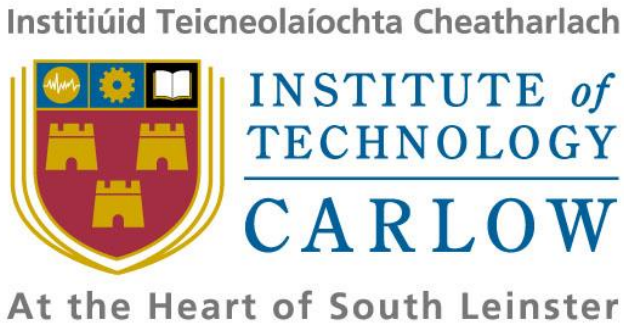

4th Year Final Project Research Manual 2

BSc (Honours) Software Development

Name: Alex Matthews
Student ID: C00208942
Supervisor: Dr. Oisin Cawley

Contents

1. Abstract

This document will outline the research which was undertaken in preparation for developing part of a mobile application which will improve the quality of life for people suffering from coeliac disease. This part of the application will help people to track their food intake to help compliance with their diet. The research will cover already existing applications which provide similar functionalities to the ones being developed for this part of the project. It will also cover technologies which could possibly be used to develop the application.

2. Introduction

Coeliac disease is an illness that causes people to have an intolerance to gluten. In Europe, 1%-2% of people suffer from coeliac disease and the number of people being diagnosed with coeliac disease is increasing. One of the biggest problems among patients is compliance with their diet. At present there is no education system to educate teenagers about the long-term risks of gluten ingestion. It has recently become possible to measure gluten ingestion using a simple test. This test will form the basis for a new application that will hopefully be beneficial to coeliac patients, mainly teenagers.

The project is on the Erasmus+ program and it is a joint effort between students, doctors and lecturers from Ireland, Amsterdam, Spain and Austria. The goal of the project is to build a mobile application for people with coeliac disease which will help them in their everyday life. It will help them in multiple ways including monitoring gluten intake, tracking food, educating about the disease, storing test results and tracking symptoms. This document and final year project will focus on the food tracker part of the application. The food tracker will allow people with coeliac disease to track their food intake in an effort to increase their compliance rate.

This document will first discuss the research which was carried out on other applications that provide the same functionality as is being developed. It will discuss the good and the bad points about each of the mobile applications which were researched and explore why some are more popular than others. All research of similar mobile applications was carried out by downloading the application onto an Android device and using the it until all functionality was explored.

The research which was carried out on possible technologies that could be used to build this section of the application will also be outlined. The research that was carried out on possible technologies for the server-side code and for the database will also be discussed. As the nature of the project requires all technologies to be agreed upon by all involved, a standard set of technologies has already been suggested by the project team for front-end development. This also means that the recommended technologies in this document may not be right for the project as a whole.

3. Need Finding - Amsterdam

As part of the project, participants travelled to Amsterdam for five days to meet with the rest of the team. The main purpose of this trip was to better understand the problem and drill down into what functionality might be included in the final application. At the beginning of the week the main goal was to understand the disease a bit better and to try and work out the main problem areas that a coeliac patient might have in their life. For the duration of the week, the team was split into 3 smaller teams. The teams tried to get into the mind set of a coeliac patient by thinking about how the disease affects their life. Once the teams had a better understanding of the problems a patient faces in their everyday life, they started brainstorming specific functionalities which target those problems. The teams then ruled out ideas which didn't target the problems and were eventually left with the functionalities that were thought to be most beneficial.

The next goal was to build some working prototypes of the proposed functionality. This was done on paper and on wireframe programs that build clickable user interfaces. Once each team had a working prototype that could be tested, they each carried out field tests on the campus grounds of the University. With this feedback in mind, the teams revised their prototypes. At the end of the week, each team presented their prototype to the lecturers and received feedback. This concluded the week. Some of the artefacts from the week can be viewed below in *Figure 1, 2, 3.*

