School of Electronic Engineering and Computer Science

Final Report

Programme of study:
BSc (Hons) Computer Science

Project Title: Diabeticare

Supervisor:
Dr. Raul Mondragon

Student Name:
Boran Cek

Final Year
Undergraduate Project 2022/23

Date: 2nd May 2023

Queen Mary
University of London

Abstract

The aim of this project is to conduct an effective investigation and provide insights into the alarming rise in diabetes prevalence. Additionally, this study seeks to explore the potential benefits of leveraging technology to tackle this pressing issue. Technology has been widely used to gain a deeper understanding of the challenges we face and to develop solutions that improve human well-being. In line with this, the proposed project, Diabeticare, aims to leverage modern technology to address the rapid growth of diabetes among the global population. Diabeticare is envisioned as an application designed to support individuals with diabetes by providing them with remote access to their doctor or General Practitioner (GP) through a messaging service. This feature is expected to significantly reduce waiting times for general queries, enabling patients to better manage their condition.

The research methodologies employed in this project are designed to facilitate a comprehensive understanding of the problem at hand. Specifically, secondary research will be conducted to collect raw data from individuals impacted by diabetes, in addition to surveys and studies conducted by other researchers to broaden the scope of knowledge on the topic. Additionally, the project aims to gather relevant data, identify user requirements, and develop a working application. To ensure that the development of Diabeticare adheres to the highest standards, this project will also apply software engineering principles, including domain analysis and requirement exploration. The goal of this approach is to build a robust and effective solution that best supports individuals with diabetes.

In summary, this project aims to provide an effective solution to the growing issue of diabetes through the innovative use of technology. The comprehensive analysis and testing conducted will ensure that Diabeticare meets the set objectives and delivers a reliable and efficient application to assist those affected by diabetes in managing their condition.

Contents

Chapter 1: **Introduction**

1.1 Background

Diabetes is a chronic condition that can have serious implications for key organs such as the heart, blood vessels, and eyes over time. This condition is characterised by the inability of the body to produce and use insulin effectively, which is essential for regulating blood sugar levels. When the body cannot generate sufficient insulin, excessive sugar remains in the blood, potentially leading to severe outcomes such as death or even limb amputations. Diabetes is classified into two types: type one, in which the body's immune system attacks and destroys the cells responsible for insulin production, and type two, in which the body fails to produce sufficient insulin.

The escalating rates of diabetes within the general population have become a pressing concern. According to a study by Saeedi et al (Saeedi et al, 2020), published in ScienceDirect, an estimated 4.2 million deaths among individuals aged 20-79 are attributable to diabetes alone. Furthermore, diabetes contributes to 11.3% of global deaths, with Europe accounting for 31.4% of such fatalities. In addition, Douglas Broom (Broom, 2020) from the World Economic Forum (WEF) has characterised diabetes as a "silent epidemic," which claims approximately 7 million lives each year. He also highlights that the number of diabetes cases is projected to increase by 134% by 2045, which is a significant cause for concern. These sources demonstrate the magnitude of the threat posed by diabetes to the general population and underscore the limitations of existing solutions.

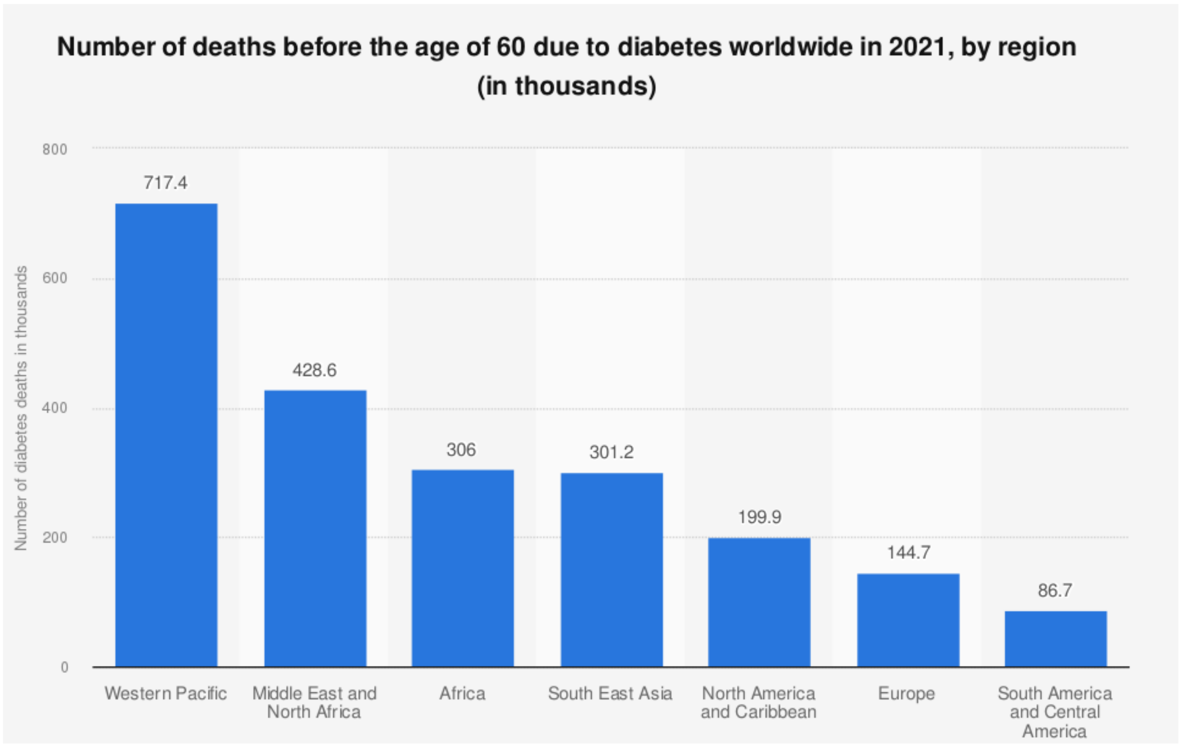


Figure 1: Diabetes around the world (Broom, 2020)

1.2 Problem Statement

Effective communication between patients and General Practitioners (GPs) has been a longstanding concern. Patients often encounter issues such as prolonged waiting times to see their GP, which can delay the timely addressing of their concerns, and difficulty remembering the details of their appointments. Notably, research by Roy P.C. Kessels (Kessels, 2003) in the Journal of The Royal Society of Medicine (JRSM) highlights the widespread issue of patients forgetting a significant portion of the medical information communicated to them by healthcare professionals, ranging from 40-80% of information immediately following a consultation. Kessels further suggests that an increase in the volume of information provided by healthcare professionals can exacerbate this issue, making it more difficult for patients to actively recall what was said. Additionally, Kessels explains how verbal communication may be a contributing factor to patient's forgetfulness during appointments.

It has been observed that patients often struggle to recall details of their appointments when the information is conveyed verbally, thereby underscoring the potential effectiveness of written or transcribed modes of communication. In light of this, Diabeticare endeavours to introduce a messaging service within its application to facilitate the transcription of information exchanged between patients and GPs. The aim is to mitigate the issue of forgetfulness by ensuring that critical pieces of information are not lost, thus enabling healthcare professionals to effectively communicate with their patients.

It is worth noting that the majority of existing diabetes management software systems do not offer a messaging service, revealing an unexplored gap in the market. While some applications on the market attempt to provide general support for patients with diabetes, they often do not directly address the key issues faced by patients. I firmly believe that adopting a more targeted approach to address patients' primary concerns, such as information forgetfulness, could be an effective solution in combatting the rising prevalence of diabetes globally. While implementing a messaging service may have potential drawbacks, such as excessive messaging, Diabeticare will endeavour to ensure that the experience is pleasant for both patients and healthcare professionals. For instance, one possible solution would be to limit the number of messages that can be sent per day, a method that will be explored in the future.

1.3 Aim

The problem that Diabeticare aims to assist is the issue of diabetes mismanagement and therefore focuses on trying to minimise the levels of fatalities seen by diabetes alone. One major way of doing this is by introducing a feature whereby patients can message their GP's. This allows patients to avoid forgetting what their GP had told them in appointments which can occur through 1:1 or telephone consultations. The ability to message GP's can help reduce time wasted where patients are reportedly waiting for up to 3 weeks to see their GP's face to face due to the COVID backlog that occurred during the pandemic. As a result of this, my project aims to help minimise the effects of diabetes on patients' life expectancy through keeping patients as close to their General Practitioners as much as possible, whilst also supporting patients through other means within the use of the Diabeticare application such as being able to access prescriptions through a PDF format. Furthermore, in order to maintain user retention, a gamification element will be added to Diabeticare which would result in continuous user engagement.

1.4 Objectives

In order to achieve the aim of this project, the following objectives must be met:

• Have a functioning mobile application possibly by using React, Node.js and npm

• To critically assess and have an accurate understanding of already built solutions and identify their benefits and drawbacks

• Reach a respectable level of knowledge of the research conducted in the area of diabetes management through articles, newspapers and/or journals

• To develop a mobile application that allows diabetes patients to manage their condition & message their General Practitioner (GP)

• Conduct primary research to gain rich and insightful pieces of data (interviews, surveys, questionnaires and/or focus groups)

• Conduct SWOT (Strengths, Weaknesses, Opportunities, and Threats) analysis on applications already in existence

• To include a simple to use interface that is comfortable for all users

• To act in accordance with the rules and regulations set out for the project

• Execute thorough software testing procedures such as Unit and Integration testing to ensure the expected functionality of the application is observed

• Have a fully functional login authentication system by using Google OAuth

• Implement a gamification element to boost app usage. For example, if the user engages with the application for 7 days in a row, then they are entered into a lottery competition.

• Conduct agile project management techniques to optimise the implementation process and allow workflow of the project to run smoothly as much as possible

1.5 Research Questions

Here are the key questions that this project aims to address:

[1] To what extent can the implementation of technology serve as a catalyst for mitigating the occurrence of diabetes cases worldwide?

[2] What are the potential strategies for enhancing communication channels between diabetic patients and their general practitioners?

[3] How can a messaging service between diabetic patients and General Practitioners be implemented?

[4] What are the potential implications and outcomes of incorporating gamification elements into diabetes management applications, and how can these elements facilitate sustained engagement among diabetic patients?

[5] What are the current trends in telemedicine for diabetes management, and how effective are they?

[6] What are the barriers to adoption of technology for diabetes management, and how can these barriers be overcome?

[7] What are the key features of successful diabetes management apps, and how can developers ensure that their apps meet the needs of patients and healthcare providers?

[8] How can technology be used to improve access to diabetes care in underserved or rural communities?

Chapter 2: **Literature Review**

2.1 Literature Review

In recent years, a growing body of research has highlighted the challenges associated with managing diabetes, including issues with communication, information recall, and the complexity of diabetes-related terminology. These challenges have renewed interest in leveraging technology to support patients in effectively managing their condition. This view is shared by Diabetes UK (Diabetes UK, 2015), which has identified a lack of confidence in managing diabetes among patients. In fact, Diabetes UK reports that 42% of people with diabetes express concern about managing their condition, a finding that Chief Executive Barbara Young describes as "extremely worrying." These findings underscore the critical need for innovative solutions, such as the proposed Diabeticare application, to support individuals with diabetes and improve their confidence in managing their condition.

Several authors have recognised the significance of poor communication between doctors and patients on patient outcomes. Tiwary et al (Tiwary et al., 2019), from the Department of Internal Medicine and Oxford University Clinical Research Unit, have conducted a study exploring the importance of communication between doctors and patients. Their case study analysis demonstrates that poor communication can potentially be the difference between life and death. Tiwary et al have estimated that approximately 27% of medical malpractice cases are a result of communication failures, indicating that a substantial number of adverse events occur due to breakdowns in doctor-patient communication, often due to a lack of understanding of medical terminology or difficulty in recalling critical information provided during medical appointments. The case study provided by Tiwary et al underscores the negative consequences of miscommunication in the doctor-patient relationship.

Case study 2 highlights the consequences of poor communication in the doctor-patient relationship. The patient in this case, a 40-year-old man, was prescribed anti-tubercular therapy and prednisolone but failed to take the former due to inadequate communication. Consequently, he was admitted to Patan Hospital with significant physical discomfort. This case study underscores the critical role that effective communication plays in promoting positive health outcomes for patients.

Moreover, the significance of effective communication between doctors and patients is further supported by a study conducted by the Joint Commission International (Joint Commission International, 2018), which investigated the level of literacy and English language proficiency among patients in the United States. The study findings highlight the importance of communication, as 49.2% of patients with limited English proficiency experienced some degree of physical harm, compared to 29.5% of patients with a good level of English proficiency. This underscores the need for clearer communication channels for patients who do not understand the various medical terminologies used by doctors. The study also found that patients with limited English proficiency had a higher likelihood of experiencing permanent or severe harm, or even death, compared to 1.4% of those with a high level of English proficiency. The percentage of patients facing more serious problems due to lower levels of English proficiency is nearly three times higher than those with higher levels of proficiency.

Moreover, Alotaibi et al (Alotaibi et al, 2014) conducted a study that delved deeper into the potential benefits of utilising modern technology, such as mobile device systems, to improve diabetes management. Specifically, they investigated the effectiveness of the SAED application, which reads a user's daily blood glucose level and generates a logbook to track their glucose levels over time. The study revealed that out of the 1657 participants who used the SAED application, there were reduced levels of hbA1c, a measure of blood glucose level. The statistical analysis also showed a confidence interval of 95%, indicating high satisfaction among users who participated in the study. These findings highlight the potential of mobile and technological devices in managing diabetes more effectively.

In addition, Bonoto et al (Bonoto et al., 2017). from JMIR Publications conducted a randomised controlled trial to evaluate the efficacy of mobile applications in assisting diabetic patients in their treatment. The study found significant improvement in the level of hbA1c, which measures average blood glucose level, among patients who interacted with diabetes-related applications. This research further supports the use of technology in managing diabetes, as it has been found to be effective in assisting patients in controlling their condition.

On the other hand, a number of studies suggest that technology may not be an answer for addressing issues related to miscommunication, information recall, and understanding of terminology in diabetes care. Accenture researchers Kaveh Safavi and Brian Kalis (Safavi and Kalis, 2020) conducted a study on the usage of digital health in the United States, which found that the percentage of individuals using health applications dropped from 48% in 2018 to 35% in 2019. Figure 8 of the study shows that patient trust in doctors is very high, at 84%, while trust in technology companies is low, at just 45%. These figures suggest that the usage of health applications is expected to decline further, indicating that such applications may not be widely embraced by the general public. This could pose challenges for applications like Diabeticare, which may struggle to achieve key objectives such as reaching a large number of diabetes patients.

Ahn and Stahl's (Ahn and Stahl, 2019) study highlights the disadvantages and inaccuracies associated with diabetes applications. These findings suggest that many health applications, including diabetes related ones, lack government regulation, which may result in providing users with incorrect information, potentially causing harm to the user's health and safety. In their investigation of 46 health-related applications, Ahn and Stahl found that 70% of these applications did not provide documentation for their calculation methods to generate user recommendations. Additionally, 59% of applications allowed users to proceed with calculations despite missing input(s), which may lead to inaccurate information being provided to the user unknowingly. Therefore, these findings underscore the importance of regulation and reliability in health applications such as Diabeticare, should it be released to the public. As such, Diabeticare must seek regulation by the NHS in order to establish user trust and ensure the accuracy of its recommendations.

