A Project Report On

"Doctor Appointment App"

For

"Siddhanath Medical Services, Kolhapur"

Submitted By

JAY DATTATRAY YELE (18108)

SUBMITTED TO

SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

For the partial fulfillment of the internal credit work of

MASTERS OF COMPUTER APPLICATIONS SEM – III

UNDER THE GUIDANCE OF

DR. RUPALI PAWAR

Through

Zeal Education Society's

Zeal Institute of Business Administration, Computer Application & Research (ZIBACAR)

Sr. No. 39, Narhe, Pune -411041, Phone No.:67206031

(Approved by A.I.C.T.E., New Delhi, Recognized by DTE, Govt. Maharashtra & Affiliated to S.P.P.U. Pune)

2022-2023

ZEAL EDUCATION SOCIETY'S

ZEAL INSTITUTE OF BUSINESS ADMINISTRATION, COMPUTER APPLICATION AND RESEARCH (ZIBACAR)

NARHE | PUNE | INDIA

PUN CODE: IMMP013170 DTE CODE: 6152 AISHE CODE: C-41828

Certificate

This is to certify that the Mini project report entitled, "**Doctor Appointment App" for "Siddhanath Medical Services"** being submitted here for the internal work of the degree of **MASTER OF COMPUTER APPLICATIONS (SEM-III)** to Savitribai Phule Pune University, Pune is the result of the original project work completed by **Jay Dattatray Yele, 18108** under my supervision and guidance of **"Dr. Rupali Pawar"** and to the best of my knowledge and belief, the work embodies in this Project has not formed earlier the basis for the award of any Degree of similar title or any other University or examining body.

Place: Pune

Date: / /2023

Dr. Rupali Pawar	**Dr. Rupali Pawar**	**Dr. Rupali Kalekar**	**Prof. Pandurang Patil**
Project Guide	**Project Coordinator**	**HOD, MCA**	**Director**

Internal Examiner **External Examiner**

DECLARATION BY STUDENT

To,

The Director,

ZIBACAR, Pune

I undersigned hereby declare that this project titled "**Doctor Appointment App**" written and submitted by me to SSPU, Pune, in partial fulfillment of the requirement of the award of the degree of **MASTER OF COMPUTER APPLICATIONS (MCA-II) SEM-III** under the guidance of **Dr. Rupali Pawar**, is my original mini project work.

I further declare that to the best of my knowledge and belief, this project has not been submitted to this or any other University or Institute for the award of any Degree.

Place: Pune

Date: / /2023

Jay Yele (18108)

ACKNOWLEDGEMENT

I extend my sincere gratitude to Honorable **Shri. Sambhajirao Katkar**, President, ZES Pune, **Dr. Rishikesh Kakandikar**, Head, ZES Management Programmes, **Dr. Pandurang Patil**, Director of ZIBACAR, and **Dr. Rupali Pawar** for allowing me to carry out the study and for his constant encouragement, valuable suggestion and guidance during their search work.

I extend my special thanks to **Dr. Rajesh Kashyap, Dr. Babasaheb Mohite, Prof. Madhavi Shamkuwar, Prof. Kirti Samrit, Prof. Dharmendra Singh, Dr. Rupali Kalekar and Dr. Rupali Pawar** their kind cooperation and inspiration.

I extend my special gratitude to my dearest family members and friends who encouraged and motivated me to complete the project report.

Place: Pune

Date: / /2023

Jay Yele (18108)

INDEX

Chapter 1

Introduction

1. Introduction:

- **Course Name:** Master in Computer Application (Semester II).
- **Student Name**: Jay Dattatray Yele (MC21066).
- **Project Title:** Doctor Appointment App.
- **Name of Internal Guide:** Prof. Rupali Pawar.
- **Date Of Submission:** 28/10/2022

The proposed system is to make an online web application for easily taking appointment of a patient see the schedule of doctors, so that everyone can get information about doctor's availability, time period, and send request to any doctor for medicine. Doctors and patients can also easily communicate with each other from anywhere. This project is aimed at developing an online application for patient to appointing doctors. Users have to logging in the system to be able to take appointment of a doctor. Doctors have to logging to see his appointments. The proposed system could be accessed from any corner of the world on net.

1.1. Organization Profile:

- **Client Name:** Dr. Ramesh Suryawanshi.
- **Location & Address:** Sadashivnagar, Kagal, Kolhapur, 416235.
- **Contact Number:** 9373030413.
- **Email Id:** Drutkarsh0905@gmail.com
- **About Organization:** The Siddhanath Medical Services are situated in Sadashivnagar, Kagal, Kolhapur. They provide hospital services. The Siddhanath Medical Services have started in pandemic period. So many patients are visited daily. The Dr. Ramesh Suryawanshi Was started this medical services.

1.2. Project Abstract:

Online Doctor appointment app in hospital today necessitate a competent administration when handling patients, patient details which serves as a key factor for the flow of business transactions in Siddhanath Medical Services. Unfortunately, the current record management system leads to misplacement of during details, Patient details and doctor record of reports and insecurity to records. This research project is aimed at computerizing all the records about patients, hospital and doctors. In order to achieve this goal, a through system study and investigation was carried out and data was collected and analysed about the current system using document and data flow diagrams. Errors made on hand held calculators and dealt out completely. The method used to develop the system include iterative, logical and entity relationship diagram were used to design the system.

1.3. Existing System:

Under manual Online Doctor System, you have to first wait in line to take appointment for the doctors and wait for your time to have meet with them and discuss on your health problems. As you have to provide your information and other reports many times at different places such as the medicine store which is again a burden of carrying documents. You have to be present physically at the doctor's cabin. Patients have to visit on another day of after some hours to take their health reports which involves extra care person with patients anytime. Under manual system, the only accepted payment method is by cash and if patients due to some reasons are not having cash on time may face difficulties and not able to get treatment.

- In this system patient want to visit the hospital directly.
- Wait in line for register their self for taking appointment.
- Patient want to submit all details asking by receptionist.
- Then they give the appointment time and date to patient.
- Patient waits hours and hours in hospital.
- After that they discuss problem with doctor.
- All the system is manual.

1.4. Scope of System:

The Doctor Appointment Booking App include various modules as follows:

Patient Module:

The main objective of this module is providing all the functionality related to patient. It tracks all the information and details of the patient. We have developed all type of CRUD (Create, Read, Update and Delete) operations of the patient.

Doctor Module:

The main objective for developing this module is provide all the functionality related to doctor. It tracks all the information and details of the doctor. We have developed all type of CRUD (Create, Read, Update and Delete) operations of the doctor. This is a role-based module where admin can perform each and every operation on data but the doctor will be able to view only his/her data, so access level restrictions has also been implemented on the project.

Appointment Module:

The main aim for developing this module is to manage the doctor appointment. This Appointment Module is an important module in this project Doctor Appointment System which we have developed on PHP and MySQL. So, all appointment will be managed by admin and patient and doctor will be able to see the appointment.

1.5. Operating Environment:

Client-Side Hardware:

- RAM: 4GB
- Storage: 32 GB
- Operating System: Android 10+

Server-Side Hardware:

- RAM: 8GB
- CPU Speed: 5 GHZ
- Hard Disk: 1TB
- SSD: 128GB
- Operating System: Windows

Server-Side Technologies:

- **Frontend Technologies:**

1. Android
2. Java 18.0.1

- **Backend Technologies:**

1. Firebase

1.6. Brief Description of Technology Used:

Android Studio:

- Android Studio is an Integrated Development Environment (IDE) developed by Google for creating Android applications. It is the official IDE for Android app development and provides a comprehensive set of tools and features for building, testing, and deploying Android applications.
- Android Studio is built on top of the IntelliJ IDEA platform and includes a code editor, code analysis tools, debugging tools, an emulator, and various other features to make the app development process smoother and more efficient. It also supports various programming languages, including Java, Kotlin, and C++.
- Android Studio has a user-friendly interface and provides an easy-to-use drag-and-drop interface for designing app layouts. It also includes built-in templates for common app types, such as a basic activity, login activity, and more.

- Overall, Android Studio is a powerful and efficient tool for building Android apps, and it continues to be the go-to choice for developers looking to create high-quality Android applications.

Java:

- Java is a popular, general-purpose, high-level programming language that was first released by Sun Microsystems in 1995. It is an object-oriented language, meaning it is based on the concept of classes and objects, which provide a powerful and flexible way to organize and structure code.
- Java is known for its "write once, run anywhere" (WORA) approach, which means that code written in Java can run on any platform that has a Java Virtual Machine (JVM), without the need for platform-specific modifications.
- Java is widely used for building a variety of software applications, including web applications, mobile applications, desktop applications, and enterprise applications. It is also used for building server-side applications, such as web servers, application servers, and databases.
- Java has a large and active developer community, which has contributed to the development of various libraries, frameworks, and tools that make it easier to build Java applications. Some popular frameworks and tools for Java include Spring, Hibernate, Maven, and Eclipse.
- Overall, Java is a powerful and versatile programming language that continues to be widely used for building a wide range of software applications.

Firebase:

- Firebase is a mobile and web application development platform, owned by Google, that provides developers with a suite of tools and services to build and manage scalable and secure applications.
- Firebase offers a range of features, including real-time database, cloud storage, authentication, messaging, and analytics. The real-time database is a cloud-hosted NoSQL database that allows developers to store and synchronize data in real-time. The cloud storage allows developers to store and retrieve user-generated content, such as images, audio, and video. The authentication service

provides secure user authentication using email and password, phone number, or social media credentials.

- Firebase also includes a messaging service that enables developers to send notifications and messages to users, and an analytics service that helps developers track user engagement and measure the performance of their applications.
- Firebase is designed to be easy to use, with a simple and intuitive interface that makes it easy for developers to set up and manage their applications. It also provides extensive documentation, tutorials, and a supportive community to help developers get started quickly and easily.
- Overall, Firebase is a powerful platform for mobile and web application development, providing developers with a wide range of features and tools to build and manage scalable and secure applications.

1.6.1. Operating System Used:

Windows OS:

- Windows is a popular operating system (OS) developed by Microsoft. It is used on desktops, laptops, tablets, and other devices, and is known for its user-friendly interface and wide range of features.
- Windows is designed to be easy to use, with a graphical user interface (GUI) that allows users to interact with their computer using a mouse, keyboard, or touch screen. It also includes a wide range of built-in applications, such as a web browser, email client, media player, and more.
- One of the key features of Windows is its ability to support a wide range of hardware and software. This makes it easy for users to customize their computers to meet their specific needs, whether that's for gaming, video editing, or other specialized tasks.
- Windows also includes a range of security features, such as built-in antivirus software, firewall protection, and encryption tools, to help protect users from online threats and keep their data safe.
- Overall, Windows is a versatile and user-friendly operating system that is widely used around the world, and continues to be a popular choice for both

personal and business use.

Android OS:

- Android OS is a popular mobile operating system (OS) developed by Google. It is used on a wide range of devices, including smartphones, tablets, and smartwatches.
- Android OS is known for its flexibility and customization options, allowing users to customize the look and feel of their device with themes, wallpapers, and widgets. It also supports a wide range of apps, including popular social media, gaming, and productivity apps.
- One of the key features of Android OS is its open-source nature, which means that developers can easily modify and customize the code to create their own custom versions of the OS. This has led to the development of many custom ROMs, which provide users with additional features and customization options.
- Android OS also includes a range of security features, such as built-in malware protection, app permissions, and encryption tools, to help protect users from online threats and keep their data safe.
- Overall, Android OS is a flexible and customizable mobile operating system that is widely used around the world and continues to evolve with new features and updates.

1.6.2. NoSQL Use to Build Database:

- NoSQL database refers to a non-relational database that provides a flexible and scalable alternative to traditional relational databases. Unlike traditional databases, which store data in tables with a fixed schema, NoSQL databases store data in various ways, such as key-value pairs, documents, and graphs.
- NoSQL databases are designed to be highly scalable and can handle large amounts of data and high traffic volumes. They are often used for big data and real-time web applications that require rapid data processing and storage.
- NoSQL databases provide a flexible data model that can be easily adapted to meet changing business requirements. This makes it easy to add or modify data

fields without requiring changes to the entire database schema.

- NoSQL databases also support distributed data storage, which means that data can be stored across multiple servers, improving performance and availability. They are also highly fault-tolerant, with built-in replication and backup mechanisms to ensure that data is not lost in the event of a hardware or software failure.
- Overall, NoSQL databases are a powerful and flexible alternative to traditional relational databases, providing a scalable and fault-tolerant solution for big data and real-time applications.

Chapter 2

Proposed System

2. Proposed System:

The proposed project is a smart appointment booking system that provides patients or any user an easy way of booking a doctor's appointment online. This is android application that overcomes the issue of managing and booking appointments according to user's choice or demands. The task sometimes becomes very tedious for the compounder or doctor himself in manually allotting appointments for the users as per their availability. Hence this project offers an effective solution where users can view various booking slots available and select the preferred date and time. The already booked space will be marked yellow and will not be available for anyone else for the specified time. This system also allows users to cancel their booking anytime.

- The system will be design for the patient who want to fix appointment with doctor.
- This app helps the doctor to accept the appointment as per there schedule.
- First Patient want to submit their all details for registration.
- Then they have login into the system.
- After that they select the doctor for treatment.
- Then doctor or staff check details and confirm their appointment.
- All process will do by any place in online mode.

2.1 Study of Similar System:

- To study similar systems for doctor appointment apps, there are several popular apps and systems you can explore:
- Zocdoc: Zocdoc is a popular doctor appointment app that allows users to search for doctors, book appointments, and receive reminders. It also provides features like insurance verification, patient reviews, and the online check-in.
- Practo: Practo is another doctor appointment app that allows users to search for doctors, book appointments, and receive reminders. It also provides features

like online consultations, medicine delivery, and health records management.

- HealthTap: HealthTap is an app that connects users with doctors for online consultations and second opinions. It provides features like video consultations, symptom checkers, and personalized health recommendations.
- Doctolib: Doctolib is a popular doctor appointment system used in Europe. It allows users to search for doctors, book appointments, and receive reminders. It also provides features like online booking for group practices, waiting list management, and medical file sharing.
- Epic Systems: Epic Systems is a healthcare software provider that offers a range of tools for healthcare organizations, including appointment scheduling, patient records management, and billing.
- By studying these similar systems, you can gain insights into the features and functionality that are important for a doctor appointment app, as well as the design and user experience best practices.

2.2. Feasibility Study:

A feasibility study for a doctor appointment app would typically include the following key components:

- **Market analysis:** The first step is to determine the demand for such an app in the target market. This would involve researching the current state of the healthcare industry, identifying existing players in the market, and assessing the level of competition. You would also need to identify the target audience and their needs and preferences.
- **Technical analysis:** The next step is to assess the feasibility of developing the app from a technical perspective. This would involve identifying the required technology and infrastructure, assessing the skills and expertise of the development team, and estimating the development time and cost.
- **Financial analysis:** The financial feasibility of the app would involve estimating the costs involved in developing and launching the app, as well as the potential revenue streams. You would also need to consider the ongoing costs of maintaining and updating the app, and estimate the break-even point

and potential return on investment (ROI).

- **Legal analysis:** You would need to assess the legal and regulatory requirements for launching a doctor appointment app in the target market. This would involve identifying any legal or regulatory restrictions or requirements related to data privacy, security, and healthcare laws.
- **Risk analysis:** Finally, you would need to conduct a risk analysis to identify any potential risks or obstacles that could impact the success of the app. This would involve assessing the potential impact of risks such as technical issues, market competition, and legal and regulatory compliance.

By conducting a thorough feasibility study, you can gain a clear understanding of the potential opportunities and challenges involved in launching a doctor appointment app, and make informed decisions about whether or not to proceed with the development and launch of the app.

2.3. Objectives of Proposed System:

The system aims to help the patients to take appointment online through internet and track their records through it. Existing system has been facing problems due to its paper-based appointment system. The increase in the number of patients visiting, it has become difficult to manage the appointment system manually. The purpose of this project is to solve these complications by creating custom-built database software to manage the appointment system. For the receptionist it makes easy to set date and time for the treatment of the patient to the relevant doctor. Doctor enters medical prescription and receptionist takes the print. It also helps to maintain doctor’s consultation fee automatically.

- To make a way of easy appointment booking.
- To provide immediate service.
- To reduce time of appointment booking.
- To improve hospital system faster and easy.
- To book appointment from any location.
- To interact with doctor from any location.

2.4. Users of System:

In this Doctor Appointment App there are two users Doctor and Patient.

Patient:

The main objective of this module is providing all the functionality related to patient. It tracks all the information and details of the patient. We have developed all type of CRUD (Create, Read, Update and Delete) operations of the patient.

Doctor:

The main objective for developing this module is provide all the functionality related to doctor. It tracks all the information and details of the doctor. We have developed all type of CRUD (Create, Read, Update and Delete) operations of the doctor. This is a role-based module where admin can perform each and every operation on data but the doctor will be able to view only his/her data, so access level restrictions has also been implemented on the project.

