# 2. Literature Review

## 2.1. Introduction

In this chapter, the research into the literature of mobile technologies was documented. The literature reviewed for NDMA came from the realm of nutrition and dietary needs. Different technologies and tools were reviewed for their designed purpose for the requirement of NDMA. Two of the key areas investigated were Usability methods and evaluation tools. Both were reviewed for their application in both mobile performance and usability. Mobile performance and usability are crucial areas of NDMA’s development. This is due to the user experience and user interface as the chosen areas of complexity for NDMA. Different solutions from industry were investigated. This was completed to obtain key knowledge about previous attempts to solve the user requirements. Three areas of concern during the review were: different types of technology in the domain area, other useful strategies or research relating to the application’s developers aim. After this investigation was explored, two decisions were made in relation to the researched areas. The first decision was to provide a critical analysis of the technologies and tools used. The second was to use the analysis to improve the requirements of NDMA. These decision and steps were documented too

## 2.2. Research Topic 1

The first research was an investigation was conducted into the 10 usability Nielsen’s Heuristics. The 10 usability heuristics was **1-“developed by Jakob Nielsen together with Rolf Molich in the early 90's”**. The reason behind the terminology of heuristics **is 2)-“because they are broad rules of thumb and not specific usability guidelines”.**  These metrics are evaluated as one of the biggest requirements is user experience for NDMA. **3)-“User Experience is one of the hottest topics in day today designer’s life. To make our product success and stand out from our competitors, we should ensure that we are treating our users in the best way”**. As such, “the heuristic review is one of the useful methods to figure out the usability issues”. Each of the protocols are describe below in relation to NDMA user interface and user experience.

**Visibility of System Status**

In NDMA interaction with the user, it should inform the status of the application in an ongoing basis. This would keep the user engage and understanding of the processes going on during engagement. For example, if the user accesses the logging system, it should respond with the speed expected. If there is a reason why it cannot, it should inform the user to be patient.

**Match between System & the Real World**

The language and actions conveyed by NDMA should match what the person expects of NDMA. For example, if the person visits the advisor system, they should get advice from the system. If they visit the logging system by going through the advisor system, the user would more likely be confused.

**User Control & Freedom**

NDMA should provide the user full control and freedom over the logged data. This means the type of food they log, how it is logged, and the contents logged. The user should also be able to edit the logged details too.

**Consistency & Standards**

Across the application of NDMA, there should be a consistent layout and look for NDMA. This includes the button looks, overall presentation of NDMA and consist in the language used. Standards for NDMA is to professional look and functionality. If this is not the case, majority of the users would probably not come back to use the application.

**Error Prevention**

NDMA should prevent errors that would mess up the application’s technology. This is by enforcing credentials in certain areas of the application. An example would be logging into NDMA using a unique username, which is stored by NDMA.

**Recognition rather than Recall**

NDMA should provide tools inside the application that is used by other mobile and/or nutritional applications. This would allow the user to use NDMA at a quicker rate. It also ensures the user doe not delve out of the comfort zone when using NDMA. This would encourage the user to return and keep using NDMA.

**Flexibility & Efficiency of Use**

NDMA, through the logging and advisor system, should provide flexibility and efficiency of use. This can be completed by ensuring both the logging system and advisor system has various options. This would provide flexibility in the usage of NDMA. A way to provide efficiency is to provide options that allow the user to complete the tasks in the minimum number of steps.

**Aesthetic & Minimalist Design**

The overall design of NDMA should be visually aesthetic to the eye. This should be in terms of the overall design. The application should look professional and polished. The colour scheme should blend well together in the application. NDMA should also be designed using minimalist approach. This would encompass NDMA overall look to provide the functionality that encapsulate its requirements. Not to provide more functionality that clusters the design of NDMA.

**Help users recognise, diagnose, and recover from errors**

If any errors occur doing the usage of NDMA, the application should be able to handle it. The handling process would be able to allow the user to editing the changes in the error made. In the areas where the users cannot make changes to, NDMA should be able to assist the user to return the application to a stable condition.

**Help & Documentation.**

NDMA should provide details and access to assistance for the usage of the application. In areas where it cannot provide the assistance, it should provide documentation for usage and guidelines. This could be through providing links and direction to such assistance.

## 2.3. Research Topic 2

The second topic researched was alternative mobile application technologies to NDMA. This investigation would be conducted by analysing the applications functionality. This was then compared with the developer team’s ambitions behind the design approach. The evaluation of the chosen technologies was split into two different areas. The first was the criteria they would be evaluated by. The second was what was getting evaluated on the application. The chosen criteria’s, which the mobile applications were evaluated by, were the 10 Nielsen’s Heuristics and 7 Best Practices to Overcome Mobile App Usability Issues. The chosen areas for evaluation were the following: “User Experience” (UX), “User Interface” (UI), “Design” and “Ease of use”.

The two technologies chosen for review were “Fitbit” and “mySugr”. Both were found on the “Play Store” for the Android devices, thus are mobile applications. The reason for choosing these applications was their logging capability as a main feature. Since NDMA has a logging system, a measure against these industry applications seemed like a reasonable choice. Their UI, UX, Design and ease of use were evaluated too.