After analysing the situation from various angles, it is clear that while diabetic applications can be useful for patients, it is important to consider the potential drawbacks as well. The findings revealed two significant aspects. Firstly, the critical challenges associated with ineffective communication between doctors and patients, and the negative impact it can have on diabetes management. Secondly, how technology can assist in addressing issues related to diabetes mismanagement. Nonetheless, further research is needed to gain a more comprehensive understanding of diabetic applications.

2.2 Competitor Analysis

2.2.1 Diabetes:M

Diabetes:M is a mobile application that serves as an efficient tool for monitoring and managing diabetes. The application provides a diverse range of features to assist individuals living with diabetes, including a logbook function that enables users to record essential information such as glucose readings and carbohydrate consumption. Moreover, Diabetes:M features analytical charts that present users with a comprehensive overview of their glucose, blood pressure, and hbA1c history, allowing them to gain valuable insights into their condition management over time. However, it is worth noting that Diabetes:M's user feedback approach may be inadequate, as evidenced by the removal of the bolus advisor feature - a prominent feature that users actively engaged with based on Google Play Store reviews. This lack of attention to user feedback may result in customer attrition, with users feeling undervalued and unappreciated.

2.2.2 Fooducate

Fooducate is a mobile application that is designed to educate its users about the food they consume. Like other similar applications, Fooducate features a logbook to monitor food intake and analyse sugar consumption through user input data. In addition to these common features, Fooducate offers a unique diet coach that assists users in maintaining a healthy and balanced diet. One notable limitation of Fooducate is its exclusive accessibility through the Google Play Store, limiting its potential user base to Android users only. This exclusion of iOS users may have implications for the application's overall reach and adoption. Moreover, to access advanced features, users must pay a subscription fee, potentially deterring some users and negatively affecting user retention rates. These factors underscore the importance of providing accessible and cost-effective options to maximise the application's appeal and utility to users.

2.2.3 mySugr

mySugr is a diabetes focused mobile application that places a significant emphasis on logging and tracking diabetes-related data. The application offers valuable features, including the ability to view user-inputted data up to 3 months, providing users with a comprehensive overview of their diabetes management effectiveness. Additionally, mySugr can connect with the Apple Health app on iOS devices, enabling the collection of data based on user physical activity. Despite its beneficial features, mySugr's subscription-based model may hinder its accessibility to users with financial constraints. The application's essential features, such as insulin dose calculation and blood sugar level logging, are only available through a subscription, limiting its potential user base. This limitation may cause potential users to opt for other applications or deter them from using mySugr altogether.

In conclusion, the conducted competitor analysis provides valuable insights into the features and user reactions of existing diabetes-focused mobile applications. These findings are instrumental in identifying market gaps and determining areas for improvement. One such gap identified is a lack of applications that prioritise educating diabetes patients on the disease. While applications such as Diabetes:M emphasise tracking and managing diabetes, they fail to address crucial factors such as poor communication and medical jargon that contribute to patient confusion and low understanding of the disease. Previous studies have shown that diabetes patients often leave doctor appointments feeling inadequately informed due to difficulties in understanding medical terms and communication barriers. Addressing this gap in the market by developing an application that focuses on educating diabetes patients in easy-to-understand language could help improve patients' understanding and management of the disease.

Chapter 3: **UML Diagram & Requirements**

3.1 Use-Case Diagram

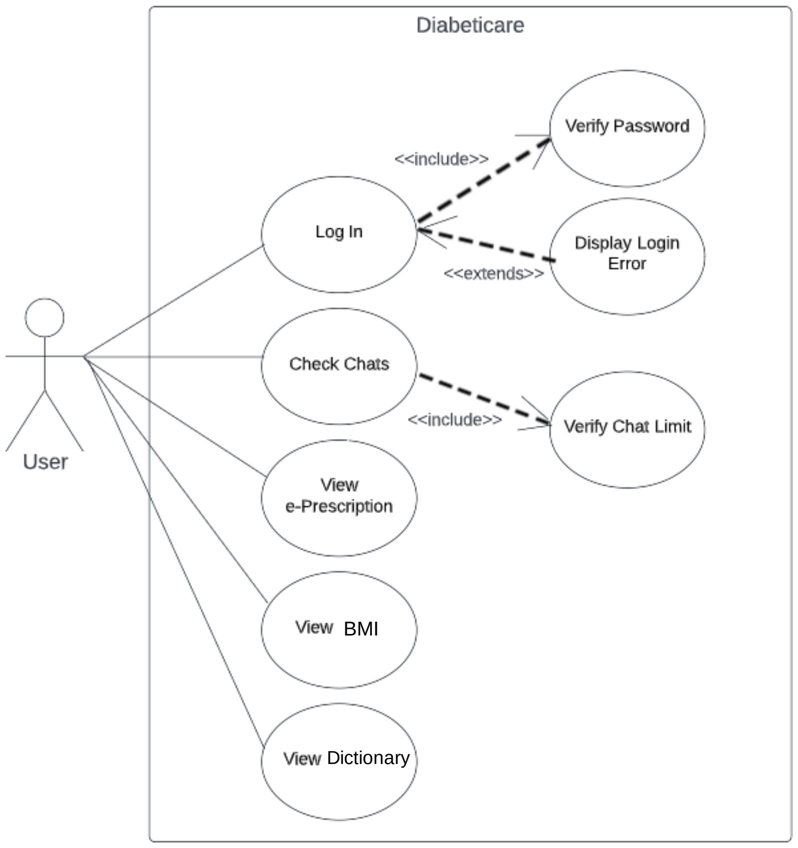


Figure 2: UML Use Case Diagram

Explanation of Key Use Cases:

•**Log In:** This use case is the aspect of the application that the user will see when first launching the application. They are required to go through the authentication process by entering their login information.

•**Verify Password:** When the user types in their login details, the system itself is going to verify the password against what is stored in the database before completing the log in process. 'Verify Password' is an 'include' relationship as whenever the base use case is executed, the 'include' use case is executed as well. In this scenario, once the user enters their details, the system automatically checks to verify the details.

•**Check Chats:** This use case demonstrates the ability of the user to be able to check their most recent chat logs. This part of the application will also allow the user to engage and communicate with their doctors/general practitioners.

•**View e-Prescriptions:** This use case illustrates the ability of the user to check any new prescriptions that they have received from their doctor. It will be presented in PDF format which can easily be read by the users preferred pharmacy.

•**View Notes:** This use case shows how the user can check their notes feature of the application. This is where the user can write down anything that they want to ask their doctor in relation to their diabetic condition.

•**View Dictionary:** The 'View Dictionary aspect of the application allows the user to read through a dictionary of terminology related to diabetes in order to educate themselves on what each word means. This will help the user to understand what doctors are saying during appointments which would help reduce the level of poor communication.

3.2 Project Requirements

To ensure the successful completion of a project, it is imperative to meticulously identify all necessary requirements (Functional and Non-functional requirements in Appendix F and E respectively). These requirements are broadly classified into hardware and software related components whereby hardware requirements encompass vital physical components, while software requirements may encompass essential operating systems, programming languages, databases, and other relevant software applications required to implement the project. It is crucial to address these requirements at an early stage to guarantee that the requisite resources and tools to deliver the project are ready at hand in order to be conducted successfully.

3.2.1 Hardware Requirements

• **WiFi or cellular data** – To ensure the seamless operation of the application, it is crucial to guarantee uninterrupted access to internet connectivity, be it through WiFi or mobile data. For instance, when a patient attempts to send a message to their General Practitioner, the successful transmission of the message relies on their connectivity to the internet. Failure to establish a reliable connection would lead to an unsuccessful transmission, ultimately resulting in the General Practitioner's inability to receive the message.

• **Storage** – To ensure the faultless operation of the e-Prescription feature, it is imperative for the patient to possess an adequate amount of storage on their device to accommodate the e-Prescriptions they receive from their General Practitioner. Insufficient storage capacity may impede the download of e-Prescriptions, ultimately hindering timely access to medication.

• **Display** – The readability and ease of navigation are crucial factors that determine the effectiveness of Diabeticare's services for the patient. Hence, a high-quality display would significantly enhance the user experience. For instance, clear visibility of the General Practitioner's response to the patient's messages is essential to prevent potential misinterpretation of critical information that may have life-threatening implications.

3.2.2 Software Requirements

Diabeticare is an integrated system developed using a multitude of software applications, which work in harmony to ensure seamless functionality. The following table illustrates the software components and packages utilised in this project, along with their respective version numbers.

Software	**Version**
JavaScript	1.5.0
React Native	0.70.5
React	18.1.0
Node.js	19.7.0
Expo	47.0.12
Firebase	9.17.1
React-navigation/native	6.1.3
React-navigation/native-stack	6.9.9
Expo-checkbox	2.2.2
Expo-print	12.0.0
Expo-sharing	11.0.1
Expo-status-bar	1.4.2
React-native-elements	3.4.3
React-native-gifted-chat	2.0.1
React-native-render-html	6.3.4
React-native-safe-area-context	4.4.1
React-native-screens	3.18.0
React-native-svg	13.8.0

XCode Simulator	14.3

Figure 3: Software requirements table.

Chapter 4: **Design and Method**

This particular section is dedicated to examining the application of the project from a design and methodological standpoint. The focus will be on clarifying the underlying rationale behind the various principles that were employed, as well as explaining the project management methods that were used to facilitate effective planning and control measures to the greatest extent possible.

4.1 Architectural Design

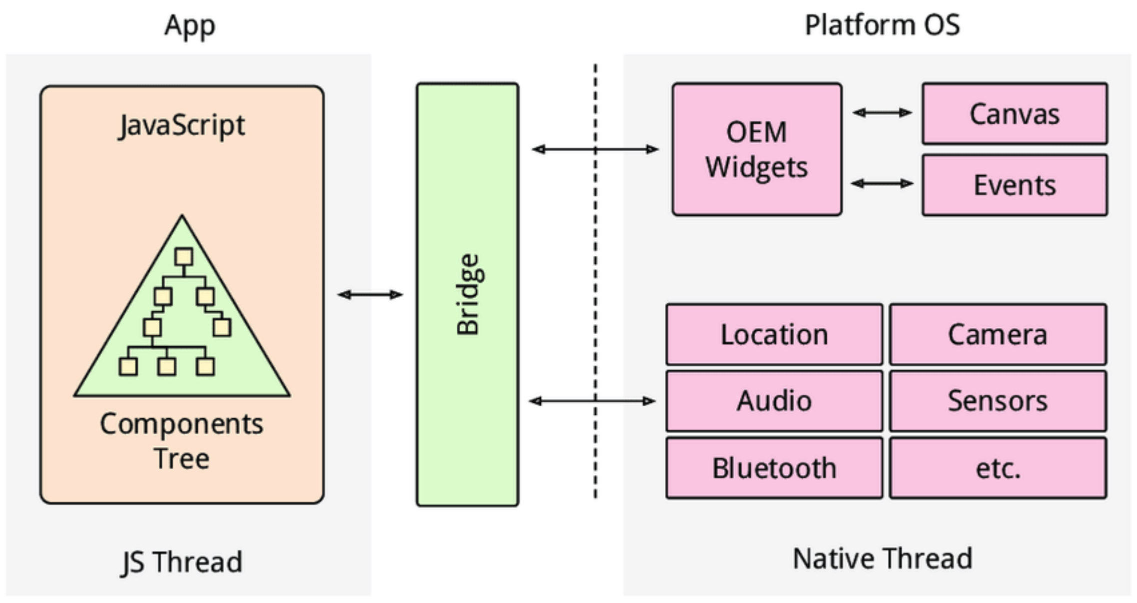


Figure 4: Overview of the React Native architecture (Image from: https://www.researchgate.net/figure/Overview-of-the-React-Native-architecture_fig8_355670705)

During the development of Diabeticare, the React Native framework was employed as the primary framework due to its robust and flexible nature, enabling the creation of high-quality native applications with the efficiency of JavaScript and React. As highlighted earlier, the JavaScript thread serves as the cornerstone of the architecture, executing the application code and controlling the user interface rendering and logic. Thus, it is crucial from a design perspective to maintain the smooth operation of the thread to ensure a responsive and seamless user interface.

To achieve optimal performance, React Native employs the Bridge, depicted in Figure 5, to facilitate message and event exchange between the JavaScript thread and the Native thread. When the JavaScript thread needs to access a module on the Native thread, it dispatches a message to the Bridge, which in turn conveys the message to the relevant native module. Upon receiving the message, the Native module processes it and responds to the Bridge, which then forwards the response back to the JavaScript thread. This process occurs asynchronously, allowing for the concurrent execution of other tasks while awaiting a response. By leveraging the Bridge, React Native ensures effective communication and seamless execution across both threads, resulting in a responsive and fluid application experience.

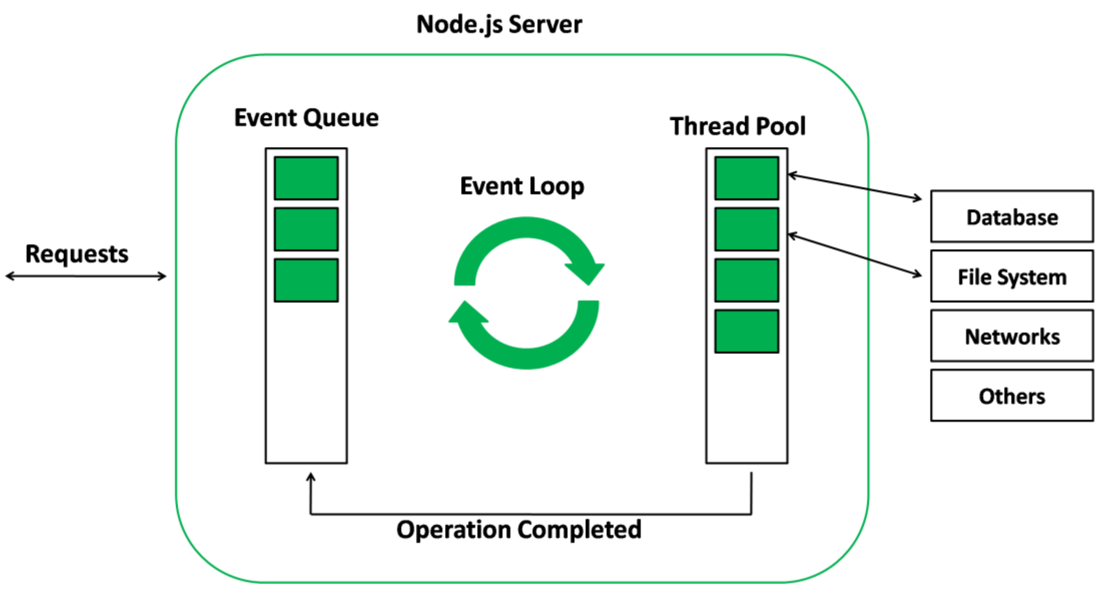


Figure 5: Overview of the Node.js architecture (Image from: https://www.geeksforgeeks.org/node-js-event-loop/)

As illustrated above, the Node.js server is a crucial component of the Diabeticare design, leveraging its powerful server-side JavaScript runtime environment. The Node.js architecture is built upon an event queue, which maintains a specific data structure that contains a queue of events and their corresponding callback functions that are awaiting processing when the event loop becomes available. The event loop, in turn, continuously monitors the event queue for new events and executes the associated callback functions on a single thread, ensuring seamless event processing.