Chapter 3

Analysis And Design

3.1. System Requirements (Functional & Non-Functional Requirements):

3.1.1 Functional Requirements:

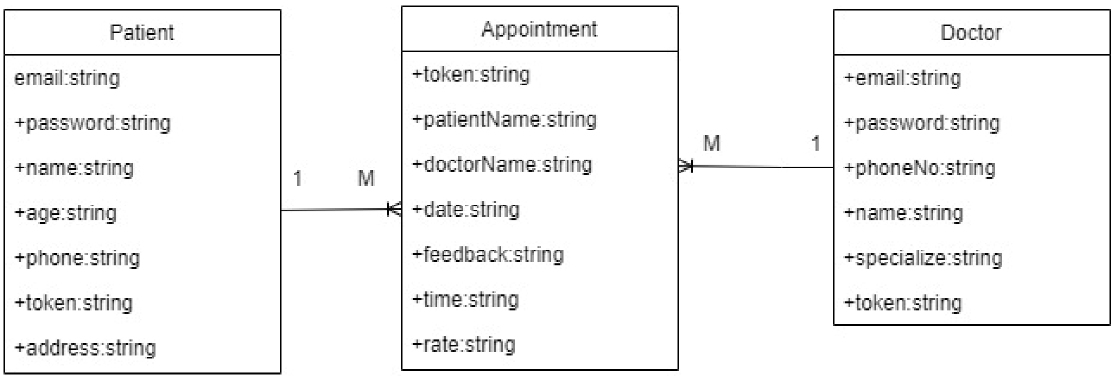
- **Patient Login:**
  Patient authentication is required to access the web application.
- **Patient Registration:**
  Patient needs to register with the details like email, name, PhoneNo, address etc.
- **Authorization:**
  User has different roles to access the application. There are 3 roles to access the application viz., patient, doctor, admin.
- **Location Access:**
  Application needs to access the GPS location of the user. User needs to allow the location access to application.
- **Book Appointment:**
  Patient is able to book the appointment with appropriate data.
- **Reports:**
  Admin can access all the records of appointments.

3.1.2 **Non-Functional Requirements:**

- **Security:**
  Each member is required to have an individual password Administrators have the option of increasing the level of password security their members must use. For maximum security, each member must protect their password.
- **Reliability:**
  System will prompt the user if any incorrect input is made. To handle data consistency, DBMS software is used. User can easily work through the different menus and buttons.

- **Maintainability:**
  Proper documentation is available for further upgradation and maintenance. User will be trained enough to handle the minor changes required.
- **Availability:**
  The system shall be available all the time.
- **Portability:**
  System is independent of hardware specification. It can run on any operating system. System is independent of browser compatibility.
- **Performance:**
  The performance of our product is at its best if stored locally, as the response time will be much faster. If the product accessed via Internet, the performance is limited by the connection speed. The only foreseen limitation is that of web server response.
- **User Friendly:**
  Our system is very easy to use and user friendly. Its GUI is very attractive and understandable to the common users.

3.2. Data Model:



3.3. Table Structure:

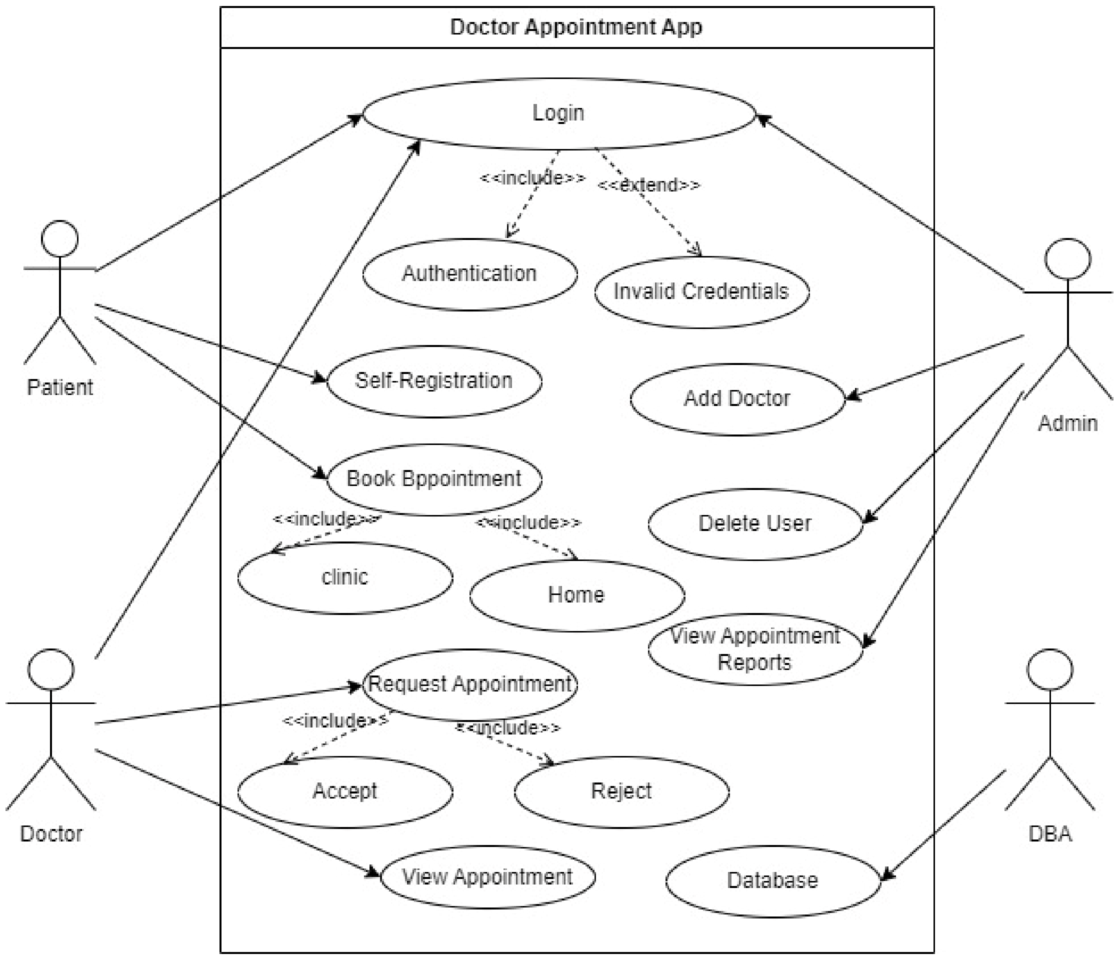
Data Dictionary:

Sr. No	Field Name	Data Type	Description
1	address	String	Address of patient
2	age	String	Age of patient
3	date	String	Appointment date
4	doctorEmail	String	Doctor email id
5	doctorName	String	Name of doctor
6	doctorSpecialize	String	Doctors' specialization
7	email	String	Email id for login users
8	feedback	String	Feedback by patient
9	meetingLocation	String	Location home or clinic
10	name	String	Name of user
11	password	String	Users' password
12	patientAddress	String	Patient local address
13	patientAge	String	Age of patient
14	patientEmail	String	Patient email
15	patientName	String	Patient name
16	patientPhone	String	Patient contact details

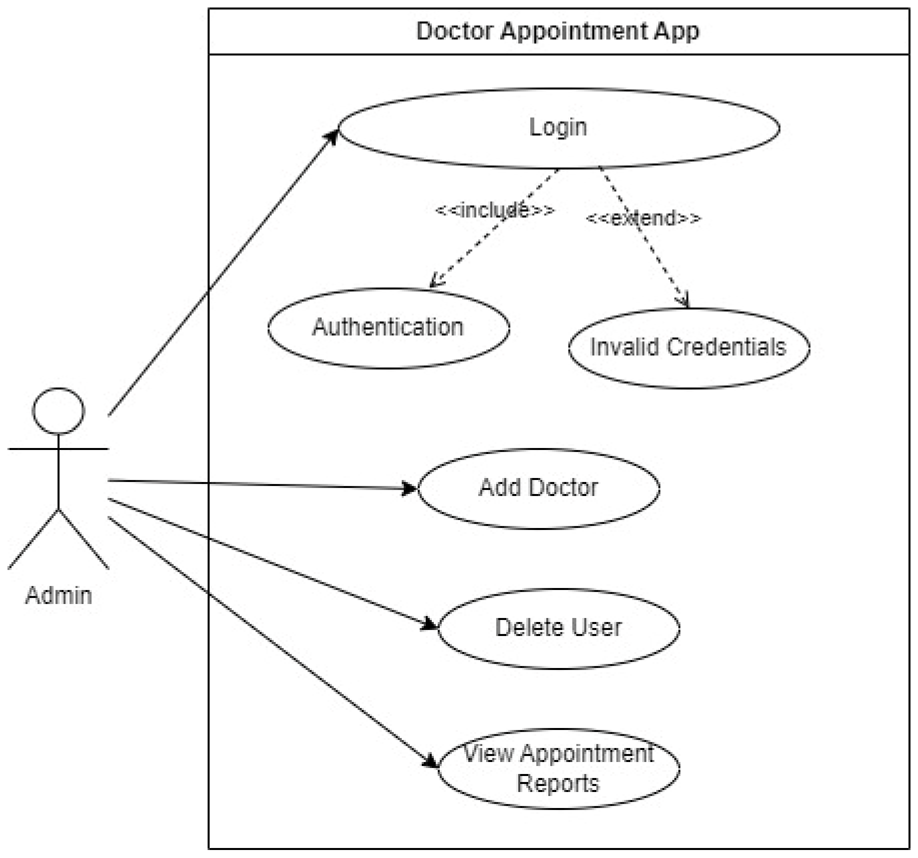
17	phone	String	Phone no for contact
18	phoneNo	String	Phone no in appointment records
19	rate	String	Ratings given by patient
20	specialize	String	Doctor specialization
21	time	String	Appointment time
22	token	String	Appointment token

3.4. Use Case Diagram:

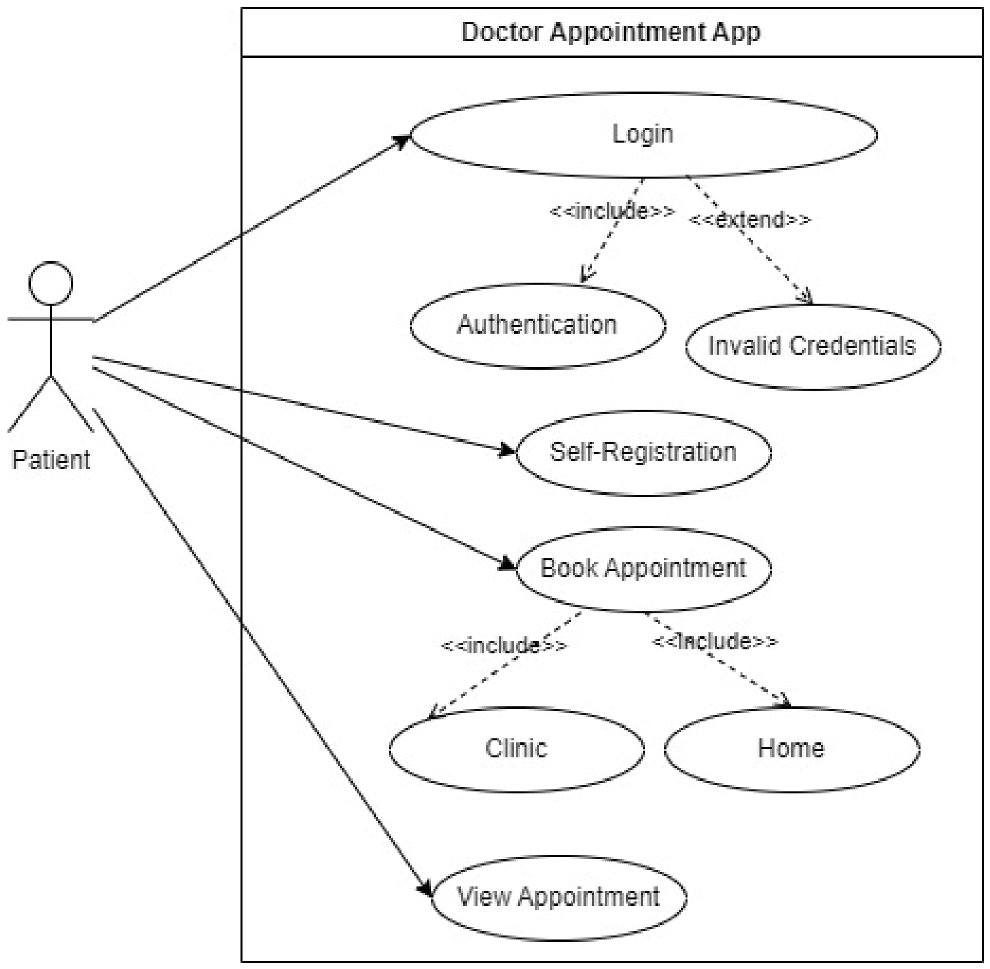
3.4.1. Global Use Case Diagram:



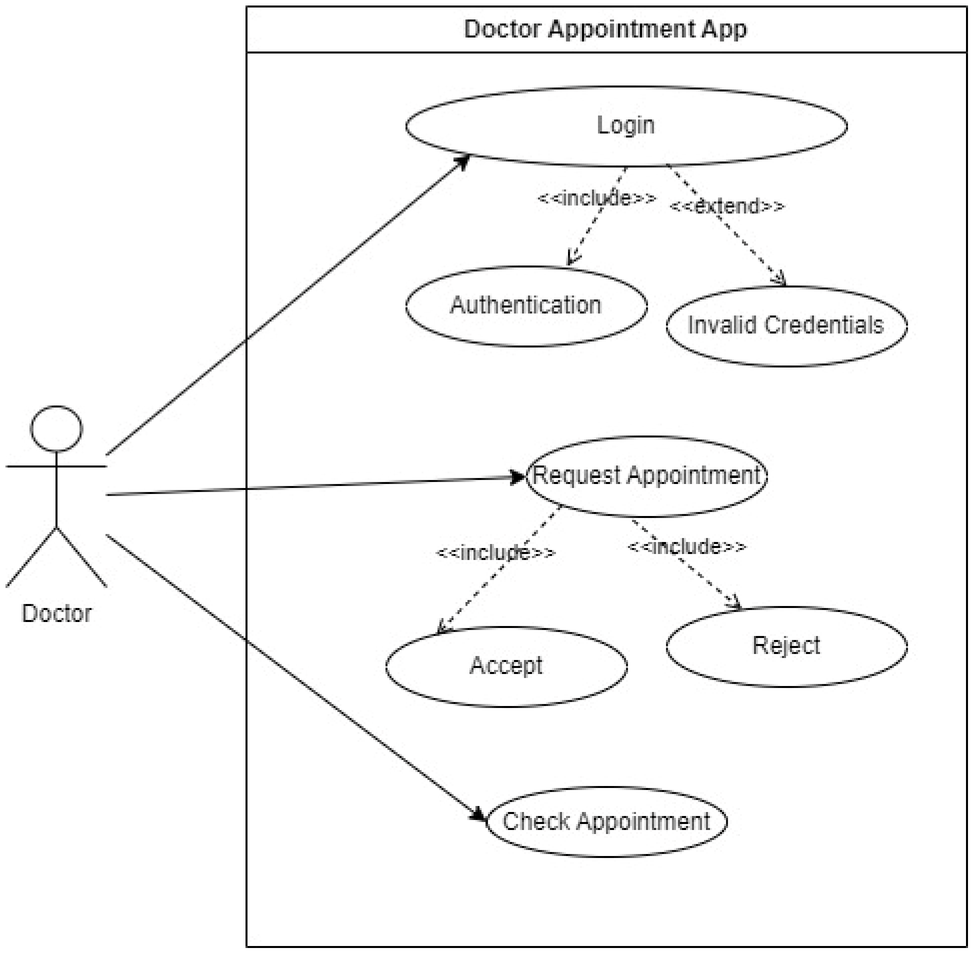
3.4.2. Admin Use Case Diagram:



