ASHESI

ASHESI UNIVERSITY COLLEGE

MOBILE APPLICATION TO TRACK DIET OF DIABETICS: USING TELEMEDICINE TO ENHANCE HEALTHCARE DELIVERY

APPLIED PROJECT

B.Sc Computer Science

NANA ADWOA OWUSU DARKO

2018

ASHESI UNIVERSITY COLLEGE

Mobile Application to Track Diet of Diabetics: Using Telemedicine to Enhance Healthcare Delivery

APPLIED PROJECT

Applied Project submitted to the Department of Computer Science, Ashesi University College in partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Science

Nana Adwoa Owusu Darko

April 2018

DECLARATION

I hereby declare that this Applied Project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

……………………………………………………………………………………………….……

Candidate's Name:

……………………………………………………………………………………………..…

Date:

………………………………………………………………………………………………..

I hereby declare that preparation and presentation of this Applied Project were supervised in accordance with the guidelines on supervision of Applied Project laid down by Ashesi University College.

Supervisor's Signature:

……………………………………………………………………………………………….……

Supervisor's Name:

……………………………………………………………………………………………..…

Date:

………………………………………………………………………………………………..

ACKNOWLEDGEMENT

I am grateful to my lecturers from the various departments who have taught me over the years to give me the knowledge to write this paper. I thank my supervisor Dr Elena Rosca for her immense contribution to this project. She advised and reviewed my work constantly to make sure I was on the right track. Her knowledge in the topic was key to the success of this project.

I also want to extend my gratitude to Mr David Ebo Adjepon-Yamoah who always found time out of his busy schedule to review the implementation of my code. His knowledge and in Web Technologies and Software Engineering contributed substantially to the implementation of this application.

ABSTRACT

This project researches on how telemedicine can enhance healthcare delivery for diabetics in Ghana, and based on the findings, builds a technological tool to help diabetics track the level of sugar they consume.

Using the Korle-Bu Diabetic Centre as a prototype, a session was conducted to observe activities at the centre, and an interview with the Director of the department was held to gain insight into the state of the hospital. 50 patients were also interviewed to find out how often and why the visit the centre. The interviews revealed that the main purpose of patients' visit to the Korle-Bu Diabetic Centre was to check blood sugar levels, and this showed low levels of self-management among patients.

My research showed that telemedicine could be used as tool for remote consultation, electronic record keeping and for self-management of chronic diseases Based on these findings, this project built an application to improve self-management of diabetes by allowing patients to track the amount of sugar they consume in their daily meals. The app also facilitates remote consultation and electronic record keeping. This app contains a large database of the nutritional values of Ghanaian foods and is built specifically for the Ghanaian community.

In the long run, the app will be developed further to allow connections with devices such as the fitbit which track weight gain and loss, and heartbeat rate. This will enable the user to get more holistic data and reassurance that the figures being computed are indeed right.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Background and Motivation

In July 2010, the National E-Health Strategy was launched by the Government of Ghana. The aim of this strategy was to streamline the regulatory framework for health data management, build a health sector capacity wide enough to accommodate e-health solutions, work towards achieving paperless record keeping, and increase access to healthcare to bridge the equity gap in the health sector (Afarikumah, 2014). As at 2014, a research carried out by Ebenezer Afarikumah of the Accra Institute of Technology to provide an overview of e-health activities in Ghana showed that there were only 22 pilot projects under e-health in Ghana. Contributors to these 22 projects ranged from international organizations, institutions dedicated to the promotion of e-health, and multinational companies through their corporate social responsibility. At that time, some of these projects had been suspended indefinitely while others were still in the implementation phase. Although e-health seems to have limitations in its role in Ghana presently, it is a growing area of interest to many researchers who are working to explore the opportunities it can offer in the remote delivery of and access to healthcare services, especially in developing countries with rural areas (Afarikumah, 2014).

E-health an acronym for Electronic Health, refers to the electronic transfer of health-related data through an electronic connectivity to enhance healthcare delivery. It could also be defined as the application of Information Communication and Technology (ICT) to all operations across the healthcare industry (Ouma and Herselman, 2008).

One of the fast-growing fields under e-health is Telemedicine. Telemedicine refers to the remote delivery of healthcare services such as consultations and assessments using telecommunications technologies (Sood et al., 2007).

According to Stella Ouma and M.E Herselman of The World Academy of Science and Technology in Trieste, Italy, Telemedicine has two categories which are real time and pre-recorded. Real-time telemedicine allows participants to send and receive information immediately with unsubstantial delay, whereas pre-recorded telemedicine stores information, encapsulates the information and then forwards it to the receiver for subsequent reply (Ouma and Herselman, 2008).

The benefits of telemedicine include the fact that it serves as a medium for record-keeping as applications can save patient records which can be viewed by both the patient and the doctor. Most importantly, it eliminates travelling cost as healthcare services can be provided remotely, and encourages self-management which is important in treating chronic diseases. Telemedicine has the potential to revolutionize the delivery of healthcare in Ghana, especially in the area of focus of this paper which is healthcare delivery at the Korle-Bu Diabetic Centre. This project uses the Korle-Bu Diabetic Centre in Accra- Ghana as its prototype in building the application to track the sugar consumption of diabetics.

From visiting the Korle-Bu Diabetic Centre and interviewing the Director Dr Atease, my research showed that the centre has two prep nurses, two consultation nurses and two doctors from the Korle-Bu Medical School who see 200-250 patients every day. The patients start arriving around 6 am to queue in order to see the doctors early enough to avoid spending the whole day at the hospital. There is no appointment system as the patients are attended to on a first-come first-serve basis. This leaves no guarantee of meeting with a doctor because of their inability to attend to the large number of patients within the 8 am to 5 pm working hours.

The two doctors on call share one consulting room with two desks where they pile the hardcopies of patient records. The prep nurses take the vitals of the patients while they wait for consultation. The consulting nurses call the patients into the consulting room in pairs, and assign one patient to each doctor. With this style of consulting there is no privacy for any of the two patients.

Normally, patients visit the centre to have their blood sugar checked. As at the year 2002, patients checked their blood sugar levels and HGA1C monthly or bimonthly through having their blood drawn. They had to rely on these numbers until their next visit to the hospital which would usually be in another month (Blankson, 2002). In recent years, the centre has adopted the use of test strips, lancets, and a lancet device to check patient blood sugar levels which has sped up the process. However, utilizing remote consultation services and keeping electronic journals on mobile devices to track patient eating habits and sugar consumption levels can help the centre cater for the growing number of patients.

Furthermore, according to Peace FM news, a fire engulfed the centre on the morning of 23rd August 2013, resulting in the loss of patient records. My research revealed that efforts had been made to set up an electronic back-up system but the system remained inefficient and faulty.

This project seeks to solve the problems the Korle-Bu Diabetic Centre faces in delivering its healthcare through creating a phone application to track sugar consumption levels among the technologically savvy patients who visit the centre. Using a mobile phone application will make the functions of the app very personalised. Per the 2012 Pew Research Center’s Internet and American Life Project, 85% of adults and 77% of adolescents own a mobile phone, with half of these respective percentages using smartphones (Eng & Lee, 2013). In the last five years, the number of medical applications has grown to over 13,000 and 6,000 for iPhone and Android users respectively. Due to its widespread dissemination, mobile technology holds great promise for transforming the health care delivery system (Eng & Lee, 2013).

This application will allow patients to input their daily meals in their right quantities, and view a chart of the amount of sugar they consume each day. This will facilitate self – management on the patient's part. Additionally, doctors can view the vitals and statistics of the patient's health through the application, and schedule an interview with them if necessary.

The application will allow electronic patient record keeping which can be retrieved at any time by both the doctor and patient. It will improve the delivery of healthcare through making consulting more personalised. Ultimately, it will cut down drastically on the number of patients who visit the centre each day, and relieve the work load on the doctors and nurses on call.

According to the 2012 National Policy for the Prevention and Control of Chronic Non-Communicable Diseases, 9% of Ghana's 25 million population suffer from diabetes. According to the International Diabetes Foundation, 330,000 Ghanaians have undiagnosed diabetes, and 8,500 people died from diabetes in 2012 (Global Press Journal, 2015). This project will ultimately help the growing population of diabetics in Ghana to manage their disease better with help from doctors and other healthcare practitioners.

1.2 Related Works

The first e-health application built for diabetics in Ghana was called Dr Diabetes, a web application that helps people assess their risk of diabetes. This application was built by Raindolf Owusu the CEO of OasisWebSoft during his time as a student of Methodist University College in 2011. The application asks users the frequency of their consumption of sugary foods and alcohol, and levels of exercise, and based on their responses, the app tells the user whether they are at risk of developing diabetes, and suggests preventive measures (Global Press Journal, 2015). The limitation of this application is that it only assesses an individual's risk of developing diabetes. It does not give the user a sense of their blood sugar levels and as such this application does not eliminate the need to visit the hospital. As a matter of fact, to cater for this gap, it suggests regular hospital visits to its users, to get their blood sugar levels checked by doctors. This practice does not encourage self-management of diabetes. However, research shows that self-management is very crucial in the treatment of chronic diseases such as diabetes.

In a research by Andrea A.G Nes from the Department of Behavioural Sciences in Medicine at the University of Oslo, it came to light that smartphone applications are a promising tool for supporting patients with diabetes (Nes et al, 2012). Smartphones are portable devices which users carry around everywhere, making it easy for them to log their meals wherever they are. For diabetics, the smartphones can be used for electronic record keeping of meals consumed, and serve as a medium through which doctors can view the eating habits of patients to inform a remote consultation. The aim of Andrea's research was to test the feasibility of a web based application delivered through a smartphone to support self-management of diabetics (Nes et al, 2012). Over a three-month period, the participants of this research entered their eating behaviour, prescriptions, physical activities and emotions into the application three times each day. In addition, they had to input their blood glucose levels in a morning diary which a therapist had access to (Nes et al, 2012). The therapist immediately assessed the information and sent personalized feedback to the patients. The participants reported positive lifestyle changes after the three months (Nes et al, 2012). With this methodology, the therapist only had access to the patients' blood glucose levels, hence all feedback was based on the blood glucose levels and not on the eating habits of the patients. Combining the eating habits and blood glucose levels of patients would have provided the therapist with holistic data to inform diagnosis. For more robust and reliable diagnosis formed through remote consultation, the healthcare practitioner must have access to all patient information pertaining to eating habits, food information and blood sugar levels. Mobile apps which allow these features prove to be very effective in remote healthcare delivery and self-management.

Studies by Anoop Rao (Rao, 2010) who is part of the faculty at Stanford University School of Medicine, have indicated that sharing of self-monitored of blood glucose (SMBG) data and subsequent feedback from the health care provider (HCP) can help achieve glycaemic goals such as a reduction in glycated haemoglobin. This research showed that, being able to record, analyse, seamlessly share, and obtain feedback on the SMBG data using an app might potentially benefit patients (Rao, 2010).

According to Eirik Arsand (Arsand et al, 2012) a professor in e-health who researches around how to use mobile technologies to improve self-management of health, self-management is very important in achieving diabetes treatment goals, and mobile phones can support this immensely. Arsand's research revealed that some of the basic functionalities that the most effective mobile health applications have include SMS, a mobile phone diary to enhance reflection, communication between the healthcare professionals and patients, food information, blood glucose modelling, automatic data transfer, and motivational and vital user interfaces (Arsand et al, 2012). Although this article explained how certain features of the system can enhance its use, it is not localized for the Ghanaian market. The food information available is for only American indigenous dishes and the app cannot give information on Ghanaian dishes.

1.3 Project Summary

This project focuses on using telemedicine to improve healthcare delivery for diabetics and enhance self-management of the disease. The project will use a smartphone as a medium of delivering remote consultation and assessment in real – time.

At the end of this project, a phone based application called DiaCare will be built. This application will enable users to perform the following;

- Input daily meals and their quantities
- View a graph model of their sugar consumption level
- Contact their doctors and schedule appointments
- Receive feedback and suggestions in real-time from their doctors

All these features are included in the mobile application that this project seeks to create to give the user a personalized and trustworthy experience as they interact with the application.

This application will be an HTML 5 application. HTML 5 is a programming language which is used to create web applications. The application will then be compressed into an android phone application using Cordova phonegap, which is a framework which transforms web apps into phone apps. It will employ the hybrid model, using PHP for the server-side communication and JavaScript for the client-side communication. PHP is a programming language which allows data to be sent to the database for storage, while JavaScript is the scripting language that connects the data from the HTML page to the PHP page. The graphs will be drawn using Google's Chart.js library. It will run on a web server which can be accessed by both the patient and the healthcare professional.

CHAPTER 2: REQUIREMENTS

2.1 Overview

The end-users of this application are the patients, who are diabetic in this case, and a certified doctor who can assess the patient's statistics and inform them of necessary actions to take. These two key stakeholders were interviewed in the requirements discovery process, to ensure that the final product meets their specific needs directly.

This project used the Boehm's spiral model in its requirement engineering process. The method mentioned above was chosen because of its combination of waterfall and iterative models which allows the requirements to be developed in phases, and changes made based on feedback from the user (Boehm, 1988). The process of the Boehm's spiral model used in the requirement engineering process of this project is outlined below;

- Requirements discovery: This is the basic level at which information about the proposed system is gathered and divided into user requirements and system requirements. Using the viewpoint oriented approach, the user requirements were identified through the interactor viewpoints.
- Domain viewpoints used internet standards of other diabetes applications to set system requirements.
- Requirements classification and organization: This level deals with grouping the requirements to suit the two users of the application which are the patient and the doctor.
- Requirements prioritization and negotiation: This level deals with modifying certain features of similar or existing apps to suit the Ghanaian user. Since the application this

project seeks to build is being developed from scratch, other applications are only used as a guide.

- Requirements documentation: This is the process of recording all processes involved in developing the application software, and producing a detailed document of these events at the end of the development process.

This chapter employs Use-case and activity diagrams to elaborate on the functionalities of the application. The use-cases put into perspective the requirements of the actors (end-users) and their possible (scenarios) uses of the application. A list of typical use-cases for this application are listed below;

- Grant access to a patient and a doctor to use the application
- Input meals and quantities
- View sugar consumption level using a graph model
- View general information about suggested meals based on blood sugar consumption levels
- View doctor's information
- View patient's information and sugar consumption level
- Schedule an appointment
- Receive SMS confirmation of booked appointment

Further in this chapter, non-functional requirements, system requirements, and limitations of the project will be discussed in detail as well.