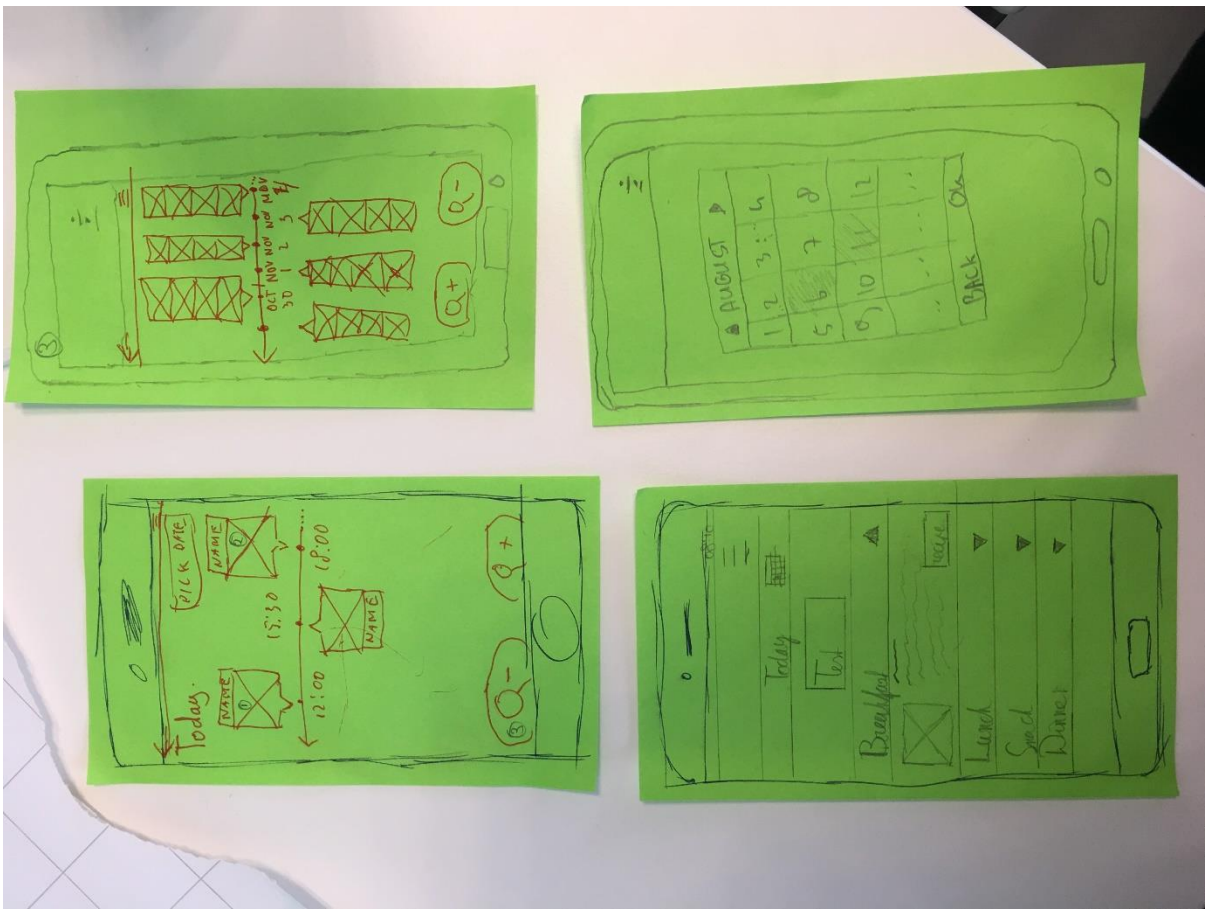

Figure 1 – Amsterdam Artefact

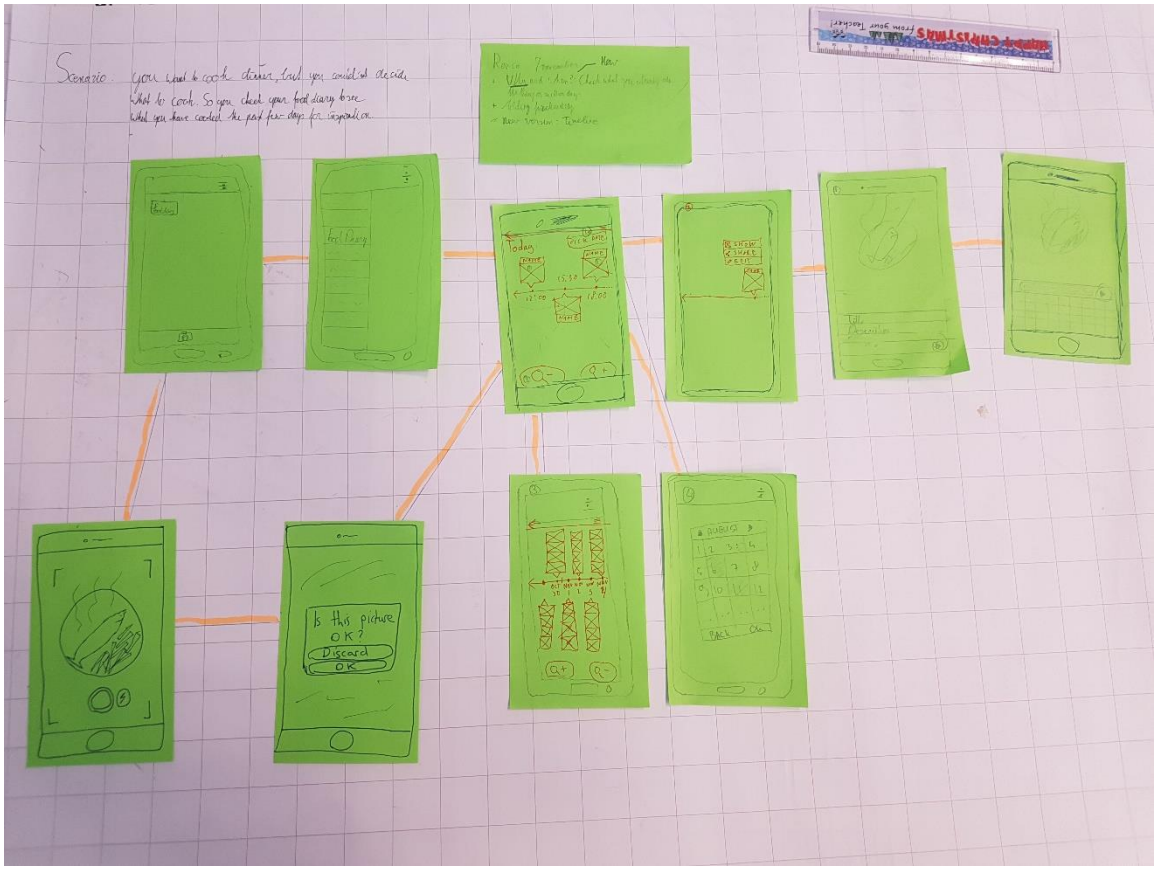

Figure 2 – Amsterdam Artefact

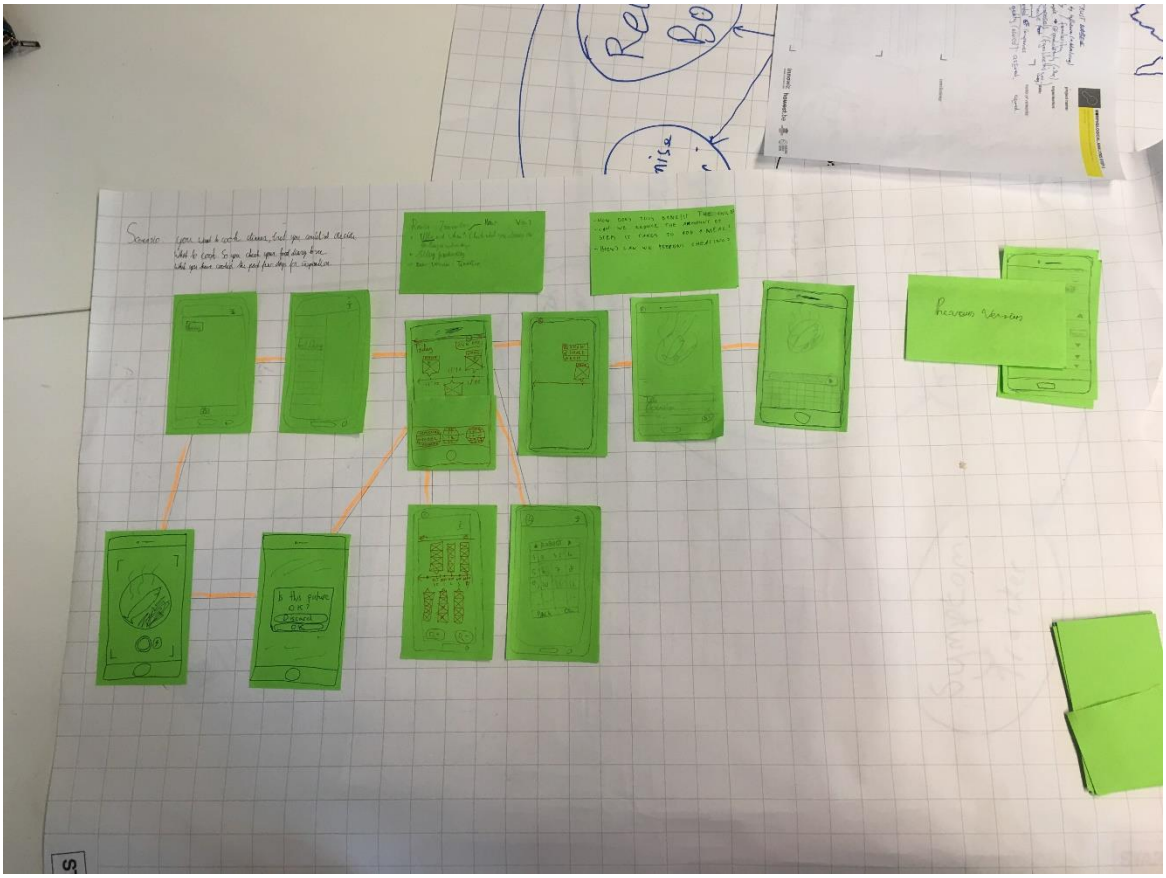

Figure 3 – Amsterdam Artefact

3. Existing Applications

3.1. Android Application – Food Diary

"Food Diary" by My Daily Bits LLC is an application which aims to help users lose weight, maintain weight and gain weight by allowing them to track what they eat, exercise, weight and allows them to save recipes. The application has a simple, professional looking user interface which is laid out well. There doesn't seem to be many advertisements on the free version of the application but there is some. There are two options for the premium version of the application. Users can choose to subscribe and pay €7.99 every three months, or they can choose to pay €27.99 to permanently unlock all of the premium features. Premium features include no advertisements, a much larger food database, the ability to email yourself a copy of the diary, and more configuration options, including the ability to enter up to eight separate meals a day. The application has over one million downloads and is rated 4.1 stars out of five. It is also on top of the list for a search for 'food diary' on the Play Store [1]. The home screen of the application and the information screens can be seen in *Figure 4*.

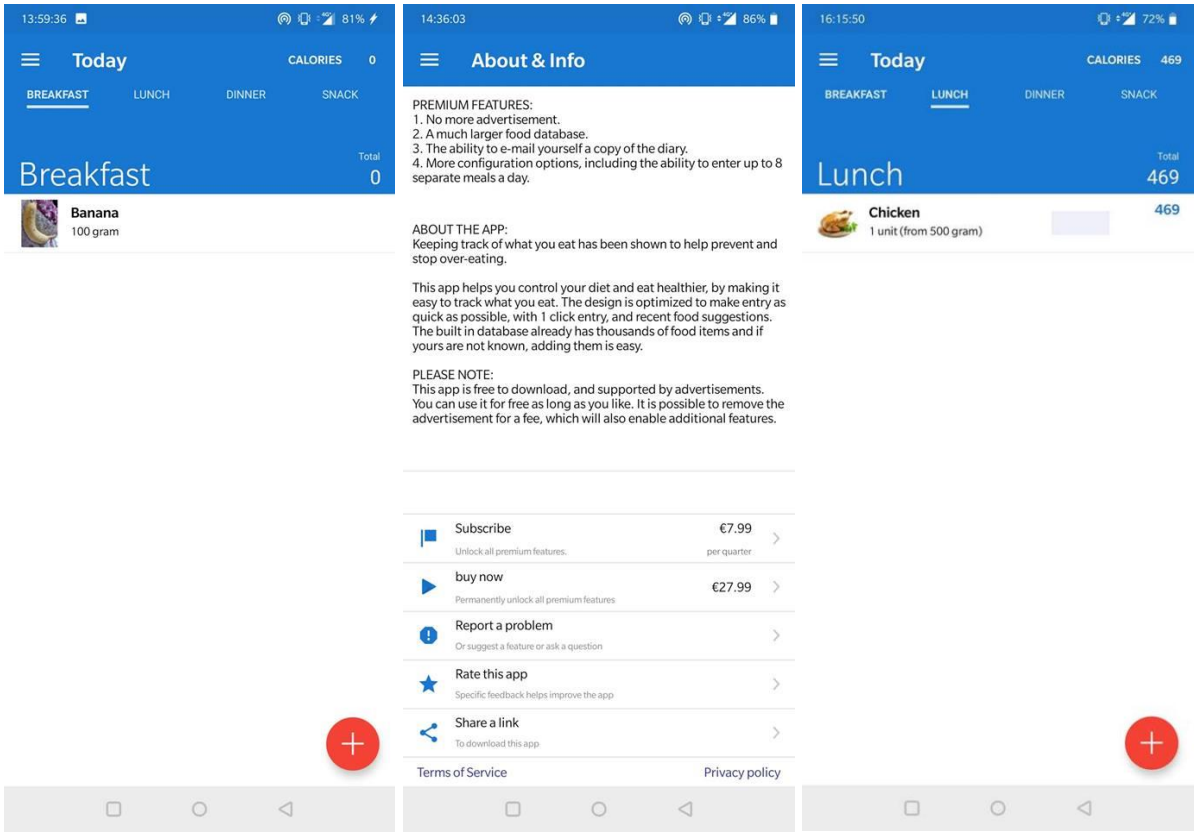

Figure 4 – "Food Diary" Home and About screens

The home screen of the application has a simple and user-friendly layout and it shows the user's food diary entry for the current day. It also shows the number of calories consumed on the day and has a button to open the main menu. The main menu has a total of nine options which are as follows; Today, Exercise, Weight, Journal, Report, Recipes, Goals, Settings, and About & Info. The main menu can be seen below in *Figure 5.* The home screen,

which is the “Today” option on the main menu, consists of four tabs; breakfast, lunch, dinner and snack. Users of the application can select a tab based on which meal they want to add food to. They can then press the plus button and begin to add a food item to their diary for that meal.

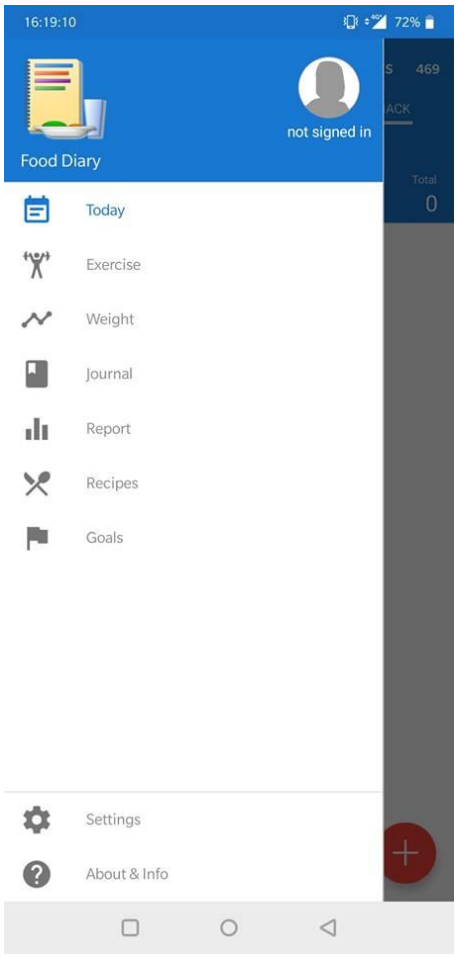

Figure 5– Main Menu

This screen allows the user to search for the food item from a large database. It also allows them to add a note to the entry or choose one of their saved recipes. They are also shown a long list of common items for that meal. They can also add a new food item if it is not present in the database. If they pick an item which is in the database they will be brought to a screen where they can specify the amount they had, what form it was in and the nutritional value. Since the chosen item was in the database, there are estimated values for the nutritional information which can be changed. A picture of the item is also visible. Users can also add a picture of the nutritional information of the item. The user can press the tick in the top left of the screen to save their changes. If the user chooses to add a new item they will be brought to the same screen as before, but they will have to enter all the information themselves and they can also add a picture to the item. These screens can be seen in *Figure 6.*