Additionally, the Thread Pool plays a critical role in handling computationally expensive tasks. Whenever a function requires significant processing power, it is offloaded to one of the worker threads, which is responsible for handling computationally intensive tasks. This allows the main thread to focus on other tasks simultaneously, enabling the application to handle multiple requests efficiently and resulting in an overall improvement in performance. The Thread Pool is an essential feature of Node.js in terms of enabling the construction of scalable and high-performance applications capable of handling large volumes of data and traffic.

4.2 Norman’s Design Principles

The development of Diabeticare was centred around the effective utilisation of Norman's Design Principles, which are fundamental in achieving user interfaces that are intuitive, user-friendly, and efficient. These design principles are composed of three key elements:

Natural Mapping – refers to designing interfaces that closely correspond to real-world actions they represent. This principle was primarily implemented in the application's icons, where the home button, for instance, is accompanied by a small house icon, indicating that the user will be directed to the home page when pressed.

Feedback – emphasises the importance of providing immediate feedback to users upon interacting with a part of the application. Users must see the result of their actions and comprehend the action they took. This element is particularly emphasised in the chat system, where patients can see their message wrapped in a blue bubble once they have sent it to their general practitioner, indicating that the message has been delivered to its intended target.

Affordance – signifies the need for clarity in an object's properties, explaining how it can be interacted with based on its visual appearance and physical properties. This feature is particularly prominent in the BMI calculator of the application, where the 'calculate' button appears and feels like it can be pressed, owing to its colour, shape, and the text on it that implies its intended purpose.

4.3 Agile & Waterfall method

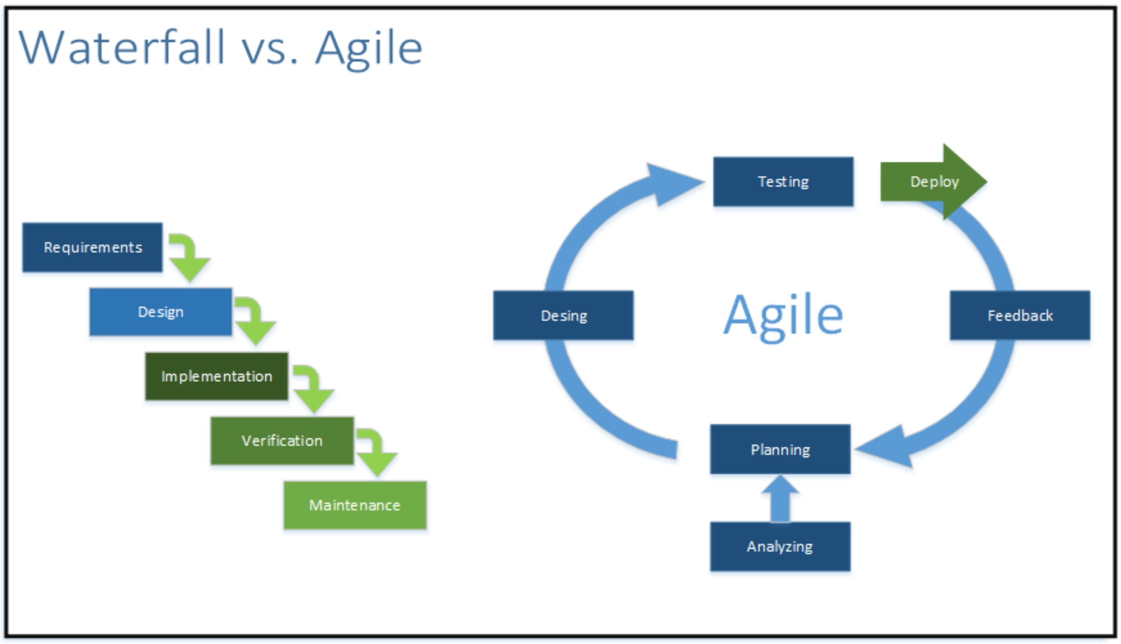


Figure 6: Overview of the steps involved in the Agile and Waterfall methodologies. (Image from: https://www.devteam.space/blog/agile-vs-waterfall-which-methodology-is-right-for-your-project/)

The software development methodologies of both agile and waterfall have proven to be effective in managing the development lifecycle. For the implementation of Diabeticare, a hybrid model was employed, utilising both methodologies simultaneously to ensure maximum effectiveness in project management.

The waterfall method was employed during the planning phase to effectively define project requirements and scope, as this methodology follows a sequential approach where each stage of planning/development must be completed before moving on to the next. This structure and focus on planning were beneficial during the project's planning phase.

In contrast, the agile method was used during the development phase, prioritising short iterations, and breaking down tasks into more manageable chunks. This approach was effective in detecting and addressing issues early on in the development process, rather than later on, which can be more complex and time-consuming. Overall, the hybrid approach provided a well-balanced strategy for the development of Diabeticare.

4.4 Application Walkthrough

The application walkthrough (Appendix G-J) entails a visual walkthrough of the Diabeticare application, highlighting its distinctive features and functionalities.

Chapter 5: **Implementation**

This chapter will delve into the details of the decision-making process and the rationale behind the selection of a specific platform or technology, which was instrumental in paving the way for Diabeticare's success. It aims to justify the vitalness of these technologies in ensuring the fulfilment of Diabeticare's objectives while emphasising their practical application.

5.1 Rationale for Platform and Technologies

5.1.1 iOS

The idea of progressing to build Diabeticare on an iOS device was not an easy decision, however there were some benefits to starting off with an iOS build rather than Android which we will go onto explore.

Development Environment – In contrast to iOS device application development, Android is often regarded as a more complex operating system due to its less developer-friendly environment. This is largely attributed to the larger range of screen sizes and display quality variations that exist within the diverse pool of mobile phone manufacturers that adopt the Android operating system. Conversely, the limited range of screen sizes on Apple's iOS devices has made it a more straightforward operating system to focus on, as it provides an easier task for developers to ensure a consistent and comfortable user experience.

For Diabeticare, it was deemed more appropriate to construct the application on an iOS system due to the stability of screen sizes and long-term support for them, a result of Apple's limited range of screen sizes and their persistent support for them. Android's market share among mobile phone devices is undoubtedly more significant, but this ubiquity leads to potential ambiguity in application development due to the overabundance of different screen sizes and display qualities. Ultimately, the choice to develop on iOS was driven by the desire to maximise user comfort and ensure a consistent experience for users, making it the most suitable option for the development of Diabeticare.

User Experience – Compared to its Android counterpart, iOS devices are widely perceived to be more upscale in nature. This perception is primarily attributed to their superior design and display quality, which aims to provide users with maximum satisfaction. Given the critical role of messaging in a diabetes management application, it is imperative that users can view messages as clearly as possible to avoid misinterpreting vital information provided by their General Practitioner. The importance of devices with high-quality displays in such an application cannot be overstated, as even slight inaccuracies in information retention by the user may lead to unwanted outcomes. Therefore, the premium nature of iOS devices is particularly well-suited for the development of a diabetes management application where message clarity is of utmost importance.

App Store Quality Standards, (Apple, 2019)– It is widely acknowledged that the App Store on iOS devices maintain a rigorous and stringent quality control system, which distinguishes itself from the quality standards observed on Android's Google Play store. As a result of the exacting quality control measures imposed on iOS applications, users are assured of a reliable, secure, and user-friendly experience when utilising applications like Diabeticare. By meeting the stringent quality control requirements, Diabeticare aims to foster and maintain a high level of user trust, thereby enabling users to utilise the application with utmost confidence.

5.1.2 JavaScript and React Native

The use of JavaScript as the primary programming language for constructing Diabeticare is rooted in its effective use in generating interactive content and dynamic user interfaces. Given the extensive interactivity requirements of Diabeticare, JavaScript was deemed an ideal choice for constructing the application. The adoption of React Native was motivated by its advantages of fast development, streamlined maintenance, and cross-platform compatibility, the latter of which is a feature that Diabeticare aims to leverage when expanding onto other operating systems, such as Android, in order to save time and effort.

5.1.3 Node.js

Diabeticare effectively leverages the node.js runtime environment to optimise its application architecture. The event-driven design of Diabeticare ensures a highly scalable and efficient application where it employs an event loop to process input/output operations asynchronously, facilitating the ability of the application to manage a large volume of requests while maintaining optimal performance and stability. Additionally, the extensive library of node.js offers a surplus of tools and resources, which greatly assist in the development of complex applications such as Diabeticare.

5.1.4 Expo

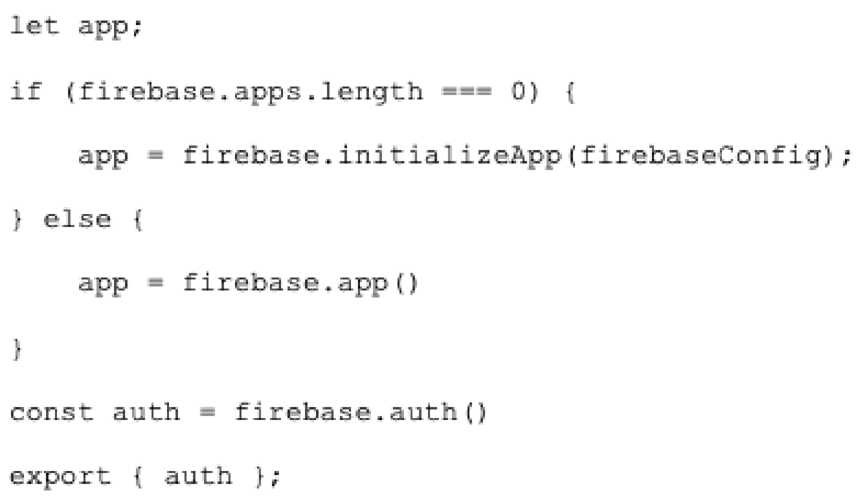
During the development stages of Diabeticare's construction, Expo played a critical role in ensuring the application's desired behaviour and user experience. By leveraging Expo's comprehensive development environment, our team was able to build and deploy mobile applications while viewing them in an iOS simulator. Additionally, the easy-to-use device APIs provided by Expo enabled the construction of Diabeticare in high-quality due to its improved efficiency, therefore facilitating seamless interaction with the application as we progressed through the implementation phase. Expo was instrumental in being able to visually track application progress, ensuring timely and effective issue resolution. Therefore, Expo remains one of the most vital tools in the implementation part of the project.

5.2 Essential API’s, Packages and External Libraries

5.2.1 Firebase User Authentication API

Firebase is widely recognised as a leading Backend-as-a-Service (BaaS) platform, providing developers with a comprehensive suite of tools for building mobile and web applications. Among its many features, Firebase offers a highly robust authentication solution that seamlessly integrates with application development, enabling developers to provide secure and efficient login systems for their users.

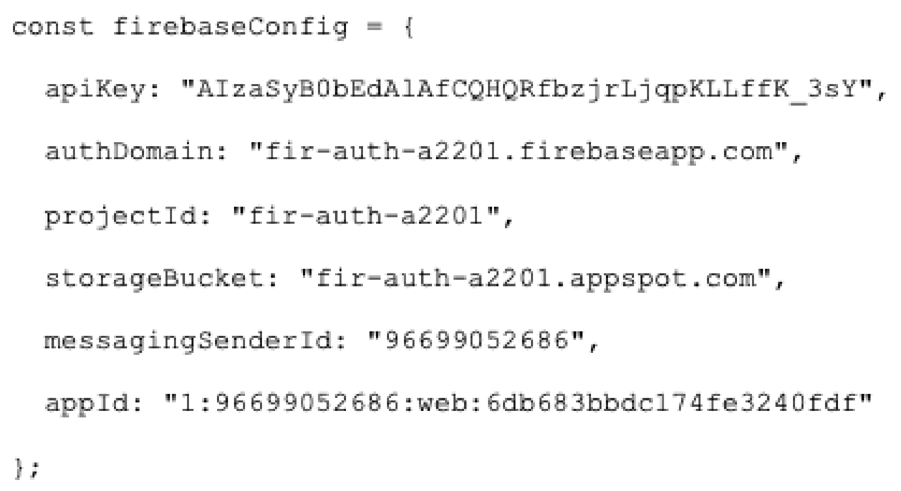
The decision to leverage Firebase's authentication capabilities was driven by its reputation for delivering fast and reliable authentication solutions, while significantly reducing development time. One of the most appealing aspects of Firebase is its ease of use, which provides an intuitive API that effortlessly interacts with Diabeticare to grant access to authenticated users, thus bolstering the security of the platform and reducing the risk of unauthorised access and potential harm to users. To enable seamless interaction with Firebase, the initialisation process of the application was imperative. This involved integrating Firebase into the system through the following code snippet:



```
let app;

if (firebase.apps.length === 0) {

    app = firebase.initializeApp(firebaseConfig);

} else {

    app = firebase.app()

}

const auth = firebase.auth()

export { auth };
```

Figure 7: Code to initialise Firebase.

Upon completion of the initialisation process, integration of the Firebase configuration into the project becomes essential. This configuration comprises the ‘`apiKey`’ property of the ‘`firebaseConfig`’ object and acts as a secret token that serves a crucial aspect in ensuring that access to Diabeticare is restricted solely to authorised users. The integration of the Firebase configuration can be seen in the code snippet below:



```
const firebaseConfig = {

  apiKey: "AIzaSyB0bEdAlAfCQHQRfbzjrLjqpKLLffK_3sY",

  authDomain: "fir-auth-a2201.firebaseapp.com",

  projectId: "fir-auth-a2201",

  storageBucket: "fir-auth-a2201.appspot.com",

  messagingSenderId: "96699052686",

  appId: "1:96699052686:web:6db683bbdc174fe3240fdf"

};
```

Figure 8: Firebase configuration.

In essence, Firebase's authentication solution is a key asset for Diabeticare, providing a high level of security and reliability that is essential in today's increasingly connected and data-driven world.

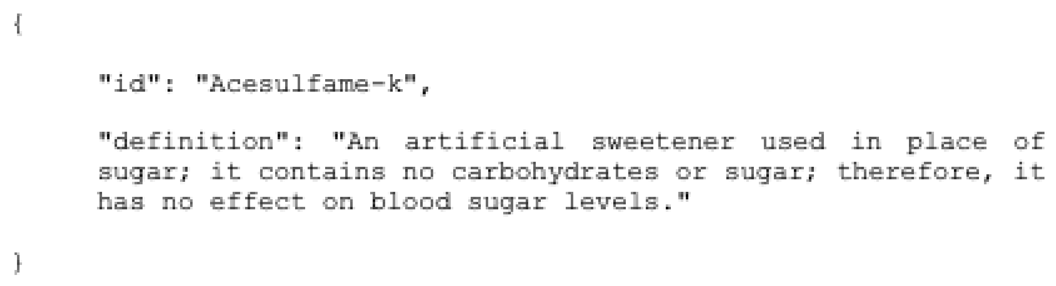
5.2.2 GiftedChat Package

The GiftedChat package is a versatile and powerful chat user interface component designed for integration within React Native applications. It is constructed using a combination of both functional and class-based components in the JavaScript programming language and offers comprehensive message data management capabilities while providing a streamlined user interface for both sending and receiving messages, making it an essential component in the development of the Diabeticare application.

