DOCHUB

NAME

SESSION YYYY/YYY

FACULTY OF XXXXXX

UNIVERSITY

MONTH DAY

ABSTRACT

TABLE OF CONTENTS

[Abstract II](#_Toc291065738)

[Table of contents III](#_Toc291065739)

[List of tables V](#_Toc291065740)

[List of figures VI](#_Toc291065741)

[LIST OF ABBREVIATIONS/ SYMBOLS VII](#_Toc291065742)

[List of appendices VIII](#_Toc291065743)

[CHAPTER 1 INTRODUCTION 1](#_Toc125875912)

[1.1 Overview 1](#_Toc125875913)

[1.2 Problem Statement 2](#_Toc125875914)

[1.3 Objectives 3](#_Toc125875915)

[1.4 Project Scope 3](#_Toc125875916)

[1.5 Project Schedule 5](#_Toc125875917)

[1.6 Report Organisation 6](#_Toc125875918)

[CHAPTER 2 LITERATURE REVIEW 8](#_Toc125875919)

[2.1 Introduction 8](#_Toc125875920)

[2.2 Existing Systems 10](#_Toc125875921)

[CHAPTER 3 ANALYSIS AND DESIGN 16](#_Toc125875922)

[3.1 Overview 16](#_Toc125875923)

[3.2 Considered Application 16](#_Toc125875924)

[3.3 Use Case 17](#_Toc125875925)

[3.4 Application Prototype 21](#_Toc125875926)

[3.5 Back-end Analysis 27](#_Toc125875927)

[CHAPTER 4 IMPLEMENTATION 37](#_Toc125875928)

[4.1 Overview 37](#_Toc125875929)

[4.2 High-Fidelity Prototype 37](#_Toc125875930)

[4.3 Backend Implementation 47](#_Toc125875931)

[CHAPTER 5 TESTING AND EVALUATION 53](#_Toc125875932)

[5.1 Overview 53](#_Toc125875933)

[5.2 Testing Procedure 53](#_Toc125875934)

[CHAPTER 6 CONCLUSION 60](#_Toc125875935)

[6.1 Concluding Remarks 60](#_Toc125875936)

[6.2 Future Work 60](#_Toc125875937)

[REFERENCES](#_REFERENCES_1) 2

[APPENDICES 2](#_APPENDICES)

LIST OF TABLES

[Table 1.1: Task distribution 5](#_Toc125875938)

[Table 2.1: Overall Comparison of Related System 13](#_Toc125875939)

[Table 4.1: API Definition 49](#_Toc125875940)

[Table 5.1: Unit Testing Module for Authentication Function 54](#_Toc125875941)

[Table 5.2: Unit Testing Module for Appointment Function 54](#_Toc125875942)

[Table 5.3: Unit Testing Module for User Profile Function 55](#_Toc125875943)

[Table 5.4: Unit Testing Module for Virtual Consultation Function 55](#_Toc125875944)

[Table 5.5: Unit Testing Module for Feedback Function 55](#_Toc125875945)

[Table 5.6: Usability Testing Module for Authentication Function 56](#_Toc125875946)

[Table 5.7: Usability Testing Module for Appointment Function 57](#_Toc125875947)

[Table 5.8: Usability Testing Module for User Profile Function 58](#_Toc125875948)

[Table 5.9: Usability Testing Module for Virtual Consultation Function 58](#_Toc125875949)

[Table 5.10: Usability Testing Module for Feedback Function 58](#_Toc125875950)

LIST OF FIGURES

[Figure 1.1: Gantt Chart Diagram 6](https://d.docs.live.net/12155eb2cacb7a61/Work%20Files/Kemi/Thesis/FYPDocumentationTemplate201509.docx#_Toc125875951)

[Figure 2.1; And.do To do List Interface 14](#_Toc125875952)

[Figure 2.2:Appointment Planner Application Interface 15](#_Toc125875953)

[Figure 3.1: DocHub Use Case Diagram 18](#_Toc125875954)

[Figure 3.2: DocHub Task Graph 23](#_Toc125875955)

[Figure 3.3: Low Fidelity Prototype (a)Registration page (b)Login page (c)Doctor detail page (d)Home page (e)Doctor list (f)Create appointment page 25](#_Toc125875956)

[Figure 3.4: System Architecture 28](#_Toc125875957)

[Figure 3.5: Entity Relationship Diagram 30](#_Toc125875958)

[Figure 3.6: Context Diagram 31](#_Toc125875959)

[Figure 3.7: Data Flow Diagram 31](#_Toc125875960)

[Figure 3.8: Login Flow Chart 32](#_Toc125875961)

[Figure 3.9: Registration Flow Chart 33](#_Toc125875962)

[Figure 3.10: Get Doctor 34](#_Toc125875963)

[Figure 3.11: Appointment Flow Chart 35](#_Toc125875964)

[Figure 3.12: Medication Flow Chart 36](#_Toc125875965)

[Figure 4.1: Authentication Screen (a) SignIn;(b) SignUp; (c) Forget-Password 38](#_Toc125875966)

[Figure 4.2:Patient Home Screen (a) Default;(b) With dropdown menu 39](#_Toc125875967)

[Figure 4.3: Doctor Home Screen (a) Default;(b) With dropdown menu 39](#_Toc125875968)

[Figure 4.4: Account Screen (a) Default;(b) Edit Profile; (c) My Feedback 40](#_Toc125875969)

[Figure 4.5: Appointment Screen (a) Book Appointment;(b) Book Appointment with Mode options; (c) Appointment List; (d) Appointment Detail; (e) Cancel Appointment Menu 42](#_Toc125875970)

[Figure 4.6: Doctor Screen (a) Doctor List;(b) Doctor List with Filter; (c) About Doctor; (d) Doctor Feedbacks; (e) Give Feedback 43](#_Toc125875971)

[Figure 4.7: Patient Screen 44](#_Toc125875972)

[Figure 4.8: Medication Screen (a) Medication List;(b) Pop-Up Medication 45](#_Toc125875973)

[Figure 4.9: (a) My Doctor List; (b) My Patient List 46](#_Toc125875974)

[Figure 4.10: Virtual Consultation Room 46](#_Toc125875975)

LIST OF ABBREVIATIONS/ SYMBOLS

LIST OF APPENDICES

[**Appendix A: XXX** 2](#_Toc346727700)

[**Appendix B: XXX** 2](#_Toc346727701)

# INTRODUCTION

### Overview

Every day, a number of interactive systems are being developed in many application domains, including as healthcare, transportation, and education, to enhance their processes. Recent polls have shown that the use of technology, notably smart devices like iPads and Android tablets, has increased productivity across most industries, including the health industry. According to reports, more than 70% of doctors working in US hospitals use either smartphones, tablets, or both as their working device (Colley et al., 2015). People go to cities as a result of urbanization and technological growth, which makes them crowded. how many individuals change their lives and go to these places in search of better living circumstances. Every institution, especially healthcare facilities, have become more busier than expected. It is estimated that over 2.8 billion people are predicted to live in 750 of the biggest cities in the globe by 2030. (Vize, R., 2017).

Numerous healthcare problems, like hospital overpopulation and a lack of medical supplies, may be seen. Not only that, patients' waiting time also lengthen when they wait to be processed and seen by a doctor. During this time patients may roam about the hospital's grounds, making it more difficult for employees to find them. In addition, health industry has recently been on high alert as a result of the ongoing Covid 19 outbreak and has used a number of techniques to streamline its operations. In hospitals, patients are instructed to abide by tight engagement guidelines and are only expected to present in instances that are of the utmost importance. In order to stop the virus from spreading, fewer individuals will be in the hospitals at one time as a result. However, due to the new guidelines patients with less urgent issues get less attention from hospitals and physicians because interactions between patients and doctors are restricted to meetings that are of the utmost importance.