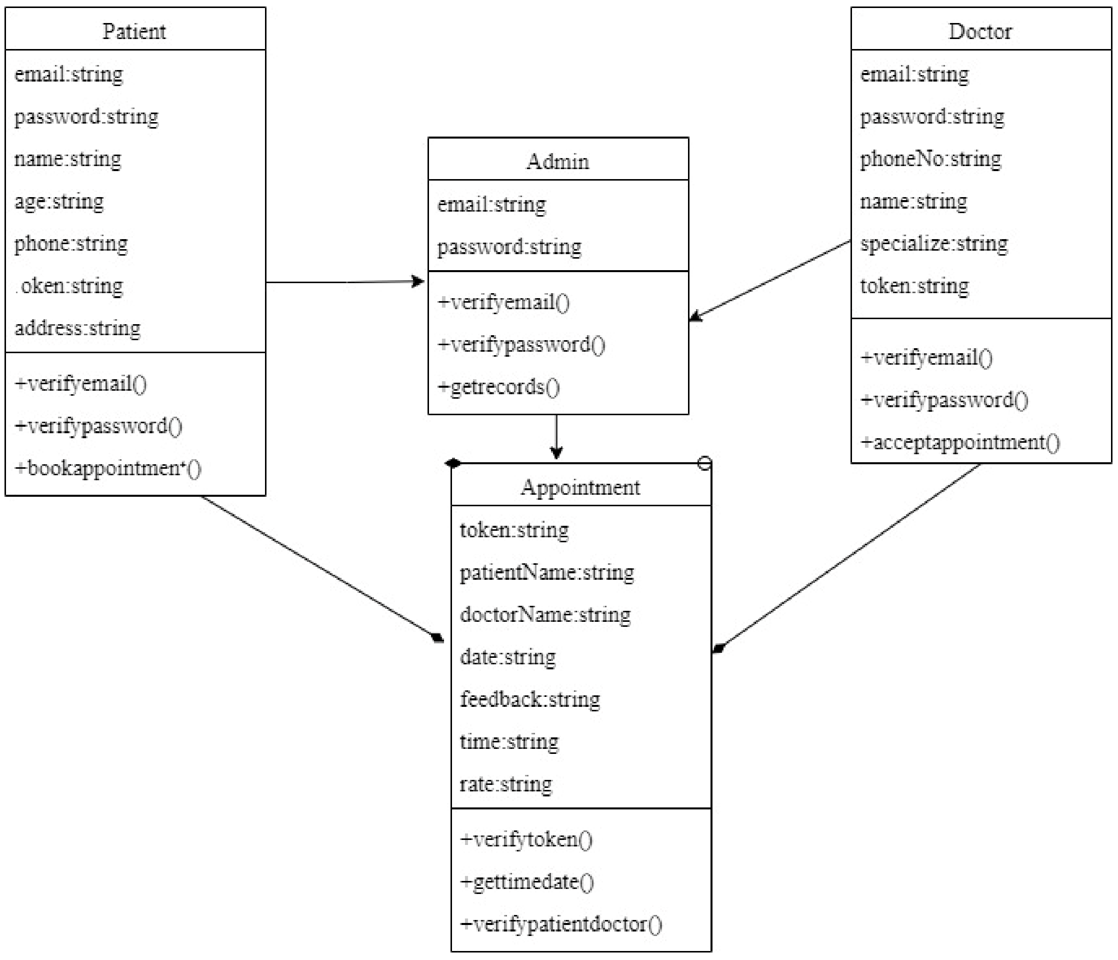
3.4.3. Patient Use Case Diagram:



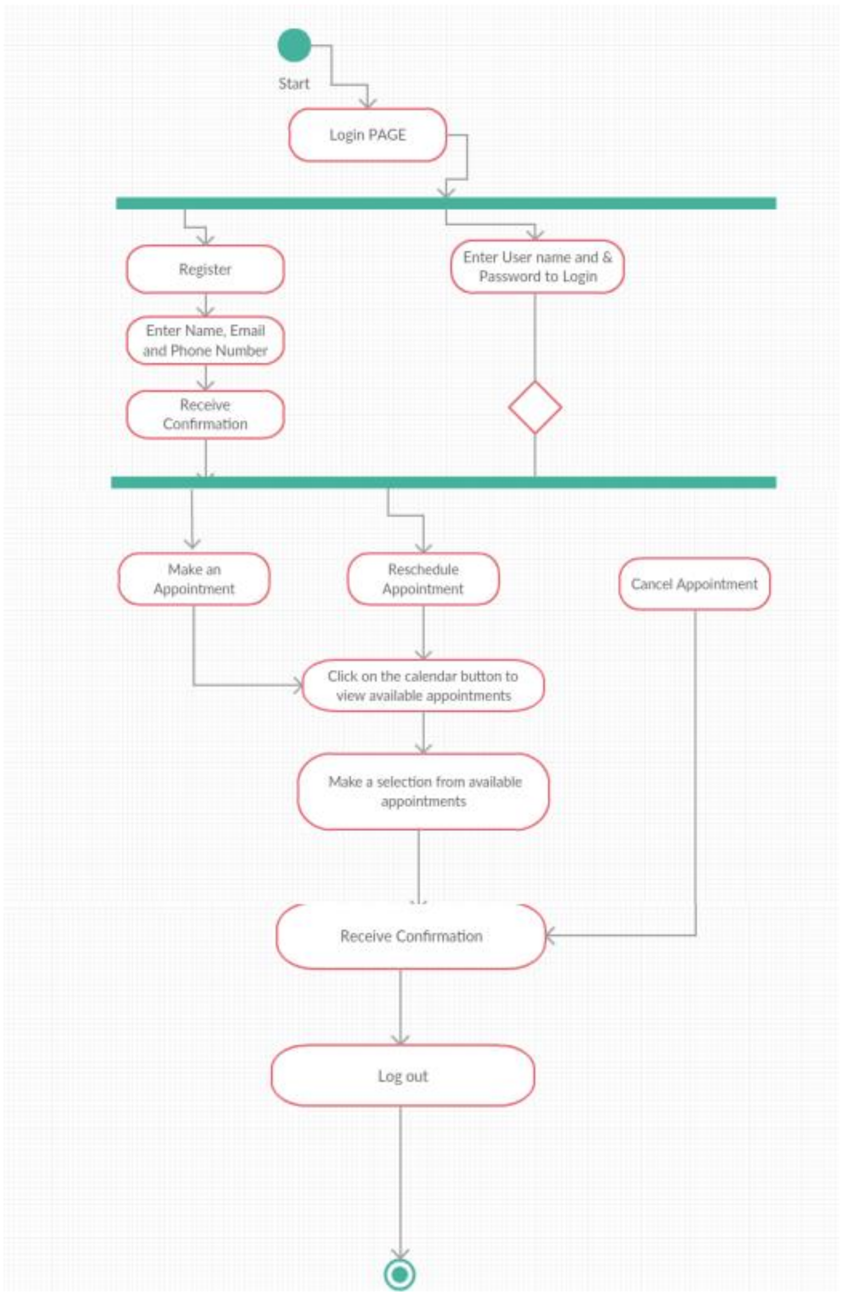
3.4.4. Doctor Use Case Diagram:



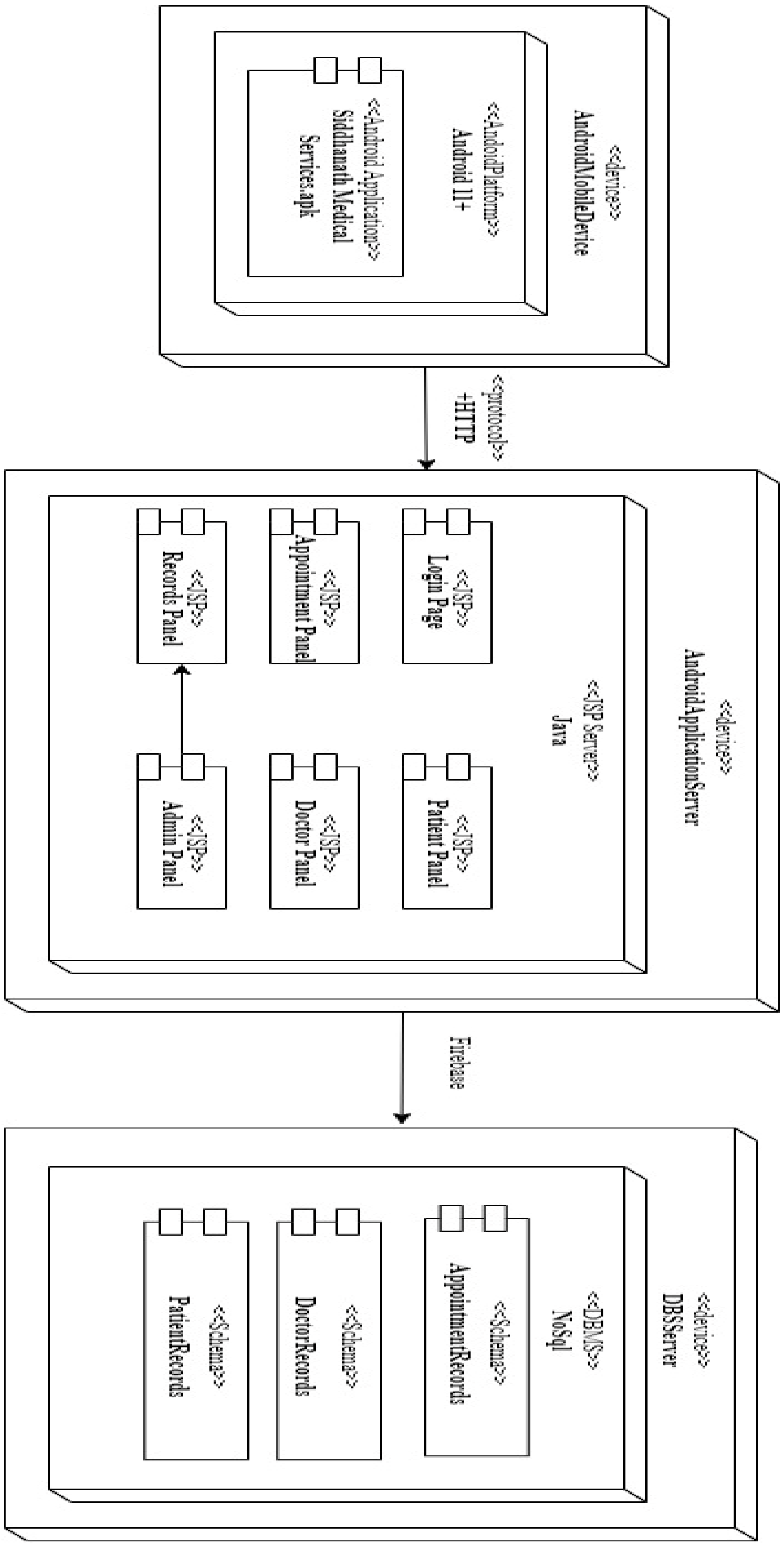
3.5. Class Diagram:



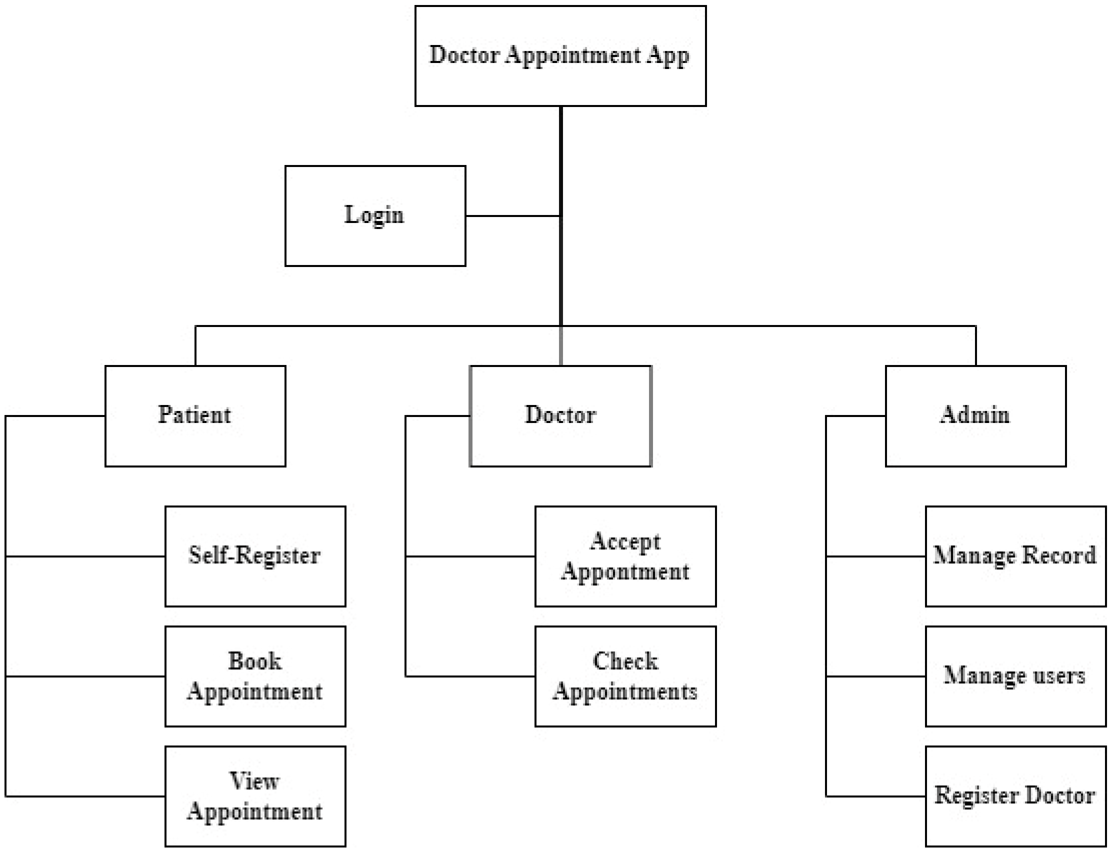
3.6. Activity Diagram:


Start
Login PAGE
Register
Enter User name and & Password to Login
Enter Name, Email and Phone Number
Receive Confirmation
Make an Appointment
Reschedule Appointment
Cancel Appointment
Click on the calendar button to view available appointments
Make a selection from available appointments
Receive Confirmation
Log out

3.7. Deployment Diagram:



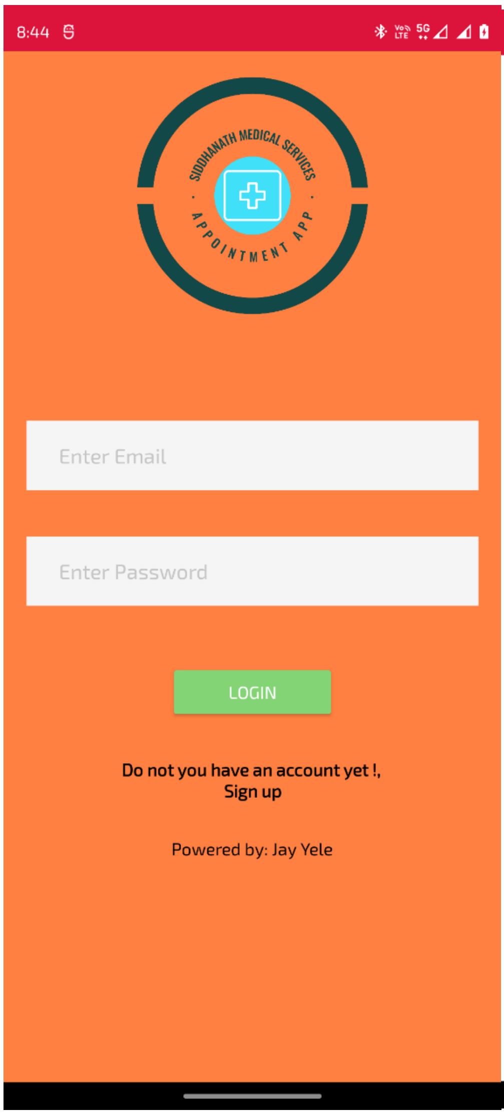
3.8. Module Hierarchy Diagram:

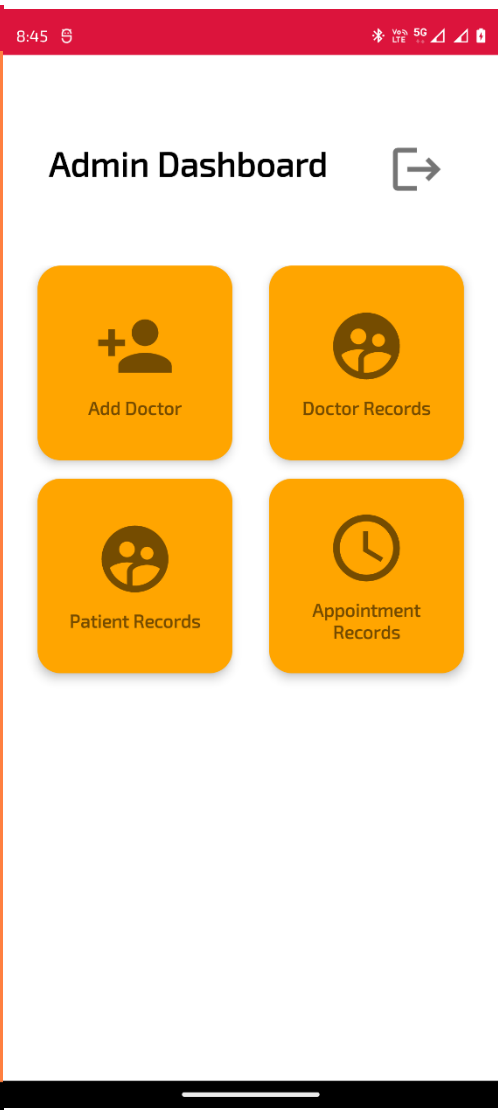


3.9 User Manual/ Sample Input and Output Screens:

Admin Panel:

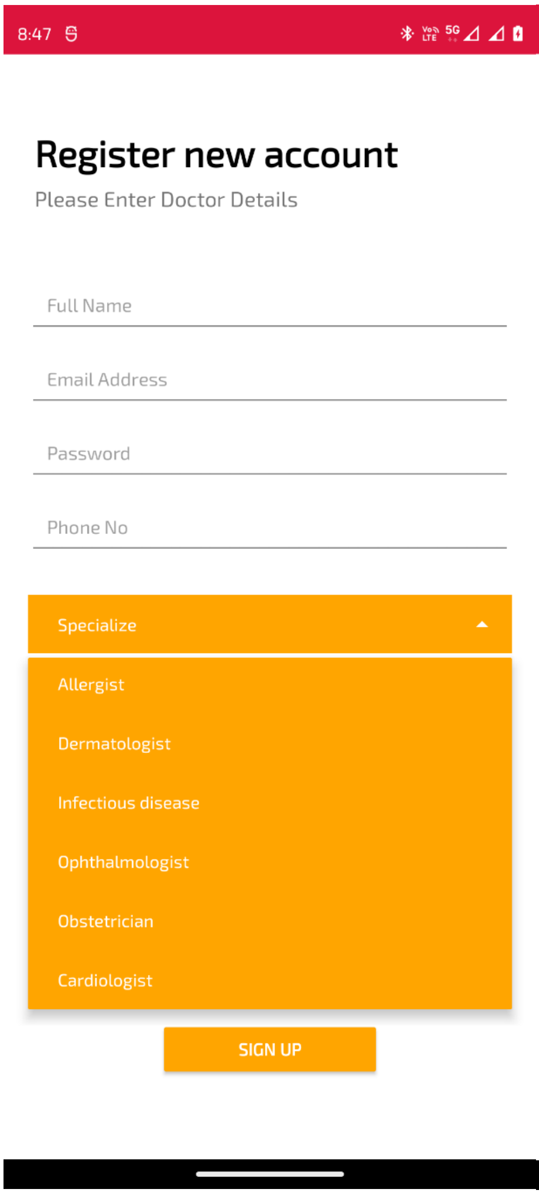
Admin login and admin dashboard.