**Fitbit**

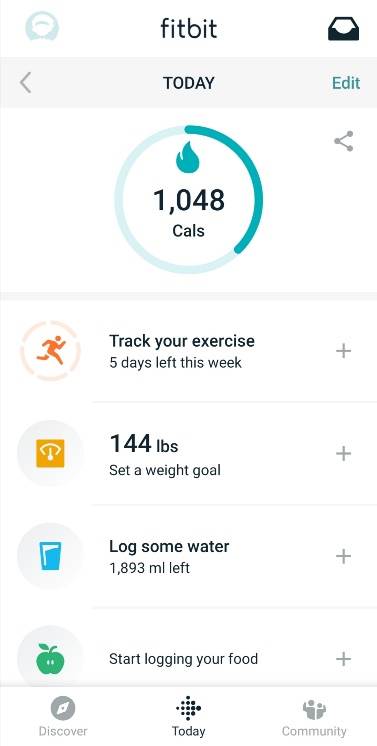
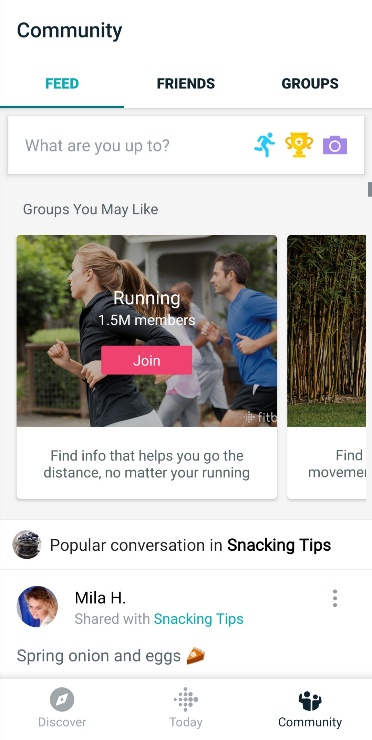
According to Play Store Application specification, Fitbit “is dedicated to helping people lead healthier, more active lives”. Fitbit was designed to assist the user in monitoring their physical condition. This is by logging key information, such as the diet, water intake, exercise tracking and a weight goal. Through the logging process, the user can use Fitbit to ensure their overall health is in the best condition. This would assist the user in their sleeping, eating, exercising and day to day routines. Fitbit can also monitor the heart rate by syncing up with multiple external devices, such as Fitbit watch, which were designed to track the user details.

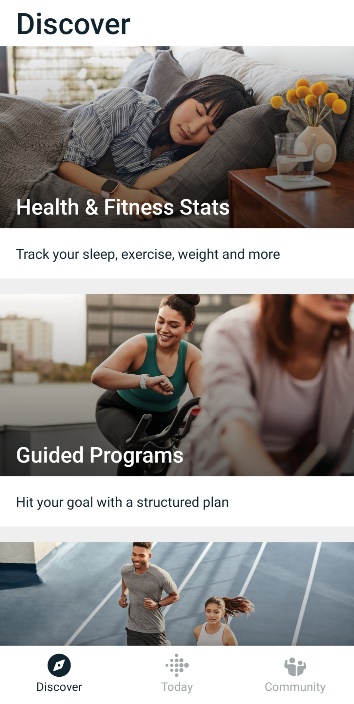
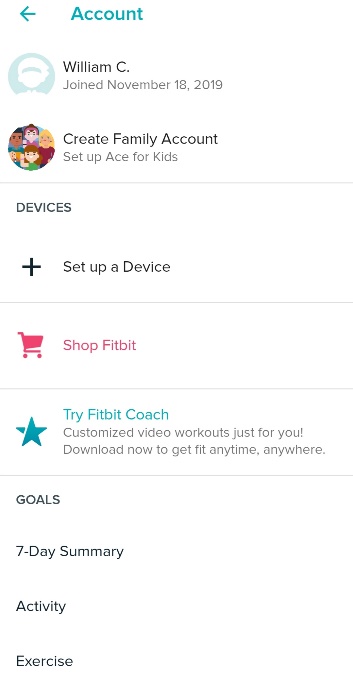
The areas of Fitbit which were evaluated were their UI, UX, Design and ease of use. Fitbit was evaluated by the criteria of the 10 Nielsen’s Heuristics and 7 modern mobile usability factors.

|  |  |  |
| --- | --- | --- |
| **10 Usability Heuristics for User Interface** | **Rating Metrics** | **Reason** |
| Visibility of system status | 9 | The application responses quickly to the inputs throughout the application. In areas where it is less than the initial desired speed, it indicates to the user to wait while it loads |
| Match between system and the real world | 8 | Majority of Fitbit application uses terminology familiar to its domain. The contextual languages and symbols lead to areas that are expected. An example would be “+” button on a add section leads to adding the contents (in which it relates to) |
| User control and freedom | 6 | In using the application, there are access to numerous features in which the user can control and choose from. However, in the logging of the food aspect, such as calories, the textual input needs to have exact measurements for the application to work. While this amount of freedom and control can be good for some, it can be too much for others. |
| Consistency and standards | 9 | The application keeps the general makeup design across the designed pages. The overall design look of Fitbit, such as the buttons and the UI, is polished and smooth. There seems to be a professional look to the application as well. |
| Error prevention | 8 | The error prevention, for the logging aspect, indicates there are certain areas that are necessary to fill as part of the logging aspect. The only issue seems to stem from accurate input through textual fields, which could mess up the analysis system later. It does allow the user to modify the details later if this error occurs |
| Recognition rather than recall | 8 | The context of the application uses familiar layout and buttons. |
| Flexibility and efficiency of use | 4 | In the core aspect of the application, the logging of the details is inefficient and not flexible to use. This is through capturing the user logged details to the exact digits. The amount of textual input in the other areas of the application reduces the flexibility and efficiency of the app |
| Aesthetic and minimalist design | 8 | The layout of the application is visually appealing and has a good look to it. If the user were to focus on themself and the application use, it has been minimised for efficient use. This is through the logging aspect and getting the feedback on it, such as graphically interface etc. If the user were to see what other people have said, or other activities to do, the design, in theory, is minimised but the content people put up makes it not so much. |
| Help users recognize, diagnose, and recover from errors | 7 | Whenever an error has occurred, there is a few features to correct this. The first would be a return part, in case the user went to a part accidently. The second would be a cancel section, if the user were in the middle of logging food and changed their mind. The third would be the ability to modify details. This would be in case of a section where the information needs to be updated. |
| Help and documentation | 6 | The access to the help and documentation through the navigation bar is not very clear. The widget is not the conventional of accessing the nav bar. The user have to scroll down to find the help. When they do, there are too much information to provide assistance to the user. |

|  |  |  |
| --- | --- | --- |
| **7 Best Practices to Overcome Mobile App Usability Issues** | **Rating Metrics** | **Reason** |
| Platform Usability | 7 | The application, through the simple design, layout and language use, it is pretty simple to get right started on the application. The only downside is the logging process of the application being slightly more complicated then needed. |
| Provide Value Right Away | 6 | While the value of the application is provided right away, such value is insigiThe value provided is based |
| Simple Navigation | 8 | The content is clear enough to allow the user to get from one part of the system to the next. This is from either logging the food data to finding local activities related to the activities of interest |
| Clear & Concise Content | 9 | The comprehension level of the application matches what I would have expected from the application itself. As I look through the different areas of the application, I understand the content I am looking as it is clear enough. |
| Minimize the Number of Steps | 6 | There is a number of steps that are required to get the application to provide the correct value and use to the user. While there is a attempt, such as quick log areas, this is a insignificant in the overall process of the application. In the other areas of the application, it was clear to try minimise the number of required steps. |
| Reduce Scrolling | 7 | The major parts of the application require minimum scrolling for each page. When looking at other people activities, or looking for additional information, that is when scrolling is required for the usage of the application. |
| Consider Landscape Orientation | 1 | Does not even consider it or allow it. Maybe only for accessing the camera to take photos on the application |

