# Design and Development of a Fitness Application

# Final Year Project

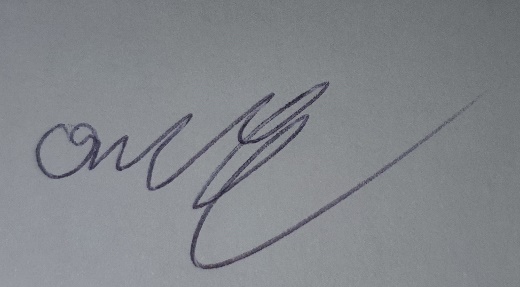
## 2021

# STATEMENT OF ORIGINALITY

##### CS3D661 Individual Project

This is to certify that, except where specific reference is made, the work described within this project is the result of the investigation carried out by myself, and that neither this project, nor any part of it, has been submitted in candidature for any other award other than this being presently studied. Any material taken from published texts or computerized sources have been fully referenced, and I fully realize the consequences of plagiarizing any of these sources.

Student Name (Printed): Owain Mahoney

Student Signature: ………………

Registered Course of Study: Computer Science

Date of Signing: 30/04/2021

# Abstract:

As time advances more and more people are finding it easier to involve technology in their activities such as exercise. It could be helpful to track and monitor, helping to improve their experience while exercising and hopefully giving people more reason to exercise and improve their own health. This project will endeavour to help people do this, by developing a mobile application to track the movement of people using GPS to observe a user's exercise routine and give back helpful information such as distance travelled, calories burned and the speed at which they are currently travelling. In conclusion, this project with exception of a few bugs that are easily fixed as explained in this report, this was a particularly good success at using GPS location to inform the user of useful information and store it for later analysis.

Contents:

[Design and Development of a Fitness Application 1](#_Toc70673065)

[Final Year Project 1](#_Toc70673066)

[2021 1](#_Toc70673067)

[STATEMENT OF ORIGINALITY 1](#_Toc70673068)

[Abstract: 2](#_Toc70673069)

[Introduction: 4](#_Toc70673070)

[Project Background & Problem Statement 4](#_Toc70673071)

[Aim & Objectives. 4](#_Toc70673072)

[Reasons for this Project 5](#_Toc70673073)

[Language of choice 5](#_Toc70673074)

[Background research: 6](#_Toc70673075)

[Existing Solutions: 6](#_Toc70673076)

[Samsung health: 6](#_Toc70673077)

[Fitness 22: 7](#_Toc70673078)

[Leap Fitness Step Counter: 9](#_Toc70673079)

[RunKeeper: 10](#_Toc70673080)

[For my solution: 10](#_Toc70673081)

[Xamarin: 11](#_Toc70673082)

[Microsoft Visual Studio 2019: 11](#_Toc70673083)

[MVVM Pattern: 12](#_Toc70673084)

[SQLite: 13](#_Toc70673085)

[Geolocation: 13](#_Toc70673086)

[Threading Tasks: 13](#_Toc70673087)

[System Diagnostics: 13](#_Toc70673088)

[Research Summary: 13](#_Toc70673089)

[Prototyping: 14](#_Toc70673090)

[Back-end Prototype: 14](#_Toc70673091)

[Back-end Methods: 14](#_Toc70673092)

[Back-end Model: 14](#_Toc70673093)

[Front-end Prototype: 15](#_Toc70673094)

[Methodology: 17](#_Toc70673095)

[Front End: 17](#_Toc70673096)

[Main Page: 17](#_Toc70673097)

[Exercise Page: 17](#_Toc70673098)

[Progress tracker Page: 18](#_Toc70673099)

[Profile Page: 19](#_Toc70673100)

[Back End: 20](#_Toc70673101)

[Main Page: 20](#_Toc70673102)

[Exercise Page: 20](#_Toc70673103)

[Profile Page: 20](#_Toc70673104)

[Progress Tracker Page: 21](#_Toc70673105)

[Testing: 21](#_Toc70673106)

[What we are testing: 21](#_Toc70673107)

[How it is being Tested: 22](#_Toc70673108)

[Results: 23](#_Toc70673109)

[Evaluation: 23](#_Toc70673110)

[Aims and objectives: 23](#_Toc70673111)

[Research: 23](#_Toc70673112)

[Language of choice: 24](#_Toc70673113)

[For my Solution: 24](#_Toc70673114)

[Prototyping: 24](#_Toc70673115)

[Methodology: 24](#_Toc70673116)

[Testing: 25](#_Toc70673117)

[Future progress: 25](#_Toc70673118)

[Appendix: 27](#_Toc70673119)

[Code: 27](#_Toc70673120)

[Bibliography: 27](#_Toc70673121)

[Appendix II: 28](#_Toc70673122)

# Introduction:

## Project Background & Problem Statement

Lately more and more people are trying to get into exercise, more than one in four Americans currently use one product or the other: 8% are actively using only a fitness tracker, 9% are actively using only a mobile health app, and 10% are actively using both. Combining both current and past users, nearly half of Americans (45%) have at least tried one or more of the digital health products: 13% have tried or currently use only a fitness tracker, 12% have tried or currently use only a mobile health app, and 20% have tried both products. (Gallup, 2020). Also, people like to keep accurate records of their achievements whether that is distance achieved, or time spent doing an activity. Something this project end goals is to aid this by keeping accurate records during the activity such as running, walking and other forms of exercise.

Main parts of this project will include using GPS location to track the user location on their phone as they do the activity so it will have to be built in a mobile application connected to a database and make live updates to it. Other uses can be using simple math workout speed and overall distances. To do this the application would need the user’s personal data such as weight, age, gender, and height. (Create a Fitness App l Guide: Tech Stack, Cost, Features, 2020)

## Aim & Objectives.

The aim of this project is to build a mobile application for users to able to track their progress during and after exercise. Other objectives of the project are as follows:

* To use secure programming practices such as MVVM. To ensure decent quality code such as ready ability and efficient algorithms.
* To do background research on other fitness application to see what I could include in my solution.
* The running algorithm gives the user accurate outputs all the time even during the time it is in use.
* The application is easy to follow and does not confuse the user i.e., the units will be shown in either miles or kilometres and the app will allow for easy navigation.
* To adhere to the Gantt chart in all timings for the project to be completed on time.

## Reasons for this Project

Following the stats from (Anurina, 2021) In the first quarter of 2020, the number of downloads of health and fitness apps reached 593 million. Health and fitness app statistics project that by the end of 2020 apps from this category will have been downloaded 656 million times. It could be worth quite a lot of money to develop an application like this. In particular, the application that suffered the most from this phenomenon were applications that focused on outdoor activities such as walking, running and cycling. The reason they suffered in 2020 was due unprecedented circumstances that affected the ability of these application to be effective, as people couldn’t exercise as freely outside.

## Language of choice

Xamarin was chosen as the language of choice due to it being a cross platform framework, one of the main advantages of this language is that it easily lends itself to C#, due to C# being a supporting language of Xamarin. The Integrated Development Environment (IDE) that will be used is Visual Studio, which will allow for testing and the ability to easily identify errors. The use of Xamarin also comes with a few downsides, such as the file size being on the larger side, other issues could be the application will be restricted to being a simple application and be incapable of supporting heavier graphics. Luckily, graphics is not the focus of this project, so this is mitigated (Mahoney, 2020)

Other languages that were considered for use within this project include Java and Kotlin. The first and foremost advantage of using Java for app development is that Java supports OOPS (Object Oriented Programming) concepts and is more competent as they are scalable, extensible, and flexible. It comes with a rich library of default design patterns and other best practices. Java is one of the most used programming languages to develop any good business mobile app. And for entrepreneurs and huge businesses, it can provide the best suitable application that is user-friendly. Kotlin can easily get rid of aggravations as well as obsolescence of Java. Kotlin is a clear compact and dynamic language. Kotlin can maximize the productivity of the developer’s team as it takes very little time to write and also you can deploy it pretty fast. The reasons that Java was not chosen for this project are that the programmer for this project is not very proficient in the use of Java and would take extra time to learn this language. The uses of Kotlin would restrict the user to have to use an Android phone and Apple would not be compatible with these languages. The same is true for languages such as swift and Objective-C which are created solely for apple products.