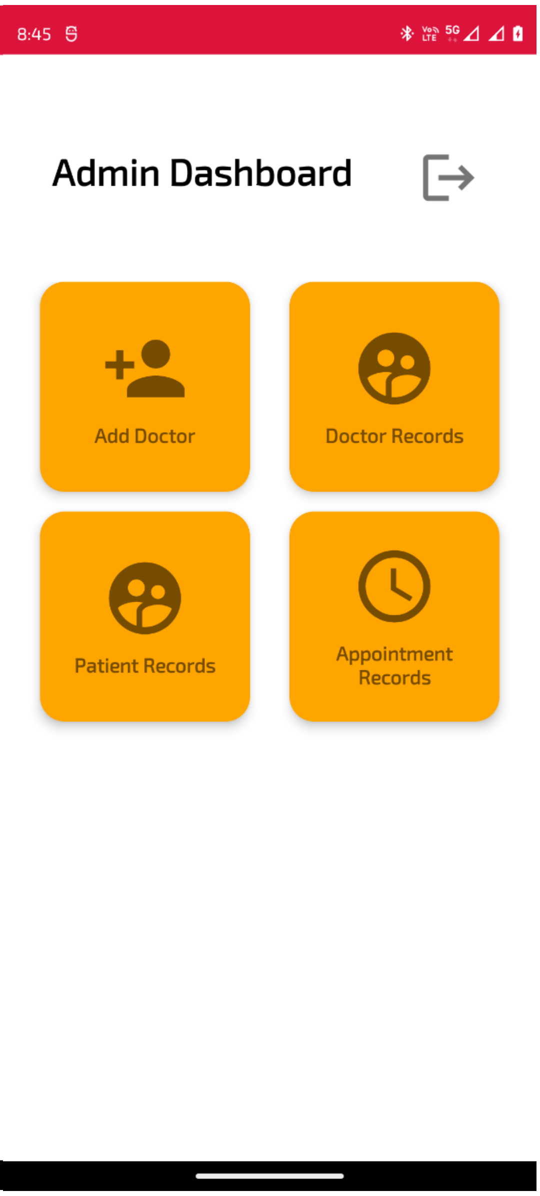




Admin Register Doctor:

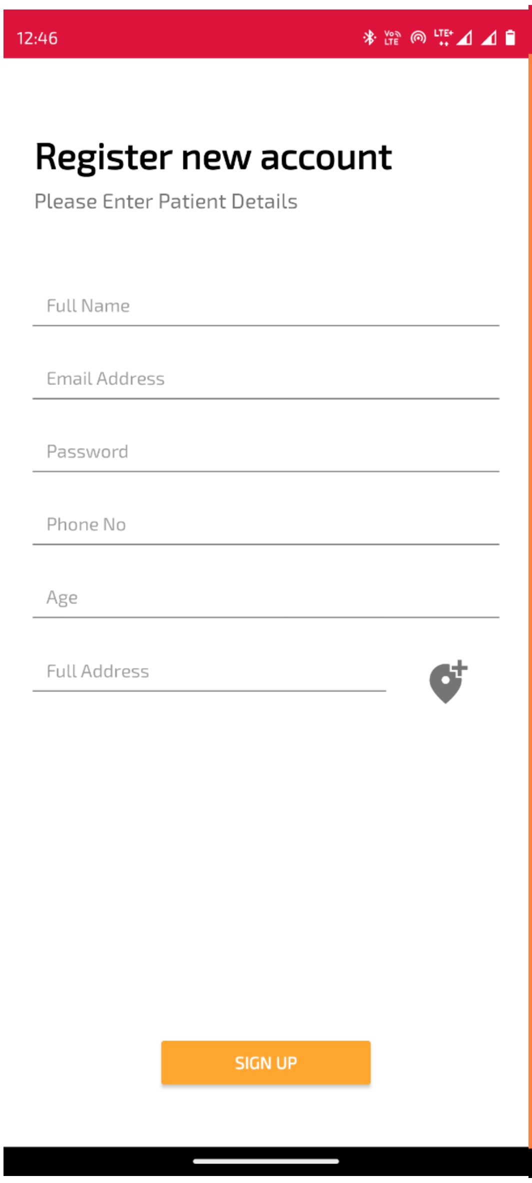
Admin add doctor successfully in system.

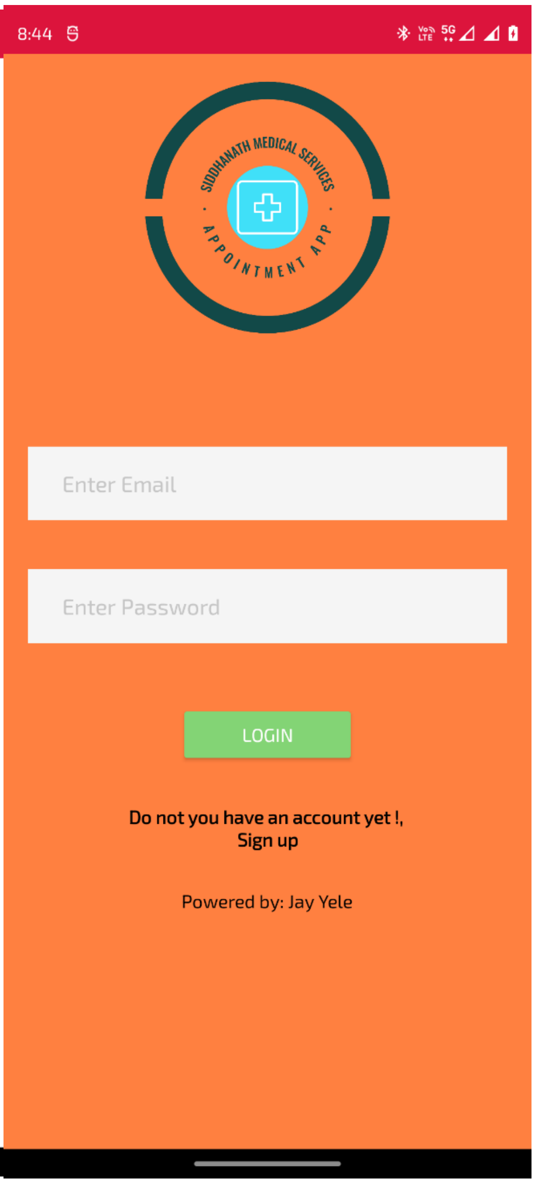




Patient Registration:

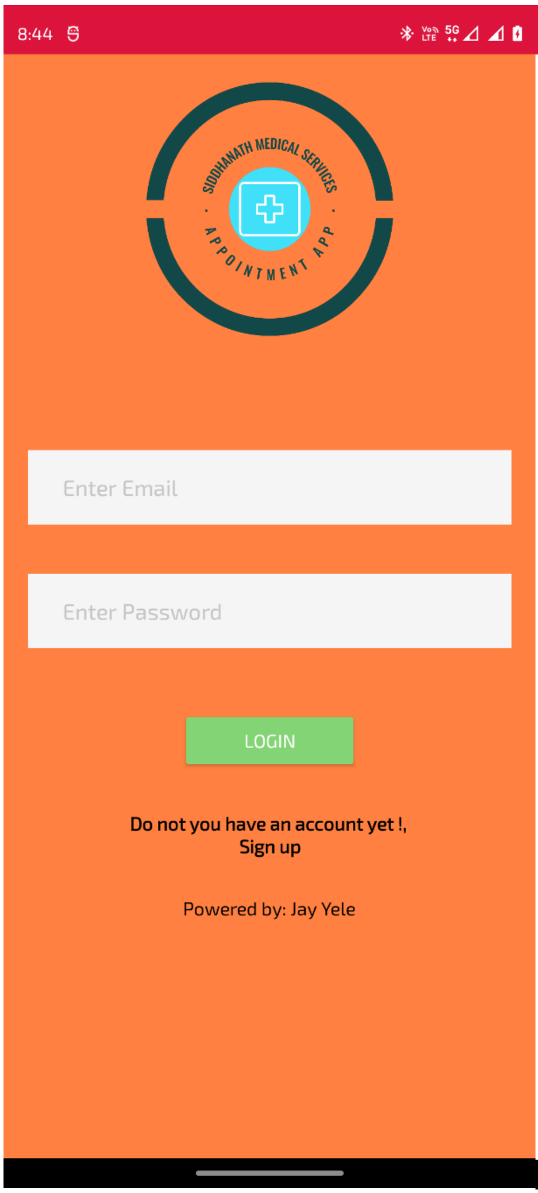
The patient self-register successfully.

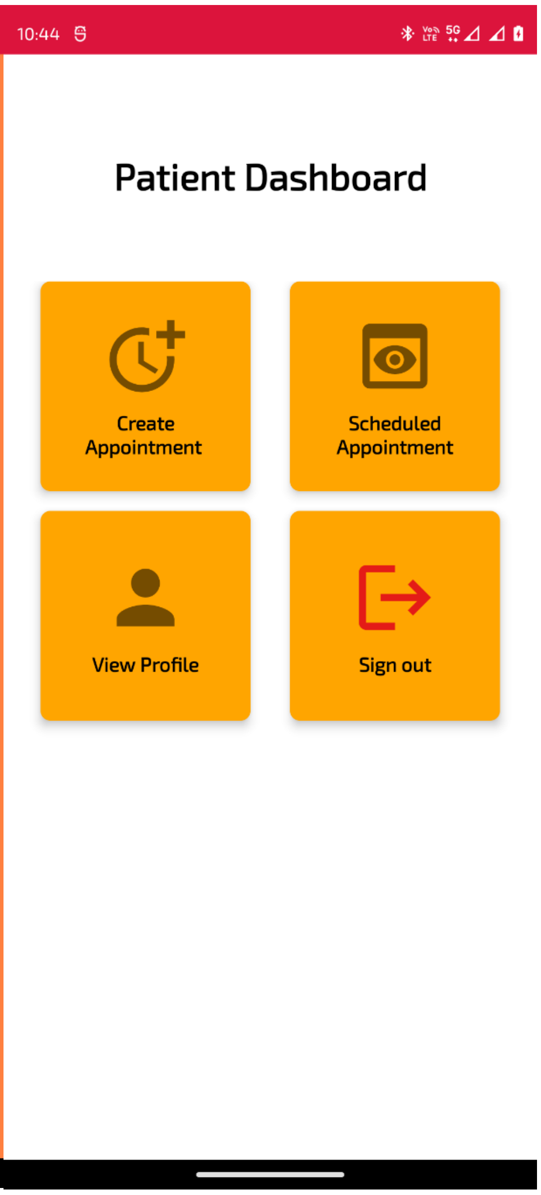




Patient Login and Dashboard:

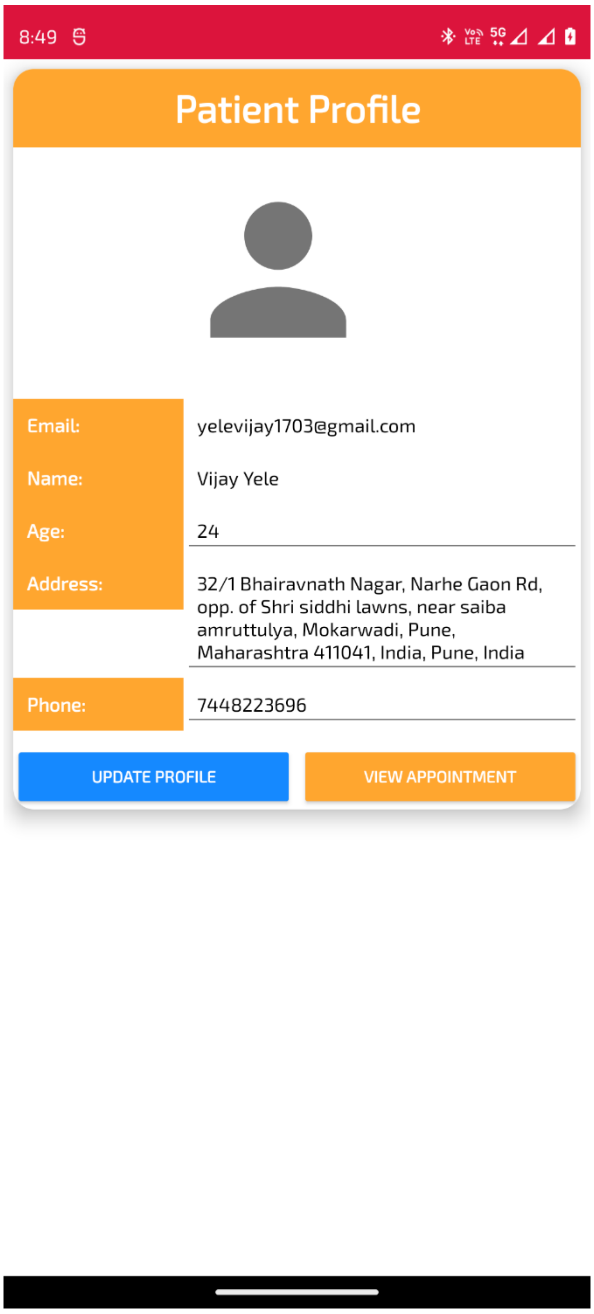
Patient Login and enter in the system.





Patient Profile:

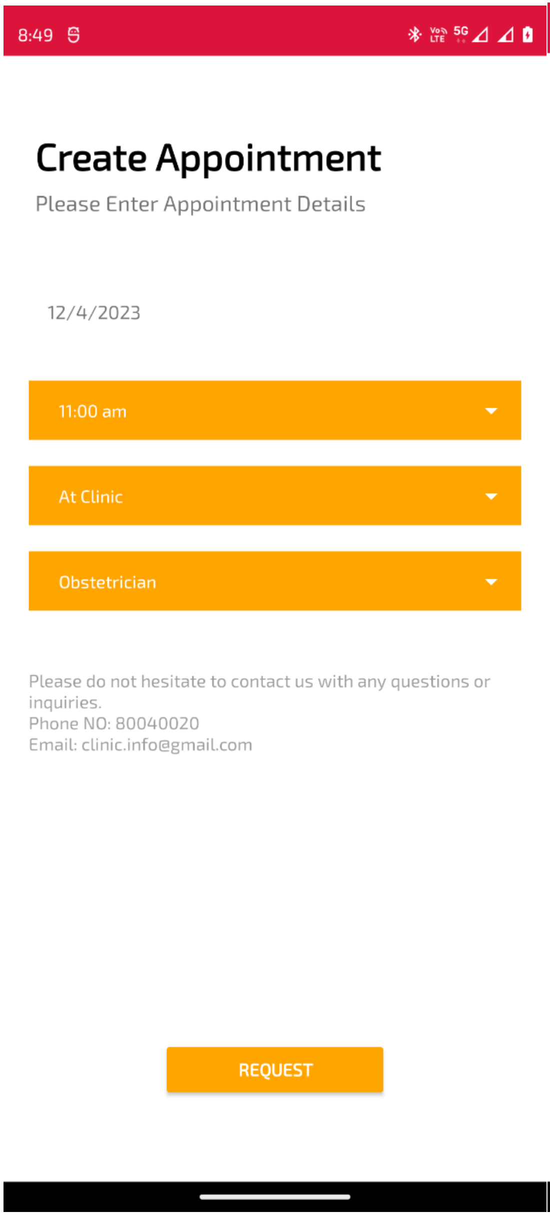
Patient Profile show patient details.