In Nigeria, the health care facilities are filled with overwhelmed doctors who have a lot of patients to attend to within a short period of time and which has proved abortive because the few doctors available are not able to deliver. It was established by the National Primary Health Care Development Agency (NPHCDA), which is a parastatal of the Nigeria’s Federal Ministry of health that there is a ratio of four to ten thousand in terms of doctors to Nigerians (Muanya et al., 2022). In addition, Doctors wages in Nigeria in comparison to other developed country is underwhelming, therefore, they are on strike for the major parts of the year. In 2022 alone, doctors were on strike three time over the demand of allowances for treatment carried out on Covid-19 patients (Adebowale-Tambe, 2022). The absence of an implemented healthcare information system has escalated the effects of these problem, whereby hospitals are generally congested with patients and no doctors to attend to them. This is evidently due to the fact that most patient cannot ascertain the availability of their doctors at any given time.

Because of this, improved resource management and raising hospital productivity are essential to improving patient care. The major objective of the system in the proposed project is to develop seamless communication between patients and various kinds of hospital physicians through a mobile application. With the aid of the mobile application patients who don't need to be at the hospital physically may engage with their physicians in some way. Consequently, fewer individuals will be at the hospital, which will also result in shorter wait times for those who need physical consultations with the physicians.

### Problem Statement

Interaction between doctors and patients is a crucial aspect of what is done in the health industry (Dennis & Newman, 2017). A number of structural techniques have been used to aid with this process. Despite the fact that patients now have restricted access to their physicians on a consistent basis, the recent Covid-19 epidemic has made these issues worse, endangering the structural process as it is now. Due to the physicians' failure to respond to their patients' urgent needs, this constraint has led to patients' transitioning from minor medical issues to more serious ones.

Patients seldom have access to doctors' availability until they are on the hospital grounds. The majority of the time, people who need a doctor's consultation from a certain hospital department come to the hospital in the hopes of finding one who is on call. In the event that no physicians are available, the patient is required to wait until a spot opens up; if none do, the patient is advised to reschedule or make appointment for another time.

### Objectives

The objectives of the proposed system are as follows:

* Design a mobile application where patient and doctors can have a seamless interaction for an appointed time created by patient and confirmed by the doctor.
* Provide a platform for easy evaluation of Doctor performance and service evaluation through patient remarks
* Provide a platform on the designed mobile application for patient to make appointment for either physical consultation or virtual through the application based on doctors’ availability.

### Project Scope

The project scope is determined by how many modules are projected to be developed for this system. The three module types are general, medical, and patient. The general module's features are used by both patients and doctors. The other two modules, patient and doctor, are intended for use by patients and physicians, respectively.

#### General Modules

The general modules are components of the system that is used by both patients and doctors. The general modules include the following modules.

* Authentication
* Database
* Mobile Interface/Interaction

The authentication module serves as a security layer for the system. The following functions are in the authentication module: (Registration, Login, Forget Password, Logout). All the functions present in the authentication module is to be utilized by the patients. The doctors do not require the registration function because their details are placed in the system by the system administrator. The database module provides connection and access to the system database. Every information about all the users in the system is saved in the database. The database module contains all the database function which allows the user to access their information in the database. The last module is the mobile interface module. This module provides link from one page to another page in the system. In addition, the interaction through chat, video and audio call is serviced by this module.

#### Patient Module

The patient module holds functions that are specifically for the patients. This module consists of the following functions.

* Profile
* Find Doctor
* Appointment
* Medication
* Feedback

The profile module provides the patients with functions to view their profile and edit/update specific information on their profile. The find doctor module provide function to help patients to find doctors based on their departments and expertise in the hospital. The exact location of the doctor office in the hospital can be accessed through this function. The appointment module provides functions to help patients view their appointment, make appointment, or cancel and existing appointment. The medication module provides function to help patients view their prescriptions from the doctor or make a new request for a medication. The feedback module allow patients provide feedback about previously encountered doctors.

#### Doctor Module

The doctor module is holds functions that are specifically for the doctors. This module consists of the following functions.

* Appointment
* Medication

The appointment module provides functions to help doctors view their appointment. All the appointment slots are created by the system administrator based on the working hours of the doctor. The medication module provides function to help doctors view existing prescriptions based on the patients and make new prescriptions or medical recommendations to patients.

### Project Schedule

This section illustrates the progress of the development of the system. The time frame of each process which includes development and documentation is illustrated in a task distribution table and the Gantt chart diagram.

Table 1.1: Task distribution

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Start Date** | **End Date** | **Duration (Days)** |
| Chapter 1: Introduction   * Requirement gathering * Project scope and Report Organization | 15-08-2022 | 26-08-2022 | 12 |
| Chapter 2: Literature Review   * Existing similar systems * Existing design Platform | 22-08-2022 | 23-09-2022 | 32 |
| Chapter 3: Analysis and Design   * System development life cycle method * System Interface design * Database Design | 24-09-2022 | 15-10-2022 | 22 |
| Chapter 4: Implementation   * Front-end Design * Back-end Design * System integration and server setup | 01-10-2022 | 12-11-2022 | 43 |
| Chapter 5: Testing/Evaluation | 13-11-2022 | 24-12-2022 | 41 |
| Chapter 6: Conclusion | 20-12-2022 | 25-12-2022 | 6 |

Chart, timeline

Description automatically generated

Figure 1.1: Gantt Chart Diagram

### Report Organisation

This report is organized based on different chapters. These chapters include, introduction, literature review, methodology, design, implementation, and conclusion.

In chapter 1 Introduction, we discuss the overview of the entire system which involves the motivation behind the development of the system, the problem statement and the objectives. In addition, the expected time of completion of the system is determined and illustrated in a table and a Gantt chart diagram.

In chapter 2 Literature Review, we evaluate different approaches in the design of mobile applications. The programming languages that are suitable, advantages and disadvantages. Also, we discuss existing systems that are similar in functions to the system we intend to design. At the of these chapter, all the system are evaluated and compared based on their advantages and disadvantages.

In chapter 3 Analysis and Design, we modelled the design of the system based on the system development life cycle, using the prototyping model. The tools that are used in the development of the system is discussed in this chapter and the reason why these tools are employed. Furthermore, the model of the system is created based on its functions and the user interface. The system database design is also discussed in this chapter.

In chapter 4 Implementation, in-depth discussion on the implementation process of the system is discussed in this chapter. The file management of the system is illustrated as well as all the tools and libraries used in the development of the system.

In chapter 5 Evaluation and Testing, each of the objective of the system are tested and evaluated to meet the requirements stated in the chapter 1 of this paper.

Finally, chapter 6 Conclusion explains the entire overview of the project and the entire work that has been done. This will explain the first phase of the project and the second phase of the project. Also, suggestion on future improvements that can be carried out on the system.

# LITERATURE REVIEW

### Introduction

The requirements and objectives of many areas of our life has been altered as a result of the frequent and fast developments in technology. Additionally, improvements in mobile technology have greatly enhanced the way we go about our everyday lives. Because smartphones and other mobile devices are such an integral part of our everyday lives, a variety of software has been developed to cater to the requirements and expectations of individuals. Doctors and other medical professionals often employ smart gadgets to lessen the stress of their everyday job in a variety of medical specialties (Colley et al., 2015). However, the current Covid-19 pandemic has significantly impacted hospital operations, including but not limited to the amount of patients the facilities can house at one time. In order to create place for urgent cases, non-extreme ones that need physicians' attention are delayed. Therefore, there was an introduction of virtual interactive medium to complete consultation carried out between doctors and patients. This is mostly done over mobile devices and this process is called Telemedicine (Morishita et al., 2021).

Telemedicine, sometimes known as interactive telemedicine, has been used in Colombia to facilitate communication between medical personnel and patients. Through currently available video conference platforms like Microsoft Teams, healthcare services are given to patients. This platform was selected due to its promise to preserve patients' private information, as well as its validity, integrity, and system accessibility. Following the consultation, patients are sent PDF files containing information on the consultation through email. Over 50,000 thousand patients were consulted through interactive telemedicine in the first nine months of Colombia's Covid-19 declaration, demonstrating the viability of this concept. The efficiency produced a 91.5% total success rate, according to the records (Fernanda Escobar et al., 2021). Telemedicine was initially authorized in Japan in 1997 and was specifically designed for patients in distant places who needed dental treatment but had little access to a hospital. They must have initial in-person meetings before moving on to follow-up ones that may be conducted through online consultation. The was just implemented to support the in-person consultation. In 2019, access is extended to additional areas and regulations are updated as a result of various successes in the telemedicine industry. But the first visit still has to include a face-to-face conversation. The limitation on using telemedicine during the first hospital visit was ultimately abolished by the Japan Ministry of Health, Labour, and Welfare in 2020 as the Covid-19 epidemic expanded throughout the country (Morishita et al., 2021).