Furthermore, the package offers exceptional levels of customisability, empowering developers to tailor the chat UI to the unique requirements of the Diabeticare application. With real-time messaging capabilities and an adaptable architecture, the GiftedChat package represents an ideal choice for the development of scalable and robust chat functionality.

5.2.3 wordsList.json File

The wordsList.json file is the source of terms related to diabetes, designed to facilitate user understanding and reduce any ambiguity that may arise which was an issue that was identified as a critical requirement during the research phase of the project. JSON was chosen as the storage format due to its inherent advantages, such as ease of use in reading and writing, lightweight design, and support for complex data structures and the use of the JSON format enables seamless addition or removal of terms, ensuring the continued relevance and usefulness of the dictionary feature to the Diabeticare application. Each term is issued a set of properties which include an `‘id’` and a `‘definition’.` For example, to add the word ‘Acetone’ into the JSON file it would be written in the format of the following code snippet below:



```
{

      "id": "Acesulfame-k",

      "definition": "An artificial sweetener used in place of
      sugar; it contains no carbohydrates or sugar; therefore, it
      has no effect on blood sugar levels."

}
```

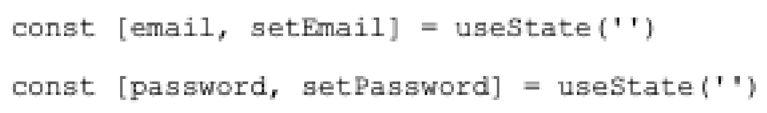
Figure 9: Code snippet depicting the addition of the term 'Acetone' to the wordsList.json file.

5.3 Development of Core Functionalities

This section will provide an in-depth examination of the essential functions necessary for the development of Diabeticare. Each feature will be thoroughly explained, with supporting code snippets that illustrate the underlying logic of the implementation process.

5.3.1 Login and Account Creation

In considering the importance of safeguarding the security and privacy of patients, it is imperative that the application grants authorisation to users before granting access to its features. Additionally, the functional requirements analysis emphasises the necessity of incorporating a login system utilising the Firebase authentication system, a service provided by Google. Failure to establish a connection between the user and Firebase will impede access to the application, as a result of either by entering incorrect login credentials, or by not having authorised access granted by the NHS, which is responsible for providing login credentials. To ensure seamless operation of the login system, the backend necessitates the declaration of two state variables: `'email'` and `'password'`, as illustrated in the following code snippet:



```
const [email, setEmail] = useState('')

const [password, setPassword] = useState('')
```

Figure 10: Code to declare the state variables.

Figure 8 depicts the declaration of two state variables using the 'useState' hook from the React Native library. The 'useState' function functions as an array that stores a pair of values - the current state, and a corresponding function that updates that state. Notably, the `'email'` and `'password'` state variables are initialised as empty strings, and the `'setEmail'` and `'setPassword'` functions are employed to update the respective state variables.

Moreover, after the receipt of the patient's email and password from the NHS for the Diabeticare administrators, the system proceeds to store this information by leveraging the Firebase Authentication Library in an effective manner. The mechanism employed to then login is illustrated below:

```
const _Login = () => {
        auth
        .signInWithEmailAndPassword(email, password)
        .then(userAuthData => {
                const user = userAuthData.user;
                console.log('Login success with email: ‘, user.email); //Output seen in terminal
})
}
```

Figure 11: Code to depict logging in functionality.

In the context of Figure 10, the `'_Login'` function is utilised to facilitate the patient login process, via the input of email and password details, and with the aid of Firebase Authentication. The `'auth'` object used within this function is a specific instance of Firebase Authentication which serves the purpose of authenticating the user before proceeding further.

The `'signInWithEmailAndPassword'` method is invoked using the email and password arguments provided, effectively verifying the input credentials against the Firebase Authentication system. Upon successful authentication, the `'.then'` method is triggered with `'userAuthData'` as the argument and `‘user’` as the object, which contains the users email for further use in the application as seen in the home page.

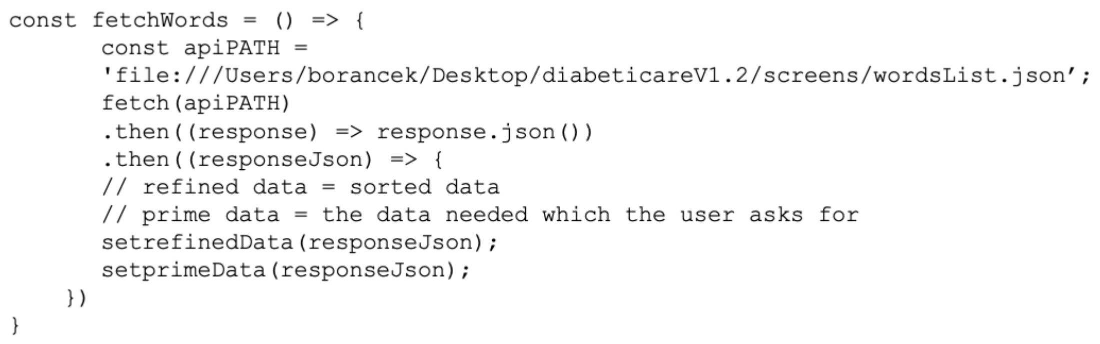
5.3.2 Dictionary Feature

To develop the dictionary component of the application, a careful process was employed to manage and refine the API data. Initially, the implementation involved the declaration of three pivotal variables essential for the optimal performance of the dictionary functionality, as demonstrated in the subsequent code below:

```
const [refinedData, setrefinedData] = useState([]);
const [primeData, setprimeData] = useState([]);
const [search, setsearch] = useState('');
```

Figure 12: Code snippet depicting the declaration of three state variabels.

As illustrated in Figure 10, `'refinedData'` and `'primeData’` are initialised as empty arrays, awaiting data input, while the `'search'` variable is initialised as an empty string, intended to store the word that the user seeks the definition of. The next aspect of the dictionary feature includes that ability to perform network requests to fetch the words list API which was explored in the previous chapter which is seen in the following code snippet:

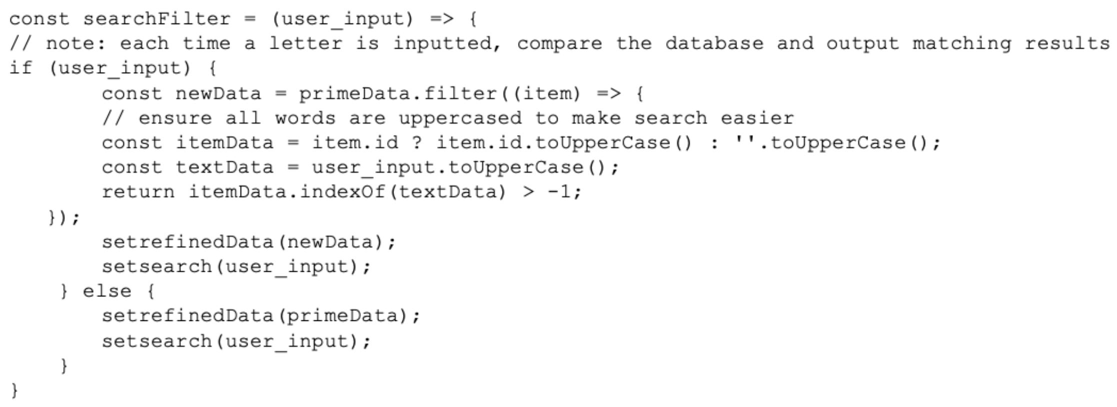


```
const fetchWords = () => {
       const apiPATH =
       'file:///Users/borancek/Desktop/diabeticareV1.2/screens/wordsList.json';
       fetch(apiPATH)
       .then((response) => response.json())
       .then((responseJson) => {
       // refined data = sorted data
       // prime data = the data needed which the user asks for
       setrefinedData(responseJson);
       setprimeData(responseJson);
    })
}
```

Figure 13: Illustrating the network request function to fetch the words list API.

The `'fetchWords'` function retrieves data from a locally stored JSON file specified within the `'apiPATH'` variable. This function employs the `'fetch(apiPATH)'` method to send a GET request to the location of the JSON file, thus acquiring the list of words. Upon receipt of the data, the `'setrefinedData'` and `'setprimeData'` fields are updated to use the state of the data variables.

Furthermore, a crucial aspect of the dictionary functionality lies in its capacity to facilitate word searches for users experiencing confusion. This search capability is intended to offer support to patients, as one of the challenges encountered by diabetic individuals discovered in the research aspect of the project, is the use of puzzling medical terminology by their general practitioners. The search function of the feature has been implemented through the `'searchFilter'` function, as illustrated in the code snippet below:



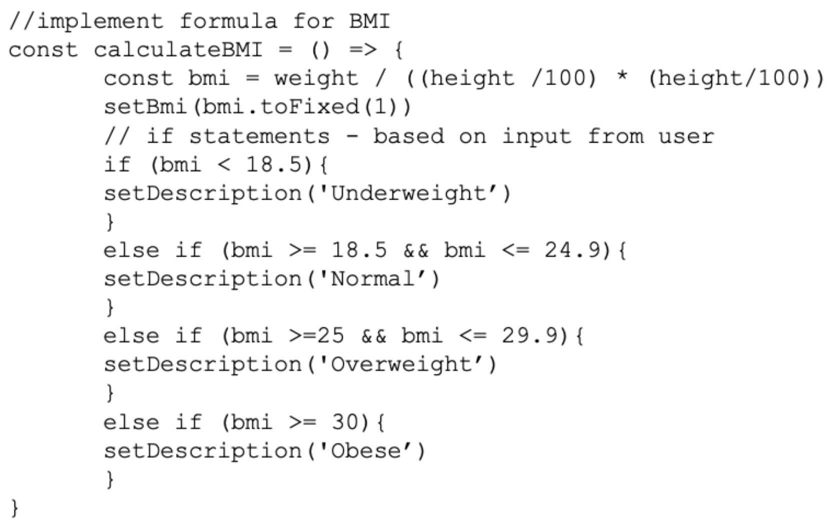
```
const searchFilter = (user_input) => {
// note: each time a letter is inputted, compare the database and output matching results
if (user_input) {
        const newData = primeData.filter((item) => {
        // ensure all words are uppercased to make search easier
        const itemData = item.id ? item.id.toUpperCase() : ''.toUpperCase();
        const textData = user_input.toUpperCase();
        return itemData.indexOf(textData) > -1;
   });
        setrefinedData(newData);
        setsearch(user_input);
    } else {
        setrefinedData(primeData);
        setsearch(user_input);
    }
}
```

Figure 14: Depicting the `'searchFilter'` function.

The function named `'searchFilter'` receives a textual input from the user and assigns it to a variable named `'user_input'`. The function aims to utilise this input to filter the data contained in a JSON file called `'wordsList'`. Specifically, the function compares each word in the file with the inputted text by filtering out the characters that do not match through the use of the `'indexOf(textData)'` method where it is used to check if `'itemData'` (converted to upper case) contains the same characters as `'textData'` (converted to upper case) by comparing their indexes. In the event that the textual input supplied by the user is absent from the data set, the `'indexOf()'` method would produce a return value of -1. Consequently, the condition established by the comparison `‘itemData.indexOf(textData) > -1’` would be evaluated as false. Hence, the words in the data set that does not meet this condition would not be included in the filtered data.

5.3.3 BMI Calculator

The procedure for constructing the BMI calculator can be elucidated through the following sequence of steps. Firstly, two state variables representing the patient's weight and height are declared and initialised as empty strings, in anticipation of user input. Specifically, the variables are denoted as follows: `'const [weight, setWeight] = useState('')'` and `'const [height, setHeight] = useState('')'`. The functions `'setWeight'` and `'setHeight'` are used to facilitate the updating of the `'weight'` and `'height'` variables, respectively. Subsequently, the computation of the user's BMI is executed through a function that adheres to a predefined set of instructions to gain the BMI value, as demonstrated below:

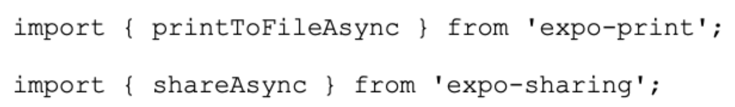


```
//implement formula for BMI
const calculateBMI = () => {
       const bmi = weight / ((height /100) * (height/100))
       setBmi(bmi.toFixed(1))
       // if statements - based on input from user
       if (bmi < 18.5){
       setDescription('Underweight’)
       }
       else if (bmi >= 18.5 && bmi <= 24.9){
       setDescription('Normal’)
       }
       else if (bmi >=25 && bmi <= 29.9){
       setDescription('Overweight’)
       }
       else if (bmi >= 30){
       setDescription('Obese’)
       }
}
```

Figure 15: Algorithm to work out the patients BMI index.

5.3.4 e-Prescription downloader in PDF

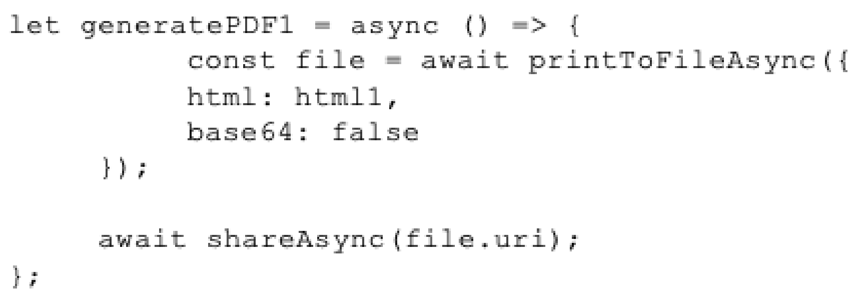
The e-Prescription component of the application encompasses the functionality that enables users to download a designated prescription dispatched by their General Practitioner. This feature accelerates the process of replenishing medication by facilitating the retrieval of a new set of medicines prescribed by the patient's General Practitioner. To execute this functionality, the e-Prescription feature must first retrieve the prescription from the NHS system, possibly through using an API, and subsequently permit the patient to download the prescription onto their device for further usage. Following the importation of the indicated files, as demonstrated in the subsequent code below:



```
import { printToFileAsync } from 'expo-print';

import { shareAsync } from 'expo-sharing';
```

Figure 16: Import statements.

The system would then seek to generate a PDF version of the prescription through the use of `'printToFileAsync'` which allows the user to print the data onto the file of their choice locally asynchronously. The `'printToFileAsync'` function would take in two arguments where the first argument represents the content that needs to be printed and secondly, the options that are available to specify as seen in the code snippet below:



```
let generatePDF1 = async () => {
            const file = await printToFileAsync({
            html: html1,
            base64: false
      });

      await shareAsync(file.uri);
};
```

Figure 17: Illustrates the method that downloads the prescription.