2.2 Application Users

This mobile application has two main users;

- Patient
- Doctor

The functions these two users can perform with the application can be grouped into four main tasks: (1) Inputting meals, (2) Viewing sugar consumption levels, (3) Receiving meal suggestions (4) Scheduling an appointment.

2.2.1 Inputting meals

The patient will open a page which has two input fields and a submit button. The first input field will allow for entry of the name of the meal, and the second input field will allow for entry of the quantity. The user will then click the submit button once all values are entered.

2.2.2 Viewing blood sugar

The patient can view a bar graph of the level of sugar they consume based on their food consumption. The graph will turn red when the sugar consumption level is at alarming rates, and will remain blue when within safe levels. A doctor can also view the sugar level of a selected patient.

2.2.3 Receiving meal suggestions

The patient will receive regular meal suggestions which will be generated based on the sugar consumption level of the patient. To help the patient keep the sugar level within safe boundaries, meals which are high or low in sugars will be suggested to the patient depending on their level of blood sugar.

2.2.4 Scheduling an appointment

The patient can select an available date from the doctor's calendar which will be shown with unavailable dates blocked out. The doctor will have to block out their unavailable dates and times. Once a patient selects an appointment date and time, the doctor will receive an SMS notification, and then will send a confirmation in response to the patient. A doctor can contact a patient through a phone call or an SMS when necessary, and in cases where an appointment is required, the patient will schedule it using the process described above.

2.3 Usage Scenario Viewpoints

The use case diagram Figure 2.1 below displays all the possible interactions the patient and the doctor can have with the application.

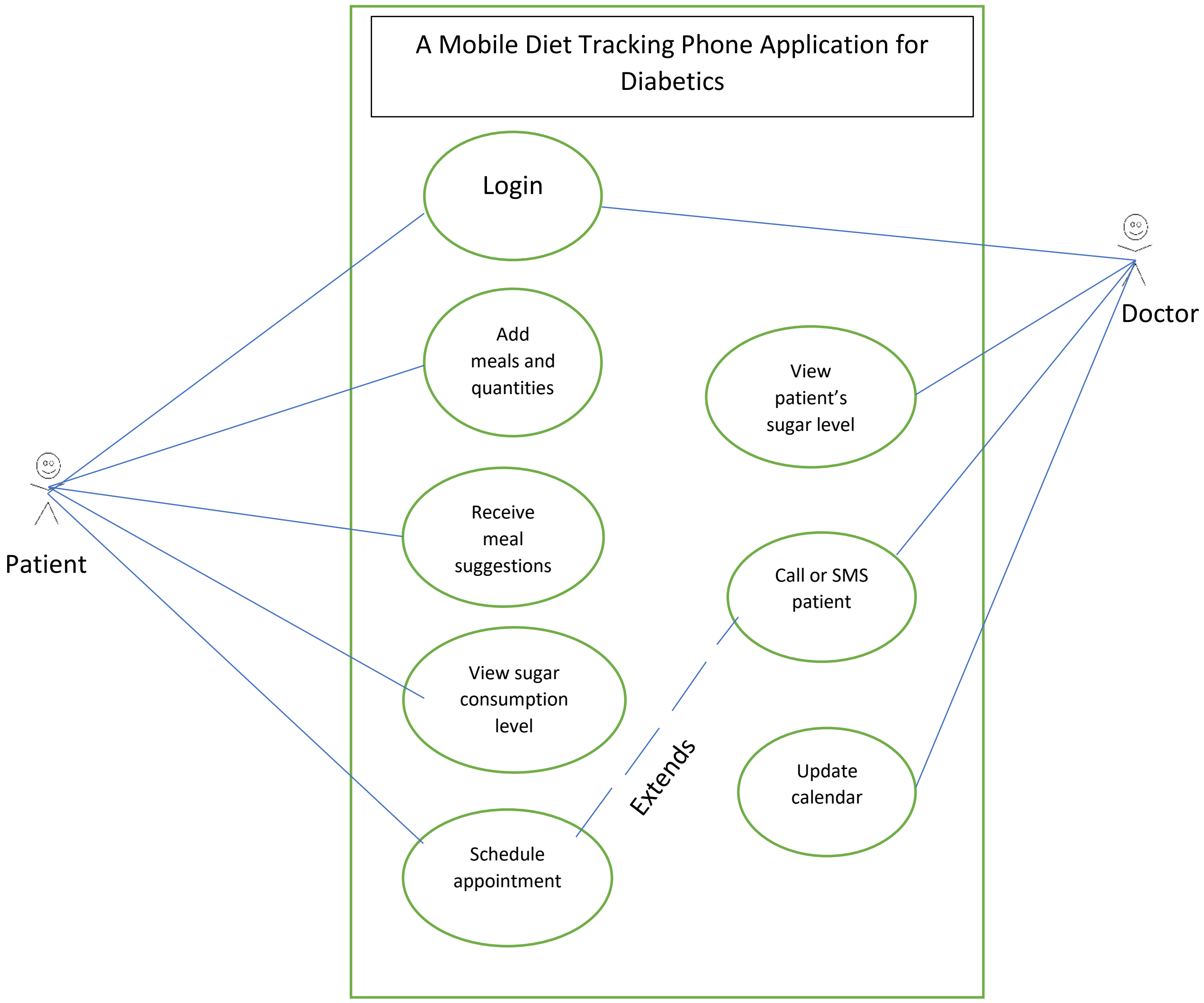

Figure 2.1 Use case diagram of DiaCare mobile app

2.3.1 The Process of Logging In

The application will allow patients and doctors to log into the application using their emails and passwords. These users will have to register first, by submitting their full names, phone numbers, emails and passwords which will be stored in the database. Once registration is successful, they will be sent an SMS notification which will prompt them to login. The email and password submitted during login is compared to what is stored in the database and if there is a match, they will be redirected to the dashboard page.

- Pre-conditions: The user must register by submitting all the required registration details using the app.
- Post-conditions: The registered user can log into the app anytime and input meals, view blood sugar, schedule appointment.

Figure 2.2 below shows the activity diagram of the login activity

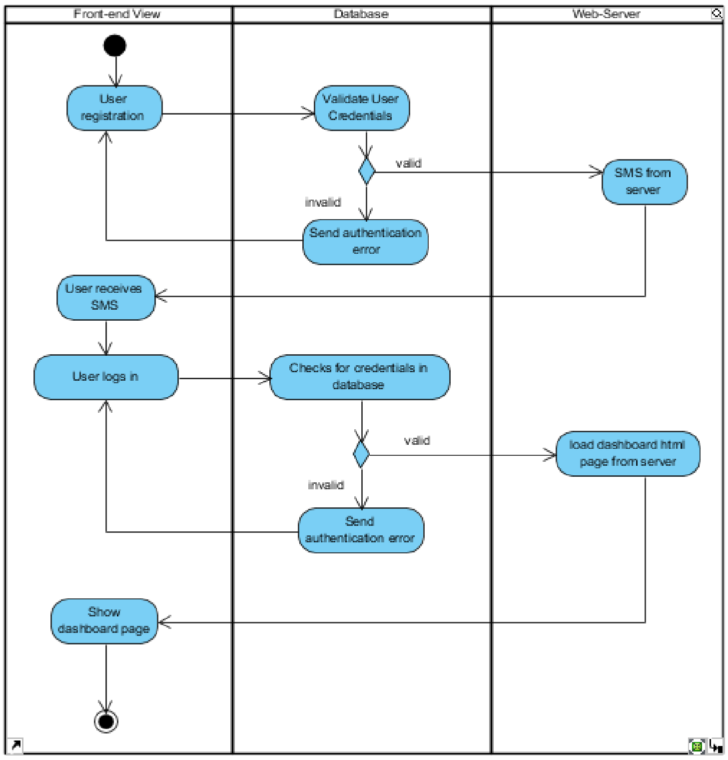

Figure 1.2 Activity diagram of login process

Figure 2.2 Activity Diagram of the Login Process

2.3.1.1 Explaining the activities in the activity diagram of the login process

1. User registration begins the activity of logging in. The user inputs their name, email, phone number and password as shown in Figure 2.3 below

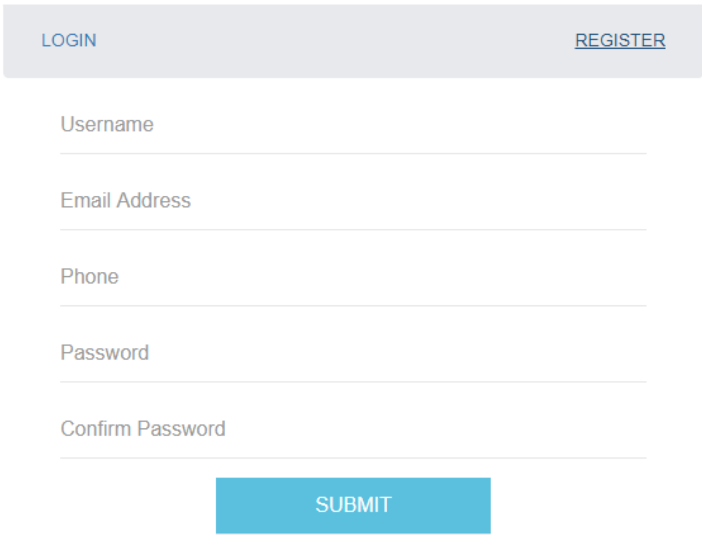

Figure 2.3 Registration form for DiaCare app

2. The form validates the user input to check that all information provided is valid.
3. If it is not valid, the user is prompted to input the credentials again.
4. All valid registration credentials are stored into the database.
5. An SMS is sent from the Npontu SMS Server to the user to confirm successful registration.
6. The user receives an SMS confirmation.
7. The user can then log into the application using the form as shown in Figure 2.4 and Table 2.1 below;

Table 2.1: Required information for login

Data Element	Description	Format	Required?
Email	Unique email for the app	String, any length, in the format name@domain.com	Yes
Password	Unique password for locking user sessions	String, any length of combined ASCII characters.	Yes

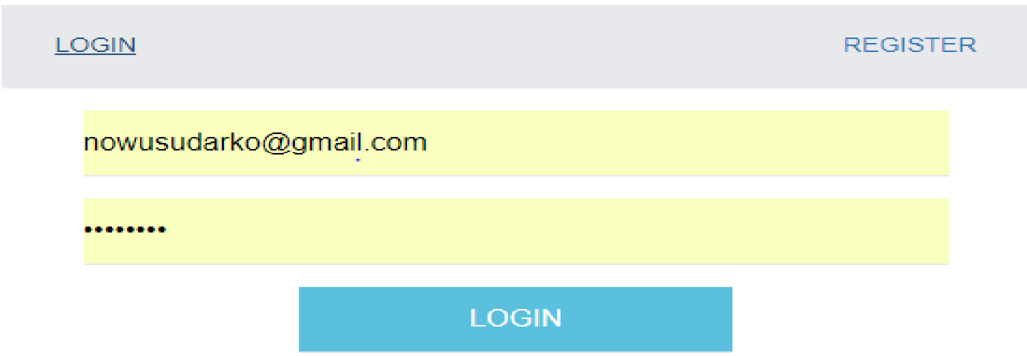

Figure 2.4: Login form for DiaCare app

8. The login credentials are validated by checking for the inputted credentials in the database
9. If these credentials are not found in the database, the user is prompted to type them again
10. Once these credentials find a match in the database, the web server loads the dashboard page

11. The dashboard page is then rendered in the web browser and the user is redirected to that page

2.3.2 The Process of Inputting Data

This application will allow a patient to input meals in the quantities in which they were consumed. Once a user is successfully logged in, they can select a button on the dashboard page which will load a modal with a form to the input meals page. On that modal the user will have two text fields. One will allow for entry of the name of the meal, and the second text field will allow for the amount to be input. The user will have a submit button which will send the data to the database. This data will be compared to a pre-populated database of a range of meals and the quantity of sugar present in each portion. This database will be used to calculate the sugar in the meals the user inputs, and the resulting values will be used in populating the bar graph to show the level of sugar the user consumes.

- Pre-conditions: The user must input all components of their meals in their right quantities. Example; Rice- 1 cup, Beans- 1 cup, Chicken- Drumstick.
- Post-conditions: The computed sugar levels will be used to update the bar graph

Figure 2.5 shows an activity diagram of the process of inputting a meal.

