DEVELOPMENT OF AN ONLINE DOCTOR APPOINTMENT BOOKING SYSTEM FOR MOUNTAIN TOP UNIVERSITY HEALTH CENTER

By

OLUJULO, OLUWATOMIWA ENOCH

18010301060

A PROJECT SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE AND MATHEMATICS, COLLEGE OF BASIC AND APPLIED SCIENCES, IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE

2021

DECLARATION

I hereby declare that this project has been written by me and is a record of my own research work. It has not been presented in any previous application for a higher degree of this or any other University. All citations and sources of information are clearly acknowledged by means of reference.

_____

OLUJULO, OLUWATOMIWA ENOCH

_____

Date

CERTIFICATION

This is to certify that the content of thos project entitled, "**Online Doctor Appointment Booking System for Mountain Top University**", was prepared and submitted by **OLUJULO OLUWATOMIWA ENOCH,** in partial fulfillment of the requirements for the degree of **BACHELOR OF SCIENCE IN COMPUTER SCIENCE.** The original research work was carried out by him under by supervision and is hereby accepted.

_____ (Signature and Date)
Mr. J.A BALOGUN
Supervisor

_____ (Signature and Date)
Dr. Matthew O. Adewole
Coordinator, Department of Computer Science and Mathematics

DEDICATION

This project is dedicated to God almighty for the successful completion of my BSc. in Computer Science.

ACKNOWLEDGMENTS

I owe my profound gratitude to God Almighty who gave the strength, wisdom, and courage, divine help, and provision to me from the beginning to the completion of this work. I express my gratitude to my major supervisorS, Prof. P.A. Idowu and Mr. J. Balogun, for their guidance and support in ensuring the successful completion of this research. God bless you richly sirs. I sincerely appreciate the Dean, College of Basic and Applied science for his fatherly advice, guidance, and teachings. My heartfelt gratitude goes to the Head of Department, Computer Science and Mathematics – Dr. Adewole, and all other members of staff of the Department of Computer Science: Dr. (Mrs.) F.A. Kasali, Mr. I.O. Ebo, Dr. Igiri, Miss Anabelle, Late Dr. Oyetunji, Mr. Falana.

I will forever be grateful to my parents Mr. and Mrs. Olujulo, who sacrificed wealth, time, and other resources for the sake of my success; and my siblings for their consecutive support and prayers.

Furthermore, I must appreciate the University Chaplain and the Chaplaincy unit for their love, care, and prayer for me, and all Mountain Top University colleagues and friends for their prayers and support in one way or the other. God bless you all.

ABSTRACT

The aim of this study was to produce an information system that will provide a means of scheduling appointments with the doctors at the university remotely through a mobile device resulting in the reduction of waiting time at the university health center. This study involved the identification of the user and system requirements, specificaion of the system design with the UML language and implementation of the system.

A review of related works was done to identify the expected behaviour of appointment booking systems. Informal interviews were conducted with the students and staff of the university to elicit user requirements. UML diagrams were used to specify the design of the system. The system's frontend and user interfaces were inplemented using the react-native framework, the backend was implemented using google's cloud firestore, firebase authentication and Node JS.

The implementation results of the system proved that the system could allow patients book doctor appointments with the university healthcenter, allow nurses assign match patients with available doctors, allow doctors report medical data after every appointment and reshedule appointment with patients if there it is required.

This study concluded that using online appointment booking systems would help overcome challenges that are faced in the manual approach and help boost the output of the healthcare facilitators as well as improve customer expirience.

Keywords: Information systems, Health information systems, Online appointment booking systems.

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study.

The health center is a facility in an institution that provides all sorts of healthcare services to respective patients in the institution. It is a necessary feature, an integral part of human life, that is very important to every university as health care services are a necessity and in critical cases, lack of timely access to health care services could lead to death, (Akinode J.L, 2017). The most significant workers are the doctors and the nurses. An informal survey of the mountain top university showed that it has a total of two doctors, about 10 nurses, and an estimated value of 2,000 students and academic staff. This estimation is minus the non-teaching staff. The patient-to-doctor ratio is about a thousand patients to a single doctor or 250 patients to a nurse. The health center is understaffed so issues like long waiting times or work overloading of doctors and nurses are inevitable. These problems are not occurring at the mountain top university health center alone. According to the world health organization (W.H.O), the estimated waiting time of patients across the world is about three hours, (Jude Altema, 2020). Waiting time simply means a period in which one must wait for a specific action to occur after that action is requested or mandated, (Idowu & Adeosun, 2014). In the context of health-related service provision, waiting time refers to the total amount of time it takes a patient to get the services of a doctor when they visit the health center. The total waiting time is usually a total of the service wait time, the time it takes the nurses at the health center to attend to patients, the total time it takes them to search for the patient's hospital files, to take vitals of the patient and then forward the files to the doctor that's supposed to attend to the patients and the queue wait time, which is the total amount

of time the patients at the health center spend waiting to be attended to by the nurses and then waiting to be attended to by the doctor.

An appointment is an arrangement made by two or more individuals. It's usually at a specific place and at a particular time. An appointment booking system is an application created to automate the process of booking doctor appointments done by patients in the health center. A good appointment booking system would result in increased satisfaction for both the patients (students and staff) and the staffers of the university, and as such, it can be referred to as an important component of the health center, (Clue, 2020).

1.2 Statement of the Problem.

Traditionally, before the arrival of the covid-19 pandemic, there have been prevalent issues of long service and long queue waiting times when patients go to get treated at the university health center. Now, in addition to the already existing problems, there's a pandemic and the health center is one of the most vulnerable institutions as it is where sick people come to get treated. The mountain top university health center has a structural limitation and can effectively contain a maximum of about ten people at a time. A very effective way to reduce long service and queue wait time and effectively curb the spread of the coronavirus pandemic would be to automate the process of booking appointments at the health center. This way, only patients that have been scheduled to see the doctor would be available at a time and it'll help boost the service experience of the patients and the work morale nurses as the patients can now make better use of their time and the workload of the nurses would be effectively reduced.

1.3 Aim and Objectives of the Study.

This study aims to produce an information system that will provide a means of scheduling appointments with the doctors at the university remotely thus eliminating waiting time at the university health center. The objectives of this study are to;

i. identify the requirements of the system.
ii. specify the design of the software.
iii. implement the software.
iv. test the software.

1.4 Proposed Methodology of the Study.

To meet up with the objectives of the study stated above the following methods will be adopted;

a) A discussion will be held with the nurses and doctors at the health center and some colleagues to elicit the functional requirements of the system.
b) Observation of other software solutions that have already been developed will be undertaken to find other functional and non-functional requirements of the system.
c) UML diagrams, like the use case, sequential and architectural diagrams, would be used to graphically specify the design of the system.
d) A cross-platform mobile application framework, React Native, will be used to implement a mobile application for the students and staffers that would use the system.
e) A web-development framework, React, would be used to develop a web application for the health center staff that would use the system.
f) Firebase would be used to implement the database for the system.

g) Alpha and Beta testing would be done by me and some of the target users of the system to ensure that the system meets up to its requirements.

1.5 Significance of the Study.

The primary beneficiaries of the appointment booking system are the students and staff of the Mountain top university. The introduction of an appointment booking system would reduce the amount of time they would normally waste, waiting in line to receive the services of doctors or nurses. The nurses and doctors are also beneficiaries in that their workload would be reduced since they can now spread the patient appointment throughout the entire day. This would in turn boost their work efficiency and morale. Spreading doctor appointments throughout the day would also help to curb the spread of the corona as overcrowding of the health center waiting rooms would be eradicated.

1.6 Scope and Limitations of the Study.

The software will be developed with Mountain Top University in mind although it will have the potential to be incorporated into various universities in Nigeria. The software to be developed would only run on the android and iOS mobile operating system (for the students and staff) and any operating system (for the nurses). Some of the challenges that might be experienced during development include;

i. Slow internet connectivity on the school campus.
ii. Limited time available to completely develop the proposed system.

1.7 Definition of Terms

a API – Initialism of application programming interface. A set of functions and procedures allowing the creation of applications that access the features or data of an

operating system, application, or other services.Distributed Systems – Also referred to as distributed computing, refers to the process of combining the computing power of several machines to simultaneously perform a single computational task in an organized unified way.

b JSON – Javascript Object Notation, is a data format used to represent structured data. It is based on JavaScript syntax.

c Moore’s law - Moore's Law refers to Moore's perception that the number of transistors on a microchip doubles every two years, though the cost of computers is halved. It implies that the compute power of machines increases every year and its costs reduce.

d No-SQL Databases – Any database that doesn’t store its data as relations (in tabular form) is referred to as a No-SQL database. Other storage structures could include graphs, collections, documents, maps, or JSON.

e Operating Systems – The operating system is system software that controls and manages access to hardware resources such as memory, CPU time, disk space, and input and output devices. It also allows concurrent access to these resources with scheduling algorithms.

f Programming language – A set of reserved words and symbols used by computer programmers inside computer programs to instruct the computer on how certain tasks should be performed.

g Prototype – A prototype could be a physical model of a system or the initial release of a proposed system built to study it or test it for required functionalities.

h Software developer – It is a person that develops software. She/he usually writes code to communicate with a computer.

CHAPTER TWO

LITERATURE REVIEW

2.0 Introduction

This chapter consists of conceptual reviews and thematic reviews of related literature that help to understand the factors affecting existing health information systems and conditions for the successful implementation of an online doctor appointment booking system.

2.1 Information Systems

An information system is a system created to collect data, store it, process it and, produce information. Information system design involves the integration of different components like procedures, people, technology, etc. to collect, store and process data, to be used by organizations to simplify some of their activities and facilitate decision making, (Zwass, 2020). Information systems coordinate the flow of and maintain information, and this information can be used by different organizations for different purposes, (The art of service., 2019).

A computer information system usually referred to as plainly a computer with software running on it, is an information system that consists of users and computers that process information, (Zwass, 2020). Most computer information systems are created to support the operational, management, and operational activities of organizations, and the department in organizations in charge of these systems are referred to as information services, (Techopedia, 2011).

2.1.1 Dimensions of an information system

There are different aspects of information. Geeks for geeks, (Geeks for geeks, 2019), specified three dimensions of information systems.

a. Organizational Dimension – Information systems are usually used by organizations. These organizations usually have different norms, practices, and operating procedures that should ideally be embedded within them. They include;
    i. Functional specialties
    ii. Business processes
    iii. Culture
    iv. Political interest groups.
b. Management Dimension – They include the managers and owners of the organization. They are usually responsible for making managerial decisions for organizations and information systems provide them with decision-making tools to help educate the process.
c. Technology Dimension – They include technology used by managers and employees of an organization. It consists of computer hardware and/or software, data management technology, network or telecom technology, etc. Managers of ever-changing operation environments like business organizations, use these tools to cope with change.

2.1.2 Classes of information systems

Lauren Christiansen, (Christiansen, 2021), grouped all information systems into three distinct classes namely;

a. Operational Management Systems – Transaction processing, knowledge management, and office automation systems fall into this category. They are used by the employees of an organization to manage daily activities.
b. Tactical Management Systems – They are used by the managers of an organization. They handle semi-structured data and include management information systems.

c. Strategic Management Systems – The are used by the owners and executives of an organization. The decision support and executive support systems fall into this category.

2.1.3 Components of an information system

Information systems are the integration of components to collect, store and process data to support the decision-making process, (Geeks for geeks, 2019). Information systems have five components; computer hardware, computer software, telecom, databases, human resources, and procedures, (Gregersen, 2019). These components are explained in detail in the following

a. Computer Hardware – It includes all the physical devices used by organizations to process information. Hardware devices are of varying sizes. They range from small mobile devices like smartphones to large immobile systems like supercomputers. Peripherals like keyboards, mouse, modems, network cables, etc. that work with computers are also considered hardware devices. The arrival of the Internet of things, which allows for the communication of data between hardware devices like basic home appliances or cars, computer hardware is gradually becoming a popular feature in modern homes, (Gregersen, 2019). Operating systems alongside processors are also used to facilitate the use of computer hardware devices, (Geek for geeks, 2021). Large organizations also use distributed systems, a network of multiple systems with powerful processors located on different sites. These distributed systems are viewed by the users, through mobile devices or laptops, as one system and details like implementation or number of connected users or current system being used are masked from the user. The bewildering advancement in computer hardware over the years, however, hasn't resulted in unbearable costs. In

fact, the costs of hardware rapidly decreased and the processing speed and storage capacity of storage devices have increased following Moore's law, (Zwass, 2020).

b. Computer Software – A computer software is a set of instructions used to instruct a computer, or a hardware device, to perform a task. The hardware usually needs to be instructed on what tasks to be performed and this is the role of software. Software developers usually specify these tasks using a programming language. Computer software can be divided into two categories. Application software, which could be word or spreadsheet programs, mobile apps, or other software, is usually created to perform specific tasks. Examples of application software include; word processing or spreadsheet applications e.g., Microsoft Office, database tools e.g., XAMP, games like PES 21, bespoke applications built for specific companies, for instance, a software developed to automate the process of booking orders in a confectionary, or generic applications built by organizations and available commercially for use by individuals. Computer software could also be delivered as a service (software as a service), and be accessed from the cloud over the web. The other category, system software, is normally used to control hardware devices like the computer system, to manage data, programs, system resources (i.e., memory, CPU, etc.), and to provide a means by which users can control a system via a graphical interface. The primary system software is the operating system, like the Windows, iOS, Android, macOS, or Linux operating systems. (Zwass, 2020).

c. Telecommunications – Telecommunications are used to network multiple computer systems and facilitate the communication of information between them. Communication over a network could be performed through wired or wireless mediums. Examples of wired media include; coaxial cables, ethernet cables, USB cables, fiber optics, etc. Wireless technology is based on the transmission of radio

waves and microwaves such as Wi-Fi. The network needs of an organization determine the network configuration to be employed. If all the components that need to communicate with each other are on a particular site, local area networks (LANs) are used. Metropolitan area networks (MANs) are used when these components are spread over a mildly large area, a city for instance. For communications over a large area, Wide area networks (WANs) are used. They are usually used by large organizations, with data centers on multiple sites. Peer to Peer networks, with a central control system, are enable mass sharing of data, (Zwass, 2020).

d. Database and data warehouses – A database is storage for a collection of raw unprocessed data to be used by other components of an information system, which is usually retrieved by querying it using a particular criterion, (Gregersen, 2019). Typical examples of a database include customer purchase history or sales records. The contents of a data warehouse could be prehistoric data, or an archive of data that has been collected over a certain period. Data warehouses leverage cheap storage to store very large amounts of data, which can be mined to retrieve information that is useful to an organization. The mined information can be used to gain new customers, better satisfy existing ones, or determine what type of product to be produced. Data in databases or data warehouses can be managed using a database management software, (Geek for geeks, 2021). A database management software (DBMS), is software that provides a method to create, update, store and retrieve data stored in a database. It enables sharing of data between end-users and application programmers or sharing between multiple applications, (Hoffer, Ramesh, & Heikki, 2011). Most database management software use SQL to structure their data. Some example of DBMS includes MySQL, Mongo DB,

Firebase, PostgreSQL, etc. The massive data collections of different forms of data (structured, quantitative, text, etc.), on the web, results in a concept known as big data. The analysis of these large amounts of data is has proven very useful in decision-making scenarios, (Zwass, 2020).

e. Human Resources and procedures – IT personnel are an essential part of the information system. They include programmers, business and systems analysts, and designers, database administrators, computer operators, computer security specialists, etc. Workers in organizations are now being trained to fully utilize information systems. Procedures are documentations used by workers operating or maintaining information systems, or embedded in autonomous software to control them, (Zwass, 2020).

2.1.4 Types of information systems

The majority of the available textbooks generally group information systems as a hierarchy of systems namely transaction processing, management information, decision support, and executive information systems in the afore-stated order as represented in figure 2.1. Although the analogy is still useful, the introduction of modern technology, however, has resulted in the development of new information systems that no longer fit into this original categorization, (Laundon & Laundon, 1988).

Some of the types of information systems are discussed in detail in the following parts of this section.

a. Transaction Processing Systems – A good example of a transaction is the successful sale of a good at a shop. They are used by individuals in the operations

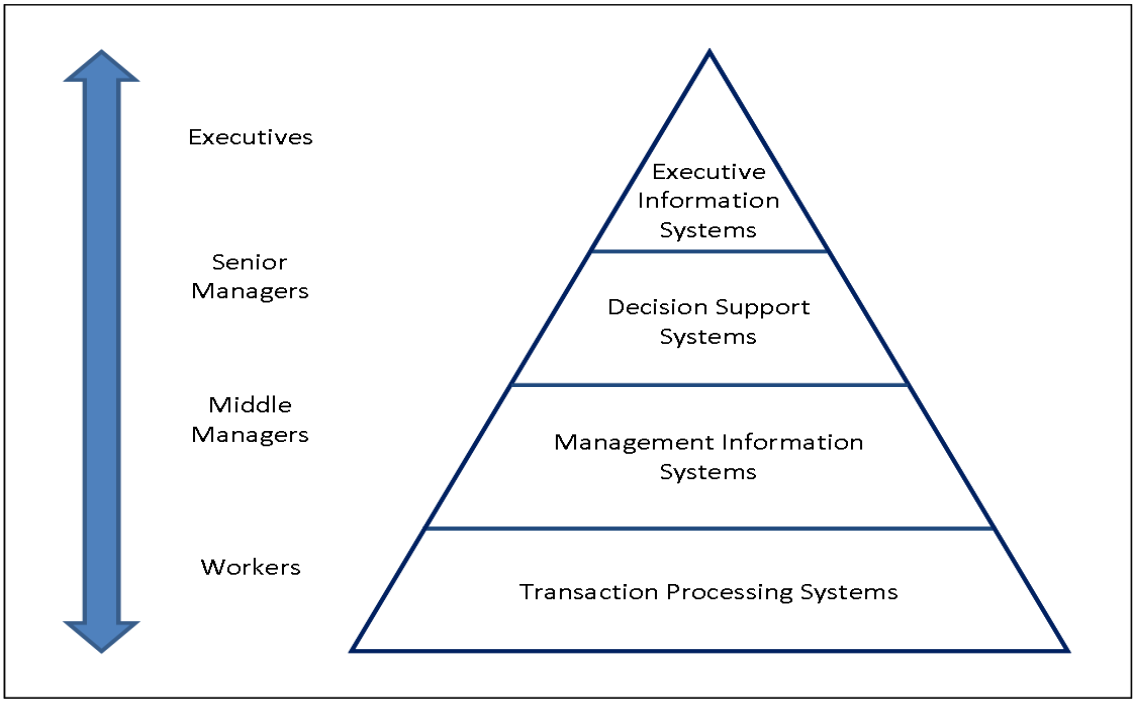

Figure 2.1: Hierarchy of Information Systems, (Laundon & Laundon, 1988).

