

# **DOCEASY- MOBILE APPLICATION FOR MANAGING DOCTOR APPOINTMENTS**

## **A PROJECT REPORT**

*Submitted by,*

<b>Sagili Mythili</b>	<b>- 20201CCS0078</b>
<b>Ravuri Venkata Koushik</b>	<b>- 20201CCS0150</b>
<b>Chrisilda shamayah Babu Anthony</b>	<b>- 20201CCS0088</b>

*Under the guidance of,  
Dr. Mohana S D*

*in partial fulfilment for the award of the degree of*

**BACHELOR OF TECHNOLOGY  
IN**

**COMPUTER SCIENCE AND ENGINEERING (CYBER SECURITY)**

**At**



**PRESIDENCY UNIVERSITY  
BENGALURU**

**JANUARY 2024**

# PRESIDENCY UNIVERSITY

## SCHOOL OF COMPUTER SCIENCE & ENGINEERING

### CERTIFICATE

This is to certify that the Project report "**DOCEASY- MOBILE APPLICATION FOR MANAGING DOCTOR APPOINTMENTS**" being submitted by **Sagili Mythili, Chrisilda Shamayah Babu Anthony, Ravuri Venkata Koushik** and bearing roll number(s) **20201CC0078, 20201CCS0088, 20201CCS0150** in partial fulfilment of requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering (Cyber Security) is a bonafide work carried out under my supervision.



**Dr. MOHANA S D**  
Assistant Professor  
School of CSE  
Presidency University



**Dr. S.P. ANANDA RAJ**  
Professor & HoD  
School of CSE  
Presidency University



**Dr.C.KALAIARASAN**  
Associate Dean  
School of CSE&IS  
Presidency University



**Dr. SHAKKEERA L**  
Associate Dean  
School of CSE&IS  
Presidency University



**Dr. SAMEERUDDIN KHAN**  
Dean  
School of CSE&IS  
Presidency University

**PRESIDENCY UNIVERSITY**  
**SCHOOL OF COMPUTER SCIENCE & ENGINEERING**  
**DECLARATION**

We hereby declare that the work, which is being presented in the project report entitled "**DOCEASY-MOBILE APPLICATION FOR MANAGING DOCTOR APPOINTMENTS**" in partial fulfilment for the award of Degree of **Bachelor of Technology in Computer Science and Engineering (Cyber Security)**, is a record of our own investigations carried under the guidance of **Dr. Mohana S D, Assistant Professor, School of Computer Science & Engineering, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

Name	Roll No	Signature
Sagili Mythili	20201CCS0078	s. Mythili
Ravuri Venkata Koushik	20201CCS0150	R.V.Koushik's
Chrisilda Shamayah Babu Anthony	20201CCS0088	Chrisilda Shamayah Babu Anthony

## **ABSTRACT**

Mismanagement within a large institution such as a hospital or clinic gives rise to various complications, such as longer wait times, delayed and erratic healthcare, and the need to work overtime for healthcare providers and related staff. This paper discusses all the undertaken measures regarding research, implementation, and development of 'DocEasy' an application developed to eliminate the previously mentioned issues, thus ensuring a smooth and seamless consultation experience for any healthcare institution that incorporates it. Developed on Android Studio and by implementing services provided by Firebase, the application (which can be accessed by physicians and desk clerks) boasts several features, such as reducing wait times by scheduling, the ability to hand over appointments to a substitute physician (in the event that the primary physician is on vacation, or overcome by an emergency) ensuring uninterrupted healthcare, the receptionist will be able select physicians based on the patient's corresponding symptoms, and the provision for doctors to view their patient's profile ahead of the appointment

## **ACKNOWLEDGEMENT**

First of all, we are indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected **Dr. Md. Sameeruddin Khan**, Dean, School of Computer Science Engineering &, Presidency University for getting us permission to undergo the project.

We record our heartfelt gratitude to our beloved Associate Deans **Dr. Kalaiarasan C** and **Dr. Shakkeera L**, School of Computer Science Engineering & Information Science, Presidency University and **Dr. S.P. Anandaraj**, Head of the Department, School of Computer Science Engineering(Cyber Security), Presidency University for rendering timely help for the successful completion of this project.

We are greatly indebted to our guide **Dr.Mohana S D**, School of Computer Science Engineering, Presidency University for his inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the University Project-II Coordinators **Dr. Sanjeev P Kaulgud**, **Dr. Mrutyunjaya MS** and also the department Project Coordinator. **Ms. Manasa C M**.

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

**Sagili Mythili**

**Ravuri Venkata Koushik**

**Chrisilda Shamayah Babu Anthony**

## **LIST OF TABLES**

<b>Sl. No.</b>	<b>Table Name</b>	<b>Table Caption</b>	<b>Page No.</b>
1	Table 9.2.1	Receptionist Signup Activity	33
2	Table 9.2.2	Doctor Signup Activity	34
3	Table 9.2.3	Receptionist Login Activity	35
4	Table 9.2.4	Doctor Login Activity	35
5	Table 9.2.5	New Appointment Activity	36
6	Table 9.2.6	Doctor Applying Leave	37
7	Table 9.2.7	Patient Creation Activity	38

## **LIST OF FIGURES**

<b>Sl. No.</b>	<b>Figure Name</b>	<b>Caption</b>	<b>Page No.</b>
1	Figure 4.1.1	Data-Flow Diagram	23
2	Figure 6.1	Architecture Diagram	27
3	Figure 7.1	Gantt Chart	30
4	Figure 11.1.1	Code Execution 1	44
5	Figure 11.1.2	Code Execution 2	44
6	Figure 11.1.3	Code Execution 3	45
7	Figure 11.1.4	Code Execution 4	45
8	Figure 11.1.5	Code Execution 5	46
9	Figure 11.1.6	Code Execution 6	46
10	Figure 11.1.7	Code Execution 7	47
11	Figure 11.1.8	Code Execution 8	47
12	Figure 11.1.9	Code Execution 9	48
13	Figure 11.1.10	Code Execution 10	48
14	Figure 11.1.11	Code Execution 11	49
15	Figure 11.1.12	Code Execution 12	49
16	Figure 11.2.1	Onboarding Screens (1)	50
17	Figure 11.2.2	Onboarding Screens (2)	50
18	Figure 11.2.3	Receptionist login activity	51
19	Figure 11.2.4	Receptionist logout activity	51
20	Figure 11.2.5	Receptionist Home Page	52
21	Figure 11.2.6	Doctor Home Page	52
22	Figure 11.2.7	All Patients	53
23	Figure 11.2.8	Post Consultation List	53
24	Figure 11.2.9	All Doctors	54
25	Figure 11.2.10	Post Consultation List	55

## **TABLE OF CONTENTS**

<b>CHAPTER NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
	<b>ABSTRACT</b>	<b>i</b>
	<b>ACKNOWLEDGMENT</b>	<b>ii</b>
<b>1.</b>	<b>INTRODUCTION</b>	<b>10</b>
<b>2.</b>	<b>LITERATURE REVIEW</b>	<b>12</b>
<b>3.</b>	<b>RESEARCH GAPS OF EXISTING METHODS</b>	<b>20</b>
<b>4.</b>	<b>PROPOSED METHODOLOGY</b>	<b>21</b>
	<b>4.1 Methodology</b>	<b>21</b>
	<b>4.2Algorithm</b>	<b>24</b>
<b>5.</b>	<b>OBJECTIVES</b>	<b>26</b>
<b>6.</b>	<b>SYSTEM DESIGN &amp; IMPLEMENTATION</b>	<b>27</b>
<b>7.</b>	<b>TIMELINE FOR EXECUTION OF PROJECT</b>	<b>30</b>
<b>8.</b>	<b>OUTCOMES</b>	<b>31</b>
<b>9.</b>	<b>RESULTS AND DISCUSSIONS</b>	<b>32</b>
	<b>9.1 Software and Hardware Requirements</b>	<b>32</b>
	<b>9.2Data set</b>	<b>33</b>
<b>10.</b>	<b>CONCLUSION</b>	<b>40</b>
	<b>REFERENCES</b>	<b>41</b>
	<b>APPENDIX-A: PSUEDOCODE</b>	<b>44</b>
	<b>APPENDIX-B: SCREENSHOTS</b>	<b>50</b>
	<b>APPENDIX-C: ENCLOSURES</b>	<b>55</b>

## CHAPTER-1 INTRODUCTION

Imagine this you enter a building, its walls stark white and its citizens clad in mint green. They rush to tend to those they have taken an oath to protect, to nurture. Suddenly, there is a surge of these care seekers, swarming through the front door. Chaos abounds, and the scrub clad folk struggle to provide their undivided care and attention to them. Dr. Smith sighs as his shift has been extended due to a surge of unexpected appointments. Even though he is a general physician, he is obligated to consult with younger patients due to the increased number of summer illnesses. His vacation plans seem bleak as the other doctors are swamped with work as well. Now, imagine this you enter the same building, its walls stark white and its citizens clad in mint green. They are calm and composed in their offices, tending to those they have taken an oath to protect and to nurture. Patients have already been sorted and are recommended physicians based on their symptoms before they have even set foot into the premises. Dr. Smith must leave early today.

It's the first day of summer and he cannot wait to get to the beach. He doesn't have to worry about his patients as they will continue to receive their healthcare from Dr. Kapur till he returns. The second scenario seems much more appealing than the first, doesn't it? It is all because of 'DocEasy', an application that helps doctors plan their appointments, handle out-patient and in-patient care, and handover patients to other physicians of the same specialization in case of leaves for seamless continuity of care. This application a beacon of digital innovation simplifies the once-arduous task of managing doctor appointments. It sweeps away the antiquated system of pen and paper, replacing it with a streamlined, digital solution.

The result? A dramatic reduction in wait times and a seamless booking experience. Doctors and receptionists alike find it easy to navigate the application's user-friendly interface, making the process of scheduling and managing appointments a breeze. And in the dynamic world of healthcare, where doctors may be unavailable due to emergencies or holidays, the application works like a charm. DocEasy effortlessly allows doctors to hand over their appointments to their colleagues, ensuring uninterrupted care for Patients. In the heart of the hospital, amidst the chaos and volatility of life, this application will stand as a testament to the power of technology to improve healthcare. Welcome to the future, where managing doctor appointments is just a tap away.