Other parts of this project that were being considered were the different architectures or framework. One such Framework was MVC, which is an architectural pattern that separates an application into three main logical components Model, View, and Controller. Hence the abbreviation MVC. Which is described by a website (MVC Tutorial for Beginners: What is, Architecture & Example, 2021). Another website gives a few advantages of using this framework, one such advantage is that MVC supports rapid and parallel development. If an MVC model is used to develop any particular web application, then it is possible that one programmer can work on the view aspect while the other can work on the controller aspect to create the business logic of the web application. Following this logic, this way, the application developed using the MVC model can be completed three times faster than applications that are developed using other development patterns. (Six Benefits Of Using MVC Model For Effective Web Application Development, 2021) whereas another site gives disadvantages and the main reason I did not go with this framework It is hard to understand the MVC architecture. MVC architecture must have strict rules on methods. There are not many disadvantages to this type of architecture, and the disadvantages are not so large and are very easy to ignore in comparison with all the benefits we get. (Jithin, 2016)

# Background research:

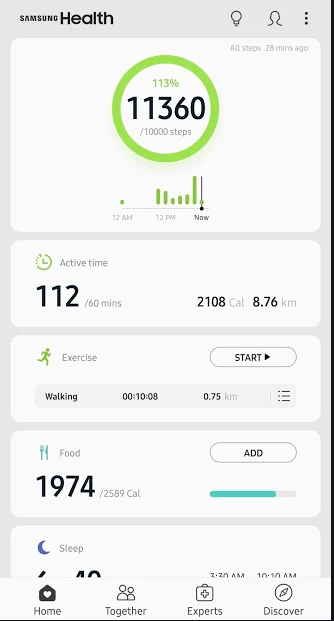
Researchers like (Gabbiadini 2018) say “Recent studies suggest that about 6 out of 10 users have installed a fitness tracking application on their smartphone.” But it also shows that “more than 59% of adults do not engage in sufficient daily physical activity and much remains unknown with regard to the effectiveness of mobile applications.” Which would mean that fitness application must be at least to some people not good enough reason to do enough exercise. Some people go the other way entirely even going to say that fitness application can be a detriment to people doing exercise, as stated by (Concannon, 2020) “people who use the app to praise and public endorsements are more likely to become obsessed with exercise and suffer high-stress levels in the long run.” As well as “fitness sharing apps can certainly help seed and sustain exercise routines, but there is a danger that some users may develop obsessive tendencies.”

Other areas of health applications that have been researched include calorie burn rate depending on the exercise (Loria, 2017) “Fortunately, there's a science-backed way to calculate how many calories you burn doing almost anything. Sure, there are apps out there that will help you calculate how many calories you burn on your run or your bike ride, but this goes deeper than that.” This equation will be used in my program to determine how much a user might burn, (Loria, 2017) later explains that every activity has a value called a "MET value" (Metabolic equivalent) which calculates the energy required for that activity. One MET is approximately 3.5 millilitres of oxygen consumed per kilogram (kg) of body weight per minute as said by (Roland, 2019). Using the MET value, you can work out the calories burned during exercise (Loria, 2017) Multiplying MET value by weight in kilograms tells you calories burned per hour. (Roland, 2019) A MET is a ratio of your working metabolic rate relative to your resting metabolic rate. Metabolic rate is the rate of energy expended per unit of time. It is one way to describe the intensity of an exercise or activity. So, from this research will need to be done on the different MET values for the different exercises which (ProCon.org, 2019) covers extensively and will give the values that will be used in the program.

## Existing Solutions:

### Samsung health:

Alternative solutions include Samsung Health, which is a built-in application to all Samsung phones. The app has multiple features including handling the service connection, as well as inserting, reading, updating and deleting health data. The application holds and manages predetermined data types such as counting steps, done via pedometer, and nutrition values. Other parts of the application need to handle checks and requests for permissions to handle health data from the user. Samsung Health Android SDK's Service provides 3 tracker tile templates, each tile template is designed for its proper usage and different tile templates can be chosen whenever your application posts the tracker's tile. Things that were liked about this project was the different attributes that were on display, such as today's calories burned, steps and exercise completed. The project could try to replicate having useful information on display on the main page rather than a separate page for it. Also, the navigation at the bottom is simple but clear in its purpose and where each button leads.



This picture is figure 1.1 of Samsung’s main/first page and main display or information.

### Fitness 22:

This was application was found by the recommendation of (Hindy, 2020). Fitness22 Running Distance Tracker is exactly what the name implies, it tracks your distance if you decide to take a walk or a run. It has a bunch of standard features, like total distance, average pace, pace per mile (or kilometre), a run log, and other helpful features. You can also pass your music through the app for some motivation and the app tells you when you have run a mile (or kilometre) and what your pace was during that mile (or kilometre). This application had the best-looking UI with dark background and the white and green words and numbers this makes it easier for the user to see that important information clearer. This project might use a similar output display, on the tracking page as its simple but the useful the user while they are exercising.



Figure 1.2 of Fitness 22 coach page displaying exercise information of the user.



Figure 1.3 Fitness 22 exercise tracking page showing live data to the user.

### Leap Fitness Step Counter:

Also recommended by (Hindy, 2020) was this application. Leap Fitness Step Counter is a good, simple step counter app. It tracks your steps, requires no sign-in, and it does not use GPS tracking for your steps. That makes it an excellent app for privacy and simplicity. It only tracks your steps, tracks your progress, and make sure you hit your goals every day. There are some backups and restore options along with some visual theming options. There is a free version and a pro version that removes ads. It is a competent step tracker app for those who need something basic. Like the previous example this solution has similar output, they both do not show information such as how many glasses of water the user has drank through the day. The fancy graphics are nice to look at but don’t add any functionality to the application, the same could be achieved by just showing a percentage or state number of steps out of target number of steps and you would get the same information.



Figure 1.4 main page of Leap Fitness Step Counter showing today’s date to the user.

### RunKeeper:

RunKeeper tracks walks, runs, and any other physical activity. From the casual stroller to the 5K runner and those building up to a marathon, this GPS app boasts a community of 50 million users and is suitable for everyone. You can build, save, and discover new routes with GPS to keep your workout stimulating. It is easy to stay motivated with RunKeeper. You can join challenges to push yourself, win rewards for your workouts, and share your progress with friends. You can also create your own challenges, invite friends to join you, and encourage and cheer each other on along the way. From this example you can see that sharing your information as way to motivate people to do exercise, this could help with the some of the down sides that fitness application has shown to have.

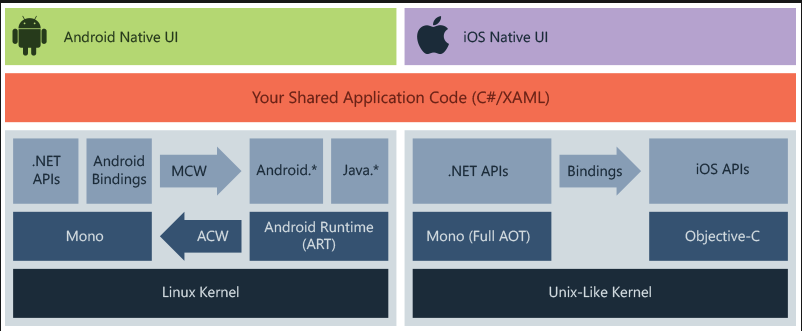
## For my solution:

Research for this project was more directed the learning the language and framework of Xamarin. Main points of interest were the GPS plugin and SQLite and the data required to make the app function as stated in my aims and objective of the project.