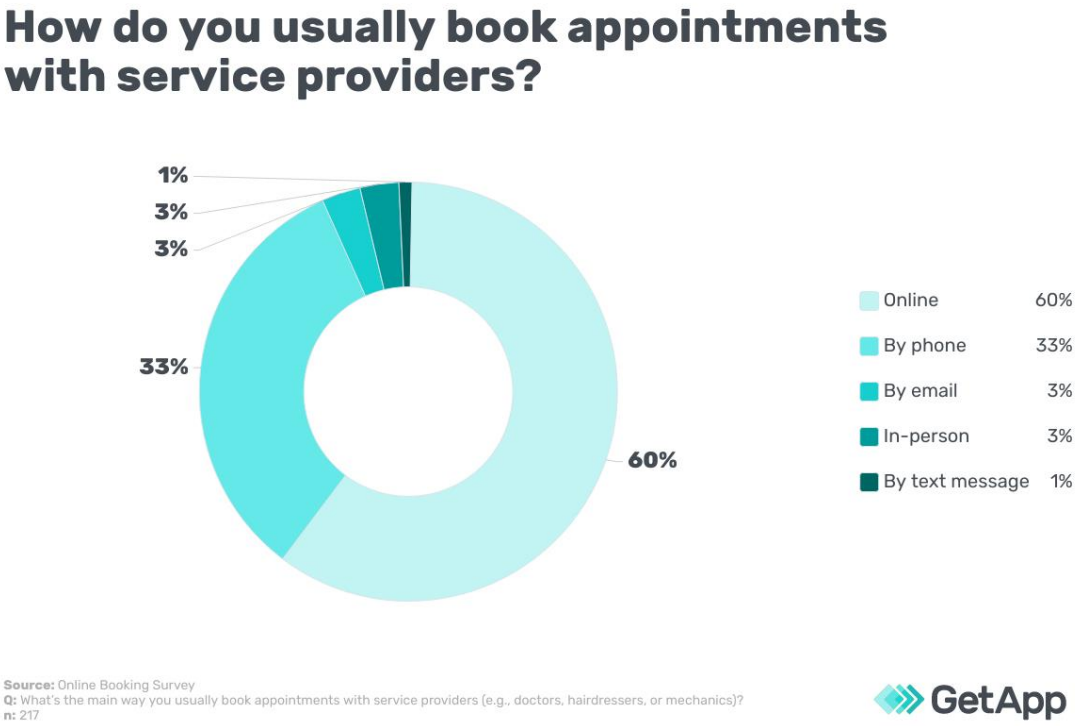

Figure 2.2 Survey of Appointment Booking Systems, (Hedges, 2021).

section of an organization, (The art of service., 2019). Transactions occur daily in most organizations and there's a need to properly handle them to ensure the successful running of organizations. The nature of an organization's business determines what kinds of transactions are being run daily. In a transportation organization, for example, transactions may include placing bus ticket orders, receipts of successful orders, etc. while in financial institutions transactions could involve cash withdrawal or deposit. Some transactions like employing new staff or making orders are common among organizations. A good transaction processing system would support organizations in managing transactions, (Krukeja, 2021)

b. Management Information Systems – They are particularly designed to be used by managers of organizations. It assists in the control and/or planning responsibility of managers. It makes use of information that has been gathered by the transaction processing systems and processes it to generate reports or product details that are in turn used by the managers. Typical examples of data generated from management information systems include work summary, or quarterly price of sales made by the sales department of an organization, (Krukeja, 2021).

c. Decision Support Systems – Decision support systems are powerful information systems often used by senior managers of organizations to make decisions. This information system leverage data mining and other powerful analytics tools to provide information that educates the decision-making process in organizations. (Zwass, 2020) describes two types of decision support systems. A model-driven system, where a data set, i.e., sales for the last quarter, is relatively used on an existing model, and a data-driven system where data mining techniques are

applied to data generated over some time, stored in data warehouses to obtain useful information.

d. Knowledge Work Information Systems – These systems are designed to ensure that knowledge and technical skills are inculcated into an organization, (Krukeja, 2021). They provide methods to gather knowledge and ensure that knowledge gathered is properly used. This knowledge could be images or documents found in patents, design methodology, competitive products, etc. Applications like Microsoft SharePoint are used to implement this kind of information system, (Zwass, 2020). (Krukeja, 2021), also identified some examples of this kind of information systems, some of which includes computer-aided design systems, for creation or reimplementation of designs using computer graphics applications, virtual reality systems, which make use of computer graphics software to produce simulations that are very realistic and financial workstations, which help to

e. Executive Support Systems – It is an extension of the decision support system, usually used by top-level executives of firms to educate their decision-making process by providing an overview of the organizations' performance, (The art of service., 2019). Judgments made with the assistance of an executive support system are non-routine decisions that have far-reaching implications for the entire business and thus require very critical analysis. Consequently, they usually have better compute, networking capabilities, and stronger display options, to present useful information in the forms of plots or charts that help top executives make critical decisions, compared to other information systems. Executive support systems help answer questions like what business an organization should start, or how to beat competitors in the market.

2.2 Health Information Systems.

A health information system is a system that is designed to receive, store, analyze and output data gotten in a healthcare facility either public or private. They also include all the systems that handle health data used by health providers or organizations. Examples of health information systems include electronic medical records, patient management software, patient portals, remote patient monitoring, etc. (Brook, 2020). Health information systems help to better the quality of health care, it also reduces the cost of operations of health care facilities and generally improves and organizes the administration process, (Sinhasane, 2019).

2.2.1 Benefits of health information systems

Sinhasane (2019), outlined some benefits of health information systems which include:

a. Improved Patient Care – These systems provide health workers with user-friendly software that boosts communication between them and patients and efficiently deliver health services to them by collecting and analyzing patient information like diagnosis report, allergies, appointment history, etc.

b. Co-ordinated Treatment Process – With the Healthcare Information System, it is simple to transfer the secured data of patients. The majority of patients with serious illnesses require care from a team of specialists. Quality care can be ensured by quickly transmitting records and reports across multiple departments in any healthcare facility. It is possible to implement a healthcare information system in primary secondary or tertiary institutions.

c. Reduction in costs – Costs in healthcare are reduced when patient data is stored digitally since the need to print documents would be greatly reduced and hospitals can be rid of the old filing system.

2.2.2 Types of health information systems.

There are different kinds of health information systems being used by different health institutions around the world. Some of the types of health information systems as highlighted by (Shrushti, 2020) are discussed in the following parts of this section.

a. Medical management systems – These systems handle the administrative aspects of the health care facility. Tasks like scheduling appointments, clerical works, storing and retrieving documents are usually taken care of by this type of health information system. It handles online payments and billings but usually doesn't contain the medical data of patients. Nowadays, medical management software's being used in healthcare facilities irrespective of their size be it a single practitioner or a big hospital.

b. Electronic health records - Another sort of hospital management information system is electronic health records, and it is primarily concerned with the medical information of the patients. Before its introduction doctors had to manually input the medical details of all their patients. Nowadays, they simply input it into the electronic health records system. It allows for seamless sharing of data between doctors of different specialties or wards in a system and access to data can be done from a mobile phone or a computer, and this gives room for faster treatments. It is concerned with the management of patient data, which includes information on a patient's medical history, allergies, and other factors.

c. Electronic Prescribing Software – These systems are used by healthcare facilities to generate patient prescriptions electronically. It enables them to transmit prescriptions automatically to pharmacies in a matter of clicks, and when the patient arrives, the store is already stocked with their supplements. The design

reduces any need for handwritten notes, which are often difficult to comprehend, and instead generates a prescription that is easy to grasp.

d. Remote Patient Monitoring – RPM systems are gaining recognition at an alarming rate and it's for the right reasons. They make use of medical sensors to monitor a patient and send patient data from anywhere they are to databases at healthcare facilities. These systems allow doctors to foresee and take care of events that might have caused unfortunate situations. The RPM lessens the cost of readmission to the hospital. It is resulting in higher-quality medical care. In the long run, the information gathered can be used to examine the health of larger populations.

e. Patient Portals – These systems provide avenues for patients to review their medical records at any given time, through any device. It usually contains their medical data such as appointment history, medications, treatments, etc. They also allow patients to book doctor appointments, view and/or make payments online, using their devices. In addition, some patient portals allow patients to communicate with healthcare experts. Rather than standing in line for hours to make an appointment, patients may now simply log in, check their doctor's availability, and review their lab reports.

2.3 Appointment Booking Systems.

An appointment is an agreement between two people to meet at a certain time at a specific place at a particular time. There are different approaches to appointment booking employed by organizations nowadays. A manual approach, that usually involves customers or patients, requesting appointments via mail or a phone call, and a representative of the organization replying with an available date. This bickering usually

goes on for a long time as there might be conflicts in the available time vacancies of the customer and/or the organization. This usually has a negative effect on the organization's performance and company resources end up being wasted. (Appointment-plus, 2015). The other approach leverages information technology and automates the entire appointment booking process. Customers or patients usually just book appointments online from the comfort of their homes via an appointment booking system. This approach is gaining recognition and it is being inculcated into many large and small businesses. Online booking systems eliminate all the issues associated with manual booking and reduce the costs of providing services in organizations as well as optimizes the overall performance of its employees. It is also very cheap to acquire and is acceptable as these systems are usually easily accessible through a mobile device. (Laptop's or mobile phones), (Appointment-plus, 2015). Online appointment systems can be used by a variety of organizations. Healthcare facilities, saloons, spas, restaurants, or any organization that provides services to people. Additionally, (Hedges, 2021) states that according to a survey of about 200 people carried out in North America, about 70 percent of the participants indicated that the manual approach was frustrating and preferred the online appointment booking approach, as graphically illustrated in figure 2.2 above.

The online approach to appointment booking also has some shortcomings. Lateness or no-show of customers is one of its biggest problems. It is not possible to force users to show up or come early when they book appointments and as such, time and resources might end up being wasted. Organizations using appointment booking systems should also be ready for an influx of customers at times and unavailability of customers at other times, (Team-fourUp., 2018).

2.3.1 Appointment Booking in Healthcare Facilities.

Healthcare facilities provide health services to people. Around the world, healthcare facilities usually experience an influx of patients daily. (Michas, 2020), illustrated research that proved that about 612,000 people visit the hospital on daily basis. Access to healthcare providers in countries like Nigeria is very poor. (Onyedifenu, 2020), estimates the ratio of available doctors to patients in Nigeria to be 1 doctor to 2753 patients. Consequently, patients around the world experience issues of long wait and service time. Many healthcare facilities now implement appointment booking systems to automate appointment booking and regulate access to health practitioners. These systems help to curtail prevalent issues of long wait and service times often experienced by patients in hospitals. They also help to improve customer satisfaction, patronage, and control the wastage of hospital resources.

2.3.2 Benefits of Online Appointment Booking Systems to Health Facilities

There are a lot of advantages that come along with implementing appointment booking systems to automate appointment booking in health care facilities. Some benefits of health information as indicated by (Reddy, 2020), are discussed in the following part of this section.

a. Ensures proper allocation of resources – The conventional appointment scheduling method demands a significant amount of time and resources. The receptionist must check the availability of doctors, manually fill out all patient data, and remind people of their appointments. This is a time-consuming procedure that may or may not be perfect. Furthermore, human error may be readily reduced by using an online appointment scheduling system. This system automates the entire process, saving time and reducing the healthcare workers' workload.

b. Boosts patient satisfaction – In the conventional appointment booking technique, hospital or clinic staff must manually schedule an appointment and send reminders indefinitely. When a clinic or hospital adopts an appointment scheduling program, staff can pay more attention to their patient's requirements. Staff can focus on providing the patient with the care they need upon arrival at the healthcare center, giving them time to do something other than making an appointment by hand.

c. Better ability to process and store patient data – Before the introduction of health information systems, patient information was recorded in a book or manually entered into a system. All patient information can now be maintained, updated, and managed correctly through the use of an online appointment scheduling system. This health record allows the hospital or healthcare practitioner to access a patient's medical data and deliver the best possible care.

d. Helps to save time – A doctor's visit can be made easier by arranging it online. It eliminates the need for patients to remain in a waiting room for extended periods. Patients save time with less messy and more scheduled visits to the doctor. As a result, patient satisfaction is increased and long-term loyalty is guaranteed.

2.4 Software Development Life Cycle (SDLC)

The software development process consists of a series of activities that takes place during the development of software. The period between a software's inception when its implemented for use and tested is its life-cycle. This time frame can be infinite as software usually have capabilities to reflect a change in its requirements. The software development life-cycle is a process used by software development companies to develop software in a way that guarantees quality, (Techopedia, 2011). It comprises a well-

described plan detailing how to create, maintain and evolve the software to be developed. The SDLC describes the stages involved in creating and/or improving the quality of software, (Tutorialspoint, 2021)**.** A graphical representation of the phases of software development is illustrated in figure 2.3.

2.4.1 Software Development Lifecycle Phases.

The software development lifecycle is divided into four distinct phases, (Sommerville, 2011). These phases are discussed in detail in the following parts of this section.

a. Software Specification – During this stage user and system the requirements of the system are identified and the constraints on the software's operation and development (i.e., development cost) are defined. It's a critical stage as errors in this stage are expensive and would go on to cause troubles during the design and implementation of the system. The four major activities that take place during this stage are;

   i. Feasibility study – A feasibility study is usually undertaken at the beginning of the project to determine if developing software from scratch is necessary or its cost constraint i.e. if capital exceeds profit.

   ii. Requirements analysis – The requirements of the system are elicited through analysis of legacy systems, or through discussions with target users or organizations the software is being developed. Prototypes can be used to model the system behavior and educate the requirement specification process.

   iii. Requirement specification – It is the process of transforming the requirement into formal documentation. Its content includes user requirements, usually an abstract representation used and understood by

the end-users of the system, and the system requirements containing implementation instructions to be used by the programmer.

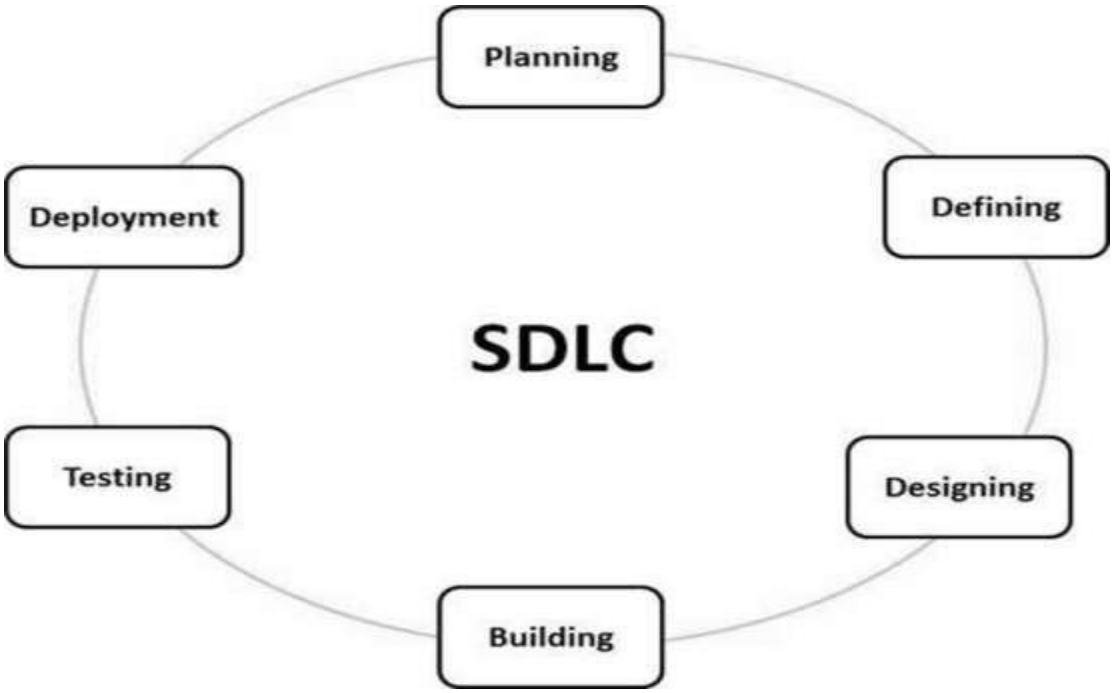

Figure 2.3 Software Development Lifecycle, (Tutorialspoint, 2021).