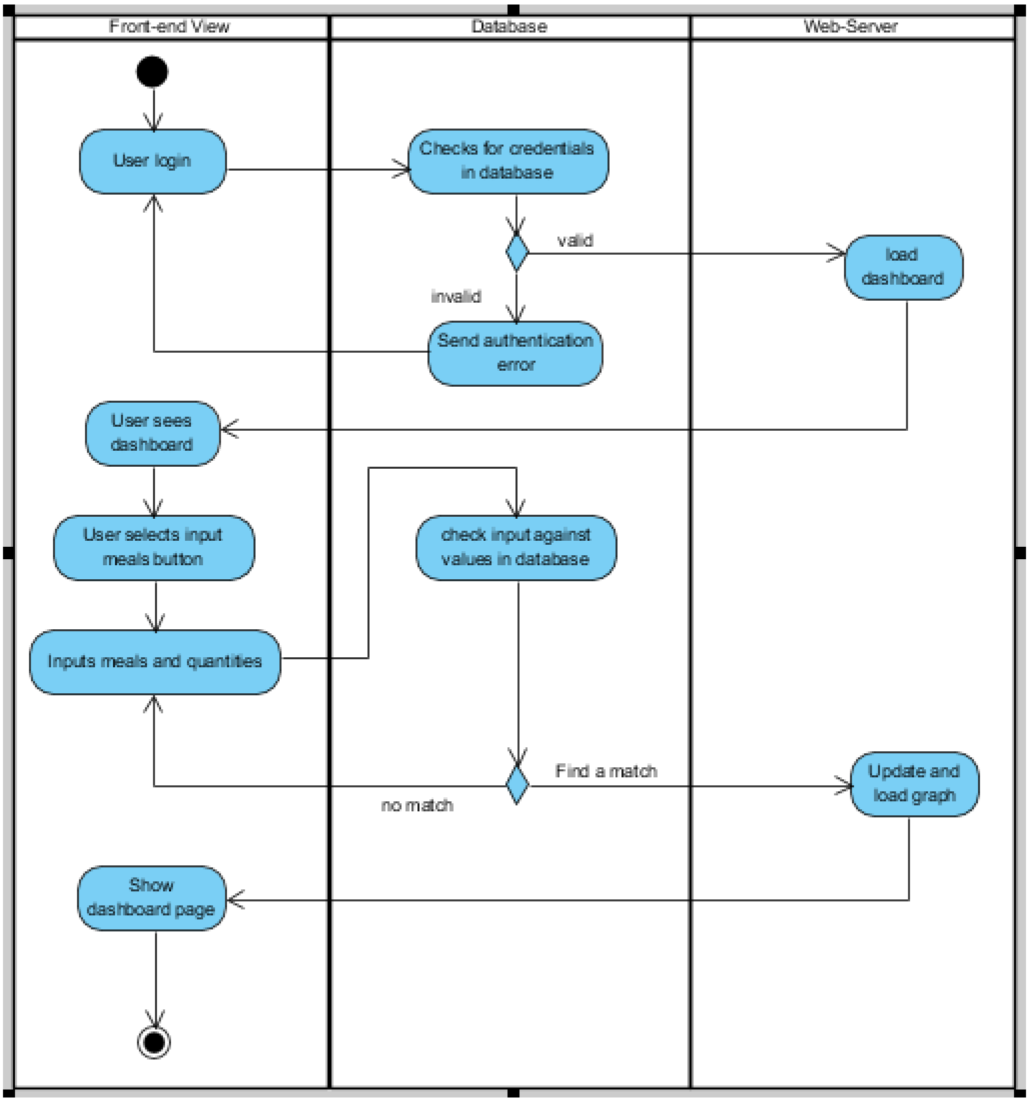

Figure 2.5: Activity diagram of the process of inputting a meal

2.3.2.1 Explaining the activities in the activity diagram of the process of inputting a meal

1. The user logs in using their email and password
2. The submitted values are checked against a database to see if they are valid and exist in the database
3. If no match is found, the user is prompted to submit the credentials again
4. If a match is found, the user is redirected to the dashboard page
5. The user selects the input meals button shown in the image below

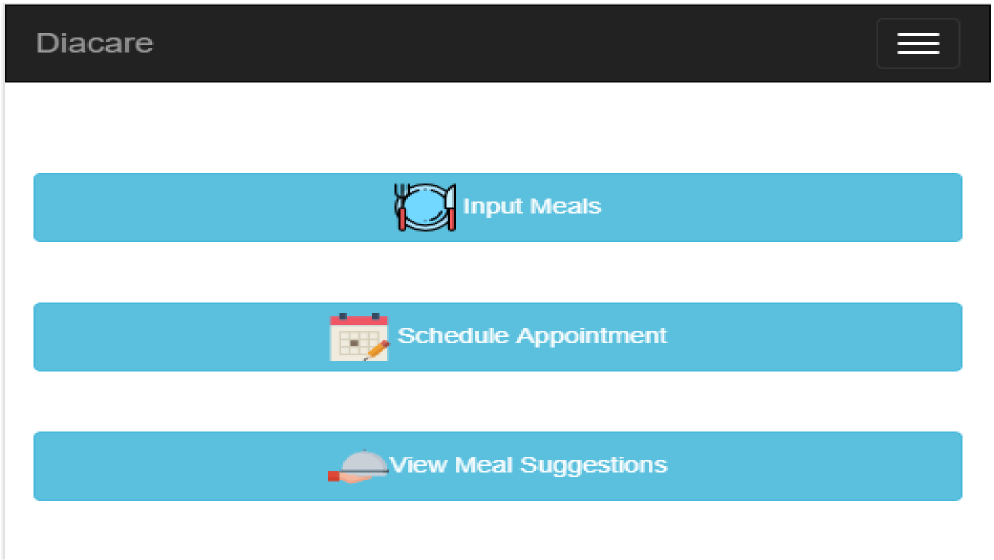

Figure 2.6: Buttons on the dashboard

6. A modal with a form pops up and the user fills two input fields, one takes the name of the meals, and one takes the quantities. Table 2.2 below shows the required information needed to input meals and Fig 2.7 shows the modal

Table 2.2: Required information to input meals

Data Element	Description	Format	Required?
Name of meal	Name for the meal as normally called in the Ghanaian society	String, any length	Yes
Quantity	Quantity measured is simple metrics or parts such as cups, oz, chicken breast or drumstick etc.	String, any length	Yes

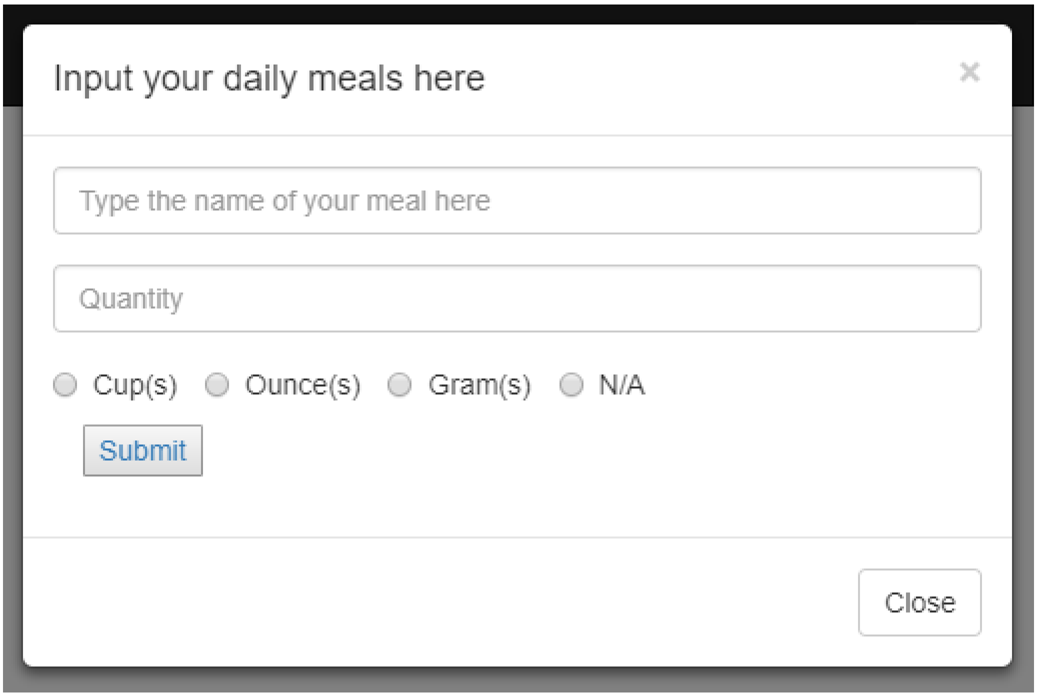

Figure 2.7: An image of the input meals form

7. The user submits the values entered and the values are compared against values in a pre-populated database
8. If the meal is not found in the database, the sugar level cannot be computed and the user is prompted to enter another name of the meal which is more common
9. If the meal is found in the database, the amounts of sugars in its components are calculated and the information is sent to the server
10. The server updates the bar graph and reloads the graph for the user to view

2.3.3 The Process of Viewing Sugar Consumption Levels

This application allows both the patient and the doctor to view the graph of the sugar consumption level of the patient. Using a bar graph, the user can visualise the sugar consumption level. These users will login with their credentials, and then for the patient, they can view the graph on the dashboard page. However, the doctor, after logging in, will be redirected to a page of all patients. A patient will then be selected, and the information of that patient will show, including the sugar consumption level of that patient as a graph.

- Pre-conditions: The user must provide authentic login credentials, and be granted access to the dashboard if the user is a patient, or to the list of all patients page if the user is a doctor.
- Post-conditions: The user can view the sugar consumption level as a bar graph

Figure 2.8 below shows an activity diagram of viewing the sugar consumption level from the doctor's side

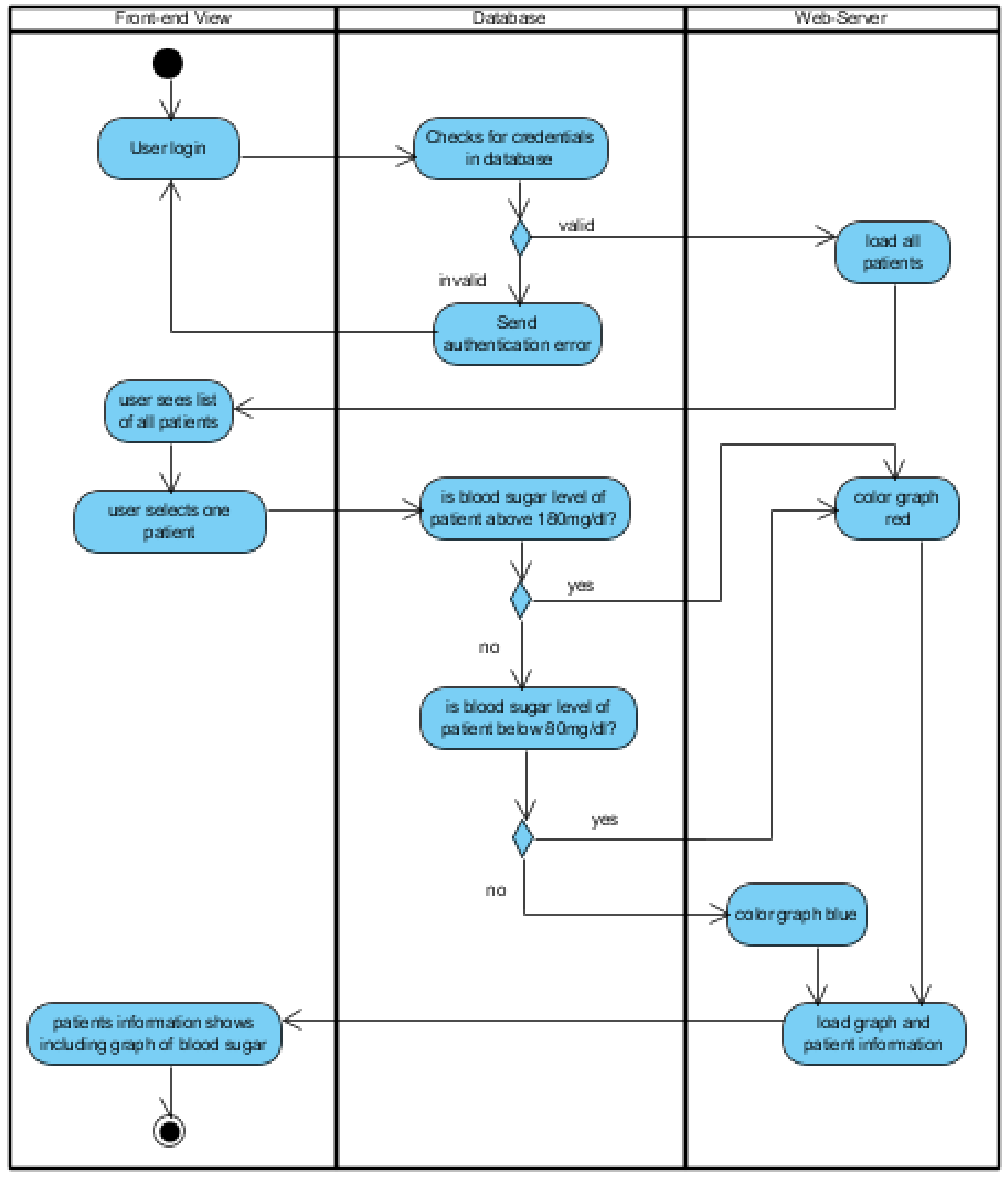

Figure 2.8: Activity diagram of viewing sugar consumption level from the doctor'sside

Figure 2.9 below shows an activity diagram of viewing sugar consumption level from the patient's side

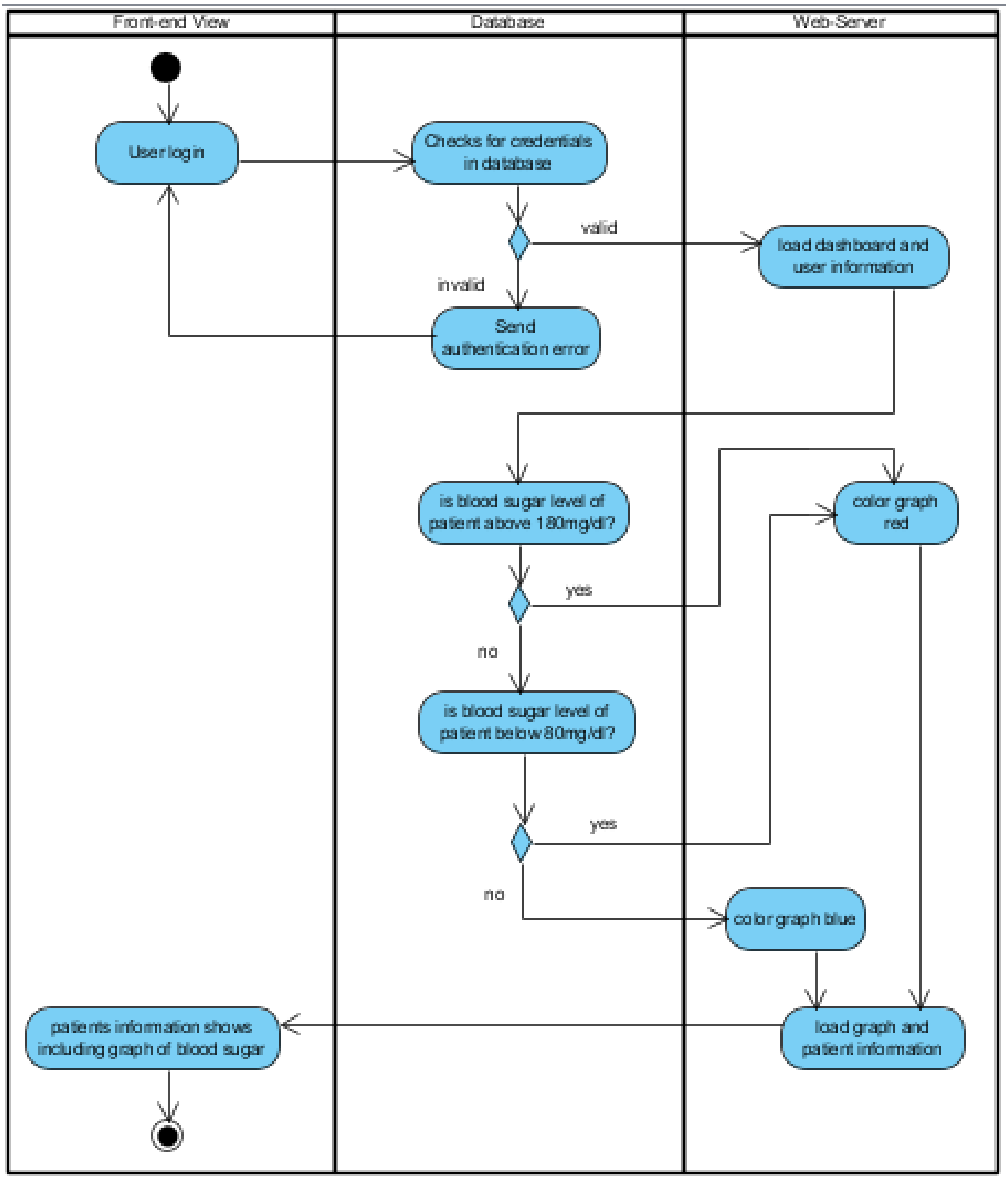

Figure 2.9: Activity diagram of viewing sugar consumption level from patient's side

2.3.3.1 Explaining the activities in the process of viewing sugar consumption levels from the patient and doctor's side

1. Both the patient and the doctor will have to login to start this process. They will input their email and password
2. The provided credentials are checked against the database to see if they have been registered to use the app
3. If they have not been registered, a match will not be found in the database and the user (patient or doctor) will be prompted to register or crosscheck the submitted credentials for errors
4. If the user is a doctor, after a successful login, they will be redirected to a page which lists all patients.
5. The doctor will then select from the list of patients
6. The selected patient's information will then be loaded from the database onto the server
7. The server will check for the sugar consumption level as computed and stored in the database.
8. If the sugar consumption level is above 180mg/dl or below 80mg/dl using American standards, the graph will be a red colour to show unsafe sugar consumption levels, otherwise if it falls anywhere within the values above, it will be blue in colour to show safe sugar consumption levels.
9. If the user is a patient however, then after a successful login, they will be redirected to the dashboard page where the graph of their sugar consumption level is displayed automatically once the page loads. The previous step above is followed in determining the colour of the graph to be shown on the dashboard as well.