|  |  |  |
| --- | --- | --- |
| **Fitbit Application Evaluation** | **Rating Metrics** | **Reason** |
| User Interface Design | 8 | The interface had a professional look to it. Simple, intuitive design which was appealing to the eyes in terms of aesthetics. |
| User eXperience Design | 7 | The experience, during the testing process, was pleasant. There weren’t really any difficulties in understanding what the application was trying to accomplish in its design. However, the downfall was the logging process. This found to be more complicated then expected if the user didn’t the details they were entering. It could be interpreted the application design was for professionals, their team and other people who had knowledge of the user input. |
| Ease of use | 7 | It was easy to use, access different information and areas which was expected. The different options were very quickly understood too. The only area was the logging aspect. Unless its user are the type of person whom measures their food to the exact details, this feature is too overly complicated. |
| Overall Design | 7 | The overall design is nice. It functions very well and easy to use. |

**mySugr**

An app “to manage your diabetes and HbA1c”. It was ranked “the top diabetes app by Healthline 3 times”. The functionalities of mySugr are:

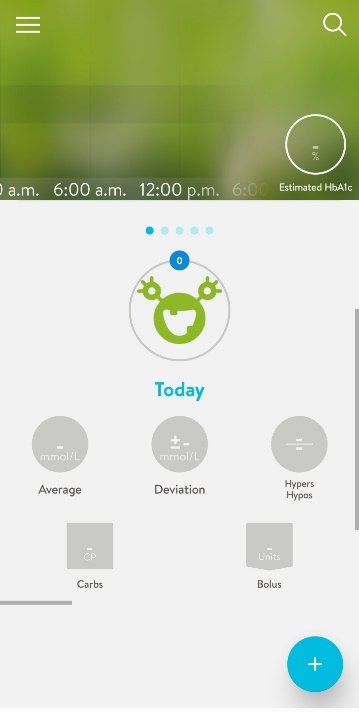
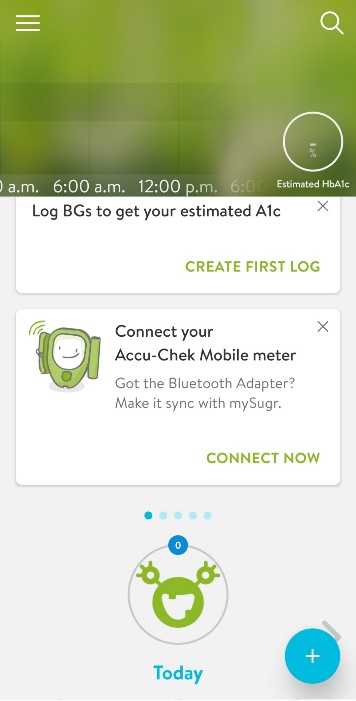
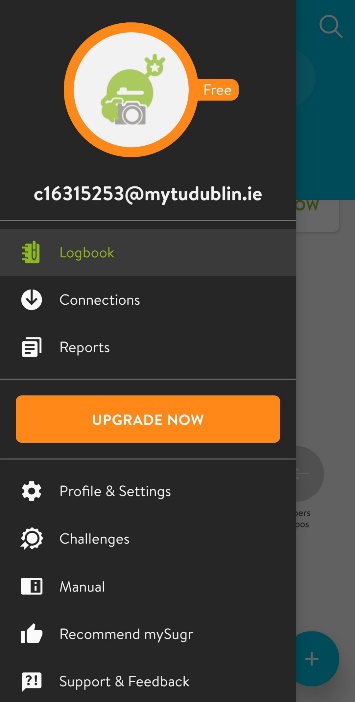
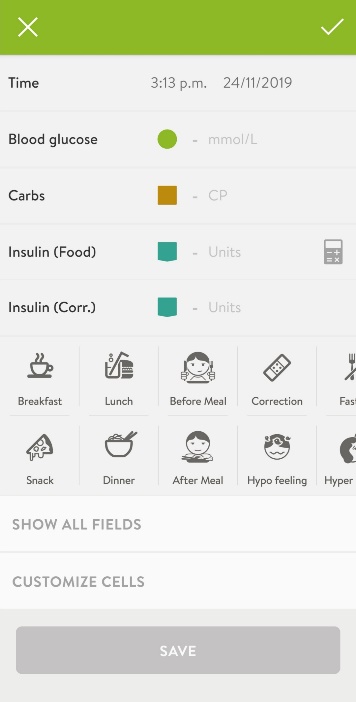
* access to easy and personalised dashboard (including diets, meds, blood sugar etc)
* clear blood sugar level graphs
* estimate HbA1c
* motivating challenges and feedback
* medical analysis (daily, weekly and monthly)
* detailed reports for the doctor
* secure data backup, which incorporates regulatory compliance, quality and safety for the user.

The areas of mySugr which were evaluated were their UI, UX, Design and ease of use. Fitbit was evaluated by the criteria of the 10 Nielsen’s Heuristics and 7 modern mobile usability factors.