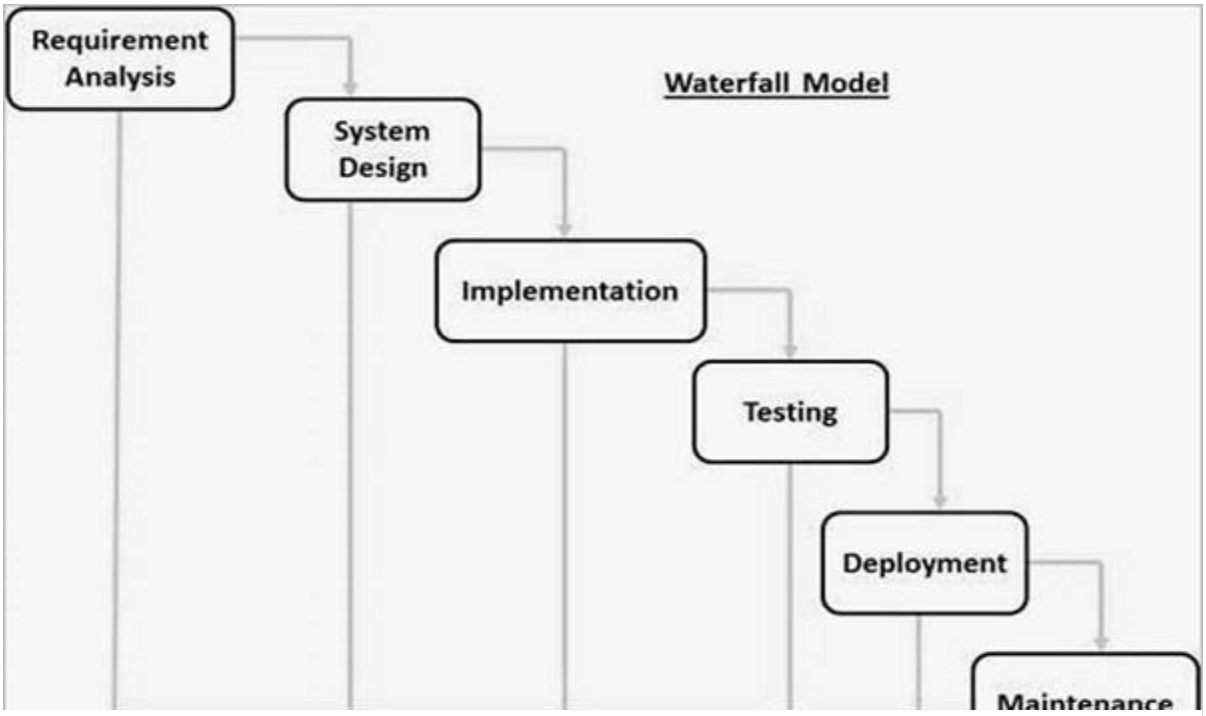

Figure 2.4 The Waterfall Model, (Sommerville, 2009).

iv. Requirement validation – This stage checks the requirements documents to identify errors and fix them. It ensures that the requirements specified are feasible and complete.

b. Design and Implementation – During this stage, the system specification is usually converted into a working system. Firstly, the structural description of the system to be designed, the system design, is defined. Then the design is implemented usually in a programming language. The major activities in the software design phase include;

   i. Architectural design – The overall structure of the system, its subcomponents, and the relationship between them are determined in this process.

   ii. Interface design – The interface, i.e., the communication media, between the interface is designed in this process. A good interface design allows components to easily communicate with each other.

   iii. Component design – The functionality of the individual components in the system is implemented by the software programmer. Alternatively, it could be a list of modifications to be made to a re-usable component to be used by the system.

   iv. Database design – Depending on whether a new database will be created or whether an existing database, an API would be used, the systems data structure and how they will be stored on the database are described.

c. Software Validation – Software validation (testing) is done to ensure that the system conforms with the requirement specification and system design. The stages of software validation are:

i. Development testing – It is usually done by the developers of the system. Each component of the system is tested in isolation without other sub-components. Test automation tools, i.e. JUnit, are usually used for component testing.

ii. System testing – All the system component is integrated and the system is tested as a whole. This test ensures that no unanticipated errors arise from the interaction of system components with each other.

iii. Acceptance testing – This test is usually carried out together with the end-users of the system. User data as opposed to simulated is used to test the system. This system also helps to identify omissions in the software requirements documents.

d. System Evolution – In today's world, user and business requirements are constantly changing and software systems need to have abilities to cope with these changes. Software evolution is the process of changing software to reflect a change in its requirements and is usually a continual process.

2.4.2 Software development lifecycle models.

An SDLC model is an abstract framework that defines all the processes that would be involved in the development of a software project from its initiation to implementation. Each software model follows a unique pattern peculiar to itself. ISO/IEC 12207 is the international standard that defines all the tasks required for developing and maintaining software (Tutorialspoint, 2021). There are several models but they are generally classified into two, plan-driven, agile, or a combination of both, (Sommerville, 2011).

Plan-driven models are most suitable for large and safety-critical systems that have a long development lifespan, large development teams working from different teams, and systems with little need for change since these models have little tolerance for change,

and changing large systems can be very expensive. Agile models on the other hand are used for small or medium-sized systems with short development lifespans. They are most suitable for business systems with constantly changing environments and/or requirements as they usually put measures in place to tolerate change. These models include little or no documentation and usually focus on developing a working system quickly, (Sommerville, 2011).

a. Waterfall model

It's a plan-driven model. It is one of the oldest SDLC models and it is most for large engineering projects. When using this model to develop software you plan every process phase in advance and complete the phase before moving on to the next phase, and each phase is properly documented, (Sommerville, 2011). An illustration of the model is represented in figure 2.4.

(Sommerville, 2011) highlights the phases of the waterfall model and they are discussed in the following parts of this section.

i. Requirement elicitation and analysis – The systems requirements are gathered by discussions with the users of the system.
ii. System design – The system's hardware and software requirements and the system's overall architecture are usually defined at this phase of the life cycle.
iii. Implementation and Unit testing – The systems design is usually implemented in the program or program unit. Unit testing involves testing each program unit to ensure it meets requirements.
iv. Integration and System testing – All the program units are integrated into a complete system and the system is tested to ensure it meets the requirements.
v. Operation and Maintenance – It is the longest phase in the life cycle. It involves installing the system for practical use and modifying it to reflect changes.

b. Iterative model

It is an agile model also referred to as an incremental model. In this model, an initial version of the system is created, and users experiment with the system to properly define its requirements until a final system is built i.e., the system is built in increments, (Tutorialspoint, 2021). This system is suitable for systems to be used in business environments since the costs of accommodating change are low. In this model, software requirement elicitation and specification, design and development are intertwined as opposed to being stand-alone. This model is not suitable for large or safety-critical projects however because as new increments are added, the overall structure of the system tends to degrade and this model doesn't prioritize staged development of systems or documentation of progress. An illustration of the iterative model is represented in figure 2.5.

c. Re-use oriented software engineering.

In most software systems, there is some software reuse. This typically happens informally when people working on the project know about designs or code that are comparable to what is required. They seek them out, change them as necessary, and integrate them into their system. Reusable software components and an integrative framework for their composition are the foundations of reuse-oriented techniques, (Sommerville, 2011). The stages of this model are discussed in detail in the following parts of this section.

i. Component analysis – A search for components to implement the requirements specification is conducted based on the requirements specification. In most cases, there is no exact match, and the components that can be employed only give a portion of the desired functionality.

ii. Requirements modification – Using the knowledge about the components that have been discovered, the requirements are examined at this step. They're then

tweaked to reflect the available components. If changes aren't possible, the component analysis activity can be re-entered to look for other options.

iii. System design with re-use – The system's framework is designed or an existing framework is reused during this phase. The designers take into account reusable components and organize the framework to accommodate them. If reusable components aren't available, new software may have to be created.

iv. Development and Integration – The components and COTS systems are merged to build the new system, and software that cannot be obtained outside is produced. In this concept, system integration might be a part of the development process rather than a distinct operation.

d. Extreme programming

Extreme programming is probably the most well-known agile method. It was introduced by pushing already available agile methods to extreme levels. It is a new software process model where user requirements are usually expressed using scenarios and implemented as a series of tasks. A group of programmers usually work together to develop and test each task and new code must pass test cases before they are integrated into the system. In extreme programming, it is possible to develop multiple versions of a system in a day, (Sommerville, 2011). Some characteristics of extreme programming as indicated by (Sommerville, 2011) include;

i. The system is developed incrementally and released in increments. User scenarios are the basis for deciding functionalities to be implemented next.

ii. The customer representative is usually a part of the development team and this representative defines the acceptance tests for the system.

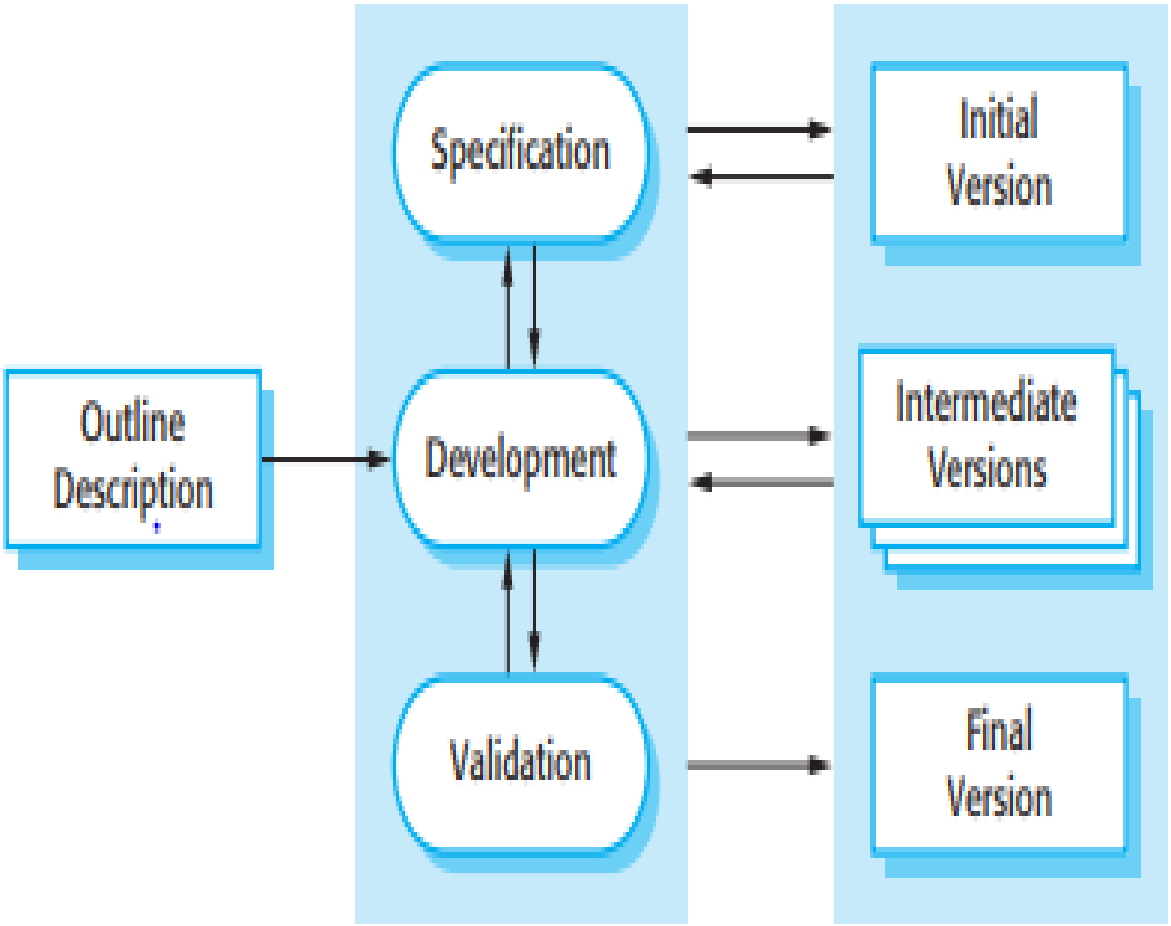

Figure 2.5 The Iterative Model, (Sommerville, 2009).

iii. Pair programming, joint ownership of the system code, and a sustainable development approach that does not require long working hours promote the involvement of people rather than processes.

iv. It creates an avenue for change since it releases the system incrementally to the users, tests first before development, constantly refactors code to eliminate deprecation, and continuously adds new functionality.

v. Maintaining simplicity is facilitated by frequent refactoring that improves code quality and by utilizing basic designs that do not unduly anticipate future modifications to the system.

Extreme programming, however, has some disadvantages. One of them is that constantly changing the system results in a degradation of the system's architecture which causes implementation and design issues. Since programmers code in pairs, the difference in the skill level of programmers usually causes compatibility issues and reduces the effectiveness of refactoring, (Sommerville, 2011).

2.5 Unified Modelling Language (UML)

UML is a set of diagrams that can be used to visualize a software program. Grady Booch, James Rumbaugh, Ivar Jacobson, and the Rational Software Corporation developed the notation to be used for object-oriented design, but it has subsequently been expanded to cover a broader range of software engineering projects. The Object Management Group (OMG) now recognizes UML as the standard for modeling software development and it has been responsible for its management since 1997, (SmartDraw, 2021).

In 2005, the ISO also approved the UML as an approved ISO standard, and it has been occasionally revised up till the most current version. Unified modeling is a pictorial

language used for making blueprints for software and non-software systems. There are different types of UML modeling, (Tutorialspoint, 2021). They include;

a. Structural modeling – It represents the structural features of the system. It usually doesn't describe the behavior of the system, just its component and methods to assemble them. Examples are class and component diagrams.
b. Behavioral Modelling – It describes the interaction between components in the systems. It describes the dynamic nature of the system. Examples include activity diagrams, use-case diagrams, and interaction diagrams.
c. Architectural Modelling – It represents the framework of the system and it consists of the structural and behavioral elements of the system. An example is the package diagram.

2.5.1 UML models

i. Use-case diagrams – They illustrate the functionality of the system or a part of the system and its interaction with external agents. It also describes all the instances where a system can be used, (GeeksforGeeks, 2019). (Tutorialspoint, 2021), highlights several purposes of the use-case diagrams;
    a. Used to elicit requirements for a system.
    b. Identify factors (external and internal), influencing the system.
    c. To understand the external structure of the system.
ii. Activity diagrams – They illustrate all the activities that are carried out in a system and the control flow from one activity to another, (Sommerville, 2011). (Tutorialspoint, 2021) highlights several purposes of the activity diagram;
    a. Illustrate the control flow of activities in the system.
    b. Describe the sequence of activity in a system.
    c. Describes the parallel, concurrent and branched flow of the system.

iii. Class diagrams – They are used when creating object-oriented systems to show their classes and the relationships between these classes, (Sommerville, 2011). They are static diagrams, used for describing different parts of a system and creating code for a software application, (Tutorialspoint, 2021). (Tutorialspoint, 2021) highlights several purposes of class diagrams;

   a. Analysis and design of the static view of an application.
   b. Describe the function of systems and subsystems.
   c. Identify classes of the system.
   d. Forward and reverse engineering.

iv. Sequence diagrams – They illustrate the relationships between actors and objects, objects and objects in a system. They show the series of interaction that exists during a particular activity, (Sommerville, 2011). Some of the purposes of sequence diagrams are highlighted below

   a. Understand requirements for new and existing systems, (GeeksforGeeks, 2019).
   b. Describe the functionality, and the order of functionalities of objects in a system, (GeeksforGeeks, 2019).

2.6 Software Development Tools.

Software development tools are used by programmers or software developers to create, debug and deploy application programs. They are usually tools used by developers to create other software, (Goodfirms, 2019). Examples of software tools include software frameworks, compilers, interpreters, and integrated development environments (IDE's). These tools generally help programmers communicate with machines and get information from programmers, i.e., an automated software testing tool would help a programmer test

the functionality of a program, and so on. Software development tools could be software development frameworks for developing application software, or IDE's, automated testing tools, compilers, environments for running programs, etc. Some software development tools are discussed in detail in the following parts of this section.

i. Software Frameworks - A software framework is software providing generic functionality that can be altered using user-specific code to add additional functionalities. Popular examples of software frameworks include React-native, Flutter, Xamarin, JIT compilers, etc.

ii. Integrated development environment (IDE) – They are software applications that provide facilities for developers to create software applications. Popular examples include VS Code, Sublime text, Note++, etc.

iii. React Native – It is an open-source cross-platform development tool that allows software developers to develop iOS, Android OS, Windows OS, Mac OS applications from a single code base and provides native features for the devices that software is being built to run on. It was developed by Facebook in 2015, (O'Reily, 2015).

iv. Expo - Expo is a framework and a platform used to build React applications. It is a set of equipment and offerings constructed around React Native and native systems that assist you to develop, build, deploy, and shortly iterate on iOS, Android, and net apps from the identical JavaScript/TypeScript codebase, (cromacampus, 2021).

v. Node – Node.js is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a web browser. Node.js is a popular, lightweight web framework for beginners, and it is used by many big companies like Netflix and Uber.

2.7 Review of Related Works.

This section provides a review of similar literary works produced by other researchers in the field of health information systems.

(Idowu & Adeosun, 2014), at the Obafemi Awolowo University worked on developing a dependable online appointment booking system for NHIS outpatient in Nigerian teaching hospitals with a MySQL database, a WAPM server, PHP, and dream weaver. Its software used to book appointments is web-based and made use of SMS or mail messages to remind the users of their appointments or to notify them of changes made to their appointments. The system also provided a means for patients to monitor their health records.

(Maryam, 2018), at the Tampere University of Applied Sciences developed an online polyclinic appointment and database management system in Pakistan, with HTML, CSS, and JS on the client-side and PHP on the server-side. The developed software allows users to book appointments on a website. The software also provides a medium for patients and doctors to communicate to reduce the need for patients to visit the clinic. It also links to the pharmacy so the pharmacist can see prescriptions and dispense drugs.

(Akinode J.L, 2017), at the Federal Polytechnic Ilaro, Nigeria, worked on the design and implementation of a patient appointment and scheduling system using Angular Js for frontend, Ajax framework for handling the client-server requests, and MySQL for the backend. He developed a webpage from which patients could book appointments and made implemented a calendar using ajax to automate the appointment scheduling process.

(Garko & Mahmud, 2017), at the Department of Computer Science Federal University, Dutse, Jigawa State. Nigeria, worked on the design and implementation of an outpatient management system, with the waterfall software process model, and the Visual Basic

.Net and Microsoft SQL software tools. They successfully developed a web-based appointment booking system and made use of SMS/email to notify users of changes made to their appointments or remind them of their appointments. The system helps patients monitor their appointment records and hides sensitive information like X-rays, lab results, etc.