Pediatricians in Italy have used telemedicine for diagnosis and information sharing. Different doctors employ a variety of techniques to do the duty, including personal email, instant messaging software, hospital mobile phones, personal cell phones, etc. The use of telemedicine in Italy is very limited. This is as a result of insurance companies' incapacity to follow therapy in order to pay hospitals for services rendered. Because it requires people to pay out-of-pocket in Italy, telemedicine is often disapproved of by both patients and medical professionals. As a result, compared to other big cities, the impact of the Covid-19 epidemic in Italy was shown to be more severe (Omboni, 2020). We may draw the conclusion that the use of mobile devices in medicine is practical and that doctors can consult with their patients via virtual procedures, as is common in the nations mentioned above. However, this is based mostly on developed countries or standard health care system with standard patient record management.

In recent study of the effect of Telemedicine in Africa with Mali as a case study, one of the problems encountered is the digital divide that’s in the country. The access to digital technology isn’t universal and in the country which creates a problem of implementing a digital health care system (Geissbuhler et al., 2003). This can be said to be the case for other West African countries including Nigeria. Proving health services through mobile devices can be difficult due to this setback. However, this doesn’t prevent the implementation of the system to 40% of West African who is considered to be operating a mobile device (OECD, 2021). In the next section, we will be evaluating existing systems created for medical services to reduce problems encountered when providing services and enhance health care productivity.

### Existing Systems

The aim of the system to be designed is to reduce the rate of clusters at health centre and improve the effectiveness of the medical practitioners using an appointment system. In the medical domain, the goal of an appointment system is to offer a flawless method and establish an extraordinary congruence between patients, doctors and other stake holders in the implementation of services rendered in the hospital. The most common methods used in show schedule and availability of doctors or medical practitioners at hospitals involves either putting up sign-up sheet on the doors or release a fixed time over a period which patients can reserve a consultation time. When managing consultation times, for instance, if the doctor just releases the consultation time, several patients may wish to meet at the same time, which might cause congestion in the hospital. These approaches do not respond to dynamic changes or cancellations of appointments.

People who recognized the potential of the internet and resolved to reap its full rewards created online appointment systems to enhance the dynamic process of the appointment system. In actuality, developers concentrated on the issues with the standard appointment procedure so they could create software and websites that would address these issues. There are similarities and distinctions among these systems despite the fact that they are not all the same. The following systems that were created by various individuals in various locations worldwide will be discussed and their performance will measure against the following features which are expected to be present in the proposed system.

* Online Feature
* Mobile Application
* Cross-platform Compatibility
* Reminders
* Location
* Notification
* Recurring appointment
* Virtual Consultancy (Audio, Video, and Text)

The system called ‘Mwa3edk’ was developed by (Odeh et al., 2019) with the aim of providing an appointment booking platform for patients to have access to doctors in medical clinic by transferring the process to an online technology. The system consists of three parts which includes the mobile application, a webpage and a central database. The mobile application is used by both patients and doctors to manage all their affairs regarding appointments. The webpage is the administrator section of the system to manage the system user and database content. The central database connects all the clinics, hospitals and users registered on the system. In addition to the appointment system, an embedded expert system is integrated into the system which allow patients perform simple queries about their problem to get fast diagnosis without having to see or make an appointment with the doctor. The expert system requires an extensive knowledge base depending on the amount of diagnosis the system is able to provide.

“BOOKAZOR” by (Anish Kumar S, 2019) is a web-based used for booking appointment in hospitals, streams of parlor and architects within a geographical area. The main functionalities provided by the system is to allow user make appointment with different job providers like doctors, architects, lawyers, maid etc. The appointment for a job provider is stored as request which is updated on the job providers route periodically by a jog scheduler. The job providers availability is updated based on the time provided by the job providers. The users on the system are only able to make appointment with job providers that are geographically nearby. Job providers on the system can register as a company or an individual, with details about the job description and category of job provided during the registration process. To assist users in navigating the system's pages, the system also included a chatbot. When a chatbot presents a question that cannot be resolved automatically, the admin staff is notified and will respond appropriately.

“MeetUp”, an appointment booking system designed using Flutter and Django framework by (Bhangale et al., 2021) was utilized in the educational domain for teachers and students to schedule meetings. The major advantage of this system over the previous mobile application designed is the ability to be used across different platforms like Android and iOS. Due to the functionality of the design environment used, the developer is only required to write just one code which can be converted to both Android and iOS mobile application. The Flutter framework used in the design doesn’t depend on any platform because it has its own widgets and design parameters. The teacher upon registration uploads their timetable and the system automatically generates a free time slot for the teacher which can be selected during booking by the students. However, this makes changes in schedule difficult to update because every update in schedule will require the teacher to upload a new timetable. This process of making changes is not flexible for the Teachers. In addition, appointments which has been made before the schedule is updated are cancelled which will require the student to make a new on, this makes the entire process cumbersome.

A mobile application for Doctor appointment scheduling (MADAS) is designed by (Usharani et al., 2021) with the major difference compare to previous mobile application designed been its feature to auto assign doctors to patients. The system allow patients pick time and date of appointment with the kind of doctor which they intend to have consultation. With this information, the system assigns an available Doctor available at this particular time to the patient or present a list of Doctors available to choose from if there are more than one available. The Doctor is therefore notified of the new appointment. In addition, this feature allow patient to make appointment in real-time. The application also offers a live video appointment with the Doctor.

Although not every system is the same, as was previously noted, they have all been created to make scheduling visits with doctors easier or in some cases other form of service providers. The virtual consultation feature makes the proposed application special and differentiates it from the other existing application. Although the proposed application from (Usharani et al., 2021) have the virtual consultation feature but its only focus on video consultation. The proposed system would implement other modes of virtual consultation like audio calls and text conversations.

Table 2.1: Overall Comparison of Related System

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Features | Online Feature | Mobile Application | Cross Platform Compatibility | Reminders | Location | Notification | Virtual Consultation |
| Systems |
| Proposed System | Checkmark outline | Checkmark outline | Checkmark outline | Checkmark outline | Checkmark outline | Checkmark outline | Checkmark outline |
| Mwa3edk (Odeh et al., 2019) | Checkmark outline | Checkmark outline | Close outline | Checkmark outline | Checkmark outline | Checkmark outline | Close outline |
| BOOKAZOR (Anish Kumar S, 2019) | Checkmark outline | Close outline | Close outline | Close outline | Checkmark outline | Checkmark outline | Close outline |
| MeetUp (Bhangale et al., 2021) | Checkmark outline | Checkmark outline | Checkmark outline | Checkmark outline | Close outline | Checkmark outline | Close outline |
| MADAS (Usharani et al., 2021) | Checkmark outline | Checkmark outline | Close outline | Checkmark outline | Close outline | Checkmark outline | Checkmark outline |

In terms of designs related to the proposed system, mobile applications which are generally used with great popularity are evaluated.

#### WhatsApp

The WhatsApp application is a popular social media messaging application. The reason for review of this application is a possible adoption of messaging, video and voice conversation style for the virtual consultation function.

#### Any.do

Any.do is a popular calendar to-do list scheduler with over 10 million downloads in the Google play store. The application requires user to be authenticated either through google, Facebook or email to access the application functions. The application provides user with Calendar to organize their task, which could be an appointment.

The application also provides option to synchronize with phone calendar. However, this application require manual addition of tasks to be carried out. The application is also a paid application which require monthly or yearly subscription.

Table

Description automatically generated with low confidence

Figure 2.1; And.do To do List Interface

#### Appointment Planner

The appointment planner is a useful application which is not as popular as the previously reviewed applications. The application do not require user authentication to use the application functions. The application provides user with predefined time slots for everyday to organize their appointments.