### Xamarin:

(Johnson and Britch, 2020) explains Xamarin is an open-source platform for building modern and performant applications for iOS, Android, and Windows with .NET. Xamarin is an abstraction layer that manages communication of shared code with underlying platform code. Xamarin runs in a managed environment that provides conveniences such as memory allocation and garbage collection.

Xamarin enables developers to share an average of 90% of their application across platforms. This pattern allows developers to write all their business logic in a single language (or reuse existing application code) but achieve native performance, look, and feel on each platform.



Xamarin.Forms is more than just a cross-platform UI library, it is a full application framework that includes everything you need to build mobile apps. This includes cross-platform navigation, animation APIs, dependency service and a messaging centre.

Advantages to using Xamarin given by (Johnson and Britch, 2020) is the complete binding for the underlying SDKs – Xamarin contains bindings for nearly the entire underlying platform SDKs in both iOS and Android. Additionally, these bindings are strongly typed, which means that they are easy to navigate and use and provide robust compile-time type checking and during development. Strongly typed bindings lead to fewer runtime errors and higher-quality applications. Modern language constructs – Xamarin applications are written in C#, a modern language that includes significant improvements over Objective-C and Java such as dynamic language features, functional constructs such as lambdas, LINQ, parallel programming, generics, and Xamarin.Essentials offers a unified API to access common resources across all three platforms. Shared code can significantly reduce both development costs and time to market for mobile developers.

# 

### Microsoft Visual Studio 2019:

Microsoft Visual Studio is described by (Microsoft Visual Studio - Wikipedia, 2021) as an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) from [Microsoft](https://en.wikipedia.org/wiki/Microsoft). It is used to develop [computer programs](https://en.wikipedia.org/wiki/Computer_program), as well as [websites](https://en.wikipedia.org/wiki/Web_site), [web apps](https://en.wikipedia.org/wiki/Web_app), [web services](https://en.wikipedia.org/wiki/Web_service) and [mobile apps](https://en.wikipedia.org/wiki/Mobile_app). Visual Studio uses Microsoft software development platforms such as [Windows API](https://en.wikipedia.org/wiki/Windows_API), [Windows Forms](https://en.wikipedia.org/wiki/Windows_Forms), [Windows Presentation Foundation](https://en.wikipedia.org/wiki/Windows_Presentation_Foundation), [Windows Store](https://en.wikipedia.org/wiki/Windows_Store) and [Microsoft Silverlight](https://en.wikipedia.org/wiki/Microsoft_Silverlight). It can produce both [native code](https://en.wikipedia.org/wiki/Machine_code) and [managed code](https://en.wikipedia.org/wiki/Managed_code). Microsoft Visual Studio, like any other [IDE](https://en.wikipedia.org/wiki/Integrated_Development_Environment), includes a [code editor](https://en.wikipedia.org/wiki/Code_editor) that supports [syntax highlighting](https://en.wikipedia.org/wiki/Syntax_highlighting) and [code completion](https://en.wikipedia.org/wiki/Autocomplete) using [IntelliSense](https://en.wikipedia.org/wiki/IntelliSense) for [variables](https://en.wikipedia.org/wiki/Variable_(programming)), [functions](https://en.wikipedia.org/wiki/Subroutine), [methods](https://en.wikipedia.org/wiki/Method_(computer_science)), [loops](https://en.wikipedia.org/wiki/Program_loops), and [LINQ](https://en.wikipedia.org/wiki/LINQ) queries. IntelliSense is supported for the included languages, as well as for [XML](https://en.wikipedia.org/wiki/XML), [Cascading Style Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets), and [JavaScript](https://en.wikipedia.org/wiki/JavaScript) when developing web sites and [web applications](https://en.wikipedia.org/wiki/Web_application). Autocomplete suggestions appear in a [modeless](https://en.wikipedia.org/wiki/Modeless) [list box](https://en.wikipedia.org/wiki/List_box) over the code editor window, in proximity of the editing [cursor](https://en.wikipedia.org/wiki/Cursor_(computing)). In Visual Studio 2008 onwards, it can be made temporarily semi-transparent to see the code obstructed by it. The code editor is used for all supported languages.

### MVVM Pattern:

The first thing that was researched in this project, was the MVVM model due to it being a good layout for the code in the effort to try keep the code manageable and easy to read.

The first part of the MVVM is the model. The model is referred to as the domain object. The model represents the actual data and/or information we are dealing with. The key to remember with the model is that it holds the information, but not behaviours or services that manipulate the information. It is often a challenge to keep a model completely “clean.” By this a true representation of “the real world.” For example, a contact record may contain a last modified date and the identity of the modifying user (auditing information), and a unique identifier (database or persistence information).

The view is what most of us are familiar with and the only thing the end user really interacts with. It is the presentation of the data. The view takes certain liberties to make this data more presentable. In MVVM, the view is active. As opposed to a *passive view* which has no knowledge of the model and is completely manipulated by a controller/presenter, the view in MVVM contains behaviours, events, and data-bindings that ultimately require knowledge of the underlying model and view model.

The view model is a key piece of the triad because it introduces *Presentation Separation*, or the concept of keeping the nuances of the view separate from the model. The view model also exposes methods, commands, and other points that help maintain the state of the view and manipulate the model as the result of actions on the view, and trigger events in the view itself.

The first thing that was found from the View Model research is the need to use binding context to connect the view to this part of the code and that the view part only needs to declare and call like any class.

(Model-View-ViewModel (MVVM) Explained, 2021)

MVVM allows for separating the visual part of the app (the user interface, or UI) from the related code it becomes possible to have specialists in each area work on related items at the same time. The theory is designers can work on the UI at the same time as developers are working on the code without needing to have them both work on the same files at the same time.

MVVM is not great for every situation as described by (Copsey et al., 2017) “For simple UI, M-V-VM can be overkill. In bigger cases, it can be hard to design the ViewModel up front in order to get the right amount of generality. Databinding for all its wonders is declarative and harder to debug than nice imperative stuff where you just set breakpoints.”. This does not apply to this application because it is not a small project and requires a level a readability that is only accomplished using the MVVM structure

### SQLite:

SQLite is an embedded SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file. SQLite is a compact library. With all features enabled, the library size can be less than 600KiB. SQLite generally runs faster the more memory you give it. Nevertheless, performance is usually quite good even in low-memory environments. SQLite is very carefully tested prior to every release and has a reputation for being exceptionally reliable. Most of the SQLite source code is devoted purely to testing and verification. (About SQLite, 2021)

### Geolocation:

The Geolocation class is available at Xamarin.Essentials API. Xamarin.Essentials plugin provides 20+ cross-platform APIs for mobile application development. Xamarin.Essentials API works with all Xamarin. Forms, Xamarin. Android, Xamarin. iOS, or UWP applications that can be accessed from shared code. It is used to retrieve the device's current geolocation coordinates. Android, iOS, and UWP offer unique operating system and platform APIs that developers have access to all in C# leveraging Xamarin. This will be the most important part of this project as this enables the developers to actually track the user. Having a good tracking library will mean that the application user can be tracked accurately without the use of complex code.

### Threading Tasks:

Async programming is all the rage in mobile app development for good reasons. Using async methods for long running tasks, like downloading data, helps keep your user interface responsive, while not using async methods, or the improper use of async/await, can cause your app’s UI to stop responding to user input until the long-running task completes. This will make my program run more efficiently and appear better to the user, however there are issues with this process. (Jon, 2017) also explains that any method that calls an async method (i.e., a method that has the async keyword in its signature) should use the await keyword when calling the async method. Not using the await keyword when calling an async method can result in exceptions that are thrown being [swallowed by the runtime](https://blogs.msdn.microsoft.com/pfxteam/2011/09/28/task-exception-handling-in-net-4-5/).