Email:	yelevijay1703@gmail.com
Name:	Vijay Yele
Age:	24
Address:	32/1 Bhairavnath Nagar, Narhe Gaon Rd, opp. of Shri siddhi lawns, near saiba amruttulya, Mokarwadi, Pune, Maharashtra 411041, India, Pune, India
Phone:	7448223696

Patient creates an appointment:

The patient booked appointment successfully.



Create Appointment

Please Enter Appointment Details

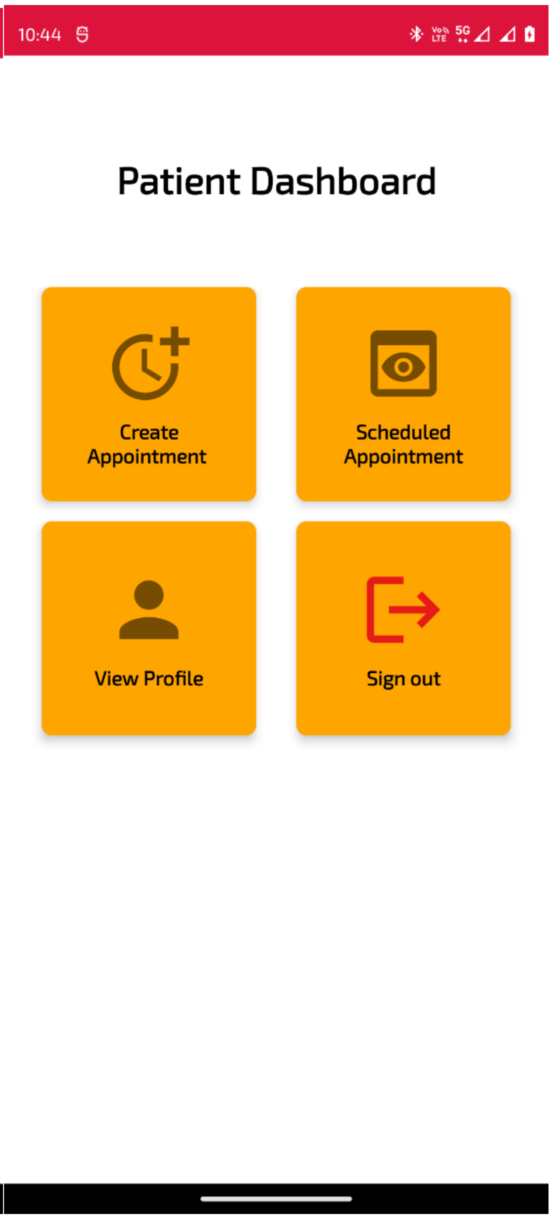
12/4/2023

11:00 am

At Clinic

Obstetrician

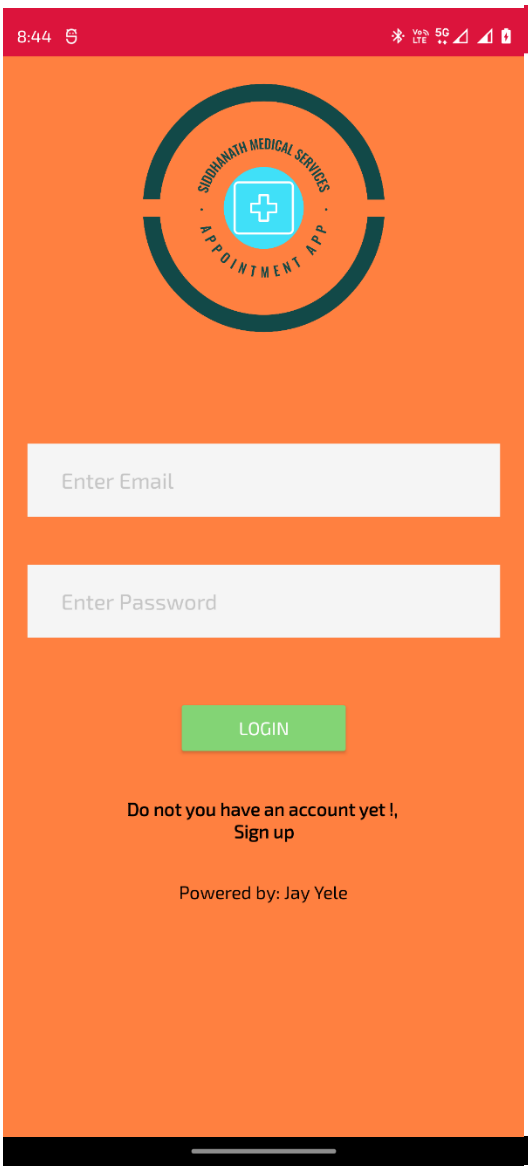
Please do not hesitate to contact us with any questions or inquiries.
Phone NO: 80040020
Email: clinic.info@gmail.com

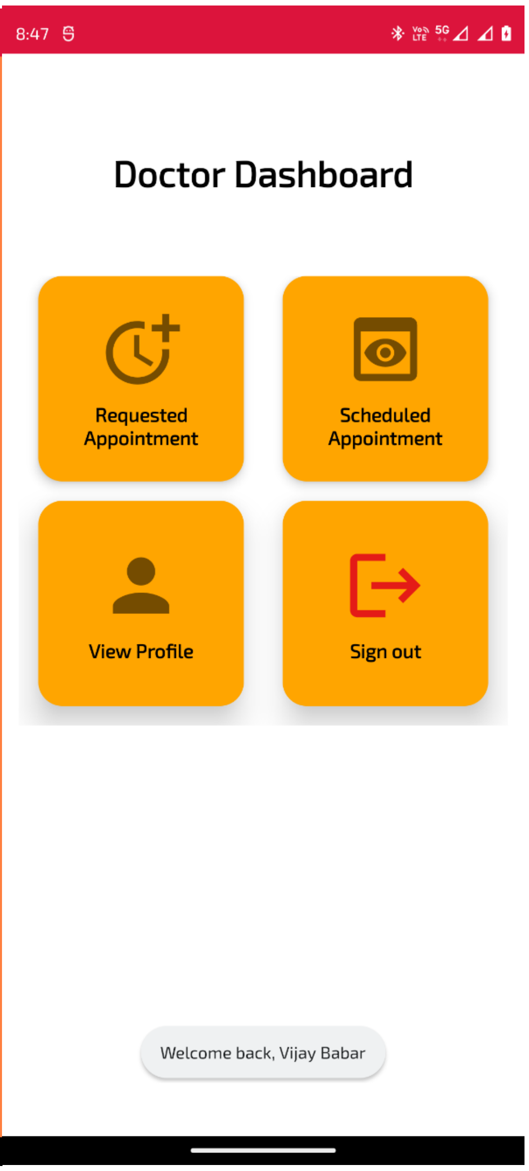


Patient Dashboard

Doctor Login and Dashboard:

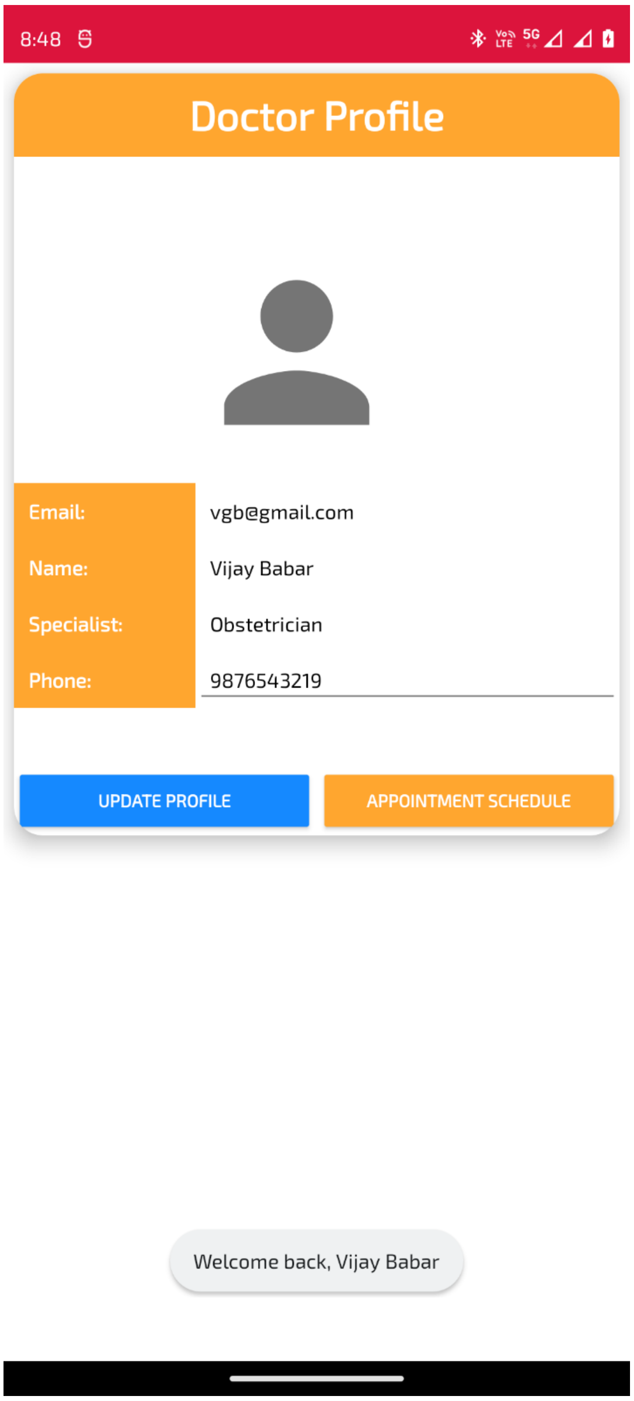
Doctor login successfully in the system.





Doctor Profile:

Doctor profile show doctor details.

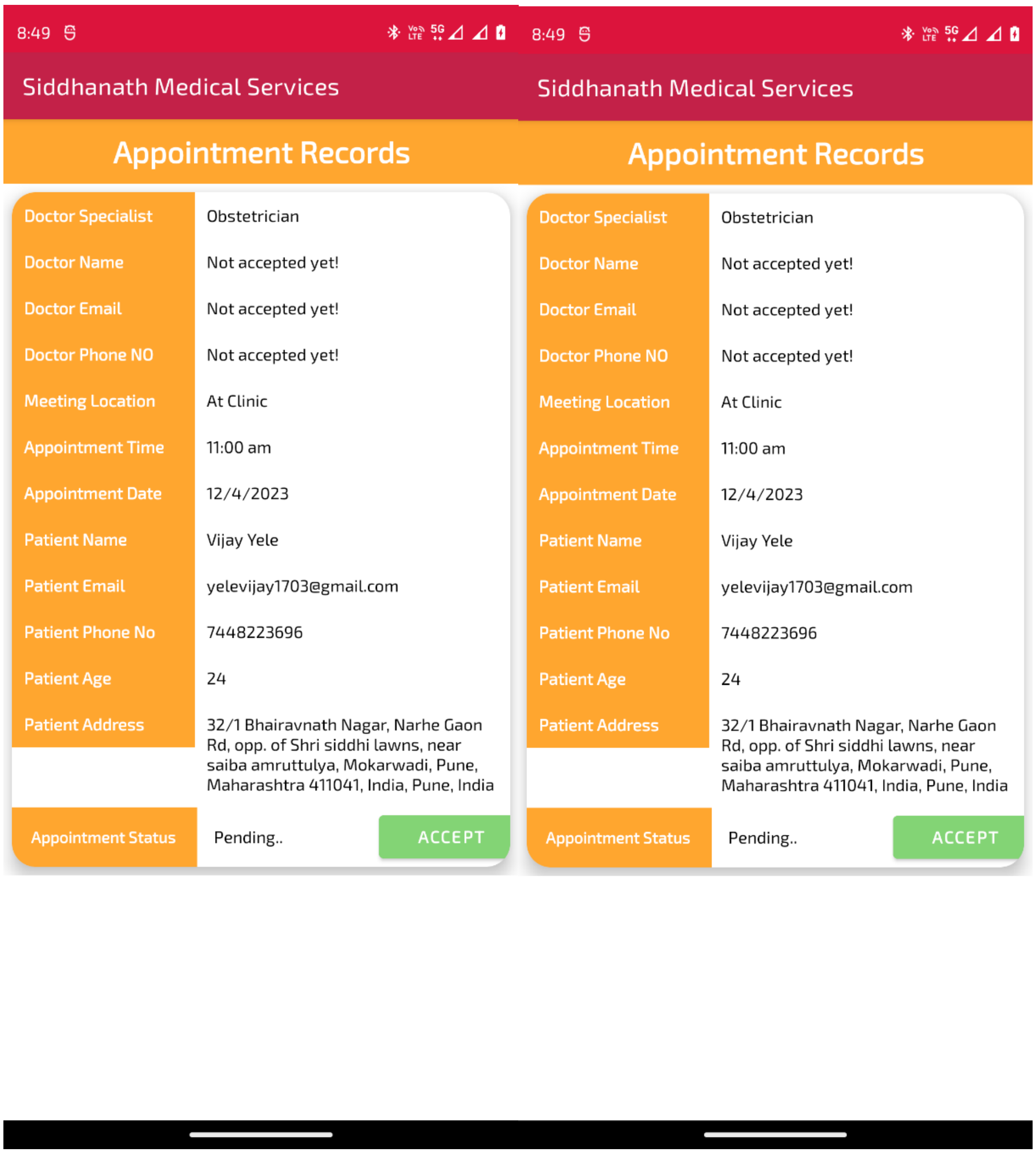


Email:	vgb@gmail.com
Name:	Vijay Babar
Specialist:	Obstetrician
Phone:	9876543219

Welcome back, Vijay Babar

Requested Appointment:

The doctor accepts the appointment successfully.



8:49

Siddhanath Medical Services

Appointment Records

Doctor Specialist	Obstetrician
Doctor Name	Not accepted yet!
Doctor Email	Not accepted yet!
Doctor Phone NO	Not accepted yet!
Meeting Location	At Clinic
Appointment Time	11:00 am
Appointment Date	12/4/2023
Patient Name	Vijay Yele
Patient Email	yelevijay1703@gmail.com
Patient Phone No	7448223696
Patient Age	24
Patient Address	32/1 Bhairavnath Nagar, Narhe Gaon Rd, opp. of Shri siddhi lawns, near saiba amruttulya, Mokarwadi, Pune, Maharashtra 411041, India, Pune, India
Appointment Status	Pending..

8:49

Siddhanath Medical Services

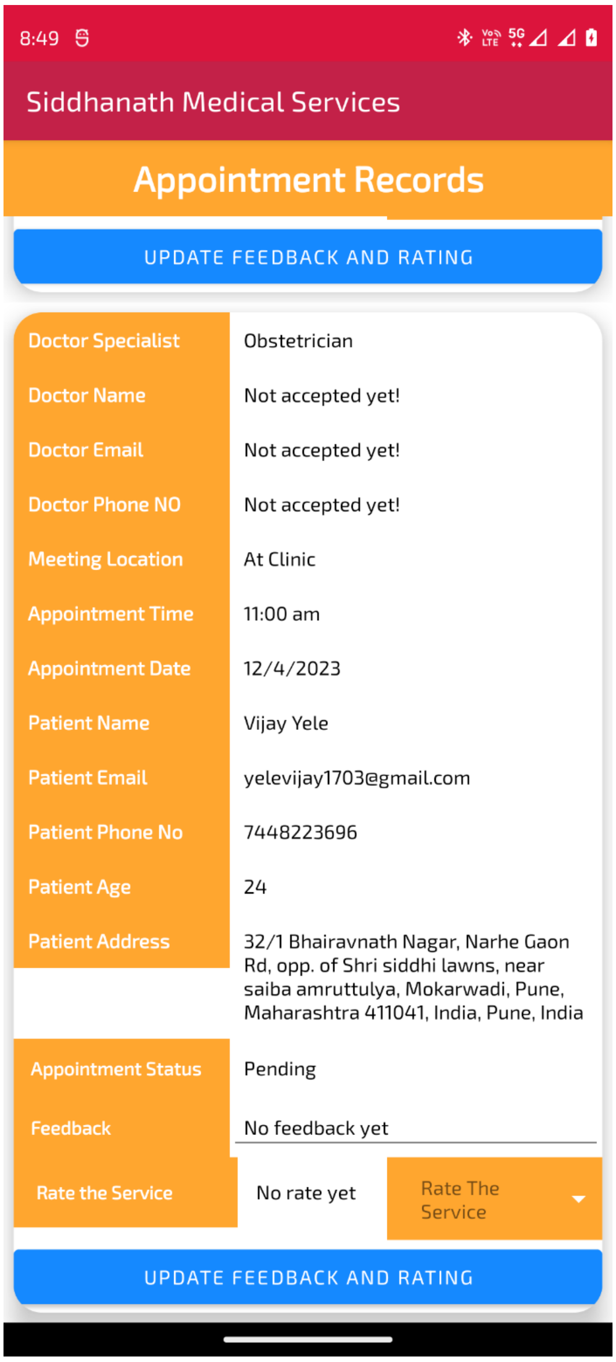
Appointment Records

Doctor Specialist	Obstetrician
Doctor Name	Not accepted yet!
Doctor Email	Not accepted yet!
Doctor Phone NO	Not accepted yet!
Meeting Location	At Clinic
Appointment Time	11:00 am
Appointment Date	12/4/2023
Patient Name	Vijay Yele
Patient Email	yelevijay1703@gmail.com
Patient Phone No	7448223696
Patient Age	24
Patient Address	32/1 Bhairavnath Nagar, Narhe Gaon Rd, opp. of Shri siddhi lawns, near saiba amruttulya, Mokarwadi, Pune, Maharashtra 411041, India, Pune, India
Appointment Status	Pending..