(Xie & Or, 2017) conducted a study on the association between waiting times, service times, and patient satisfaction in an endocrinology outpatient department. They recruited endocrinology patients from a teaching hospital in China, 18 years and older, and surveyed them. They concluded that although reducing wait times may not always be possible (due to lack of resources or staffing limitations), it remains possible to improve some patient-centered aspects of care services, which can mitigate the patients' dissatisfaction with waiting periods and enable more positive perceptions regarding the services they receive

CHAPTER THREE

SYSTEM DESIGN

3.0 Introduction.

This section provides an overview of the software development process. The extreme programming software process model was the SDLC model employed for the development of this project. Multiple versions of the system were created and exposed to experimentation by the target users of the device and other principal stakeholders until a stable version was created.

3.1 Method of Identification of User and System Requirements.

This section gives an overview of the system and user requirements of the system, and how these requirements were gathered. Some of the requirements for the system to be built were gathered through informal discussions with the users of the system, i.e., patients (students and staff of MTU) and healthcare workers (doctors and nurses at the MTU health center). Other requirements were gathered by a deep review of related literature, online articles, books, and an analysis of existing systems to properly understand the expected functionality of the system. After developing an initial version of the system, the software was made available to target users and other important stakeholders of the system to experiment with. This activity helped to refine the requirements of the system and elicit new requirements that had not previously been identified. This exposure also ensured that the users of the systems were well familiarized with the systems interface to ensure ease of use.

3.1.2 User requirements

This section gives an overview of the elicited requirements of the system, in an abstract high-level language, for the users of the system. These requirements are functionalities

the system is supposed to make available to the users of the system. They are highlighted below;

i. It shall provide an avenue for the different users to book doctor appointments.
ii. It shall provide methods for patients to cancel created appointments
iii. It shall provide the means to register patients and healthcare facilitators with the system.
iv. It shall provide an avenue for the nurses to validate appointments that have been booked and assign approved appointments to doctors.
v. It shall a means for healthcare facilitators to report the diagnosis of and prescription given to patients after every medical session.
vi. It shall notify patients when their appointments have been canceled or when new appointments have been scheduled for them by the healthcare facilitators.
vii. It shall provide all the aforementioned requirements with a user interface that is easy to use and understand.

3.1.3 System requirements

This section provides a detailed description of the elicited requirements for the senior stakeholders and developers of the proposed system. The system requirement of any system is usually described in terms of the functional and non-functional requirements. The functional and non-functional requirements of the system are highlighted in the following parts of this section.

a. Functional requirements

They are the functionalities provided by the proposed software. They include;

i. The software shall provide a page through which patients would book doctor appointments.
ii. The software shall provide avenues for the patients to cancel appointments.

iii. The software shall provide a page for nurses to view booked appointments

iv. The software shall provide an avenue for nurses to validate appointments and assign patients to doctors

v. The software shall provide avenues for nurses to cancel appointments and update patients when their appointments get canceled.

vi. The system shall provide a screen from which doctors can view validated appointments.

vii. The system shall provide a means for the doctor to report patient medical data after appointments and schedule appointments if need be.

b. Non-functional requirements

They are the constraints in the services rendered by the system. They include;

i. The system shall authenticate users with their email and passwords before entering the system.

ii. The system shall authorize users before granting them access to certain data stored in the software.

iii. The system shall provide an interface that is easy to use and easy to understand to the users.

iv. The system shall render all of its services in a timely fashion

v. The system shall store all of the user credentials and appointment data in cloud storage with only authorized access.

c. Software and hardware requirements

The system software and hardware requirements were gathered from an informal study of the target users. The majority of the patients that would use the system are students and all students have access to a tablet. The majority of the staff and healthcare facilitators at the university make use of mobile devices. With this information in mind, the hardware

and software requirements of the proposed system were gathered. These requirements are highlighted in the following parts of this section.

i. Hardware Requirements – They are the standard hardware components required of any device that would use the system. They include;
    a. A mobile device (phone or tablet).
    b. Internet connectivity (minimum 3G).
    c. 1 gigabyte of R.A.M.
    d. 200MB disk space.
    e. Single-core processor
ii. Software Requirements – They are the standard software component required of any device that would use the software. They include'
    a. Android OS (version 4.4 or higher) or iOS (version 9 or higher).
    b. Android SDK 16 or higher.

3.2 System Design Models

Several UML models were used to specify the design of the system and structure its overall architecture. These models provided graphical representations of the interactions between several users of the system and the system, the hierarchy of activities in the system and the flow of control between these activities, and the overall architecture of the system. These graphical representations are illustrated in the following parts of this section

3.2.1 Use case diagrams

The use case diagram illustrates the relationship between several users and the system. There are four different types of users of the proposed software. They are the patients

(student and staff), nurses, doctors, and admin. Their roles are described in the following parts of this section.

i. Patient (student and staff) – The patients primarily make use of the software to book doctor appointments and cancel booked appointments if there's a need. The other activities they can perform include edit profiles view a log of current and/or past appointments, etc. The patient use case diagram is illustrated in figure 3.1 below.

ii. Nurse – The nurses view a list of every appointment booked for the day, validates them, and assign patients to doctors. The nurse is also in charge of registering patients with the system. The use case diagram is illustrated in figure 3.2 below.

iii. Doctor – The doctor views a list of validated appointments that have been appointed to him/her. The doctor schedules appointments if there's a need and reports the medical data of every doctor session i.e., the prescriptions and/or diagnosis. The use case diagram is illustrated in figure 3.3 below.

iv. Admin – The admin manages all the activities in the system. She/he creates updates and deletes data objects in the database. The admin also registers healthcare facilitators with the system. The use case diagram is illustrated in figure 3.4 below.

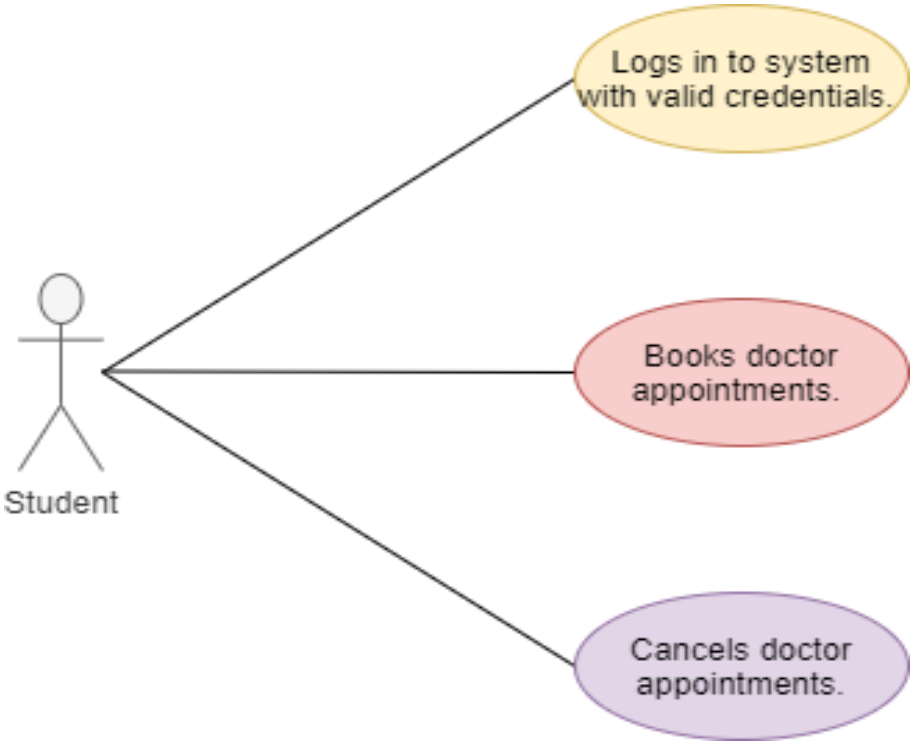

Figure 3.1 Use-case for Patients.

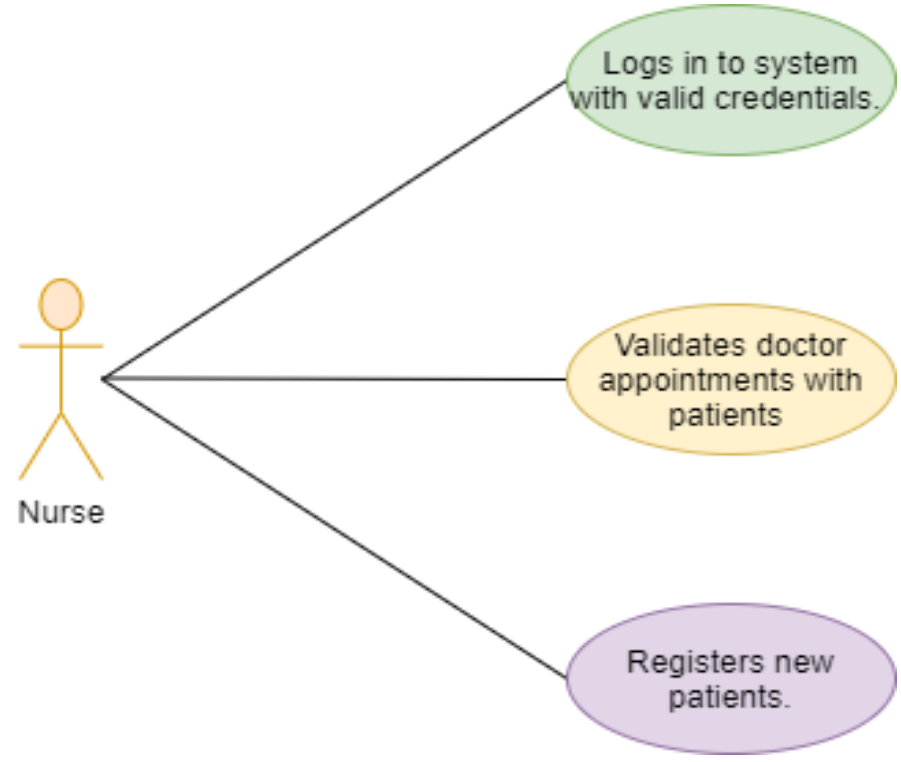

Figure 3.2 Use-case for Nurse.

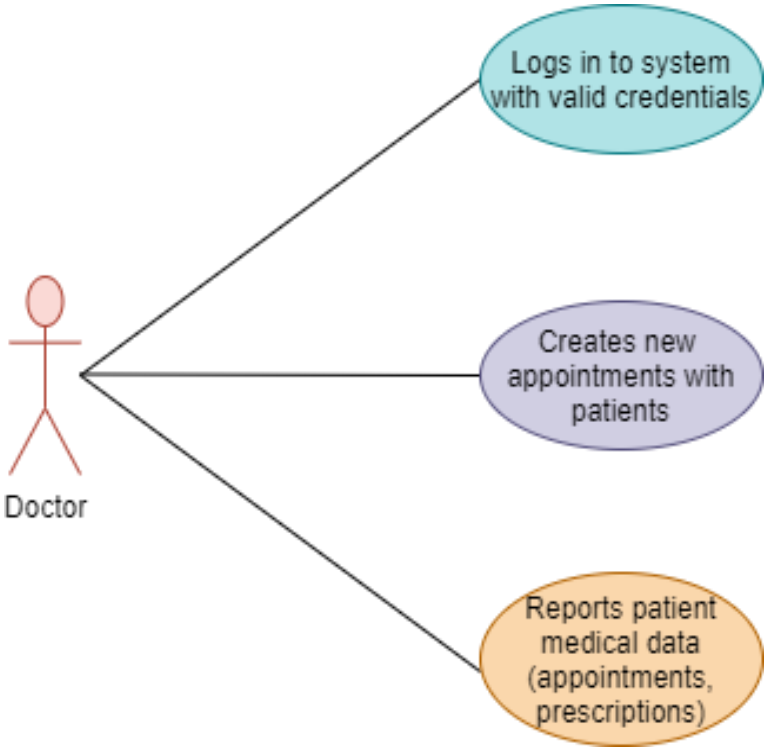

Figure 3.3 Use-case for Doctors.

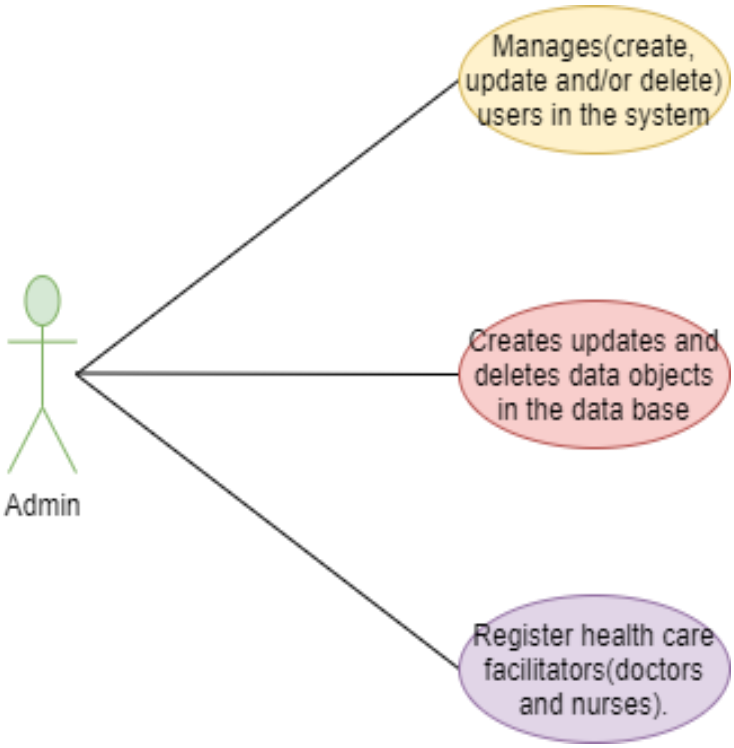

Figure 3.4 Use-case for Admin.

3.2.2 Sequence diagram

The sequence diagram illustrates the sequence of activities that usually happens in the system. The sequence of activities in the proposed system is illustrated in figure 3.5 below. The diagram graphically represents several sequences of activities that occur within the proposed system i.e., user's login into the system via an interface in the software, the application logic (responsive code) checks if the email and password conform with the standard email and password nomenclature, and the system checks if that user exists. If it does, the user is redirected to the navigation flow he's authorized to access, and if it is not, he is informed that the email and/or password doesn't exist and that he should try again. It also shows the sequence of activities that occur when the healthcare facilitators (doctors and nurses) use the system.

3.2.3 Activity diagram

The activity diagram illustrates all the activities that occur within the system and the flow of control between these activities. It shows the standard order of activities in the software system. The user logs in to the system and is directed towards a navigation route depending on what his or her role in the system is. The activity diagram for the proposed system is illustrated in figure 3.7 below.

If the user is a patient, he or she is directed to the patient navigator, where the user will gain access to a screen that would allow him to book appointments and another that would allow users to view and/or cancel appointments. Nurses get navigated to screens where they could register patients, view booked appointments, and cancel or validate them. Doctors get navigated to screens where they can view validated appointments, schedule appointments with patients, and report a log of the medical data retrieved from the doctor appointments.