|  |  |  |
| --- | --- | --- |
| **10 Usability Heuristics for User Interface** | **Rating Metrics** | **Reason** |
| Visibility of system status | 9 | As the user interact with the different components of the system, it provides small gestures of interaction back. There seems to be no area where the loading is done slow (maybe in the report generation section – be simple to highlight a waiting process there) |
| Match between system and the real world | 8 | The language and layout used are what would be expected from the target audience. It was clear the target audience was those whom had diabetes. A lot of the users for the application would be familiar with the terminology specific with their diabetes |
| User control and freedom | 8 | The application provides the user full control over what areas to access and log as part of the diabetes. The amount of options available could be overwhelming for some users. However, there is ways to go back and modify as necessary. |
| Consistency and standards | 7 | The application has a consistent layout across application, in terms of the buttons, graph placements etc. The language used is also just as consistent. |
| Error prevention | 8 | The application does not allow logging of invalid data, as this would make the application insufficient. There are clear steps taken to prevent this, such as mandatory fields to enter. The only potential errors not captured is the logging of all the details and making deliberate errors while ensuring the compulsory fields are correctly filled |
| Recognition rather than recall | 7 | Majority of the application contains features that are standard practise across mobile application development. The only unique parts are the section where the data gets logged and areas related to diabetes. |
| Flexibility and efficiency of use | 7 | The application is very efficient and flexible in the area where it functions. From providing numerous ways which the application allows the logs to occur, it is flexible to allow the user to choose how to complete the tasks. While the choice allows the flexibility, it sacrifices efficiency in the same area. |
| Aesthetic and minimalist design | 9 | The layout is simple, smooth and intuitive. The simple design enables immediate understanding of how to interact with the application. The main icon of mySugr moves with the user and has graphical elements. This includes the action of the eating motion when the user logs their diet. The colour scheme of the entire application matches up with one another too. It only changing a minor section to display the difference between the current logged data and the trends, such as weekly or monthly as examples. |
| Help users recognize, diagnose, and recover from errors | 7 | The application has the ability to allow the user to edit any detail in the application, such as logged data. This means if there was a mistake happened during the logging process, they can go back and change it. In other areas where changes cannot occur, there is no potential for the user to make a mistake. In saying that, the part of complexity of the logging process itself could leave to errors |
| Help and documentation | 8 | The access to the help and documentation is easily found and clear. The layout of the application indicates the lack of need for any. By checking out the navigational bar and scrolling down, there is a link to the documentation and contact details in case more assistance is needed. |

|  |  |  |
| --- | --- | --- |
| **7 Best Practices to Overcome Mobile App Usability Issues** | **Rating Metrics** | **Reason** |
| Platform Usability | 7 | Overall design of mySugr makes the usability very easy to use in its platform. The only critic is the logging process of the application. Because this is a essential part of the application, it makes the application less usable due to its difficultly. |
| Provide Value Right Away | 7 | The application only provides value when information gets logged. This is the complicated aspect due to the amount of choices available. However, as you progress through the logging process, the system gives out points. This itself provides ongoing value for the user as they log their food details. |
| Simple Navigation | 9 | The process of getting from one part of the application to the next was a simple task to perform within the application, as each part of the application behaved according to the described. Ie, + button, in the case of this application, was believed to mean add a new log. After clicking it, the + button led to adding a new log. |
| Clear & Concise Content | 7 | The content of the application was simple enough to understand for majority of the application. The only exception was the logging process, which was mildly eased by some areas using icons. However, the amount of options and choices can make the logging process overcomplicated. |
| Minimize the Number of Steps | 7 | All the steps seem to have been minimised to the best possible. However, the logging process is complicated by the amount of options available to the user straight away. One way this could have been reduced would be providing the options to the user. As they complete more of the section, more of it opens up. |
| Reduce Scrolling | 8 | MySugr displays the need for a small amount of scrolling. However, there have been minimised as best as possible to make it easier for the user. |
| Consider Landscape Orientation | 0 | Does not even consider the landscape mode for the use of the application. Probably due to the belief of not needing it for the application |

|  |  |  |
| --- | --- | --- |
| **MySugr Application Evaluation** | **Rating Metrics** | **Reason** |
| User Interface Design | 8 | The application overall interface is simple and intuitive. The colour scheme which was used works well across the board and blends well together. |
| User eXperience Design | 7 | The overall experience of using the application was intuitive, pleasant and easy to use. The app is very functional and designed well in functionality provided. The logging part of mySugr is the only part of the user experience where there was difficulty. The interaction with the icon is pretty interesting too. |
| Ease of use | 7 | mySugr overall was easy to use. The context and layout of the application behaved as expected. The navigational was easily accessible too. The only aspect which had moderate experience was the logging aspect. The used terminologies and layout made the experience, for that case, moderate rather the easily usable. |
| Overall Design | 9 | The overall design of the application is inviting and easily understood. Each aspect of mySugr blends well together. It is also professional designed with art and graphics in mind, giving it a more interesting outlook in its design. |

**Overall Evaluation of Industry Mobile Applications**

|  |  |  |  |
| --- | --- | --- | --- |
| **10 Usability Heuristics for User Interface** | **Fitbit Rating Metrics** | **MySugr Rating Metrics** | **Average Rating Metrics** |
| Visibility of system status | 9 | 9 | 9 |
| Match between system and the real world | 8 | 8 | 8 |
| User control and freedom | 6 | 8 | 7 |
| Consistency and standards | 9 | 7 | 8 |
| Error prevention | 8 | 8 | 8 |
| Recognition rather than recall | 8 | 7 | 7 |
| Flexibility and efficiency of use | 4 | 7 | 5 |
| Aesthetic and minimalist design | 8 | 9 | 8 |
| Help users recognize, diagnose, and recover from errors | 7 | 7 | 7 |
| Help and documentation | 6 | 8 | 7 |

|  |  |  |  |
| --- | --- | --- | --- |
| **7 Best Practices to Overcome Mobile App Usability Issues** | **Fitbit Rating Metrics** | **MySugr Rating Metrics** | **Average Rating Metrics** |
| Platform Usability | 7 | 7 | 7 |
| Provide Value Right Away | 6 | 7 | 6 |
| Simple Navigation | 8 | 9 | 8 |
| Clear & Concise Content | 9 | 7 | 8 |
| Minimize the Number of Steps | 6 | 7 | 6 |
| Reduce Scrolling | 7 | 8 | 7 |
| Consider Landscape Orientation | 1 | 0 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Mobile Application Evaluation** | **Fitbit Rating Metrics** | **MySugr Rating Metrics** | **Average Rating Metrics** |
| User Interface Design | 8 | 8 | 8 |
| User eXperience Design | 7 | 7 | 7 |
| Ease of use | 7 | 7 | 7 |
| Overall Design | 7 | 9 | 8 |

Fitbit and MySugr had similarities and differences. In their differences lies each other strengths and weakness. The functionalities of both applications are the same: getting logged data from their users and displaying the user trends. The trends are displayed textually and using graphs. That is where the similarities end. The difference are described in a few ways. The first would be the type of data logged. For MySugr, the process is to combine every detail into one log (food, time of day logged, diabetes information etc). With Fitbit, it is just food and water at the start. With the food log, it goes though the different steps and what you logged and when. These steps are broken down more in Fitbit. The activities provided by Fitbit, in terms of exercise and other activities, extend more then just the logging feature. This is something MySugr doesn’t offer. What MySugr offers which Fitbit seem to not have is the capability of downloading the logs into a report format file. Both, in their different respective ways, are the same in the overall evaluation of UI, UX, Ease of use and overall design. The only aspect where there is more difficulty then necessity is the logging aspect of both applications, It Is completed though textual input for both applications, indicating both are designed for specific measurements. It could have been easier to input logged data through generic input as an option.