Figures 2.10 and 2.11 below show the sugar consumption levels in safe and unsafe ranges.

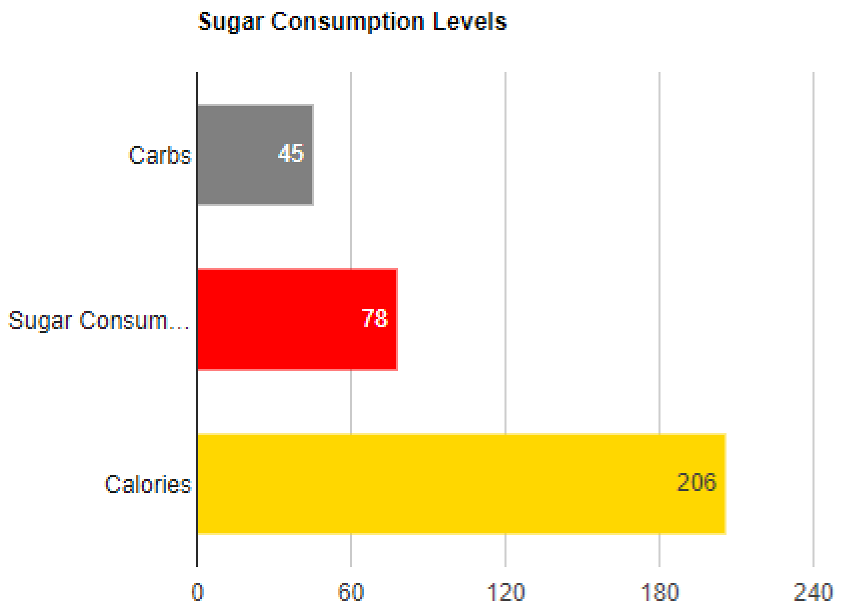

Figure 2.10: Showing sugar consumption level below 80mg/dl in unsafe range

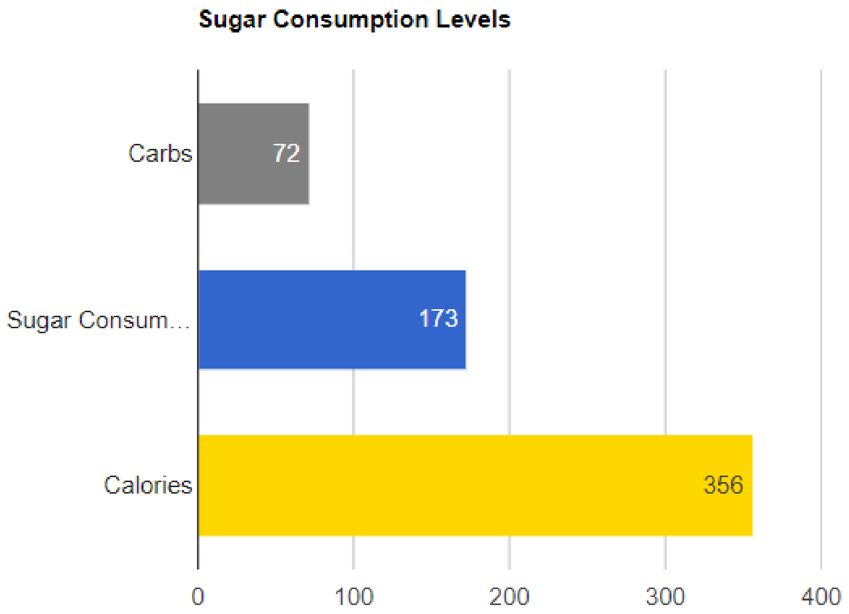

Figure 2.11: Showing sugar consumption level within safe range (between 80-180mg/dl)

2.3.4 The Process of Receiving Meal Suggestions

This application regularly suggests foods for the user (patient) based on their sugar consumption levels. This process starts by assessing the sugar level of the patient as stored in the database. The app will then suggest and encourage the consumption of foods which are either high or low in sugars in order to keep the sugar consumption level of the patient within a safe range which according to the National Diabetes Association is between a low of 80mg/dl when fasting (waking up in the morning), and a high of 180mg/dl after meals.

Figure 2.8 below shows an activity diagram of the process of suggesting meal

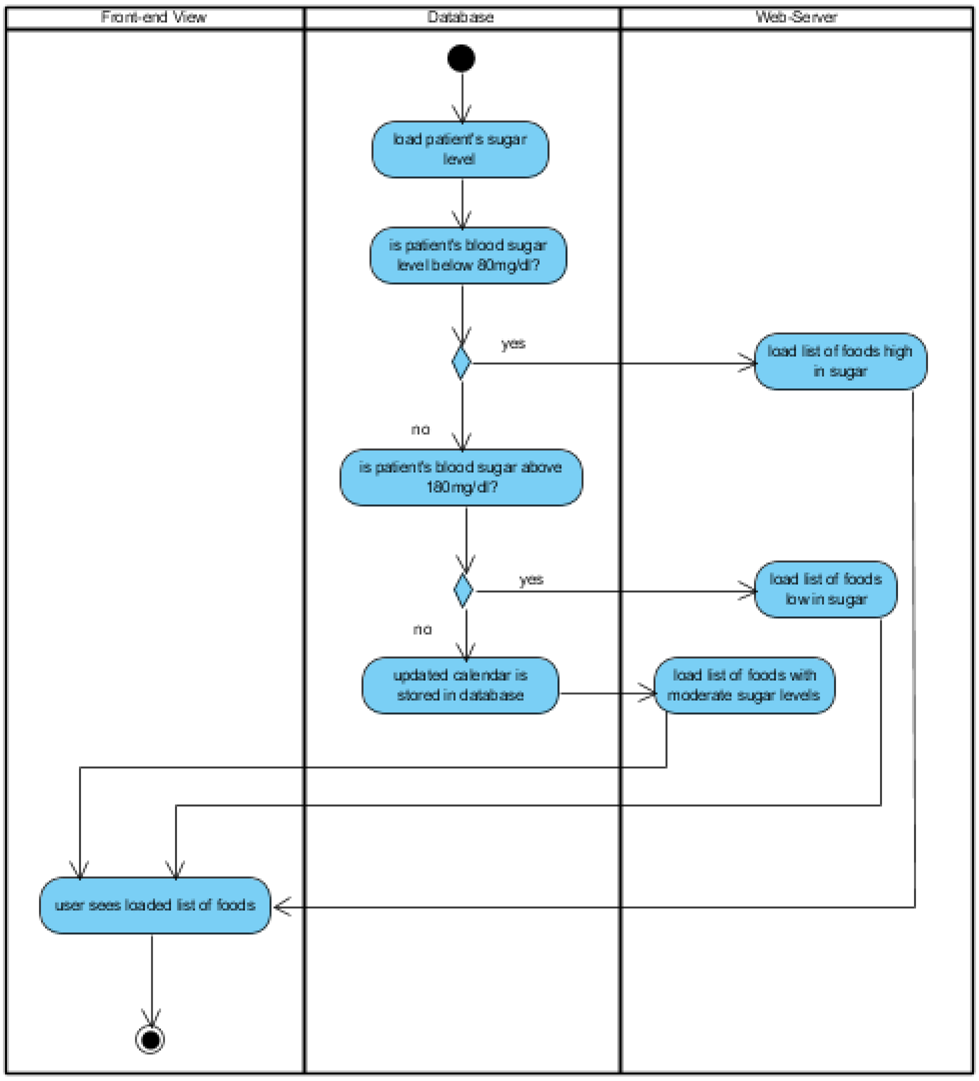

Figure 2.12: Activity diagram of the process of suggesting meals for the patient

2.3.4.1 Explaining the activities in the process of suggesting meals for the user

1. An SQL query loads the patient's sugar consumption level from the database
2. The value is assessed
3. If it is lower than 80mg/dl, a list of foods high in sugar are loaded from the database and sent to the server which then loads the page for the user to see this list
4. If the sugar level is higher than 180mg/dl a list of foods low in sugar are loaded for the user to see
5. If the sugar consumption level does not fall below 80mg/dl or above 180mg/dl, then it means it falls within the safe range, and a list of foods with moderate amounts of sugar are loaded for the user to see

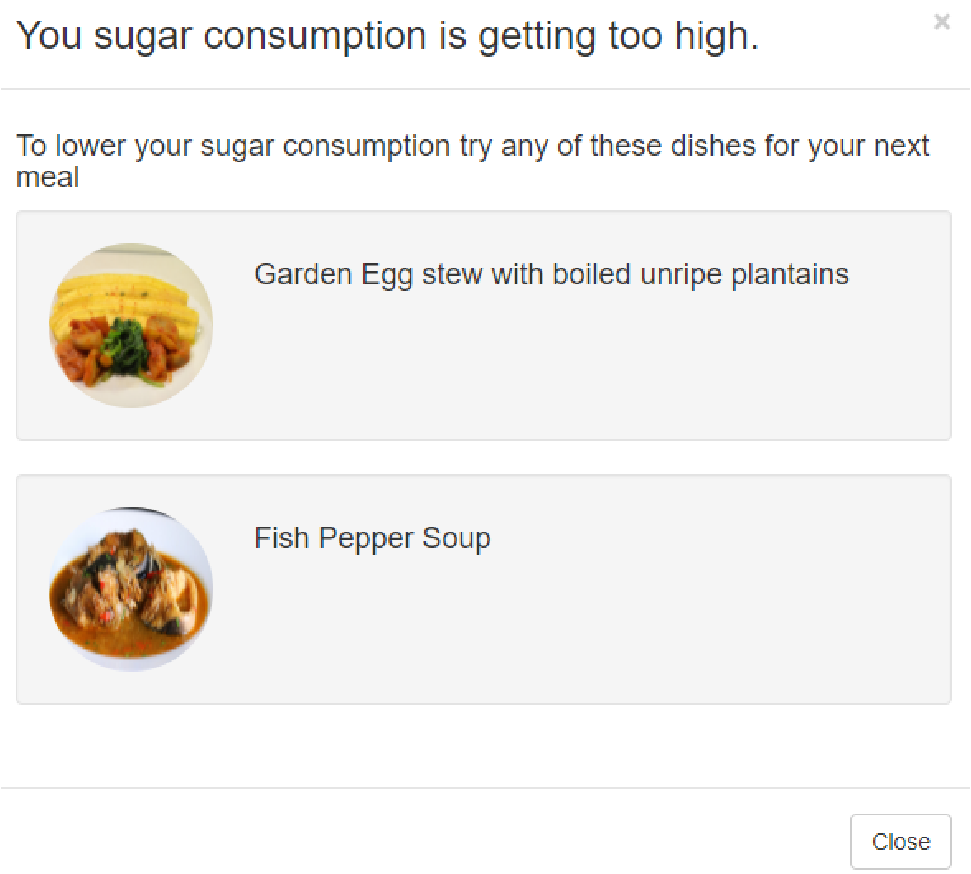

Figure 2.13: Image of the app suggesting meals which are low in sugar content

2.3.5 The Process of Scheduling an Appointment

The application allows a user to schedule an appointment. So far in this project, it is only the process of scheduling an appointment that has distinct procedures for the two users of the application. These procedures are enumerated below;

- For the prime user which is the patient, the process starts when they login successfully. Once they are redirected to the dashboard, they can select the 'schedule appointment' button which will redirect them to their doctor's calendar. They will then select a proposed appointment date and time and wait for to be contacted by their doctor through SMS or phone call for confirmation
- For the secondary user which is the doctor, after a successful login, they are redirected to a page which lists all their patients. At the top right-corner they select the 'update calendar' button. This button loads their calendars on which they must block all unavailable dates, and save the update. The doctor will also receive SMS messages whenever a patient schedules an appointment, and it will be the doctor's duty to respond in a timely manner.