Calendar

Description automatically generated A picture containing table

Description automatically generated

Figure 2.2:Appointment Planner Application Interface

# ANALYSIS AND DESIGN

### Overview

In this chapter, an outline of the application's specifications that will be implemented is provided. In order to get some ideas, we looked at several other applications that were similar. Following the completion of the analysis, a low-fidelity prototype was developed to illustrate the initial concept for the application's final form. After that, a high-fidelity prototype was constructed, and it was put through user testing as well as Nielsen's heuristics analysis. At the end of the chapter, a quick description of the various tools and services that are utilized in the process of putting the application into action is provided.

### Considered Application

DocHub is an application focus on helping doctors and patients with activities carried out during interaction in the hospital. There are two main users of this application, Doctors and Patients. The application also has a system administration who manages the content of the application. Since the application is designed as an in-house software, Doctor information are directly inserted into the system by the admin after employment. However, Patients and register on the application anytime. Patients should be able to check Doctors availability based on predefined appointment slots by the hospital administrator. Each doctor is displayed with information about their profession like profile picture, specialization, years of experience and more.

Added value to this standard information is the feature to allow patient consult with doctors without going to the hospital. With the appointment slot allocated to a patient, the patient can decide to perform consultation virtually, either through video chat, voice chat or text. Documents om current state of patient and prescriptions can be shared with this medium.

In addition, patient can rate doctor performance in the review section where other patients are able to read. Only patients who have previously consulted with the doctor will be given access to make reviews about the Doctors service. The purpose of this is to avoid misleading reviews about doctors.

In conclusion, DocHub is an application focused on one domain, simplifying Doctor-Patient interaction in the hospital. With the added feature of virtual consultation, Patients do not need to be present on hospital ground except the absolutely have to. It should not be difficult to utilize the application, and the user interface should be clear.

### Use Case

From the specification stated above, the use cases and use case scenarios were created. The use case diagram is shown in Figure 3.1. It shows every use case from the Doctor and Patient perspective. Technically, there is a role of the system administrator who manages the content of the application, but it is skipped due to lack of importance from the application perspective.

As shown in the use case diagram in Figure 3.1, the application has several use cases which includes:

* Use Case 1: Login
* Use Case 2: Register
* Use Case 3: View Doctors
* Use Case 4: Make Appointment
* Use Case 5: Write Reviews
* Use Case 6: Upload Prescription
* Use Case 7: Virtual Consultation
* Use Case 8: Update, Edit, Cancel Appointment
* Use Case 9: Logout

Diagram

Description automatically generated

Figure 3.1: DocHub Use Case Diagram

#### Use Case 1: Login

The user can login into the application, this function works similarly for both Doctors and Patients.

**Pre-Condition:** N/A

**Basic Flow**

1. A user launches the DocHub Application and lands on the login page.
2. The application check for previous login session and if it doesn’t exist stay on the page and if does redirect to the Application user home screen.
3. Users enter Username and Password
4. System validates user credentials and redirect user to home screen.

**Extension**

4a\_1. Application notifies user of invalid username or password or invalid account

4a\_2. Application clears login inputs and return to initial state.

#### Use Case 2: Register

The user can register on the application, this function is specifically for only the Patients.

**Pre-Condition:** The user must be on the registration screen.

**Basic Flow**

1. The user input credential which includes email and password
2. System validates credentials and redirect user to login screen.

**Extension**

2a\_1. Application notifies user of exiting user email address.

#### Use Case 3: View Doctors

The user can display all the doctors in the hospital in a list view. This function is specifically for Patients.

**Pre-Condition:** User must be on the doctor list screen

**Basic Flow**

1. Doctor list is shown in form of a card in list view.
2. User can pull down to refresh list.
3. User can tap on the doctor card and is redirected to the detail doctor view.

#### Use Case 4: Make Appointment

The user can make appointment with any Doctor. This function is specifically for Patients.

**Pre-Condition:** User must be on the Doctor profile.

**Basic Flow**

1. From the Doctor page, make appointment with doctor
2. Select a date to show available time slots for the day
3. Select a suitable time slot and confirm appointment.

**Extension**

2a\_1. System notifies user of no time slot available for the date.

#### Use Case 5: Write Reviews

The user can write reviews on doctors that have previously consulted with. This function is specifically for patients.

**Pre-Condition:** User must be on my doctor list view page

**Basic Flow**

1. From my doctor list view page, user select doctor card.
2. Select write review.
3. Write review and submit.

#### Use Case 6: Upload Prescription

The user can write reviews on doctors that have previously consulted with. This function is specifically for doctors.

**Pre-Condition:** User must be on my patients list view page.

**Basic Flow**

1. From my doctor list view page, user select patient card.
2. Select upload prescription.
3. Select image/document from mobile device and submit.

#### Use Case 7: Virtual Consultation

The Patient can start virtual consultation with Doctor if they have a valid appointment with the Doctor

**Pre-Condition:** User must be on my doctor list view page

**Basic Flow**

1. Users select the start consultation meeting with Doctor
2. Select consultation option (Voice Chat, Video Chat, Text Chat)

**Extension**

1a\_1. System notifies user of invalid appointment time.

1a\_2. Ask user to make appointment to have virtual consultation.

#### Use Case 8: Update, Edit, Cancel Appointment

The user can change appointment and cancel appointments with Doctors. This function is specifically for Patients.

**Pre-Condition:** User must be on my appointments list view page.

**Basic Flow**

1. Users select the from list of upcoming appointment.
2. Update dates or slots of appointment

**Extension**

1a\_1. System notifies user of no upcoming appointments.

2a\_1. System notifies user of no time slot available for the date.

#### Use Case 9: Logout

The user can logout from the application. This function works similarly for both Patients and Doctor

**Pre-Condition:** User must be previously logged into the application.

**Basic Flow**

1. Users select the logout option.
2. User current session is cancelled and redirected to the login page.

### Application Prototype

The development of a working prototype is an essential part of the process of creating a software product. Different design ideas can be presented with the assistance of prototypes, which can also be easily tested, assessed, and modified. Although there are many different prototyping processes, the end goal is always the same: to create a visual concept of the finished product. The use of prototypes is not only beneficial from an aesthetic standpoint; they are also an integral part of user experience research and may be used to determine in advance which aspects of the user interface should be modified.

There is no single, agreed-upon concept of what prototypes ought to look like or how they ought to be developed. The prototype can be created in a variety of forms, from a straightforward sketch on paper to an intricate application that is pixel perfect. Throughout the entirety of the creation process, many prototypes may be developed for testing purposes.

In the first stages, a prototype with low fidelity, also known as Lo-Fi, is typically constructed. The application can be assessed and user-tested with Lo-Fi to determine whether or not the desired design concept is useful and understandable for end users. After the Low Fidelity (Lo-Fi) prototype is completed, the High Fidelity (Hi-Fi) prototype is developed. Hi-Fi should behave as a fully functional application on the target platform and emerge from Lo-Fi, which is the starting point. Once more with the help of Hi-Fi, the application will be tested and assessed with users.

A Lo-Fi prototype is a means of translating high-level concepts into artefacts that are both tangible and capable of being tested. The ability to evaluate and test the functionality of the product before it even has a visual look is the most important functionality that Lo-Fi prototypes offer. One of the benefits of Lo-Fi is that it is an inexpensive and quick method of proposing prototypes. Lo-Fi, on the other hand, is not complicated and cannot provide more advanced forms of interactivity because of this. In the early phases of the creation process, Lo-Fi should be utilized to facilitate the rapid creation of a prototype and the gathering of feedback from users.

The Hi-Fi prototype is built once the Lo-Fi prototype is finished. This application's actual developed version is modelled after its appearance and behavior as closely as feasible in this prototype. Hi-Fi should be developed on the platform that will be used, and it should function in the same way as the finished product. The interaction with the user interface should be made to be more sophisticated, and the input from user testing should be improved. In comparison to a low-fidelity prototype, a prototype that looks like an actual application encourages more natural user behavior and enables the user to provide feedback that is both more specific and more relevant.

Finally, because Lo-Fi prototypes are only tested internally with a select group of users, they can undergo rapid iterations. Since the Hi-Fi is more expensive to make, it should be made and tested only after the Lo-Fi prototype has been accepted.

#### DocHub Prototype

A task list was made after the specification was written and finalized. The list of tasks is written from the user's point of view, with each item detailing a specific step that must be taken. As the list of tasks grows in length, it is typically converted into a task graph. The task graph is not defined strictly, but it should include all tasks and all screens in the application. The task graph will facilitate managing the implementation of the prototype.