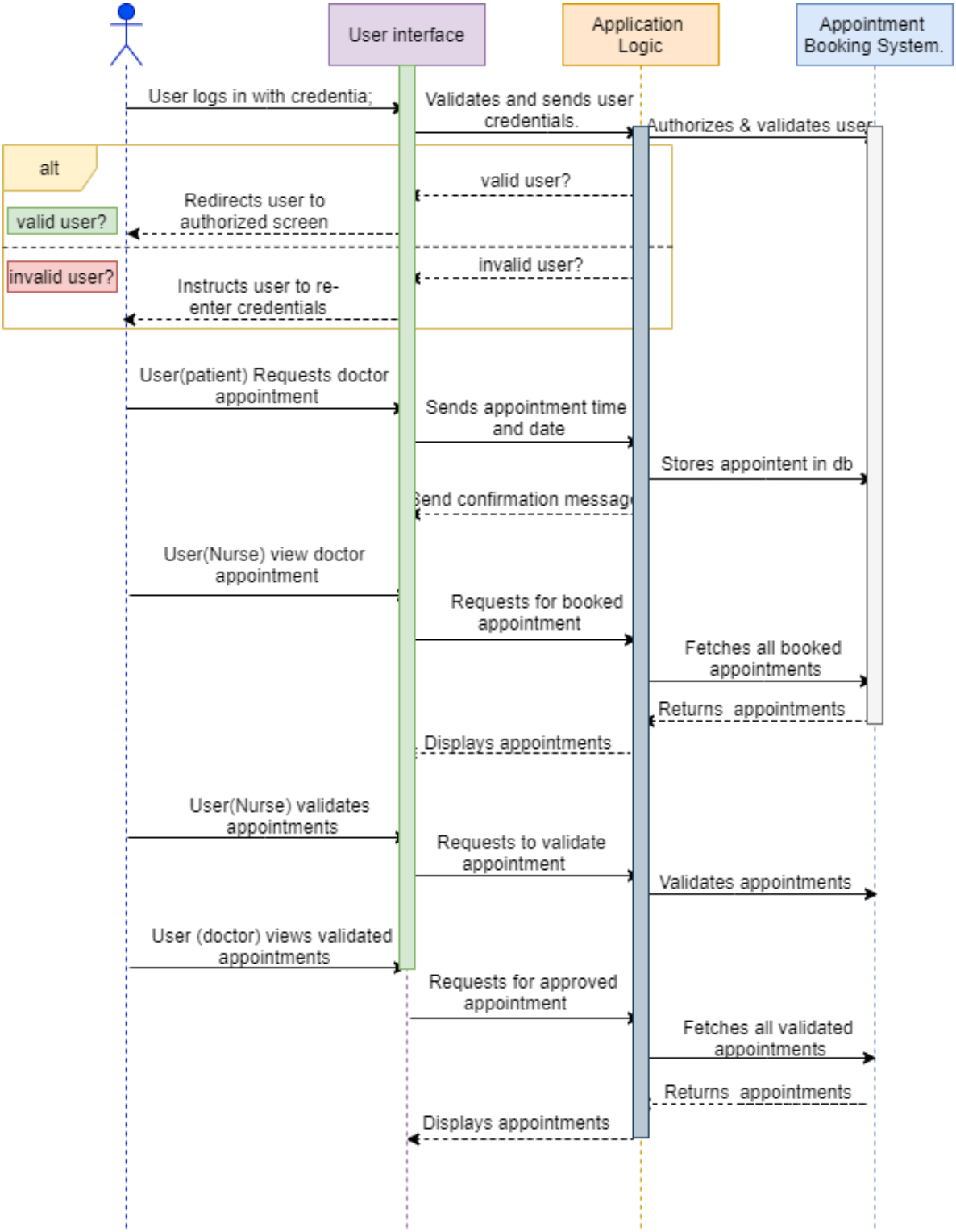

Figure 3.5 Sequence Diagram of Activities in the System

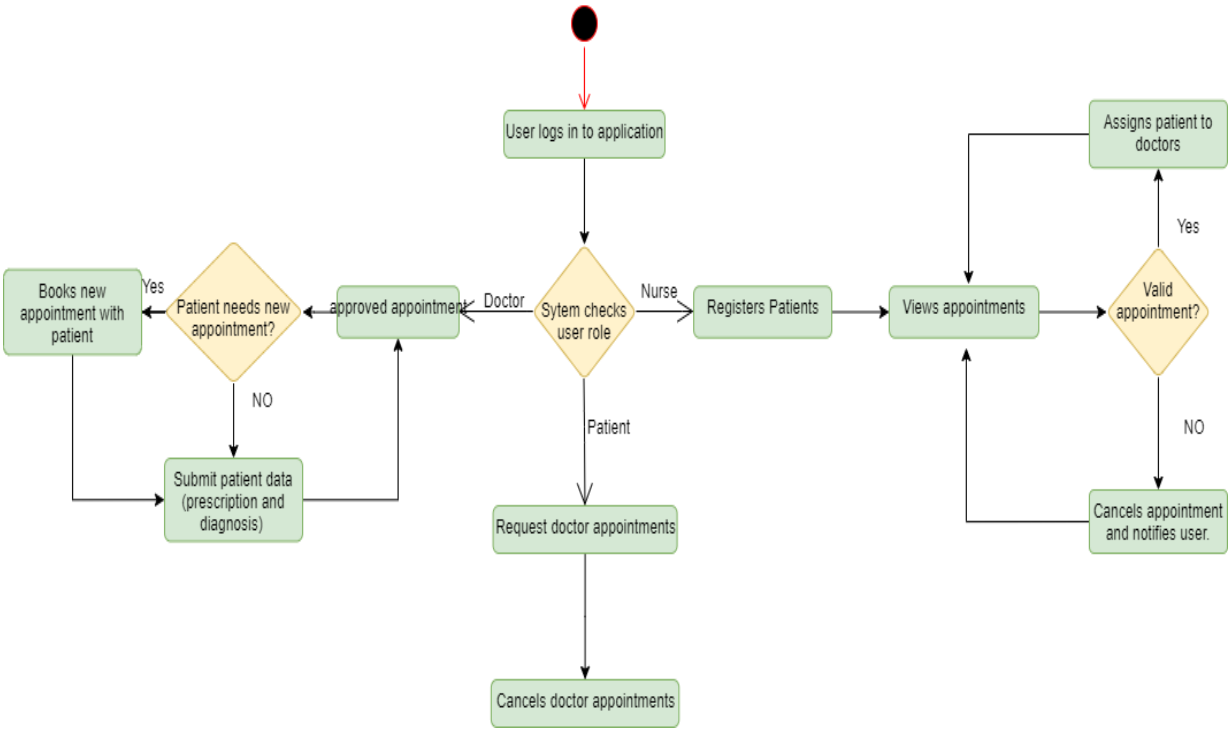

Figure 3.6 Activity Diagram

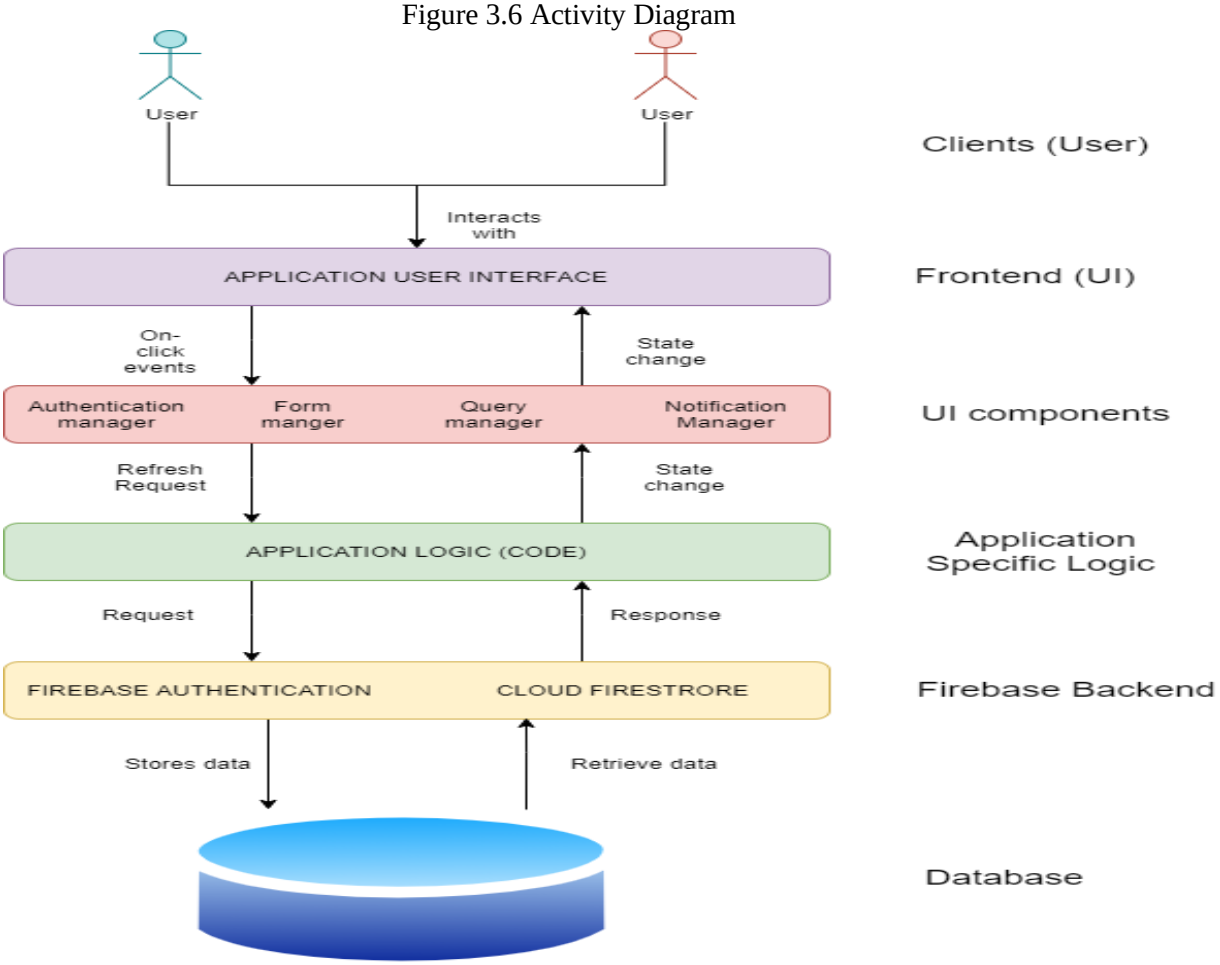

Figure 3.7 Architectural Diagram

3.2.4 Architectural Diagram

The architectural diagram illustrates the organization and architectural structure of all the components in the system. It also helps to understand how the components of the system communicate with each other. The proposed software consists majorly of six components. The users of the system, the user interface they communicate with, react-native UI components, application logic implemented in javascript, firebase backend, and a database. The architectural diagram of the proposed system is graphically illustrated in figure 3.8 above.

3.3 System Implementation

This section provides an overview of the methods used to actualize the system design specified in the previous section. The proposed software was implemented using a software framework, react-native, that allows for cross-platform development of mobile applications, to ensure that the software runs seamlessly on both Android OS and iOS devices. The front-end and the back-end implementation of the proposed software are discussed in detail in the following section.

3.3.1 Frontend implementation

The user interface for the software was designed with Adobe XD. This design was implemented on the react-native framework with javascript. The UI components of the software were built modularly to enable the re-use of code across the applications. A react-native library, @react-native-community/datetimepicker, was used to create a date-time picker for booking doctor appointments. Other react-native libraries, Formik and Yup, were used to validate the forms in the application. An animation library, Lottie view, was used in the software to provide useful animations that promote ease of use of the system and keeps the user aware of the state of the system at all times. React native

navigation, was used to configure the navigation for the different user types of the software. React-native libraries, Secure store, App-loading, and React-context were used to persist authenticated users and cache data to improve the performance of the software. Firebase tools are used to query data from the database.

3.3.2 Backend implementation

The backend for the software was implemented using firebase. The users are authenticated and authorized with firebase authentication. The software data is stored in cloud-firestore in JSON format (key-value pairs). The user data is collected when a new user is registered with the system and stored in could firestore in key-value pairs. The user data consist of the userId – string, name – string, matric number – number, email – string, password – alphanumeric, and display name – string. Each user's role (patient, doctor, or nurse) is also stored when a new user is registered with the application. Appointment data are also stored in the database as an array of objects to simplify the database query process. Appointments usually have properties of id – date, time – date, date – date, patient_name – string, patient_Id – string, and appointment_status – Boolean. The appointments are usually stored in two collections. One is accessible by patients and the other accessible to only the healthcare facilitators, as it helps to prevent data integrity. The patient medical data, consisting of bp – number, temperature – number, prescription – string, and diagnosis – string is also stored in the database after every doctor appointment.

3.4 Software Testing

A library used for testing javascript programs, Jest, was used to test the software. Several test cases were developed for testing the different subcomponents of the software in isolation from other components. The interfaces between these components were also

tested to ensure that communication issues don’t arise when components try to communicate with other components. The system as a whole was also tested to ensure that the system measures up to the system requirements. Beta testing was carried out, by the developers, some of the target users, and stakeholders of the system to ensure that no requirements were omitted when the requirements were being gathered and that the system is easy to use.

CHAPTER FOUR

IMPLEMENTATION AND ANALYSIS OF RESULTS

4.0 Introduction

This chapter provides an overview of the implementation of the system design specified in the design document. A detailed explanation of the database implementation, user interfaces, and application logic would be discussed in the following parts of this chapter.

4.1 Database Implementation

Google's firebase backend service was used to implement the backend for the software. The data generated from the software were stored as JSON (key-value pair) objects, in unique documents grouped as a collection.

4.1.1 Authentication and authorization

User registration with the system is handled by Google's firebase authentication services. The users register with a verified email and a password of their choice that is at least 6 characters long. After every successful registration, a user object with display_Name, email, password, auth_Token, and user_Role properties is created. The user_Role property is used to control access to data.

4.1.2 Database

Data generated in the app is stored in Google's cloud firestore, in a JSON, (no-SQL) database. The database's architecture requires you to store data or data sets as documents, and a group of identical documents is stored as a collection of documents. Four different collections of data were stored in our database.

i. User data – User information collected when new users are registered with the software is stored in the database. This information includes the user Full name – string, matric number – number, and e-mail – string, and the user_role (patient,

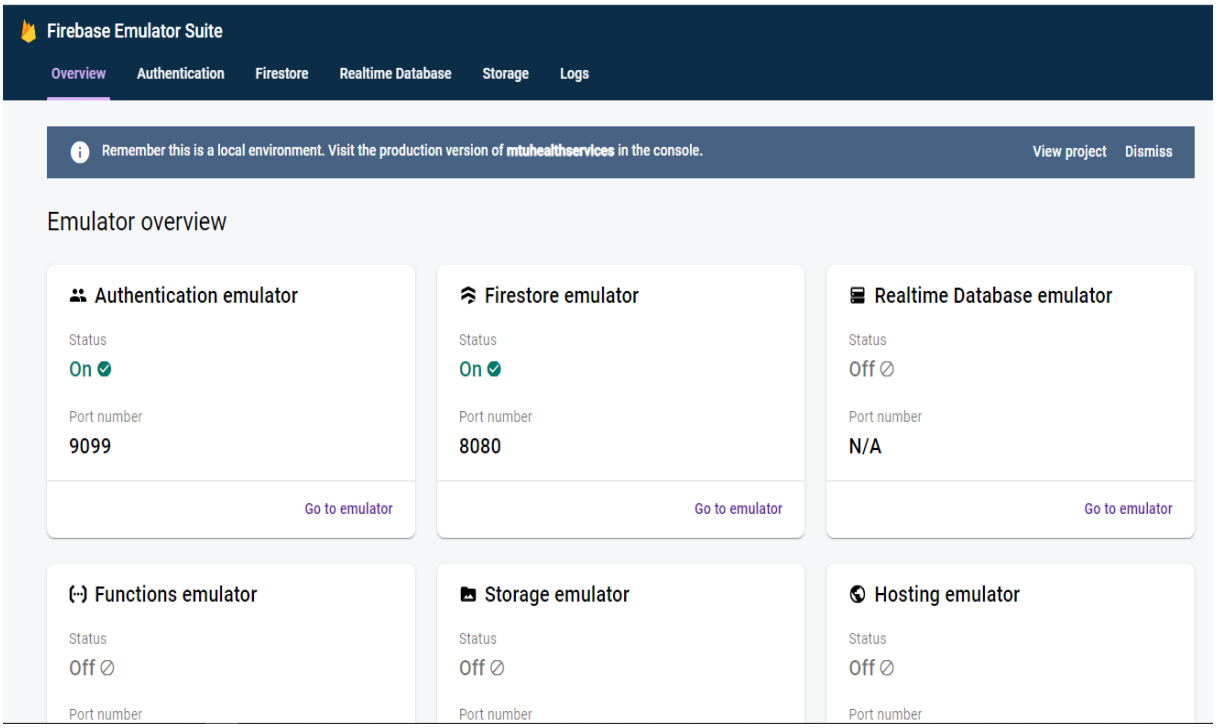

Figure 4.1 The Firebase Emulator Suite

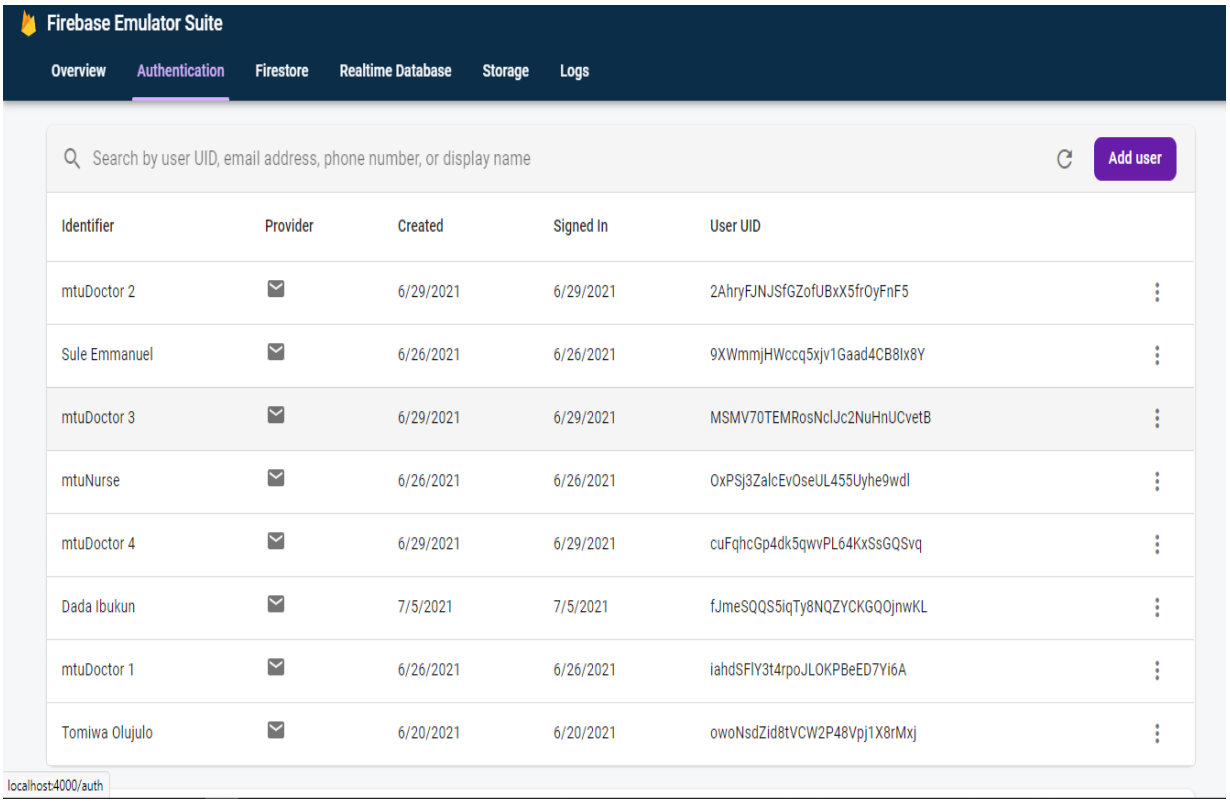

Figure 4.2 Users Registered with the system

doctor or nurse) - string. A unique id is also created to uniquely identify each user. Each user's data is stored in a document with its id identical to the user's id and all the user documents are stored in a user collection. The figure below shows how the user object is stored in the database.

ii. Appointment Log – The appointment log collection contains documents of all the appointments booked by the patients in the application and is illustrated in figure 4.4. Each document is an array of all the appointments that have been booked by a user and it's uniquely identified by the users' id. Each appointment has properties of appointment_Time – string, appointment_Date – string, appointment_Status – Boolean, and appointment_Id – Date used to uniquely identify each appointment.

iii. Current Appointments – It is a collection of every appointment that has been booked by all the users of the application and it is illustrated in figure 4.5. This data is used by the health care facilitators. The appointments are booked in a single document with an id "allAppointments". Each appointment has properties of appointment_Time – string, appointment_Date – string, appointment_Status – Boolean, patient_Name, patient_Id, and an appointment_Id – Date, used to uniquely identify each appointment.

iv. Medical-Log – This collection contains the patient medical data generated after every doctor session. It contains documents of medical data of each patient, uniquely identified by the user's id. The medical log is illustrated in figure 4.6

4.2 Frontend Implementation

The user interface for the software was designed with Adobe XD and was implemented in the react-native framework with javascript. Several screens were developed with React Native-UI components to provide a beautiful and easy to use interface for the users to perform their tasks. Some of the UI interfaces are explained in detail in the following part of this section.