Figures 2.8 below shows the activity diagram of scheduling an appointment from the patient's side

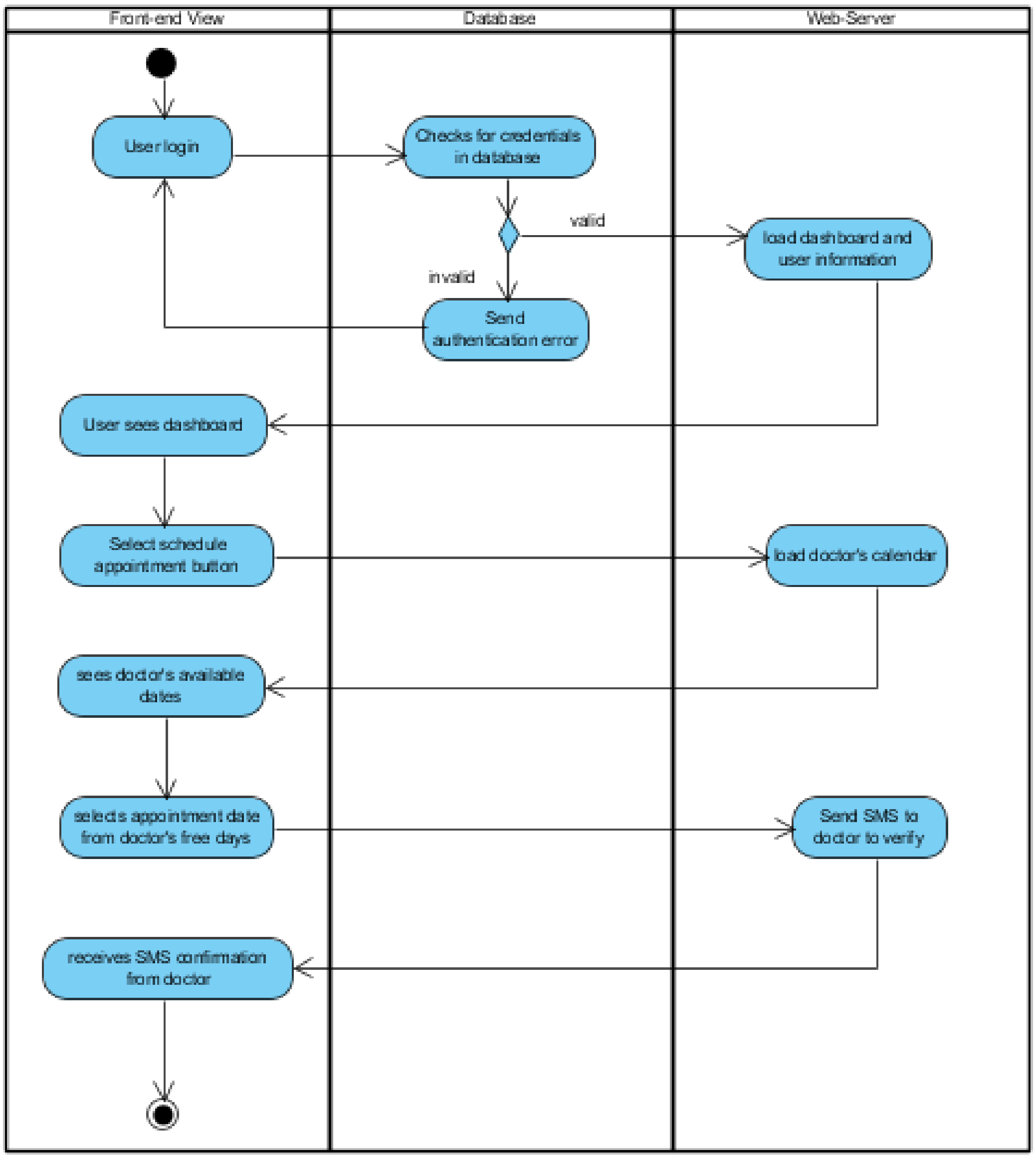

Figure 2.14: Activity diagram of the process of scheduling an appointment from patient's side

2.3.5.1 Explaining the activities in the process of scheduling an appointment as a user

1. The user submits login credentials
2. These credentials are validated using the existing database of registered users
3. If validation fails, the user is prompted to register or log in again
4. If login is successful, the user is redirected to the dashboard page
5. The user then selects the 'schedule appointment' button on the sidebar

6. The user's appointed doctor's calendar is loaded onto the page from the server
7. All the doctor's unavailable dates are blocked out
8. The user can only select from a list of available unblocked dates and times
9. Once the user selects, the SMS server sends the doctor an SMS to the doctor for verification
10. The patient then waits to receive a confirmation SMS from the doctor

Figure 2.9 shows an activity diagram of the process of scheduling an appointment from the doctor's side

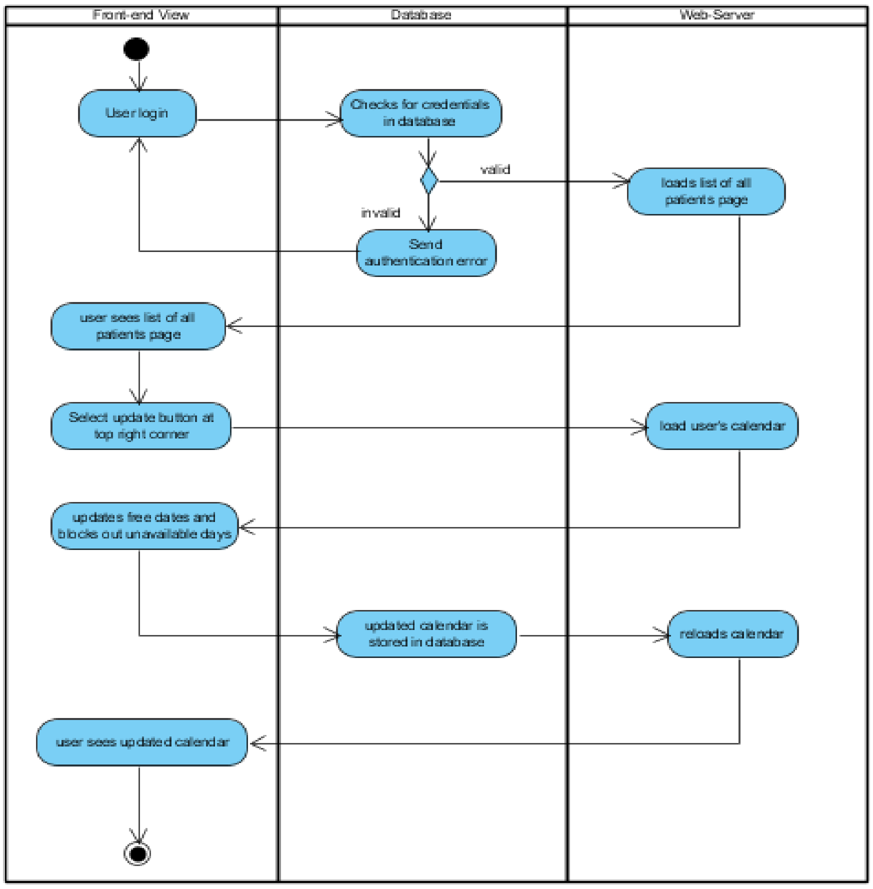

Figure 2.15: Activity diagram of scheduling an appointment from the doctor's side

2.3.5.2 Explaining the activities in the process of scheduling an appointment as a doctor

1. The user which is the doctor in this case must submit the required credentials to log into the application
2. The credentials are validated by comparing against values in the database
3. If there is no match found for the submitted credentials, validation fails and the user is prompted to register or check the submitted credentials for errors and re-attempt
4. If log in is successful, the user is redirected to a page with a list of all patients
5. The user then selects 'update calendar' at the top right corner of the page
6. This button loads the users calendar from the database
7. The user then blocks out unavailable dates or adds available days and times
8. The user then saves the changes
9. The updated calendar is stored in the database and reloaded to reflect the changes
10. The user can now see the updated calendar
11. The user must respond to pending appointment requests from patients through SMS

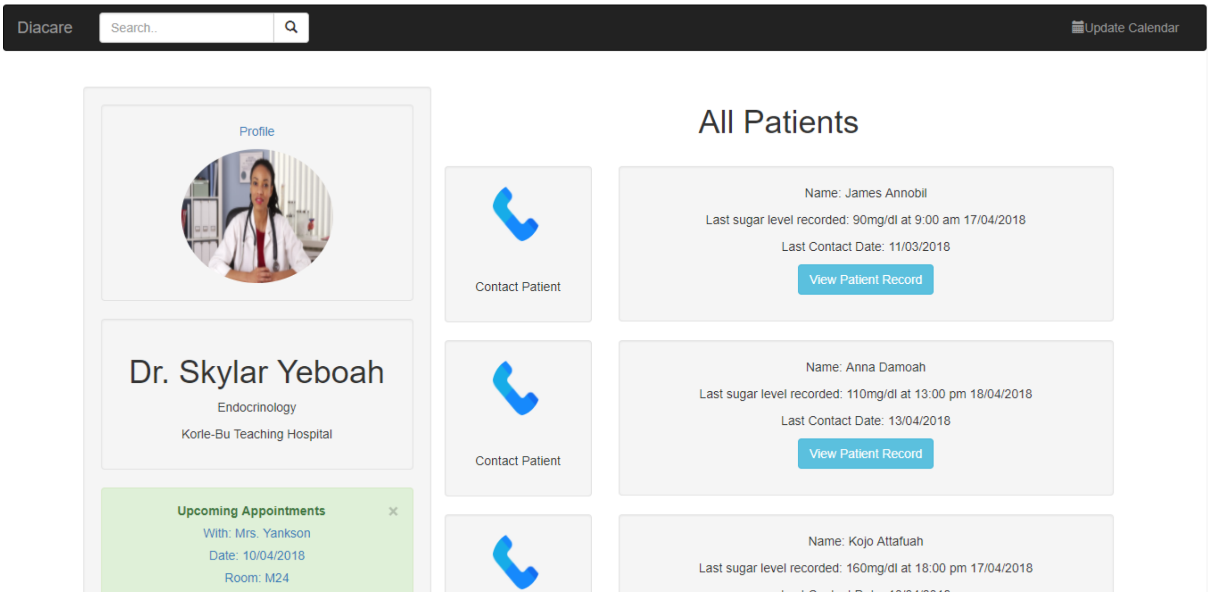

Figure 2.16: Image of doctor's dashboard page used in contacting patients

2.4 Non-Functional Requirements

Non-functional requirements are normally used to edify or enhance the functional requirements defined, to ensure that the system works at its best. The functional requirements pertaining to this project are listed below;

- **Availability**

  The phone application must be available to its users always through a reliable internet connection.

- **Usability**

  The users should find it easy using and navigating the application to obtain their desired results. The user interface should be interactive, well designed, and make use of icons and symbols for easy identification and understanding.

- **Reliability**

  The users should be able to rely on the app to perform its intended tasks without errors. They should also be able to rely on the feedback obtained from this application to perform life-style and health decisions

2.5 System Requirements

System requirements define how the system provides certain user requirements. The four main user tasks for this project are (1) Inputting meals, (2) Viewing sugar consumption levels, (3) Receiving meal suggestions, (4) Scheduling an appointment.

Table 2.3 below shows the system requirements necessary for the system to achieve the four main tasks.

Table 2.3: System Requirements

Description	Requirements
Graphics	Smartphone Screen
Application Server	Apache
Database	MySQL
SMS Server	Npontu
Browser	Chrome, Safari, Opera, Firefox

2.6 Project Scope

The scope of this project is heavily constrained by time and resources, and in the event of further development, the future works listed below can be developed;

- Synchronize with external health tool such as Fitbit to monitor heart rate
- Use the smart phones in-built pedometers to track steps the user takes in a day
- Enable the user to enter their daily exercise routine
- Plan and suggest daily exercise routines for the user
- Track the sugars that the user burns in their daily activities
- Enable the user to input their meals using pictures

CHAPTER 3: HIGH-LEVEL ARCHITECTURE

3.1 Overview

This project makes use of a three-tier client-server architecture as shown in Figure 3.1. A client-server architecture has a client side, a database server, and an application server forming the its three tiers. The application server tier can be divided further into three layers which are the presentation layer, the application layer, and the data layer. The presentation and application layers connect to the web application server tier, while the data layer connects to both the database server tier and the web application server tier. This is because the data layer acts as an intermediary or bridge between the application server tier and the database server tier. The data layer carries information from the application, and stores it in the database.

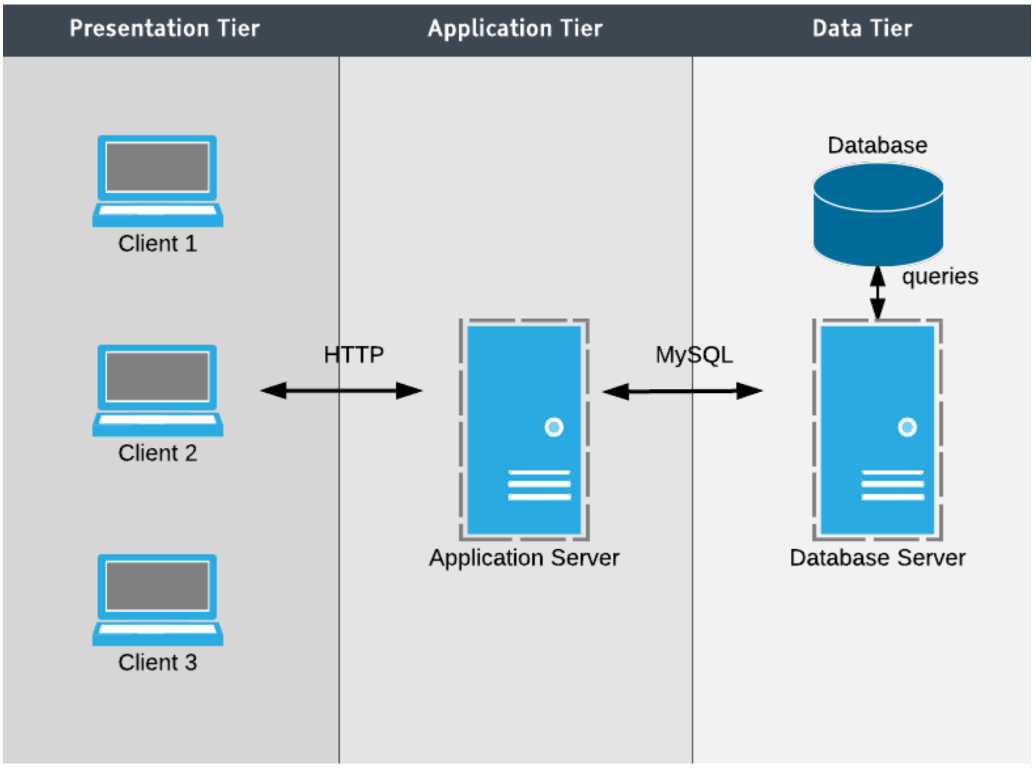

Figure 3.2: DiaCare app system architecture

The three layers are explained into detail below;

- The presentation layer deals with the organization of the front-end view. It is responsible for arranging features of the web page to meet the design as coded in the HTML code. It manages the view of the application as seen on various devices with different screen sizes and resolutions. The presentation layer runs on the browser of the client, but is generated by the application server.
- The application layer runs directly from the application server. It processes all the components required by the application such as sessions, data frames and IP addressing, and binds these components into one unit which is considered as the robust application.
- The data layer deals with storing all user information acquired from the application server into the database on the database server. Without the data layer when users enter their data into the app, the data will be lost as soon as the app is closed. The data layer helps store the required information from the app into the database.

