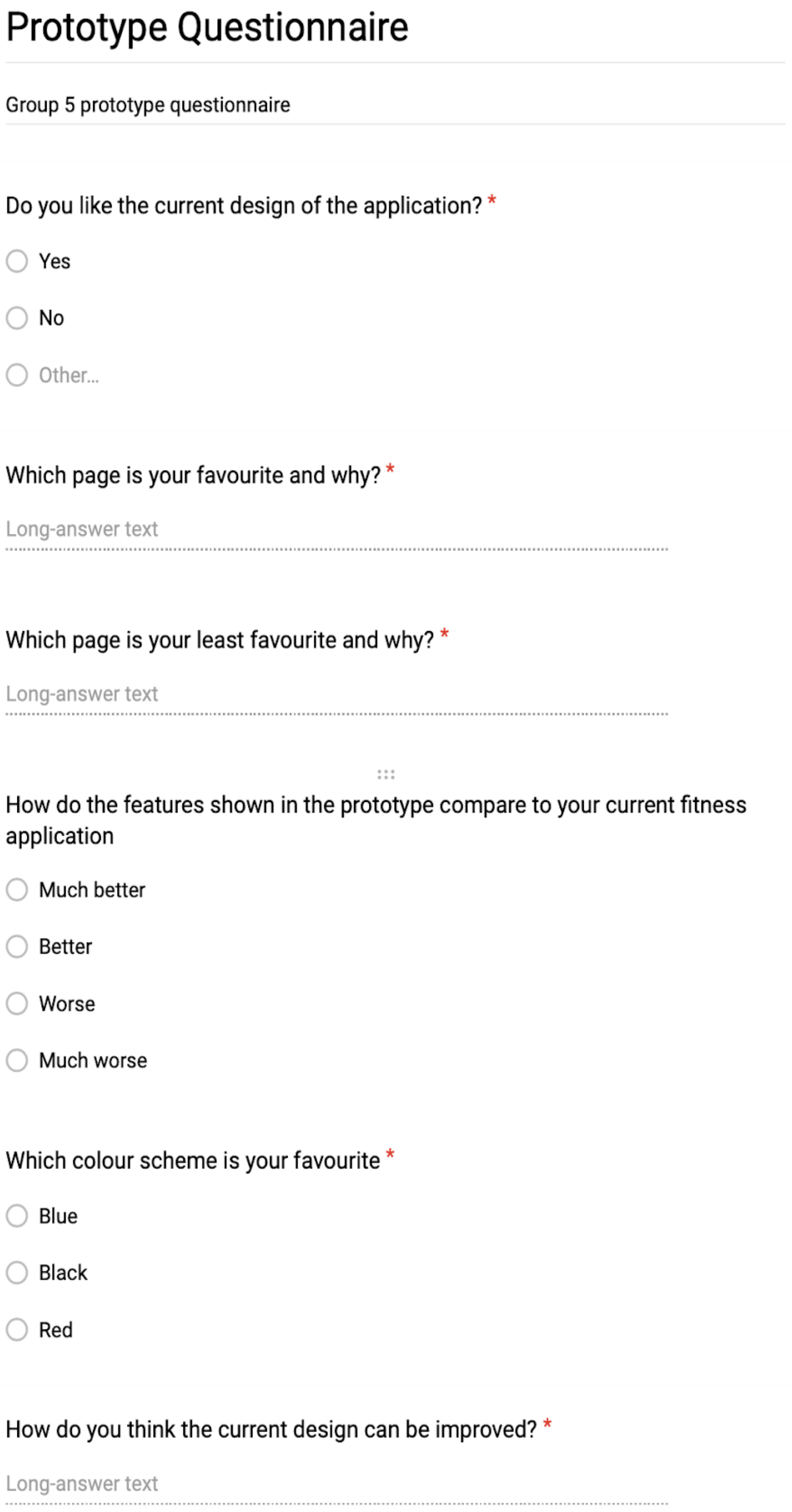
Upcoming deadlines

- Final report due — 29 Apr — AA GJ IC MN SK T
- + Add another card

Complete Report

- Chapter 1 - Introduction — SK
- Chapter 2 -Literature Review — MN
- Chapter 3 - Development Record — SK
- Chapter 4- Formative Evaluation — GJ SK
- Chapter 5- Design and Implementation — T
- Chapter 6 - Quality Assurance — AA SK
- Chapter 7 - Summative Evaluation — IC MN
- Hand in report as first draft for the supervisor to mark. — 9 Apr
- + Add another card

APPENDIX B: Prototype questionnaire and feedback from stakeholders



Prototype Questionnaire

Group 5 prototype questionnaire

Do you like the current design of the application? *

○ Yes

○ No

○ Other...