DocHub’s task graph is listed in Figure 3.2, the green rectangles represent screens, and the blue rectangles represents the task user can carry out on the screen. The entry point of the application is represented with a red bordered rectangle.

Graphical user interface, diagram, application

Description automatically generated

Figure 3.2: DocHub Task Graph

1. **Low Fidelity Prototype**

Based on the definition of the task graph, the low fidelity prototype can be made. Despite the application been designed for multiple platforms, we focus the base design on Android and its Material layout for example the *appBar* with the application title and *Menu, Buttons* and *Tabs.*

In the first instance all rough prototype of the expected screens from the task graph are drawn on paper. This will give the idea of what the app screen look like and considered layouts. After that, the Miro prototyping tool is used to mimic the pencil sketch pf the layout.

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface

Description automatically generated with low confidence (a) (b) (c)

Diagram

Description automatically generatedGraphical user interface

Description automatically generatedGraphical user interface, text

Description automatically generated

(d) (e) (f)

Graphical user interface, application

Description automatically generated

(g)

Figure 3.3: Low Fidelity Prototype (a)Registration page (b)Login page (c)Doctor detail page (d)Home page (e)Doctor list (f)Create appointment page

(g)Appointment list

1. **High Fidelity Prototype**

The design implementation of the high fidelity protype including the testing process will be defined in the next chapter during the implementation process. However, the requirement needed from the High-Fidelity prototype is defined using the Nielsen Heuristic. Therefore, the evaluation of the system interface will be based on these Heuristics.

#### Nielsen Heuristic

According to Jakob Neilsen, 1994, user interface design should follow 10 general principles to give user good usability experience. These principles are called heuristics which are a set of rules which the user interface should be judged upon by a small group of evaluators. Each of the rules are described briefly based on the article from (Jakob Neilsen, 1994),

1. **Visibility of system status:** Users of the system should constantly be kept up to date about what is happening by the system providing relevant feedback within a reasonable amount of time.
2. **Match between system and the real world:** Instead of using terminology that are exclusive to the system, the interface for the system should make use of words, phrases, and ideas that are already recognizable to the user. Observe the norms of the actual world and arrange the material such that it appears in a logical and natural sequence.
3. **User control and freedom:** Users frequently choose system functions by accident, and they will want a clearly defined "emergency escape" to leave the undesirable state without having to go through a long conversation. This "emergency exit" should not require the user to go through any more steps. It should be possible to undo and redo.
4. **Consistency and standards:** It is not appropriate for users to be required to consider if various words, circumstances, or activities indicate the same thing.
5. **Error prevention:** A well-thought-out design that eliminates the need for error messages altogether by preventing errors from happening in the first place is preferable in every way. Either do away with criteria that are prone to mistake completely or check for them and give users the opportunity to affirm their decision before they carry out the action.
6. **Recognition rather than recall:** Making items, actions, and alternatives accessible may help reduce the amount of mental strain placed on the user. It shouldn't be necessary for the user to have to recall information from one stage of the conversation to the next. Whenever it is relevant, the instructions for using the system need to be in plain sight or within an easy reach of the user.
7. **Flexible and efficiency of use:** The expert user's contact with the system may often be sped up using accelerators, which are hidden from view of the beginner user. As a result, the system can accommodate the needs of both novice and experienced users. Make it possible for users to customize frequently used activities.
8. **Aesthetic and minimalist design:** Information that is not relevant or that is only sometimes required should not be included in dialogues. Every additional piece of information that is shared in a conversation acts as competition for the important pieces of information and reduces the relative visibility of those pieces.
9. **Help user recognize, diagnose, and recover from errors:** Error messages have to be written in clear English (no codes should be used), they ought to accurately describe the issue, and they ought to provide constructive solutions.
10. **Help and documentation:** Even while it is preferable if the system can be utilized without the need for documentation, it is possible that such assistance and documentation will still be required. Any such material should be simple to search for, concentrate on the job that the user is attempting, specify clear actions that need to be carried out, and not be too extensive.

### Back-end Analysis

In this section, the technical analysis of the technology for the back-end services is discussed. The DocHub API service will be designed specifically to connect the storage database and the mobile client. The means of communication is shown in Figure 3.4.

Graphical user interface

Description automatically generated with medium confidence

Figure 3.4: System Architecture

1. HTTP request is made by the mobile client to the DocHub API.
2. The request is processed and if appropriate query to the storage is issued, the required data is obtained.
3. The result is return to the mobile client through the DocHub response.

Technology to be used to build API should fulfil the following requirements:

* The access to API must be secure either from storage or mobile client.
* The service provided by the backend should be scalable when needed.
* The maintenance cost of API should be minimal.

There are many technologies which can be used to achieve the requirement. One of the methods of implementing an API is implementing from scratch with technologies like .NET Core, RESTful API etc. In this project, The RESTful API implemented using PHP framework. The API uses HTTP request to GET, POST and DETELE data accordingly. The API is deployed to a service hosting provider. Other predefined APIs and resources will be introduced in the implementation state of the application design as required.

#### Storage Analysis

Information from the mobile application is required to be stored in the database. The database management system which is chosen to be used for this application is the MySQL. In order to design a database, we need to find all the entities in the system and their relationships. This can be derived from the system business rule.

To complete the design of the database we need to show the business rule of the system. The business rule is basically a statement that is used to impose constraints on specific aspect of the database. In this project the entities in the system are derived from the system requirement. These are Patients, Doctor, and Appointments. The business rules for this entity are as follows:

* Each Patient may make appointment with one or many Doctor. Each Doctor may have one or many Patients.
* Each Appointment is made by one Patient for only one Doctor.
* Each medication is issued by a doctor and for a patient. A patient may have many medication and a doctor may issue many medication.

#### Entity Relationship Diagram

Graphical user interface

Description automatically generated

Figure 3.5: Entity Relationship Diagram

#### Dataflow

This section of the report describes the context diagram of the system's architecture with an emphasis on the information flow into and out of the application's interface to the database. Based on the functions present in the system the context diagram as shown in Figure 3.6 is the simple representation of the flow of data from the user to the system and vice versa.

A picture containing diagram

Description automatically generated

Figure 3.6: Context Diagram

However, to show detailed representation of data based on each function, the level 1 data flow diagram is created show in Figure 3.7.

Diagram

Description automatically generated

Figure 3.7: Data Flow Diagram

#### Process Flow Chart

Based on the designed dataflow diagram, there are four processes which facilitate the entire DocHub application. The processes are:

* **Authentication**

Authentication process is used by both user types (patients and doctors). The authentication process facilitates user login process and user registration process. The Figure 3.8 shows the flow chart of user authentication process for login.

Diagram

Description automatically generated with medium confidence

Figure 3.8: Login Flow Chart

The registration process is only available to the patients since they are the only user allowed to register on the system based on the requirement. The flow chart of the registration process is shown in Figure 3.9.

Diagram

Description automatically generated

Figure 3.9: Registration Flow Chart

* **Get Doctor**

The get doctor function allows user to select a specific category of doctors or department. The default result will show the entire list of doctors in the hospital. The flow chart for the process is shown in Figure 3.10.

A picture containing diagram

Description automatically generated

Figure 3.10: Get Doctor

* **Appointment**

The appointment process has two parts, first is to make the appointment which is available for one user which is the patient. The other part is updating the appointment which available to both users patient and doctors. The flow chart of the process is shown in Figure 3.11.

Diagram

Description automatically generated

Figure 3.11: Appointment Flow Chart

* **Medication**

The medication process is used by both users but are used for different purpose. The patient only have access to view their medication while the doctor can create medications. The Figure 3.12 shows the flow chart for the process.

Diagram

Description automatically generated

Figure 3.12: Medication Flow Chart

# IMPLEMENTATION

### Overview

A prototype is an interactive mock-up of a product that shows how it will function before it is built. It can provide an overview of a mobile application's functionality, user procedure, and flow. Users may also move from page to page and use all of the features in the prototype. The pages in the prototype will be described in this section of the report based on the low fidelity design from the previous chapter. In the implementation of the prototype, we have the frontend which is the high-fidelity prototype. In addition, the implementation process of the backend which includes the database and APIs is described in this chapter. The high-fidelity descriptions shows how the interface will look like which will be shown using the image. The backend is the connection to the database which is implemented using a PHP, and MySQL.