This project used this architecture because of its easy adaptation to all types of devices. This chapter explains into detail the structural organisation of the presentation layer, the application layer, and the data layer, and the functions they perform when it comes building the DiaCare app.

3.2 Presentation Layer

The presentation layer manages how the user interface of the DiaCare app is organized. It deals with how the components of the user interface are represented and viewed by the user. The two main users of the DiaCare app which are the patient and the doctor will use this app for data entry, visualization or viewing, and communication thus the salient features the user interface must offer the users of this application include;

- Visualization of important user information which in this case is the graph of the sugar consumption level of the patient
- Easy data entry using well designed forms with appropriate labels
- Fast responses on tasks performed with the app. For example, a user should receive a toaster notification on whether a task was successful or erroneous

To elaborate on the structure of the presentation layer, this section will analyse it in terms of user interface, navigation, and communication. This project will consider these three areas as very important to the robust functioning of the presentation layer of the DiaCare app.

3.2.1 User Interface (UI)

The user interface mainly refers to the layout of the page and how elements such as buttons and icons are positioned to enhance the user's interaction with the app. A key factor in designing a good UI is consistency. With the DiaCare app, there is a consistency in the design across all the pages from both the patient's side and the doctor's side. The forms used across this application have a consistency in button colours, placeholders, modals and labels to enhance the input collection strategy of the application. See screenshots of system pages in *Appendix B* which show the UI theme of the app.

3.2.2 Navigation

The navigation of the app should be easy for the user. The user should have no difficulty in using the app to perform tasks such as inputting a meal from start to finish. To ensure easy navigation of the app all the linked pages are listed on the sidebar which is available on every page of the app. This will aid the user in easily locating all the pages they wish to access.

3.3.3 Communication

Communication deals with the user's interactions with the app. It encompasses all toasters and alerts that provide the user with feedback on the status of actions performed. It involves actions such as registration, logins, and SMS requests. To perform these functions, the system uses AJAX calls to communicate between the HTML file and PHP file which sends or requests information from the database. AJAX is popularly known for its fast response time which is achieved because of the reduced amount of data exchanged between the app and the web server.

3.3 Application Layer

According to Microsoft, the application layer contains application logic which deals with retrieving, processing, transforming and managing application data (Microsoft, 2009). These activities when narrowed down to the context of this project will include;

- Visualising the sugar consumption levels of the patient
- Retrieving and displaying data from the database for the user
- Capturing data from forms filled by the user and sending it to the database to be saved

The main application logic used in building the DiaCare app can be divided into the PHP code which is used for loading from and storing information in the database, and the JavaScript library from Chart.js used to generate the graph of the patient's sugar consumption.

3.4 Data Layer

The data layer deals with transferring the required information needed to access the database to service user requests. This project makes use of MySQL database, and to access this database, PHP files are used. The required information needed to access the database is passed using the PHP files. Every PHP file contains;

- database connection parameters such as database username, and password
- database connection and query syntax
- table names and column names to show where the information should be stored in the database

This section describes into detail the database design.

3.4.1 Database Design

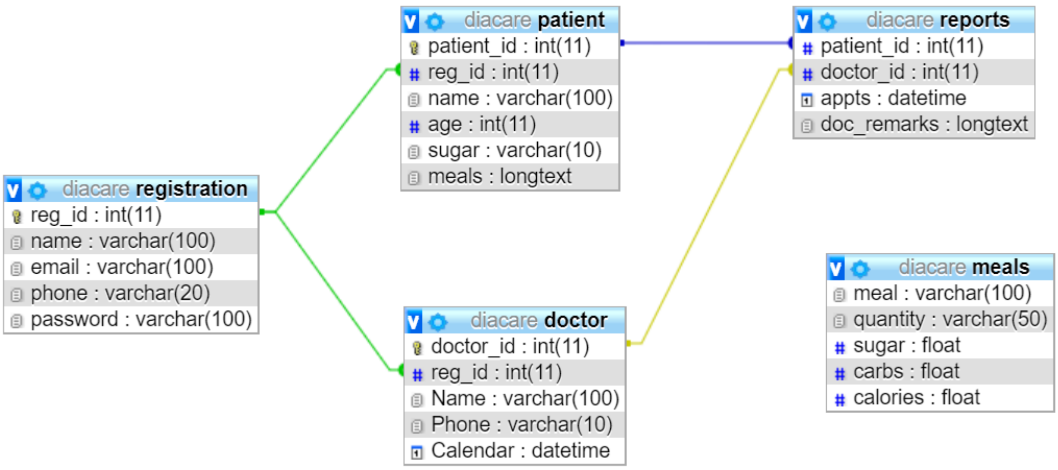

Figure 3.2: Diagram of DiaCare app database

This project made use of a database called Diacare with five tables namely; registration, patient, doctor, meals, and reports. These tables are explained below.

- REGISTRATION table stores all user's data such as name, phone number, email and password. These are the required credentials needed to gain access to the app through login
- PATIENT table stores the health information of the patient, just as a typical patient record file would in a hospital. This table stores the patient's name, unique ID number as a primary key, the patient's age, the patient's sugar level, and the meals the patient consumes.
- DOCTOR table stores the doctor's information which includes their names, phone numbers, and their calendars
- REPORTS table logs all communication between doctors and patients. Here all appointments whether past or pending will be stored, including all the doctor's remarks, suggestions and prescriptions for the patient as well
- MEALS table stores a list of all meals which are high, low or moderate in sugar content. This is the table used in suggestion meals for the patient to stabilize their sugar levels

In creating this database, the types of information that the users of the app will typically request for or like to save for future decision making were considered. The information the app would require in making decisions to favour the functionality of the app were also put under consideration. The information was then grouped under tables and columns which had primary keys to make finding and retrieving information very easy. These primary keys double as foreign keys in other tables to help create a relationship between the tables and prevent duplicate entries or redundant information. In creating this database, integrity rules were enforced to ensure that (1) A primary key was never null, and (2) foreign keys always referenced primary keys in the parent table. Additionally, parameters were used in the SQL queries as well as try and catch blocks to help handle SQL injection attacks.

CHAPTER 4: IMPLEMENTATION

4.1 Overview

The DiaCare app is a phone app that allows a diabetic to track their consumption of sugar. The app allows users to input their daily meals, computes the level sugar in each meal and updates a bar graph with the obtained values. This application enables users to visualise sugar consumption levels through the graph. The app also allows for the remote delivery of healthcare counselling since doctors can also log in to view the sugar consumption levels of patients and contact them to discuss important health changes and decisions.

This chapter focuses on describing the implementation of the app in terms of the three layers which are the application layer, the presentation layer, and the data layer. The Rapid Application Development Model was used in the implementation of the layers because it allowed for development of the layers to be developed in parallel as if they were individual projects, and when complete were combined to form the finished product for the overall project.

4.2 Application Layer

- **Tools**

  The tools used in implementing the application layer included JavaScript and JQuery libraries for the client-side scripting, and PHP for the server-side scripting. These tools were used because they form a widely acclaimed stack for web development.

- **Techniques**

  Some of the scalability techniques used in developing this app include caching from the HTML local storage which reduces the response time of tasks requested by the user.

This reduces latency and increases response time of tasks. The modularity techniques used included storing the code in folders such as CSS, JavaScript, HTML, src, which allow for easy location for re-use, and easy modification.

- **Tasks**

  The main tasks implemented are, inputting meals, visualizing patient's sugar consumption level using a bar graph, and scheduling appointments with doctors. Other sub tasks include storing and loading information from the database.

4.3 Data Layer

- **Tools**

  For the data layer the tools used included the MySQL database from the XAMPP localhost server. The server-side scripting was done using PHP files which include SQL connection code, queries and database parameters. These tools also form part of the XAMPP stack which is commonly used in web development.

- **Techniques**

  Using PHP files to connect to the database ensured that if changes happen in the database, only the PHP files will be affected. Minor changes have to be made to a specific section of the code if the need arises. This is as a result of employing the object - oriented approach in the implementation of this code which eliminates the need to alter the entire code when minor changes have to be made.

- **Tasks**

  Designing and creating the schema for the database is the main task involved in the data layer.

4.4 Presentation Layer

- **Tools**

  The tools used in the implementation of the presentation layer are HTML 5, JavaScript jQuery, AJAX, CSS and the Bootstrap library. The presentation layer is also hosted on the XAMPP server.

- **Techniques**

  Employing the use of the bootstrap framework made the development of the front-end very rapid. The framework offered pre-built buttons, forms, and navbars which cut down the implementation time, and introduced a form of consistency in the UI.

- **Tasks**

  The main tasks included in implementing the presentation layer were the design of a consistent theme across all pages of the app. It also included creating the design of pages to match their required purpose by strategically positioning the elements on the page to make understanding and navigation easy for the user.

CHAPTER 5: TESTING AND RESULTS

Testing is a stage of software engineering where the developed product is run iteratively with the required input, to check if it produces the desired output. The aim of testing is to detect errors in the software, and to make sure that it meets its requirements (Sommerville, 2010). This project used unit testing, system testing, and user testing in it testing phase. Unit testing and system testing were applied in testing the three layers which are the presentation layer, application layer, and the data layer.

See *Appendix A* for a list of test cases and results.

5.1 Presentation Layer Testing

In the presentation layer, the flow of the pages was tested to ensure that all buttons function as they should, the pages link to the correct pages and all hyperlinks work. The forms were also tested to check that they submit the information that has been entered into the database, and manual tests to check for error messages for wrong input. At the basic level automated tests in the HTML code checks for unclosed tabs, misspelt elements and duplicate id's.

- **Syntax Test**

  Using an online HTML validator, the HTML syntax was tested to ensure that they conform to the HTML 5 standards. This test helps to arrange elements in the right order.

- **Forms**

  All the forms in the app were tested to ensure that the right labels and placeholders were positioned with the right input fields. The database was checked after each submission to ensure that the data was submitted in the correct format. Also, error testing was checked by entering wrong information into the forms and checking the error messages which popped up.

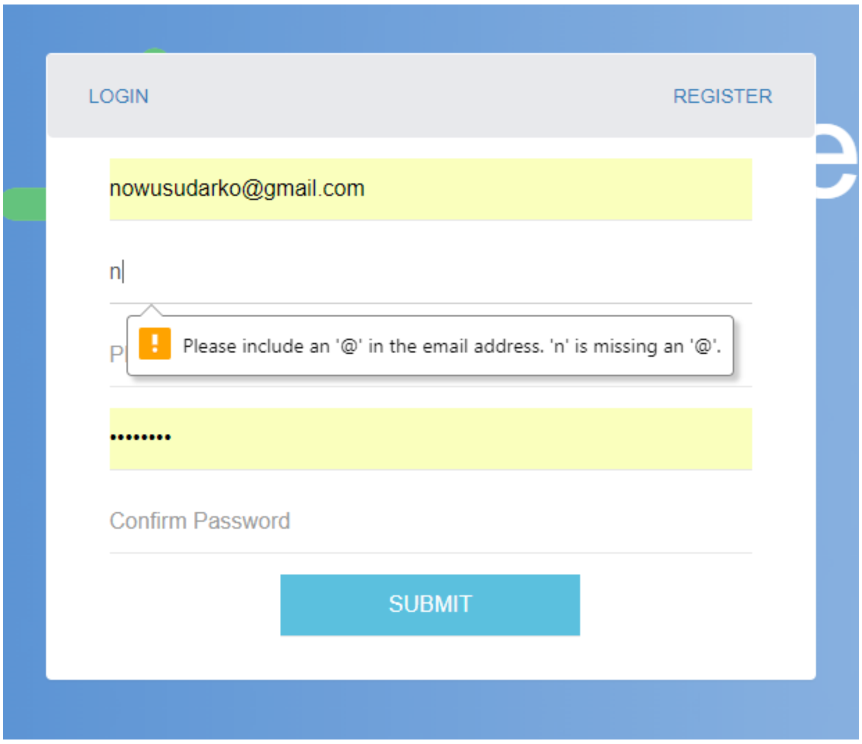

Figure 5.1: Screenshot showing error handling of wrong login activity

5.2 Application Layer Testing

The main tests were performed on the AJAX calls and the functions written in JavaScript. Required input was submitted to the functions and the if the output matched the expected output, then the test was successful.

5.3 Data Layer Testing

Testing in the data layer involved testing the Database Access Object (DAO), the connection to the database, the queries and finally the schema.

- **DAO Testing**

  Code to connect to the database was written in a PHP document, and the response was recorded. If the response for a positive output was obtained, then it means the connection to the database was successful.

```php
<?php
  include_once('../Server/adb.php');

  class DAOTest extends adb
  {
    function adb(){}

      public function testConnection(){
        if($this->connect()==true){
          echo "Database connection succesful";
        }else{
          echo "Failed to connect to database";
        }
      }

  }
?>
```

Figure 5.2: Testing DAO database connection

- **Schema Testing**

  Before data is sent to the database, the tables and columns are tested to check that they exist, and their structure is set up correctly to accommodate all incoming input. Testing the schema ensures that information is saved and retrieved correctly from the database.

- **Queries**

  To test queries, the required parameters were passed to the query and if the desired information was retrieved, then the query was correct.

```php
try {
    //connect to the database
    $con = mysqli_connect("localhost", "root","","diacare");


    $name = $_POST["name"];
    $email = $_POST["email"];
    $phone = $_POST["phone"];
    $password = $_POST["password"];



    // our SQL statements
    $query = "INSERT INTO registration (name,email,phone,password) VALUES ('$name', '$email', '$phone' , '$password')";
    $result = mysqli_query($con,$query);

    if ($result) {
        sms($phone, "You have successfully registered your account with DiaCare");

        echo "Successfully registered";
    }
    else{
        echo "Error ".mysqli_error($con);
    }

    }

catch(Exception $e)
    {
    // roll back the transaction if something failed
    //$conn->rollback();
    echo "Error: " . $e->getMessage();
    }
```

Figure 5.3: Screenshot of error handling of queries before sending data to the database

5.4 User Testing

User testing was done using the cognitive approach. Ten diabetics from the Korle-Bu Diabetic Centre who used android smartphones who were selected at random and three diabetic students chosen by the Ashesi Health Centre tested the app. These users were asked to perform the four main functions of the app which are; (1) Inputting meals, (2) Viewing sugar consumption levels, (3) Receiving meal suggestions (4) Scheduling an appointment.