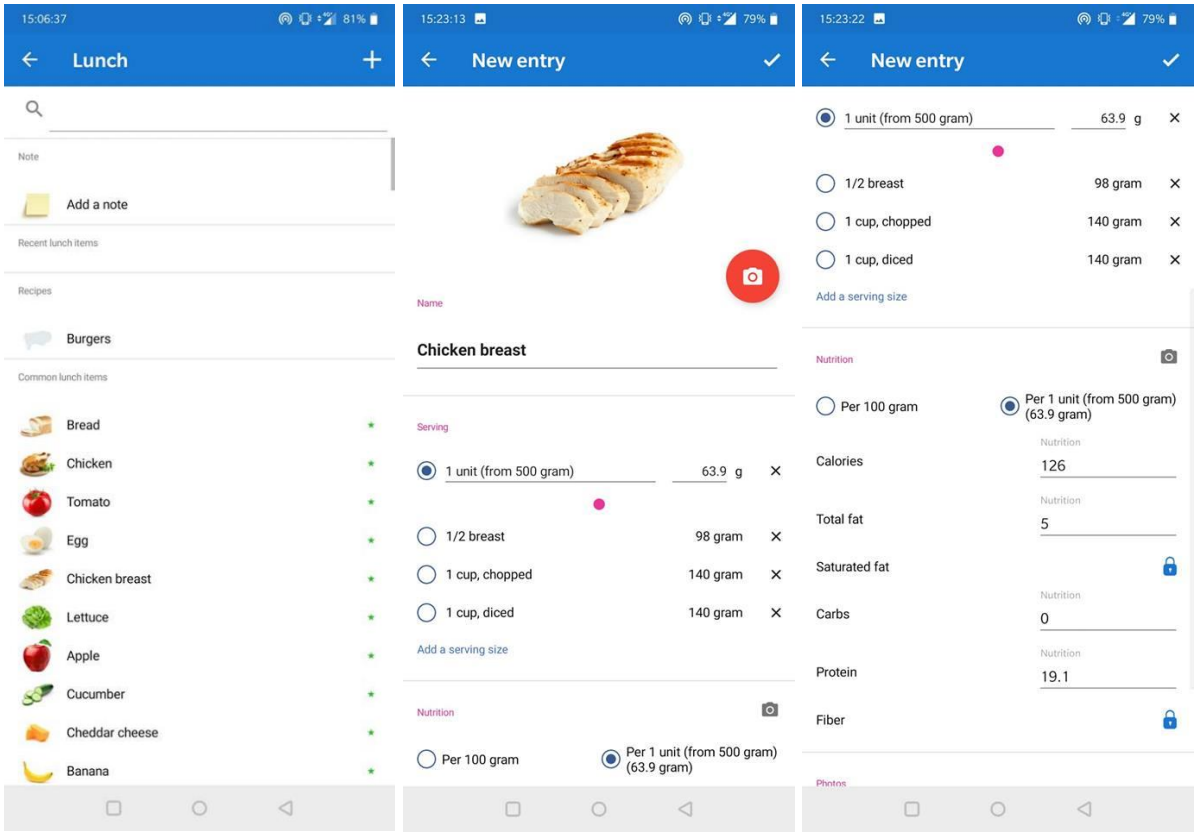

Figure 6 – Add food to meal

The second main menu item is Exercise. When this option is selected the user is brought to a screen which is similar to the home screen. The primary function of this screen is to allow the users to record exercise they have done on a specific date. There is an advertisement on this screen which detracts from the appearance of the application When the user presses the plus button to add an exercise they are brought to another screen which shows a list of common exercises to choose from. The user can either choose one of these options or they can add their own exercise. Once the user chooses one of the options they are taken to a screen where they can specify the date they did the exercise, the duration, calories burnt, speed, and they can add a note. This is a useful feature for this application but would not fit the application being developed for the coeliac project. The Exercise screens can be seen in *Figure 7.*

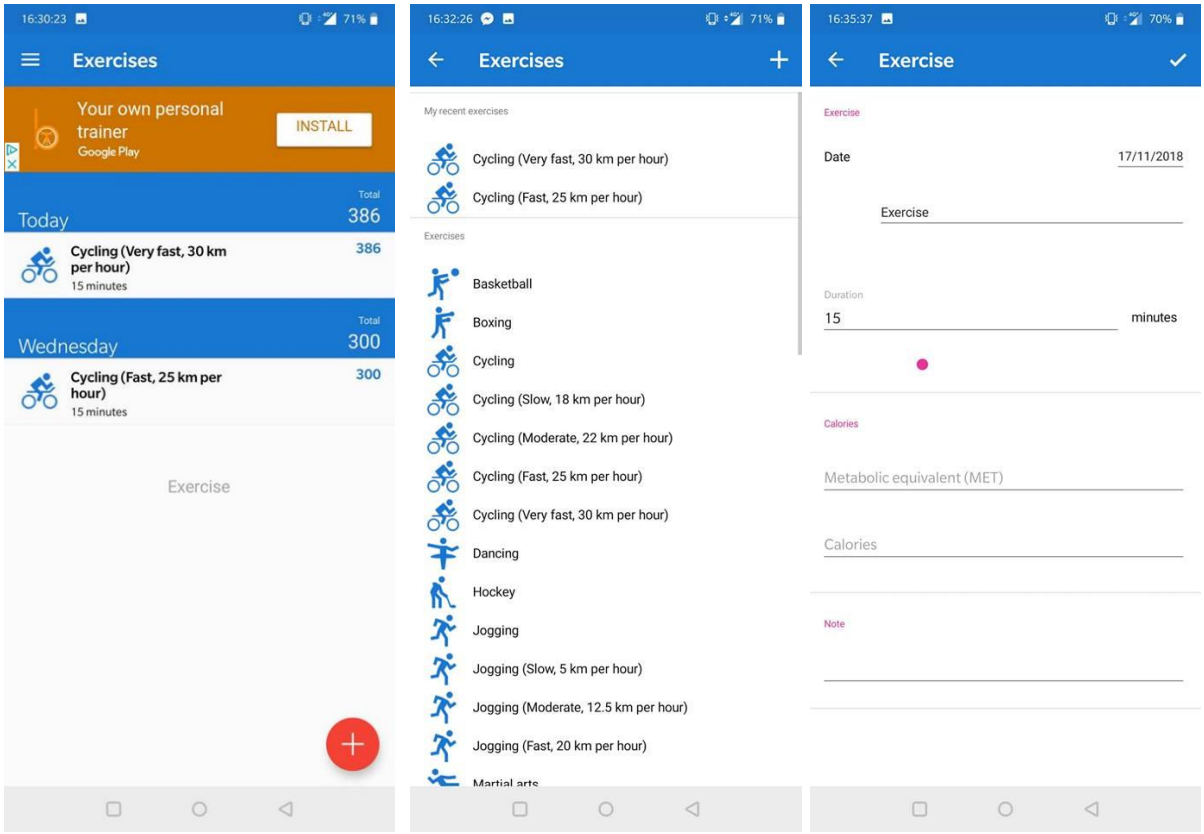

Figure 7 – Exercise Functionality

The next menu item on the list is the Weight section. This part of the application aims to help users track the progression of their weight. The screen shows a line graph with time on the x-axis and weight on the y-axis. The graph shows the change in a person's weight over time. The can add data points to the graph by pressing the plus button. They are presented with a popup which allows them to enter a weight and a date. Once they save the data, it is plotted onto the graph. These screens can be seen in *Figure 8.*

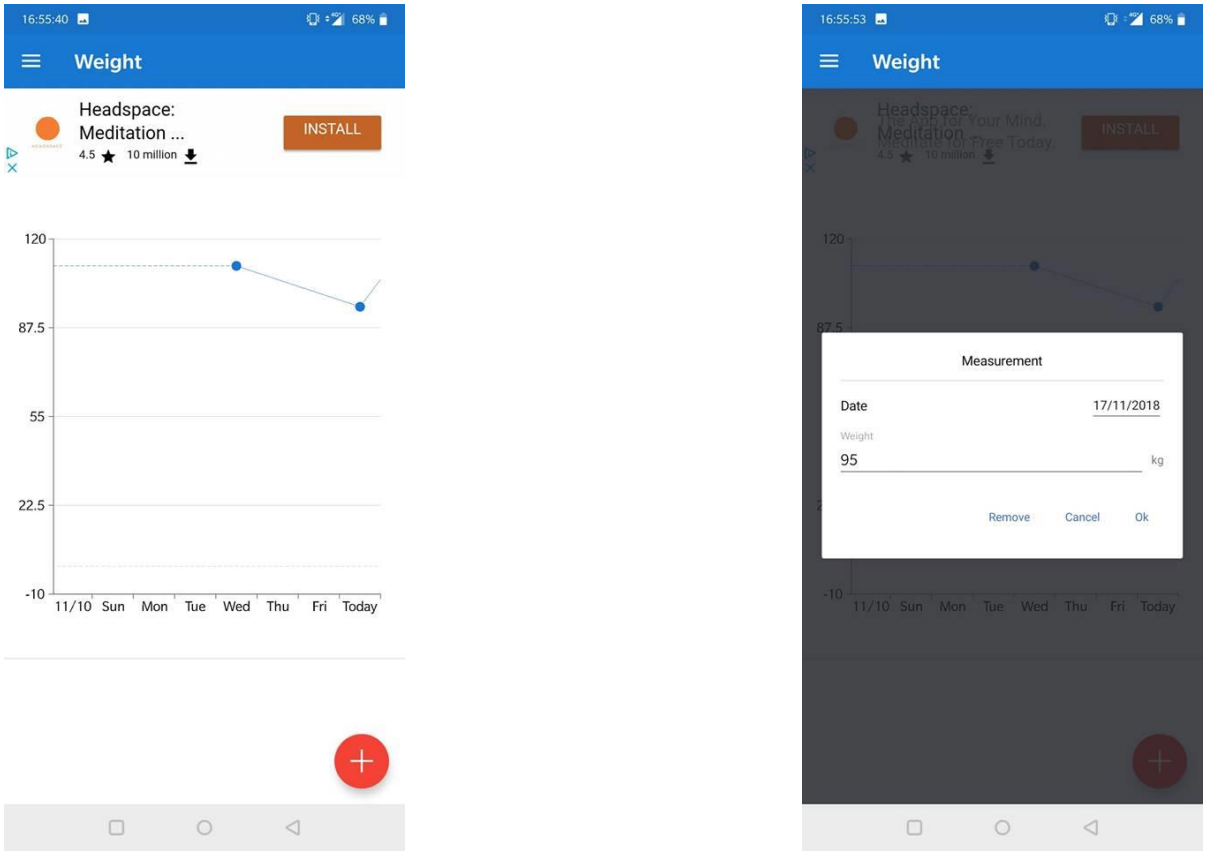

Figure 8 – Weight Functionality

The next menu item on the list is the actual Journal. This is where all of the diary entries are saved and displayed. At the top of this page there are columns with different diet measurements. Some of the measurements are calories, fat, carbs and protein. Depending on which one is chosen, the diary entry will show a different value depending on what the user entered when making the entry. Below the columns is the list of diary entries. The list is continuously loaded as the user scrolls down. Each diary entry shows the food that was eaten and the exercise that was carried out. Each entry also has an edit button and send button. The edit button allows the user to add meals or exercise to a specific day. The send button allows the user to send a specific diary entry by email. There is also an option to send the whole journal by email. These screens can be seen in *Figure 9.*