5.3.5 Chat Feature

As mentioned in the previous section, the chat feature of Diabeticare is supported by the use of the Gifted Chat library in order to assist in the building of the chat interface. The gifted chat along with other types of components are imported through the use of the following code:

```
import {Bubble, GiftedChat, Send, InputToolbar} from 'react-
native-gifted-chat';
```

After importing the `'GiftedChat'` library into the project, it is necessary to start with creating a function that keeps a record of the messages state that will be sent by the user. To achieve this, a function known as `'onSend'` is created, which houses an empty array of messages. This array serves as a repository for previous patient messages as well as any other prior communications, thereby enabling the construction of a comprehensive chat history. This is accomplished through the use of the `'GiftedChat.append'` method within the `'onSend'` function. By concatenating the new `'messages'` array with the pre-existing `'previousMessages'`, a single array is formed, which is then stored in the `'setMessages'` function as seen in the code snippet below:

```
const onSend = useCallback((messages = []) => {
      setMessages((previousMessages) =>
       GiftedChat.append(previousMessages, messages),
      );
}, []);
```

Figure 18: Depicting the creation of the 'onSend' function.

Furthermore, the "`renderSend`" function, which assumes responsibility for accessing the rendering of the text input and the send message button, is declared in order to provide the user with a clear understanding of the message input process and facilitate the sending of their message to the General Practitioner. The code snippet below provides the means to access and customise the send message function:

```
const renderSend = (props) => {
       return (
       // render send button
       <Send {...props}>
       <View style={{top: 0}}>
       <MaterialCommunityIcons
       // send button customisation
       name="send-circle"
       style={{marginBottom: 5, marginRight: 5 }}
       size={32}
       color="#2e64e5"
   />
```

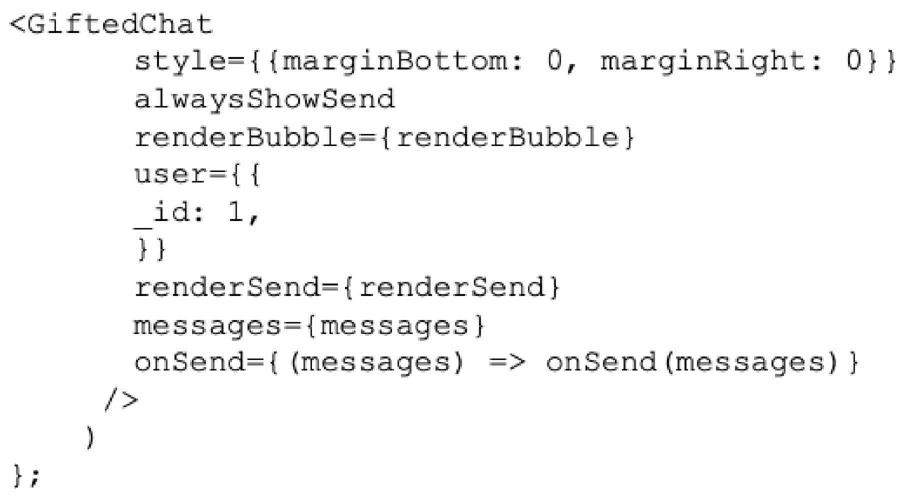
Figure 19: The implementation of the 'renderSend' function.

Furthermore, to ensure a seamless user experience, it is imperative to declare and customise the message bubbles. As such, the "`renderBubble`" function is defined to access the message and set the background colour of the bubbles to blue, thus enabling the white text to stand out, allowing the patient to read the message with utmost clarity and avoid any possible misinterpretations. The code snippet presented below shows the customisation process of the message bubbles:

```
const renderBubble = (props) => {
// render sent messages design
  return (
      <Bubble
        {...props}
      wrapperStyle={{
      //background of the already sent chat
      // blue and white = best visually
      right: {
      backgroundColor: '#2e64e5',
      },
      }}
      // render already sent chat
      textStyle={{
      right: {
       color: '#fff',
      },
     }}
   />
 );
};
```

Figure 20: Implementation of the 'renderBubble' function.

Additionally, after laying out the customisation for the chat interface to fully function, and then display itself on the screen, the GiftedChat component from the GiftedChat library must be declared as shown below:



```
<GiftedChat
        style={{marginBottom: 0, marginRight: 0}}
        alwaysShowSend
        renderBubble={renderBubble}
        user={{
        _id: 1,
        }}
        renderSend={renderSend}
        messages={messages}
        onSend={(messages) => onSend(messages)}
      />
    )
};
```

Figure 21: Declaration of the 'GiftedChat' component.

Within the declaration of the `'GiftedChat'` component, the main takeaways are:

• The `'messages'` variable used to pass an array of chat messages which will then be displayed on the interface

• The `'onSend'` variable which will be called when the patient sends a message

• The `'user'` variable to define the current user in the chat where they have an 'id' property of 1.

• The `'renderBubble'` variable being called that displays the customised chat bubbles.

• The `'alwaysShowSend'` variable being used to ensure that the send button is always visible to the user even if there is no text in the input field.

• The `'renderSend'` variable used to pass the function that uses the customised send button, referring to the 'renderSend' function mentioned earlier.

Chapter 6: **Testing**

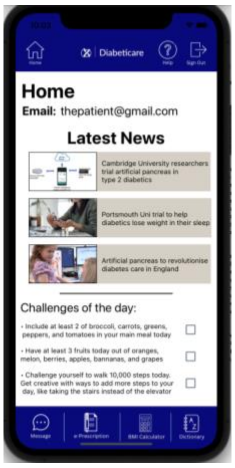
This particular section is dedicated to conducting a comprehensive evaluation of Diabeticare's features in both Blackbox and Whitebox (supported by Jest Testing Tool) testing environments. This approach is being taken to ensure that a sufficiently broad range of testing scenarios are covered, thus enhancing the overall quality of the testing outcomes.

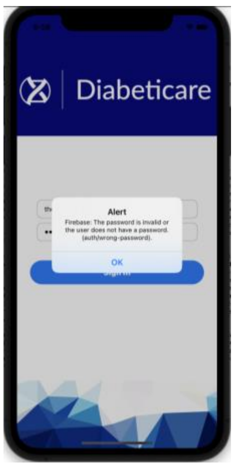
6.1 Blackbox Testing

This section focuses on the use of manual Blackbox testing where during the test, the inner workings of the application are disregarded, and diverse approaches are employed to identify potential user interface dysfunction, incorrect behavioural responses, and crashes.

6.1.1 Test Case 1: Login

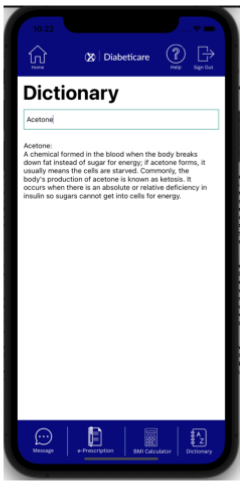
This test case is designed to assess the operational functionality of the login system, specifically focusing on how the application will respond to both correct and incorrect user credentials. The following table outlines the results of the test:

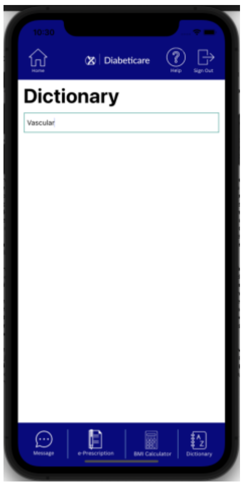
Test Case	Steps	Expected Result	Actual Result	Success / Failure
Correct credentials	1. Enter email as 'thepatient@gmail.com' 2. Enter password as 'mileend'. 3. Press the sign in button. 4. The application will verify the accuracy of the provided credentials by comparing them to the data stored within the Firebase Authentication System.	Navigation to the home page with the patients email visible	 Home Diabeticare Help Sign Out Message e-Prescription BMI Calculator DictionaryHome Email: thepatient@gmail.com Latest News Cambridge University researchers trial artificial pancreas in type 2 diabetics Portsmouth Uni trial to help diabetics lose weight in their sleep Artificial pancreas to revolutionise diabetes care in England Challenges of the day: • Include at least 2 of broccoli, carrots, greens, peppers, and tomatoes in your main meal today ☐ • Have at least 3 fruits today out of oranges, melon, berries, apples, bannanas, and grapes ☐ • Challenge yourself to walk 10,000 steps today. Get creative with ways to add more steps to your day, like taking the stairs instead of the elevator ☐	Success

Incorrect credentials	1. Enter email as 'thepatient@gmail.com' 2. Enter password as 'mileend1'. 3. Press the sign in button. 4. The application will verify the accuracy of the provided credentials by comparing them to the data stored within the Firebase Authentication System.	Alert to indicate incorrect credentials	 Alert Firebase: The password is invalid or the user does not have a password. (auth/wrong-password). OK	Success

6.1.2 Test Case 2: Search for a word in the Dictionary

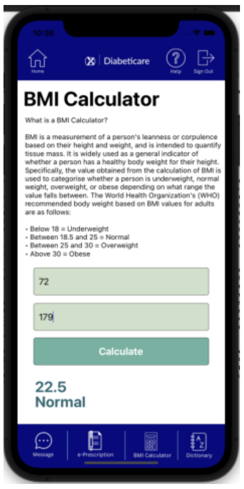
This test case is designed to assess the operational functionality of the dictionary feature, specifically focusing on how the application will respond to in recognising whether a word provided by the user is present within the system or not. The following table outlines the results of the test:

Test Case	Steps	Expected Result	Actual Result	Success / Failure
Word present in the system.	○ Navigate to the dictionary feature. ○ Enter the word 'Acetone'. ○ The application will perform a database query to determine if the provided word exists within the system. In the event that a match is found, the application will proceed to return the word and its corresponding definition.	Display the requested word and its corresponding definition on the user's screen.	 10:22 Home Diabeticare Help Sign Out Message e-Prescription BMI Calculator Dictionary Dictionary Acetone Acetone: A chemical formed in the blood when the body breaks down fat instead of sugar for energy; if acetone forms, it usually means the cells are starved. Commonly, the body's production of acetone is known as ketosis. It occurs when there is an absolute or relative deficiency in insulin so sugars cannot get into cells for energy.	Success

Word not present in the system.	○ Navigate to the dictionary feature. ○ Enter the word 'Vascular. ○ The application will perform a database query to determine if the provided word exists within the system. In the event that a match is found, the application will proceed to return the word and its corresponding definition	Requested word does not appear with its corresponding definition.	 Dictionary Vascular	Success

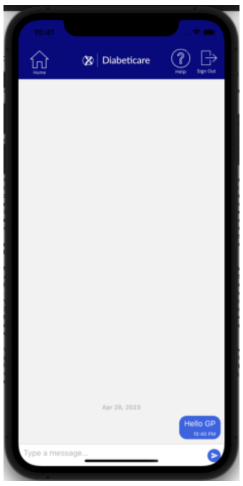
6.1.3 Test Case 3: Receive Correct BMI Number

This test case is designed to assess the operational functionality of the BMI Calculator feature, specifically focusing on how the application will respond when receiving weight and height data input from the user. The following table outlines the results of the test:

Test Case	Steps	Expected Result	Actual Result	Success / Failure
Input weight and height data	○ Navigate to BMI Calculator ○ Input weight as '72' ○ Input height as '179' ○ Press calculate	Output to be '22.5' and 'Normal'	 BMI Calculator What is a BMI Calculator? BMI is a measurement of a person's leanness or corpulence based on their height and weight, and is intended to quantify tissue mass. It is widely used as a general indicator of whether a person has a healthy body weight for their height. Specifically, the value obtained from the calculation of BMI is used to categorise whether a person is underweight, normal weight, overweight, or obese depending on what range the value falls between. The World Health Organization's (WHO) recommended body weight based on BMI values for adults are as follows: • Below 18 = Underweight • Between 18.5 and 25 = Normal • Between 25 and 30 = Overweight • Above 30 = Obese 72 179 Calculate 22.5 Normal	Success

6.1.4 Test Case 4: Send a Text Message

This test case is designed to assess the operational functionality of the chat feature, specifically focusing on how the application will respond a request by the user to send a message. The following table outlines the results of the test:

Test Case	Steps	Expected Result	Actual Result	Success / Failure
Send message	○ Navigate to the message feature. ○ Type 'Hello GP' in the text box below ○ Press the send message icon on the right side of the screen	Message sent and displayed in a blue bubble with timestamp	 10:41 Home Diabeticare Help Sign Out Type a message... Apr 28, 2023 Hello GP 10:40 PM	Success

6.2 Whitebox Testing

Whitebox testing is a rigorous software testing technique that involves examination of the internal structure and operations of an application. This testing methodology provides deep insight into the internal code, data flow, and development architecture, allowing a comprehensive understanding of the application's functioning. During the testing process, automated Unit Test cases will be designed and executed to ensure that the code is operating seamlessly and accurately. This approach to testing is highly effective in detecting defects and vulnerabilities that may not be apparent in Blackbox testing.

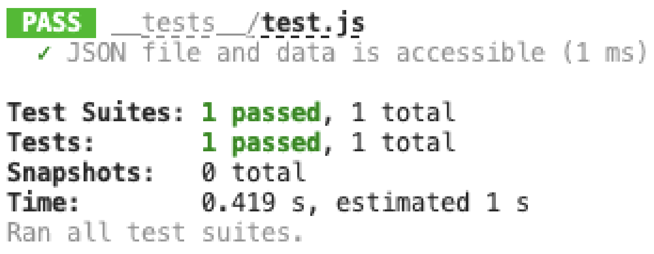
6.2.1 Test Case 1: wordsList.json file is accessible.

The aim of this test is to see if the JSON file used by the dictionary feature is able to access the file and use the data inside of that file. The test case is written as:

```
import fs from 'fs';

test('JSON file and data is accessible', () => {
        const filePath = './screens/wordsList.json’;
        const fileExists = fs.existsSync(filePath);
        expect(fileExists).toBe(true);
});
```

Figure 22: Test Case to see if JSON file is accessible.

The result of the test is as follows:



```
PASS __tests__/test.js
  ✓ JSON file and data is accessible (1 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        0.419 s, estimated 1 s
Ran all test suites.
```

Figure 23: Result of Test Case 1

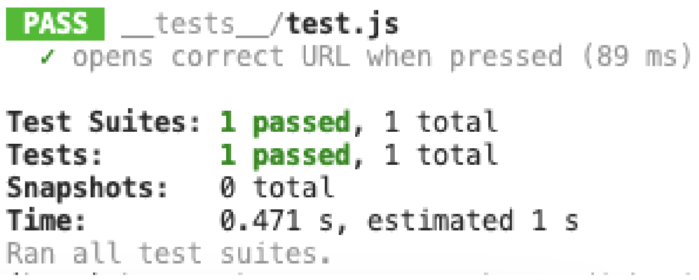
6.2.2 Test Case 2: <TouchableHighlight> component.