## 2.4. Research Topic 3

The third researched topic was the types of technologies researched for each tier of NDMA. The researched was split into three sections for each of the three tiers. The first-tier technological requirement was the view tier. The research conducted was on the available mobile technologies with their integrated development environments. The second-tier technological requirement was the application tier. The type of research conducted was the suitable middleware technology. Finally, the last tier researched was the data access tier. Technologies appropriate for this layer were lastly researched. An investigation was also conducted into several other technologies for their application into NDMA. They were into potential cloud services, APIs, Operating Systems, Programming Languages and other technical tools. This was to see how they would assist the development of NDMA.

**Mobile Technologies and their integrated development environment**

A variety of mobile technologies were investigated for NDMA front view. As each were delved into, a number of questions were drafted up. These questions were how the researched technology would fit alongside each of NDMA requirements. The requirements talked about were the technical, user and the feature requirements. These were the main priority of the project NDMA. As part of this research, numerous sites containing the reviews of popular hybrid mobile applications technologies, used by the developing community, were looked up. Upon such review, an overview of the four most popular mobile technologies specifications were drafted up. These specifications were then compared against the requirements of NDMA. These four technologies were PhoneGap IDE, Ionic, React Native and Xamarin. An outline of each technology is displayed below.

**PhoneGap**

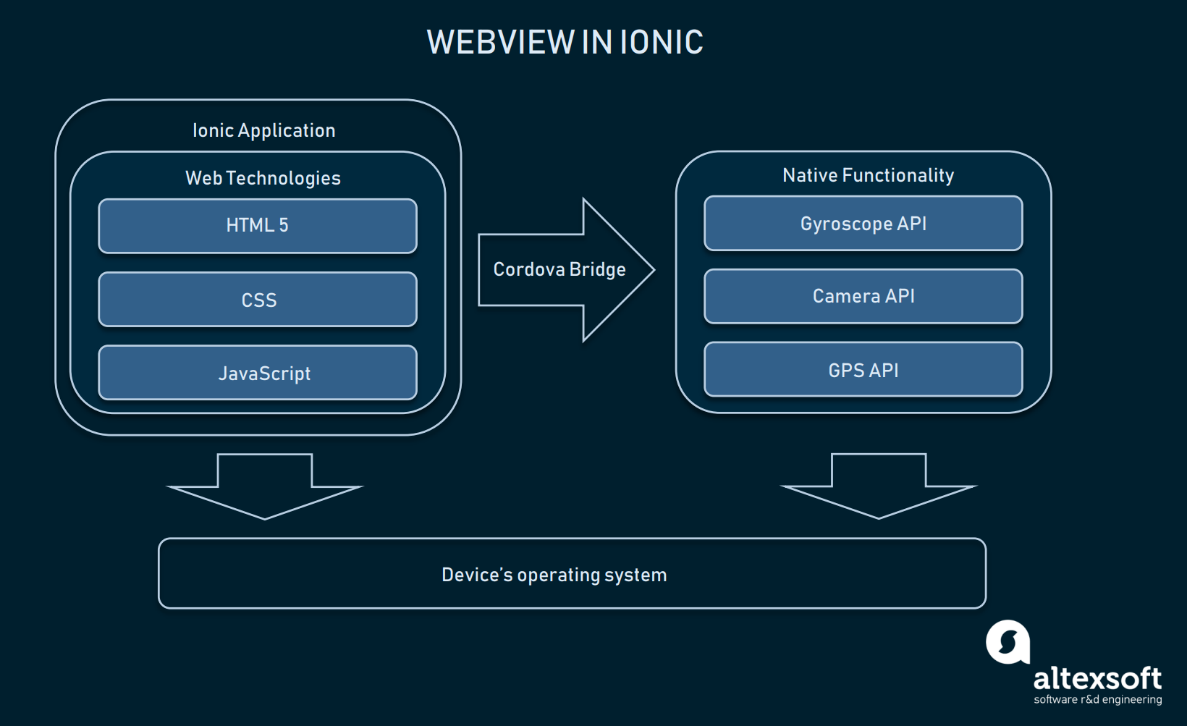
By wording of Orion Info Solutions Website, “PhoneGap is open source freely available app which can be run on different platforms. There is rapid increase in these types of apps as they are easy to maintain and save both time and money.” This is due to the applications, which were created under PhoneGap tools and technologies, are developed using modern web technologies (HTML5, CSS3, JS libraries). As such, they provide “easy testing and maintenance” to the community. They also allow for usage without learning any “additional skills”. They support “multiple platforms such as Android, iOS, Window” through providing software that are “easy to access”.

Objectively, downsides are part of the application. Due to knowledge “PhoneGap Apps are very poor in performance; they are not recommended for the gaming technologies as compared to native apps”. They also fail “in providing the access and control to the user” due to their “slow processing”. The same apps “become inefficient” when working with the same “native apps”. As a result of such design decisions, the layout of PhoneGap apps are not “as good as native apps”. 

**Ionic**

Through insight of the Altexsoft Website, Ionic was created in “2013 as an open-source SDK for hybrid mobile applications”. This has resulted in “more than 5 million apps built using it”. The growth of the “strong [growing] community” is access to the “Concise documentation” which was built with the application. Some of Ionics’s strengths lies in the ability to “providing platform-specific UI elements … for iOS and Android”. This would engage many types of developers with two goals in mind. The first would be to target as many mobile platforms as possible. The second goal would be reducing the amount of repetitive or redundant work required to meet such target users. The overall aim is to combine these goals at simultaneal rates. One way to accomplish that is to incorporate “quick prototyping” as part of their development.