ACCEPT

Doctor Check Schedule Appointment:

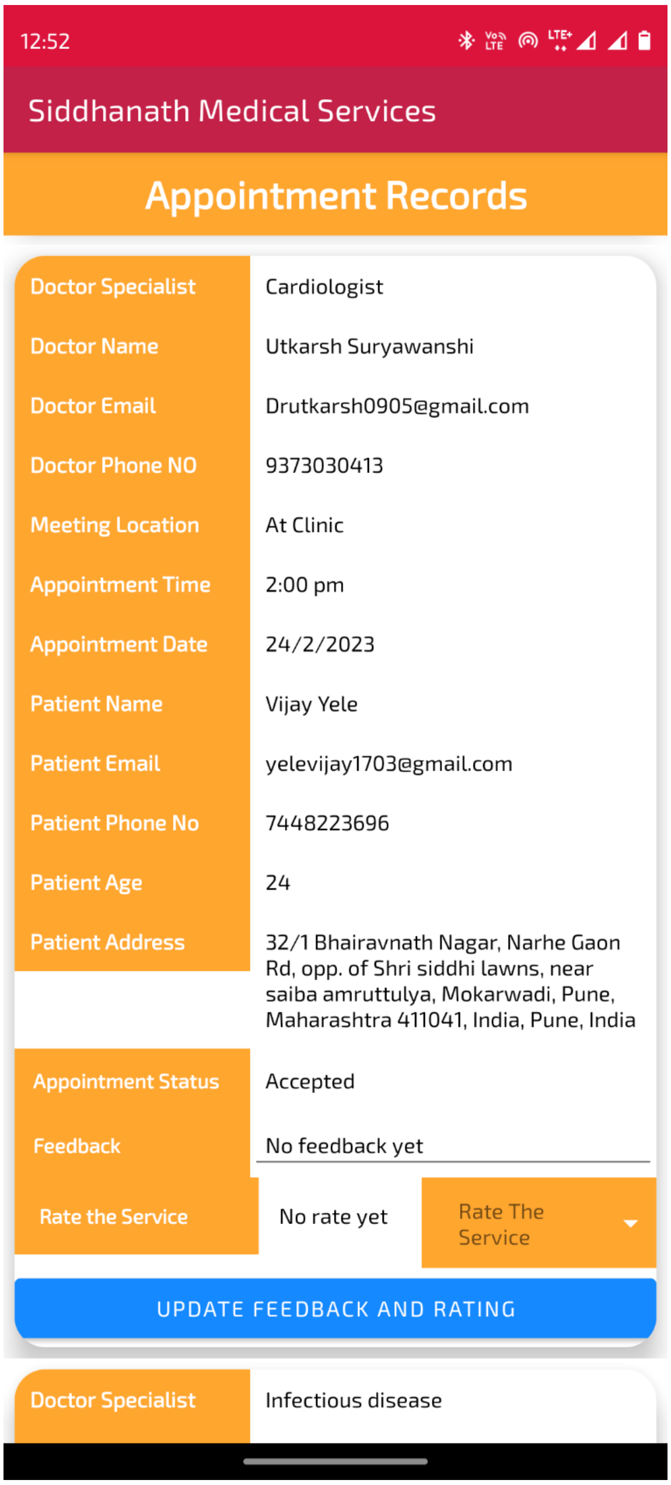
Doctor checks the appointment schedule.



Doctor Specialist	Obstetrician
Doctor Name	Not accepted yet!
Doctor Email	Not accepted yet!
Doctor Phone NO	Not accepted yet!
Meeting Location	At Clinic
Appointment Time	11:00 am
Appointment Date	12/4/2023
Patient Name	Vijay Yele
Patient Email	yelevijay1703@gmail.com
Patient Phone No	7448223696
Patient Age	24
Patient Address	32/1 Bhairavnath Nagar, Narhe Gaon Rd, opp. of Shri siddhi lawns, near saiba amruttulya, Mokarwadi, Pune, Maharashtra 411041, India, Pune, India
Appointment Status	Pending
Feedback	No feedback yet
Rate the Service	No rate yet

Patient Check Appointment Status:

Patient see their appointment status and give feedback and ratings.



12:52

Siddhanath Medical Services

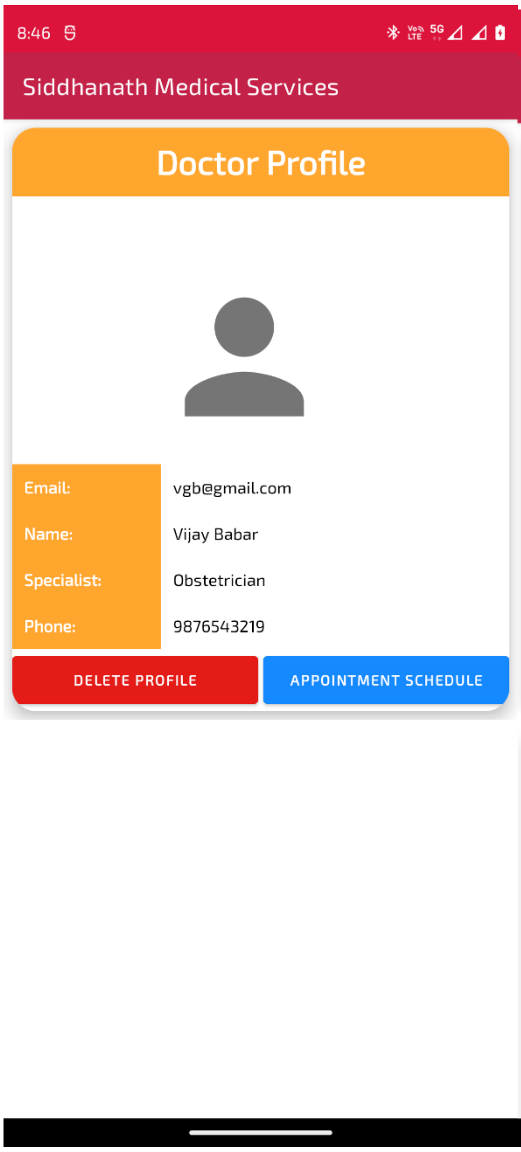
Appointment Records

Doctor Specialist	Cardiologist
Doctor Name	Utkarsh Suryawanshi
Doctor Email	Drutkarsh0905@gmail.com
Doctor Phone NO	9373030413
Meeting Location	At Clinic
Appointment Time	2:00 pm
Appointment Date	24/2/2023
Patient Name	Vijay Yele
Patient Email	yelevijay1703@gmail.com
Patient Phone No	7448223696
Patient Age	24
Patient Address	32/1 Bhairavnath Nagar, Narhe Gaon Rd, opp. of Shri siddhi lawns, near saiba amruttulya, Mokarwadi, Pune, Maharashtra 411041, India, Pune, India
Appointment Status	Accepted
Feedback	No feedback yet
Rate the Service	No rate yet

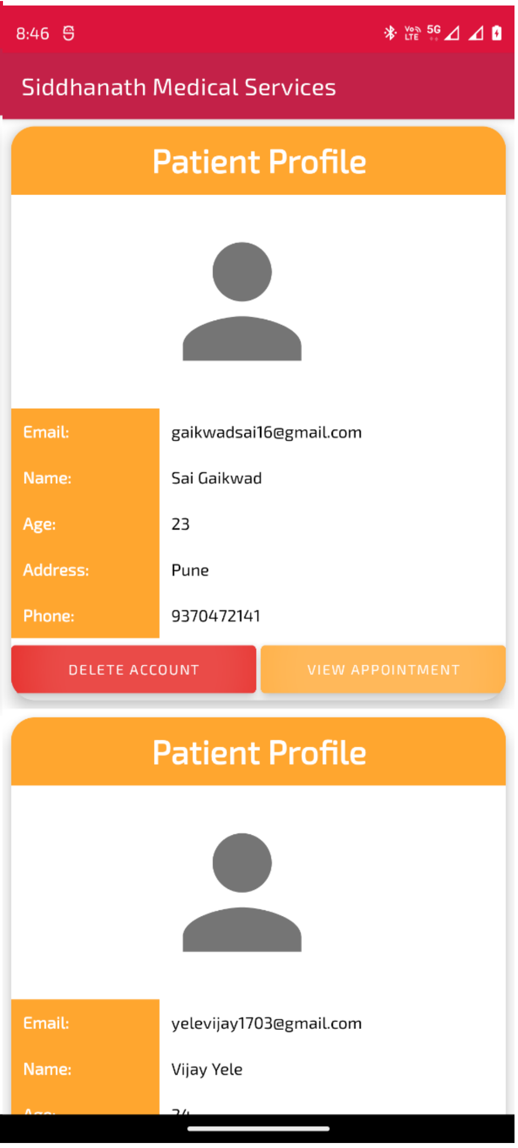
Doctor Specialist	Infectious disease

Admin Check Doctor, Patient, and Appointment Records:

Admin can see doctor and patient records.



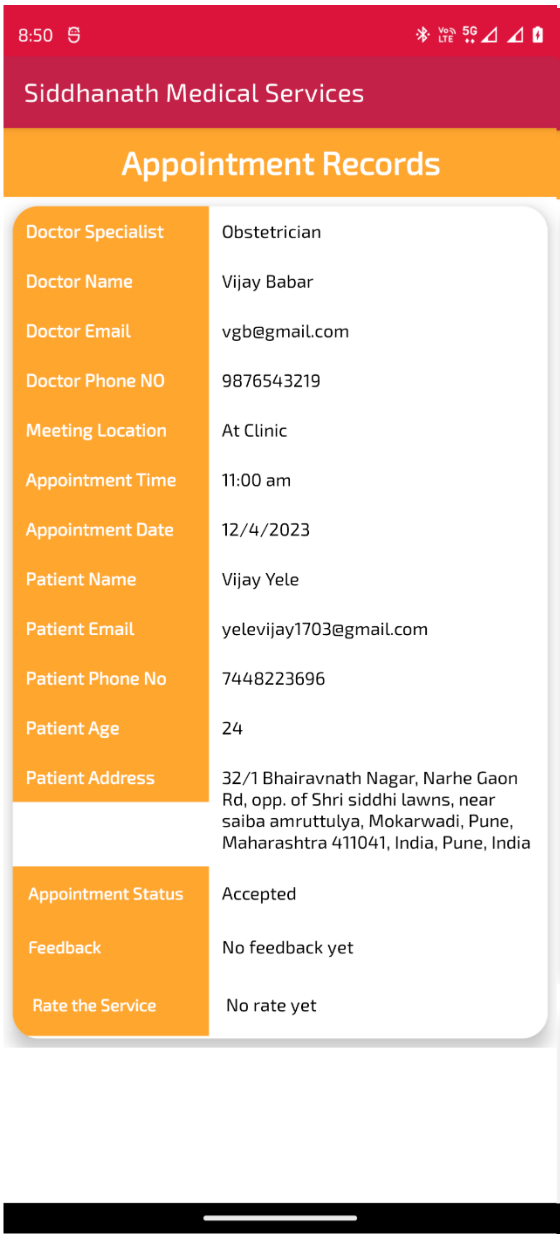
Email:	vgb@gmail.com
Name:	Vijay Babar
Specialist:	Obstetrician
Phone:	9876543219



Email:	gaikwadsai16@gmail.com
Name:	Sai Gaikwad
Age:	23
Address:	Pune
Phone:	9370472141

Email:	yelevijay1703@gmail.com
Name:	Vijay Yele

Admin can see all appointment records.

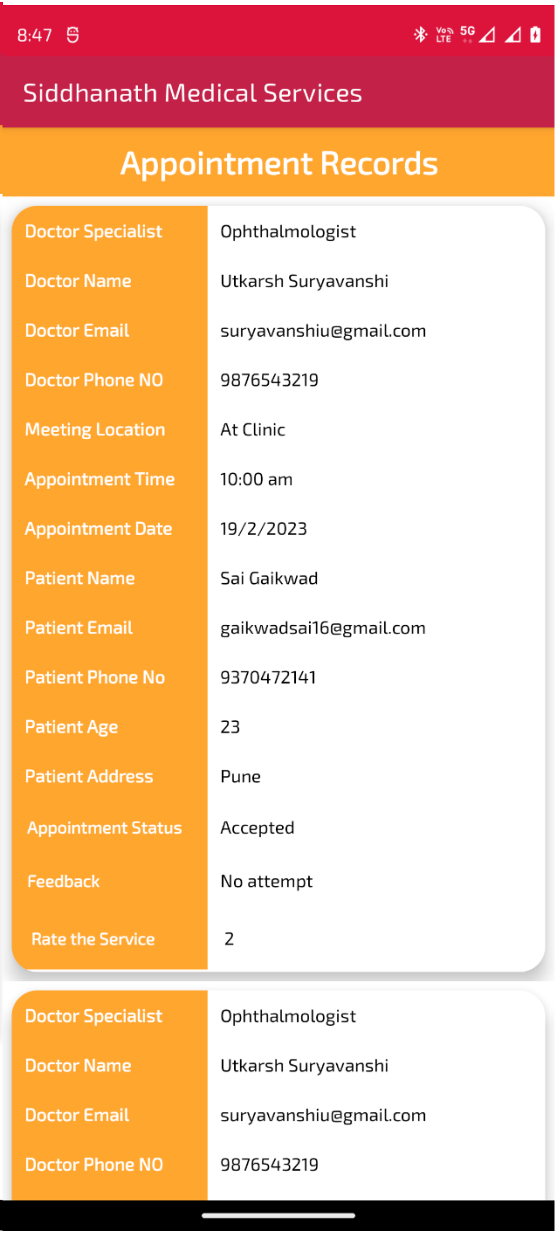


8:50

Siddhanath Medical Services

Appointment Records

Doctor Specialist	Obstetrician
Doctor Name	Vijay Babar
Doctor Email	vgb@gmail.com
Doctor Phone NO	9876543219
Meeting Location	At Clinic
Appointment Time	11:00 am
Appointment Date	12/4/2023
Patient Name	Vijay Yele
Patient Email	yelevijay1703@gmail.com
Patient Phone No	7448223696
Patient Age	24
Patient Address	32/1 Bhairavnath Nagar, Narhe Gaon Rd, opp. of Shri siddhi lawns, near saiba amruttulya, Mokarwadi, Pune, Maharashtra 411041, India, Pune, India
Appointment Status	Accepted
Feedback	No feedback yet
Rate the Service	No rate yet



8:47

Siddhanath Medical Services

Appointment Records

Doctor Specialist	Ophthalmologist
Doctor Name	Utkarsh Suryavanshi
Doctor Email	suryavanshiu@gmail.com
Doctor Phone NO	9876543219
Meeting Location	At Clinic
Appointment Time	10:00 am
Appointment Date	19/2/2023
Patient Name	Sai Gaikwad
Patient Email	gaikwadsai16@gmail.com
Patient Phone No	9370472141
Patient Age	23
Patient Address	Pune
Appointment Status	Accepted
Feedback	No attempt
Rate the Service	2

Doctor Specialist	Ophthalmologist
Doctor Name	Utkarsh Suryavanshi
Doctor Email	suryavanshiu@gmail.com
Doctor Phone NO	9876543219

Chapter 4

Coding Snippets

4.1. Coding

activity_login.xml:

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

    android:layout_width="fill_parent"

    android:layout_height="match_parent"

    xmlns:tools="http://schemas.android.com/tools"

    android:orientation="vertical"

    android:gravity="center_horizontal"

    android:background="@color/teal"


tools:context="com.example.clinicmanagementsystem.AuthAuthenticatioAndRegistra
tion.Login">