### System Diagnostics:

The library for system diagnostics is usually used to debug or measure the performance of code. In this project the stopwatch class from the system diagnostics namespace will be used to measure the amount of time that the user has been doing the exercise. Other uses of system diagnostics given by (Namespace: System.Diagnostics, 2002) include the EventLog component, which provides functionality to write to event logs, read event log entries, and create and delete event logs and event sources on the network. The EntryWrittenEventHandler provides a way to interact with event logs asynchronously. Supporting classes provide access to more detailed control, including permission restrictions, the ability to specify event log types (which controls the type of default data that is written with an event log entry), and iterate through collections of event log entries. For more information about these tasks, see the EventLogPermission, EventLogEntryType, and EventLogEntryCollection classes.

### Research Summary:

From this, research has shown that many people have tried to do this sort of thing before with mixed results, a lot of the time many people find it useful, but a lot more people could use this kind of application. There is evidence suggesting that mobile phone fitness applications can be counterproductive as people can find it off putting from doing exercise which is not good for people’s health. So, for this project I will try to make sure it is more inviting but gives the user as much control as possible.

Research done on exciting solutions shows there are many already available on the market and going in to the few that have been researched, it seems that there are many similarities, such as the design, or the tracking points like steps, calories, and distance. This research will influence the design of the project using the good parts of the designs and outputs this project can use for its own.

It was particularly important to research the different libraries that the project will be using. It showed how each of the library’s worked so development would go a lot faster and less likely to run into bugs and errors, also this allows for cross platform use as all the libraries are able to be used by both Apple and Android phone users. This research will expedite the development process, as the development team will not be bogged down by errors, and it take too long to discover bugs.

The research into MVVM structure allowed the project to have cleaner code, and stopping code being bundled in a big and unmanageable script that would be more prone to errors and bugs. It also allows one use binding context to monitor user making changes in the view using INotifyPropertyChanged. This research will affect the project by enforcing best coding practice.

# Prototyping:

## Back-end Prototype:

### Back-end Methods:

GPS request: Retrieving a location class from the method created by the GetLocationAsync method which creates an instance of the location class and populates it with the attributes. This method takes place during a in progress exercise.

Data process: After retrieving location data, the information is then processed in the View Model by translating the longitude and latitude attributes from the new point and the last point we can use the distance from the kilometres function then display this information for the user to see.

Data entry: When the data been processed the new location point that was retrieved is saved in the on-phone system database setup with SQLite, the saved information does by inserting a new class of point and using the Insert Async which add the new point to the database asynchronously. This data can be then later accessed.

Personal data retrieval: when data is required for the display or for editing, the data is needed to be retrieved so that it can be edited, a few examples such as main page data to be shown, the profile page where attributes such as name, age, weight and height can be changed etc other example include during exercise when calculating total distance of the user and other such relevant things.

### Back-end Model:

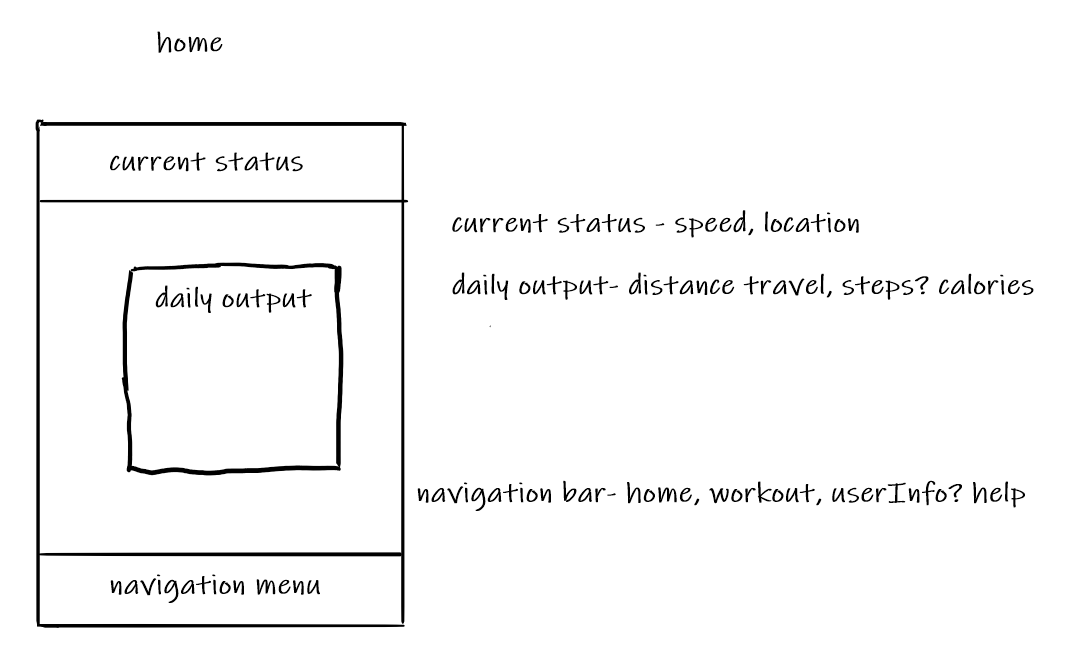
The model that will hopefully be employed will be the SQLite tables that are formatted in a CS file. These tables will be for a user table which will hold current data, such as name and other personal information, some of which that will be used to augment and make more accurate in the algorithm running, other points to save in the person table is the data collected on the users from the algorithm tracking the users exercises.

Other models in the background are the GPS request information, when the algorithm needs this information the model generates the information using the model information like the accuracy of the location data how long the system should wait before timing out and the result is given as an exception and some alternatives to this problem.

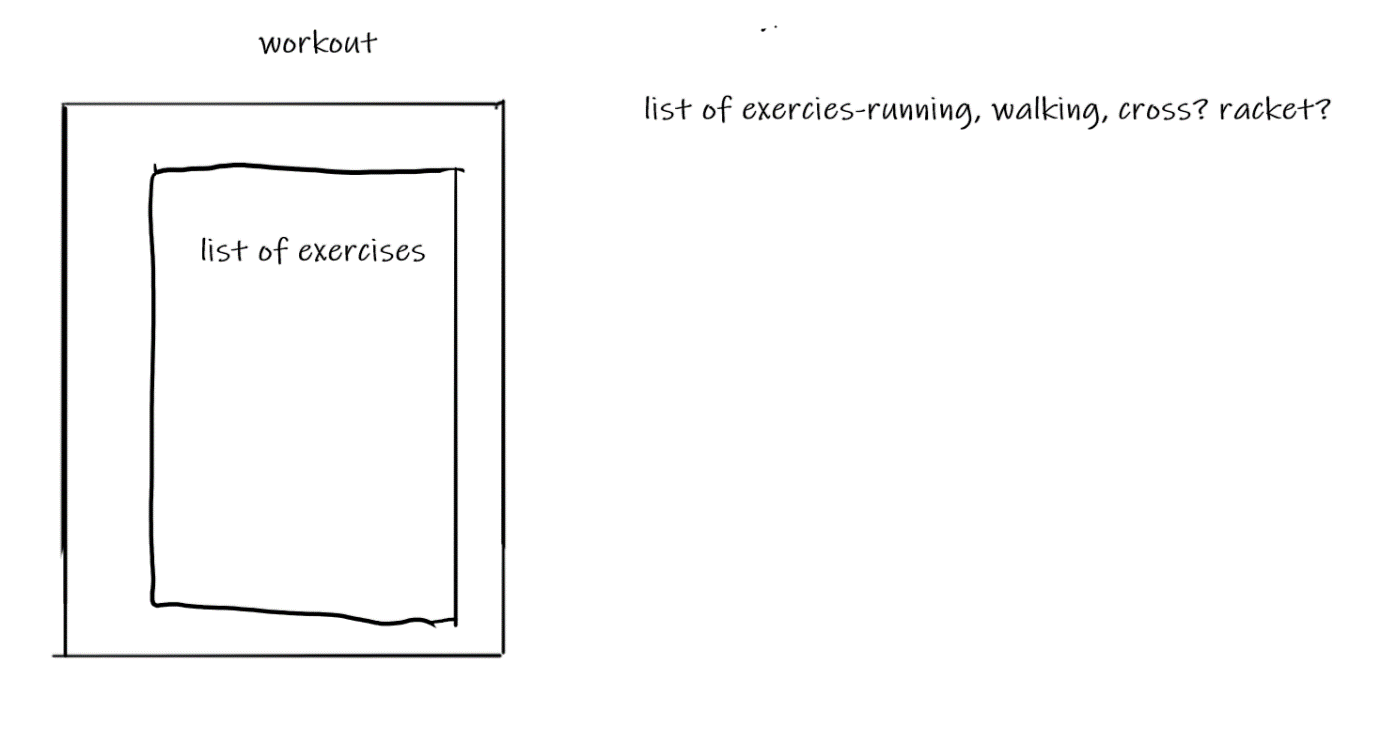
## Front-end Prototype:

The figures 2.1, 2.2 and 2.3 are early design ideas for the application.