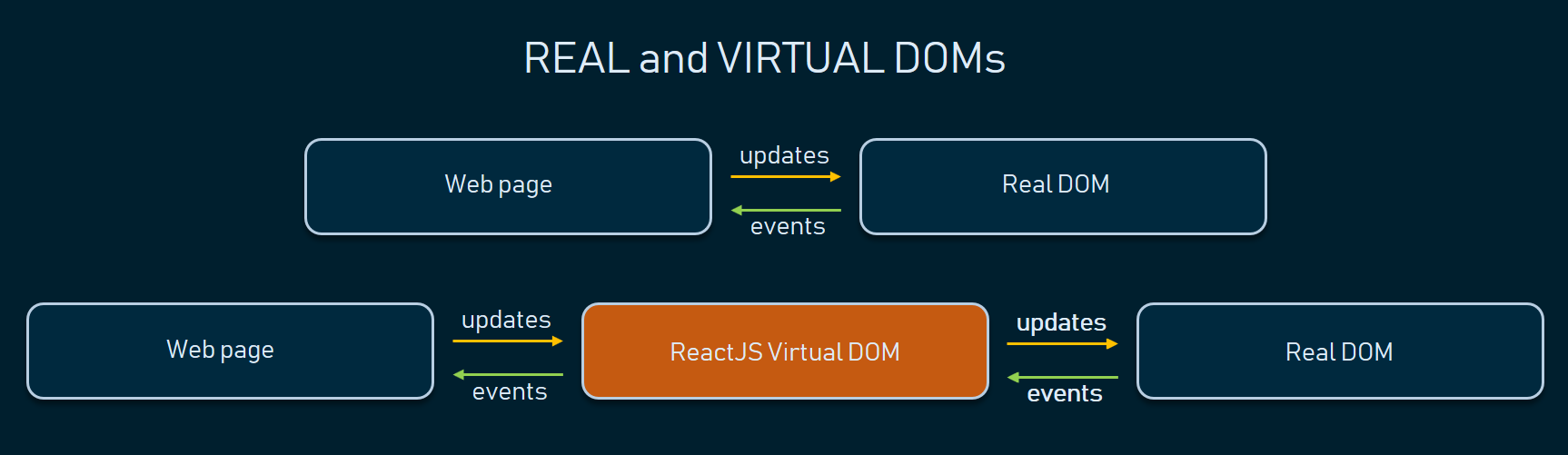
However, a few areas of weakness were discovered from the usage of Ionic. The first starts with “Absence of hot reloading”. This is a feature which allows developers to create changes in a layout file and see the result in real time. This, combined with any application developed using Ionic, results the system being a “Plugin-dependent system”. This can result in potential “security issues” for the application, if there are no security files supported by Ionic. The final issue, which had a significant part in the decision making, is the “performance is lacking [when compared] with native applications”.



**React Native**

The Altexsoft Website states React Native is “a hybrid mobile-app development framework for iOS and Android.” React Native has been created by Facebook and is open source for growth of community developers. One of Reactive Native strengths was being built using the tools of ReactJS. This is “a JavaScript library that [uses] the speed of JavaScript”, making applications “highly dynamic and responsive to user input “. Another React Native strength is its ability to combine native application development with JavaScript UI development. This would result in a speedy development in comparison to some other applications. The speed of the development comes from the usage of the “Virtual DOM in ReactJS”. This ensures both” user experience” and “developer’s work” are improved simultaneously.

A con of React Native is the tight coupling of business logic. This is an issue when system’s design finds “HTML in [the] JavaScript”. This would make it difficult to effectively maintain the application in the long term. The “Documentation”, in comparison to the “High pace of development” of React Native, is found to be lacking. As a result, it was found by the community, there is “Lagging SDK Updates” for the React Native libraries.

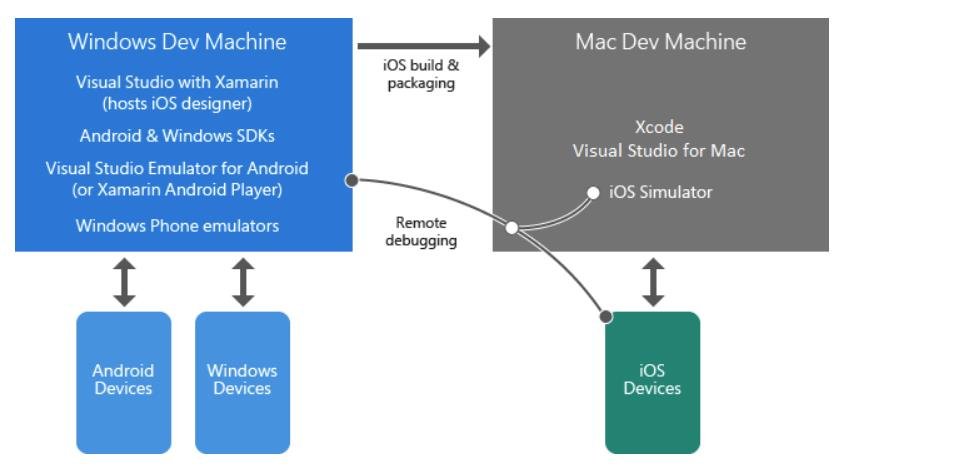
****

**Xamarin**

The last tool being covered today, also evaluated by the Altexsoft Website, is Xamarin. “Xamarin is a tool used for cross-platform mobile app development that allows engineers to share about 90 percent of code across major platforms”. “It is based on the Microsoft technology stack and already has a community of over 1.4 million developers”.

Being “Open Source Technology with Strong Corporate Support” gives Xamarin a competitive advantage against other IDEs. The support which is provided by the community is “Full Hardware Support”. This is also provided developers from private institutions. This as a result would grant “Simplified Maintenance”. Such efforts would ensure one goal from the community. Any application developed from Xamarin are equipped with “Performance Close to Native” applications.

Imperfections and Drawbacks from Xamarin can be drawn. Due to the corporate support, there is a “High Cost [of development] for Professional and Enterprise Use”. There is also a strict guideline of the required language needed. Applications derived from Xamarin ide usually are a “Larger App Size”. These drawbacks result in “Basic Knowledge of Native Languages” are required to build such applications. The last drawback is the lack of “Compatibility Issues with Third-Party Libraries and Tools”. This is due to mistrust of larger cooperation’s from the potential of hackers.