## CHAPTER-2 LITERATURE SURVEY

- I. The case study found several issues with traditional scheduling systems such as unpunctual patients, long waiting times, and delayed healthcare. As a result, the scheduling of medical appointments, which is the fundamental basis for most non-urgent health care services, is currently undergoing significant changes to encourage patient participation. Patients have greater accessibility and more options when it comes to how they want their appointments to go when they use online resources as a medium.
- II. A patient appointment and scheduling system with Angular JS for the frontend, Ajax framework for handling client-server request and Sqlite3 and MYSQL for the backend was developed. One important component of healthcare services is scheduling appointments. Numerous applications and methods of solving appointment scheduling issues in healthcare systems have been put forth. An advantage of automating clinical services and operations in different medical facilities is the appointment and scheduling system. Through an automated scheduling system, outpatients can register online, make and cancel appointments, and manage their details. The system is expected to decrease patient wait times and boost physician productivity.
- III. Incompetent scheduling systems, along with extended waiting times, and patient satisfaction during outpatient clinic visits, treatments, and services are addressed in this discussion. The term "waiting time" describes the duration of time a patient must endure while waiting for their appointment or consultation to commence. The duration from the patient's admission to the outpatient clinic to the time the patient received his or her prescription is how it is defined. It can also mean the

whole span of time a patient spends from the time they schedule their appointment until their appointment with a doctor begins.

- IV. The homegrown healthcare startup, 'Practo', and entails its features, success, as well as drawbacks. Using the healthcare app Practo, users may locate medical practices, schedule an immediate appointment, and have online consultations with physicians. In addition, users can ask medical issues and receive responses from the professionals on hand. Users of the service can also schedule private online consultations with any doctor of their choosing. In addition, the app lets users save their appointments and preferred providers in addition to viewing previously scheduled appointments. The application also has a navigation tool that makes it easier for users to locate the practices. Additionally, the software lets users select an appropriate time window in which to schedule an appointment. This application's shortcoming is that it lacks information about the days during which a particular physician can be reached at a specific hospital
  
- V. Prototype Model is used to develop this system. As for the hardware and software used to develop this system, it is MySQL Database and programming language use is PHP and JavaScript. By developing this system, it will reduce the number of calls for an appointment and avoid the morning rush for an urgent appointment. The current standard operating method for scheduling appointments and registering patients in a healthcare setting is laborious and time-consuming. The "Medical Appointment Application" is a web-based mobile application designed to manage the scheduling of appointments for a few medical organizations, irrespective of the services they offer within a specific region. To view

the appointments made by users, the patients, the practices will need to register on the web portal. The ‘Medical Appointment’ application will assist users and patients in scheduling appointments. Additionally, this system is developed using the Prototype Model. PHP and JavaScript were employed as programming languages, while MySQL Database was the hardware and software utilized in the development of this system. By creating this system, fewer calls will be made in search of an appointment, and the morning rush for an urgent appointment would be avoided. Additionally, it might lessen the need to recruit extra receptionists, saving a substantial amount of work. Additionally, it saves the user time by removing the need to haggle over a convenient appointment time with the receptionist. With the help of this technology, they will be able to operate more profitably, effectively, and efficiently than they currently do with the cumbersome appointment process.

- VI. The usage of user reviews to determine the characteristics that affect user star ratings. Additionally, the study evaluates user reviews of nine currently available healthcare apps and makes strategic recommendations for raising user happiness and app ratings for healthcare providers. Based on the literature analysis and domain expertise, the authors have categorized the reviews into the twelve most important elements. On the other hand, the digital revolution of healthcare delivery has been significantly influenced by variables like lockdown/COVID, expenditure of time and money, resource effectiveness, convenience, and responsiveness. Demand for mobile health (mHealth) systems has surged in recent years considering the COVID-19 pandemic, and the preference of customers for doorstep services. Nevertheless, there is currently an absence of academic study on the variables impacting customer reviewsbased consumer

experiences and satisfaction levels on these platforms. By applying topic-modelling methodologies, it was determined that the influencing factors (predictors) divided them into two main categories: motivating association and strategic adoption. The research's conclusions indicate that providing a good user experience on m-health platforms is significantly influenced by time and money, convenience, responsiveness, and availability. This study offers crucial information about the variables that affect how well a user interacts with a mobile health app. These results will help application and business model developers comprehend the fundamental needs for creating a user-focused application. Additionally, if the mHealth sector develops throughout time, researchers will benefit from it by having methodological guidelines to follow when discovering novel characteristics in the future.

- VII. Mobile application development is a significant emerging field that is booming as a result of the rapid growth in the ownership of handheld cellular devices. Moreover, a lot of programs fall short of their goal of drawing in the target audience, which is based on their User Interface (UI) and User Experience (UX). As a result of this, developers frequently find it difficult to live up to users' expectations. A variety of studies that examined various elements of design and the user experience of mobile applications utilizing UX/UI have been conducted up to this point. Nevertheless, a large number of these previously conducted surveys only addressed a small number of the issues at hand and neglected to take into account other important factors including language, feedback, context, user behaviour, emotions, and control, visualization/graphics, and usability. The preferences and impressions of a diverse range of interested individuals have been gathered by the researchers of this study in their

pilot survey, which took into account all of the previously described features. Mobile application developers could utilize these choices as a guide, providing them with useful data. Their suggested method would assist designers and developers of mobile applications in concentrating on the specific UI/UX issues of those applications in light of their pertinent contexts. This study presents an unbiased comparison of the major UI and UX aspects that define a mobile application interface. A wide spectrum of concerned individuals' points of view and interests were discussed. Mobile application developers can use these choices as a reference, which provides them with helpful information. The authors' suggested method would assist designers and developers of mobile applications in concentrating on specific UI/UX issues related to their applicability. This study provides an in-depth comparison of the major UI and UX aspects that define a mobile application interface. When creating and developing different mobile applications, user interface (UI) and user experience (UX) are the two essential components. Therefore, when creating the application, developers should take into consideration everything described previously. The younger generation anticipates seamless mobile application engagement with their electronic devices. An application's aesthetic (UI) fosters users to interact with it more, while its functionality (UX) helps make a good first impression that encourages further engagement and pleasure from users. To design a mobile application, both components must be completely in equilibrium.

VIII. Implementation of system is mostly concentrating on receiving notifications through android applications Web-based notification systems have always had the capacity to become more prominent in our lives. With almost every aspect of our lives managed by handheld devices, it is obvious that we as forgetful human beings would like to be

notified about the important events in life such as meetings, appointments, commencement of events, and so on. Web notifications are largely used in the academic realm where schools and colleges are equipped with these systems to make the lives of students much more productive, all while notifying their guardians about their progress. These systems offer a solution for student information maintenance through an easy-to-use interface by utilizing an all-encompassing way to keep apprised of the different tasks that a school manages. Using the Android application, parents may obtain comprehensive information about their child's education, providing facts on attendance, fees owing, results, reports on progress, vital notices, school event details, instructor information, and more. In order to facilitate communication between parents and college teachers, the system also includes an inquiry message option. An effective technology as such has also been used in the healthcare realm, although there seems to be a lot of room for improvement.

- IX. SMTP is a widely adopted email standard, enables servers to send and receive messages. It operates as a push protocol for sending emails, complemented by POP or IMAP for retrieval. Most email systems that transmit messages over the internet utilize SMTP to transfer messages from one server to another; clients can use POP or IMAP to retrieve the messages. Moreover, emails are often sent from an electronic mail client to a mail server using SMTP. This is the reason that when we configure our utilities, we need to know the POP or IMAP server as well as the SMTP server. SMTP is used as the common mechanism for transporting electronic mail among different hosts within the transmission control protocol/Internet protocol (TCP/IP) suite. It is an application layer protocol an application that enables us to send messages electronically

over the internet is email. It offers a real-time, low-cost, and environmentally friendly way to distribute statistics to individuals. Nevertheless, one may also send non-textual content files as attachments sent in binary streams, such as sound and imagery. The most commonly used application of the World Wide Web was and still is electronic mail. It was one of the first things people accomplished online. Emails can also be shared between users of different Internet providers and in public and private networks that are not connected to the internet. This paper examines the protocols that are secured by SMTP (simple mail transfer protocol) and internet protocol (IP). The following are some recent general or experimental SMTP extensions:

- guide for numerous carrier environments,
- global delivery repute and deposition notifications
- content material conversion
- message monitoring
- internationalized scope with submission carrier extension for destiny message release

- X. An Android mobile application portal for doctor and patient booking was developed, aiming to improve the quality of healthcare by improving appointment times. They developed an effective booking system to enable all stakeholders to manage their bookings and daily work. The system also addresses privacy and security issues that are fundamental in medical applications. Feedback from all stakeholders was generally positive, and when all stakeholders tested the system, it showed high performance. The availability of high-speed Internet technologies and smart phones, doctors and patients need to improve their medical interactions. In recent times, parents have been put under heavy workloads and pressures, enabling them to forget important details about

their children, particularly vaccination milestones. Furthermore, a lack of awareness of the symptoms of the disease increases the likelihood of many serious and fatal diseases. Therefore, it is important to establish a system of appointment and consultation for medical vaccinations for children under the age of six. In this paper, the authors developed an effective reservation system using mobile applications based on Android and interactive websites. A centralized database was developed to process appointments and manage health records. Four modules were developed, i.e. the Patient Module Doctor Module, the Nurse Module and the Manager Module. Their proposed system aims to improve the quality of healthcare by improving appointment times. In addition to this, the system has reduced the time and effort required to carry out vaccination appointment processes and to ensure that vaccinations are delivered on time.

## **CHAPTER-3 RESEARCH GAPS OF EXISTING METHODS**

### **1. High Performance, Scalability, and Availability:**

Healthcare systems must be available to provide prompt care because, in the event of a malfunction, patient's lives are at risk. Performance and scalability are similar. The systems should function without interruption and smoothly scale on demand to accommodate increasing demands.

### **2. Preference for Speaking Over the Phone:**

Instead of making appointments online, many people would rather give their doctors a call. One way to address this would be to offer patients a phone option whereby they may make or cancel appointments and obtain access to their medicines and medical records.

### **3. Privacy and Security Issues:**

Sensitive and private healthcare data must be shielded from abuse and illegal access. Encryption, authentication, authorization, and auditing systems that guarantee data confidentiality, integrity, and availability can be put into place to do this.

### **4. Unpunctual Patients:**

Patients who are late for their appointments can seriously interfere with the efficient use of scheduling systems. These patients can set off a chain reaction of events that could result in longer wait times for other patients, hurried consultations, and more stress for medical staff. Moreover, it may lead to an underutilization of resources when spaces designated for patients who arrive late are left empty.