    <ImageView

        android:layout_width="200dp"

        android:layout_height="231dp"

        android:layout_gravity="center_horizontal"

        android:layout_marginTop="10dp"

        android:layout_marginBottom="60dp"

        android:scaleType="fitXY"

        android:src="@mipmap/logoc" />
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="150dp"
    android:background="@color/browser_actions_bg_grey"
    android:orientation="horizontal"
    android:layout_margin="20dp"
    android:padding="10dp">



    <EditText
        android:id="@+id/EditTxtAdminEmailLogin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="10dp"
        android:background="#00000000"
        android:hint="Enter Email"
        android:inputType="textEmailAddress"
        android:padding="8dp"
        android:textColor="#000000"
        android:textColorHint="#c6c6c6" />
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="25dp"
    android:background="@color/browser_actions_bg_grey"
    android:orientation="horizontal"
    android:layout_margin="20dp"
    android:padding="10dp">



    <EditText
        android:id="@+id/EditxtAdminPasswordLogin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="10dp"
        android:background="#00000000"
        android:hint="Enter Password"
        android:inputType="textPassword"
        android:padding="8dp"
        android:textColor="#000000"
        android:textColorHint="#c6c6c6" />
</LinearLayout>


<Button
```

```
    android:id="@+id/BtnLogin"
    android:layout_width="144dp"
    android:layout_height="50dp"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="30dp"
    android:layout_marginBottom="2dp"
    android:text="LOGIN"
    android:backgroundTint="@color/Green"
    android:textAllCaps="false"
    android:textColor="#FFFFFF"
    android:textSize="15sp" />
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="30dp"
    android:gravity="center_horizontal"
    android:onClick="onSignupPatient"
    android:textStyle="bold"
    android:textColor="@color/black"
    android:text="Do not you have an account yet !, \nSign up"
    android:textSize="15sp" />
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
        android:layout_marginTop="30dp"

        android:gravity="center_horizontal"

        android:textStyle="normal"

        android:text="Powered by: Jay Yele"

        android:textColor="@color/black"

        android:textSize="15sp" />

</LinearLayout>
```

Login.java:

```
package com.example.clinicmanagementsystem.AuthAuthenticatioAndRegistration;


import android.app.Activity;

import android.app.ProgressDialog;

import android.content.Intent;

import android.os.Bundle;

import android.text.TextUtils;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;


import androidx.annotation.NonNull;


import com.example.clinicmanagementsystem.Activities.AdminPanel;
import com.example.clinicmanagementsystem.Activities.DoctorPanel;

import com.example.clinicmanagementsystem.Activities.PatientPanel;

import com.example.clinicmanagementsystem.R;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.AuthResult;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.QuerySnapshot;


public class Login extends Activity {


    ProgressDialog mProgressDialog;

    Button btnLogin;

    EditText edtEmail, edtPassword;

    FirebaseFirestore firebaseFirestore;

    boolean CheckUser = false;

    private FirebaseAuth firebaseAuth;


    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_login);
        edtEmail = findViewById(R.id.EditTxtAdminEmailLogin);
        edtPassword = findViewById(R.id.EditxtAdminPasswordLogin);
        btnLogin = findViewById(R.id.BtnLogin);
        mProgressDialog = new ProgressDialog(this);
        firebaseAuth = FirebaseAuth.getInstance();
        mProgressDialog.setMessage("Please wait, Logging in..");
        firebaseFirestore = FirebaseFirestore.getInstance();
        btnLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                checkLoginDetails();
            }
        });
    }

    private void AdminLogin(String email, String password) {
        //we take the email and password to check if the usr is admin or not
        firebaseAuth.signInWithEmailAndPassword(email,
password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful()){
                    Toast.makeText(Login.this, "Success Login as Admin",
Toast.LENGTH_LONG).show();
```

```
                Intent intent = new Intent(Login.this, AdminPanel.class);
                startActivity(intent);
                finish();
            }else {
                DoctorLogin(email,password);
            }
        }
    });
}

public void DoctorLogin(String email, String password) {

    FirebaseAuth.getInstance().signOut();
    firebaseFirestore.collection("DoctorRecords")
            .get()
            .addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
                @Override
                public void onComplete(@NonNull Task<QuerySnapshot> task) {
                    if (task.isSuccessful()){
                        for (int i=0; i<task.getResult().getDocuments().size(); i++)
                        {
                            // fetch data
                            String fname =
task.getResult().getDocuments().get(i).getString("name").trim();
```

```
                                String femail=
task.getResult().getDocuments().get(i).getString("email").trim();

                                String fphoneNo =
task.getResult().getDocuments().get(i).getString("phoneNo").trim();

                                String fspecialize =
task.getResult().getDocuments().get(i).getString("specialize").trim();

                                String fpassword =
task.getResult().getDocuments().get(i).getString("password").trim();

                                String ftoken =
task.getResult().getDocuments().get(i).getString("token").trim();



                                if (femail.equals(email) && fpassword.equals(password)){

                                    Toast.makeText(Login.this, "Welcome back, "+
fname,Toast.LENGTH_LONG).show();

                                    Intent intent = new Intent(Login.this, DoctorPanel.class);

                                    intent.putExtra("fname", fname);

                                    intent.putExtra("femail", femail);

                                    intent.putExtra("fspecialize",fspecialize);

                                    intent.putExtra("fphoneNo", fphoneNo);

                                    intent.putExtra("ftoken", ftoken);

                                    startActivity(intent);

                                    finish();

                                    CheckUser = true;

                                }

                            }
```

```
                    if (CheckUser == false){
                        PatientLogin(email, password);
                    }
                }else {
                    Toast.makeText(Login.this, "Error, Check Internet
connection!",Toast.LENGTH_LONG).show();
                }
            }
        });


    }
    public void PatientLogin(String email, String password) {


        FirebaseAuth.getInstance().signOut();
        firebaseFirestore.collection("PatientRecords")
                .get()
                .addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
                    @Override
                    public void onComplete(@NonNull Task<QuerySnapshot> task) {
                        if (task.isSuccessful()){
                            for (int i=0; i<task.getResult().getDocuments().size(); i++)
                            {
                                // fetch data
                        String fname =
task.getResult().getDocuments().get(i).getString("name").trim();

                        String femail =
task.getResult().getDocuments().get(i).getString("email").trim();

                        String fpassword =
task.getResult().getDocuments().get(i).getString("password").trim();

                        String fage =
task.getResult().getDocuments().get(i).getString("age").trim();

                        String faddress =
task.getResult().getDocuments().get(i).getString("address").trim();

                        String fphone =
task.getResult().getDocuments().get(i).getString("phone").trim();

                        String ftoken =
task.getResult().getDocuments().get(i).getString("token").trim();


                        if (femail.equals(email) && fpassword.equals(password)){

                            Toast.makeText(Login.this, "Welcome back, "+
fname,Toast.LENGTH_LONG).show();

                            Intent intent = new Intent(Login.this, PatientPanel.class);

                            intent.putExtra("fname", fname);

                            intent.putExtra("femail", femail);

                            intent.putExtra("faddress",faddress);

                            intent.putExtra("fage", fage);

                            intent.putExtra("fphone", fphone);

                            intent.putExtra("ftoken", ftoken);

                            startActivity(intent);
```

```
                                finish();

                                CheckUser = true;

                            }

                        }

                        if (CheckUser == false){

                            Toast.makeText(Login.this, "Wrong Email or Password
!",Toast.LENGTH_LONG).show();

                        }


                    }else {

                        Toast.makeText(Login.this, "Error, Check Internet
connection",Toast.LENGTH_LONG).show();

                    }

                }

            });


    }

    private void checkLoginDetails() {

        if(!TextUtils.isEmpty(edtEmail.getText().toString()) &&
!TextUtils.isEmpty(edtPassword.getText().toString())){

             //login

            //initLogin(edtEmail.getText().toString(), edtPassword.getText().toString());

            AdminLogin(edtEmail.getText().toString(), edtPassword.getText().toString());

        }else{

            if(TextUtils.isEmpty(edtEmail.getText().toString())){
```

```
            edtEmail.setError("Please enter a valid email!");
            return;
          }if(TextUtils.isEmpty(edtPassword.getText().toString())){
            edtPassword.setError("Please enter password!");
            return;
          }
        }
    }
    public void onSignupPatient(View view) {
        Intent intent = new Intent(Login.this, PatientRegistration.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK |
Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
        finish();
    }
    @Override
    public void onBackPressed() {
        super.onBackPressed();
        Intent homeIntent = new Intent(Intent.ACTION_MAIN);
        homeIntent.addCategory( Intent.CATEGORY_HOME );
        homeIntent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(homeIntent);
    }
}
```

activity_patient_register.xml:

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout

    xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:orientation="vertical"


tools:context="com.example.clinicmanagementsystem.AuthAuthenticatioAndRegistra
tion.PatientRegistration"

    android:background="@color/teal_200">



    <TextView

        android:id="@+id/txtViewTitle"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:layout_marginStart="25dp"

        android:layout_marginTop="60dp"

        android:layout_marginBottom="5dp"

        android:text="Register new account"

        android:textColor="@color/black"

        android:textStyle="bold"
        android:textSize="30sp"
        android:layout_marginLeft="25dp" />
    <TextView
        android:id="@+id/txtSubTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="25dp"
        android:layout_marginBottom="50dp"
        android:text="Please Enter Patient Details "
        android:textSize="17sp"
        android:layout_marginLeft="25dp" />

    <EditText
        android:id="@+id/EditTxtPName"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:layout_marginBottom="10dp"
        android:hint="Full Name"
        android:inputType="textPersonName"
        android:padding="15dp"
        android:textSize="15sp" />
```

```
<EditText
    android:id="@+id/EditTxtPEmail"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginBottom="10dp"
    android:hint="Email Address"
    android:inputType="textEmailAddress"
    android:padding="15dp"
    android:textSize="15sp" />

<EditText
    android:id="@+id/EditTxtPPassword"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginBottom="10dp"
    android:hint="Password"
    android:inputType="textPassword"
    android:padding="15dp"
    android:textSize="15sp" />
```

```
<EditText
    android:id="@+id/EditTxtPPhoneNo"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginBottom="10dp"
    android:hint="Phone No"
    android:inputType="phone"
    android:padding="15dp"
    android:textSize="15sp" />


<EditText
    android:id="@+id/EditTxtPAge"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginBottom="10dp"
    android:hint="Age"
    android:inputType="phone"
    android:padding="15dp"
    android:textSize="15sp" />
<LinearLayout
```

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal"
android:weightSum="100"
>
<EditText
    android:id="@+id/EditTxtPAddress"
    android:layout_width="0dp"
    android:layout_weight="90"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="10dp"
    android:layout_marginBottom="10dp"
    android:hint="Full Address"
    android:inputType="text|textMultiLine"
    android:padding="15dp"
    android:textSize="15sp" />

<ImageView
    android:id="@+id/edlocationImage"
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:padding="15dp"
    android:src="@drawable/ic_location" />
    </LinearLayout>
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginRight="20dp"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="15dp"
        android:layout_marginBottom="15dp"
        android:layout_weight="1">

        <Button
            android:layout_width="180dp"
            android:layout_height="wrap_content"
            android:layout_alignParentBottom="true"
            android:layout_centerHorizontal="true"
            android:layout_marginBottom="30dp"
            android:backgroundTint="@color/teal_700"
            android:onClick="onRegisterPatientBtnClick"
            android:text="Sign up"
            android:textColor="@color/white"
            android:textSize="14dp"
            android:textStyle="bold" />
    </RelativeLayout>
</LinearLayout>
```

PatientInfo.java:

```
package com.example.clinicmanagementsystem.PatientAPI;


public class PatientInfo {
    private String name, email, password, age, address,phone, token;


    public PatientInfo(String name, String email, String password, String age, String
address, String phone, String token) {
        this.name = name;
        this.email = email;
        this.password = password;
        this.age = age;
        this.address = address;
        this.phone = phone;
        this.token = token;
    }


    public PatientInfo() {


    }


    public String getPhone() {
        return phone;
    }
```

```
public void setPhone(String phone) {
    this.phone = phone;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getEmail() {
    return email;
}

public void setEmail(String email) {
    this.email = email;
}

public String getPassword() {
    return password;
}
```

```
public void setPassword(String password) {
    this.password = password;
}

public String getAge() {
    return age;
}

public void setAge(String age) {
    this.age = age;
}

public String getAddress() {
    return address;
}

public void setAddress(String address) {
    this.address = address;
}

public String getToken() {
    return token;
}
```

```
    public void setToken(String token) {
        this.token = token;
    }
}
```

activity_doctor_register.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"

tools:context="com.example.clinicmanagementsystem.AuthAuthenticatioAndRegistra
tion.DoctorRegistration"
    android:background="@color/teal_200">


    <TextView
        android:id="@+id/txtViewTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="25dp"
        android:layout_marginTop="60dp"
        android:layout_marginBottom="5dp"
        android:text="Register new account"
        android:textSize="30sp"
        android:textColor="@color/black"
    android:textStyle="bold"
    android:layout_marginLeft="25dp" />
<TextView
    android:id="@+id/txtSubTitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="25dp"
    android:layout_marginBottom="50dp"
    android:text="Please Enter Doctor Details "
    android:textSize="17sp"
    android:layout_marginLeft="25dp" />

<EditText
    android:id="@+id/EditTxtDName"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginBottom="10dp"
    android:hint="Full Name"
    android:inputType="textPersonName"
    android:padding="15dp"
    android:textSize="15sp" />
```

```
<EditText
    android:id="@+id/EditTxtDEmail"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginBottom="10dp"
    android:hint="Email Address"
    android:inputType="textEmailAddress"
    android:padding="15dp"
    android:textSize="15sp" />

<EditText
    android:id="@+id/EditTxtDPassword"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginBottom="10dp"
    android:hint="Password"
    android:inputType="textPassword"
    android:padding="15dp"
    android:textSize="15sp" />
```

```
<EditText
    android:id="@+id/EditTxtDPhoneNo"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginBottom="10dp"
    android:hint="Phone No"
    android:inputType="phone"
    android:padding="15dp"
    android:textSize="15sp" />


<com.jaredrummler.materialspinner.MaterialSpinner
    android:id="@+id/spinnerID"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginTop="20dp"
    android:textColor="@color/white"
    app:ms_background_color="@color/teal"
    android:visibility="visible"
    app:ms_hint="Specialize" />
<RelativeLayout
```

```
        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:layout_marginRight="20dp"

        android:layout_marginLeft="20dp"

        android:layout_marginTop="15dp"

        android:layout_marginBottom="15dp"

        android:layout_weight="1">


        <Button

            android:layout_width="180dp"

            android:layout_height="wrap_content"

            android:layout_alignParentBottom="true"

            android:layout_centerHorizontal="true"

            android:layout_marginBottom="50dp"

            android:backgroundTint="@color/teal"

            android:clickable="true"

            android:onClick="onRegisterBtnClick"

            android:text="Sign up"

            android:textColor="@color/white"

            android:textSize="14dp"

            android:textStyle="bold" />


    </RelativeLayout>

</LinearLayout>
```