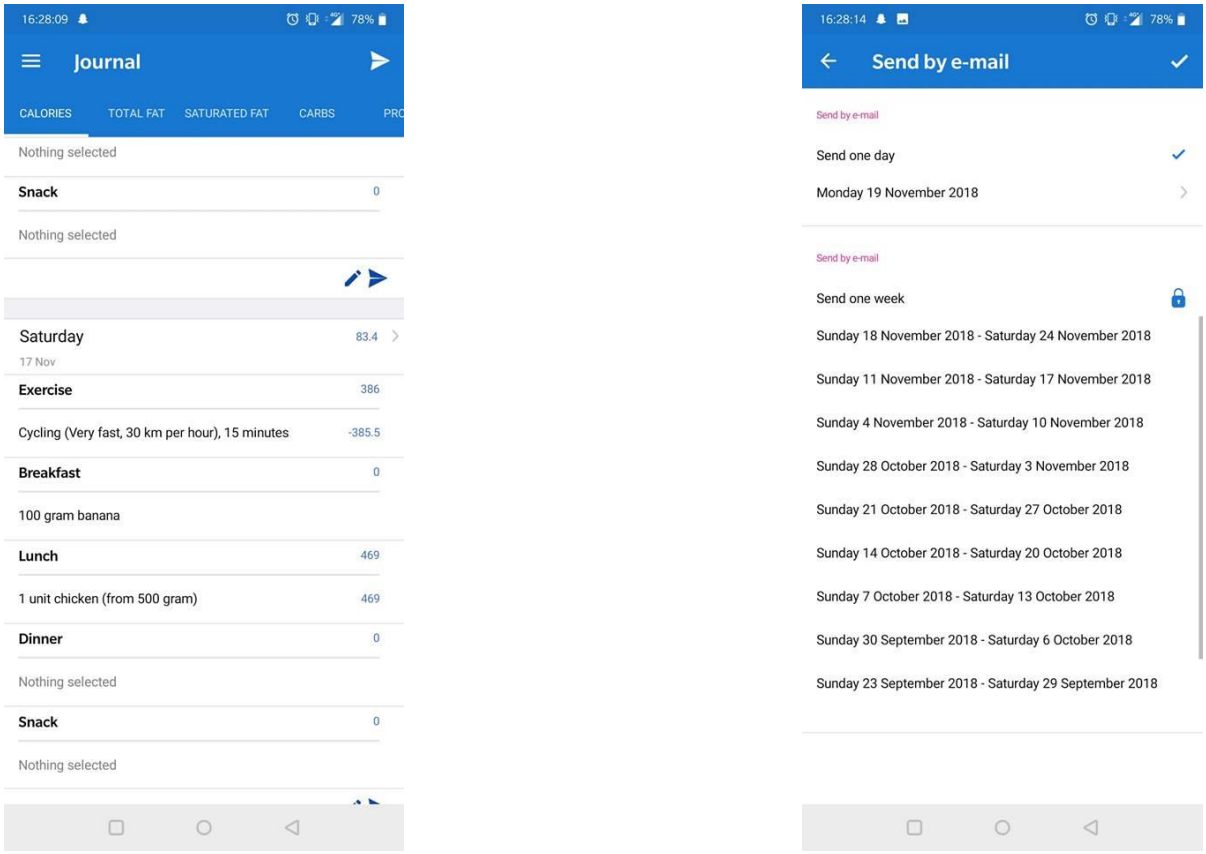

Figure 9 – Journal Functionality

The next item on the main menu is the Report functionality. This screen provides users with graphs which indicate their intake of protein, fat, calories, and carbohydrates for each meal. The first graph on the screen shows calorie intake, the second shows fat, the third shows carbohydrates, and the fourth shows protein. The graphs have different colour bars for each meal. Each graph also shows an average intake for the week. The graphs can be seen in *Figure 10.*

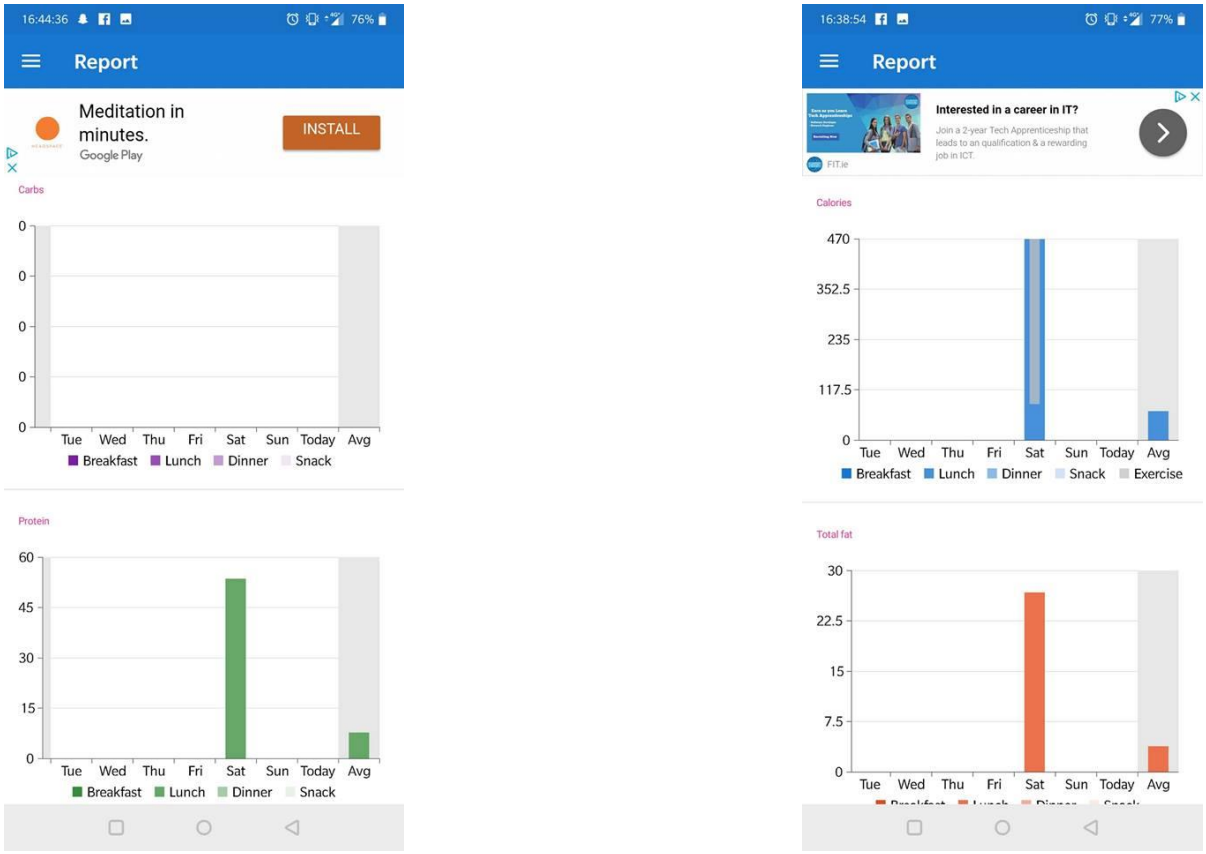

Figure 10 – Report Functionality

The second last main menu item is the Recipes section. This section is very simple. It has a list of recipes which were added by the user and there's an add button to allow the user to add another recipe. When this button is pressed, the user is brought to the "Add a recipe" page. This page first asks the user for the name of the recipe. It also allows the user to add a photo of the food. The page has a button for adding ingredients to the recipe. When this button is pressed, a list of ingredients is shown. The user can choose from the list or add their own recipe, much like adding food to the diary. They can then specify how many meals the recipe makes and description of how to cook it. See *Figure 11* for screens.

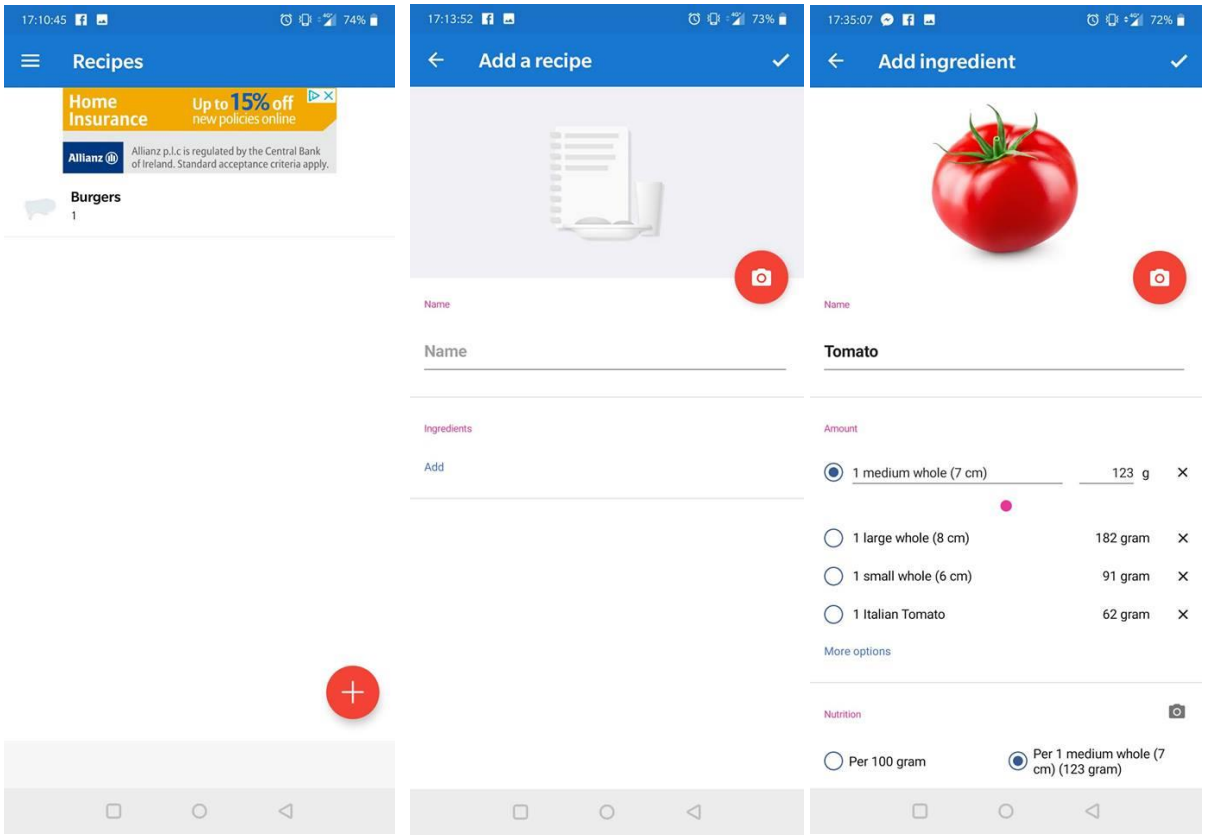

Figure 11 – Recipe Functionality

The last piece of functionality on the main menu is the Goals section. This part of the application allows users to set various daily goals. At the top of the page, a user can set a personal goal. Users can also set target amounts for calories, fat, saturated fat, carbohydrates, protein and fibre. Users can add new goals which they can pick from a list of different vitamins, minerals etc. The list consists of many different options but most of them are locked behind premium membership. Some of these screens can be seen in *Figure 12.*