Which page is your favourite and why? *

Long-answer text

Which page is your least favourite and why? *

Long-answer text

How do the features shown in the prototype compare to your current fitness application

○ Much better

○ Better

○ Worse

○ Much worse

Which colour scheme is your favourite *

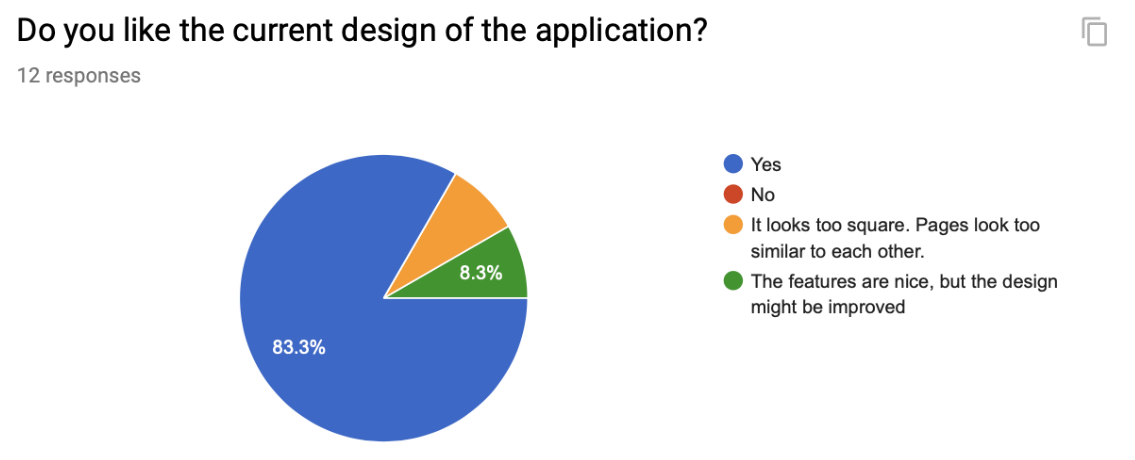
○ Blue

○ Black

○ Red

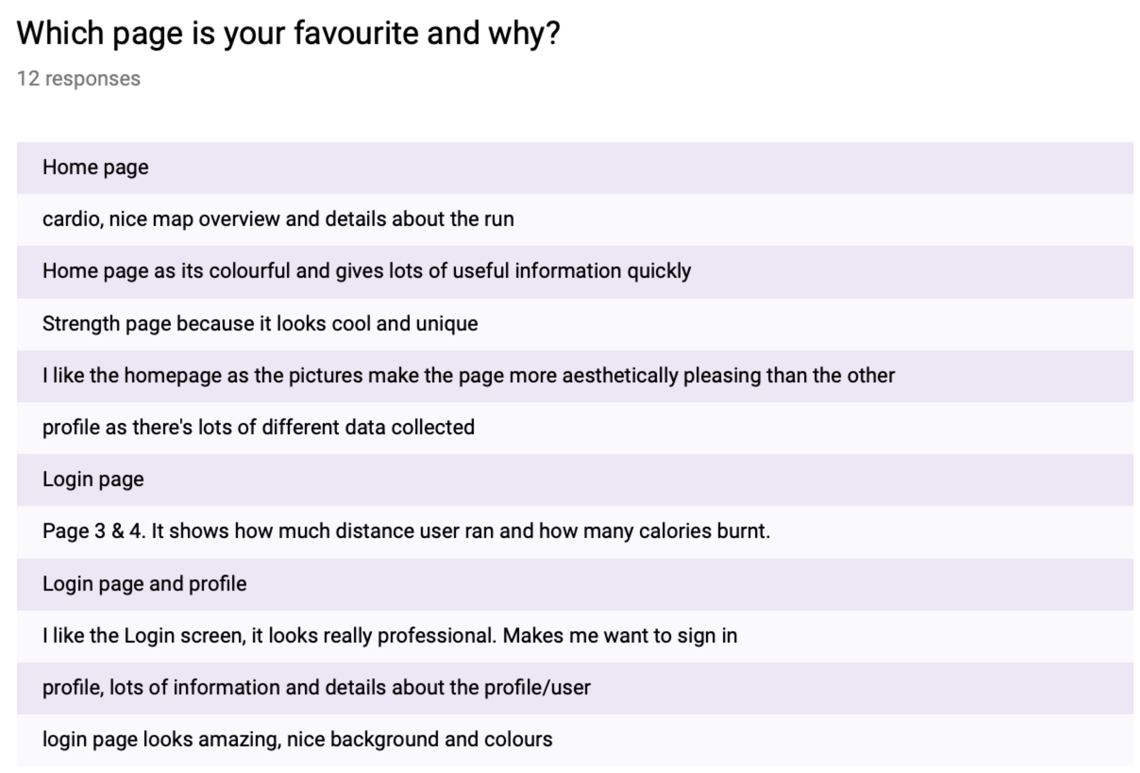
How do you think the current design can be improved? *

Long-answer text



Do you like the current design of the application?

12 responses



Which page is your favourite and why?