The participants had to use the app over three days. Two nurses, one from the Ashesi Health Centre and the other from the Korle-Bu Diabetic Centre were allowed to access the patients' records over the three days, to monitor their vitals. The nurses could navigate the app, and monitor patients' sugar consumption and eating habits well. The patients found the record keeping and suggestion of meals very useful. They were able to visualise their sugar consumption levels using the graph, and adjust their eating habits accordingly. However, the patients asked for a call functionality to replace the SMS functionality used in contacting their doctors to schedule appointments. Overall, the patients found the app very useful and personalized to fit the Ghanaian culture.

CHAPTER 6: CONCLUSION

The aim of this project was to design and implement a phone application which will allow diabetics to track the amount of sugar they consume in their diet.

The user requirements demanded that both doctors and patients should be allowed to log into and use the app. Both users should be able to view the sugar consumption level of the patient, and schedule appointments. The users should be able to retrieve patient data whenever necessary. The application fulfils these requirements however there is room for expansion and improvement of the app, such as connection to a peripheral Lancet Device to read patients' blood sugar levels.

Limitations of the finished product include its inability to be used across iOS and Windows platforms because it is an android app.

To improve the application, future development can include the following:

- Connection to Lancet Device to read patient's blood sugar directly
- Connection to a fit bit to read patient's heart rate
- Using the pedometers installed in the patients' phone to track steps taken in a day
- Read calories and sugars used up in daily routines
- Suggest work-out plans

This project introduces a way of managing chronic diabetes through using the DiaCare app. This application encourages self-management of diabetes, provides a reliable medium of storage of patient data, and facilitates remote consultation. Although there is still room for improvement, this app will go a long way to revolutionize healthcare delivery in Ghana.

References

Abbey, E. (2013). Korle-Bu Diabetic Centre Burnt. *Modern Ghana*. Retrieved from https://www.modernghana.com/news/485278/korle-bu-diabetic-centre-burnt.html

Afarikumah, E. (2014). Electronic Health In Ghana: Current Status And Future Prospects. *Online Journal of Public Health Informatics*, *5*(3). http://dx.doi.org/10.5210/ojphi.v5i3.4943

Chetley, A. (2006). *Improving Health, Connecting People: The Role of ICTs in the Health Sector of Developing Countries* (7th ed.). Infodev. Retrieved from http://www.infodev.org/infodev-files/resource/InfodevDocuments_84.pdf

Mensah, L. (2015). *Young Ghanaian Tech Enthusiast Creates App to Ward off Diabetes*. *Global Press Journal*. Retrieved 18 April 2018, from https://globalpressjournal.com/africa/ghana/young-ghanaian-tech-enthusiast-creates-app-to-ward-off-diabetes/

Ouma, S., & Herselman, M. (2008). *E-health in Rural Areas: Case of Developing Countries* (pp. 1-7). World Academy of Science, Engineering and Technology. Retrieved from https://pdfs.semanticscholar.org/342e/c3adb2b9085ddb76093535ca53b0c0ef9001.pdf

Sood, S., Mbarika, V., Doarn, S., Prakash, N., & Merrell, R. (2007). *What Is Telemedicine? A Collection of 104 Peer-Reviewed Perspectives and Theoretical Underpinnings | Telemedicine and e-Health*. *www.liebertpub.com*. Retrieved 16 April 2018, from http://doi.org/10.1089/tmj.2006.0073

Vogel, L., Goertz, L., Shani, S., Boots, M., Dorval, L., & Wang, N. (2016). A Mobile-based Healthcare Utilization Assessment in Rural Ghana. *Procedia Engineering*, *159*, 366-368. http://dx.doi.org/10.1016/j.proeng.2016.08.394

Bibliographies

Arsand, E. et al. "Mobile Phone-Based Self-Management Tools for Type 2 Diabetes: The Few Touch Application." Journal of Diabetes Science and Technology 4.2 (2010): 328-336. Web. 9 Oct. 2017.

Erik Arsand is a Professor in e-health and researches around how to use mobile technologies to improve self-management, mainly using diabetes as the case. Apart from this research paper, his other works include reviewing features of mobile diabetic applications. In this article Mobile Phone-Based Self-Management Tools for Type 2 Diabetics, he explores how such tools can be designed for supporting lifestyle changes among people with type 2 diabetes, using a group of 12 patients during a 6-month period. The study used focus groups, interviews, feasibility testing, questionnaires, paper prototyping, and prototyping of both software and hardware components. A mobile phone-based system called the Few Touch application was developed for this project, with features such as blood glucose (BG) meter, and a software for recording food habits and providing feedback on how users perform in relation to their own personal goals. After the 6-month period, feedback showed good usability of the tested system, with several of the participants adjusting their medication, food habits, and/or physical activity. This article gave me a very practical point of view on how type 2 diabetics use and adjust to these health tracking apps. It brought to light, the most important features, and the ones which could be changed to make it more effective.

Chomutare, T., Fernandez-Luque, L., Årsand, E., & Hartvigsen, G. (2011). Features of Mobile Diabetes Applications: Review of the Literature and Analysis of Current Applications Compared Against Evidence-Based Guidelines. *Journal Of Medical Internet Research*, *13*(3), e65. http://dx.doi.org/10.2196/jmir.1874

Taridzo Chomutare, one of the authors of this journal graduated from UiT The Arctic University of Norway with a PhD in Computer Science. With his master's degree in telemedicine and e-health from the UiT, he has worked with telemedicine for nearly 10 years, developing mobile applications for people with diabetes. One of the assisting authors, Erik Arsand is a Professor in e-health and does research around how to use mobile technologies to improve self-management, mainly using diabetes as the case. Luiz Fernandez – Luque is also an assisting author who graduated from his PhD at the University of Tromso (Norway) and Computer Engineering at the University of Sevilla (Spain). He currently works as an eHealth Researcher at QCRI (Qatar Computing Research Institute-Qatar Foundation). The fourth assisting author is Gunnar Hartvigsen, PhD, who is an adjunct professor at the Norwegian Centre for E-Health Research, and is the head of the research group for medical informatics and telemedicine (MI & T) at the Department of Computer Science, Faculty of Science and Technology at UiT The Arctic University of Norway. In their research article, Features of Mobile and Diabetes Applications, studied the salient features of mobile applications for diabetes care, in contrast to clinical guideline recommendations for diabetes self-management. Their research showed that, a wide selection of mobile applications seems to be available for people with diabetes but are gaps between the evidence-based recommendations and the functionality used in study interventions or found in online markets. Also, no studies were found to evaluate social media concepts in diabetes self-management on mobile devices, and its potential remains largely unexplored. This article lacked depth and gave little but relevant information on the diabetics and how they use the app.

Eng, D., & Lee, J. (2013). The Promise and Peril of Mobile Health Applications for Diabetes and Endocrinology. Pediatric Diabetes, 14(4), 231-238. http://dx.doi.org/10.1111/pedi.12034

Dr. Donna S Eng is an Endocrinology Specialist in Grand Rapids, Michigan. She graduated with honors from University of Michigan Medical School in 2007. She has more than ten years of diverse experiences, especially in endocrinology and paediatric medicine. The assisting author for this article, Dr Joyce Lee, MD, MPH, is an Endocrinologist and Professor at the University of Michigan Medical School and School of Public Health. Her research is centred on exploring the use of emerging technologies for improving outcomes in type 1 diabetes. In their research paper titled The Promise and Peril of Mobile Health Applications for Diabetes and Endocrinology, these authors conducted a review of apps focused on endocrinology. Their findings stated that a higher number of relevant applications were found on the iPhone app store, as compared to the android marketplace. They also found that majority off the applications needed data which had to be manually input by the user; only two applications used glucometers which were attached to the phones. This research paper also explained that, although mobile health apps could potentially improve chronic disease care, they face some challenges including lack of evidence of clinical effectiveness, lack of integration with the health care delivery system and showed the need for formal evaluation. The research paper did not show an in-depth analysis of the effects off the apps on its users, however, it gave very relevant statistics of the types of apps, their functions, and their use on the market.

Rao, Anoop et al. "Evolution of Data Management Tools For Managing Self-Monitoring Of Blood Glucose Results: A Survey Of Iphone Applications." Journal of Diabetes Science and Technology 4.4 (2010): 949-957. Web. 9 Oct. 2017.

Anoop Rao is part of the faculty at Stanford University School of Medicine, Palo Alto, CA, Neonatology. He has an MSc from Massachusetts Institute of Technology and a Bachelor of Medicine and Bachelor of Surgery degree from Manipal Academy of Higher Education. In this article, Anoop Rao reviews apps and software to find out if electronic self - monitoring of blood glucose (SMBG) data management and sharing tools for the PC and smartphones may help in reducing the effort to manage SMBG data. Studies have indicated that sharing of self-monitoring of blood glucose (SMBG) data and subsequent feedback from the health care provider (HCP) can help achieve glycaemic goals such as a reduction in glycated haemoglobin. Through observation of patients who use SMBG devices, it was discovered that the use of the WaveSense Diabetes Manager allowed them to complete SMBG data and relay tasks faster than other applications. This research showed that, being able to record, analyse, seamlessly share, and obtain feedback on the SMBG data using an app might potentially benefit patients. This article presented me with tangible information on how tracking blood glucose levels could potentially benefit the technologically savvy patient, however, the article was very brief, and the details and statistics were shallow.

APPENDIX

Appendix A: Test Cases and Results

ID	Test Case	Input Data	Expected Results	Actual Results	Pass/Fail
1	Test if user logs in successfully	Email, password	See dashboard for patients (if patient) or dashboard for doctors (if doctor)	See dashboard for patients or doctor	Pass
2	Test login for unregistered user	Unregistered email and password	Login failed error message	Login failed error message	Pass
3	Test user login for valid email and empty password	Valid username	Error: “Password cannot be empty”	Error: “Password cannot be empty”	Pass
4	Test user login for valid password and empty email	Valid password	Error: “Please enter an e-mail in the format email@domain.com”	Error: “Please enter an e-mail in the format email@domain.com”	Pass
5	Check all links in patient dashboard page		Links lead to pages required	Links lead to expected pages	Pass
6	Check all links in doctor dashboard page		Links lead to required pages	Links lead to expected pages	Pass
7	Test if input meals form submits actual data	Filled form with meals and quantities	Successful submission of data	Successful Submission of data	Pass
8	Test if nutritive values are computed right	Filled form with meals and quantities	Red bar graph of sugar consumption level if above 180mg/dl or below 80mg/dl, blue bar graph of sugar consumption level if between 80-180mg/dl	Red bar graph of sugar consumption level if above 180mg/dl or below 80mg/dl, blue bar graph of sugar consumption level if between 80-180mg/dl	Pass

9	Test if doctor can contact patient	On the doctor's dashboard, click on contact patient button beside patient's information	SMS box is opened for doctor's message to be typed and sent.	Patient receives message	Pass
10	Test if patient can view doctor's calendar	Click on schedule appointment button on patient dashboard page	Doctor's calendar is shown	Doctor's calendar is shown	Pass
11	Test if patient can select date for appointmen t	Click on date on doctor's calendar	Date is blocked out. Success message received. SMS sent to doctor	Date is blocked out, success message received, doctor receives SMS	Pass
12	Generate list of foods low in sugar	Click on suggest meals button on patient dashboard	List of meals with low caloric count is loaded	List of meals with low caloric count is loaded	Pass
13	Generate list of foods high in sugar	Click on suggest meals button on patient dashboard	List of meals with high caloric count is loaded	List of meals with high caloric count is loaded	Pass
14	Generate list of meals with moderate caloric count	Click on suggest meals button on patient dashboard	List of meals with moderate caloric count is loaded	List of meals with moderate caloric count is loaded	Pass

Appendix B: Test Visualization Proving Implementation

The images and figures shown here are to demonstrate the functionality of the application.