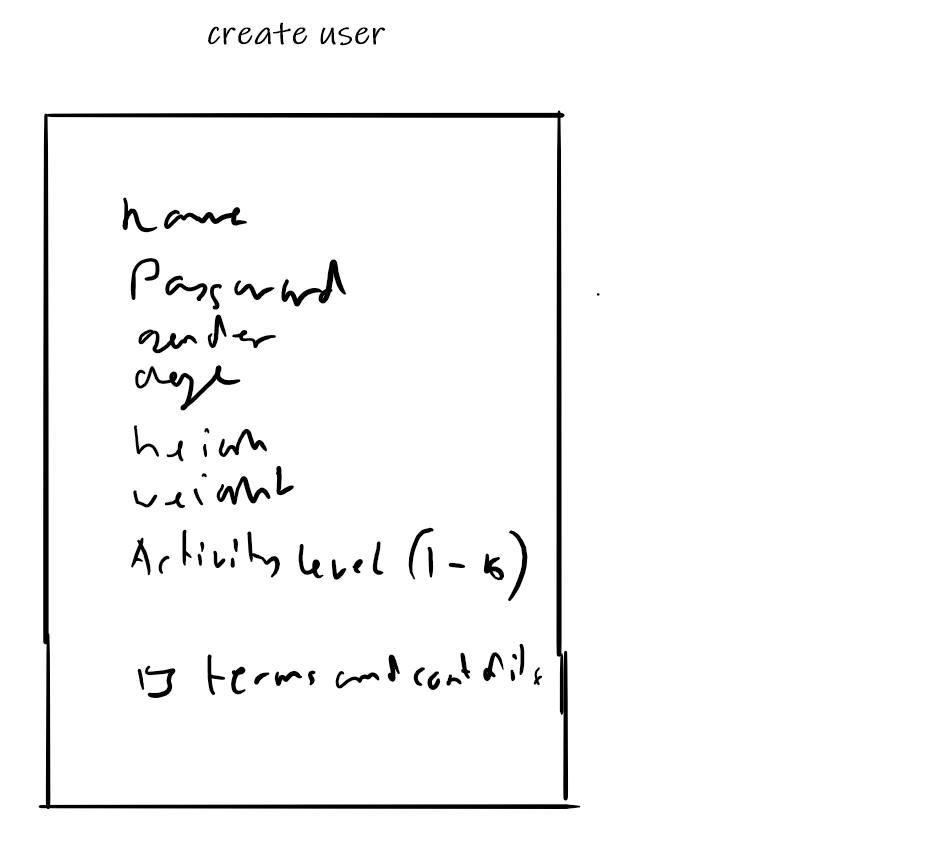
The first one figure 2.1 shows the home page that will show generic data about the person exercise history such as distance travelled and other such units.



Firgure 2.2 would be a small menu for the user to choose the excerise they decirded. Examples of the excerise than can be chosen are running, walking and racket sports.



Firgure 2.3 has been labeled create a user which is true for first time users but if the the user has already created an account they are able to edit and make chages to the profile in the same area which will automatcally load their current data in, then when changes are wanted to be made, they can do it here which prevents having to make a different page and all new code.



# Methodology:

## Front End:

The front end part of this project will be encoded in Xamarin’s XAML file which will make up the GUI. This will allow users to view current states, make changes to profile and create new recordings of exercises, this is all made possible by Xamarin front end extensions such as Xamarin essentials.

### Main Page:

The main page includes running data of the users’ previous exercise data. To show this, a simple GUI is created from a XAML file which is bonded to the ViewModel where the data is received, data displayed is distance gone the day of completion, calories on the day of completion and overall distance for the user. Also included on this page is navigation bar which can change to any of the other pages such as exercise or profile editing. The navigation bar is not represented by traditional images and is instead represented by text, which allows for the app to be more intuitive overall.

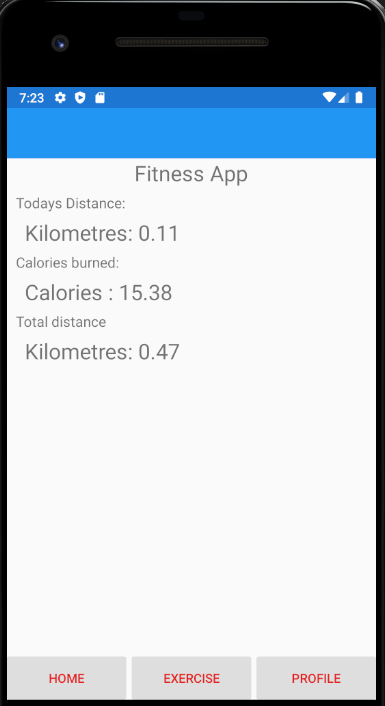


Figure 3.1 The main page of the application show the user useful information.

### Exercise Page:

This page’s main function is to allow the user to choose between which exercise they are about to do; this leads to more accurate results for calorie consumption during exercise. The choices between exercises, are at this moment, only running, walking, and cycling. These are all shown as image buttons alongside the names of the exercise, this makes it more difficult to make mistakes when making the choice between exercises. Once selected, the system knows what exercise the user intends to do and then apply the correct filters for the algorithm in the progress tracker page.

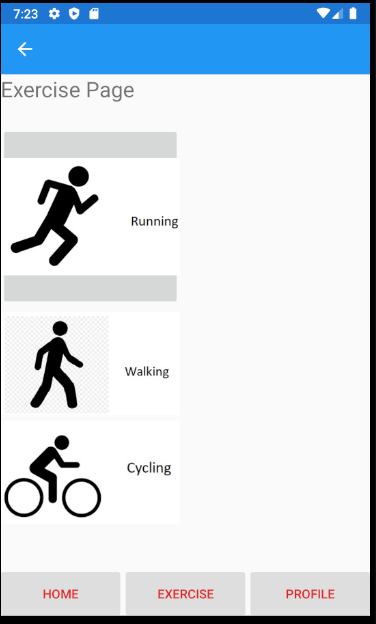


Figure 3.2 the exercise page with 3 different options of exercise to choose.