12 responses

Home page

cardio, nice map overview and details about the run

Home page as its colourful and gives lots of useful information quickly

Strength page because it looks cool and unique

I like the homepage as the pictures make the page more aesthetically pleasing than the other

profile as there's lots of different data collected

Login page

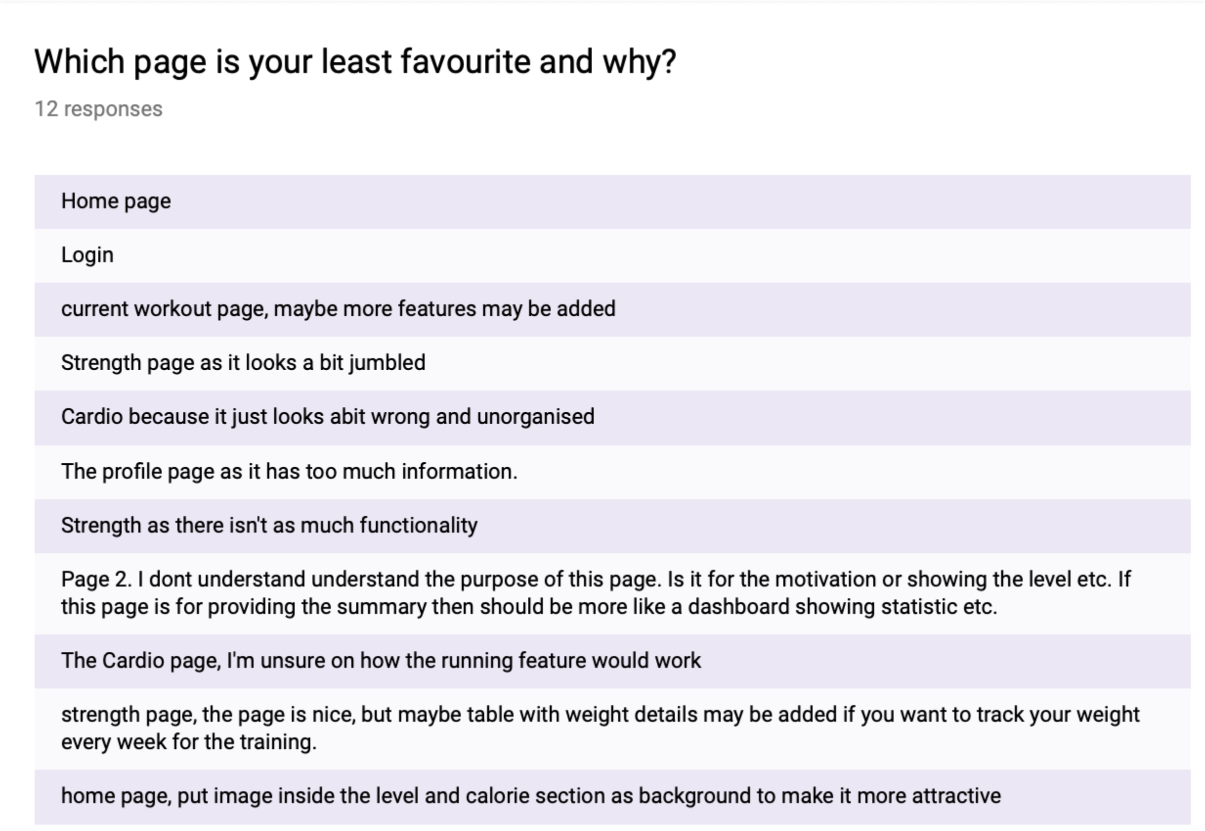
Page 3 & 4. It shows how much distance user ran and how many calories burnt.

Login page and profile

I like the Login screen, it looks really professional. Makes me want to sign in

profile, lots of information and details about the profile/user

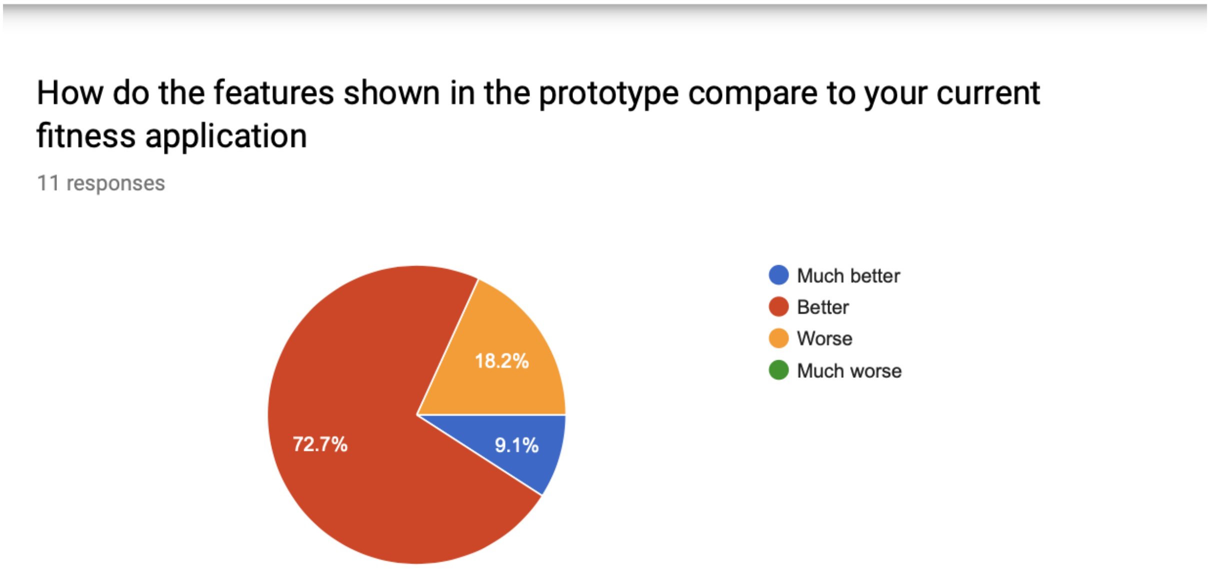
login page looks amazing, nice background and colours