**Mobile Requirements for NDMA**

The feature requirements of NDMA are:

* Diet Logging System,
* Simple, intuitive Graphical UI
* Template for diet
* Advisor / Recommender System
* Display Trends Analysis
* Cater to user goals and macronutrients nutrition deficiencies

As a result of these features, several NDMA mobile requirements were derived and a table drawn. An X was placed inside the column if the platform provided the desired requirement.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NDMA Mobile Requirements** | **React Native** | **Ionic** | **Xamarin** | **PhoneGap** |
| decoupling of the layers |  | X | X | X |
| Fast performance similar to native mobile applications | X |  | X |  |
| Ability to handle graphical elements while keeping high speed | X |  | X |  |
| Familiar language used in its development | X | X | X | X |
| Good documentation and community support |  | X | X | X |
| Ability to put the user in control and freedom | X |  | X |  |
| Use a framework that enables modern technology uses | X | X | X | X |
| Quick prototyping and development | X | X |  | X |
| **Matching Result requirements** | **6** | **5** | **7** | **5** |

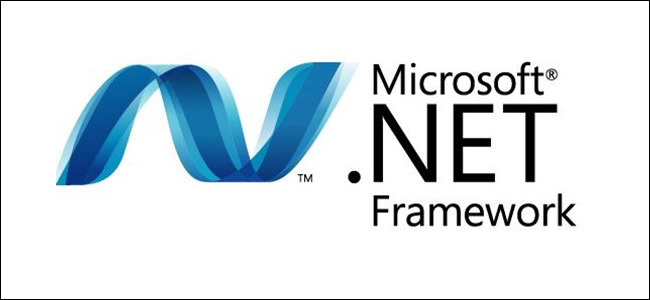
Due to being the closets to match all of the necessary requirements, Xamarin was chosen. Xamarin is a Microsoft product. This was interpreted as only few other technologies could sync up well with it. The best match would ideally be other Microsoft technologies. As a result, only a few different technologies were considered for the middleware and backend.

**Middleware Technology**

Described by Techopedia website, “The .NET framework is a software development framework from Microsoft. It provides a controlled programming environment where software can be developed, installed and executed on Windows-based operating systems”.

The main design principles behind the framework are Interoperability (allows programs functionalities to be accessed outside .NET), Common Runtime Engine, Language Independence, Base Class Library, Ease of Deployment (ensure the ease of installing programs without interfering with previously installed applications) and Security. Because of these features, .Net framework has been employed on many applications to allow separation of concerns between the different areas of the applications. These include front-end to backend communication.

Due to the .Net framework being a Microsoft product, it was presumed to work well with Xamarin technologies. A little more investigation did confirm this. This framework was chosen for NDMA as the middleware technology. This choice came from .Net meeting the technical requirements of NDMA and working well with Xamarin. With the middleware, the main requirements of NDMA was ease of communication between the front & backend, separation of data & view logic from the application logic and Security. All the other benefits of the .Net framework were seen as additional benefits.



**Cloud services**

All cloud services provide solutions. This solutions include **Infrastructure as a Service** (IaaS), **Platform as a Service** (PaaS), and **Software as a Service** (SaaS). These solutions can be used for various services, such as analytics, virtual computing, storage, networking etc. These services allow for remote connections from anywhere in the world, as one main aspect of the cloud is it resides on the world wide web. The three competitive public examples, in the present moment, are Azure Microsoft Cloud services, Amazon Web Services (AWS) and Google Cloud Platform Services. All three would provide the same role for both remote services and storage purposes. The only significant difference between each is the producer of the servicers, with Microsoft hosting Azure, Google with its own and Amazon with its web services.

The cloud services were investigated as part of the plot to hold the database of NDMA remotely. This would provide users accessibility to the cloud database through the mobile application. A requirement of NDMA was to be able to host multiple people simultaneously. As such, a cloud environment was required.



**Backend Database technology**

Cloud services were investigated for backend storage. This is due to databases services being provided as part of the cloud. Many have found to contain a form of SQL servers which can be used from local applications. In the case of the user having the inability to access the internet, SQLLite was investigated for local storage. Part of the two versions of databases, local and remote, is to sync the data with one another. This would provide the best possible experience to the users of NDMA.

**Operating Systems**

**Android**

According to lifeWire, “Android is a popular, Linux-based mobile phone operating system developed by Google”. It is an open source project which Google provides to various device manufacturers for free. From Huawei to Samsung, Android is used in each device development and is maintained to adhere to the different specifications. This would allow for diverse use of the phone, providing different phone user experiences. However, as a result is very difficult to keep updated against the different risks against the devices.

****

**iOS**

Recombu states “iOS is the mobile operating system that runs on Apple’s mobile devices, i.e. iPhones and iPads. It’s the main software that allows you to interact with your Apple phone or tablet”. While Android is versatile with the devices employed, iOS is restricted to apple products and apple software only. This makes the design more maintainable due to the company able to design the two together simultaneously while making the application safer, when compared to Android, by preventing downloads from Third-libraries sources.



Due to the plan of developing NDMA as being cross-platform mobile application, specifications of both android and iOS were investigated. This is to understand the differences and needs of both applications. The main differences between the two platform is described as OS behaviour as hardware integration. Where Apple and iOS are coupled tightly, Android has a diverse set of hardware which it is developed on. Thus, both software’s are kept in consideration when NDMA is built.

**Programming Languages**

**C#**

When reviewing “Geeks for Geeks Website”, information found about C# are describing the language as “a general-purpose, modern and object-oriented programming language”. The language was developed by a Microsoft team within the .NET initiative. This team led by Anders Hejlsberg. The language was approved by the European Computer Manufacturers Association (ECMA) and International Standards Organization (ISO). C# is similar to Java syntactically and easy for users who have knowledge of C, C++ or Java.

C# is the main programming language of Xamarin and the .Net framework. Because of this, C# was investigated to see how it can be used to implement the requirements of NDMA.

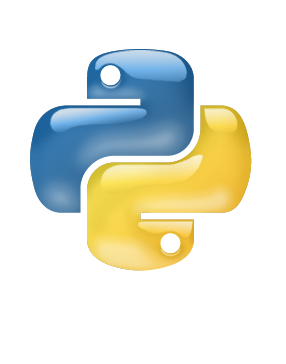


**Scripting Languages**

**Python**

Another language reviewed under “Geeks for Geeks”, Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

There were numerous food datasets which were investigated for storage within NDMA. One of the ways to investigate them properly was to run data analysis on them. Python was understood to be one of the popular languages to handle this task. As a result, Python was used for the application usage.