DoctorInfo.java:

```
package com.example.clinicmanagementsystem.DoctorAPI;


public class DoctorInfo {
    String name, email, phoneNo, specialize,password, token;


    public DoctorInfo(String name, String email, String phoneNo, String specialize,
String password, String token) {
        this.name = name;
        this.email = email;
        this.phoneNo = phoneNo;
        this.specialize = specialize;
        this.password = password;
        this.token = token;
    }


    public DoctorInfo() {


    }


    public String getPassword() {
        return password;
    }
```

```
public void setPassword(String password) {
    this.password = password;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getEmail() {
    return email;
}

public void setEmail(String email) {
    this.email = email;
}

public String getPhoneNo() {
    return phoneNo;
}
```

```
    public void setPhoneNo(String phoneNo) {
        this.phoneNo = phoneNo;
    }

    public String getSpecialize() {
        return specialize;
    }

    public void setSpecialize(String specialize) {
        this.specialize = specialize;
    }

    public String getToken() {
        return token;
    }

    public void setToken(String token) {
        this.token = token;
    }
}
```

activity_admin_panel.xml:

```
<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout

    xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    tools:context="com.example.clinicmanagementsystem.Activities.AdminPanel"

    android:background="@color/teal_200">


    <View

        android:id="@+id/bg_top_header"

        android:layout_width="match_parent"

        android:layout_height="122dp"

        app:layout_constraintStart_toStartOf="parent"

        app:layout_constraintTop_toTopOf="parent" />




    <TextView


        android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="47dp"
    android:layout_marginTop="32dp"
    android:text="Admin Dashboard"
    android:textColor="@color/black"
    android:textSize="30dp"
    android:textStyle="bold"
    app:layout_constraintBottom_toTopOf="@+id/gridLayout"
    app:layout_constraintEnd_toEndOf="@+id/bg_top_header"
    app:layout_constraintHorizontal_bias="0.211"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.472" />

  <ImageView
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:layout_marginStart="36dp"
    android:layout_marginTop="72dp"
    android:src="@drawable/ic_logout"
    android:onClick="onLogoutAdminClick"
    app:layout_constraintBottom_toTopOf="@+id/gridLayout"
    app:layout_constraintEnd_toEndOf="@+id/bg_top_header"
    app:layout_constraintHorizontal_bias="0.27"
```

```
        app:layout_constraintStart_toEndOf="@+id/textView"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.062" />


    <GridLayout
        android:id="@+id/gridLayout"
        android:layout_width="match_parent"
        android:layout_height="400dp"
        android:layout_marginTop="170dp"
        android:layout_marginBottom="60dp"
        android:alignmentMode="alignMargins"
        android:columnCount="2"
        android:columnOrderPreserved="false"
        android:padding="14dp"
        android:rowCount="3"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@id/bg_top_header"
        app:layout_constraintVertical_bias="0.0"
        app:layout_editor_absoluteX="0dp">


        <androidx.cardview.widget.CardView
            android:layout_width="0dp"
```

```
android:layout_height="0dp"
android:layout_rowWeight="1"
android:layout_columnWeight="1"
android:layout_marginLeft="16dp"
android:layout_marginRight="16dp"
android:layout_marginBottom="16dp"
android:backgroundTint="@color/teal"
android:onClick="onAddDoctorClick"
app:cardCornerRadius="20dp"
app:cardElevation="6dp">

<LinearLayout
    android:layout_width="130dp"
    android:layout_height="130dp"
    android:layout_gravity="center_horizontal|center_vertical"
    android:layout_margin="16dp"
    android:gravity="center"
    android:orientation="vertical">

    <ImageView
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:layout_gravity="center_horizontal"
        android:background="@drawable/ic_add" />
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="8dp"
            android:text="Add Doctor"
            android:textAlignment="center"
            android:textSize="16sp"
            android:textStyle="bold" />
    </LinearLayout>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_rowWeight="1"
    android:layout_columnWeight="1"
    android:layout_marginLeft="16dp"
    android:layout_marginRight="16dp"
    android:layout_marginBottom="16dp"
    android:backgroundTint="@color/teal"

    android:onClick="onDoctorRecordsClick"
```

```
app:cardCornerRadius="20dp"
app:cardElevation="6dp">

<LinearLayout
    android:layout_width="130dp"
    android:layout_height="130dp"
    android:layout_gravity="center_horizontal|center_vertical"
    android:layout_margin="16dp"
    android:gravity="center"
    android:orientation="vertical">

    <ImageView
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:layout_gravity="center_horizontal"
        android:background="@drawable/ic_records" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:text="Doctor Records"
        android:textAlignment="center"
        android:layout_gravity="center_horizontal"
```

```
            android:textSize="16sp"
            android:textStyle="bold" />
    </LinearLayout>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_rowWeight="1"
    android:layout_columnWeight="1"
    android:layout_marginLeft="16dp"
    android:layout_marginRight="16dp"
    android:layout_marginBottom="16dp"
    android:backgroundTint="@color/teal"
    android:onClick="onPatientRecordsClick"
    app:cardCornerRadius="20dp"
    app:cardElevation="6dp">

    <LinearLayout
        android:layout_width="130dp"
        android:layout_height="130dp"
        android:layout_gravity="center_horizontal|center_vertical"
        android:layout_margin="16dp"
        android:gravity="center"
        android:orientation="vertical">

        <ImageView
            android:layout_width="70dp"
            android:layout_height="70dp"
            android:layout_gravity="center_horizontal"
            android:background="@drawable/ic_records" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="8dp"
            android:text="Patient Records"
            android:textSize="16sp"
            android:textStyle="bold" />
    </LinearLayout>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_rowWeight="1"
```

```
android:layout_columnWeight="1"
android:layout_marginLeft="16dp"
android:layout_marginRight="16dp"
android:layout_marginBottom="16dp"
android:onClick="onAppointmentRecordsClick"
android:backgroundTint="@color/teal"
app:cardCornerRadius="20dp"
app:cardElevation="6dp">

<LinearLayout
   android:layout_width="130dp"
   android:layout_height="130dp"
   android:layout_gravity="center_horizontal|center_vertical"
   android:layout_margin="16dp"
   android:gravity="center"
   android:orientation="vertical">

   <ImageView
      android:layout_width="70dp"
      android:layout_height="70dp"
      android:layout_gravity="center_horizontal"
      android:background="@drawable/ic_schedule" />
   <TextView
      android:layout_width="wrap_content"
```

```
                android:layout_height="wrap_content"

                android:layout_marginTop="8dp"

                android:textAlignment="center"

                android:text="Appointment Records"

                android:textSize="16sp"

                android:textStyle="bold" />

            </LinearLayout>


        </androidx.cardview.widget.CardView>


    </GridLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

AdminPanel.java:

```
package com.example.clinicmanagementsystem.Activities;


import android.app.Activity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;


import
com.example.clinicmanagementsystem.AuthAuthenticatioAndRegistration.DoctorRe
gistration;
```

```
import
com.example.clinicmanagementsystem.AuthAuthenticatioAndRegistration.Login;

import com.example.clinicmanagementsystem.R;


public class AdminPanel extends Activity {

   @Override

   protected void onCreate(Bundle savedInstanceState) {

      super.onCreate(savedInstanceState);

      setContentView(R.layout.activity_admin_panel);

   }


   public void onAddDoctorClick(View view) {

      Intent intent = new Intent(AdminPanel.this, DoctorRegistration.class);

      intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK |
Intent.FLAG_ACTIVITY_NEW_TASK);

      startActivity(intent);

      finish();

   }


   public void onDoctorRecordsClick(View view) {

      Intent intent = new Intent(AdminPanel.this, DoctorRecords.class);

      intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK |
Intent.FLAG_ACTIVITY_NEW_TASK);

      startActivity(intent);

      finish();
```

```
    }


    public void onPatientRecordsClick(View view) {
        Intent intent = new Intent(AdminPanel.this, PatientRecords.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK |
Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
        finish();
    }


    public void onAppointmentRecordsClick(View view) {
        Intent intent = new Intent(AdminPanel.this, AppointmentRecords.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK |
Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
        finish();
    }


    public void onLogoutAdminClick(View view) {
        Intent intent = new Intent(AdminPanel.this, Login.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK |
Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
        finish();
    }
    @Override
    public void onBackPressed() {
        super.onBackPressed();
        Intent intent = new Intent(AdminPanel.this, Login.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK |
Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
        finish();
    }
}
```

Chapter 5

Testing, Limitations, Proposed Enhancement & Conclusion

5.1. Testing:

5.1.1. Test Case for Patient Dashboard:

Test Details: When the patient login into the system all functionality of the patients dashboard will work successfully.

Test Id	Test Feature	Test Description	Expected Result	Test Status
PD1	Patient Login	Fill the login data and check patient login successfully in system	"Welcome Back, (User Name)". And See Patient Dashboard	Pass
PD2	Create Appointment	Click on Create Appointment and and fill appointment details and click on request.	Show massage "requested successfully" on patient dashboard	Pass
PD3	Schedule Appointment	Enter in Schedule appointment and check appointment confirmation status and give feedback and rating	Show appointment status pending or accepted. And update feedback and rating.	Pass
PD4	Patient Profile	Click on View Profile and want edit the details.	See massage profile "update successfully".	Pass

5.1.2. Test Case for Doctor Dashboard:

Test Details: When doctor login into the system all functionality of the doctor dashboard will work successfully.

Test Id	**Test Feature**	**Test Description**	**Expected Result**	**Test Status**
DD1	Doctor Login	Fill the login data and check doctor login successfully in system	“Welcome Back, (Doctor Name)”. And See Doctor Dashboard	Pass
DD2	Requested Appointment	Enter in Requested Appointment and accept the pending appointment.	Show massage “Appointment has been accepted”	Pass
DD3	Schedule Appointment	Click on Schedule Appointment.	Show the doctor appointment list	Pass
DD4	Doctor Profile	Click on View Profile and want edit the details.	See massage profile “Profile update successfully”.	Pass

5.1.3. Test Case for Admin Dashboard:

Test Details: When the admin login into the system all functionality of the admin dashboard will work successfully.

Test Id	Test Feature	Test Description	Expected Result	Test Status
AD1	Admin Login	Fill the login data and check admin login successfully in system	"Success login as admin". And See Doctor Dashboard	Pass
AD2	Add Doctor	Click to Add Doctor enter details of doctor and sign up.	Show massage on dashboard "Doctor Added Successfully"	Pass
AD3	Doctor Record	Check how many doctor are register on system and and check their appointment schedules and also delete doctor authentication	See massage "Doctor Record Delete".	Pass
AD4	Patient Record	Check how many patients are register on system and and check their appointment schedules and also delete patient authentication	See massage "Patient Record Delete".	Pass

AD5	Appointment Records	Click on appointment records to see all the appointments details	See all appointment details.	Pass

5.2. Limitations:

- This application is only made for booking appointments.
- This is providing home facilities in nearby areas.
- Application requires the internet to book appointments.
- Only the admin can add a doctor to the application.
- Location access is needed for patients.

5.3. Future Enhancement:

- Integration with Electronic Health Records (EHRs): An appointment app that can integrate with patients' EHRs could provide a more comprehensive view of their medical history, including medications, allergies, and previous procedures or diagnoses. This information can be shared with healthcare providers to enable better-informed decisions and more personalized care.

- Virtual consultations: With the increasing popularity of telemedicine, integrating virtual consultations within the appointment app can be a great enhancement. Patients can see their doctors remotely, from the comfort of their own homes, without having to travel to the clinic. This would be especially beneficial for people with chronic conditions who require frequent check-ins.

- Automatic appointment reminders: The app can send automated appointment reminders to patients via text message or email, reducing the likelihood of missed appointments. These reminders could also include instructions for preparing for the appointment, such as fasting or bringing certain documents.

- Wait time notifications: Patients can receive notifications about estimated wait times before their appointment. This can reduce frustration and anxiety, as they will know when they can expect to be seen by their healthcare provider.

- In-app prescription refills: Patients can request prescription refills within the app and have them delivered directly to their homes, making it more convenient and accessible for patients with mobility or transportation issues.

- Ratings and reviews: Patients can leave feedback about their experience with their healthcare provider, including ratings and reviews. This can help other patients choose a provider that best fits their needs and preferences.

- Integration with wearable devices: The app can integrate with wearable devices, such as fitness trackers or smartwatches, to gather additional health data that can be shared with healthcare providers. This can provide a more complete picture of a patient's health and help with diagnosis and treatment decisions.

5.4. Conclusion:

A doctor appointment app can be a valuable tool for patients and healthcare providers. By providing a convenient and easy-to-use platform for scheduling appointments. Easy to access and the database is on the firebase online platform.

5.5. Bibliography:

For UML

https://www.youtube.com/watch?v=9CkpMm-n5iA

For Frontend:

https://dribbble.com/tags/android_ui

https://developer.android.com/develop/ui

For Backend

https://console.firebase.google.com/u/0/

https://firebase.google.com/docs/database/android/start

A

Synopsis On

"Doctor Appointment App"

For

Siddhanath Medical Services, Kolhapur

By

Jay Dattatray Yele (MC21066)

Submitted to

SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

For the partial fulfillment of the internal credit work of

MASTERS OF COMPUTER APPLICATION SEM – III

Through

Zeal Education Society's

Zeal Institute of Business Administration, Computer Application & Research (ZIBACAR)

Sr. No. 39, Narhe, Pune - 411041, Phone No.:67206031

(Approved by A.I.C.T.E., New Delhi, Recognized by DTE, Govt. Maharashtra & Affiliated to S.P.P.U.Pune)

2022-2023

Project Synopsis

1. Introduction:

- **Course Name:** Master in Computer Application (Semester III).
- **Student Name:** Jay Dattatray Yele (MC21066).
- **Project Title:** Doctor Appointment App.
- **Name of Internal Guide:** Dr. Rupali Pawar.
- **Date Of Submission:** 28/10/2022

2. Organization Profile:

- Client Name: Dr. Ramesh Suryawanshi.
- Location & Address: Sadashivnagar, Kagal, Kolhapur, 416235.
- Contact Number: 9373030413.
- About Organization: The Siddhanath Medical Services are situated in Sadashivnagar, Kagal, Kolhapur. They provide hospital services. The Siddhanath Medical Services have started in pandemic period. So many patients are visited daily. The Dr. Ramesh Suryawanshi was started this medical services.

3. Project Abstract:

Online Doctor appointment App in hospital today necessitate a competent administration when handling patients, patient details which serves as a key factor for the flow of business transactions in Siddhanath Medical Services. Unfortunately, the current record management system leads to misplacement of during details, Patient details and doctor record of reports and insecurity to records. This project is aimed at computerizing all the records about patients, hospital and doctors. In order to achieve this goal, a through system study and investigation was carried out and data was collected and analysed about the current system using document and data flow diagrams. Errors made on hand held calculators and dealt out completely. The method used to develop the system include iterative, logical and entity relationship diagram were used to design the system.

4. Existing System:

Under manual Online Doctor Appointment Booking App, you have to first wait in line to take appointment for the doctors and wait for your time to have meet with them and discuss on your health problems. As you have to provide your information and other reports many times at different places such as the medicine store which is again a burden of carrying documents. You have to be present physically at the doctor's cabin. Patients have to visit on another day of after some hours to take their health reports which involves extra care person with patients anytime. Under manual system, the only accepted payment method is by cash and if patients due to some reasons are not having cash on time may face difficulties and not able to get treatment.

- In this system patient want to visit the hospital directly.
- Wait in line for register their self for taking appointment.
- Patient want to submit all details asking by receptionist.
- Then they give the appointment time and date to patient.
- Patient waits hours and hours in hospital.
- After that they discuss problem with doctor.
- All the system is manual.

5. Proposed System:

The proposed project is a smart appointment booking system that provides patients or any user an easy way of booking a doctor’s appointment online. This is android application that overcomes the issue of managing and booking appointments according to user’s choice or demands. The task sometimes becomes very tedious for the compounder or doctor himself in manually allotting appointments for the users as per their availability. Hence this project offers an effective solution where users can view various booking slots available and select the preferred date and time. The already booked space will be marked yellow and will not be available for anyone else for the specified time. This system also allows users to cancel their booking anytime.

- The system will be design for the patient who want to fix appointment with doctor.
- This app helps the doctor to accept the appointment as per there schedule.
- First Patient want to submit their all details for registration.

- Then they have login into the system.
- After that they select the doctor for treatment.
- Then doctor or staff check details and confirm their appointment.
- All process will do by any place in online mode.

6. Objectives:

- To make a way of easy appointment booking.
- To provide immediate service.
- To reduce time of appointment booking.
- To improve hospital system faster and easy.
- To book appointment from any location.
- To interact with doctor from any location.

7. Scope Of the System:

The Doctor Appointment Booking App include various modules as follows:

7.1 Patient Module:

The main objective of this module is providing all the functionality related to patient. It tracks all the information and details of the patient. We have developed all type of CRUD (Create, Read, Update and Delete) operations of the patient.

7.2 Doctor Module:

The main objective for developing this module is provide all the functionality related to doctor. It tracks all the information and details of the doctor. We have developed all type of CRUD (Create, Read, Update and Delete) operations of the doctor. This is a role-based module where admin can perform each and every operation on data but the doctor will be able to view only his/her data, so access level restrictions has also been implemented on the project.

7.3 Appointment Module:

The main aim for developing this module is to manage the doctor appointment. This Appointment Module is an important module in this project Doctor Appointment System which we have developed on PHP and MySQL. So, all appointment will be managed by admin and patient and doctor will be able to see the appointment.

8. Limitations:

This system is made for booking appointment. In this system only two main user Doctor and Patient only. The limitation of this application is the user need the android phone. And user need to internet connection to book an appointment. User want to register first for book appointment. In the given time slot patient want to visit the hospital otherwise appointment will be cancelled.

9. Future Reference:

- Patient doesn’t need to book appoint by waiting in line.
- The details of doctor are already given they’re for book an appointment.
- Patient doesn’t need to purchase a case paper.
- The patient is able to select their own timeslot according to doctor availability.
- Doctor will check the daily appointment and make their schedule.

10. Environment:

10.1 Client-Side Hardware:

- RAM: 4GB
- Storage: 32 GB
- Operating System: Android 10+

10.2 Server-Side Hardware:

- RAM: 8GB
- CPU Speed: 5 GHZ
- Hard Disk: 1TB
- SSD: 128GB
- Operating System: Windows

10.3 Server-Side Technologies:

- **Frontend Technologies:**

1. Android
2. Java 18.0.1

- **Backend Technologies:**

1. Firebase

Place: Pune

Date: 28/10/2022

Sign of the Student	Sign of the Internal Guide	Sign of the Project Coordinator
Jay Dattatray Yele	**Prof. Rupali Pawar**	**Prof. Rupali Pawar**