Which page is your least favourite and why?

12 responses

Home page
Login
current workout page, maybe more features may be added
Strength page as it looks a bit jumbled
Cardio because it just looks abit wrong and unorganised
The profile page as it has too much information.
Strength as there isn't as much functionality
Page 2. I dont understand understand the purpose of this page. Is it for the motivation or showing the level etc. If this page is for providing the summary then should be more like a dashboard showing statistic etc.
The Cardio page, I'm unsure on how the running feature would work
strength page, the page is nice, but maybe table with weight details may be added if you want to track your weight every week for the training.
home page, put image inside the level and calorie section as background to make it more attractive

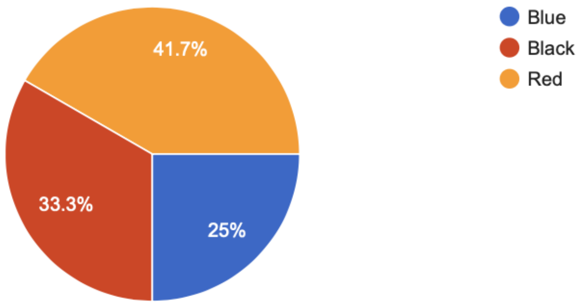


How do the features shown in the prototype compare to your current fitness application

11 responses

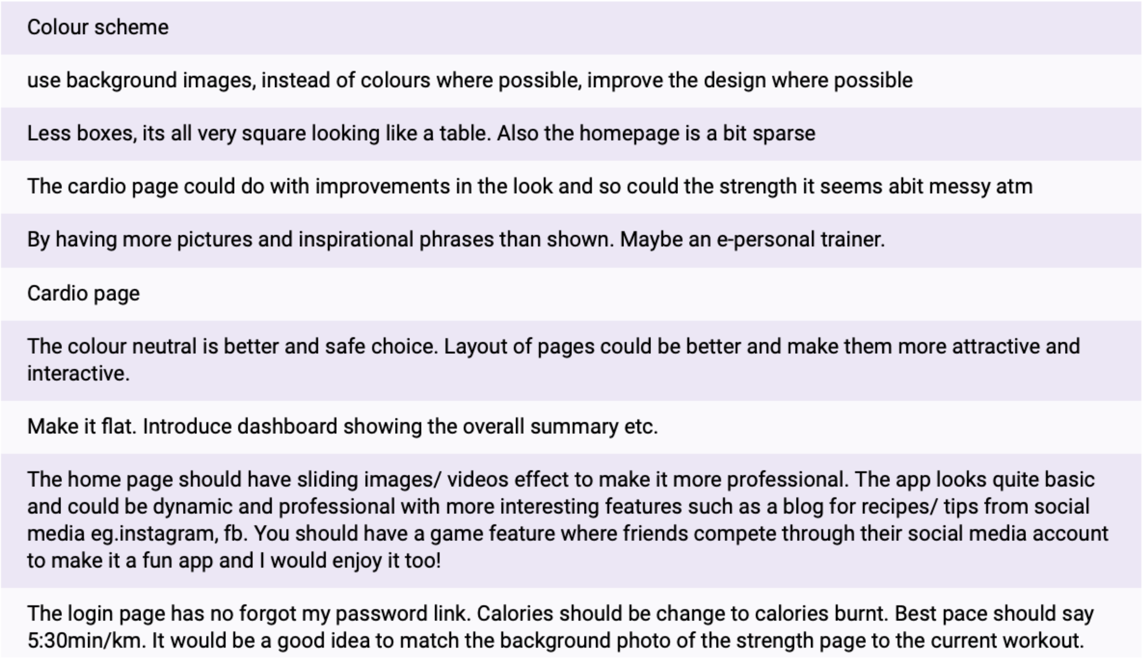
Which colour scheme is your favourite

12 responses



How do you think the current design can be improved?