|  |
| --- |
| Progress tracker Page: This is the page that tracks the progress of the user during their chosen exercise. The output displays in this screen are distance travelled so far in kilometres, the speed the user is currently travelling at in kilometres per hour, the current user calories burnt during the current exercise and the time the exercise has been going on. Other user interfaces on this page are the back button which returns you to the previous page. When you first load the page, the only button that visible is the start button, but when the application is running the tracking algorithm, the start button is hidden, and the exercise pause button and the end button are shown. When the pause button is pressed, the pause button hides, and the start button appears, and the algorithm stops. If the end button is pressed the page is given a prompt asking if the user is sure, they would wish to end the exercise to avoid unwanted clicks. After the prompt, the application returns to the main page.    Figure 3.3 the workout tracking page with display output for the user Profile Page: The profile page is where the user can make changes to their information that is being stored in the application. This includes first name, last name, age, height, weight. All this can all be saved when the user uses the button marked save to save the current information written into the information entry points which are then assigned as values. The entry for the information that requires only numbers such as age, weight, and height the keyboard is set to numeric, this is to make sure that the user cannot enter characters such as letters and special characters to a field that should be only numbers. At the bottom of the screen there is a navigation bar that has home and exercise button and profile button which navigate to their respective page, except for the profile button, which is there simply for cosmetic consistency.  Figure 3.4 profile page with entry fields for the user to input information to save on the phone. Back End:Main Page: When the main page is first loaded the function OnAppearing runs a parallel function Load command, this command is to retrieve the data from the person table in the SQLite database then the relevant data is assigned to variables that are bound to the View XAML file which creates the output for the front end. The program also checks if the data saved each time the user enters the main page, if the current date attribute day is less than the day that is stored in the database if so, it set the Today distance and calories to 0 as this marks that a new day has passed for application. The navigation bar clicked functions is assigned to functions written in the main page XAML class file, the function adds a new form of the relevant page to the model stack. Exercise Page: Navigation bar is set up so that when the home button is pressed it pops all of the model stack till it has reached the root of the stack i.e., the main page. The image buttons main difference from each other is the value they pass in the model push async function call, this is to tell the programme which MET value to use in the Progress tracker page. Profile Page: The profile page navigation is similar to the exercise page ie the home button pops the model stack to the root and the exercises button pushes an exercise page on to the model stack. In the profile viewModel file is most of the code for this page this file holds the variables that the XAML file use to display the relevant information in the entry point, this is done by binding these values rather than naming the entry points as individual variables. This allows for the program to monitor the changes the user does in the entries and update the person class as and when they make changes. To confirm the changes the user has made the user must use the save button, as this is how the data stored in the person class is actually saved to the database using the update person function, which is connected to the database operation file and runs the SQLite command update command with the person class template to update the correct record. If the user leaves the page before confirming the changes, the data they entered into the entries will be lost, and will not be there if the user return to the page. Progress Tracker Page: Unlike the other pages, this page does not have a navigation, but it uses the built-in back button which pops this page and returns to the exercise page. When the page first runs the load function to get the user’s information, when the user presses the start button, the pause and end buttons are shown by assigning true to Boolean values that are bound to the visibility attribute of the buttons and false for the start button. The command that is connected to the start button runs the main part of the page and sets a Boolean value for marking if the algorithm should be running called in progress.  After this the algorithm creates a start point by using the GPS request using the function created in the model that runs the Xamarin essentials GPS request with the all the needed information that’s covered in the research and sets the location that is returned then it enters the loop which takes another GPS request and assigns this to a new location variable so the algorithm can use the location function that takes longitude and latitude coordinates from two locations in this instance it would be the current location and the new one just created from the GPS request this gives a distance in kilometres then this value is added to the distance value, other values that are being tracked can be then derived from the new location such as speed. Calories can then worked out by following the equation that can be found in the background research but divide by 3600 which gives the number of calories burned in one second of exercise, the MET value is passed in through depending on the exercise chosen by the user, and the weight is also depending on the user input in the profile page if the user hasn’t given a weight the program uses a default value of 80kg. So, if the user wants a more accurate calorie count, they should enter their own weight in the profile page before using the application tracking, but it’s not required for use of this application.  After this the new point is then assigned the current point and the program is then stalled for a one second by the delay wait function in the Task Threading library, the program then adds the current point to database by the database function AddPoint. This marks the end of algorithm so to jumps back to the point where the program does another GPS request and repeats the algorithm again until the user interacts with the GUI button such as the pause the button this runs the pause command which sets the InProgress flag to false and then prompting the user that the program has paused and there will be no more tracking of data. This also leads to the pause button visibility to be false and start button to be true so the user can start the exercise from that point. The other button that the user can press is the stop/Exit button this also prompts the user but the user can choose to confirm this with a yes or no answer is no the program continues from its current state either in paused or still tracking the user, if the user chooses yes the tracking ends and the current distance and calories the program has acquired from tracking the user is add to the user current record of this elements like today distance, calories and total distance, after this the update user function is then add to the SQLite database using the UpdatePerson function from the operation file, then the program returns to the main page by popping to root of the model stack. Testing:What we are testing: The testing of this program involves the software being run in an emulator, which can replicate a person moving and doing exercise. This was done in the mobile android SPK manager for the main and most important part of this project, but it will need more testing in other parts of the project such as, the profile editing page that updates of the user information. If the navigation bar works by sending the user to the right pages and is consistent on every page. The main page needs testing the results are accurate results are being displayed to the program GUI output. Other tests will include the MET value of the progress tracking page that is received from the previous page. How it is being Tested: To test the navigation bar at the bottom of the pages the program will be run in android google Pixel emulator then using the buttons and seeing if the correct page loads in the correct way. Second major test will be the profile save or update functionality so what should happen is the contents of the entry fields should be compiled into a person object and then stored in the SQLite database, which we can check on the phone setting after completion. The last big test we set a route on the phone emulation that we know the accurate distance for and how fast the emulator travels along the route, with this information we can work out what the actual values should be, and we can use them to compare them with answers the algorithm gives us after completing the route given. To test the reset of the daily information such as today's distance and calories we must move the Clock of the emulator a day on and see if the system recognise it as a new day and will reset the daily information. A test plan would look like a name, small description of how you would be testing the application, the desirable outcome of the test, if the test was successful, and comments on the test most likely when the test fail how it failed and how could the fixed.    Figure 4.1 shows example test plan.    Figure 4.2 shows the application being tested in an Android Google Pixel emulator Results: The navigation bar buttons loaded all the correct pages, the only problem that might arise is if the user switches between the profile and exercise page as it just builds on the model stack until an error occurs, which typically happens when the model stack is filled, but this very unlikely as it you would have to do it many times and if the user press home once the program pops to the root model of the stack emptying the stack. The profile saves or update functionality is tested by entering test data into the entry fields and then pressing the save button, by checking the internal storage of the phone to see what information was saved in the phone through SPK device manger. The next major test was the overall tracking ability of this application and its accuracy. To test this, we set a route of 1 kilometre we also set the speed of the user being emulated to roughly walking speed estimated to be 2.7 kilometres per hour, which would mean it should take 37 minutes to complete and burn on average of 212 calories, the outcome that was presented when finishing this test was the distance was correct in there being one kilometre measured. However, the calorie count was slightly higher than expected. With all of the tests we did, on average it came out to be 230 calories burned, this still suggests that the MET value used in the algorithm is still correct and it is needed to be checked by using a debugging process.  To check if the data that we collected was being saved correctly in stored memory we checked using the SPK’s device manager which was shown to have the correct results, after completing we were able to see the buttons the user would be able to interact with while doing an exercise being tracked. All buttons respond to normal programme guidelines such as changing visibility depending upon the different states that the current programme is in.  The results to the test of the main page display updating after completing a tracked workout, it would change when the user does more tracked exercise, and then when the user returns to the main page it updates showing the up to date and relevant data, this was successful as when the user exits from the exercise tracking page it pops from the model stack to its root which is the main page, and the information has changed by a factor of the amount of exercise the user has just completed. when testing the daily reset functionality, the programme functioned as expected, except if the day went from the end of the month to beginning of the next month the programme, it didn't recognise this as a new day to reset with the daily information. This is probably due to the comparison of the new date; it only uses the day attribute but if the programme uses the day, month and year attributes of the date time variable, to use for comparison this would fix the problem because the year part of the variable does not start again or loopback this will rectify that problem as the programme will know that the day has changed moving forward. Evaluation:Aims and objectives: The aims of this project were concise with how the project was to be made but was not truly clear how it was going to be done and were too broad on the approach that should be taken. Maybe to improve them, there could be a few more to do with the development and the time taken to develop the application. Research: With the research of the pro and cons of undertaking this project it suggested that some people would find it useful for improving their lifestyle as way to encourage more exercise, but for others they might download the application but never use it or even worse find it more as turn off for doing exercise as people might not be able to reach their personal goals.  While researching the architecture of this project it was decided it was MVVM practice to use. This was due to extensive research whereupon the development team found a reputable source (Model-View-ViewModel (MVVM) Explained, 2021) which suggested using MVVM for this particular project as it can make the process faster which is good for the time constraints for developing this project.  Research into SQLite was conducted due to prior knowledge of the development team. It was chosen because it was compact language and so not taking too much memory on a smaller device such as mobile phone compared to other languages. This opinion was shared by this source (About SQLite, 2021).  The visual design of this project was influenced by market research into existing solution which helped design team form a design which took the best elements from each previous existing solution.   Language of choice: The Language of choice was more of an easy one, Xamarin, even though it is a new language it still had a lot of helpful documentation and sources for problem solving during the development of the application. Other advantages of the language being cross platform is especially useful with exceedingly minor changes the application can on both Apple and Android phones. the main reason for choosing this language was for its built-in libraries such as Xamarin essentials and SQLite which are the biggest parts to the project. The only downside to using Xamarin is the time it took to teach the developers of this project the language meaning that there was delay in starting the project. For my Solution: The decisions that were made for libraries in the application is believed are good enough to fulfil the desired outcome of this project such as the Xamarin Essentials library this makes tracking the user quite easy and does require a lot of learning time. The same can be said for SQLite as which is light weight database library with also prebuilt C# API function which does take a lot of memory on the user phone. The use of MVVM structure is another desirable choice as it keeps code spaced out and easy to read. Prototyping: The prototype front end designs were very rough as they were free hand drawn and they should have been made in more of a professional manner. A suggested way of improving the professionalism of the prototype designs would be to use a computer software that is specialised in creative design, such as Draw.io or Adobe Illustrator. The designs were not completely adhered to, such as the log in page was dropped entirely as it was later reasoned that it would not be needed. Other parts of the designs that were dropped were the tracking on the home screen and certain trackable things such as steps, but all the functionality is all there it would just need to be added to the database model and the equation to find the number of steps. Methodology: Many issues were faced during this part of the project as there are many bugs and issues with the program. Also, there were quite a number of places that could be improve upon as well, like the counting calories if the user is not moving quick enough or not moving at all the program still assumes they are burning the same number of calories as if they were doing the exercise they were meant to be doing for the user’s choice. This could be rectified by adding a function to derive the MET value by what the user current speed and or other factors. Other improvements I would like to make is how the exercise data is stored, as at the moment as it stands all the points are stored together and not divided very well, if at all, as it is possible to divide the data up by the time if the point is recorded but this is inefficient and does not take into account when the exercise is paused, this could be improved by setting the run id at the start of a exercise so the when you want a particular set of exercise points you only need the run id to retrieve all the relevant point you need. This leads on to the next improvement or step of this project that is to analysis the data that you are collecting, at the moment this is a simple gathering and displays simple information but maybe there could be a page that give information on specific exercise rather than just grouping it all together. Other bugs are the resting of the daily data, at the moment the daily data that consists of today’s distance and calories when the day changes this rests but at the end of the month when the date goes from say for example 31st to the 1st the program would not reset the daily data as the date has wrapped and is not incremented to higher number so it doesn’t reset the daily’s value, this can be fixed by not just comparing the day but also the mouth and year variable if any of these increment then the reset should take place.  The time management for the development of this project was schedule using a Gantt chart, which clearly showed the tasks that were needed to be done and how long to do it in. The Gantt chart also will let developers state how completed a task has been done. Alternatively, a Trello board could have been used to supplement the Gantt chart in monitoring progress in the development of the application. The reason for using a Trello board is the ease of use and makes certain that each member of the development team is aware of their tasks and responsibilities.  Testing:  It is believed that the testing done on this programme could have been a bit more extensive but the testing that did take place did suggest that there were a few problems with the code that could been improved upon, but the main functionality of the programme works that meets the standards of the aims and objectives of this project. The parts of the programme that could do with more extensive testing would include would be the speeds of which the programme moves between each page as they could be running inefficiently which could affect the users experience while using this application. To improve upon the testing of the project, the quality control could have tested on an actual mobile phone to makes the application works on the device that it is intended for. Furthermore, the quality control team could have also used device with an Apple operating system, as the main benefit of using the Xamarin framework is it’s cross-platform capabilities. Due to unfortunate circumstances, the quality control team does not own a device with the Apple operating system on it, and as such cannot complete a thorough testing on the application. Another avenue of testing that could have been completed is beta testing. The quality control team could have employed an outside source that could thoroughly test the application on their own device to get a better sample of undetected bugs the programme could have. This form of testing was not done for this is beyond the scope of the project and would be too arduous.  Future progress:  There are a few ideas where this programme can progress further. One idea might be adding a centralised database on a server so the information that the user generates can be stored off their phone, so they don't have to take up too much space on their phone, this would mean research into networking and passing data between the client and a server would need to be done because of a centralised database you could have multiple access points to this information such as a website where you can view your data that you collected through your phone and see it in different formats such as tables, graphs or in more broken-down list information which might be difficult for a phone to display effectively. Other parts of the project that could be improved upon is having alternative designs for the user preferences, such as light and dark modes and alternate colour palettes. Additionally, accessibility options could be considered, such as larger font sizes, audio playback and text to speech. Other option for further tracking could include a heart rate monitor. With this improvement the development team could add functionality to record other activities such as cross fit, racket sports, or even static sports like rowing. |