### High-Fidelity Prototype

The High-Fidelity is designed based on the Low-Fidelity prototype designed in the previous chapter. Additional screens are generated based on the objective and functions of the application.

#### Authentication Screen

From the application lunch, the mobile application is expected to redirect to the authentication screen. The authentication screen consists of signUp, signIn and forget-password screen. The image is shown in Figure 4.1. The required information on the signIn screen is the user email and password. On the signUp screen, the input required are email, name, password and mobile contact. The forget-password screen only require the user registered email.

Graphical user interface, application

Description automatically generated Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

1. (b) (c)

Figure 4.1: Authentication Screen (a) SignIn;(b) SignUp; (c) Forget-Password

#### Home Screen

The home screen for the user is design to provide specific functions for the different users(patient and doctors). The patient home screen is shown in Figure 4.2 and the doctor home screen is shown in Figure 4.3. The home screen for both users contains a menu button which produces dropdown menu for the user profile and logout option. Specifically for the patient home screen there are three menu option, the book appointment now button, my doctor menu and my medication menu.In addition there is a button option for patients to m

Graphical user interface, text, application, chat or text message

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated

1. (b)

Figure 4.2:Patient Home Screen (a) Default;(b) With dropdown menu

Graphical user interface, application

Description automatically generated Graphical user interface, application

Description automatically generated

(a) (b)

Figure 4.3: Doctor Home Screen (a) Default;(b) With dropdown menu

#### Account Screen

The Account screen is available for both types of user to show information about their profile. The design is similar; therefore a sample of the patient account is shown in Figure 4.4. The profile screen contains the following elements: row menu which show the highlight of the profile information, my feedback menu, help and support menu, about dochub app and lastly the terms and condition menu. The account highlight screen lets the user access the profile editing screen. The rest of the menu links to their respective name screen.

Graphical user interface, text, application, Teams

Description automatically generated Graphical user interface, text, application, email

Description automatically generated Graphical user interface, text, application

Description automatically generated

(a) (b) (c)

Figure 4.4: Account Screen (a) Default;(b) Edit Profile; (c) My Feedback

#### Appointment Screen

The appointment screen consists of different screen options depending on the function the user is carrying out. The appointment screens shown in Figure 4.5 consist of book appointment screen, book appointment screen with mode selection drop down, the appointment list screen, the appointment detail screen and the cancel appointment option screen. The book appointment screen allows the user select time slots and mode of appointment which they desire. The mode of appointment has three option based on the system requirement (physical, virtual voice call, virtual video call) as shown in the book appointment with mode dropdown screen. From the appointment list screen, user can access the appointment detail screen by selecting a specific upcoming appointment. Past appointment on the appointment list screen cannot be selected. The user can either cancel or reschedule appointment using the cancel or reschedule menus from the appointment detail screen.

Graphical user interface, text, application, chat or text message

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated Graphical user interface, application

Description automatically generated

(a) (b) (c)

Graphical user interface, application

Description automatically generated Graphical user interface, application, Teams

Description automatically generated

(d) (e)

Figure 4.5: Appointment Screen (a) Book Appointment;(b) Book Appointment with Mode options; (c) Appointment List; (d) Appointment Detail; (e) Cancel Appointment Menu

#### Doctor Screen

The doctors screen can be access when a patient tries to book an appointment. The list of doctors in the hospital is presented to the patient. Patient have option of filtering the list based on doctor departments. From the doctor list screen patient can select a doctor view details about the doctors. The doctor details screen provides information about the doctor as well feedback from previous patients regarding doctor service.

Graphical user interface, application

Description automatically generated Graphical user interface, application

Description automatically generated

(a) (b)

Graphical user interface, application, website

Description automatically generated Graphical user interface, application, website

Description automatically generated Graphical user interface, text, application

Description automatically generated

(c) (d) (e)

Figure 4.6: Doctor Screen (a) Doctor List;(b) Doctor List with Filter; (c) About Doctor; (d) Doctor Feedbacks; (e) Give Feedback

#### Patient Screen

The doctors can view a list of their patient and access the patient profile page to see information about the patient. The patient screen is shown in Figure 4.7.

Graphical user interface, application, website

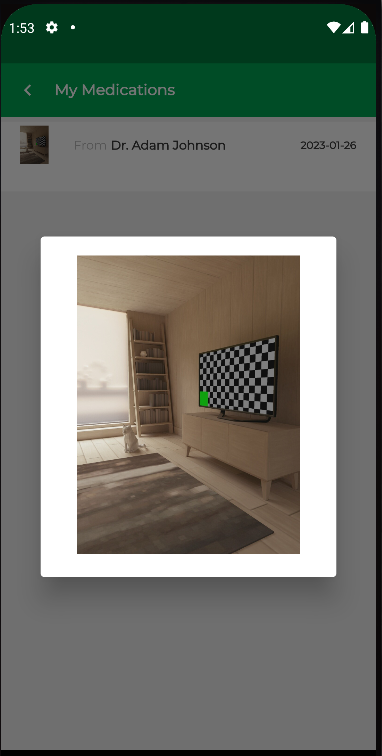
Description automatically generated

Figure 4.7: Patient Screen

#### Medication Screen

The patients can view list medications provided by the doctor from the My Medication on the patient home screen. The medication screen consists of a list of medication and when a medication is selected, the medication pop-up on the screen for easy viewing. The medication screens is shown in Figure 4.8.

Shape

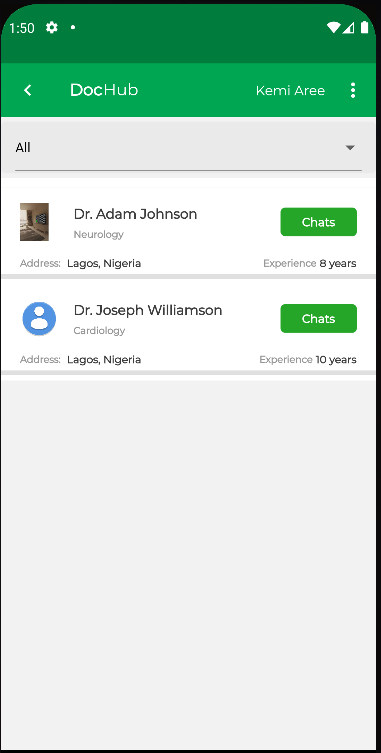
Description automatically generated with low confidence 

(a) (b)

Figure 4.8: Medication Screen (a) Medication List;(b) Pop-Up Medication

#### Virtual Consultation Screen

The virtual consultation screen can be access from the My Patient or My Doctors menu from the Patient Home Screen or Doctor Home Screen respectively. The Figure 4.9 shows the My Patient and My Doctor Home Screen. The chat button from each patient or doctor takes the user to the virtual consultation screen as shown in Figure 4.10.

 Graphical user interface, application, Word

Description automatically generated

1. (b)

Figure 4.9: (a) My Doctor List; (b) My Patient List

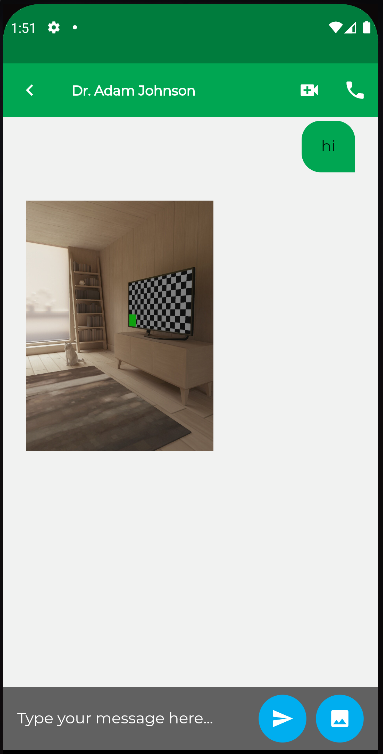


Figure 4.10: Virtual Consultation Room

### Backend Implementation

The backend of the system is implemented using PHP API. The API connects the frontend display to the SQL database information. In addtion to the defined API, other predefined package API are utilized in this Application. The following APIs are used in the implementation of the mobile application:

* Firebase Authentication
* Firebase Storage
* Firebase Firestore Database
* ZegoUIKitPrebuiltCall

All the Firebase API require a firebase account and connecting the application to the firebase environment. The ZegoUIKitPrebuiltCall is a paid service which comes with 10,000mins of free access trial. Apart from the predefined packages utilized, an API is implemented to give access to the SQL database which is used to store all the information used in the mobile application.