12 responses



Colour scheme

use background images, instead of colours where possible, improve the design where possible

Less boxes, its all very square looking like a table. Also the homepage is a bit sparse

The cardio page could do with improvements in the look and so could the strength it seems abit messy atm

By having more pictures and inspirational phrases than shown. Maybe an e-personal trainer.

Cardio page

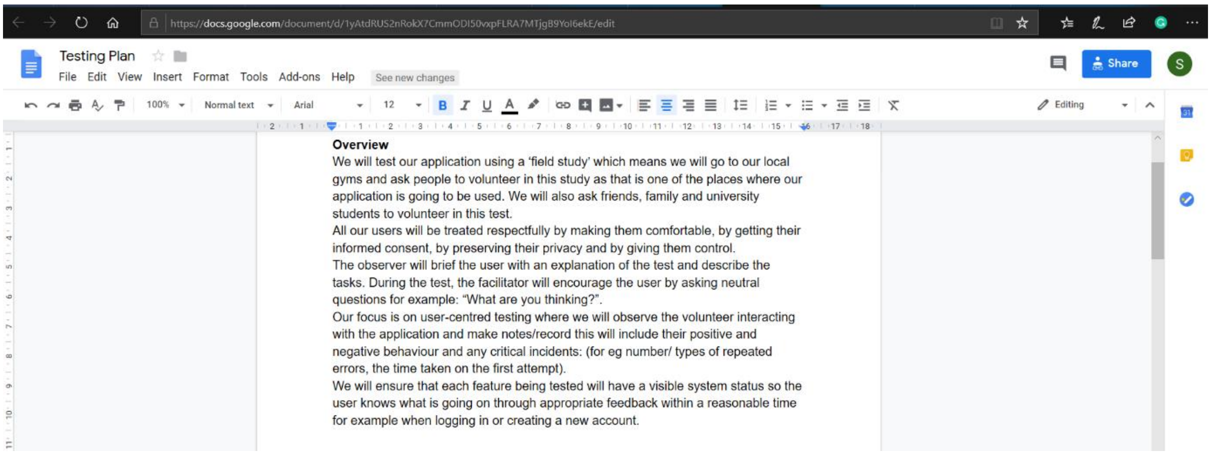
The colour neutral is better and safe choice. Layout of pages could be better and make them more attractive and interactive.

Make it flat. Introduce dashboard showing the overall summary etc.

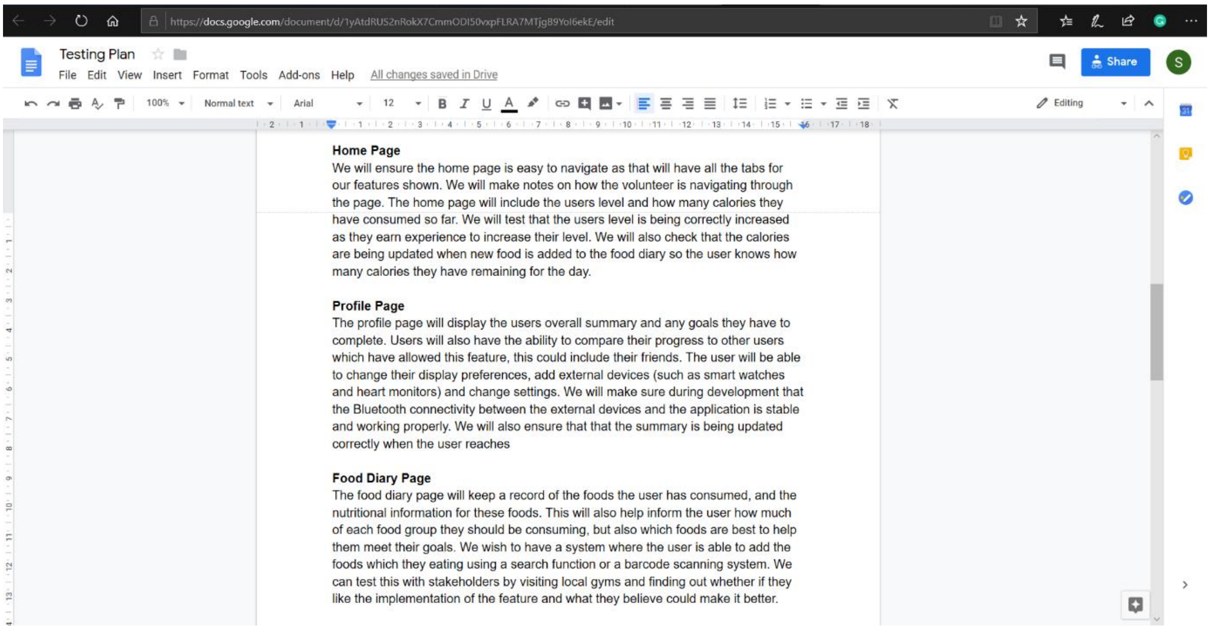
The home page should have sliding images/ videos effect to make it more professional. The app looks quite basic and could be dynamic and professional with more interesting features such as a blog for recipes/ tips from social media eg.instagram, fb. You should have a game feature where friends compete through their social media account to make it a fun app and I would enjoy it too!