4.2.1 Authentication and authorization

The users enter the system by logging in with their credentials at the login page as shown in the figure below. The admin is responsible for registering healthcare facilitators with the system and the nurses are responsible for registering users via the software's registration page as shown in Figures 4.7 and 4.8 below.

The form fields have validation schema's that have been implemented with Javascript and YUP that ensures that only the correct data type, email type, and/or password type gets inputted into the field depending on what it expects. If the credentials entered into the fields are invalid or if another user has been created with the same credentials, an error message is displayed. Depending on the user role of the user, different navigation schemas have been designed with javascript and some react native libraries so only authorized users have access to certain parts of the software.

4.2.2 Appointment booking

The appointment booking page can be used by patients to book doctor appointments. The appointment booking screen responds to on-tap events by the patients and prompts them with modals that help them select their preferred date and application times. An alert box with a success or failed message pops up and informs the user about the appointments current state depending on whether the appointment was successful or not. The appointment booking screen is illustrated in figure 4.9 below.

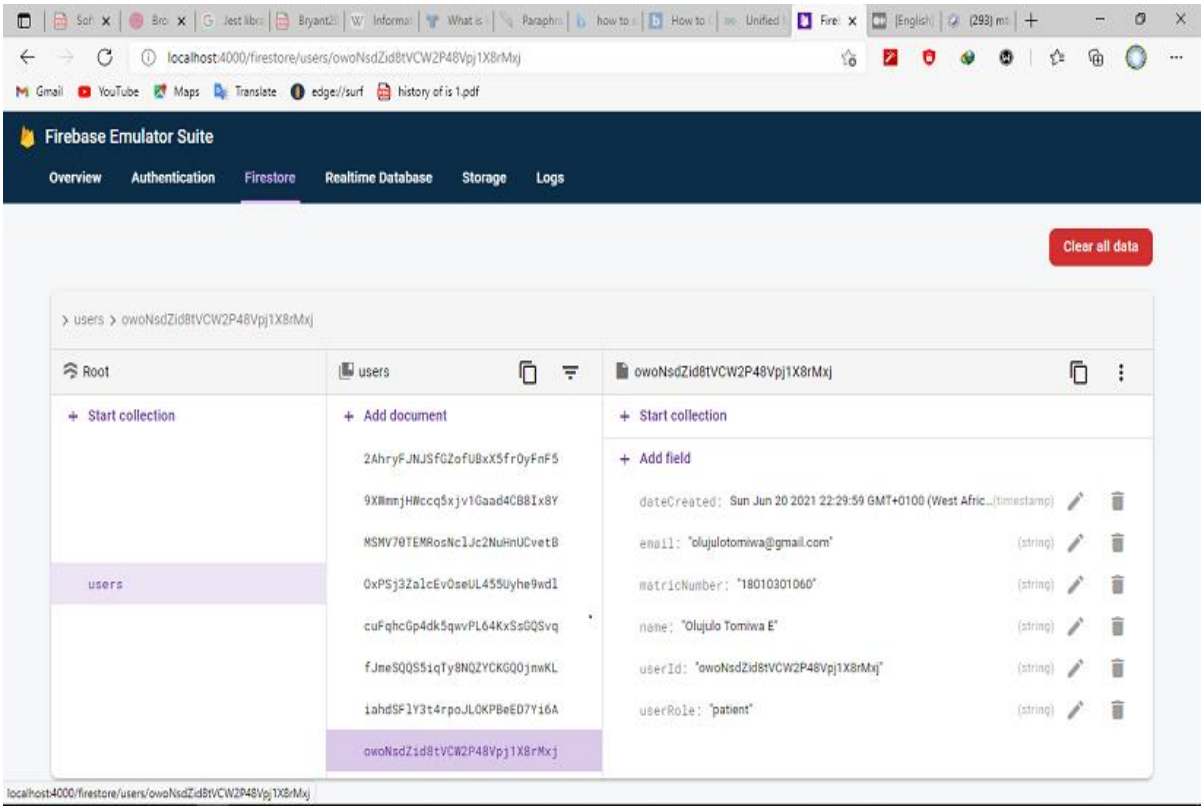

Figure 4.3 User-Information of System Users.

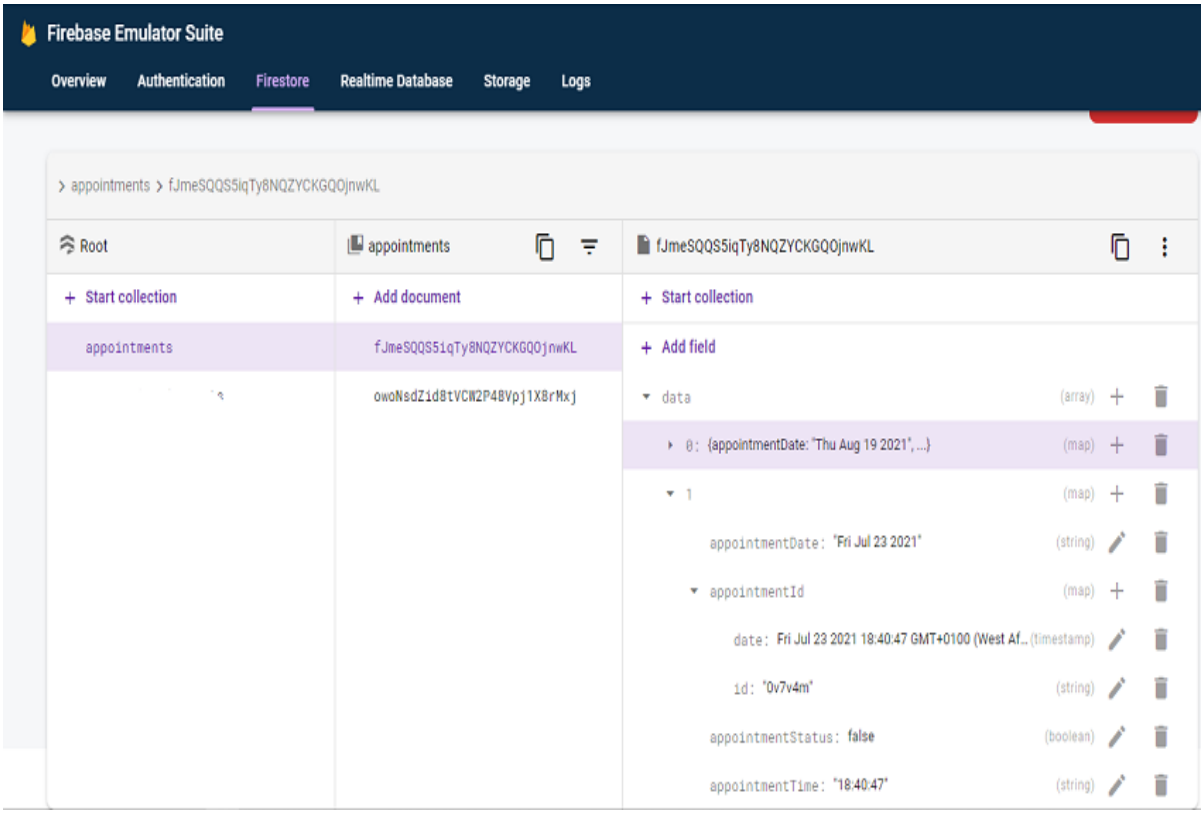

Figure 4.4 Individual User Appointments.

4.2.3 Appointment details

Every patient has an account screen from which they view their user details, view their successfully booked appointments, view a log of all their past appointments, and cancel appointment if there's a need for it. The patients get prompted with an alert box when they try to cancel an appointment to prevent mistakes. The patients can also log out of the software from the account screen. Illustrations are provided in Figures 4.10 and 4.11 below.

4.2.3 Validating an appointment

The nurses use the appointments screen to view all the lists of appointments that have been made for the day. The appointments are queried from the database with firebase syntax, filtered using javascript array accessing methods, and displayed to the nurse with react native flat-list. The nurses select appointments chooses whether to validate them and assign the appointments to doctors or to cancel them and notify the user to pick the date and time for a new appointment. The nurse appointment screen is represented in Figures 4.12 and 4.13 below.

4.2.4 Validated appointments

The doctors can view all the appointments that have been validated by the nurses via the validated_appointments_screen. The doctor also schedules new appointments for the patients if there is a need and reports medical data of patients after every appointment session.

4.2.5 Medical diagnosis

After every session with the patients, the doctor reports a log of the patients' data like her/his blood pressure, temperature, etc., to the system. The doctor can also selectively choose to reschedule appointments with the patients if there is a need to. The medical diagnostics screen is illustrated in figure 4.14 below.

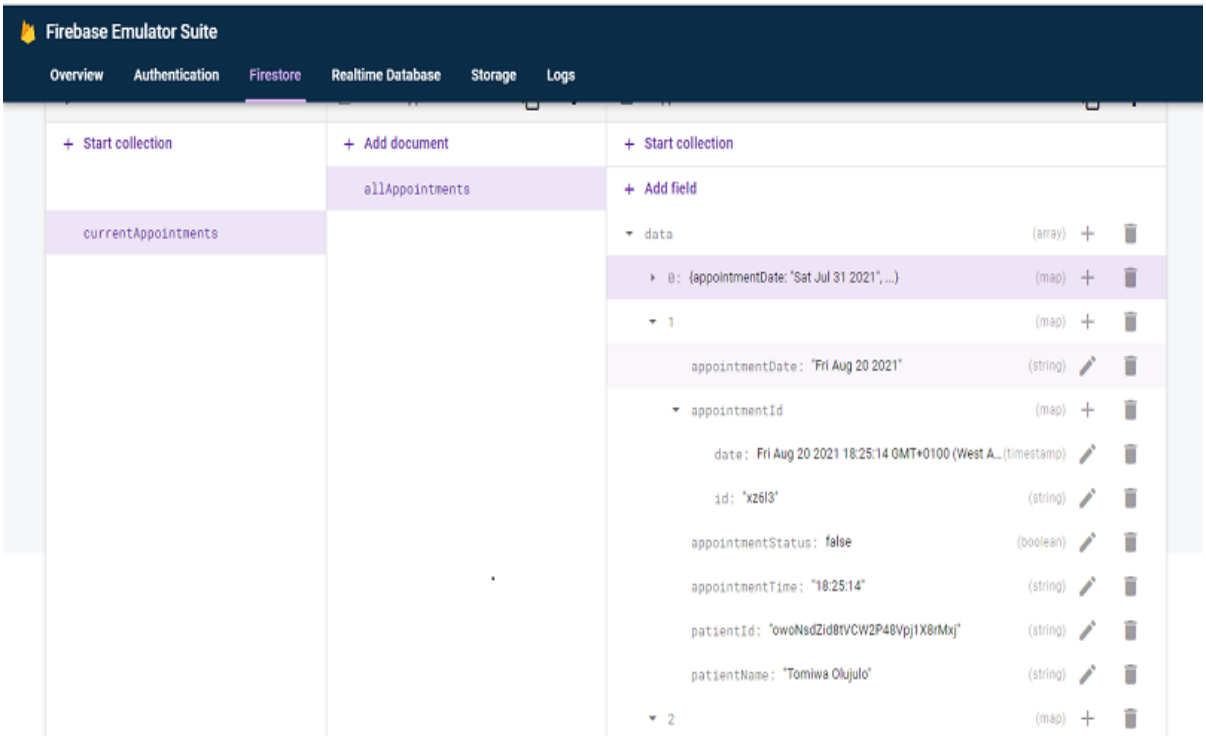

Figure 4.5 Active Appointments.

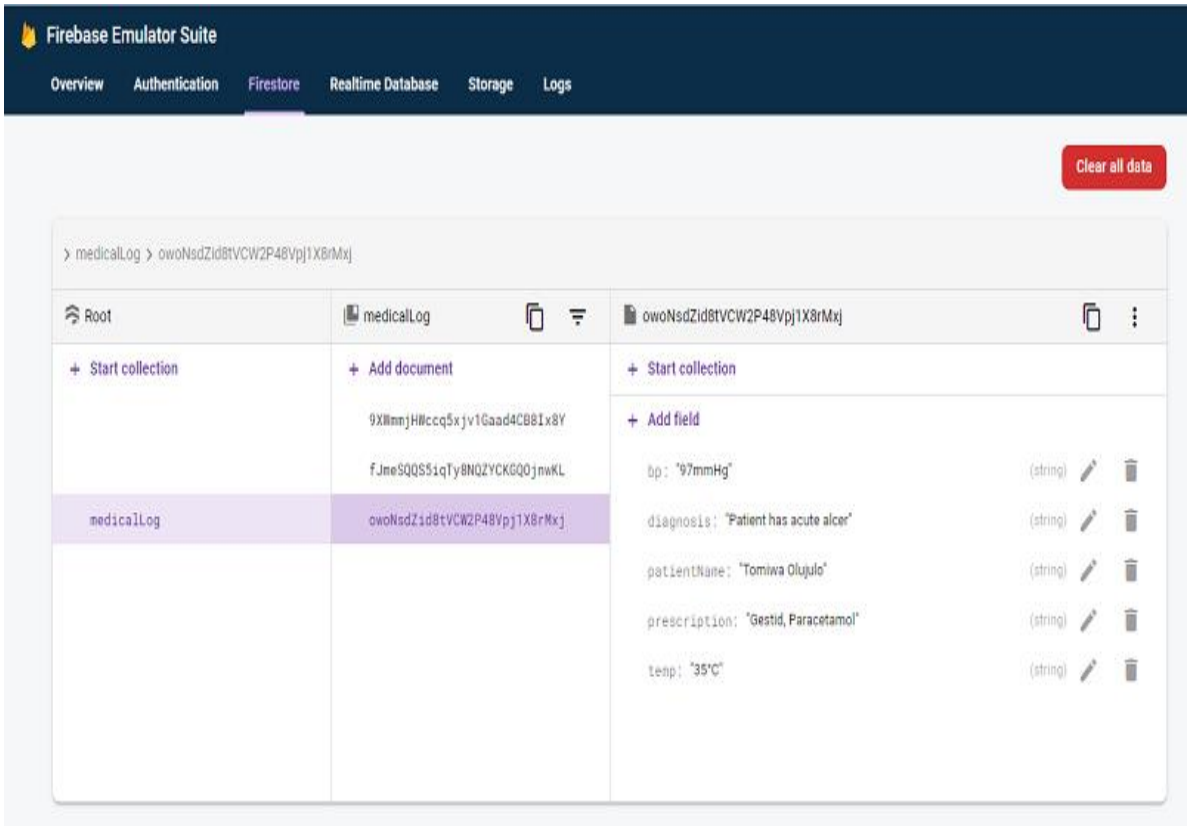

Figure 4.6 Patients Medical Log.

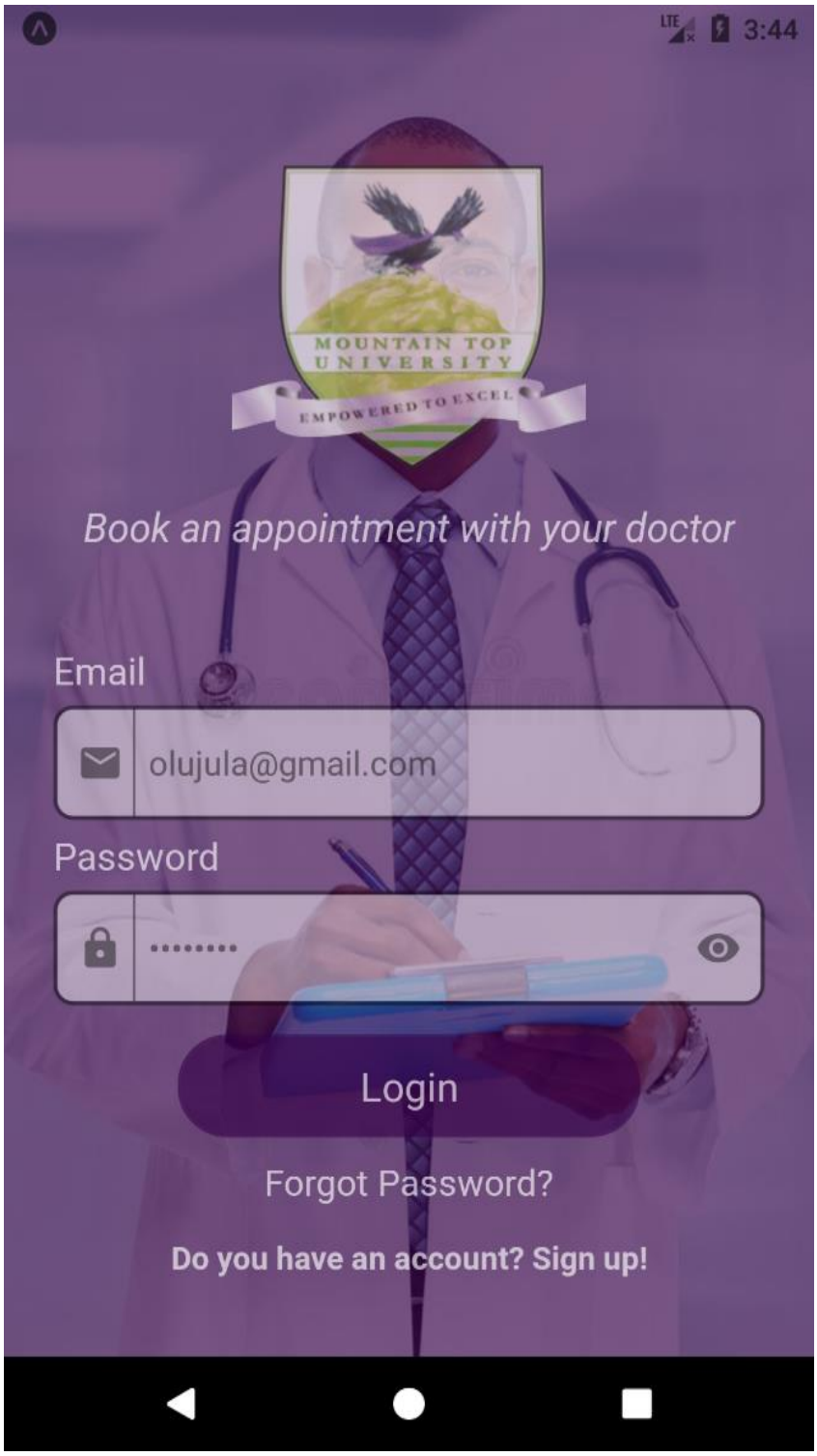

Figure 4.7 The Login Screen.