#### Firebase Authentication

The firebase authentication API allows the application to authenticate user using email and password. During registration the user credentials are saved in the firebase authentication service. This credentials can be used in the future to access their account. The function utilized in the API are:

* signInWithEmailAndPassword(input: email, password)
* signUpWithEmailAndPassword(input: email, password)
* sendPasswordResetEmail(input: email)
* signOut

ref: *project/app/lib/Services/auth.dart*

#### Firebase Storage

The Firebase storage API is used to store user medications issued by the doctor during virtual consultations. The API call require initialzing the storage.

ref: *project/app/lib/Screens/Doctor/chatPatient.dart:305*

#### Firebase Firestore Database

This Firestore database is used to save conversations between patients and their doctors during virtual consultation. The API requires the chatID which is created as the patient and doctor email address. The following fucntions are created to used the Firebase Firestore API.

* sendConversations(input: chatRoomID, messageMap)
* getConversations(input: chatRoomID)

ref: *project/app/lib/Services/firebasedatabase.dart*

#### ZegoUIKitPrebuiltCall

The Zego UIKit requires a Zego account which give user access to an appID and appSignIn key which is used to initialize the API.

ref: *project/app/lib/Patient/voiceDoctor.dart*, *project/lib/Patient/videoDoctor.dart, project/app/lib/Doctor/chatPatient.dart, project/lib/Doctor/chatPatient.dart.*

#### Defined API

The PHP defined API consist of php file which give sepcific access to different parts of the database. The total numbe of API defined for the mobile Appilcation is 25. The API calls can be found in the API ref defined below.

ref: *project/app/lib/Services/server\_handler.dart*

The definition of each API is shown in Table 4.1 stating the name, usage of the API, the expected input and output.

Table 4.1: API Definition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | API name | Usage | Input | Output |
| 1 | auth/user\_login | Get information of User after successful login | user\_email | User(Doctor / Patient) |
| 2 | auth/user\_registration | Insert user information after successful registration | User information | True / False |
| 3 | auth/update\_user\_token | Update user token after signin or signout | user\_email | True/ False |
| 4 | doctor/doctor\_profile | Get doctor information | doctor\_id | User(Doctor) |
| 5 | doctor/doctor\_update | Update doctor information | Doctor information | True/False |
| 6 | doctor/doctors | Get all doctors in the system | deparment | List(Doctor) |
| 7 | doctor/mydoctor | Get a patient doctor list | patient\_id, department | List(Doctor) |
| 8 | patient/mypatient | Get a doctor patient list | doctor\_id | List(Patient) |
| 9 | patient/patient\_profile | Get patient information | patient\_id | User(patient) |
| 10 | patient/patient\_update | Update patient information | Patient information | True/False |
| 11 | gen/appointment | Get doctor free slot for a particular date | doctor\_id, date | Slot list |
| 13 | appointment/add | Add an appointemnt information | doctor\_id, patient\_id, date, slot, mode | True/False |
| 14 | appointment/checkapp | Check if patient have a valid appointment at the time | doctor\_id, patient\_id, date, slot, mode | True/False |
| 15 | appointment/checkappfeedback | Check is patient has previously met with a doctor to allow access to give feedback | doctor\_id, patient\_id | True/False |
| 16 | appointment/delete | Delete an appointment informeation | appointment\_id | True/False |
| 17 | appointment/doctorapp | Get all appointment of a particular doctor | doctor\_id | List(Appointment) |
| 18 | appointment/patientapp | Get all appointmet of a particulat patient | patient\_id | List(Appointment) |
| 19 | appointment/update | Update appointment | Appointment information | True/False |
| 20 | appointment/updateappstatus | Update status of appointment to show if Doctor is ready for consultation | patient\_id,  doctor\_id,  mode,  status | True/False |
| 21 | feedback/add | Add feedback information | doctor\_id, patient\_id, rating, comment(not required) | True/False |
| 22 | feedback/doctorfeedback | Get all feedback of a particular doctor | doctor\_id | List(Feedback) |
| 23 | feedback/feedback\_status | Check is Patient can give feedback for a Doctor | patient\_id, doctor\_id |  |
| 24 | feedback/patientfeedback | Get all feedback of a particular patient | patient\_id | List(Feedback) |
| 25 | medication/add | Add medication information | doctor\_id, patient\_id, file | True/False |
| 26 | medication/patientmediaction | Get all medication of a particular patient | patient\_id | List(Medication) |

# TESTING AND EVALUATION

### Overview

This chapter includes the testing process carried out and expected result and output from the dochub mobile application. The aim of the testing and evaluation is to verify the mobile application fulfil all the objective stated in the first chapter.

### Testing Procedure

Functional testing is a process offering enormous advantages in software testing for the development process. If done properly, it improves communication between analysts, developers and testers. By examining the function tests whether successful or not, the entire project is objectively visible to the management. Finally, development speed increases as well-known requirements lead to less reprocessing. These tests also lead to a more clearly defined system structure. Unit testing is used in this project development.

Non-functional tests are as crucial as functional tests. It is because teams must run a mixture of various test types; the team must make both of these functional and non-functional tests more usable and more reliable. Unfortunately, application deadlines can often be rushed. If non-functional tests are not taken into account, users can experience bad performance and UX defects and cause brand damage. Worst of all, user influx applications could crash. Deficiencies inaccessibility can lead to fines of compliance a and their security could be in jeopardy. In chapter 3, we described the criteria to which we will evaluate the non-functional test using Neilsen Heuristics.

#### Unit Testing

Unit testing ensures every part delivers the required output of code developed in a system. In this unit testing phase, developers examine only the interface and the component specification. It provides code development documentation since every unit of the code has been appropriately tested independently before moving to a unit. Programmers often use the sample input and monitor its corresponding outputs. The tables 5.1 – 5.5 shows the unit testing carried out during the development of the project. Based on the testing modules all the required objectives were fulfilled by the application.

Table 5.1: Unit Testing Module for Authentication Function

|  |  |  |
| --- | --- | --- |
| **Authentication** | | |
|  | Result | Remarks |
| Register Account with valid email and password | Successful | Wrong/Empty space remain on the screen with error message displayed |
| Login as patient and doctor with valid email and password | Successful | Wrong/Empty space remain on the screen with error message displayed |
| Forget password functionality | Successful | Only valid for registered customers. |

Table 5.2: Unit Testing Module for Appointment Function

|  |  |  |
| --- | --- | --- |
| **Appointment** | | |
|  | Result | Remarks |
| View all doctors present and filter by department | Successful | - |
| Select a doctor to view doctor profile (Feedback and About) | Successful |  |
| Book an appointment with a doctor | Successful | - |
| Cancel/Reschedule and appointment | Successful | - |

Table 5.3: Unit Testing Module for User Profile Function

|  |  |  |
| --- | --- | --- |
| **User Profile** | | |
|  | Result | Remarks |
| View profile information (Doctor /Patient) | Successful | - |
| Update profile information with or without profile image | Successful | - |
| Logout of application | Successful | - |

Table 5.4: Unit Testing Module for Virtual Consultation Function

|  |  |  |
| --- | --- | --- |
| **Virtual Consultation** | | |
|  | Result | Remarks |
| Chat between doctor and patient | Successful | - |
| Send and receive medication prescription through chat | Successful | - |
| Virtual Video Consultation | Successful | Requires valid appointment time and slot |
| Virtual Voice Consultation | Successful | Requires valid appointment time and slot |

Table 5.5: Unit Testing Module for Feedback Function

|  |  |  |
| --- | --- | --- |
| **Feedback** | | |
|  | Result | Remarks |
| Provide Feedback for a doctor | Successful | Only available for patient with previous appointments with the doctor |

#### Usability Testing

Usability testing is a practice to test how simple a design seems to be with a group of real users. It usually involves monitoring users as they try to finish a given task and can also be completed for different designs. It can also be performed multiple times, from early development to the release of the product. It is to find design flaws that might overlook through usability testing. By looking at how test users behave while trying to perform tasks, easy to get vital insights into how well the design or mobile app works. These observations can be used to make improvements. The usability test for this project is carried out using a form asking users their experience based on each task given.