This test was done by extracting the touchable highlight component from the HomeScreen.js file and running it through the test that checks to see if when the component is pressed, the page navigates to the external page that it is supposed to go to, essentially ensuring that it is clickable. The test case is written as:

```
import React from 'react';
import { render, fireEvent } from '@testing-library/react-native';
import { TouchableHighlight, Linking, Text } from 'react-native';

test('opens correct URL when pressed', () => {
       const openURLMock = jest.spyOn(Linking, 'openURL');
       const { getByText } = render(
       <TouchableHighlight onPress={() =>
       Linking.openURL('https://www.bbc.co.uk/news/uk-england-cambridgeshire-
       64318194')}>
       <Text></Text>
       </TouchableHighlight>
);

fireEvent.press(getByText(''));

       expect(openURLMock).toHaveBeenCalledWith('https://www.bbc.co.uk/news/uk-
       england-cambridgeshire-64318194');
});
```

Figure 24: Testing code for test case 2

The result of the test is as follows:



```
PASS __tests__/test.js
  ✓ opens correct URL when pressed (89 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        0.471 s, estimated 1 s
Ran all test suites.
```

Figure 25: Result of test case 2

Chapter 7: **Evaluation**

This segment of the report will concentrate on the comprehensive assessment of the project, specifically delving into the successful and unsuccessful components. Additionally, we will examine the extent to which the project aligned with its predetermined goals and objectives established earlier in the academic year.

7.1 What Was Successful?

This project has achieved numerous successes as a result of the immense dedication and hard work invested in achieving its aims and objectives (as outlined in Appendix D). The original plan entailed constructing an application built on a foundation of technologies such as React Native, node.js, and npm. React Native was employed to facilitate the development of the application, node.js was used to execute JavaScript code on the server-side, while npm was employed as the primary package manager, enabling the project to possess unique and interactive features. Notably, the project strictly adhered to this objective, which undoubtedly ranks as one of its greatest successes.

Moreover, the project's capacity to analyse existing solutions proved instrumental in identifying a market gap, which in turn allowed concentration on features that were absent in comparable applications. This market gap analysis was crucial in informing the project's strategy for implementing unique features that would differentiate it from its competitors. Specifically, the identification of a gap in the market led to the development of Diabeticare's most exciting feature - a chat feature that enables diabetic patients to communicate with general practitioners in real-time. Additionally, the project's capability to attain a substantial level of expertise on the research conducted in the field of diabetes management stands as another remarkable accomplishment, as outlined in the literature review section of the report. This accomplishment enabled us to conduct a comprehensive examination of existing research through the utilisation of articles, newspapers, and other studies, leading to a profound understanding of the domain and a distinctive viewpoint on how to make the application a success as much as possible.

Another noteworthy accomplishment of the project is its ability to showcase the immense potential of technology in aiding everyday problems. Through this project, we were able to identify issues such as inadequate communication channels between patients and healthcare professionals. By leveraging the power of technology, an innovative solution was developed that enables patients to receive assistance in overcoming potential communication barriers that they may face in their relationship with their healthcare provider. This application serves as a prime example of how technology can be harnessed to effectively address the challenges faced by vulnerable populations, thus opening up opportunities to explore other ways in which technology can be used to improve people's lives.

7.2 What Was Not Successful?

As is customary with any project, successes are often coupled with areas that have the potential for improvement. One such potential improvement would be to conduct primary research in order to gain a more comprehensive understanding of the positive and negative elements of the application. In addition, primary research can provide us with valuable data from the target audience themselves, which could have been useful during the development stages of the application. While relying on secondary research was beneficial to some extent, it resulted in a limited understanding of the subject matter at hand, as secondary sources may not cover all aspects of diabetes in sufficient depth. Moreover, secondary sources of information are often prone to bias, as they may be influenced by the interpretations and opinions of the researcher, rather than the original data, resulting in insufficient quality of information.

One aspect of the project that did not meet expectations was the inability to fulfil a particular requirement, which resulted in rework and deviation from the original objective. Specifically, the aim to construct an application that allowed users to log in exclusively through Google OAuth proved to be unattainable. After implementing this feature, it became clear that testing users locally on development mode was not possible due to the requirement of a live server for Google OAuth to function. This created complexities and prolonged the development process and in response, a login system supported by Firebase was implemented as an alternative, which did not necessitate a live server and enabled more efficient testing during development.

Additionally, the project's time management was identified as an area of potential improvement. While the project was able to meet its targets and objectives within the allotted time frame, other coursework and assignments in concurrent modules demanded a substantial amount of attention, which had a negative impact on productivity and increased stress levels. Additionally, in order to adhere to the project's schedule, overtime work was conducted on multiple occasions, which resulted in burnout and reduced overall efficiency.

7.3 Limitations

One limitation of Diabeticare is its lack of a chat limiter in the messaging feature, which may lead to negative user experiences. As users can send messages without any control, the application may become inundated with unwanted or excessive messages, impeding effective communication. Moreover, this limitation may create a hostile and unsafe environment for healthcare providers, potentially leading to harassment or bullying. To address this limitation, the implementation of a chat limiter feature and effective moderation would be beneficial for future versions of the application.

Another limitation of Diabeticare is its language support, as the application is currently only available in English. This limitation may exclude diabetic patients who speak other languages, limiting the application's effectiveness and reducing its potential user base. To overcome this limitation, the application must provide support for multiple languages in the future. Doing so would increase accessibility and ensure that diabetic patients worldwide can fully benefit from the application's features.

Chapter 8: **Legal, Social, and Commercial Issues**

The construction of Diabeticare may face a variety of legal, social, and commercial issues which must be taken into consideration throughout the life of the application. The issues that may arise are as follows:

Regulatory Compliance – Ensuring regulatory compliance is a crucial aspect of any telemedicine application, including Diabeticare, to ensure user safety and protection of personal data. Compliance with various regulations differs across different regions of the world. For instance, in the European Union, Diabeticare must adhere to the General Data Protection Regulation (GDPR), (Intersoft Consulting, 2018), which emphasises the safe storage and use of personal data. Similarly, in the United States, compliance with the Health Insurance Portability and Accountability Act (HIPAA) is mandatory to safeguard sensitive patient health information. Failure to comply with these regulations may result in legal penalties, loss of user trust, and damage to the reputation of the application. Therefore, it is essential for Diabeticare to be following these regulations to maintain a safe and trustworthy platform for its users.

Privacy Concerns – The chat feature in Diabeticare plays a vital role in facilitating communication between healthcare providers and patients, and as such, it may involve the sharing of sensitive personal and health-related information. It is crucial to ensure that such information is adequately protected to comply with privacy regulations. In the United States, the Electronic Communications Privacy Act (ECPA), (Bureau of Justice Assistance, 1986), outlines that any personal health information transmitted through the chat is protected from interception or unauthorised access. As such, Diabeticare must ensure that the chat feature is designed and implemented in a way that fully complies with the provisions of the ECPA and other relevant privacy regulations. This may involve implementing measures such as end-to-end encryption, user authentication, and access controls to safeguard the confidentiality, integrity, and availability of the sensitive information being shared through the chat feature.

Security Concerns – The e-Prescription feature in Diabeticare emphasises the criticality of protecting patient's personal and health-related information to ensure compliance with privacy regulations. The implementation of the e-Prescription feature poses concerns for potential identity theft, hacking, or data breaches, underscoring the need for robust security measures such as two-factor authentication and firewalls in future updates of the application to safeguard patient data. Maintaining the confidentiality and integrity of sensitive patient information must be a top priority to instil trust and confidence in the users of Diabeticare.

Chapter 9: **Conclusions**

9.1 Future Work

Looking ahead to the future of Diabeticare, enhancing the application's functionality and user experience is a top priority. To begin with, expanding the accessibility of Diabeticare to Android users is an essential step as current data reveals that older adults tend to use Android devices more than iOS devices, as suggested by Taylor (Taylor, 2019), making it a worthwhile effort to extend the application's reach to this demographic.

Furthermore, Diabeticare could potentially leverage additional security measures to bolster its security framework. A viable solution to achieve this objective could be the incorporation of Google and Apple ID as a means of user authentication. By implementing this approach, the application can establish an additional layer of security, guaranteeing that only authorised users are granted access to the application's functionalities and sensitive data. Moreover, exploring the possibility of integrating biometric authentication methods, such as fingerprint or facial recognition, would further enhance the application's security system.

Furthermore, in order to fully understand the needs and preferences of the target audience and ensure that the future enhancements to Diabeticare meets its expectations, primary research should be carried out. This may involve administering surveys, conducting interviews, or organising focus groups with people who have diabetes, healthcare professionals, and other relevant stakeholders to solicit feedback on the current version of Diabeticare, and obtain insights into possible improvements suggested by the patients themselves. By gathering data from the target audience, we can acquire a better understanding of which features are most significant and how they can be implemented in a manner that is most effective and user friendly. Additionally, primary research can assist in identifying potential obstacles or hurdles that must be addressed prior to implementing future enhancements to the application.

Lastly, the application could be improved by integrating other noteworthy features, such as including a glucose reader, which is a feature that did not make it to the final version of the application due to the significant amount of time constraints final year of university has caused. This feature would enable users to conveniently monitor their blood glucose levels, providing them with invaluable information about their health status and empowering them to make informed decisions regarding their diet and medication. Moreover, there are other potential features that could be incorporated, such as the capability to monitor exercise and physical activity, set reminders for medication or doctor's appointments where the application can also offer users the ability to integrate the appointments they receive via messages onto their mobile phone calendars.

In summary, the future work mentioned in this section can greatly improve the user experience which is a major priority for Diabeticare to accomplish. By implementing these enhancements to the application, individuals with diabetes can be better equipped to manage their condition and improve their overall health outcomes.

9.2 Summary

In conclusion, Diabeticare has successfully achieved its primary objectives and enabled me to gain a deeper understanding of the diabetes field and the daily challenges faced by the community of diabetic patients. Through thorough background research, I was able to identify crucial issues such as inadequate communication channels and ambiguous terminology that patients face, which in turn allowed for the development of an effective solution to assist patients. The resulting Diabeticare application therefore exemplifies the significant potential that technological advancements can have in enhancing the quality of life for many individuals. This project serves as compelling evidence that when technology is used correctly, it can deliver remarkable benefits that have a profound impact on a large population.

Thanks to the use of effective development methodologies and a comprehensive analysis of project requirements, I am confident that the development of Diabeticare has been a success. The project was able to accomplish its intended outcomes and primary objectives within the predetermined timeline.

References

Broom, D. (2020). The silent epidemic that is three times as deadly as COVID. [online] World Economic Forum. Available at: https://www.weforum.org/agenda/2020/12/diabetes-silent-epidemic-world-health.

Saeedi, P., Salpea, P., Karuranga, S., Petersohn, I., Malanda, B., Gregg, E.W., Unwin, N., Wild, S.H. and Williams, R. (2020). Mortality attributable to diabetes in 20-79 years old adults, 2019 estimates: results from the International Diabetes Federation Diabetes Atlas, 9th edition. Diabetes Research and Clinical Practice, 162, p.108086. doi:10.1016/j.diabres.2020.108086. Available at: https://www.sciencedirect.com/science/article/pii/S016882272030139X#f0010

Kessels, R.P.C. (2003). Patients' memory for medical information. Journal of the Royal Society of Medicine, [online] 96(5), pp.219–22. doi:10.1258/jrsm.96.5.219. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC539473/

Diabetes UK. (2015). *42% of people with Type 2 diabetes 'not confident managing their condition'*. [online] Available at: https://www.diabetes.org.uk/about_us/news/42-percent-of-people-with-type-2-diabetes-not-confident-managing-their-condition-#:~:text=managing%20their%20condition-.

Bonoto, B.C., de Araújo, V.E., Godói, I.P., de Lemos, L.L.P., Godman, B., Bennie, M., Diniz, L.M. and Junior, A.A.G. (2017). Efficacy of Mobile Apps to Support the Care of Patients With Diabetes Mellitus: A Systematic Review and Meta-Analysis of Randomized Controlled Trials. *JMIR mHealth and uHealth*, [online] 5(3), p.e4. doi:10.2196/mhealth.6309. Available at: https://mhealth.jmir.org/2017/3/e4/

Tiwary, A., Rimal, A., Paudyal, B., Sigdel, K.R. and Basnyat, B. (2019). Poor communication by health care professionals may lead to life-threatening complications: examples from two case reports. *Wellcome Open Research*, [online] 4(1). doi:10.12688/wellcomeopenres.15042.1.

Joint Commission International (2018). *Communicating Clearly and Effectively to Patients How to Overcome Common Communication Challenges in Health Care*. [online] Available at: https://store.jointcommissioninternational.org/assets/3/7/jci-wp-communicating-clearly-final_(1).pdf.

Alotaibi, M.M., Istepanian, R.S.H., Sungoor, A. and Philip, N. (2014). An intelligent mobile diabetes management and educational system for Saudi Arabia: System architecture. [online] IEEE Xplore. doi:10.1109/BHI.2014.6864296. Available at: https://ieeexplore.ieee.org/document/6864296

Safavi, K. and Brian Kalis (2020). MAKE RECENT DIGITAL HEALTH GAINS LAST? [online] Available at: https://www.accenture.com/_acnmedia/PDF-130/Accenture-2020-Digital-Health-Consumer-Survey-US.pdf#zoom=40.

Ahn, D.T. and Stahl, R. (2019). Is There an App for That? The Pros and Cons of Diabetes Smartphone Apps and How to Integrate Them Into Clinical Practice. Diabetes Spectrum, [online] 32(3), pp.231–236. doi:10.2337/ds18-0101.

Intersoft Consulting (2018). Personal Data | General Data Protection Regulation (GDPR). [online] General Data Protection Regulation (GDPR). Available at: https://gdpr-info.eu/issues/personal-data/.

Bureau of Justice Assistance (1986). Electronic Communications Privacy Act of 1986 (ECPA). [online] Bureau of Justice Assistance. Available at: https://bja.ojp.gov/program/it/privacy-civil-liberties/authorities/statutes/1285.

Jestjs.io. (2017). Jest · Delightful JavaScript Testing. [online] Available at: https://jestjs.io/.

Apple (2019). App Store Review Guidelines - Apple Developer. [online] Apple.com. Available at: https://developer.apple.com/app-store/review/guidelines/.

Taylor, P. (2019). Smartphone OS by age 2019. [online] Statista. Available at: https://www.statista.com/statistics/1133193/smartphone-os-by-age/.