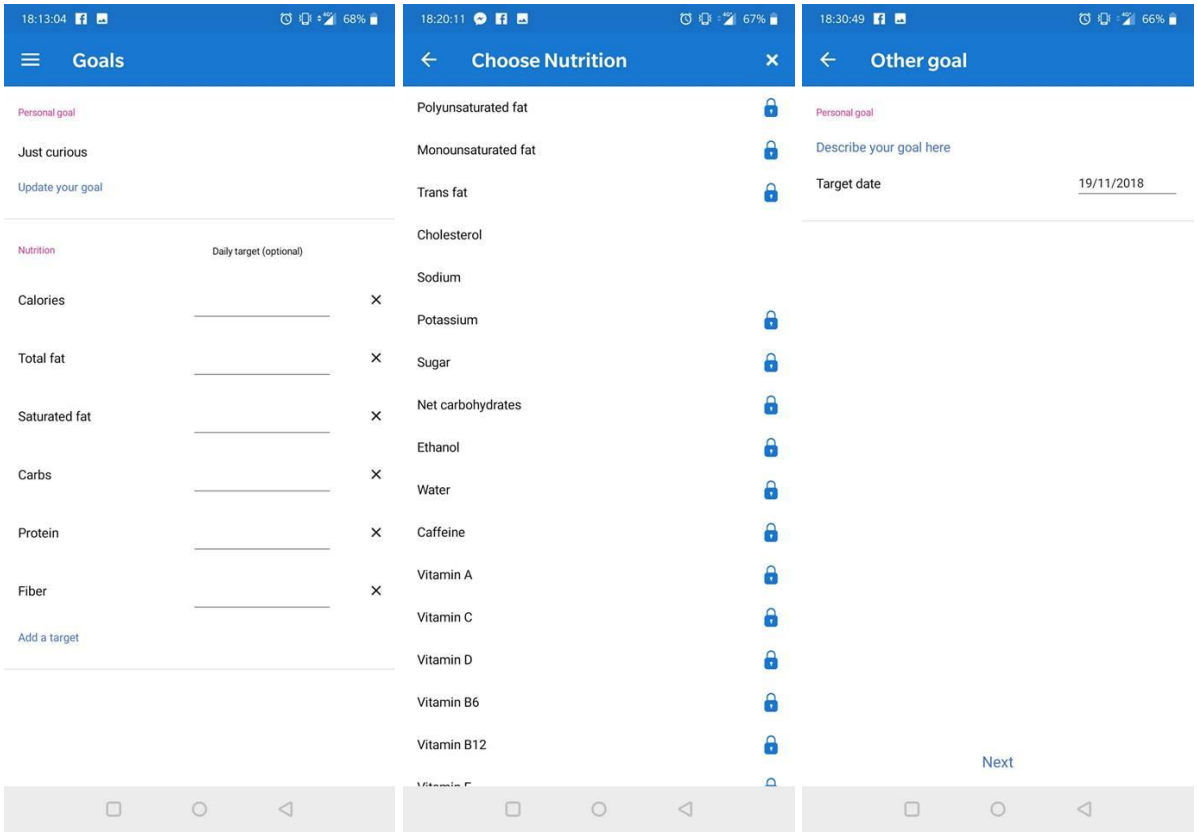

Figure 12 – Goals Functionality

The application also provides a settings menu. It provides options to sign in to the application, add a picture to your profile, choose which meals you want on your diary, choose which nutrition you want for your goals, choose which information is shown throughout the application, choose the units of measurement, and add reminders to remind the user to add an entry to their food diary. The settings menu can be seen in *Figure 13.*

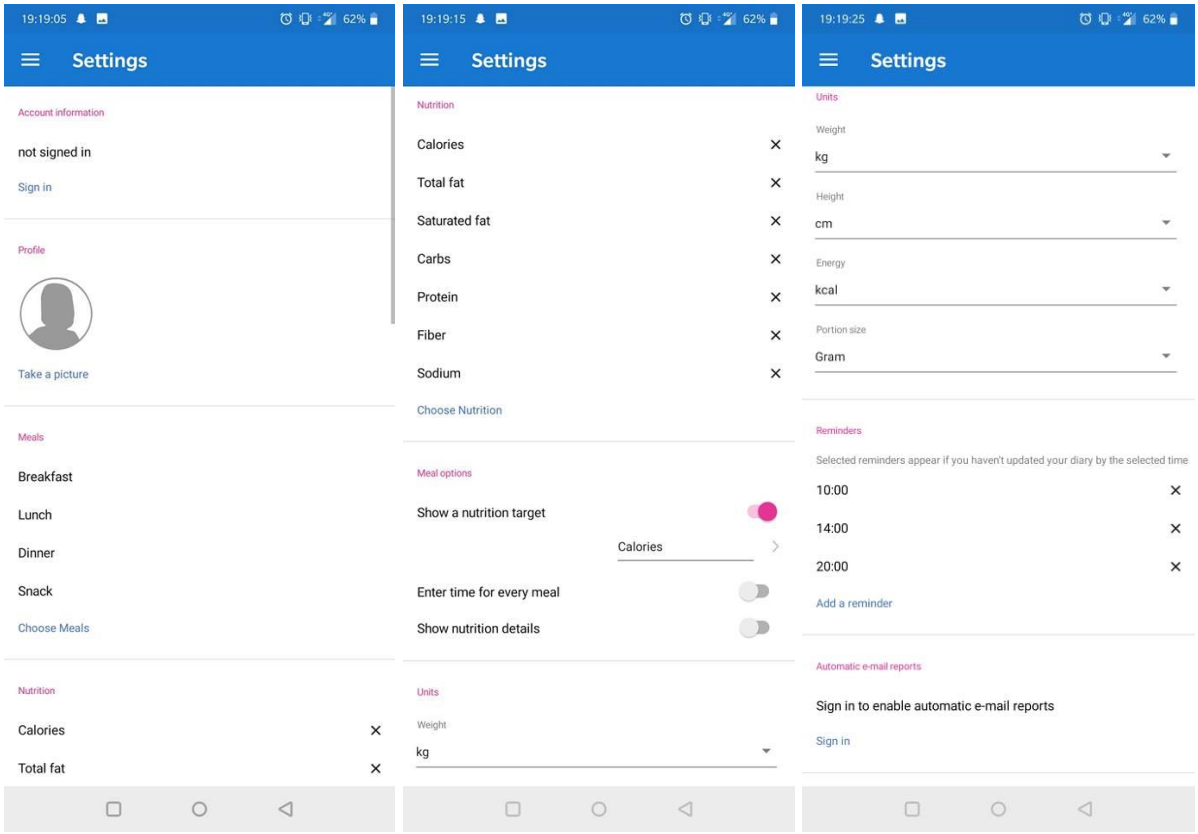

Figure 13 – Settings

Conclusion

Overall, this application provides some useful functionality which could be applied in the application being developed. The goals feature is something that might make younger users use the application in the hope of reaching their goals. This could be paired with some kind of achievements system. The food diary list of each entry will also be considered for the application being developed. There will also be some recipe functionality included in the final application. This could be a useful feature for sharing coeliac friendly recipes, giving patients more choice and options. Some graph functionality will also be implemented in the final application.

4.1. Android Application – Dining Note

"Dining Note" by 4th May Soft is an application which allows users to track their food. It intends to help users start a good eating habit by keeping track of their food intake. The quality of the applications user interface is very bad. It is not very user friendly at all and it is sometime hard to figure out how to do something. The advertisements are also very intrusive. They are positioned near the home buttons, so they are often pressed by mistake, bringing the user to their web browser. There is a premium membership which removes ads and allows the user to change the colour theme. The premium membership only costs €2.89. The application has over 100,000 downloads and has a star rating of 4.5 stars out of 5. It is usually the third result on the Google Play store when searching for food diaries.

When the application is first opened, the name of the application and a question asking "what did you eat today?" is displayed. There is also a plus button that brings the user to the next screen when it's pressed. The next screen is the food diary and it initially lands on today. From here, a user can add entries to each day by selecting the day they want and pressing the plus button in the middle of the screen. This brings up a list where the user can pick which meal they want to add. They can also add drinks and exercise. When the user makes their choice, an editable text box is displayed. The user can type what they ate into the box. They can also specify who they were with and where they were. They can also add a photo from their gallery to the entry. When the user is finished entering all the relevant information, they can press the save button. Some of these screens can be seen below in *Figure 14*.