# Appendix:

## Code:

https://github.com/owain-mahoney/Fitness-App

## Bibliography:

Gallup, I., 2020. One In Five U.S. Adults Use Health Apps, Wearable Trackers. [online] Gallup.com. Available at: <https://news.gallup.com/poll/269096/one-five-adults-health-apps-wearable-trackers.aspx#:~:text=More%20than%20one%20in%20four,10%25%20are%20actively%20using%20both.> [Accessed 25 November 2020].

Mlsdev.com. 2020. Create A Fitness App L Guide: Tech Stack, Cost, Features. [online] Available at: <https://mlsdev.com/blog/create-a-fitness-app> [Accessed 25 November 2020].

Mahoney, O., 2020. A report on the various development languages and environments in Mobile Development. p.3.

Sqlite.org. 2021. About SQLite. [online] Available at: <https://sqlite.org/about.html> [Accessed 5 February 2021].

Wintellect. 2021. Model-View-ViewModel (MVVM) Explained. [online] Available at: <https://www.wintellect.com/model-view-viewmodel-mvvm-explained/> [Accessed 12 February 2021].

Gabbiadini, Alessandro & Greitemeyer, Tobias. (2018). Fitness mobile apps positively affect attitudes, perceived behavioral control and physical activities. The Journal of sports medicine and physical fitness. 59. 10.23736/S0022-4707.18.08260-9.

Anurina, O., 2021. Create a Fitness App l Guide: Tech Stack, Cost, Features. [online] Mlsdev.com. Available at: <https://mlsdev.com/blog/create-a-fitness-app> [Accessed 24 March 2021].

Guru99.com. 2021. MVC Tutorial for Beginners: What is, Architecture & Example. [online] Available at: <https://www.guru99.com/mvc-tutorial.html#:~:text=The%20Model%2DView%2DController%20(,Model%2C%20View%2C%20and%20Controller.&text=MVC%20separates%20the%20business%20logic,graphical%20user%20interfaces%20(GUIs).> [Accessed 24 March 2021].

Brainvire.com. 2021. Six Benefits Of Using MVC Model For Effective Web Application Development. [online] Available at: <https://www.brainvire.com/six-benefits-of-using-mvc-model-for-effective-web-application-development/> [Accessed 24 March 2021].

Jithin, 2016. What is MVC? Advantages and Disadvantages of MVC - Interserver Tips. [online] Interserver Tips. Available at: <https://www.interserver.net/tips/kb/mvc-advantages-disadvantages-mvc/> [Accessed 24 March 2021].