Appendix

Appendix A - Developer Risk Assessment

Description of Risk	**Description of Impact**	**Likelihood Rating**	**Impact Rating**	**Preventative Actions**
Poor time management	Missed deadlines and missed meetings	Low	High	Conduct weekly checkpoints individually to assess if targets are being met. Also conduct regular meetings with supervisor
Loss of project	No work to submit resulting in a poor grade	Low	High	Create project back up files
Constant change in technologies	Time consuming in terms of adapting to new technologies	Medium	Medium	Analyse thoroughly which coding languages/technologies best fits my project
Poor communication with supervisor	Increased rates of ambiguity and not knowing what to do	Low	High	Conduct bi-weekly meetings with supervisor
Illness	Will cause delays	Medium	High	Brief supervisor and catch up after recovery
Insufficient testing	A poorly working application would ruin chances of high marks	Low	High	Conduct software testing techniques thoroughly to ensure the application runs smoothly

Poor project management	Increasing in confusion and would be difficult to stay on top of tasks	Low	High	Strictly follow the Gantt chart to see what is expected each week
Ethical issues	Reduced marks and application will not be able to be published	Low	High	Ensure project guidelines are adhered to at all times
Ineffective authentication during login	Higher risk of security concerns	Low	High	Ensure thorough testing of the authentication feature before deployment
Poorly functioning messaging service	Lower rates of app interaction	Low	High	Ensure that the code for messaging is tested fully
Incorrect information extracted from references	Inaccurate data gathered resulting in incorrect evaluations	Medium	High	Check that information being extracted from sources are reliable
NHS reject partnership	The messaging service will not work and users will lose trust	Medium	High	Provide detailed explanation and convince NHS as much as possible to ensure a partnership is established
iOS Developer license is rejected by Apple	Application will be unable to be published on the App store	Medium	High	Ensure the application is built to meet Apple's expected standards
Data breach / cyber attack	User information	High	High	Ensure that all passwords set are

	would be stolen and privacy will be breached, resulting in losing users trust			strong and meet the requirements set by the administrator

Figure 26: Developer risk assessment

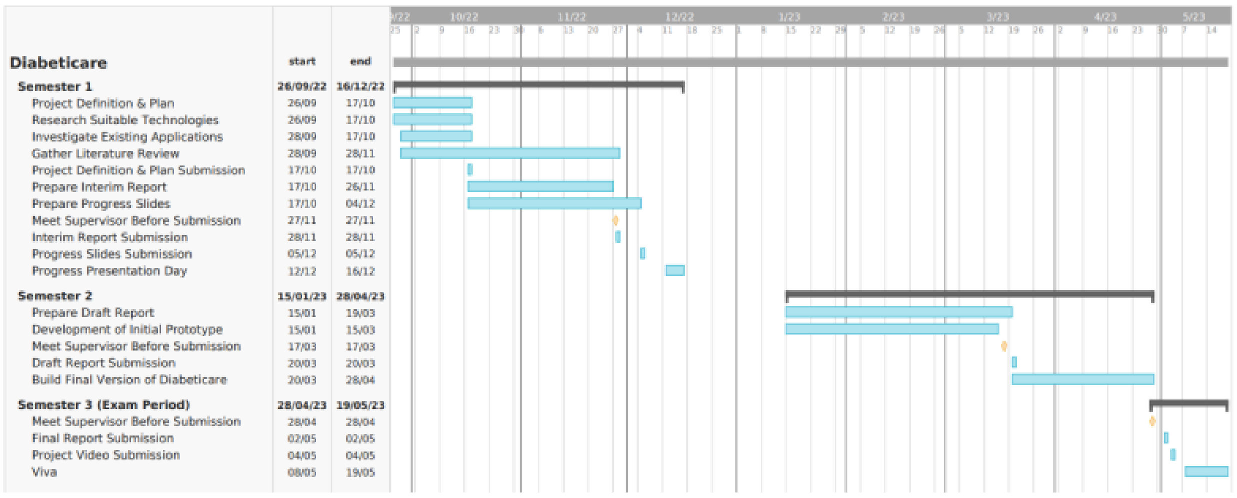
Appendix B - User Risk Assessment

Description of Risk	**Description of Impact**	**Likelihood Rating**	**Impact Rating**	**Preventative Actions**
Receiving incorrect definition of words from glossary	Providing false information for users can be the difference between life and death	Low	High	Ensure that all words are checked against a verified dictionary
e-Prescriptions do not work	User will not trust our application and may not return	Low	High	Ensure that accessing PDF prescriptions are tested
Chat limiter is ineffective	Large amounts of messages may be annoying for doctors, resulting in doctors not wanting to use the app	Low	High	Ensure that the chat limiter is tested
The application is not engaging enough	User retention will suffer	Medium	High	User feedback must be listened to, and new updates

				should match user expectations
Constant downtime	Users would see that the application is not reliable enough resulting in them pursuing alternative applications	Medium	High	Downtime should only happen during hours that are least busy, and users must be warned in advance of expected downtime
Ethical concerns are not dealt with	Users will lose trust in the application, may even be fined by UK Government	Low	High	Ensure that all rules and guidance towards ethics are carefully adhered to
Use of profanity	Doctors may receive abuse and treated unfairly	Medium	High	Ensure that certain words that are offensive are banned from being used
The application consumes too much battery power	Users may refrain from wanting to use the application	Medium	Medium	Ensure that the application is quick and efficient as much as possible

Figure 27: User Risk Assessment

Appendix C - Time Plan Gantt Chart



Diabeticare	start	end
Semester 1	**26/09/22**	**16/12/22**
Project Definition & Plan	26/09	17/10
Research Suitable Technologies	26/09	17/10
Investigate Existing Applications	28/09	17/10
Gather Literature Review	28/09	28/11
Project Definition & Plan Submission	17/10	17/10
Prepare Interim Report	17/10	26/11
Prepare Progress Slides	17/10	04/12
Meet Supervisor Before Submission	27/11	27/11
Interim Report Submission	28/11	28/11
Progress Slides Submission	05/12	05/12
Progress Presentation Day	12/12	16/12
Semester 2	**15/01/23**	**28/04/23**
Prepare Draft Report	15/01	19/03
Development of Initial Prototype	15/01	15/03
Meet Supervisor Before Submission	17/03	17/03
Draft Report Submission	20/03	20/03
Build Final Version of Diabeticare	20/03	28/04
Semester 3 (Exam Period)	**28/04/23**	**19/05/23**
Meet Supervisor Before Submission	28/04	28/04
Final Report Submission	02/05	02/05
Project Video Submission	04/05	04/05
Viva	08/05	19/05

Figure 28: Time plan of Diabeticare in Gantt Chart Format

Appendix D – Ticking Against Objectives

Objective	**Success / Fail**
Have a functioning mobile application possibly by using React, Node.js and npm	Success
To critically assess and have an accurate understanding of already built solutions and identify their benefits and drawbacks	Success
Reach a respectable level of knowledge of the research conducted in the area of diabetes management through articles, newspapers and/or journals	Success
To develop a mobile application that allows diabetes patients to manage their condition & message their General Practitioner (GP)	Success
Conduct primary research to gain rich and insightful pieces of data	Fail

(interviews, surveys, questionnaires and/or focus groups)	
Conduct SWOT (Strengths, Weaknesses, Opportunities, and Threats) analysis on applications already in existence	Success
To include a simple to use interface that is comfortable for all users	Success
To act in accordance with the rules and regulations set out for the project	Success
Execute thorough software testing procedures such as Unit and Integration testing to ensure the expected functionality of the application is observed	Success
Have a fully functional login authentication system by using Google OAuth	Fail
Implement a gamification element to boost app usage. For example, if the user engages with the application for 7 days in a row, then they are entered into a lottery competition	Success
Conduct agile project management techniques to optimise the implementation process and allow workflow of the project to run smoothly as much as possible	Success

Figure 29: Ticking against objectives.

Appendix E – Non-Functional Requirements

ID	Requirement Description
1	The system must be engaging for the user. Reviews for the application should be above 3.5 stars
2	The system must be extensible as new features may be added by the system administrator
3	The system must run on iOS device. Android implementation should come after
4	The system must log the user in as quickly as possible, ideally within 5 seconds
5	Rate of system failure must be below 3% over each quarter of the year
6	The system must be able to recover from downtime within 5 hours
7	The system must be available for maintenance from 23:00 to 03:00 every other day
8	The application should load within 4 seconds if the active user count is higher than 1500 at any given time
9	System administrator must create a username for a user that contains at least 5 characters
10	The system administrator must create a password for a user that contains at least 8 characters
11	The system should be available in at least 3 different languages

Figure 30: Non-Functional Requirements

Appendix F – Functional Requirements

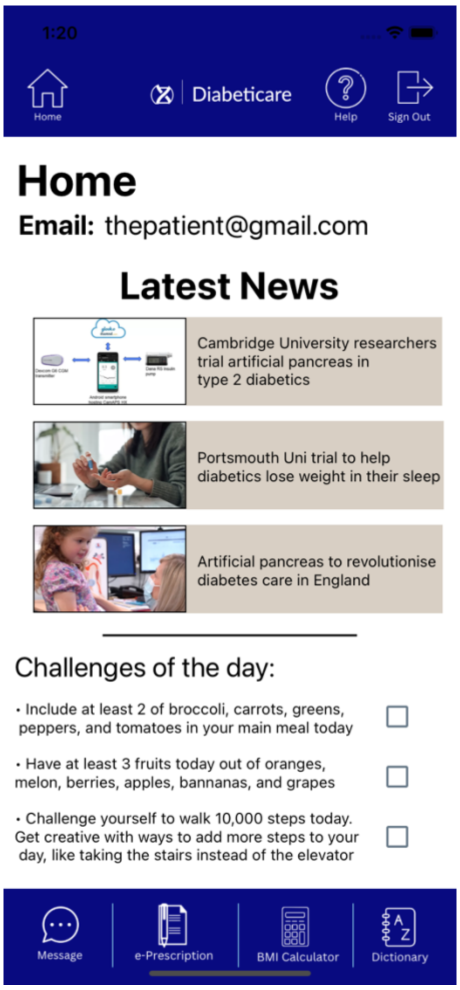
12	Users must be able to login with a username and password
13	The system must have three types of users: patients, healthcare professionals, system administrators

14	The system must be able to identify different types of users which will be reflected on the actions that they can pursue
15	The system administrator must be able to log in
16	Admin must be able to enter patients who use the system for six days in a row in a lottery game
17	The system administrator must be able to allocate a username for all users
18	The system administrator must be able to allocate a password for all users
19	The system administrator must be able to undergo maintenance with regards to the application
20	The system administrator must be able to add new features
21	The system administrator must be able to delete a user from the database
22	The system administrator must be able to update the user’s username
23	The system administrator must be able to update the user’s password
24	Users must be able to log in using the details (username and password) provided by the system administrator.
25	Patients must be able to view and send messages, given that the chat limit is not met
26	Patients must be able to view their e-Prescriptions
27	Patients must be able to view their notes
28	Patients must be able to view the glossary
29	The system must be able to use Google OAuth to verify the users log in details
30	The system must use the React framework that enables interaction within the application

31	The system must use Firebase as the database storage in order to store and retrieve messages between patients and doctors
32	The system must use node.js as the back-end runtime environment
33	The system must use npm as the default package manager

Figure 31: Functional Requirements

Appendix G – Home Page



1:20

Home | Diabeticare | Help | Sign Out

Home

Email: thepatient@gmail.com

Latest News

Cambridge University researchers trial artificial pancreas in type 2 diabetics

Portsmouth Uni trial to help diabetics lose weight in their sleep

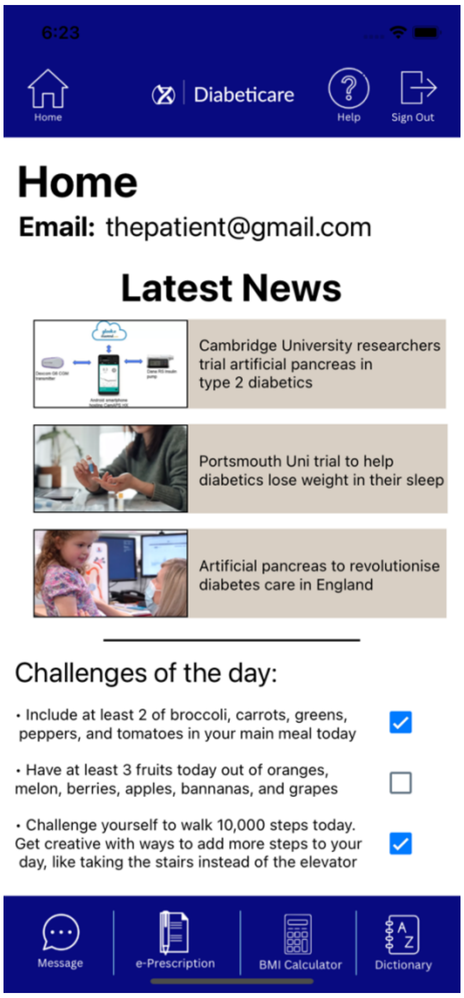
Artificial pancreas to revolutionise diabetes care in England

Challenges of the day:

- Include at least 2 of broccoli, carrots, greens, peppers, and tomatoes in your main meal today ☐
- Have at least 3 fruits today out of oranges, melon, berries, apples, bannanas, and grapes ☐
- Challenge yourself to walk 10,000 steps today. Get creative with ways to add more steps to your day, like taking the stairs instead of the elevator ☐

Message | e-Prescription | BMI Calculator | Dictionary

Figure 32: Home page.



6:23

Home | Diabeticare | Help | Sign Out

Home

Email: thepatient@gmail.com

Latest News

Cambridge University researchers trial artificial pancreas in type 2 diabetics

Portsmouth Uni trial to help diabetics lose weight in their sleep

Artificial pancreas to revolutionise diabetes care in England

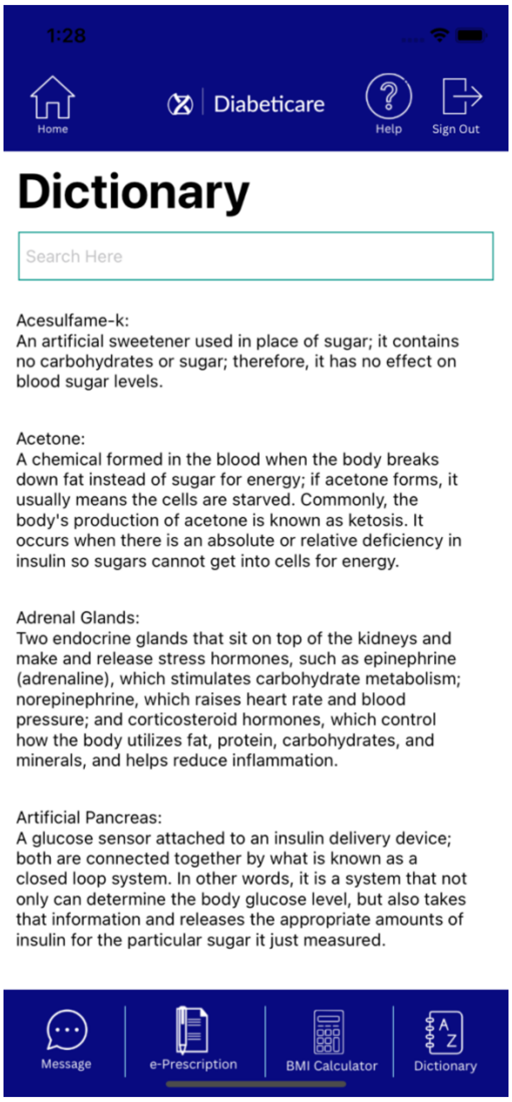
Challenges of the day:

- Include at least 2 of broccoli, carrots, greens, peppers, and tomatoes in your main meal today ☑
- Have at least 3 fruits today out of oranges, melon, berries, apples, bannanas, and grapes ☐
- Challenge yourself to walk 10,000 steps today. Get creative with ways to add more steps to your day, like taking the stairs instead of the elevator ☑

Message | e-Prescription | BMI Calculator | Dictionary

Figure 33: Home page with challenges ticked.