## CHAPTER-4 PROPOSED MOTHODOLOGY

### 4.1. Methodology:



**Figure 4.1.1 Methodology** data-flow diagram

The Figure 4.1.1 presented Data Flow Diagram illustrates the interaction between two roles: Doctors and Patients. Both the Doctor's and Receptionist's profile data is stored in the Firestore Database. The Receptionist is responsible for scheduling new appointments. When a new appointment is booked for a new patient, a corresponding patient profile is created and stored in the Firestore database. For returning patients, the Receptionist can streamline the appointment process by directly creating a new appointment without having to re-enter all personal details, as this information can be retrieved directly from the Firestore repository. The Doctor has exclusive access to view their appointment details and can input prescription details for each individual patient. These prescription

details are also stored in the Firestore database and can be reviewed by the Doctor Later. Only the Doctor has the authorization to access and check the prescription details of the patients.

#### **4.2. Algorithm:**

**Input:** Patient Details, Doctor Details

**Output:** Scheduling

**Step-1:** User can open the DocEasy and select the type of user. Shared Preferences are used to store and retrieve the user's role information. This information is checked to determine whether the user is a doctor or a recipient, and the appropriate activity is launched accordingly.

**Step-2:** Receptionist logs into Doc Easy App. Receptionist credentials are authenticated using authenticated using Firebase Authentication.

**Step-3:** The receptionist is responsible for scheduling appointments, and the details of each appointment are stored in a Firestore Database. The receptionist verifies the availability of the selected appointment by utilizing a straightforward if-else loop.

**Step-4:** Doctor logs into Doc Easy App. Doctor credentials are authenticated using Firebase Authentication. DocumentSnapshotListener code is used to listen for changes in the Firestore document containing the doctor's information. This triggers the re-fetching of appointments when the document is updated.

**Step-5:** Filter Chips are used to filter out the patients and appointment lists. The filterpatientsBySearch method filters patients based on the entered query.

**Step-6:** In the Doctor's Home page, the doctor has access to view all their scheduled appointments. Additionally, they can filter and display appointments for a specific day by selecting the desired date from the calendar for this a list view and DatePickerDialog has been used.

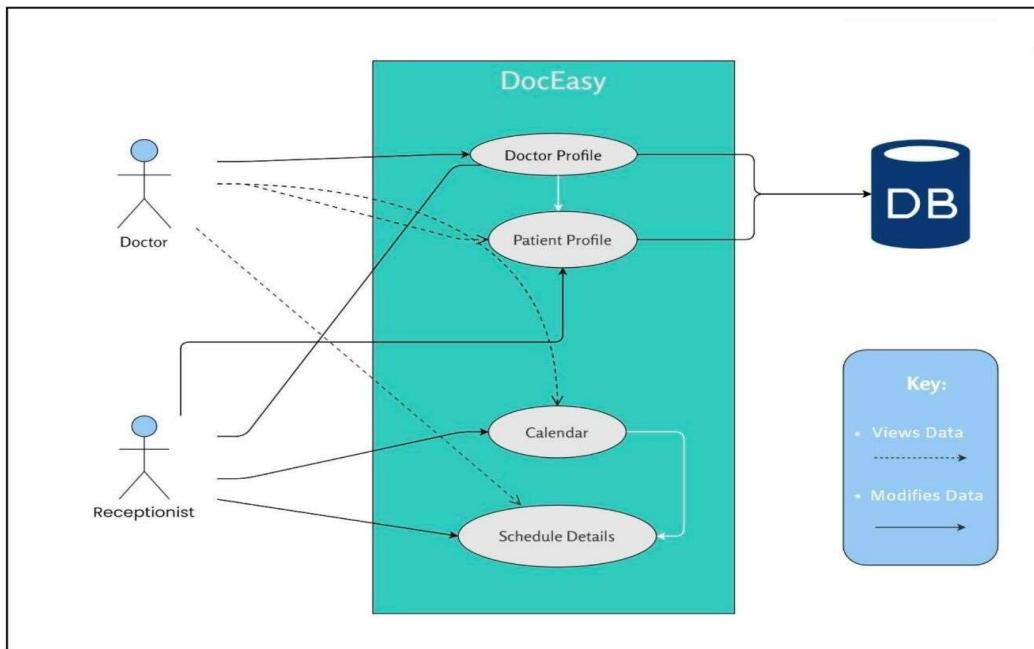
**Step-7:** Doctors can apply by entering the start date, end date and remarks. These details are stored in a HashMap.

## CHAPTER-5 OBJECTIVES

These objectives encapsulate key features and goals for healthcare application development:

- 1 Develop a secure and user-friendly healthcare application facilitating seamless communication between doctors and patients.
- 2 Implement robust authentication and authorization mechanisms for ensuring user data privacy and confidentiality.
- 3 Provide a feature-rich booking system, enabling patients to easily schedule appointments with healthcare professionals.
- 4 Ensure an intuitive user interface for accessibility, catering to both patients and doctors.
- 5 Incorporate Firebase services for reliable user authentication, real-time database updates, and cloud storage.
- 6 Establish a comprehensive user profile system, view appointment history, and communicate preferences.

## CHAPTER-6 SYSTEM DESIGN & IMPLEMENTATION



**Figure 6.1** Architecture Diagram

The Figure 6.1 depicts Architecture of the users and their respective functionalities, categorizing them into two distinct roles: doctors and receptionists. Receptionists, in their role, are responsible for scheduling new appointments and the details of these appointments are subsequently stored in the Firestore database. On the other hand, doctors have the capability to retrieve information about their patients directly from the Firestore database. The profile details of both receptionists and doctors are maintained within the database. It is important to note that only receptionists possess the authorization to create or delete appointments. On the contrary, doctors are limited to accessing appointments that are specifically designated for them, and they can input the prescription details for each individual patient which is stored in Firestores database.

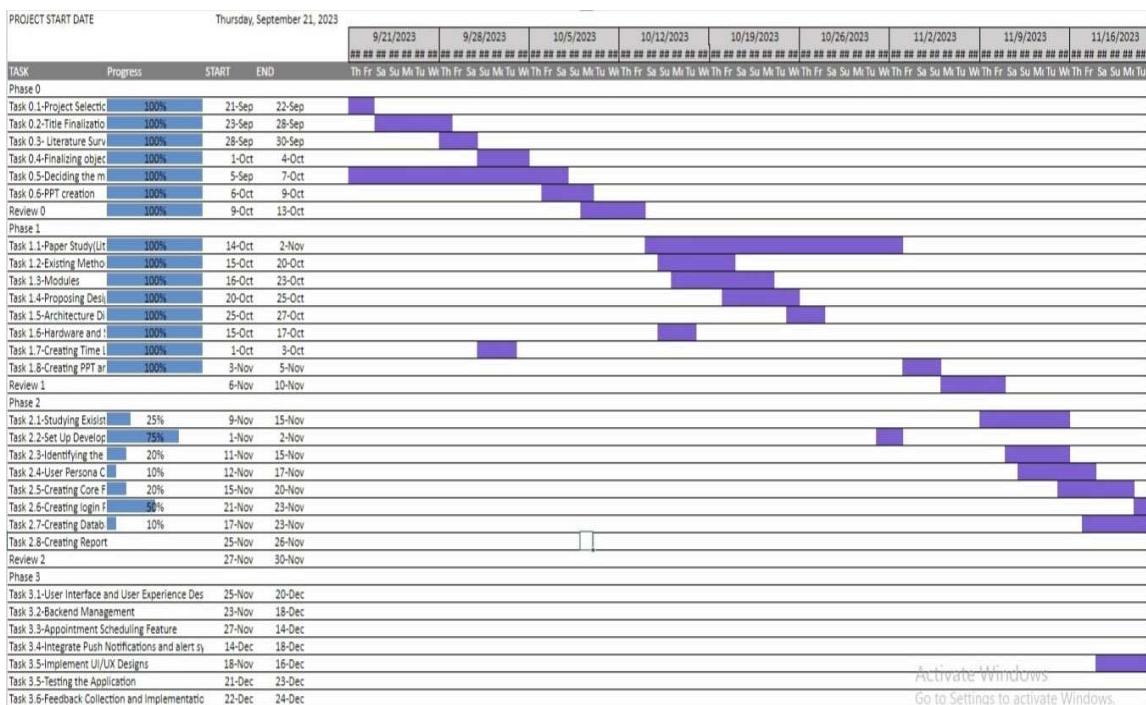
Here are some key features of Android Studio:

- Flexible Gradle-based build system: Android Studio uses Gradle as the foundation of the build system, allowing you to customize, configure, and extend the build process.
- Fast and feature-rich emulator for app testing: It comes with a fast emulator that helps you test your application on a variety of Android devices.
- Consolidated environment: Android Studio provides a unified environment where you can develop apps for all Android devices.
- Code templates and GitHub integration: These help you build common app features and import sample code.
- Extensive testing tools and frameworks: Android Studio provides a range of testing tools and frameworks to ensure your app works as expected.
- Built-in support for Google Cloud Platform: This makes it easy to integrate Google Cloud Messaging and App Engine.

Key Features of Firebase Include:

- Unlimited reporting, cloud messaging, authentication, and hosting services.
- In-built analytics dashboard that can report on 500 distinct events without any limits.
- It allows for audience segmentation, integration with other services, and facilitates app development.
- Superior performance and productivity.
- It supports a wide range of embedded applications, enhancing its flexibility.
- Compatibility with numerous operating systems.
- Being cloud-hosted, Firebase allows developers to scale on-demand effortlessly.

## CHAPTER-7 TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)



**Figure 7.1 Gantt Chart**

## CHAPTER-8

### OUTCOMES

#### **1. Efficient Appointment Management:**

DocEasy provides a sophisticated appointment management system, facilitating seamless scheduling, modification, and cancellation of appointments for both healthcare providers and patients. The user-friendly interface ensures a smooth and efficient scheduling experience.

#### **2. Appointment Delegation during Receptionist Leaves:**

A distinctive feature of DocEasy is its ability to manage appointment handovers seamlessly. In cases of receptionist leave, the application ensures continuity by enabling the transfer of appointment responsibilities to another designated staff member. This functionality ensures uninterrupted service, minimizing potential disruptions in the appointment management process.

#### **3. Streamlined In-Patient Care:**

DocEasy goes beyond standard appointment management by incorporating features tailored for in-patient care. The application offers a comprehensive overview of in-house patients, allowing healthcare providers to efficiently coordinate and manage their care. This includes monitoring bed availability, tracking patient progress, and facilitating communication among medical staff for a holistic approach to in-patient care.