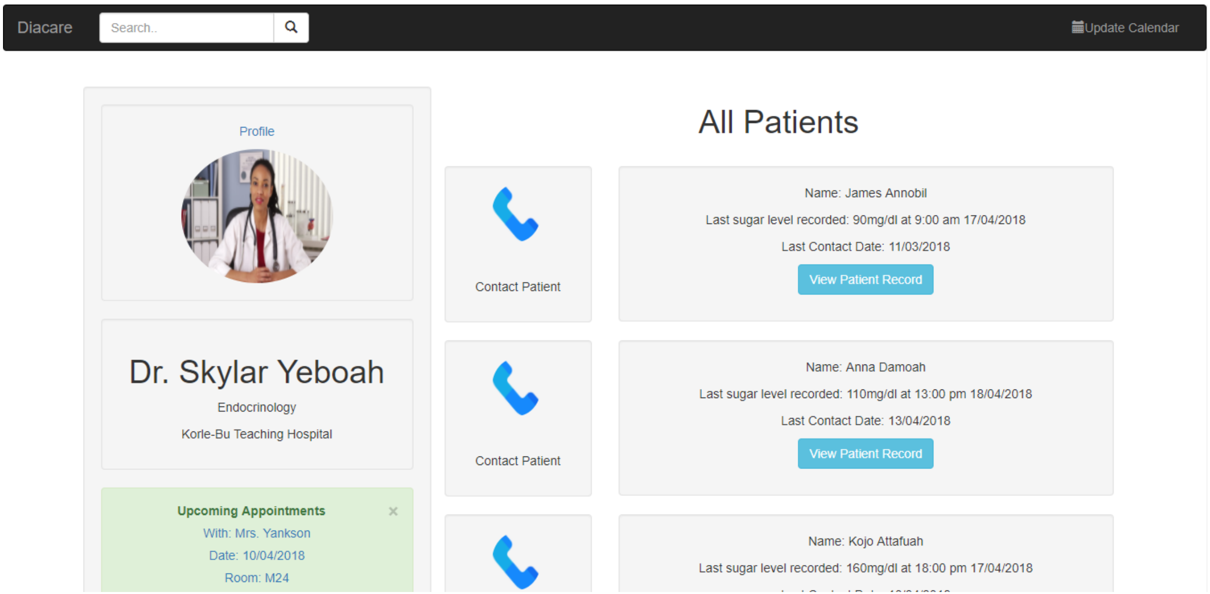

Image of doctor's dashboard

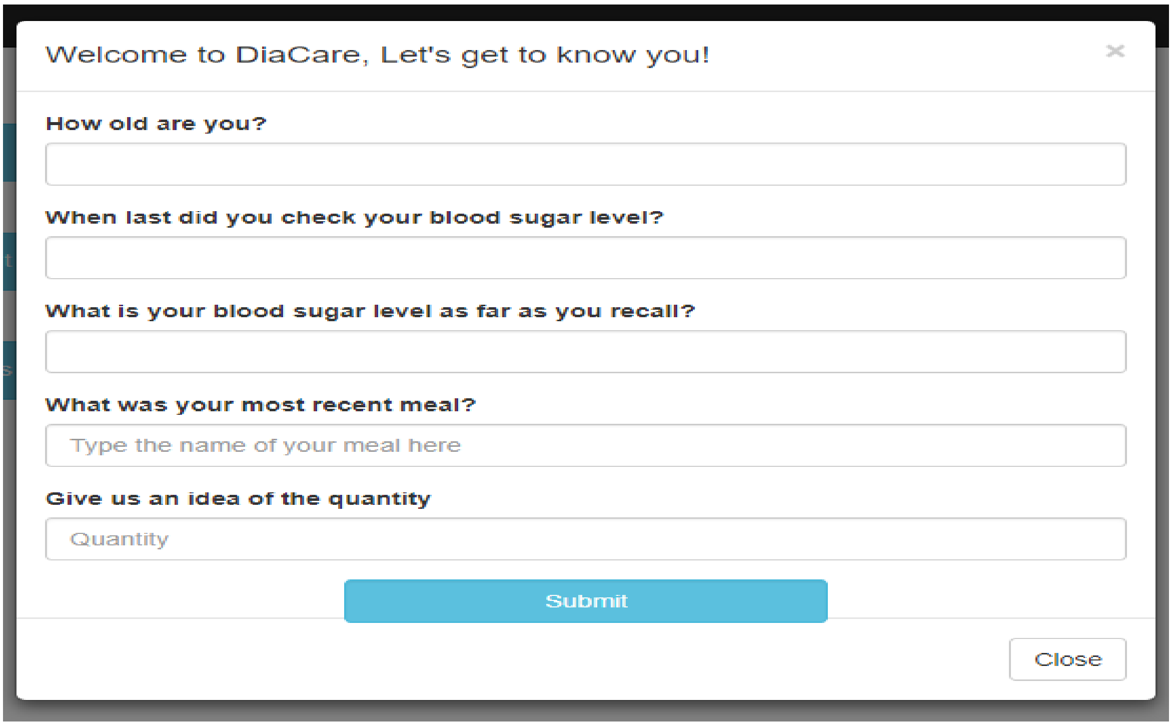

Image of modal which takes user information for the first time of using the app

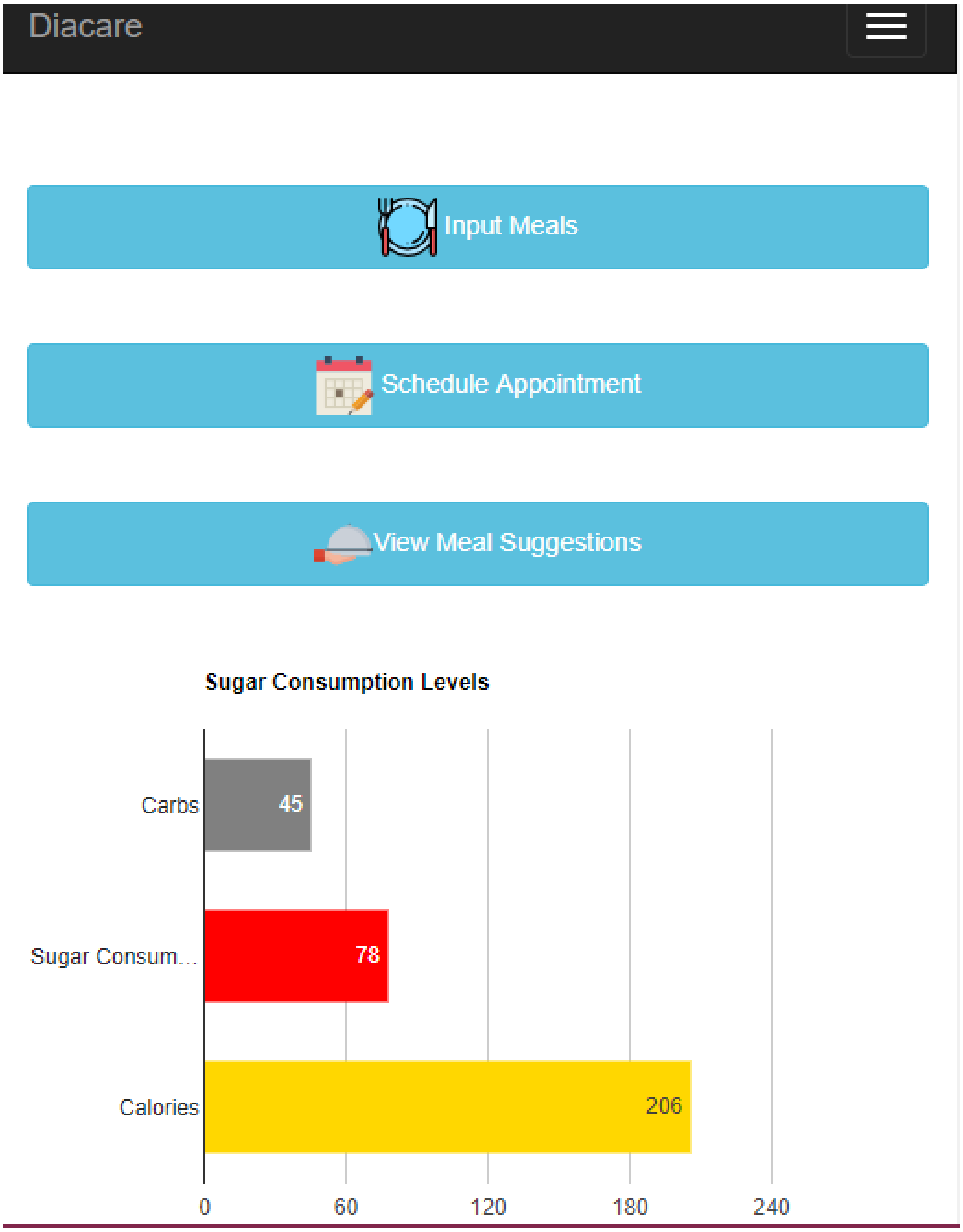

Image of patient dashboard with unsafe sugar consumption level

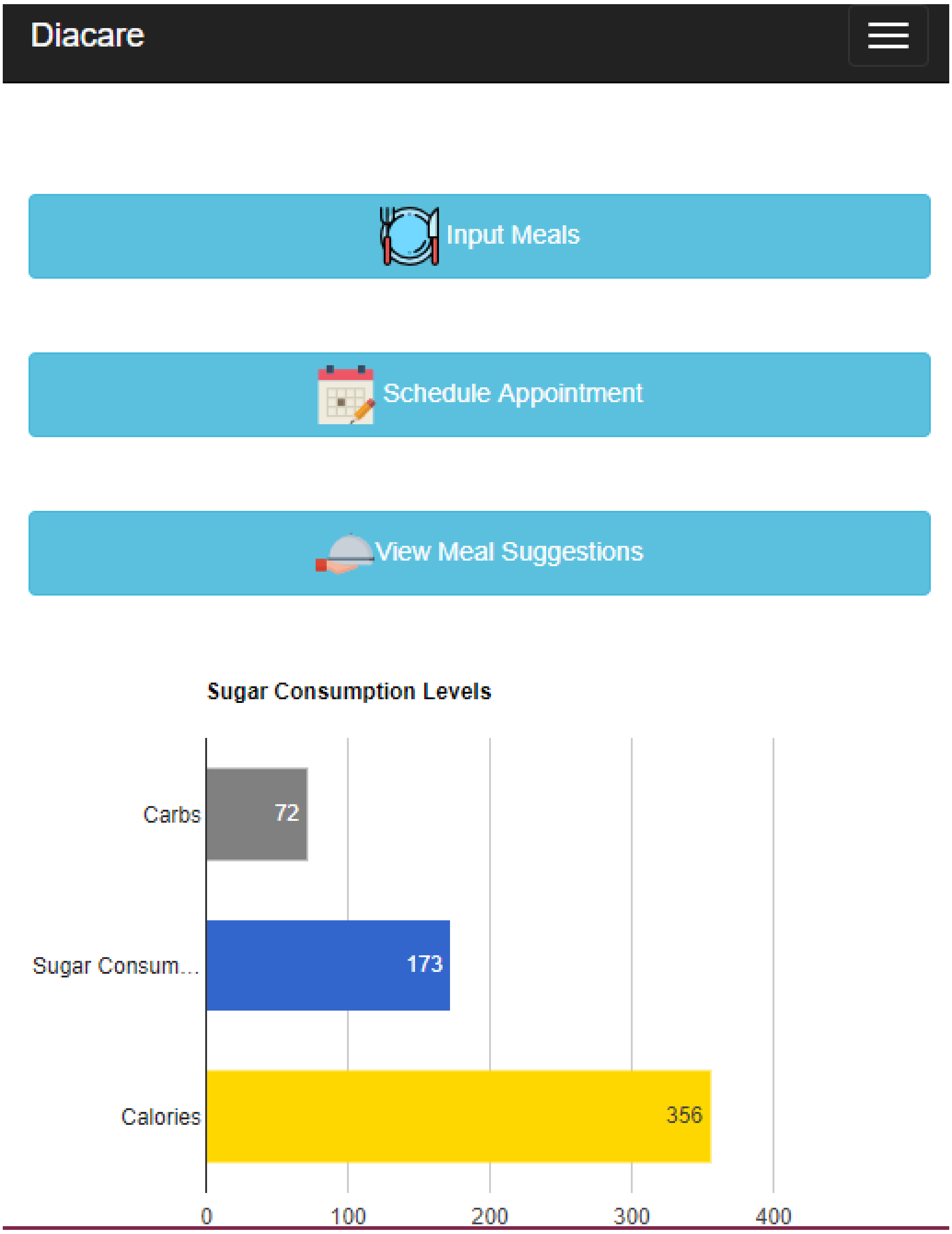

Image of patient dashboard with safe sugar consumption level

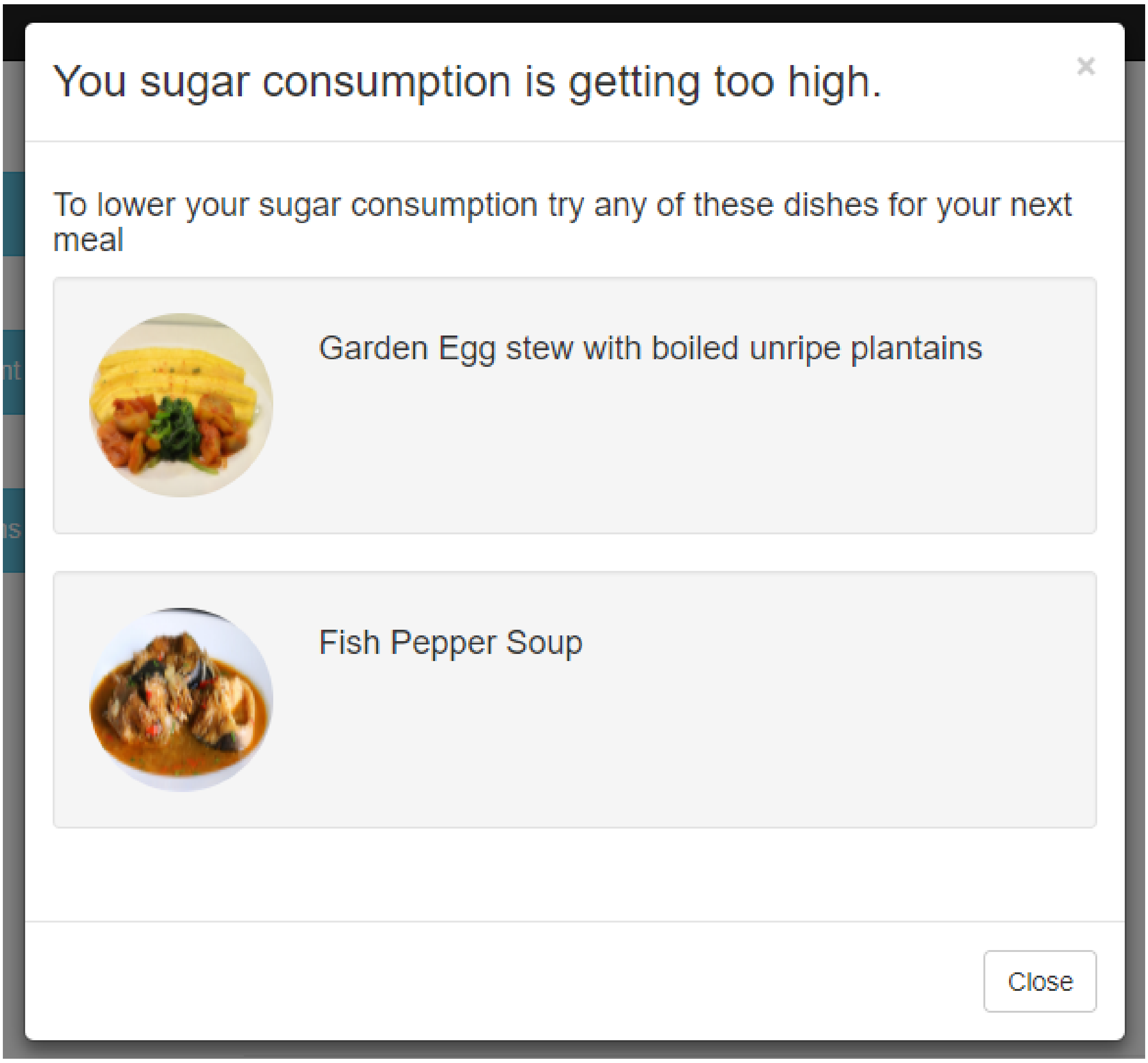

Image of meal suggestion of dishes with low sugar content