**Other Software and Tools**

**GitHub**

GitHub is a git repository hosting service and version control system. By managing different git repositories, it can manage different versions files and systems through individual or collaboratively efforts. The main features are to allow users to push, pull and merge different applications versions, making it a powerful tool in managing any type of projects developed in iterative steps.

Many industries employed the use of version control as part of professional practises. This allows them to maintain and change the state of the developed software. Changing the state would imply either committing new working changes, or rollback onto previous versions if the new changes creates undesired changes Part of developing NDMA is incorporating as much of professional practises as possible. Due to this reason, incorporating GitHub as the version control of NDMA was a necessary decision.



**Edamam Recipe Search API, Food Database API and Nutrition Analysis API**

The external source “Edamam” APIs allow the ability to integrate more types of food into the system without the need for another data set. The three APIs, Recipe Search, Food Database and Nutrition Analysis, can be used to get either food information or to get nutrional information based of the food inputted into the system. From the Recipe Search API, “numerous recipes can be searched from over 1.5 million recipes web sources and organized databases”. As such, “developers can create applications for virtually any popular diet or major health condition”. All three apis cater for either macronutrients or micronutrients.

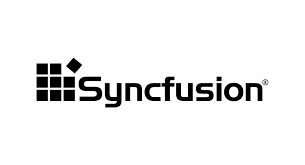
For the purpose of this application, the only API used for NDMA is the Recipe. This API was the only one that contained imagery, which was a requirement of NDMA. The other

****

**SyncFusion**

SyncFusion is a UI component suite which allows the development of professional components. This means by using their resources, there is access to professional designed plugins. This is accessed through their package Essential Studio. “Essential Studio is a software package that provides state-of-the-art solutions for start-ups and enterprises”. “It includes more than 1,600 components and frameworks for”: “WinForms, WPF, ASP.NET (Web Forms, MVC, Core), UWP, Xamarin, Flutter, Blazor, JavaScript, Angular, Vue, and React that make developers’ work easier”.

For the case of NDMA, SyncFusion was the only software found to offer development of charts for Xamarin. As a result, it was integrated and used in NDMA.



## 2.5. Other Research you’ve done

**Usability**

Usability Theory is the concept on human interaction with technology through the means of “effectiveness, efficiency, and satisfaction”. This should ensure the system is “easy to learn and remember, efficient, visually pleasing and fun to use; and quick to recover from errors”. From the same studies it is shown how people remember usage of systems combined with processing certain images.

**Assistive Technology**

Assistive Technology is the use of technology to assist people with certain disabilities. These would include vision issues, such as the blind, and providing them with appropriate solution, such as text-to-speech. Another example would be those with mobility issues and providing them with similar solutions like voice recognition.

**Modern Mobile usability**

As mobile applications availability increases to the people, the application’s requirements need to cater to ensure the user is engaged correctly. This is through ensuring the user can complete their tasks in the expected optimal time by providing the following six methods from the source website: Platform Usability, Provide Value Right Away, Simple Navigation, Clear & Concise Content, Minimize the Number of Steps and Reduce Scrolling methodologies.

**Client – Server Architecture**

By the site of ScienceDirect, the Client Server Architecture is a design in which there are two different systems interacting with each other. One side is the client, whose role is to send “requests [to] specific services or resources “and the other side is the server side, designed to be “dedicated to fulfilling requests by responding (or attempting to respond) with requested services or resources”. In the case of NDMA, the mobile application is the client and the resources it needs to access, whether it is the data from the cloud or resources from an external API, are coming from the server side.

## 2.6. Existing Final Year Projects

**Project 1**

**Title:**

Proactive Order Management System

**Student:**

Stephen Fox

**Description (brief):**

An application that allowed businesses to handle orders processes by analysing its data within the system. This system also provides businesses with information on how and when to process these orders. The order processes are placed into the system remotely through customers using a mobile application that can access and connect to the host web system.

**What was complex about this project?**

The task Scheduler manager was the most complex part of the system as many uncertain fields that could change that this system needed to calculate in order to be optimised for the overall system.

**What technical architecture was used?**

A Client – Server application with the tech tools iOS Application, Web Application and AngularJS acting as the client-side while Node.js, NuPIC, Proactive Module - Flask, Google Map Distance Matrix API and MongoDB as the server side

**Explain the key strengths and weakness as you see it**

To have an algorithm that dynamically creates task handlers based off the tasks requirements is a strength as it allows you to expand your resources and minimise data wastage at the same time.

The data involved does not seem to be protected through encryption or other methods, which could allow nearby people who could steal the data to view the data, is a major weakness of the application.

**Project 2**

**Title:**

Glucose Coach

**Student:**

Alex Kiernan

**Description (brief):**

An application designed to track people whom have type 1 diabetes to better manage their overall health by logging their blood sugar levels, their diet and their physical exertion into the coaching system. Once logged, the application will be able to provide the necessary suggestions catered to the individual user to enable them to better manage their glucose levels.

**What is complex about this project?**

The machine learning part of the overall system was the main complexity as research was needed to be conducted combined with the development of the sound process in order to properly process the user information into the system and export back accurate results to the user.

**What technical Architecture was used?**

A Client – Server architecture was used with the RESTful service links between the two, a remote relational database, a flask server and the machine learning system scikit-learn as server side while the use of a mobile application as the client.

**Explain the key strengths and weaknesses as you see it**

A key strength of the application was at the time of development, the constant monitoring of the users progress with the insulin intake was unique to the system design which provides it a competitive edge compared with similar applications

A weakness of the application would be the lack of knowledge provided back to the users whom used this application. If there were graphs that displayed over time the use of insulin intake over days and weeks, it would have made the user understand their blood sugars levels more in depth, which would have allowed them to make more informed decisions.

## 2.7. Conclusions

The main requirements for the use of NDMA are designed for the ease of use, the user experience and key polished functionality of the application. As a result of these requirements, along with the user requirements and business requirements, the technologies chosen were using Xamarin for the front-end, the .Net framework for the middleware, Azure Cloud services with for remote service and SQLite for local storage.

These technologies are best suited for the development of the application through their adaptability to the users’ requirements, the developer’s technical needs and the General User Interface requirements.