The login page has no forgot my password link. Calories should be change to calories burnt. Best pace should say 5:30min/km. It would be a good idea to match the background photo of the strength page to the current workout.

Appendix C: Testing Plan


https://docs.google.com/document/d/1yAtdRUS2nRokX7CmmODI50vxpFLRA7MTjgB9YoI6ekE/edit
Testing Plan
File Edit View Insert Format Tools Add-ons Help See new changes
Share
Editing

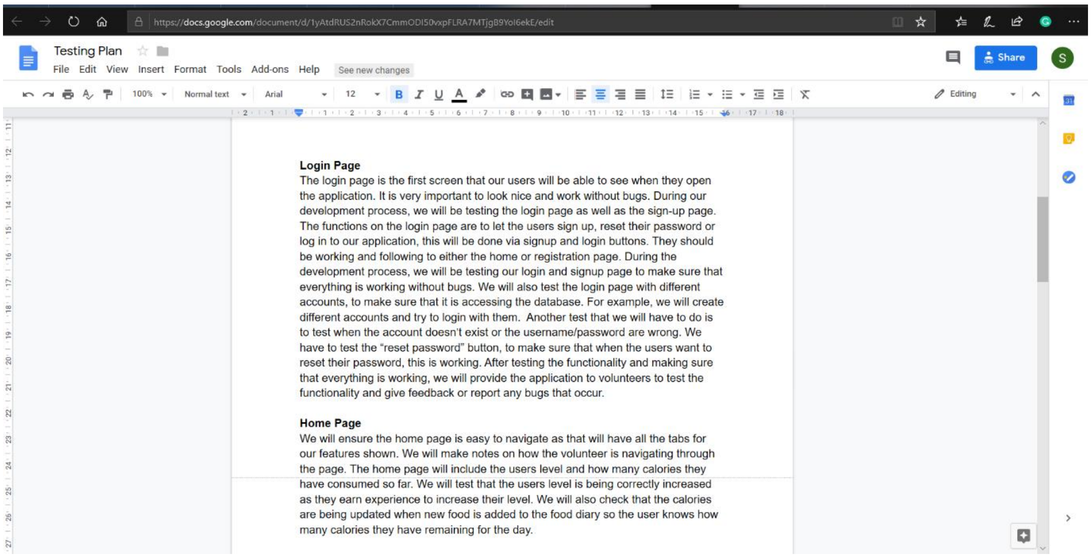
Overview
We will test our application using a 'field study' which means we will go to our local gyms and ask people to volunteer in this study as that is one of the places where our application is going to be used. We will also ask friends, family and university students to volunteer in this test.
All our users will be treated respectfully by making them comfortable, by getting their informed consent, by preserving their privacy and by giving them control.
The observer will brief the user with an explanation of the test and describe the tasks. During the test, the facilitator will encourage the user by asking neutral questions for example: "What are you thinking?".
Our focus is on user-centred testing where we will observe the volunteer interacting with the application and make notes/record this will include their positive and negative behaviour and any critical incidents: (for eg number/ types of repeated errors, the time taken on the first attempt).
We will ensure that each feature being tested will have a visible system status so the user knows what is going on through appropriate feedback within a reasonable time for example when logging in or creating a new account.


https://docs.google.com/document/d/1yAtdRUS2nRokX7CmmODI50vxpFLRA7MTjgB9YoI6ekE/edit
Testing Plan
File Edit View Insert Format Tools Add-ons Help All changes saved in Drive
Share
Editing

Home Page
We will ensure the home page is easy to navigate as that will have all the tabs for our features shown. We will make notes on how the volunteer is navigating through the page. The home page will include the users level and how many calories they have consumed so far. We will test that the users level is being correctly increased as they earn experience to increase their level. We will also check that the calories are being updated when new food is added to the food diary so the user knows how many calories they have remaining for the day.