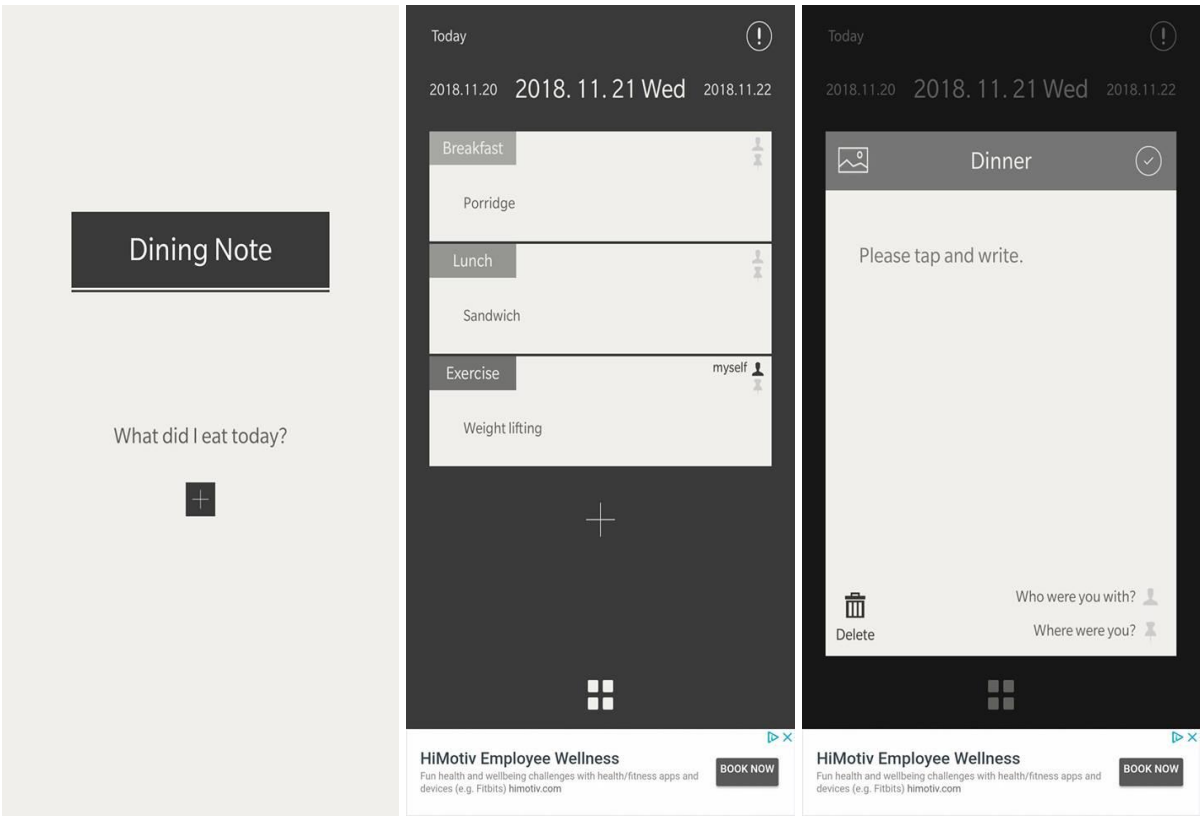

Figure 14 – Dining Note Home Screen & Add entry

Users of the application have a couple of different choices for selecting the day they want to add to. On the page after the initial welcome screen, there are dates in selectable columns. This can be seen in *Figure 14.* There is a button which looks like four squares near the bottom of the screen that when pressed brings up a calendar. From here, the user can choose which month they want to view. The application doesn't have an option for changing year so the user has to keep pressing back a month until they reach the desired year. They can also see a visual representation of the meals that were filled in on each day. This is shown with small squares on each day. The more meals that were added to the day, the more squares that will be displayed. This is useful feature for seeing any meals that were missed. There is a stats button just above the calendar. The stats screen shows how many times each meal, drinks, or exercise has been added in that month. These screens can be seen below in *Figure 15.*

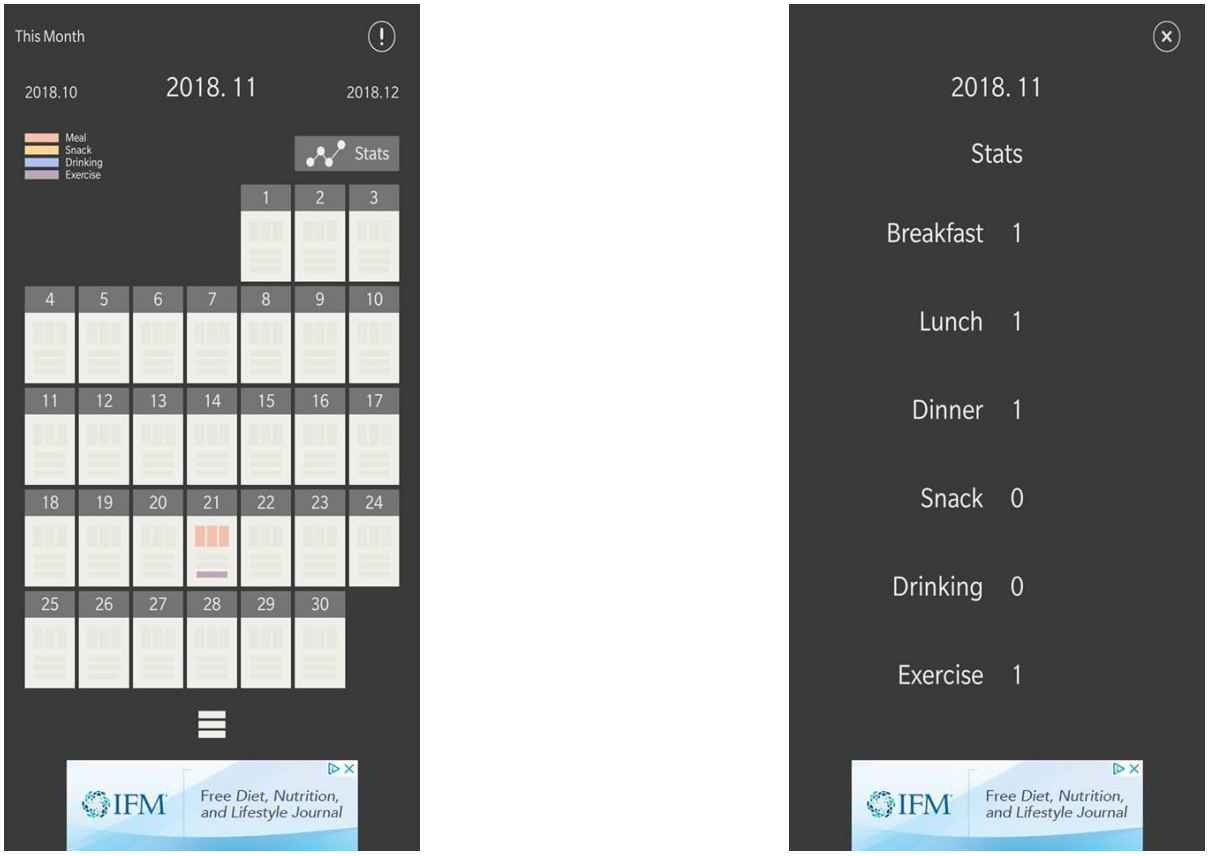

Figure 15 – Calendar & Statistics Functionality

The settings menu has a few options available to the user. The first option on the list is "Follow us on Facebook". When this option is selected, a browser is opened, and the companies Facebook page is displayed. The second option is "Send us Feedback". This option brings up the default mail application and starts a new email with the companies email address as the recipient. The user can type their feedback in the email and send it to the company. The next option is "Rate an App". When pressed, the Google Play store is opened on the applications page. The next option is "Share an App". This allows the user to share the application page on various platforms. There is also an option to set a password.

This allows the user to set a password for when the application is first opened. The last few options are “Change Language”, “Change Colour Theme”, and “Upgrade to Premium”. The settings screen can be seen in *Figure 16.*

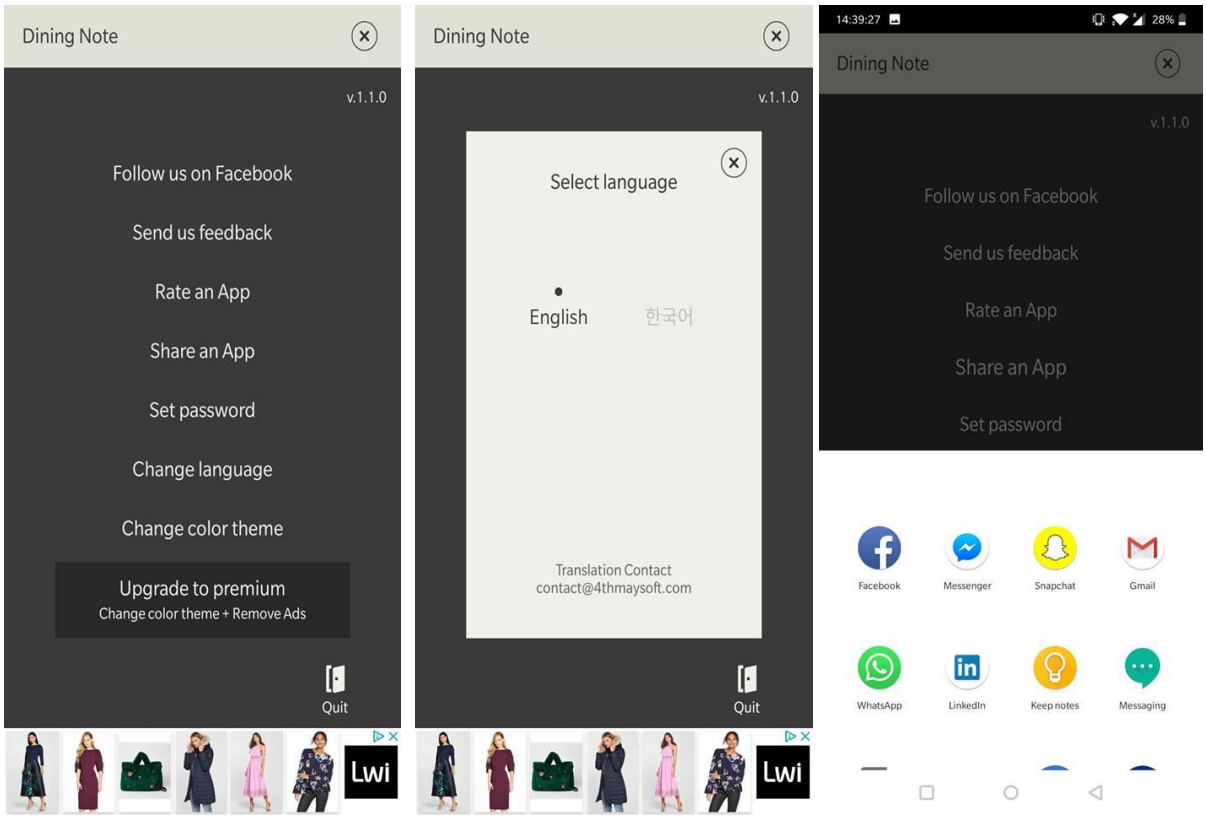

Figure 16 – Settings

Conclusion

This application is very simplistic and doesn’t offer a lot of functionality. There are some good ideas, but the implementation is poor. The calendar with the visualisation of which meals were added on which days is a useful feature for quickly seeing progress throughout the month. The basic functionality of the application being developed will be very similar to this but will be of a much higher level of quality in terms of appearance.