#### **4. Support for Doctor's Appointment Planning:**

DocEasy empowers doctors with tools to plan and manage their appointments effectively. The application's intelligent scheduling algorithm considers various factors such as consultation time, patient preferences, and doctor availability. This ensures an optimized schedule, reducing wait times and enhancing overall patient satisfaction. Additionally, the application sends timely reminders and notifications, assisting doctors in staying organized and maintaining punctuality.

## CHAPTER-9 RESULTS AND DISCUSSIONS

### **9.1. Software and Hardware Requirements:**

#### **Software Requirements:**

##### **1.Operating System:**

The application will be compatible with Android (version 8.0 and above) to cater to a wide range of users.

##### **2.Database:**

A robust database system such as Firebase cloud Firestore will be used for storing appointment details, doctor profiles, patient information, etc.

Cloud Firestore is a flexible, scalable database for developing mobile, web and server applications from Firebase and Google Cloud.

Like Firebase's Realtime Database, it syncs the data of client applications through real-time listeners and provides offline mobile and web support, enabling you to build responsive apps that work regardless of network latency or Internet connectivity.

The Cloud Firestore is also compatible with other Firebase and Google Cloud products, including Cloud Functions.

##### **3.Backend Server:**

A server-side framework for handling requests from the mobile application.

#### **Hardware Requirements:**

##### **1.Device:**

The application will be able to run on Android based smartphone devices with at least 2GB of RAM for smooth operation.

##### **2.Storage:**

Depending on the size of the completed application and the amount of data to be stored locally, a minimum of 100MB of free storage space might be required.

##### **3.Internet:**

A stable internet connection for real-time updates of appointments and other data is required.

#### **4.Screen Size:**

The application is optimized for screen sizes of 4.7 inches and above to ensure a good user experience.

#### **5.Battery:**

As the application might be used frequently throughout the day, the device should have a decent battery life.

#### **9.2. Data set:**

Sl.No.	Event	Attribute & value	Expected result
1.	Upon selecting the signup button, provide accurate and suitable information that can be utilized for authenticating the receptionist's login.	Name: Mohana Email: <a href="mailto:mohana@gmail.com">mohana@gmail.com</a> Phone number: 9000023456 Username: Mohana Password: mohana@123	successfully Signed in
2.	Upon selecting the signup button, provide accurate and suitable information that can be utilized for authenticating the receptionist's login.	Name: Manoj Email: <a href="mailto:manoj@gmail.com">manoj@gmail.com</a> Phone number: 9452236036 Username: Manoj Password: manoj@123	successfully Signed in
3.	Upon selecting the signup button, provide accurate and suitable information that can be utilized for authenticating the receptionist's login.	Name: Vikrant Email: <a href="mailto:vikrant@gmail.com">vikrant@gmail.com</a> Phone number: 9000023235 Username: Vikrant Password: vikrant@123	successfully Signed in

Table 9.2.1 Receptionist Signup Activity

Sl.No.	Event	Attribute & value	Expected result
1.	Upon selecting the signup button, provide accurate and suitable information that can be utilized for authenticating the receptionist's login.	Name: Mishra Username: Dr.Mishra Phone number: 9603375633 Speciality: ENT Specialist Experience: 12 Email: <a href="mailto:mishra@gmail.com">mishra@gmail.com</a> Working Shift: 9:00 AM -12:30 PM Password: mishra@1234	successfully Signed in

2.	Upon selecting the signup button, provide accurate and suitable information that can be utilized for authenticating the receptionist's login.	Name: Swapna Username: Dr.Swapna Phone number: 9693349049 Speciality: Dermatologist Experience: 9 Email: <a href="mailto:swapna@gmail.com">swapna@gmail.com</a> Working Shift: 9:00 AM -12:30 PM Password: swapna@1234	successfully Signed in
3.	Upon selecting the signup button, provide accurate and suitable information that can be utilized for authenticating the receptionist's login.	Name: Bhagya Username: Dr.Bhagya Phone number: 9600478922 Speciality: Gynecologist Experience: 10 Email: <a href="mailto:bhagya@gmail.com">bhagya@gmail.com</a> Working Shift: 14:30 PM - 20:30 PM Password: bhagya@1234	successfully Signed in

Table 9.2.2 Doctor Signup Activity

Sl.No.	Event	Attribute & value	Expected result
1.	Verify the receptionist's login upon entering the correct email address and password.	Email: <a href="mailto:mohana@gmail.co m">mohana@gmail.co m</a> Password: mohana@123	successfully logged in
2.	Verify the receptionist's login upon entering the correct email address and password.	Email: <a href="mailto:manoj@gmail.com">manoj@gmail.com</a> Password: manoj@123	successfully logged in
3.	Upon selecting the signup button, provide accurate and suitable information that can be utilized for authenticating the receptionist's login.	Email: <a href="mailto:vikrant@gmail.com">vikrant@gmail.com</a> Password: vikrant@123	successfully logged in

Table 9.2.3 Receptionist Login Activity

Sl.No.	Event	Attribute & value	Expected result
1.	Verify the receptionist's login upon entering the correct email address and password.	Email: mishra@gmail Password: mishra@1234	successfully logged in
2.	Verify the receptionist's login upon entering the correct email address and password.	Email: <a href="mailto:swapna@gmail.com">swapna@gmail.com</a> Password: swapna@1234	successfully logged in
3.	Upon selecting the signup button, provide accurate and suitable information that can be utilized for authenticating the receptionist's login.	Email: <a href="mailto:bhagya@gmail.com">bhagya@gmail.com</a> Password: bhagya@1234	successfully logged in

Table 9.2.4 Doctor Login Activity

Sl.No.	Event	Attribute & value	Expected result
1.	Initiating a new appointment by providing the necessary patient information.	Name: Prashanth Age: 58 Gender: Male Phone number: 9555324567 Address: Yelahanka Speciality: General Physician Doctor: Dr.Mishra In Or Outpatient: Outpatient Appointment Date: 05/01/2024 Time Slot: 10:30 AM - 11:00 AM Medical Remarks: Fever	New Appointment created Successfully
2.	Initiating a new appointment by providing the necessary patient information.	Name: Moksha Age: 25 Gender: Female Phone number: 9000345632 Address: Yelahanka Speciality: General Physician Doctor: Dr.Mishra In Or Outpatient: Outpatient Appointment Date: 05/01/2024 Time Slot: 12:30 AM - 12:30 AM Medical Remarks: Fever	New Appointment created Successfully
3.	Initiating a new appointment by providing the necessary patient information.	Name: Arjun Age: 32 Gender: Male Phone number: 9525678912 Address: Rajanakunte Speciality: General Physician Doctor: Dr.Kokila In Or Outpatient: Outpatient Appointment Date: 05/01/2024 Time Slot: 10:30 AM - 11:00 AM Medical Remarks: Fever	New Appointment created Successfully

Table 9.2.5 New Appointment Activity

Sl.No.	Event	Attribute & value	Expected result
1.	Doctor Applying Leave	Name: Dr.Mishra Start Date: 2/01/2024 Start Date: 5/01/2024 Remarks: Medical Leave	Leave applied successfully
2.	Doctor Applying Leave	Name: Dr.Kokila Start Date: 3/01/2024 Start Date: 5/01/2024 Remarks: Vacation	Leave applied successfully

3.	Doctor Applying Leave	Name: Dr.Swetha Start Date: 2/01/2024 Start Date: 5/04/2024 Remarks: Maternity Leave	Leave applied successfully
----	-----------------------	---	----------------------------

Table 9.2.6 Doctor Applying Leave

Sl.No.	Event	Attribute & value	Expected result
1.	Initiating a new appointment by providing the necessary patient information.	Name: Prashanth Age: 58 Gender: Male Phone number: 9555324567 Address: Yelahanka In Or Outpatient: Outpatient Appointment Date: 05/01/2024	New Patient created successfully
2.	Initiating a new appointment by providing the necessary patient information.	Name: Moksha Age: 25 Gender: Female Phone number: 9000345632 Address: Yelahanka In Or Outpatient: Outpatient Appointment Date: 05/01/2024	New Patient created successfully
3.	Initiating a new appointment by providing the necessary patient information.	Name: Arjun Age: 32 Gender: Male Phone number: 9525678912 Address: Rajanakunte In Or Outpatient: Outpatient Appointment Date: 05/01/2024	New Patient created successfully

Table 9.2.7 Patient Creation Activity

Table 9.1.1 Displays the information that a receptionist must enter during the Signup activity. Table 9.1.2 Displays the information that a doctor must enter during the Signup activity. Table 9.1.3 Displays the information that a receptionist must enter during the Login activity. Table 9.1.4 Displays the information that a doctor must enter during the Login activity. Table 9.1.5 Displays the information that a receptionist enters to a new appointment. Table 9.1.6 Displays the information that Doctor enters to apply leave. Table 9.1.7 Displays the information that required to make a patient creation.

## CHAPTER-10 CONCLUSION

Based on the conducted research on topics such as the history and evolution of healthcare applications, we have understood common problems that occur while using them. Our aim is to develop an application inspired by pre-existing applications, that addresses and resolves the prior discovered issues. Comprehensive study into the notification system is also required to be able to implement it in our application.

Furthermore, we intend for the mobile app to address the challenges doctors faced during appointment planning, outpatient and in-patient care management, and continuity of care during days of absence. It will provide doctors with convenient tools to streamline their work processes, leading to enhanced patient care quality and effective collaboration between medical professionals.