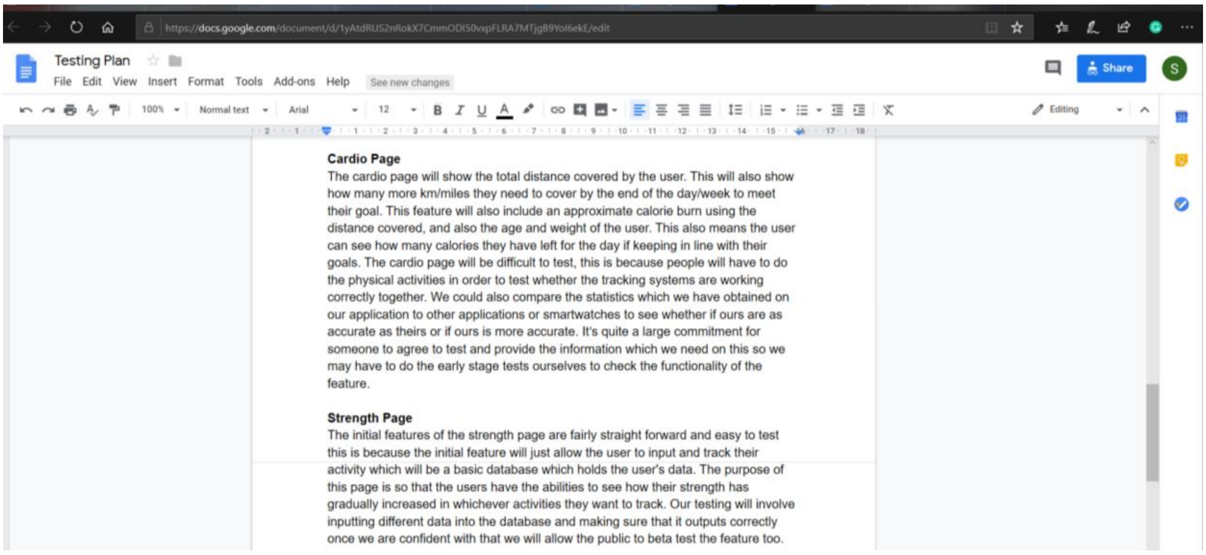
Profile Page
The profile page will display the users overall summary and any goals they have to complete. Users will also have the ability to compare their progress to other users which have allowed this feature, this could include their friends. The user will be able to change their display preferences, add external devices (such as smart watches and heart monitors) and change settings. We will make sure during development that the Bluetooth connectivity between the external devices and the application is stable and working properly. We will also ensure that that the summary is being updated correctly when the user reaches

Food Diary Page
The food diary page will keep a record of the foods the user has consumed, and the nutritional information for these foods. This will also help inform the user how much of each food group they should be consuming, but also which foods are best to help them meet their goals. We wish to have a system where the user is able to add the foods which they eating using a search function or a barcode scanning system. We can test this with stakeholders by visiting local gyms and finding out whether if they like the implementation of the feature and what they believe could make it better.


https://docs.google.com/document/d/1yAtdRUS2nRokX7CmmODI50vxpFLRA7MTjgB9YoI6ekE/edit
Testing Plan
File Edit View Insert Format Tools Add-ons Help See new changes
Share
Editing

Login Page
The login page is the first screen that our users will be able to see when they open the application. It is very important to look nice and work without bugs. During our development process, we will be testing the login page as well as the sign-up page. The functions on the login page are to let the users sign up, reset their password or log in to our application, this will be done via signup and login buttons. They should be working and following to either the home or registration page. During the development process, we will be testing our login and signup page to make sure that everything is working without bugs. We will also test the login page with different accounts, to make sure that it is accessing the database. For example, we will create different accounts and try to login with them. Another test that we will have to do is to test when the account doesn't exist or the username/password are wrong. We have to test the "reset password" button, to make sure that when the users want to reset their password, this is working. After testing the functionality and making sure that everything is working, we will provide the application to volunteers to test the functionality and give feedback or report any bugs that occur.

Home Page
We will ensure the home page is easy to navigate as that will have all the tabs for our features shown. We will make notes on how the volunteer is navigating through the page. The home page will include the users level and how many calories they have consumed so far. We will test that the users level is being correctly increased as they earn experience to increase their level. We will also check that the calories are being updated when new food is added to the food diary so the user knows how many calories they have remaining for the day.