Appendix H – Dictionary Page



Dictionary

Search Here

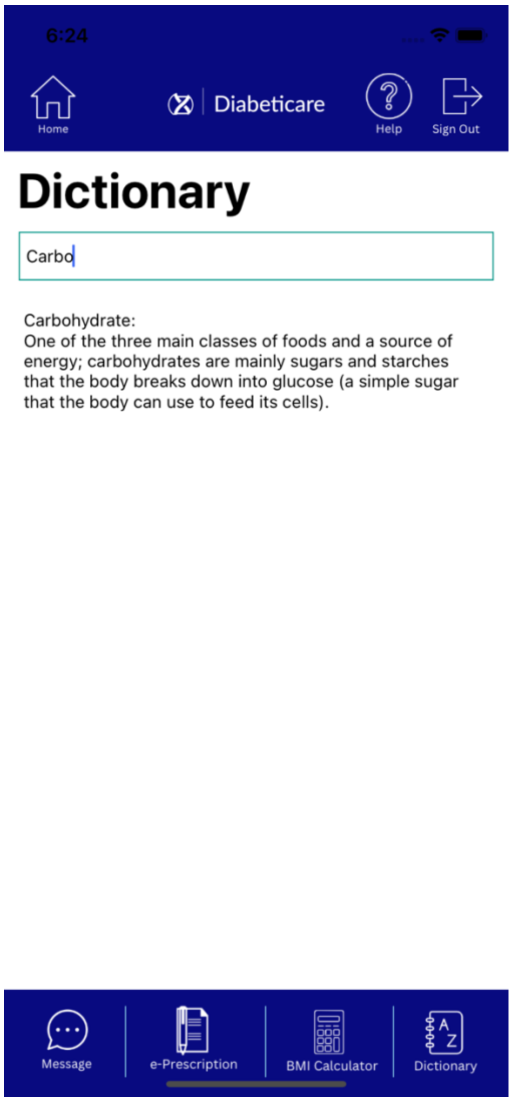
Acesulfame-k:
An artificial sweetener used in place of sugar; it contains no carbohydrates or sugar; therefore, it has no effect on blood sugar levels.

Acetone:
A chemical formed in the blood when the body breaks down fat instead of sugar for energy; if acetone forms, it usually means the cells are starved. Commonly, the body's production of acetone is known as ketosis. It occurs when there is an absolute or relative deficiency in insulin so sugars cannot get into cells for energy.

Adrenal Glands:
Two endocrine glands that sit on top of the kidneys and make and release stress hormones, such as epinephrine (adrenaline), which stimulates carbohydrate metabolism; norepinephrine, which raises heart rate and blood pressure; and corticosteroid hormones, which control how the body utilizes fat, protein, carbohydrates, and minerals, and helps reduce inflammation.

Artificial Pancreas:
A glucose sensor attached to an insulin delivery device; both are connected together by what is known as a closed loop system. In other words, it is a system that not only can determine the body glucose level, but also takes that information and releases the appropriate amounts of insulin for the particular sugar it just measured.

Figure 35: Dictionary page.



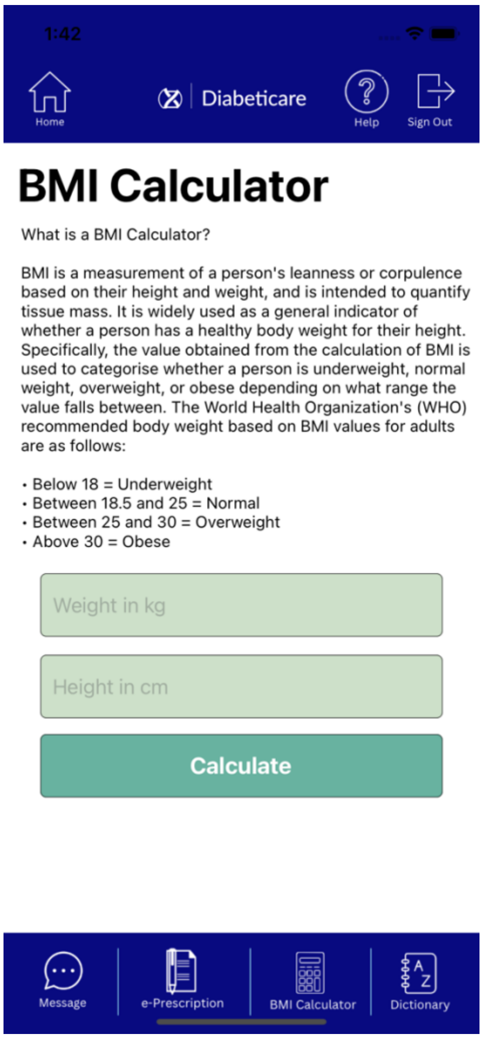
Dictionary

Carbo

Carbohydrate:
One of the three main classes of foods and a source of energy; carbohydrates are mainly sugars and starches that the body breaks down into glucose (a simple sugar that the body can use to feed its cells).

Figure 34: Dictionary page with word being searched.

Appendix H – BMI Calculator Page



BMI Calculator

What is a BMI Calculator?

BMI is a measurement of a person's leanness or corpulence based on their height and weight, and is intended to quantify tissue mass. It is widely used as a general indicator of whether a person has a healthy body weight for their height. Specifically, the value obtained from the calculation of BMI is used to categorise whether a person is underweight, normal weight, overweight, or obese depending on what range the value falls between. The World Health Organization's (WHO) recommended body weight based on BMI values for adults are as follows:

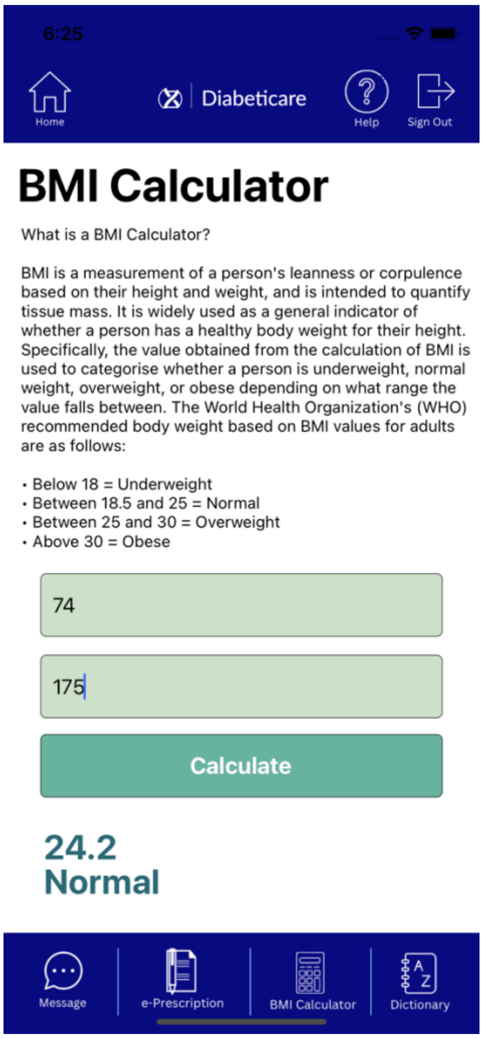
- Below 18 = Underweight
- Between 18.5 and 25 = Normal
- Between 25 and 30 = Overweight
- Above 30 = Obese

Weight in kg

Height in cm

Calculate

Figure 37: BMI calculator page.



BMI Calculator

What is a BMI Calculator?

BMI is a measurement of a person's leanness or corpulence based on their height and weight, and is intended to quantify tissue mass. It is widely used as a general indicator of whether a person has a healthy body weight for their height. Specifically, the value obtained from the calculation of BMI is used to categorise whether a person is underweight, normal weight, overweight, or obese depending on what range the value falls between. The World Health Organization's (WHO) recommended body weight based on BMI values for adults are as follows:

- Below 18 = Underweight
- Between 18.5 and 25 = Normal
- Between 25 and 30 = Overweight
- Above 30 = Obese

74

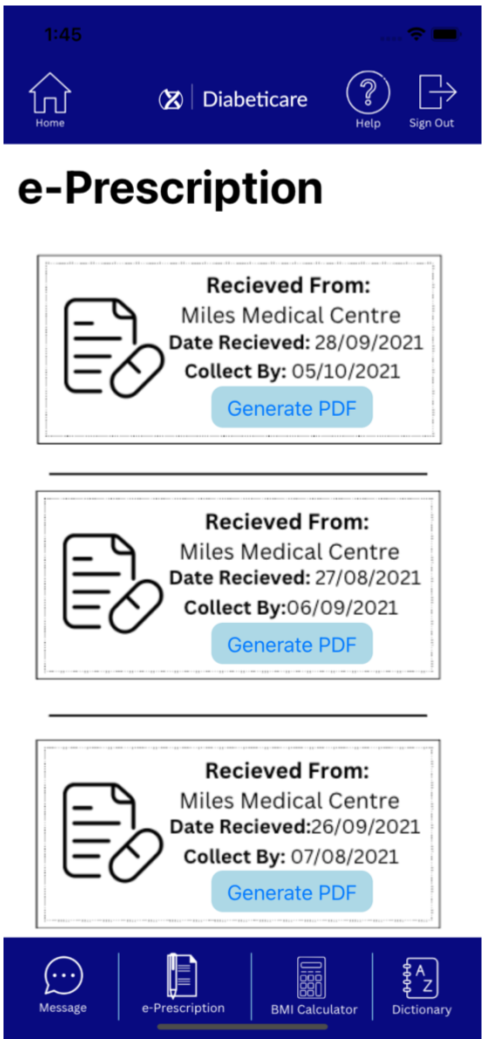
175

Calculate

24.2
Normal

Figure 36: BMI calculator with user input.

Appendix I – e-Prescription Page



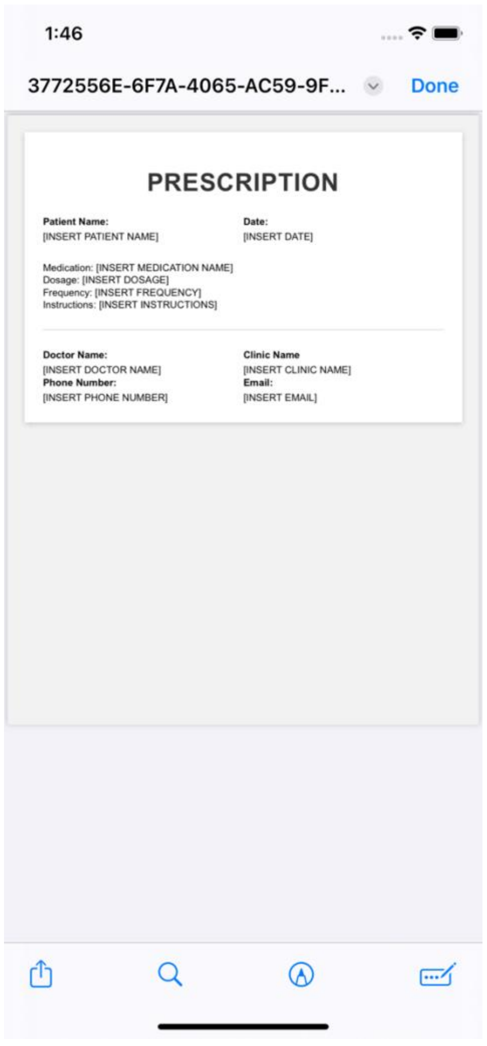
e-Prescription

Recieved From:
Miles Medical Centre
Date Recieved: 28/09/2021
Collect By: 05/10/2021
Generate PDF

Recieved From:
Miles Medical Centre
Date Recieved: 27/08/2021
Collect By:06/09/2021
Generate PDF

Recieved From:
Miles Medical Centre
Date Recieved:26/09/2021
Collect By: 07/08/2021
Generate PDF

Figure 39: e-Prescription page.



PRESCRIPTION

Patient Name:
[INSERT PATIENT NAME]

Date:
[INSERT DATE]

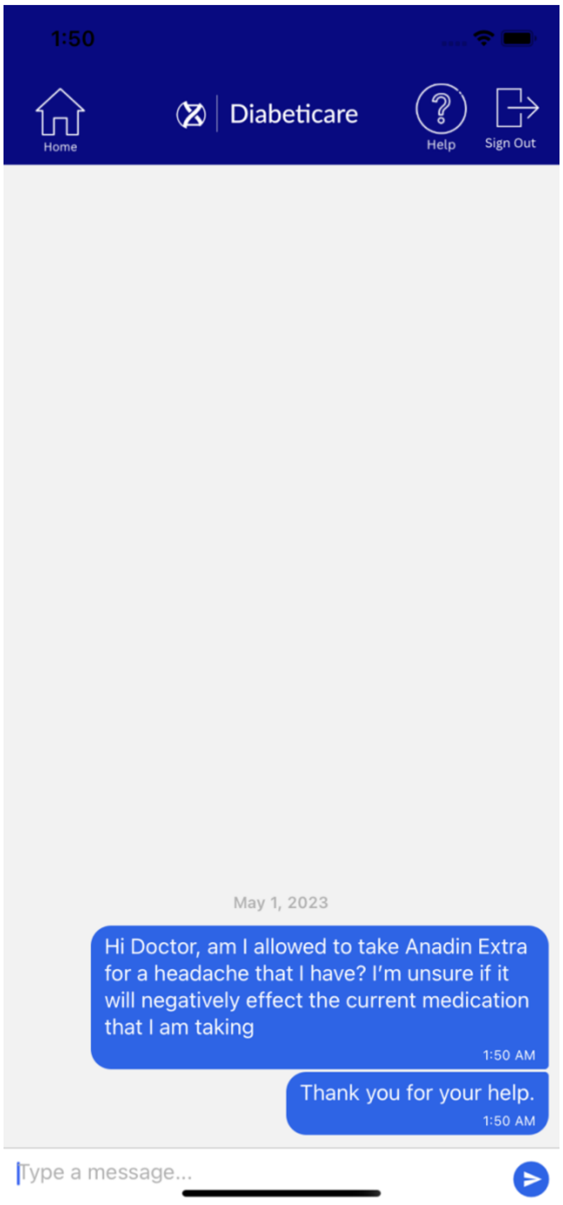
Medication: [INSERT MEDICATION NAME]
Dosage: [INSERT DOSAGE]
Frequency: [INSERT FREQUENCY]
Instructions: [INSERT INSTRUCTIONS]

Doctor Name:
[INSERT DOCTOR NAME]
Phone Number:
[INSERT PHONE NUMBER]

Clinic Name
[INSERT CLINIC NAME]
Email:
[INSERT EMAIL]

Figure 38: Result of generating PDF.

Appendix J – Chat Page



May 1, 2023

Hi Doctor, am I allowed to take Anadin Extra for a headache that I have? I'm unsure if it will negatively effect the current medication that I am taking
1:50 AM

Thank you for your help.
1:50 AM

Figure 40: Chat page.