## REFERENCES

- [1] Nazia, S., & Ekta, S. (2014). Online appointment scheduling system for hospitals—an analytical study. *Int J Innov Res Sci Eng Techno*, 4(1), 21-7.
- [2] John Lekan, Akinode. (2017). Design and Implementation of a Patient Appointment and Scheduling System. 4. 16-23. 10.17148/IARJSET.2017.41203.
- [3] Dat, Luu Tien. (2019). “Associations Between Waiting Time and Patient Satisfaction Level at Tan Phu District Hospital in Ho Chi Mihn City, Vietnam”. Master’s Theses. 115.
- [4] Chauhan, Roma, and Kumar, Amit. (2013). “Practo Technologies: the online way of life!” *Emerald Emerging Markets Case Studies*, Volume 3, Issue 7, pages 1-19. DOI: 10.1108/EEMCS-06-2013-0127.
- [1] Ismail, N. S., Kasim, S., Jusoh, Y. Y., Hassan, R., & Alyani, A. (2017). Medical appointment application. *Acta Electronica Malaysia*, 1(2), 5-9.
- [5] Pal, S., Biswas, B., Gupta, R., Kumar, A., & Gupta, S. (2023). Exploring the factors that affect user experience in mobile-health applications: A text-mining and machine-learning approach. *Journal of Business Research*, 156, 113484.
- [6] Sandesara, M., Bodkhe, U., Tanwar, S., Alshehri, M. D., Sharma, R., Neagu, B. C., ... & Raboaca, M. S. (2022). Design and Experience of Mobile Applications: A Pilot Survey. *Mathematics*, 10(14), 2380.
- [7] Jadhav, M. L. S., Mhade, M. T. R., Jagushte, M. S. S., & Gosavi, M. M. K. (2016). Web based Notification Management System with Android Application. *Int. J. Sci. Technol. Eng.*, 2(10), 1218-1220.
- [8] Vidya, K. (2020). An Overview of E-mail and Protocols Included with the IP and SMTP. *Anveshana’s Int. J. Res. Eng. Appl. Sci*, 5.
- [9] Qabajeh, M. M., Mousa, S., Saleh, H. A., & Hasan, J. A. (2023). Designing an Android Application for Electronic Booking in Health Clinics and Building Health Record System. *Journal of Advances in Information Technology*, 14(3).
- [10] Android Developer Guides: <https://developer.android.com/guide>
- [11] How to Create an Android App With Android Studio:

---

<https://www.instructables.com/How-To-Create-An-Android-App-With-Android-Studio/>

- [12] Android RecyclerView, Android Card View Example Tutorial:  
<https://www.digitalocean.com/community/tutorials/android-recyclerview><https://www.digitalocean.com/community/tutorials/android-cardview-example-tutorial>
  - [13] Everything You Should Know About Healthcare App Development in 2023  
<https://www.osplabs.com/insights/everything-you-should-know-about-healthcare-app-development-in-2022/>
  - [14] The Future of Medical App Development:  
<https://www.cprime.com/resources/blog/the-future-of-medical-app><https://www.cprime.com/resources/blog/the-future-of-medical-app-development/>
  - [15] Revolutionizing Healthcare: The Benefits And Challenges Of Booking Doctors Appointment Online:  
<https://www.kommunicate.io/blog/benefits-and-challenges-of-booking-doctors-appointment-online/>
  - [16] Doctors: Patient Wait Times are a Bigger Problem Than You Think  
[https://mobius.md/2023/07/13/patient-wait-times-are-a-bigger-problem-than-you-think/you-think/](https://mobius.md/2023/07/13/patient-wait-times-are-a-bigger-problem-than-you-think/)
  - [17] Waiting for Health Care: A Terrible and Treatable Disease  
<https://www.fraserinstitute.org/article/waiting-health-care-terrible-and-treatable-disease>
  - [18] Features and Benefits of a Hospital Management System  
<https://radixweb.com/blog/hospital-management-system-features-and-benefits>
  - [20] Hurry Up and Wait: Long Wait Times Cripple Access to Care, Health Outcomes, and Patient Experience  
<https://www.forrester.com/blogs/hurry-up-and-wait-long-wait-times-cripple-access-to-care-health-outcomes-and-patientexperience/>
-

## **APPENDIX-A PSUEDOCODE**

## Onboarding Screens:

The screenshot shows the Android Studio interface with two tabs open: `MainActivity.java` and `activity_main.xml`.

**MainActivity.java:**

```
27
28     @Override
29     protected void onCreate(Bundle savedInstanceState) {
30         super.onCreate(savedInstanceState);
31         setContentView(R.layout.activity_main);
32
33         backbtn = findViewById(R.id.backbtn);
34         nextbtn = findViewById(R.id.nextbtn);
35         skipbtn = findViewById(R.id.skipButton);
36
37         FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
38         if (user != null) {
39             // User is signed in, open their
40             SharedPreferences sharedpreferences = MainActivity.this.getSharedPreferences("name", 0);
41             String role = sharedpreferences.getString("role", "null");
42             if(role != null) {
43                 Intent i;
44                 if(role.equals("doctor")) {
45                     i = new Intent(getApplicationContext(), DocLoginPage.class);
46                 } else {
47                     i = new Intent(getApplicationContext(), RecLoginPage.class);
48                 }
49                 startActivity(i);
50                 finish();
51             } else {
52                 //Do nothing
53             }
54         }
55
56         // Harsha Vardhan Reddy
57         backbtn.setOnClickListener(new View.OnClickListener() {
```

**activity\_main.xml:**

```
8
9
10    <Button
11        android:id="@+id/skipButton"
12        android:layout_width="wrap_content"
13        android:layout_height="wrap_content"
14        android:background="#00000000"
15        android:text="Skip"
16        android:textColor="@color/buttonText"
17        android:textSize="20sp"
18        app:layout_constraintEnd_toEndOf="parent"
19        app:layout_constraintTop_toTopOf="parent" />
20
21    <androidx.viewpager.widget.ViewPager
22        android:id="@+id/slideViewPager"
23        android:layout_width="match_parent"
24        android:layout_height="match_parent"
25        android:layout_marginVertical="100dp"
26        android:contentDescription="dfg"
27
28        app:layout_constraintBottom_toBottomOf="parent"
29        app:layout_constraintEnd_toEndOf="parent"
30        app:layout_constraintStart_toStartOf="parent"
31        app:layout_constraintTop_toTopOf="parent"
32        android:hint="Skip the onboarding screens" />
33
34    <ImageView
35        android:id="@+id/ellipse1"
36        android:layout_width="wrap_content"
37
38        androidx.constraintlayout.widget.ConstraintLayout > androidx.viewpager.widget.ViewPager
```

Figure 11.1.1(Code Execution 1)

## Signup Page:

Figure 11.1.2(Code Execution 2)

## Login Page:

The screenshot shows the Android Studio interface with two open files: `DocLoginPage.java` and `activity_doc_login_page.xml`.

**DocLoginPage.java:**

```
36     @Override
37     protected void onCreate(Bundle savedInstanceState) {
38         super.onCreate(savedInstanceState);
39         setContentView(R.layout.activity_doc_login_page);
40         doc_login_email = findViewById(R.id.doc_login_email);
41         doc_login_password = findViewById(R.id.doc_login_password);
42         doc_login_button = findViewById(R.id.doc_login_button);
43         doc_signupRedirectText = findViewById(R.id.doc_signupRedirectText);
44         progressBar4=findViewById(R.id.progressBar4);
45         backButton=findViewById(R.id.backbutton);
46         dAuth=firebaseAuth.getInstance();
47         FirebaseAuth user = FirebaseAuth.getInstance().getCurrentUser();
48         if (user != null) {
49             // User is signed in, open MainActivity
50             startMainActivity(user);
51             return; // Finish the current activity to prevent going back to it
52         }
53
54         Harsha Vardhan Reddy +1
55         doc_signupRedirectText.setOnClickListener(new View.OnClickListener() {
56             @ Harsha Vardhan Reddy +1
57             @Override
58             public void onClick(View v) {
59                 Intent intent = new Intent(getApplicationContext(), DocSignupFrontPage);
60                 startActivity(intent);
61                 finish();
62             }
63         });
64         Harsha Vardhan Reddy +1
65         doc_login_button.setOnClickListener(new View.OnClickListener() {
66             @ Harsha Vardhan Reddy +1
67             @Override
68             public void onClick(View v) {
69                 String doc_email = doc_login_email.getText().toString().trim();
```

**activity\_doc\_login\_page.xml:**

```
48 <EditText
49     android:id="@+id/doc_login_password"
50     android:layout_width="match_parent"
51     android:layout_height="50dp"
52     android:layout_marginTop="20dp"
53     android:background="@drawable/white_background"
54     android:drawableLeft="@drawable/baseline_lock_24"
55     android:drawablePadding="8dp"
56     android:hint="Password"
57     android:inputType="textPassword"
58     android:padding="8dp"
59     android:textColor="@color/black" />
60
61 <Button
62     android:id="@+id/doc_login_button"
63     android:layout_width="match_parent"
64     android:layout_height="60dp"
65     android:layout_marginTop="30dp"
66     android:backgroundTint="@color/active"
67     android:text="Login"
68     android:textSize="18sp"
69     app:cornerRadius="20dp" />
70
71 <Button
72     android:id="@+id/doc_signupRedirectText"
73     android:layout_width="wrap_content"
74     android:layout_height="wrap_content"
75     android:layout_gravity="center"
76     android:layout_marginTop="10dp"
77     android:padding="8dp"
78     android:text="Not yet registered? Sign Up"
```

Figure 11.1.3(Code Execution 3)

## **Home Page:**

The screenshot shows the Android Studio interface with two tabs open: 'DocHomePage.java' and 'activity\_doc\_home\_page.xml'.

**DocHomePage.java**

```
public void logout(View view){  
    FirebaseAuth.getInstance().signOut();  
    SharedPreferences sharedPreferences = DocHomePage.this.getPreferences( MODE_PRIVATE );  
    sharedPreferences.edit().remove( key: "role" ).commit();  
    startActivity(new Intent(getApplicationContext(), DocLoginPage.class));  
    finish();  
}  
  
± Sagili Mythili +2  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_doc_home_page);  
    doc_profile=findViewById(R.id.doc_profile);  
    doc_logout_button = findViewById(R.id.doc_logout_button);  
    myAppointments = findViewById(R.id.myAppointments);  
    apply_leave=findViewById(R.id.apply_leave);  
    dAuth= FirebaseAuth.getInstance();  
    doc_user=firebasedatabase.getInstance();  
    postConsultation=findViewById(R.id.postConsultation);  
    welcome_username = findViewById(R.id.welcome_username);  
    ± Harsha Vardhan Reddy +1  
    doc_profile.setOnClickListener(new View.OnClickListener() {  
  
        ± Harsha Vardhan Reddy +1  
        @Override  
        public void onClick(View view) {  
            Intent i = new Intent( packageContext: DocHomePage.this, DocProf.class);  
            startActivity(i);  
        }  
    });  
};  
± Sagili Mythili  
myAppointments.setOnClickListener(new View.OnClickListener() {
```

**activity\_doc\_home\_page.xml**

```
<TableRow>  
    <androidx.cardview.widget.CardView  
        android:id="@+id/myAppointments"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_margin="10dp"  
        app:cardCornerRadius="20dp"  
        app:cardElevation="10dp"  
        app:cardMaxElevation="12dp"  
        app:cardPreventCornerOverlap="true">  
  
        <androidx.appcompat.widget.LinearLayoutCompat  
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:orientation="vertical">  
  
            <ImageView  
                android:layout_width="136dp"  
                android:layout_height="138dp"  
                android:layout_gravity="center"  
                android:layout_margin="10dp"  
                android:src="@drawable/patients_recors_3" />  
  
            <TextView  
                android:id="@+id/textView"  
                android:layout_width="wrap_content"  
                android:layout_height="wrap_content"  
                android:layout_gravity="center_horizontal"  
                android:text="My Appointment" />  
        </androidx.appcompat.widget.LinearLayoutCompat>  
    </androidx.cardview.widget.CardView>
```

Figure 11.1.4(Code Execution 4)

## My Appointments List View:

The screenshot shows the Android Studio interface with three tabs open:

- MyAppointments.java**: Contains logic for selecting a date, initializing a date picker, and setting up a RecyclerView adapter.
- MyAppointmentsAdapter.java**: Extends RecyclerView.Adapter and implements onCreateViewHolder, onBindViewHolder, and getItemCount methods.
- activity\_my\_appointments\_list.xml**: The XML layout for the appointment list, featuring a ListView and a TextView for empty state.