6. Mobile Development Technologies

There are many possibilities when it comes to technology for developing mobile applications. The most obvious choice would be to develop the application natively. That would mean using Java/Kotlin for Android development and Swift/Objective-C for iOS development. The problem with native development is that developers will essentially have to create two applications; one for Android and one for iOS. This is not cost or time effective and the workload will be too high for the team. This could lead to sub-optimal application.

The alternative to native development is to develop a hybrid application. A hybrid application runs on both iOS and Android using the same code base. This cuts the workload for developing a cross-platform mobile application by nearly half. As a result, the cost of developing the application is substantially reduced. There are also downsides to hybrid development such as worse than native performance and a UI which is not at the same standard as a natively developed application. As native development is ruled out for this project, this section will discuss possible hybrid development technologies.

6.1. Apache Cordova

6.1.1. Overview

Apache Cordova is an open-source mobile development framework. It allows the creation of hybrid mobile applications using standard web technologies: HTML5, CSS3, and JavaScript. This means creating applications for Android and iOS using the same code base. Applications execute within wrappers targeted to either Android or iOS, and rely on standards-compliant API bindings to access each device's capabilities such as data, sensors, network status etc. Cordova allows developers to use a special web browser called a *WebView.* A WebView allows access to device-level APIs which allow for the creation of full-featured applications without using any native code. The basic architecture of a Cordova application is shown in *Figure 17 [2].*

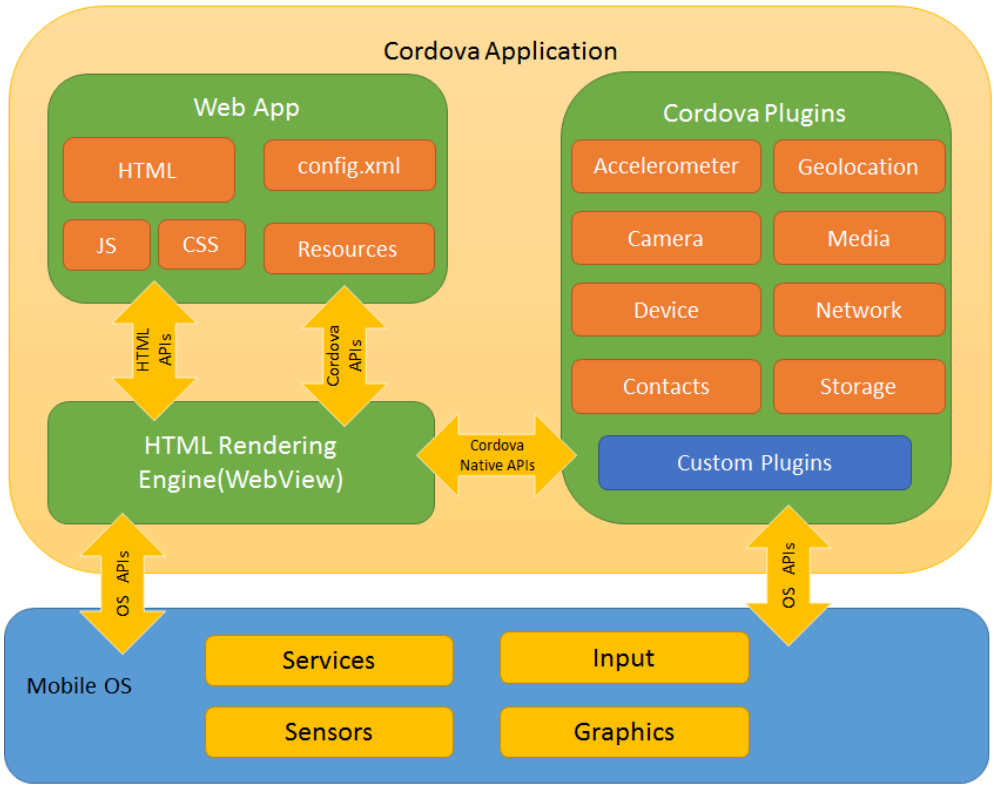

Figure 17 [2]

Cordova allows developers to work in one of two ways by taking different development paths. The first development path is called cross-platform workflow. This workflow is used if the developer wants to their app to run on as many mobile operating systems as possible, with little need for platform-specific code. It centres around the Cordova CLI which is a high-level tool that allows you to build projects for many platforms at once, abstracting away much of the functionality of lower-level shell scripts. This is the recommended configuration. The other development path is called platform-centred workflow. This workflow is used if the developer wants to build an application for a specific platform and needs to be able to modify it at a low level. This workflow needs to be used when a combination of custom native components and web-based components is needed for the application [2].

6.1.2. Advantages

Cordova is a good match for the project as it will allow concurrent development of the Android and iOS applications. This will save a lot of time and make the project more manageable for all involved. It is free and open-source which is a big plus for a project of this nature. The fact that it uses standard web development technologies also makes it desirable due to those technologies being well known by most programmers and easy to learn. Cordova is also a good choice when it comes to security. The community continuously updates its security guidelines pertaining to whitelist (access to external domains), Iframes (to show ad content from third-party sites in a secure manner), the call-back ID mechanism (retrieve information about callers), certificate pinning (check the certificate of third-party websites or servers), self-signed certificates, and encrypted storage. All of this helps keep the users and their data safe from malicious attackers [3] [4].

6.1.3. Disadvantages

One of the biggest drawbacks with Cordova is the speed of the application. The application won't be able to match the performance of a native one due to the limited resources available to Cordova's WebView. This may not be an issue for this application if it I not resource intensive. There is also a risk for high regression testing which is due to changes made for one platform introducing bugs for another platform. This is because of the use of a shared code base. Another downside is that it takes 1.5 – 2 times longer to develop a Cordova application for a single platform than it does to develop a single native application [2] [3].

Another drawback to using Cordova is that you need an Apple Mac PC to develop an application for iOS and Cordova. This is because Apple tools required to build OS X applications run only on the OSX operating system on Intel-based Macs [3]. Also, on iOS, Cordova doesn't support background processing. The WebView execution is paused when applications are in the background. This makes them unsuitable for some applications, such as an application which are integrated with beacon technology. This is due to no background processing on iOS which is necessary for Apples iBeacon technology. On Android (pre-4.4), there is fragmented WebView performance due to the limited support for newer CSS features. While Cordova takes up only 1.40% of US top-500 applications, its installs are only 0.49% which indicates that the results of the developers' attempts are not satisfying users [3] [4]. Mubaloo gives the following advice to developers who are thinking of using Cordova for their application:

"In theory, Cordova apps offer companies particular benefits, which offer an appealing way to develop across platforms. However, once you look deeper, there are factors that need to be taken into consideration to ensure that stakeholders have a thorough understanding of the compromises that are characteristic of hybrid solutions. If these don't impact what you need the app to achieve or the value for your end user, then a cross platform could be the solution for you. If in fact the compromises begin to devalue your mobile app, then a native development approach would be more in line with your overall business strategy." [4]

6.2. Microsoft Xamarin

6.2.1. Overview

Xamarin is Microsoft-owned, California based software company. The company developed Xamarin tools enable the development of cross-platform mobile applications using a shared C# code base. Xamarin can be used to build iOS, Android and Windows Phone applications with native interfaces. According to Xamarin, over 1.4 million developers were using Xamarin's products in 120 countries around the world as of April 2017 meaning that there is wealth of resources available to developers online which would come in very useful during development. Xamarin is free and open-source and is included in Visual Studio to be used by individuals, open source projects and smaller teams. A licensing fee is only required to be paid for large, private projects where the enterprise and professional editions are required [5].

Xamarin provides various technologies for developing applications. These are Xamarin.Forms, Xamarin. Android and Xamarin.iOS. Xamarin.Android and Xamarin.iOS are referred to as Xamarin native. Xamarin native allows native development of Android and iOS applications,

giving them a fully native appearance and performance, which matches an application written in Java for Android or Swift/Objective-C for iOS. It's better to use Xamarin native when the file size needs to be small, when the application has animation or a complex UI, or when the applications needs to use a native feature which is not available on all platforms [5] [6]. Xamarin native is not suitable for the application due to the added time and effort required to develop for both Android and iOS. The distinction between Xamarin native and forms can be seen in *Figure 18* [7].

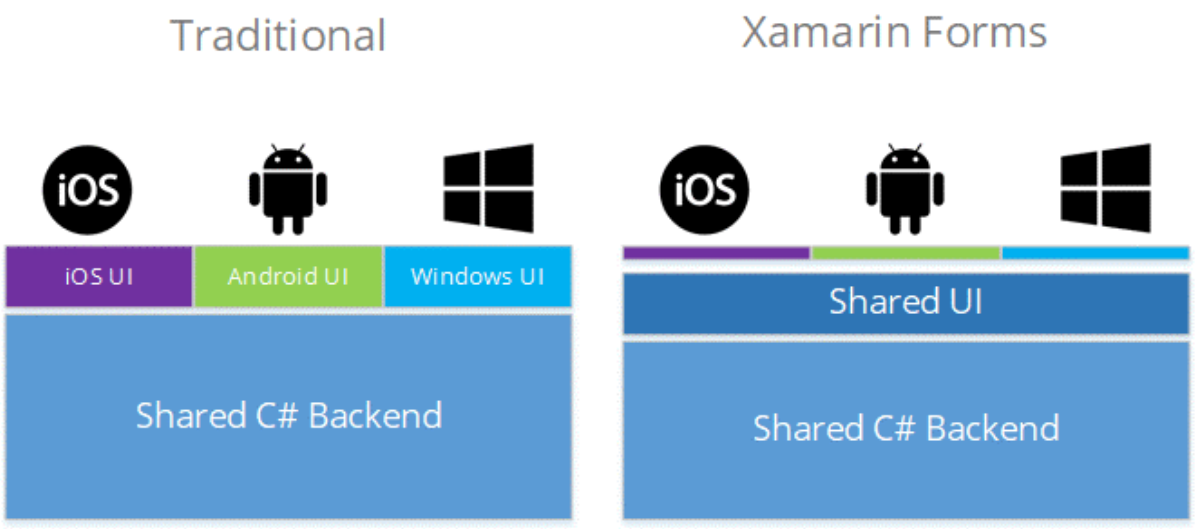

Figure 18 [7]

Xamarin.Forms is an abstraction layer that allows you to push common user interface code to multiple platforms, where if you build in Xamarin native, the UI development work is all done on the native platforms [6]. Xamarin.Forms allows the majority of the UI code to be shared across platforms making it easy to maintain and saves a lot of time due to not having to do the same work twice for different platforms. There is a common misconception that it's not possible to design good looking applications using Xamarin.Forms which is simply untrue. An example of an application which was developed using Xamarin.Forms can be seen in *Figure 19* [9]. For the purposes of this project, and to save on time and effort, only Xamarin.Forms will be considered [7] [8].