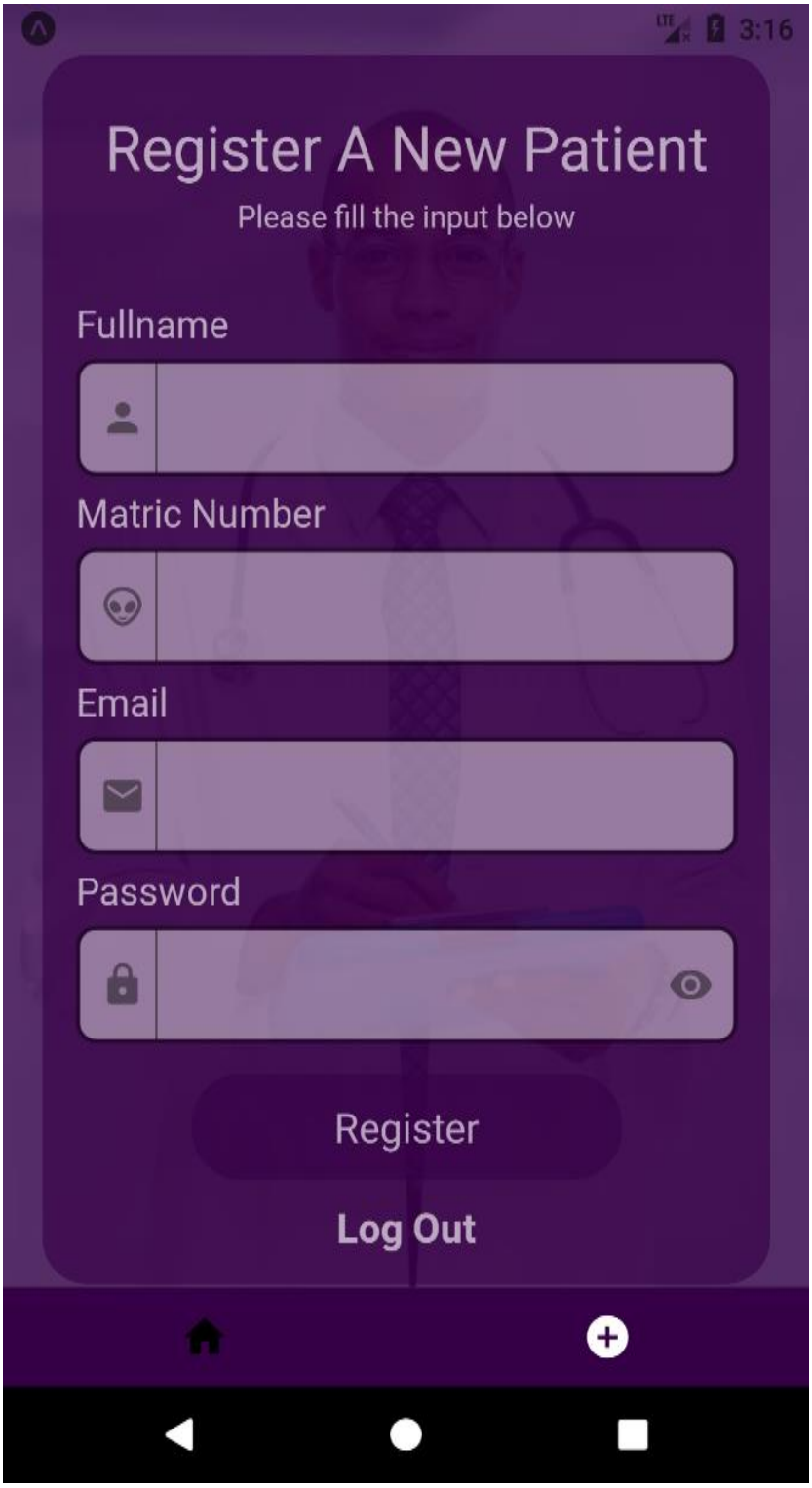

Figure 4.8 The Patient Registration Screen.

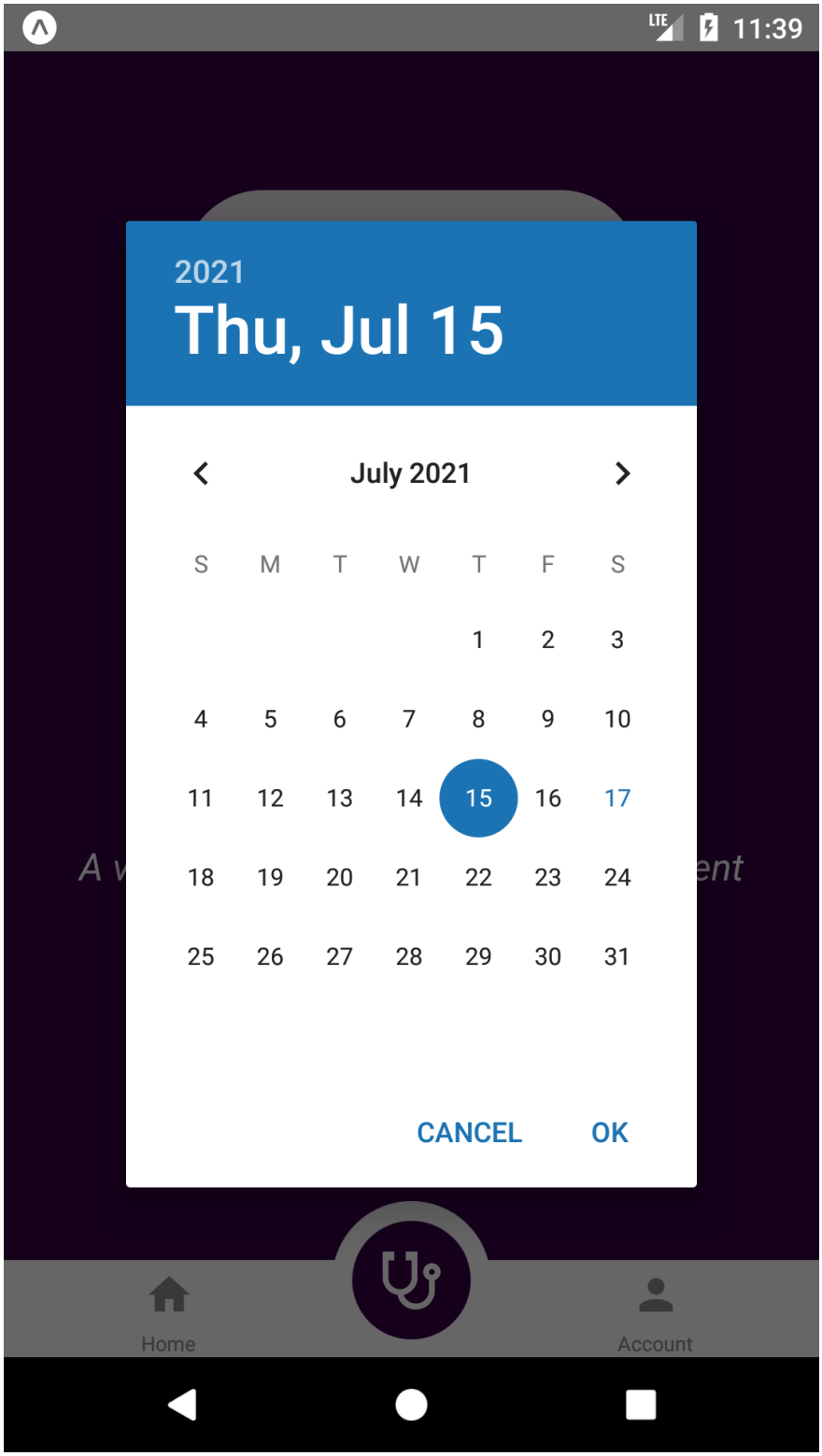

Figure 4.9 The Appointment Booking Screen.

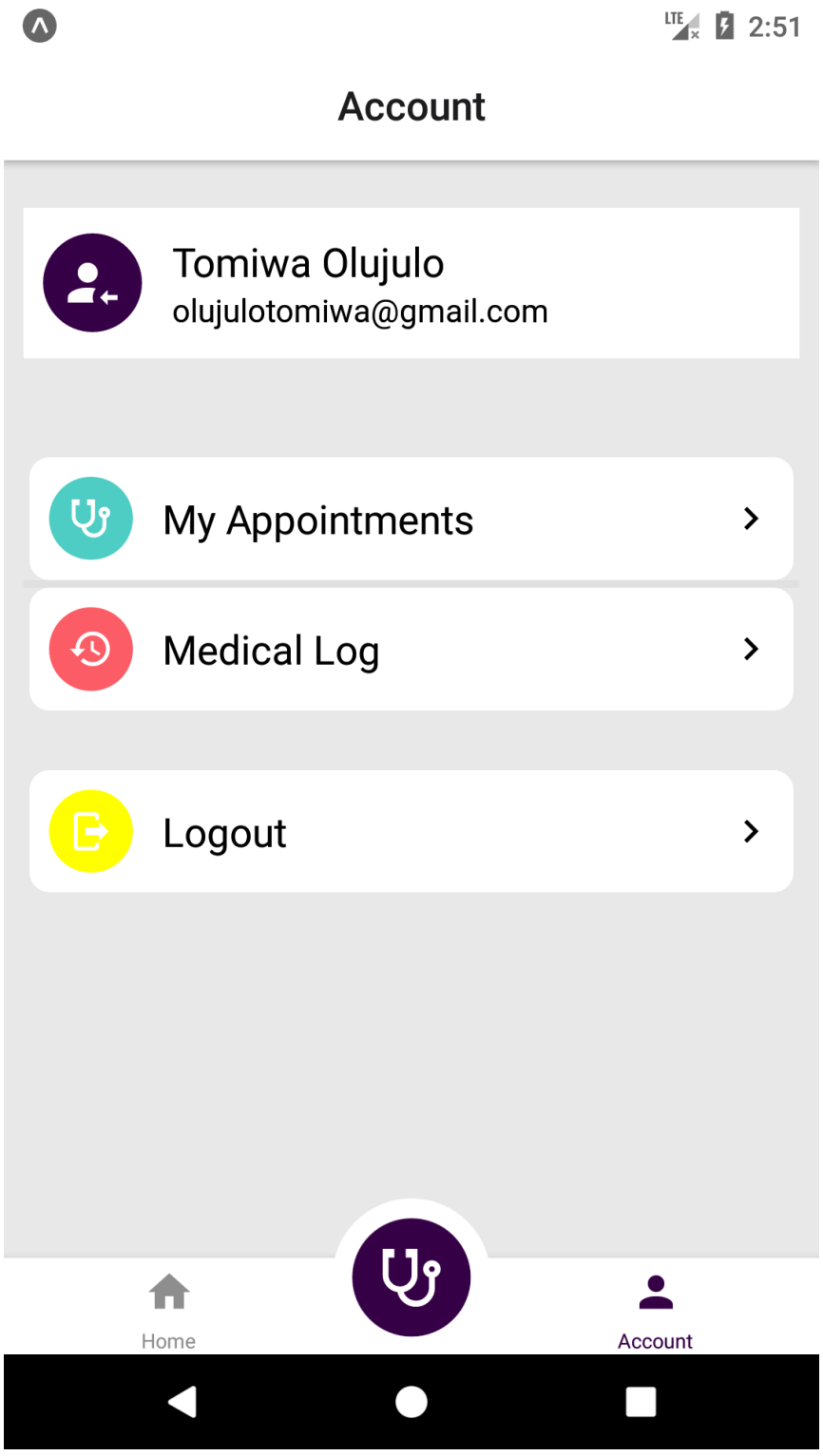

Figure 4.10 The Account Screen.