Concannon, C., 2020. The negative effects fitness apps can have on your health. [online] EVOKE.ie. Available at: <https://evoke.ie/2020/01/18/health/the-negative-effects-fitness-apps-can-have-on-your-health> [Accessed 24 March 2021].

Johnson, J. and Britch, D., 2020. What is Xamarin? - Xamarin. [online] Docs.microsoft.com. Available at: <https://docs.microsoft.com/en-gb/xamarin/get-started/what-is-xamarin> [Accessed 4 April 2021].

Loria, K., 2017. How to calculate the number of calories you burn doing anything, from running to sex. [online] Business Insider. Available at: <https://www.businessinsider.com/how-to-calculate-calories-burned-exercise-met-value-2017-8?r=US&IR=T#:~:text=Here's%20your%20equation%3A%20MET%20value,divide%20that%20number%20by%20four.> [Accessed 21 April 2021].

Roland, J., 2019. What Are METs, and How Are They Calculated? [online] Healthline. Available at: <https://www.healthline.com/health/what-are-mets#definition> [Accessed 21 April 2021].

ProCon.org, 2019. MET Values for 800+ Activities - Golf - ProCon.org. [online] Golf. Available at: <https://golf.procon.org/met-values-for-800-activities/> [Accessed 21 April 2021].

Hindy, J., 2020. 10 best fitness tracker apps for Android. [online] Android Authority. Available at: <https://www.androidauthority.com/best-fitness-tracker-apps-android-913959/> [Accessed 21 April 2021].

Jon, 2017. Getting Started with Async / Await | Xamarin Blog. [online] Xamarin Blog. Available at: <https://devblogs.microsoft.com/xamarin/getting-started-with-async-await/> [Accessed 26 April 2021].

En.wikipedia.org. 2021. Microsoft Visual Studio - Wikipedia. [online] Available at: <https://en.wikipedia.org/wiki/Microsoft\_Visual\_Studio> [Accessed 28 April 2021].

Www1.cs.columbia.edu. 2002. Namespace: System.Diagnostics. [online] Available at: <http://www1.cs.columbia.edu/~lok/csharp/refdocs/System.Diagnostics/System.Diagnostics.html#:~:text=The%20System.,event%20sources%20on%20the%20network.> [Accessed 29 April 2021].

Copsey, R., MacFarland, C., Copsey, R., Kava, A. and Auger, D., 2017. *What are the problems of the MVVM pattern?*. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/883895/what-are-the-problems-of-the-mvvm-pattern> [Accessed 30 April 2021].

Roland, J., 2019. *What Are METs, and How Are They Calculated?*. [online] Healthline. Available at: <https://www.healthline.com/health/what-are-mets#calculation> [Accessed 30 April 2021].

# Appendix II:

**SECTION A: Project Definition**

**FOR UNDERGRADUATE & TAUGHT POSTGRADUATE ONLY**

**Complete the following table with full and relevant information relating to your research.**

|  |  |
| --- | --- |
| Student Name | Owain Mahoney |
| Student Number | 17049202 |
| Student E-mail Address (please use University e-mail) | [17049202@students.southwales.ac.uk](mailto:17049202@students.southwales.ac.uk) |
| Name of Principal Project Supervisor | Alun King |
| Project Title | Fitness tracking |
| Briefly describe the project, being sure to identify any aspects that are relevant to the Ethical Evaluation in Section B.  NOTE: A project determined to be High Risk will need to include additional information in Section B to fully-specify the risks and mitigations. | Track a person’s movement and record it during exercise that the users specify i.e., running or walking etc. To track these movements, we will be using a smart phone internal GPS and dynameter and using equation to translate that into human metrics like speed and distance. |
| Please add an explanation of your study in plain English, with particular focus on any parts of your study which involve human participants. No more than 100 words. This is to help the Faculty Research Ethics Committee (FREC) to understand the project. | This application will track people through GPS plotting a travelled route they took during exercise. Using math then work out their speed. Will also record this for later revision. |

**SECTION B: Ethical Evaluation**

**FOR UNDERGRADUATE & TAUGHT POSTGRADUATE ONLY**

Consider the following points to determine the level of ethical risk your research presents:

1. Involves those who are considered vulnerable such as:

* Children under 16.
* Adults with learning difficulties.

Unless in an accredited setting, accompanied by a carer or professional with a duty of care.

1. Involves those who are considered highly vulnerable such as:

* Adults or children with diagnosed mental illness/terminal illness/dementia/in a residential care home.
* Adults or children in emergency situations.
* Adults or children with limited capacity to consent

1. Involves those who are “dependent” on others (such as teacher or lecturer to student). Unless in an accredited setting associated with normal working conditions or routines and within normal operating hours, such as a cultural institution, pre-school, school, or youth club where the research is carried out as part of professional practice such as curriculum development.
2. Requires full NHS ethical approval via the Integrated Research Application System.
3. Requires a Human Tissue Act license.
4. Involves “covert” procedures as in covert observation studies.
5. Involves anything considered “sensitive.” For example, does not carry a risk of those involved disclosing information which compromises the research (e.g., illegal activities; activities where moral opinion may differ, potential professional misconduct – work errors).
6. Induces significant psychological stress or anxiety, or produce humiliation or cause more than fleeting harm / negative consequences beyond the risks encountered in the normal life of the participants (and where the potential for fleeting “harm” is clearly detailed in the participant information sheet). If in doubt regarding definition of the above terminology please contact the research governance office.
7. Involves administration of drugs, placebos, or other substances (such as food substances or vitamins) as part of this study.
8. Involves invasive procedures (not limited to blood sampling, collection of biological samples, or passing current through a participant’s body, etc.).
9. Offers any financial inducements to participate in the study.
10. Intends to recruit serving prisoners or serving young offenders via Her Majesty’s Prison & Probation Service.

For your course, there may be specific requirements in **addition** to these, depending on the nature of the subject and how your project is assessed. You must also complete those requirements.

If **none** of the 12 points above apply, then the research can be considered **Low Risk**, *unless your course identifies additional criteria relevant to your subject that would render it High Risk*. This Section is then signed off by yourself and your supervisor, and held on file for review by FREC.

If **any** of the 12 points applies, then the research is considered **High Risk** and students must bring the matter to the attention of their research supervisor immediately. **Research cannot then commence until mitigations for the risk are agreed by FREC**. Seek advice from your Supervisor, who can help you identify mitigations of the risk or redesign as a Low Risk project.

**All students must complete the section below, in collaboration with their supervisor.**

Please strike through the statement that **does not** apply.

1. An ethics review has been completed, and the project has been identified as Low Risk.
2. ~~An ethics review has been completed, and a High Risk was identified. I agree to explain how they may be mitigated below, and agree to abide by any conditions identified at this stage, by my Project Supervisor, the School or the Faculty. I understand that High Risk projects can only proceed with approval from the Faculty Research Ethics Committee.~~

|  |
| --- |
| Issues: (Include as much information as possible to help FREC members to understand the issues. Extend onto additional pages as necessary.)  Issues with this type of application could involve GDPR as information like personal information and location with time stamps will be stored in this application. |
| Proposed mitigations: (Include as much information as possible to help FREC members to understand the mitigations. Extend onto additional pages as necessary.)  The mitigate these problems I will use an opening terms and conditions page to access the application. |
| Student’s Signature:  Date: 07/10/2020 |
| **Supervisor’s statement:** I have ensured due diligence and accountable decision making by the student. I have sought appropriate advice where required to support my judgment in this.  Supervisor’s Signature: Alun King  Date: 07/10/2020 |
| **Any false or mis-represented information contributing to this Ethical Evaluation, including attempting to pass off a High Risk project as a Low Risk project, is subject to the Student Misconduct Regulations and may also have legal repercussions.** |

Both signatures are **required** for all projects, both Low Risk and High Risk.