Figure 11.1.5 (Code Execution 5)

## Post Consultation List View:

The screenshot shows the Android Studio interface with two tabs open:

- PostConsultation.java**: Handles events from Firebase, specifically adding new prescriptions to a RecyclerView adapter.
- PostConsultationAdapter.java**: Extends RecyclerView.Adapter and implements onCreateViewHolder, onBindViewHolder, and getItemCount methods.

Figure 11.1.6 (Code Execution 6)

## Apply Leaves:

The screenshot shows the Android Studio interface with two tabs open: `ApplyLeave.java` and `activity_apply_leave.xml`.

**Java Code (`ApplyLeave.java`):**

```
42     @Override
43     protected void onCreate(Bundle savedInstanceState) {
44         super.onCreate(savedInstanceState);
45         setContentView(R.layout.activity_apply_leave);
46         startDate = findViewById(R.id.select_start_date);
47         endDate = findViewById(R.id.select_end_date);
48         remarks = findViewById(R.id.remarks);
49         submit_btn = findViewById(R.id.submit_btn);
50         firestore = FirebaseFirestore.getInstance();
51         mAuth = FirebaseAuth.getInstance();
52         startDate.setOnClickListener(v -> {
53             Calendar calendar = Calendar.getInstance();
54             DatePickerDialog datePickerDialog = new DatePickerDialog(
55                 context: this,
56                 + Sagili Mythili
57                 new DatePickerDialog.OnDateSetListener() {
58                     + usage + Sagili Mythili
59                     @Override
60                     public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {
61                         Calendar selectedDate = Calendar.getInstance();
62                         selectedDate.set(Calendar.YEAR, year);
63                         selectedDate.set(Calendar.MONTH, month);
64                         selectedDate.set(Calendar.DAY_OF_MONTH, dayOfMonth);
65                         SimpleDateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy");
66                         String formattedDate = dateFormat.format(selectedDate.getTime());
67                         startDateValue = formattedDate;
68                         startDate.setText(startDateValue);
69                     }
70                 },
71                 calendar.get(Calendar.YEAR),
72                 calendar.get(Calendar.MONTH),
73                 calendar.get(Calendar.DAY_OF_MONTH));
74             datePickerDialog.show();
75         });
76     }
77 }
```

**XML Layout (`activity_apply_leave.xml`):**

```
9
10 <Imageview
11     android:id="@+id/imageView14"
12     android:layout_width="216dp"
13     android:layout_height="151dp"
14     app:layout_constraintBottom_toBottomOf="parent"
15     app:layout_constraintEnd_toEndOf="parent"
16     app:layout_constraintHorizontal_bias="0.497"
17     app:layout_constraintStart_toStartOf="parent"
18     app:layout_constraintTop_toTopOf="parent"
19     app:layout_constraintVertical_bias="0.027"
20     app:srcCompat="@drawable/doc_leave" />
21
22 <Textview
23     android:id="@+id/select_start_date"
24     android:layout_width="wrap_content"
25     android:layout_height="31dp"
26     android:fontFamily="Gfont/poppins_bold"
27     android:gravity="center"
28     android:text="Select Start Date"
29     android:textSize="20sp"
30     app:layout_constraintBottom_toBottomOf="parent"
31     app:layout_constraintEnd_toEndOf="parent"
32     app:layout_constraintHorizontal_bias="0.471"
33     app:layout_constraintStart_toStartOf="parent"
34     app:layout_constraintTop_toTopOf="parent"
35     app:layout_constraintVertical_bias="0.311" />
36
37 <Textview
38     android:id="@+id/remarks_label"
39     android:layout_width="150dp"
40     android:layout_height="38dp"
```

Figure 11.1.7(Code Execution 7)

## All Doctors List:

The screenshot shows two tabs open in the Android Studio interface: 'AllDoctors.java' on the left and 'activity\_all\_doctors.xml' on the right.

**AllDoctors.java**

```
1 usage + Sagili Mythili
private void EventChangeListener() {
    Sagili Mythili
    doc_user.collection( collectionPath: "doc_user" ).orderBy( field: "username", Query.Direction
    Sagili Mythili
    @Override
    public void onEvent(@Nullable QuerySnapshot value, @Nullable FirebaseFirestoreRe
        if (error != null) {
            Log.e( tag: "Error", error.getMessage());
            return;
        }
        for (DocumentChange dc : value.getDocumentChanges()) {
            if (dc.getType() == DocumentChange.Type.ADDED) {
                Doctor doctor = dc.getDocument().toObject(Doctor.class);
                doctor.setMid(dc.getDocument().getId());
                doctorArrayList.add(doctor);
            }
            doctorAdapter.notifyDataSetChanged();
        }
    });
}
+ Sagili Mythili
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu, menu);

    MenuItem menuItem = menu.findItem(R.id.action_search);
    SearchView searchView = (SearchView) menuItem.getActionView();
    if (searchView != null) {
        searchView.setQueryHint("Search Here...");
    }
}
+ Sagili Mythili
```

**activity\_all\_doctors.xml**

```
<RelativeLayout
    android:id="@+id/relativeLayout"
    android:layout_width="match_parent"
    android:layout_height="84dp"
    tools:layout_editor_absoluteX="1dp">

    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:background="@color/active"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0" />

    <ListView
        android:id="@+id/listViewForDoctors"
        android:layout_width="match_parent"
        android:layout_height="631dp"
        android:layout_alignParentTop="true"
        android:layout_alignParentBottom="true"
        android:layout_marginTop="10dp"
        android:layout_marginBottom="39dp"
        android:focusable="true"
        android:focusableInTouchMode="true" />
```

Figure 11.1.8(Code Execution 8)

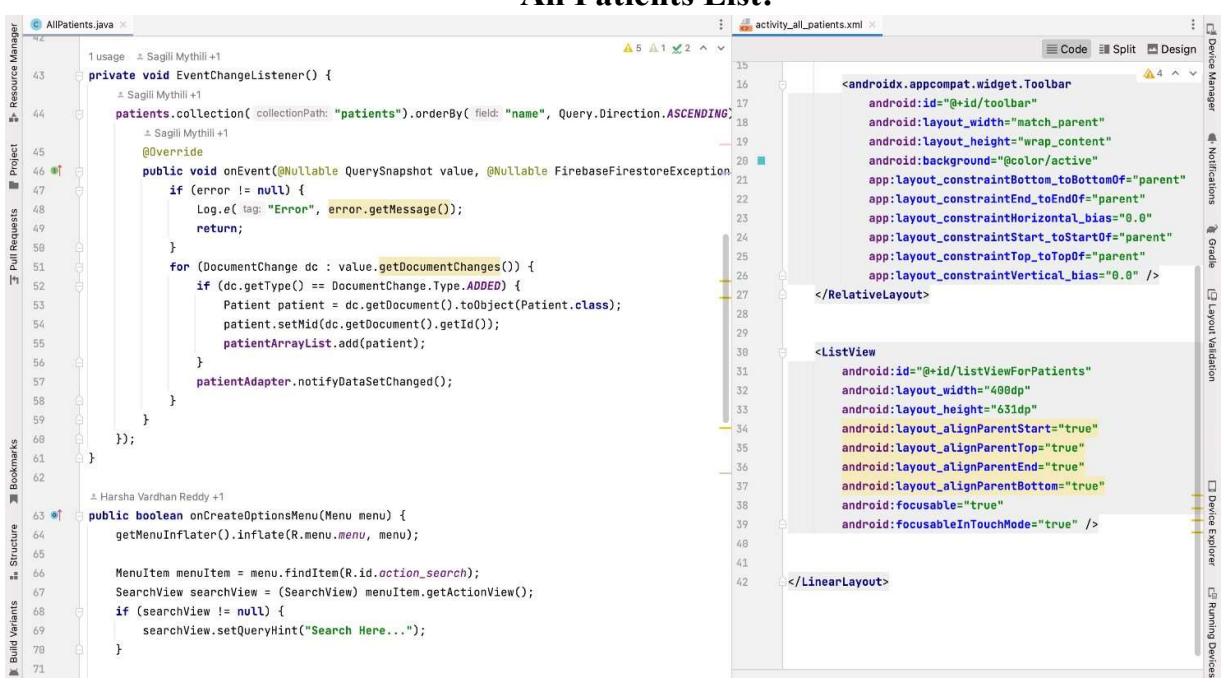


Figure 11.1.9(Code Execution 9)