There was a total of 8 users used to test the mobile application. The tasks are divided based on the functional requirements. The response is recorded based on if user is able to complete the task successfully and if they achieve positive or negative result. Remark is given against user who were unable to complete the task on their own.

Table 5.6: Usability Testing Module for Authentication Function

|  |  |  |  |
| --- | --- | --- | --- |
| **Authentication Task** | Completed | Not Completed | Remark |
| Register an account | 7 | 1 | The user requires a password of at least 8 characters but there was no message to alert the user to this, so the user was stuck on the registration page. |
| Try to register an account with an empty field | 8 | 0 |  |
| Login with Valid Credentials | 8 | 0 |  |
| Login with invalid credentials (email / password) | 8 | 0 |  |
| Use the forget password functionality | 8 | 0 |  |
| Login with new password after password changes | 8 | 0 |  |

Table 5.7: Usability Testing Module for Appointment Function

|  |  |  |  |
| --- | --- | --- | --- |
| **Appointment Task** | Completed | Not Completed | Remark |
| Book an appointment with any Doctor using the book appointment now button on the home screen | 8 | 0 |  |
| Filter Doctors by Department | 8 | 0 |  |
| Go to a Doctor profile and book an appointment from there | 8 | 0 |  |

Table 5.8: Usability Testing Module for User Profile Function

|  |  |  |  |
| --- | --- | --- | --- |
| Profile Task | Completed | Not Completed | Remark |
| View your profile from the home screen menu | 7 | 1 | Required assistance finding the profile menu option on the user home screen |
| Update your profile image | 8 | 0 |  |
| Logout | 8 | 0 |  |

Table 5.9: Usability Testing Module for Virtual Consultation Function

|  |  |  |  |
| --- | --- | --- | --- |
| Virtual Consultation | Completed | Not Completed | Remark |
| Chat with a doctor | 8 | 0 |  |
| Voice chat with a doctor | 5 | 3 | Unable to complete task due to appointment time constraints |
| Video chat with a doctor | 5 | 3 | Unable to complete task due to appointment time constraints |

Table 5.10: Usability Testing Module for Feedback Function

|  |  |  |  |
| --- | --- | --- | --- |
| Feedback Task | Completed | Not Completed | Remark |
| View Doctor feedback from doctor profile | 8 | 0 |  |
| Provide feedback | 5 | 3 | Task could not be completed because user have no previous appointment with doctor |

#### Neilsen Heuristic

The application interface is tested and evaluated based on the Neilsen Heuristics has previous described in chapter three which is one of the evaluation metrics for the mobile application. The users testing the app provide feedbacks based on the different criteria of the app evaluation. The following questions are the list of question given to the user with the response options of Strongly Agree, Somewhat Agree, Neither Agree or Disagree, Somewhat Disagree and Strongly Disagree.

The Figure 5.1 shows the response of the 8 users based on each of measurement criteria question. Based on the result we can conclude that the application performed well based on the heuristic evaluation.

# CONCLUSION

### Concluding Remarks

This project has shown how significant the problem of getting attention from medical expert due to the limited number of experts available. The proposed solution of providing a mobile application to provide a virtual environment where Patients can consult with medical professional is shown to be viable by the study of existing platforms in different countries. The proposed solution will mitigate overcrowding in hospital and also provide quick access to the doctors through virtual consultation. In addition hospitals will be able to evaluate doctor performance using the feedback from patients.

The completion of this system is guided by the SDLC using the Prototyping approach, this strategic method is enable and guide us in the implementation of the system. The major limitation that can be faced by this system is the acceptance of patient to have virtual consultation with doctor because of its novelty. The process will need some getting use to and might be rejected by the patients.

### Future Work

In future improvement of the DocHub application, additional functions can be integrated into the application. There are several other activities that are carried in the hospital that require appointment booking for example medical test from the labs. The labs can be an added function in the future app. In addition, pharmacies can be given direct access to patient medication to provide delivery instead of having to pick them up when issued by the doctor.

REFERENCES

Colley, A., Rantakari, J., & Häkkilä, J. (2017). *Dual Sided Tablet Supporting Doctor-Patient Interaction Demonstration CSCW’15 Companion*. https://doi.org/10.1145/2685553.2702672

Vize, R. (2017). How can health services keep pace with the rapid growth of cities? | Richard Vize. The Guardian. https://www.theguardian.com/sustainable-business/2017/feb/24/how-can-health-services-keep-pace-with-the-rapid-growth-of-cities

Chukwuma Muanya and Nkechi Onyedika-Ugoeze (2022, March 16) Only four doctors available to every 10,000 Nigerians. https://guardian.ng/news/only-four-doctors-available-to-every-10000-nigerians/

Nike Adebowale-Tambe (2022, July 31) Again, Nigerian doctors threaten strike over unpaid wages, poor welfare. Premium Times Nigeria Edition. https://www.premiumtimesng.com/news/headlines/546011-again-nigerian-doctors-threaten-strike-over-unpaid-wages-poor-welfare.html

Dennis, A., & Newman, W. (n.d.). *Supporting Doctor-Patient Interaction: Using a Surrogate Application as a Basis for Evaluation*.

Anish Kumar S, A. V. (2019). *BOOKAZOR - an Online Appointment Booking System; BOOKAZOR - an Online Appointment Booking System*.

Bhangale, P., Bhatt, B., Chavda, P., & uskaan Nandu, M. (2021). MeetUp: An Appointment Booking System using Flutter and Django Framework : MeetUp: Meetings Made Easy; MeetUp: An Appointment Booking System using Flutter and Django Framework : MeetUp: Meetings Made Easy. *2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*. https://doi.org/10.1109/ICOSEC51865.2021.9591786

Fernanda Escobar, M., Fernando Henao, J., Prieto, D., Paula Echavarria, M., Carlos Gallego, J., Nasner, D., Marcela Martínez-Ruíz, D., Eduardo Velasco, J., Alarcón, J., Fernanda Escobar, M., Fernando Henao, J., Prieto, D., Paula Echavarria, M., Carlos Gallego, J., Nasner, D., Marcela Martínez-Ruíz, D., Eduardo Velasco, J., Alarcón, J., Fernanda Escobar Vidarte, M., & Fundación Valle del Lili, M. (2021). 19 pandemic at a University Hospital in Colombia. *International Journal of Medical Informatics*. https://doi.org/10.1016/j.ijmedinf.2021.104589

Geissbuhler, A., Ly, O., Lovis, C., & L’haire, J.-F. (n.d.). *Telemedicine in Western Africa: lessons learned from a pilot project in Mali, perspectives and recommendations*.

Morishita, M., Takahashi, O., Yoshii, S., Hayashi, M., Kibune, R., Nakamura, T., Muraoka, K., Tominaga, K., & Awano, S. (2021). Effect of COVID-19 on dental telemedicine in Japan. *Journal of Dental Sciences*, *xxxx*. https://doi.org/10.1016/j.jds.2021.07.028

Odeh, A., Abdelhadi, R., & Odeh, H. (n.d.). *Medical patient appointments management using smart software system in UAE*. ACIT.

Omboni, S. (2020). Telemedicine during the COVID-19 in Italy: A Missed Opportunity? *Telemedicine and E-Health*, *26*(8), 973–975. https://doi.org/10.1089/tmj.2020.0106

Usharani, S., Prithivi, S., Sharmila, S., Manju Bala, P., Ananth Kumar, T., & Rajmohan, R. (2021). Mobile Application for Doctor Appointment Scheduling. *2021 International Conference on System, Computation, Automation and Networking (ICSCAN)*. https://doi.org/10.1109/ICSCAN53069.2021.9526398

APPENDICES

**Appendix A:**

**Appendix B:**