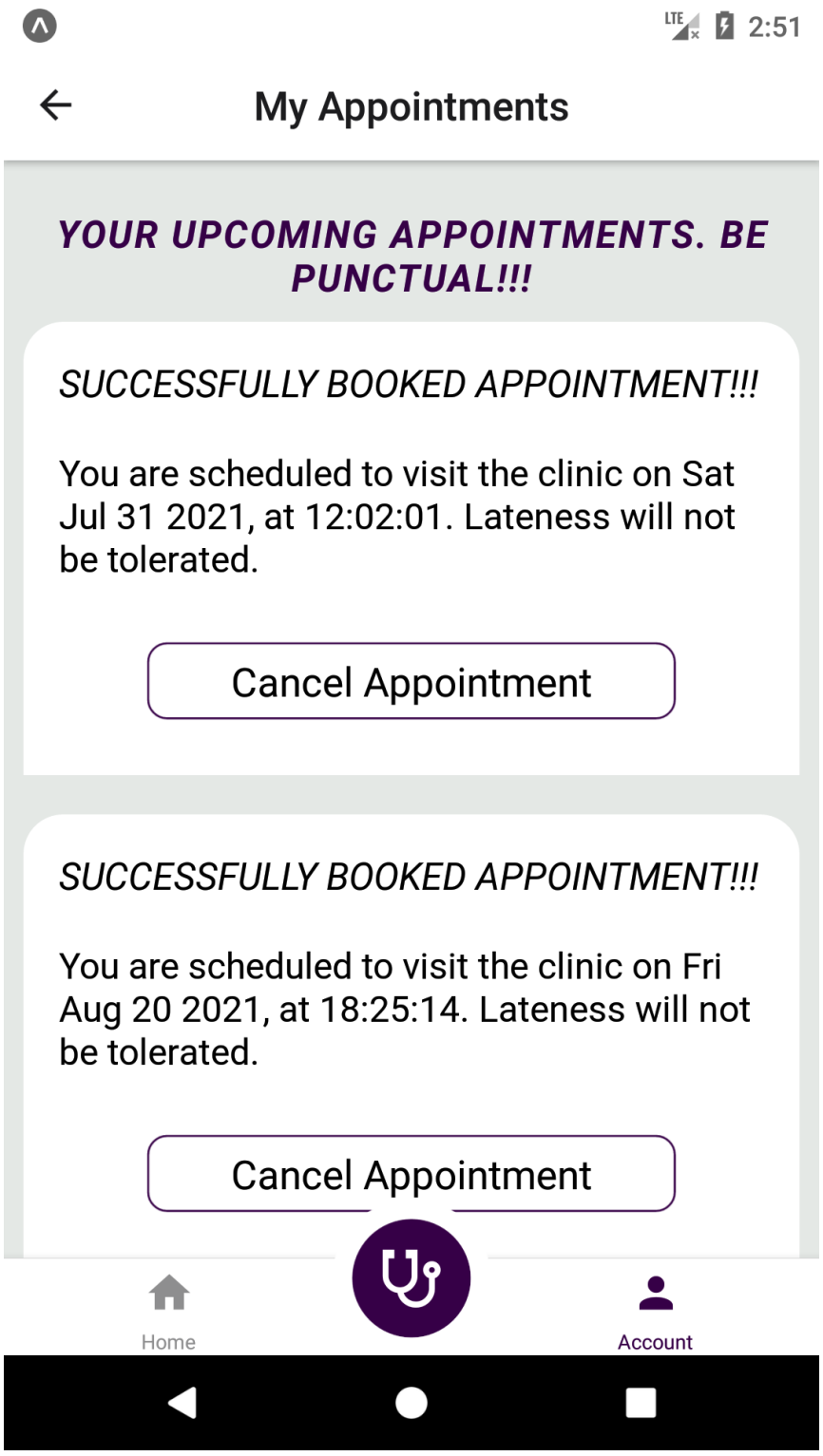

Figure 4.11 The User Appointment Screen

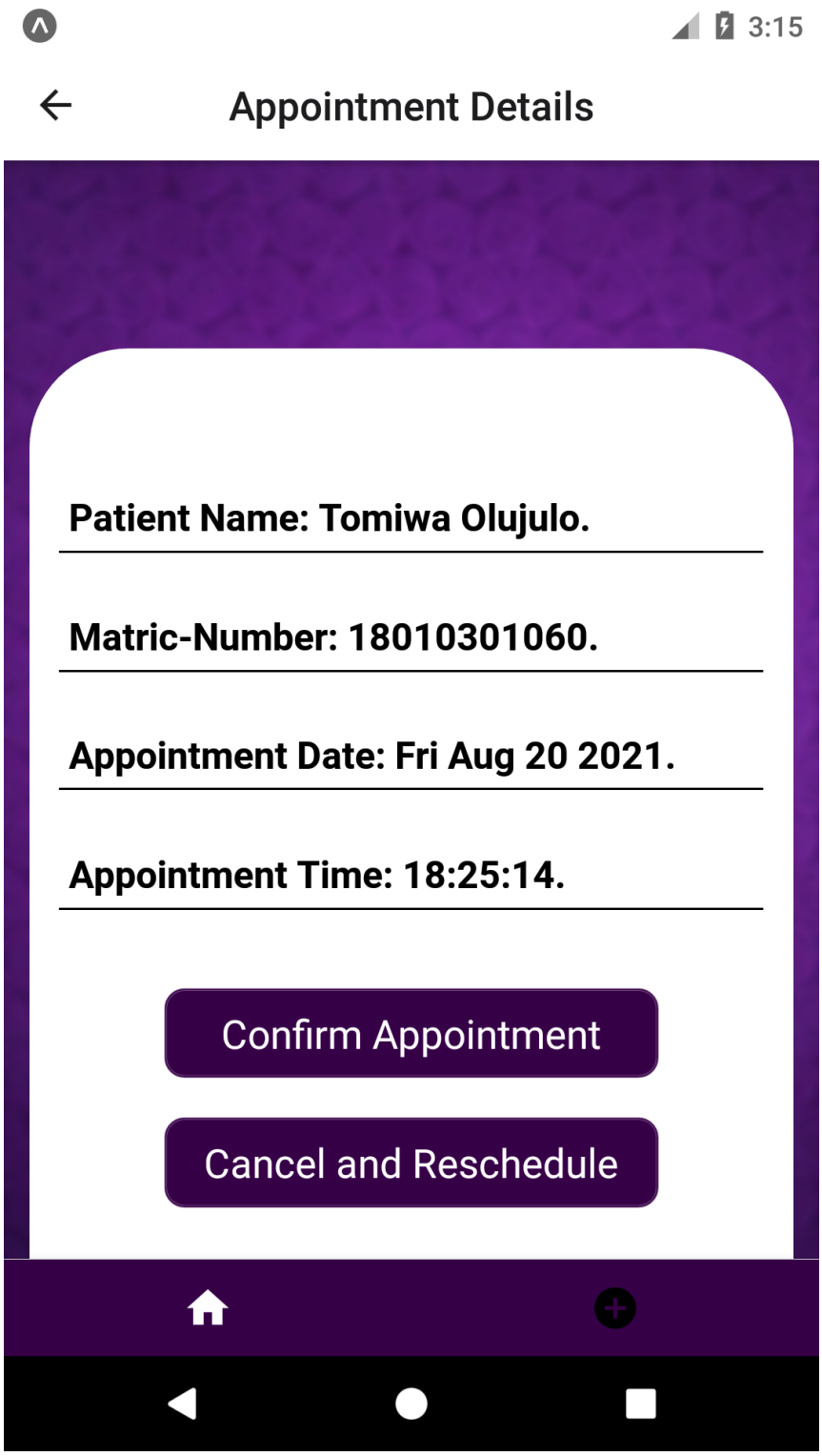

Figure 4.12 The Appointment Details Screen.

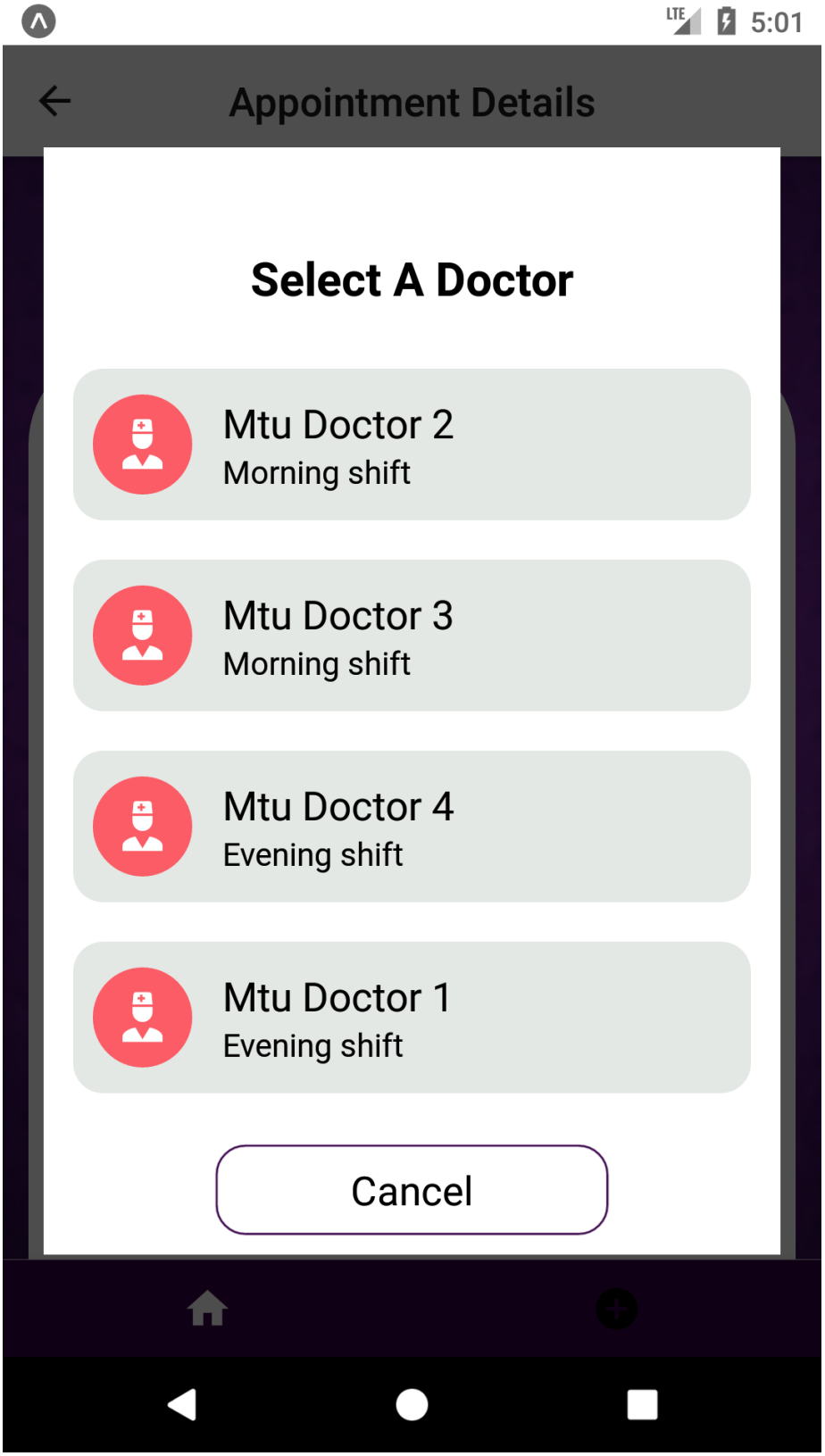

Figure 4.13 A Doctor Assignment Modal.

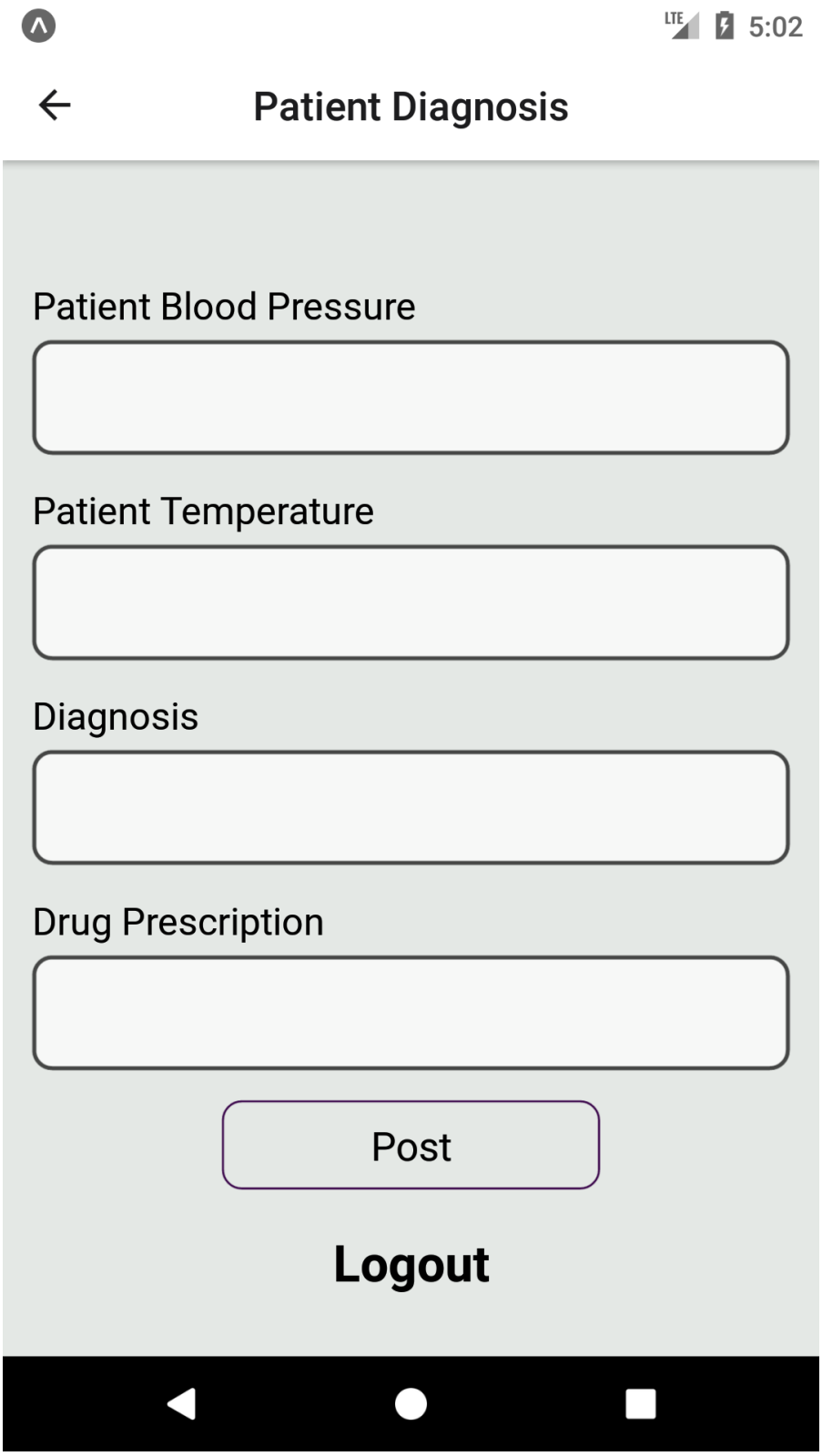

Figure 4.14 The Medical Diagnostics Screen.

CHAPTER FIVE

SUMMARY AND RECOMMENDATION

5.0 Summary

Health information systems have been around for a while and the postulations of Moore's law, as well as modern technology, have superbly aided its growth. Health facilities around the world are gradually employing health information systems to automate some of the health-related activities they carry out. A doctor appointment booking system is an example of a health information system. They are used to schedule doctor appointments to reduce waiting and service times of out-patients in a health facility. Appointment booking systems are a necessity in every healthcare facility as there are prevalent issues of waiting time that affects patient satisfaction, customer patronage, and the output of healthcare facilitators. Appointment booking systems are also useful in this COVID-19 era as they can be used to control patient traffic in healthcare facilities and curb the flow of the virus.

5.1 Recommendation

Implementing an online doctor appointment booking system would prove very useful to the mountain top university as it would help reduce the waiting time of patients and the workload of healthcare facilitators at the university. This study recommends that a scheduling algorithm should be used to schedule patient visitation to the healthcare facility to control the number of patients that would have appointments on an hourly basis.

Reference

Akinode J.L, O. S. (2017). Design and Implementation of a Patient Appointment and Scheduling System. *International Advanced Research Journal in Science, Engineering and Technology.*, 16.

Appointment-plus. (2015). *Manual Vs. Automated Appointment Scheduling.* Switzerland: appointment.plus.com.

Brook, C. (2020, December 1). *What is a Health Information System*. From Digital Guardian: https://digitalguardian.com/blog/what-health-information-system

Christiansen, L. (2021, January 6). *The six main types of information systems.* From Altametrics: https://altametrics.com/en/information-systems/information-system-types.com

Clue, R. (2020). *DESIGN AND IMPLEMENTATION OF APPOINTMENT MANAGEMENT SYSTEM (A CASE STUDY OF UNITHECH HOSPITAL)*. From Research-Clue: researchClue.com

cromacampus. (2021, March 18). *What Is Expo React Native?* From Vaccine Tours: https://www.vaccinetours.com/what-is-expo-react-native/

Garko, A. B., & Mahmud, U. (2017). DESIGN AND IMPLEMENTATION OF OUTPATIENT . *International Journal of Advanced Academic Research | Sciences, Technology & Engineering | ISSN: 2488-9849*, 15-21.

Geek for geeks. (2021, March 21). *Components of information systems*. From Geek for Geeks: https://www.geeksforgeeks.org/components-of-information-system.com

Geeks for geeks. (2019, September 6). *Types of information systems*. From Geeksforgeeeks: https://www.geeksforgeeks.org/types-of-information-system.com

GeeksforGeeks. (2019, April 1). *Unified Modeling Language (UML) | An Introduction*. From GeeksforGeeks: https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction

Goodfirms. (2019, Novemner 20). *What are Software Tools.* From Goodfirms: https://www.goodfirms.co/glossary/software-tools

Gregersen, E. (2019, March 31). *Information Systems - Definations, Examples & Facts.* From Britannica: https://www.britannica.com/topic/information-system

Hedges, L. (2021, April 7). *Online Booking Options Can Get You More Clients*. From GetApp: https://www.getapp.com/resources/research-online-booking-importance-of-appointment-scheduling/

Hoffer, J. A., Ramesh, V., & Heikki, T. (2011). *Modern database management.* New Jersey: Pearson Education, Inc.

Idowu, A. P., & Adeosun, O. O. (2014). DEPENDABLE ONLINE APPOINTMENT BOOKING SYSTEM FOR NHIS OUTPATIENT IN NIGERIAN TEACHING HOSPITALS. *International Journal of Computer Science & Information Technology (IJCSIT) Vol 6, No 4.*, 61.

Jude Altema, M. C. (2020, April). *Causes of long wait time in health consultation services and strategies to reduce them.* From The Lancet Global Health: http://www.lancetglobalhealth.com

Krukeja, S. (2021, June 7). *Six major types of information systems.* From Management Study HQ: https://www.managementstudyhq.com/six-major-types-of-information-systems.com

Laundon , K. C., & Laundon, J. P. (1988). *Management Information systems*. Macmillian.

Maryam, T. (2018, August). *ONLINE POLYCLINIC APPOINTMENT AND DATABASE MANAGEMENT SYSTEM.* From ONLINE POLYCLINIC APPOINTMENT

AND DATABASE MANAGEMENT SYSTEM: https://www.theseus.fi/bitstream/handle/10024/152390/Tufail_Maryam.pdf?sequence=1&isAllowed=y

Michas, F. (2020, July 2). *Daily average hospital census in the United States 1946-2018*. From Statista: https://www.statista.com/statistics/459736/average-daily-census-in-hospitals-in-the-us/#:~:text=This%20statistic%20displays%20the%20average%20daily%20census%20in,612%2C000%20people%20in%20hospitals%20located%20in%20the%20country.

Onyedifenu, G. (2020, March 4). *Ratio of Nigerian doctors to population is 1: 2753 – FG*. From BusinessDay: https://businessday.ng/uncategorized/article/ratio-of-nigerian-doctors-to-population-is-1-2753-fg/

O'Reily. (2015). *What is React Native.* O'Reily Media, Inc.

Reddy, N. (2020). *7 Benefits Of An Online Doctor Appointment Booking System In 2020*. From TABIIB: https://blog.tabiib.com/news/7-benefits-of-an-online-doctor-appointment-booking-system-in-2020/

Shrushti. (2020, January 5). *8 Types of Health Information Technology & Healthcare Software System*. From Software suggest: https://www.softwaresuggest.com/blog/types-of-health-information-technology-and-healthcare-software/

Sinhasane, S. (2019, June 26). *What Is Health Information System & Its Importance*. From Mobisoft: https://mobisoftinfotech.com/resources/blog/importance-of-health-information-system/

SmartDraw. (2021, July 14). *UML Diagram - Everything You Need to Know About UML Diagrams.* From SmartDraw: https://www.smartdraw.com/uml-diagram

Sommerville, I. (2011). *Software Engineering.* Boston: Pearson Education, Inc.

Team-fourUp. (2018, December 12). *ForeUp*. From Advantages and Disadvantages of Using an Online Booking System: https://blog.foreupgolf.com/advantages-disadvantages-online-booking-system

Techopedia. (2011). *Software Development Life Cycle Model (SDLC).* From Techopedia: https://www.techopedia.com/definition/25976/software-development-life-cycle-model-sdlc

Techopedia. (2011, March 6). *What is Information Systems or Information Services (IS).* From Techopedia: https://www.techopedia.com/definition/1027/information-systems-or-information-services-is

The art of service. (2019, October 24). *The six types of information systems and thier functions.* From The art of service: https://theartofservice.com/the-six-types-of-information-systems-and-their-functions.com

Tutorialspoint. (2021, June 30). *SDLC Tutorial.* From Tutorialspoint: https://www.tutorialspoint.com/sdlc/index.htm

Tutorialspoint. (2021, July 14). *UML Tutorial.* From Tutorialspoint: https://www.tutorialspoint.com/uml/index.htm

Xie, Z., & Or, C. (2017). Associations Between Waiting Times, Service Times, and Patient Satisfaction in an Endocrinology Outpatient Department: A Time Study and Questionnaire Survey. *The Journal of Health Care Organization, Provision, and Financing Volume 54: 1–10*, 5-25.

Zwass, V. (2020, November 12). *Information systems- Encyclopaedia Britannica*. From Encyclopaedia Britannica: https://www.britannica.com/topic/information-system