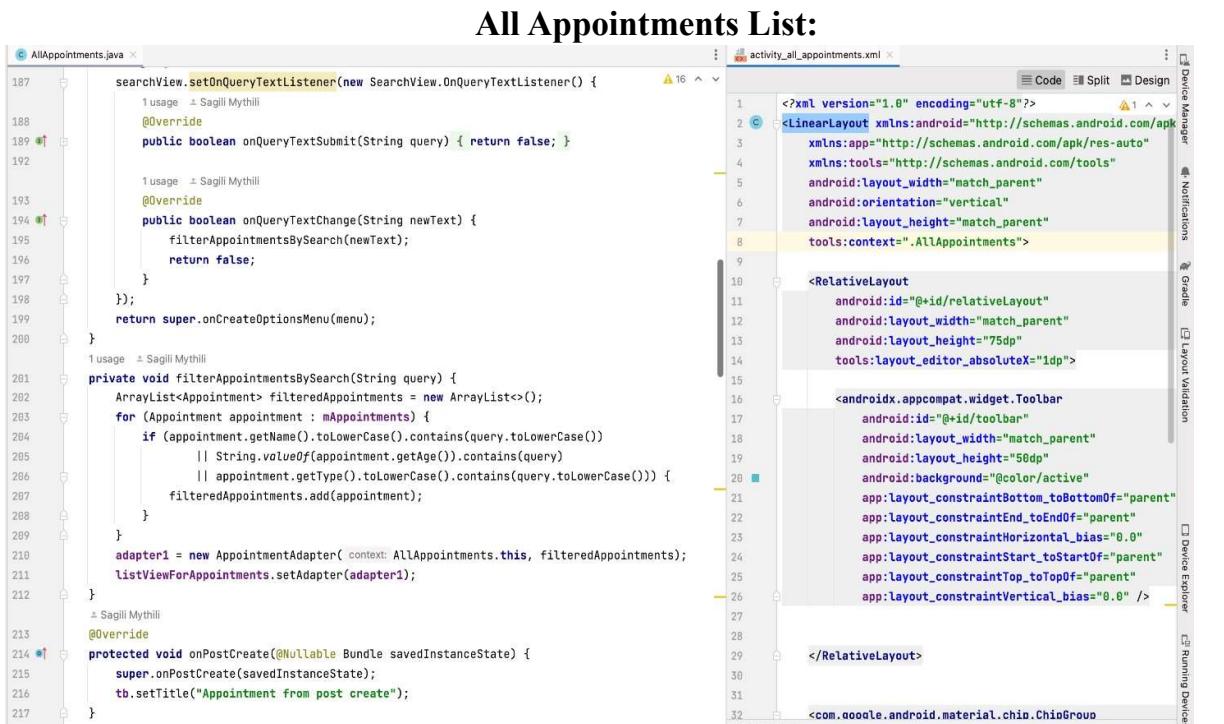


Figure 11.1.10(Code Execution 10)

## Patient Details:

The screenshot shows the Android Studio interface with two tabs open: 'PatientDetails.java' and 'activity\_patient\_details.xml'. The Java code handles the onCreate method, setting up views and retrieving data from extras. The XML file defines a ConstraintLayout with several TextViews for displaying patient information.

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_patient_details);
    name=findViewById(R.id.name);
    gender=findViewById(R.id.gender);
    phoneNumber=findViewById(R.id.phoneNumber);
    address=findViewById(R.id.address);
    bookappointment=findViewById(R.id.bookappointment);
    inOrOut=findViewById(R.id.inOrOut);
    date=findViewById(R.id.date);

    textView_age=findViewById(R.id.textView_age);
    patients= Firebase Firestore.getInstance();
    p_documentID = getIntent().getStringExtra("documentId");
    Bundle extras = getIntent().getExtras();
    if(extras!=null){
        p_name = extras.getString("name");
        age = Integer.parseInt(extras.getString("age"));
        p_gender = extras.getString("gender");
        p_phoneNumber = extras.getString("phoneNumber");
        p_inOrOut = extras.getString("inOrOut");
        p_date = extras.getString("date");
        p_address=extras.getString("address");
    }
    name.setText(p_name);
    gender.setText(p_gender);
    phoneNumber.setText(p_phoneNumber);
    address.setText(p_address);

    inOrOut.setText(p_inOrOut);
    date.setText(p_date);
}

```

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".PatientDetails">

    <TextView
        android:id="@+id/textView21"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="@font/poppins_medium"
        android:text="Gender"
        android:textColor="@color/active"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.036"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.334" />

    <TextView
        android:id="@+id/textView22"
        android:layout_width="91dp"
        android:layout_height="53dp"
        android:fontFamily="@font/poppins_medium"
        android:text="Phone Number"
        android:textColor="@color/active"
        android:textSize="20sp" />

```

Figure 11.1.11(Code Execution 11)

## Appointment Details:

The screenshot shows the Android Studio interface with two tabs open: 'AppointmentDetails.java' and 'activity\_appointment\_details.xml'. The Java code handles the onCreate method, setting up views and retrieving data from extras. The XML file defines a ConstraintLayout with several TextViews for displaying appointment details.

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_appointment_details);
    name = findViewById(R.id.name);
    gender = findViewById(R.id.gender);
    phoneNumber = findViewById(R.id.phoneNumber);
    address = findViewById(R.id.address);
    type = findViewById(R.id.type);
    doctorName = findViewById(R.id.doctorName);
    inOrOut = findViewById(R.id.inOrOut);
    date = findViewById(R.id.date);
    time = findViewById(R.id.time);
    remarks = findViewById(R.id.remarks);
    documentID = findViewById(R.id.documentID);
    textView_age = findViewById(R.id.textView_age);
    cancel_appointment = findViewById(R.id.cancel_appointment);
    appointments = Firebase Firestore.getInstance();
    p_documentID = getIntent().getStringExtra("documentId");
    Bundle extras = getIntent().getExtras();
    if (extras != null) {
        p_name = extras.getString("name");
        age = Integer.parseInt(extras.getString("age"));
        p_gender = extras.getString("gender");
        p_phoneNumber = extras.getString("phoneNumber");
        p_address = extras.getString("address");
        p_type = extras.getString("type");
        p_doctorName = extras.getString("doctor");
        p_inOrOut = extras.getString("inOrOut");
        p_date = extras.getString("date");
        p_time = extras.getString("time");
        p_remarks = extras.getString("remarks");
    }
}

```

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".AppointmentDetails"
    tools:ignore="ExtraText">

    <TextView
        android:id="@+id/textView20"
        android:layout_width="146dp"
        android:layout_height="37dp"
        android:fontFamily="@font/poppins_medium"
        android:text="DocumentID"
        android:textColor="@color/active"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.062"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.941" />

    <TextView
        android:id="@+id/textView21"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="@font/poppins_medium"
        android:text="Gender"
        android:textColor="@color/active"
        android:textSize="20sp" />

```

Figure 11.1.12(Code Execution 12)

## APPENDIX-B SCREENSHOTS

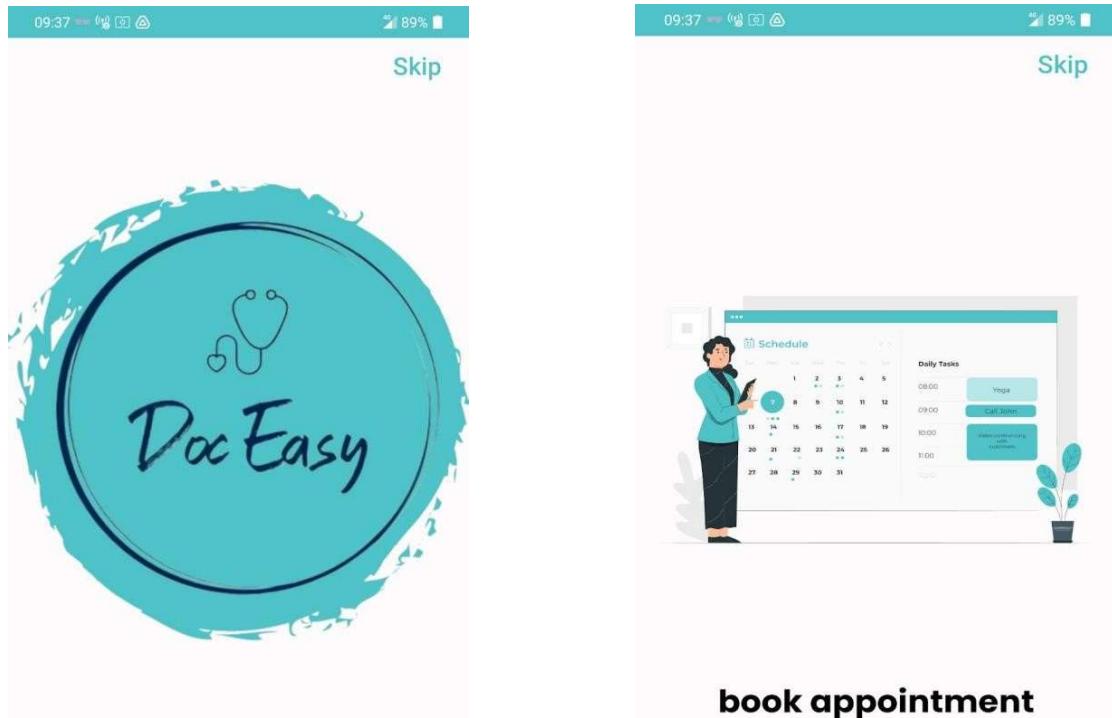


Figure 11.2.1.(Onboarding Screens (1))

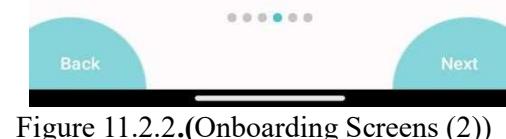


Figure 11.2.2.(Onboarding Screens (2))

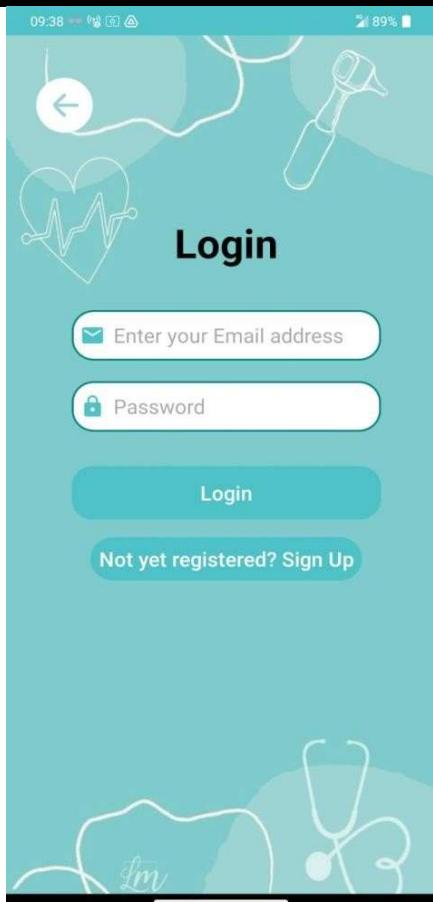


Figure 11.2.3.(receptionist login activity)