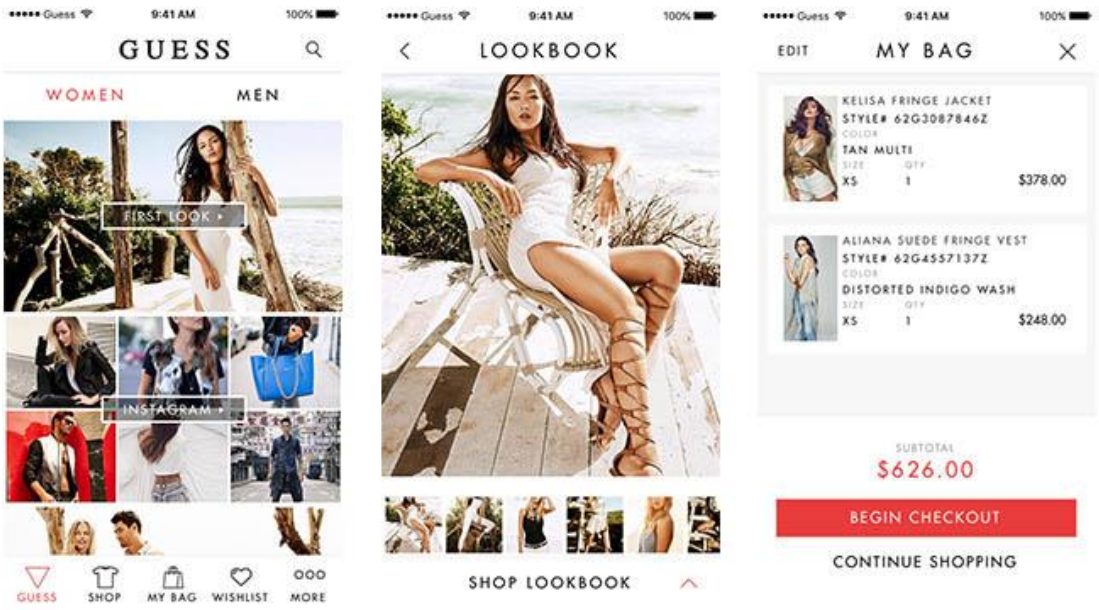

Figure 19 [9]

6.2.2. Advantages – Xamarin.Forms

One of the reasons Xamarin might be a good match for the project is that it is free for a project of this size meaning no extra cost for the project. Like Cordova, it enables concurrent development of iOS and Android applications. When compared to Cordova, Xamarin is an all-around more powerful and comprehensive platform, with significantly better tooling and debugging experiences. A big advantage of Xamarin is that the application will have a beautiful native look and feel which improves usability and makes users more likely to keep using the it. With Xamarin, the developer gets full hardware support, eliminating hardware compatibility issues and having to use plugins and specific API's to work with common devices functionality across platforms [10].

6.2.3. Disadvantages – Xamarin.Forms

One of the main disadvantages of using Xamarin is that there is a learning curve to become familiar with the technology. This is due to the developers having to become familiar with the Xamarin.Forms and its architecture, and C# if they have no prior experience in the language. They also need to learn XAML which is used to create the UI. With Xamarin, you have to use only the components provided by the platform and some .Net open source resources. This is in contrast to native applications where a lot of open-source software can be used to help develop the application. Application size is also an issue when it comes to Xamarin. A simple 'Hello World' application could take as much as 15MB. This is due to the abundance of associated libraries included in file [10].

6.3. React Native

6.3.1. Overview

React Native is an open-source JavaScript library which allows developers to build native mobile application using JavaScript. It was developed by Facebook and it's maintained by Facebook and a community of individual developers and companies. It uses the same design

as React, allowing developers to compose rich mobile user interfaces from declarative components. According to the React website, React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It lets you compose complex UIs from small and isolated pieces of code called “components”. React native applications are not “hybrid” or “HTML 5” apps. They are real mobile applications which are indistinguishable from an application written in Objective-C/Swift or Java/Kotlin. React Native uses the same fundamental user interface building blocks as natively developed applications, but the building blocks are put together using JavaScript and React. According to the website, there are thousands of applications using React Native. Some of these applications include Facebook, Instagram, Skype, Pinterest, and Uber [11]. Some sample React Native code can be seen below in *Figure 20*.

```
1.  import React, { Component } from 'react';
2.  import { Text, View } from 'react-native';
3.
4.  class WhyReactNativeIsSoGreat extends Component {
5.    render() {
6.      return (
7.        <View>
8.          <Text>
9.            If you like React on the web, you'll like React Native.
10.         </Text>
11.         <Text>
12.           You just use native components like 'View' and 'Text',
13.           instead of web components like 'div' and 'span'.
14.         </Text>
15.       </View>
16.     );
17.   }
18. }
```

Figure 20 – React Native Sample Code

React Native allows developers to build applications faster than if they were being developed with native languages. Instead of recompiling, developers can reload the application instantly. This is done with a feature called Hot Reloading. Hot Reloading allows developers to run new code while retaining the application state. This means that the application does not have to be re-compiled every time the developer wants to view new changes. React Native also allows developers to use native code if they need to. Components written in Objective-C/Swift or Java/Kotlin can be easily combined with React Native. This is useful for optimising certain aspects of the application that need to have better performance than just React Native can provide. This allows developers to build part of the application in React native and part of the application using native code. This is how the Facebook application works [11].

6.3.2. Advantages

A big advantage to using React Native for the project is that it is community driven. This means that there is a huge number of React developers to share their knowledge and expertise with the community. It also means there are huge amounts freely available components available online. Another big advantage is the amount of code reuse. The same code can be used on iOS and android. This saves a lot of time since the developer doesn’t

have to write the same piece of code twice, in two different languages. React Native allows about 90% of the code to be reused between iOS and Android.

The aforementioned Hot Reloading feature is also a huge advantage. It enables developers to immediately see their latest changes that they have made in the code. If a developer has two windows opened, one containing the code and the other showing a mobile screen as a result of the code, they can immediately see the effect of what they have changed in one screen, on the other screen. React Native is optimised well and it provides strong performance for mobile environments. It makes use of the GPU (Graphics Processing Unit), while native applications are more CPU (Central Processing Unit) heavy. Compared to hybrid applications, React Native is very fast. React Native also allows an extremely modular design which lends itself very well to this project [12]. This is due to the project having multiple individual developers working on separate components that have to be integrated into a full product.

6.3.3. Disadvantages

One disadvantage is that developers will still need to write some native code. This could be a big learning curve for developers, especially if they have no experience building native mobile applications. The speed of the application will be slower than a native one, but this is usually not a problem unless the application is heavy on resources. Another disadvantage is the security of it since it is a JavaScript based library and JavaScript is famous for its fragility. This won't be a major factor in this application as it is doesn't store data that is as sensitive as a banking or finance application [13].

Another big disadvantage is that Facebook is the key owner of the project. This means Facebook put in a lot of resources to ensure constant development of the platform. It is highly unlikely that Facebook will kill the framework overnight, but it is possible. Developers still operate on Facebooks' license which means developers are always dependent on them. Facebook has the right to revoke the license to use React if the developers get into a patent-related dispute with them [14].

6.4. Conclusion

While all of the technologies discussed would be suitable for the project, they still have their advantages and disadvantages. When compared to the other two, it's clear that using React Native for the project would produce a better looking, feeling and performing application. This is due to React Native enabling near native appearance and performance. There is also a case for Cordova due to the learning curve being less steep when compared to learning Xamarin or React Native. This is due to the developers being more familiar with standard web technologies than they are with C# and the Xamarin architecture. React Native could also have a steep learning curve due to developers having to dive deep into JavaScript when they previously only used it for form validation. React Native is also a good choice because it allows components to be very modular. This allows multiple developers to work on different feature and they are able to easily integrate each of the components. The choice of technology to build the application is a topic that needs further discussion with the team.

7. Bibliography

[1] Google. (2012). Play Store. Available: https://play.google.com/store?hl=en. Last accessed 27th Oct 2018.

[2] Apache. (2010). Cordova Documentation. Available: https://cordova.apache.org/docs/en/latest/. Last accessed 20th Oct 2018.

[3] Root Info Solutions. (2016). Top 5 Benefits of Apache Cordova/PhoneGap App Development. Available: https://www.rootinfosol.com/top-5-benefits-of-apache-cordova-phonegap-app-development. Last accessed 20th Oct 2018.

[4] Mubaloo. (2015). Cordova vs Native apps. Available: https://mubaloo.com/cordova-vs-native-apps/. Last accessed 22th Oct 2018.

[5] Wikipedia. (2011). Xamarin. Available: https://en.wikipedia.org/wiki/Xamarin. Last accessed 24th Oct 2018.

[6] Nathan Williams. (2017). Xamarin.Forms or Xamarin Native: Which should you use for your cross-platform app project?. Available: https://arctouch.com/blog/xamarin-forms-xamarin-native/. Last accessed 24th Oct 2018.

[7] Adam Pedley. (2017). Is Xamarin.Forms Making Traditional Xamarin Obsolete?. Available: https://xamarinhelp.com/xamarin-forms-making-traditional-xamarin-obsolete/. Last accessed 24th Oct 2018.

[8] Applikey Team. (2018). Xamarin Forms vs Xamarin Native: What Fits You Best?. Available: https://applikeysolutions.com/blog/xamarin-forms-vs-xamarin-native-what-fits-you-best. Last accessed 24th Oct 2018.

[9] Nathan Williams. (2017). Xamarin.Forms is Much More Capable Than You Think. Available: https://arctouch.com/blog/xamarin-forms-more-capable-than-you-think/. Last accessed 24th Oct 2018.

[10] AltexSoft. (2018). The Good and The Bad of Xamarin Mobile Development. Available: https://www.altexsoft.com/blog/mobile/pros-and-cons-of-xamarin-vs-native/. Last accessed 25th Oct 2018.

[11] Facebook. (2018). *React Native Build native mobile apps using JavaScript and React.* Available: https://facebook.github.io/react-native/. Last accessed 20th November 2018.

[12] Unknown. (2017). *5 key advantages of React Native.* Available: https://www.icapps.com/blog/5-key-advantages-react-native. Last accessed 20th November 2018.

[13] Nyma Malik. (2018). *React Native: Pros and Cons.* Available: https://citrusbits.com/react-native-pros-and-cons/. Last accessed 20th November 2018.

[14] Natalia Chrzanowska. (2017). *React Native - Pros and Cons Of Facebook's Framework.* Available: https://www.netguru.co/blog/react-native-pros-and-cons. Last accessed 20th November 2018.