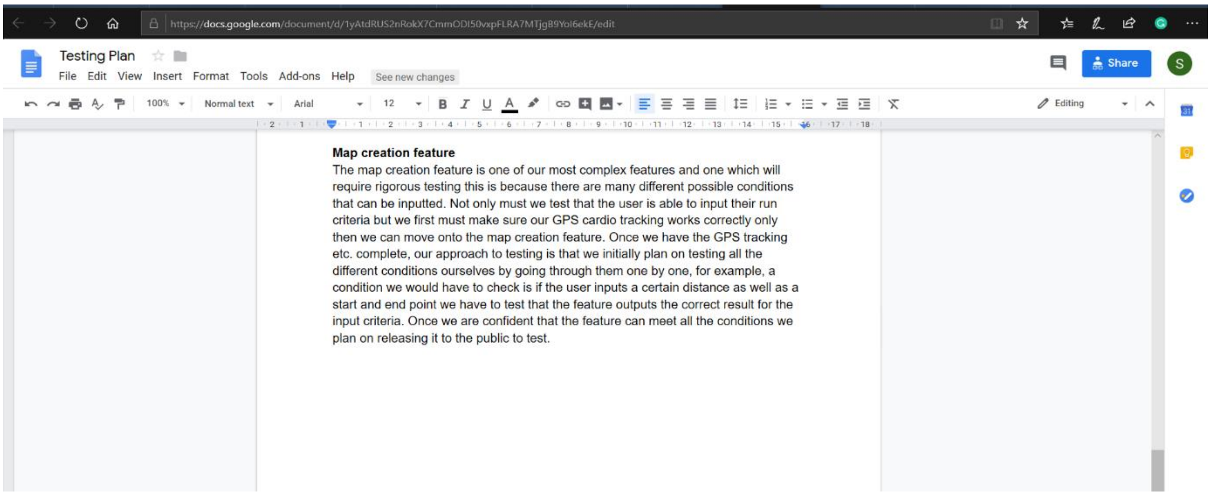

https://docs.google.com/document/d/1yAtdRUS2nRokX7CmmODI50vxpFLRA7MTjgB9Yol6ekE/edit
Testing Plan
File Edit View Insert Format Tools Add-ons Help
See new changes
Share
S
100%
Normal text
Arial
12
Editing

Cardio Page

The cardio page will show the total distance covered by the user. This will also show how many more km/miles they need to cover by the end of the day/week to meet their goal. This feature will also include an approximate calorie burn using the distance covered, and also the age and weight of the user. This also means the user can see how many calories they have left for the day if keeping in line with their goals. The cardio page will be difficult to test, this is because people will have to do the physical activities in order to test whether the tracking systems are working correctly together. We could also compare the statistics which we have obtained on our application to other applications or smartwatches to see whether if ours are as accurate as theirs or if ours is more accurate. It's quite a large commitment for someone to agree to test and provide the information which we need on this so we may have to do the early stage tests ourselves to check the functionality of the feature.

Strength Page

The initial features of the strength page are fairly straight forward and easy to test this is because the initial feature will just allow the user to input and track their activity which will be a basic database which holds the user's data. The purpose of this page is so that the users have the abilities to see how their strength has gradually increased in whichever activities they want to track. Our testing will involve inputting different data into the database and making sure that it outputs correctly once we are confident with that we will allow the public to beta test the feature too.


https://docs.google.com/document/d/1yAtdRUS2nRokX7CmmODI50vxpFLRA7MTjgB9Yol6ekE/edit
Testing Plan
File Edit View Insert Format Tools Add-ons Help
See new changes
Share
S
100%
Normal text
Arial
12
Editing

Map creation feature

The map creation feature is one of our most complex features and one which will require rigorous testing this is because there are many different possible conditions that can be inputted. Not only must we test that the user is able to input their run criteria but we first must make sure our GPS cardio tracking works correctly only then we can move onto the map creation feature. Once we have the GPS tracking etc. complete, our approach to testing is that we initially plan on testing all the different conditions ourselves by going through them one by one, for example, a condition we would have to check is if the user inputs a certain distance as well as a start and end point we have to test that the feature outputs the correct result for the input criteria. Once we are confident that the feature can meet all the conditions we plan on releasing it to the public to test.

Appendix D: Testing

1. Observation & User feedback:

User one:

Name:	Elise
Age:	23
Occupation:	Full-time postgraduate student
What training they do	They visit the gym two times a week and run once a week.
Observation	The user was able to log in and use the application with no problems.
Feedback	She said that she liked the design of the application and an application like this could help encourage her to exercise more.

User two:

Name:	Pete
Age:	52
Occupation:	Builder
What training they do	Five gym sessions per week, one run per week
Observation	He seemed confused at times, especially on the home page, not sure if he understood what the RSS feed was.
Feedback	Liked the application and said that the application would mean that he wouldn’t need to take his notepad and pen to the gym with him. He also added that the run tracking may encourage him to run more.

User three:

Name:	Gareth
Age:	52
Occupation:	Driving instructor
What training they do	One gym session per week, Three runs per week.
Observation	He managed to sign up and make an account.

Feedback	He liked the cardio page as it would allow him to have statistics of his runs (he doesn’t currently use an application to track his runs) and said he would be interested in downloading the application once it was released.

User four:

Name:	Jack
Age:	20
Occupation:	Undergraduate media student
What training they do	Six training sessions per week in the gym, no external running.
Observation	No struggles with sign up or cardio usage but seemed confused about the RSS feed on the home page.
Feedback	Isn’t sure if it would replace his current application because he doesn’t run a lot. However, he mentioned that the EXP function could make him want to exercise more as he enjoys competitiveness.

User five:

Name:	Sarah
Age:	28
Occupation:	Housewife
What training they do	One to Two gym sessions a week
Observation	She seemed to find using the application easy.
Feedback	She mentioned that push notifications may encourage her to use the application more often.

2. **Scoring table using grading criteria:**

Users	Overall design	Ease of use	Useful feature(s)
User 1	10	10	All of them
User 2	9	8	Cardio & Profile
User 3	10	10	Cardio
User 4	10	9	Connecting to Facebook, Home
User 5	10	10	All of them