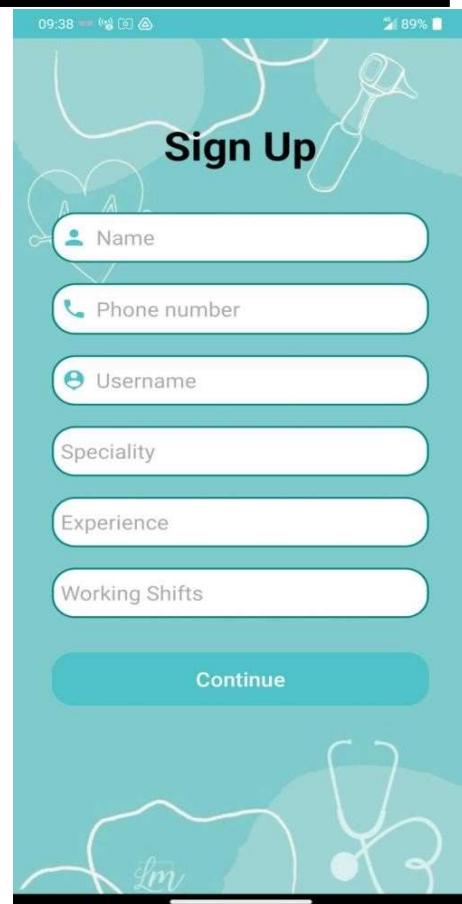
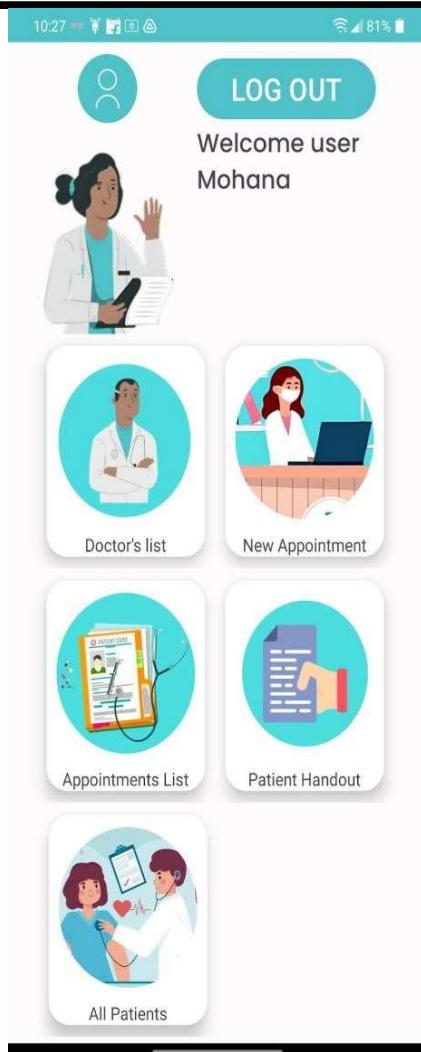


Figure11.2.4(receptionist logout activity)



11.2.5. (Receptionist Home Page)

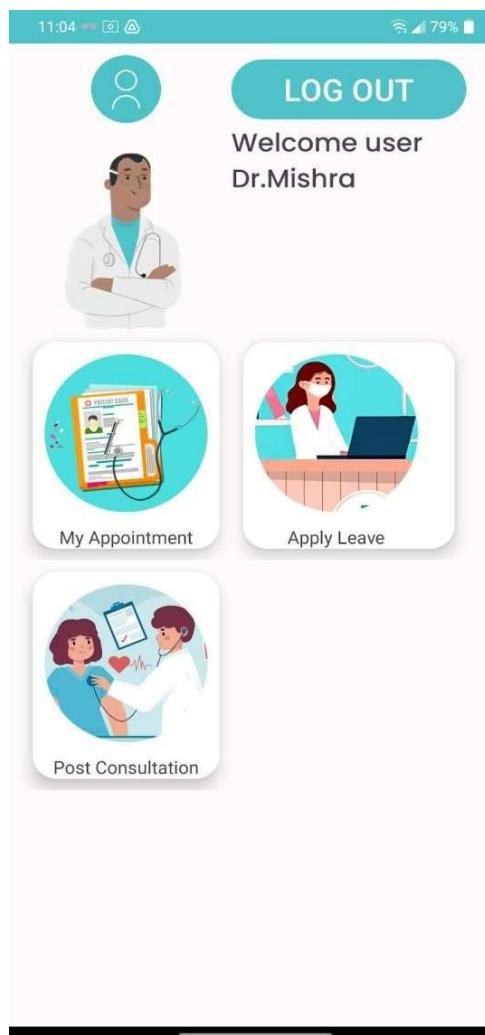


Figure 11.2.6. (Doctor Home Page)



Figure 11.2.7.(All Patients)

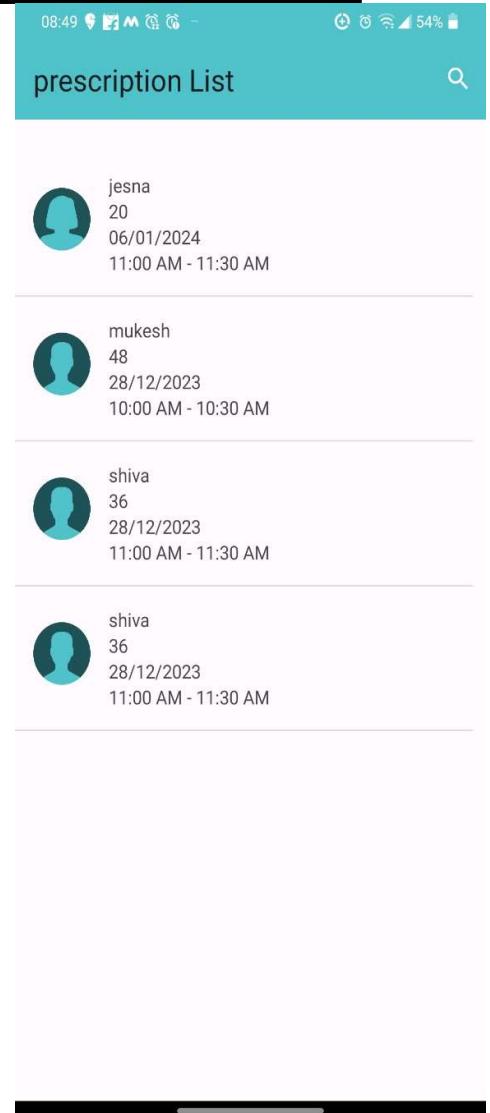


Figure 11.2.8. (Post Consultation List)



Figure 11.2.9. (All Doctors)



Figure 11.2.10. (Post Consultation List)

## **APPENDIX-C ENCLOSURES**

### **1.Acknowledgement of Journal paper submission.**

Dear Corresponding Author,

Thank you for submitting your paper entitled:

1. "A Mobile App for Doctors Appointment Scheduling"

for publication with International Journal of Advanced Computer Science and Applications (IJACSA) February 2024 Edition (Volume 15 No 2).

Your paper will be reviewed by the IJACSA technical committee and the evaluation outcome will be communicated up to 15 February 2024.

Regards,  
Editor  
IJACSA  
The Science and Information (SAI) Organization



### **2.Githublink [https://github.com/smythili-160/Doc\\_easy\\_1](https://github.com/smythili-160/Doc_easy_1)**

### **3. Similarity Index / Plagiarism Check report clearly showing the Percentage (20%).**

# DOCEASY- MOBILE APPLICATION FOR MANAGING DOCTOR APPOINTMENTS

## ORIGINALITY REPORT



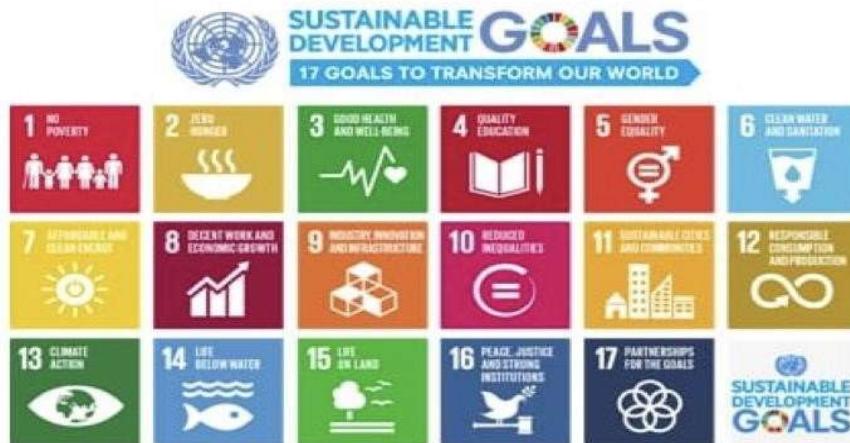
## PRIMARY SOURCES

1	<b>www.jait.us</b> Internet Source	<b>2%</b>
2	<b>Submitted to Asia Pacific University College of Technology and Innovation (UCTI)</b> Student Paper	<b>1%</b>
3	<b>Submitted to European University</b> Student Paper	<b>1%</b>
4	<b>Submitted to National Institute of Business Management Sri Lanka</b> Student Paper	<b>1%</b>
5	<b>core.ac.uk</b> Internet Source	<b>1%</b>
6	<b>Submitted to Presidency University</b> Student Paper	<b>1%</b>
7	<b>developers.google.cn</b> Internet Source	<b>1%</b>
8	<b>www.onlinesciencepublishing.com</b> Internet Source	<b>1%</b>

9	iarjset.com Internet Source	1 %
10	Submitted to Virginia Commonwealth University Student Paper	1 %
11	digscholarship.unco.edu Internet Source	1 %
12	Submitted to Eastern Institute of Technology Student Paper	1 %
13	Submitted to Laguna State Polytechnic University Student Paper	1 %
14	Submitted to University of Western Sydney Student Paper	<1 %
15	www.ijera.com Internet Source	<1 %
16	Submitted to Istanbul Aydin University Student Paper	<1 %
17	www.mdpi.com Internet Source	<1 %
18	www.studymode.com Internet Source	<1 %
19	Shounak Pal, Baidyanath Biswas, Rohit Gupta, Ajay Kumar, Shivam Gupta. "Exploring the factors that affect user experience in mobile-	<1 %

Exclude quotes Off  
 Exclude bibliography Off

Exclude matches Off



- DocEasy contributes to achieving **Sustainable Development Goal 3 - Good Health and Well-being** by enhancing healthcare efficiency and ensuring seamless patient care through digital innovation in appointment management.
- By reducing wait times, improving scheduling processes, and facilitating smooth transitions between healthcare professionals, DocEasy aligns with SDG 9 - Industry, Innovation, and Infrastructure, fostering technological advancements in the healthcare sector.
- The application's user-friendly interface promotes accessibility and inclusivity in healthcare services, aligning with SDG 10 - Reduced Inequality, as it ensures equitable access to quality medical care for patients and efficient work distribution among healthcare professionals.