

MEDICAL RECORDS MANAGEMENT

PARTH JOSHI(31455)
SAURABH DAREKAR(31461)
SUNVEG NALWAR(31464)
VARUN PATIL(31468)

Under the guidance of
Prof. Mayur Chavan

Abstract


Our Medical records management application involves a Doctor and a patient, including the patient's history, clinical findings, diagnostic test results, vaccinations and prescriptions. For the patient, the record can outline one's history and treatment plan in an easily-accessible way. For the Doctor, it can provide support about the correctness of that treatment plan.

Introduction

I hope we have briefed you about our project idea in the abstract. In this section, we will surf through the working of our management system.

The application will have 3 entities, each having separate interfaces to access the application. These 3 entities are patient, doctor and lab.

A patient can do the following things through the system:

- 
- Book a Doctor's appointment
 - Book a Lab appointment
 - View the Synopsis of a Doctor's consultation
 - View the prescriptions prescribed by a Doctor
 - View Lab reports
 - View/Update his Vaccination History



A doctor can do the following things through the system:

- Accept/Reject an appointment request.
- Upload the Synopsis and prescriptions of all of his consultations.
- View the Synopsis and prescriptions of all of his consultations.
- View Lab reports of all of his patients.
- View Vaccinations of all of his patients.




Similarly, a Lab can do the following through this system:

- Accept/reject an appointment request.
- Upload reports
- View all of its uploaded reports

Requirements

Mongo db- **MongoDB** is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

Android Studio- **Android Studio** is the official integrated development environment (IDE) for Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.



Android Volley- **Volley** is an **HTTP library** that makes networking very easy and fast, for Android apps. It manages the processing and caching of network requests and it saves developers valuable time from writing the same network call code again and again.

Flask- **Flask** is a web framework written in Python. Applications that use the Flask framework include Pinterest and LinkedIn.

REST API

- **Representational state transfer (REST)** is a software architectural style which uses a subset of HTTP. It is commonly used to create interactive applications that use Web services. A Web service that follows these guidelines is called *RESTful*. The primary or most-commonly-used HTTP verbs (or methods, as they are properly called) are POST, GET. POST is used to send data to a server to create/update a resource. GET is used to request data from a specified resource.

Working


The system stores the data in a MongoDB database.

The database can be accessed via a Flask server which uses REST API to handle HTTP requests.

An Android application is used as a GUI(Graphical User Interface) which makes use of Android Volley to communicate with the Flask server which in-turn interacts with the MongoDB database.

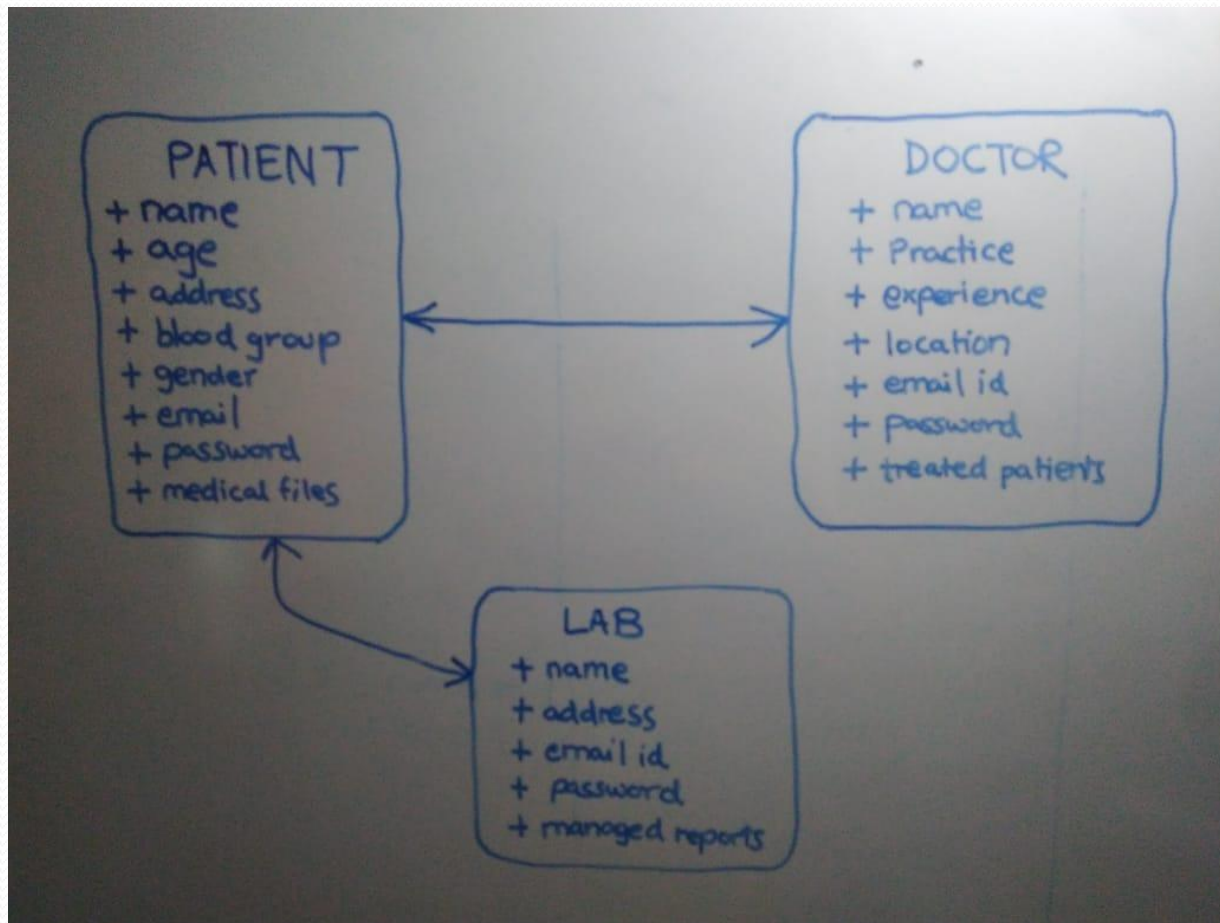
Scope

Not long ago, when a patient visited a medical office, the doctor took notes from the consultation and added them to a paper. Paper records are being replaced by electronic records which have several potential efficiencies and benefits . The transition to an all-electronic management of those records can be tricky, but not difficult to learn and eventually master.



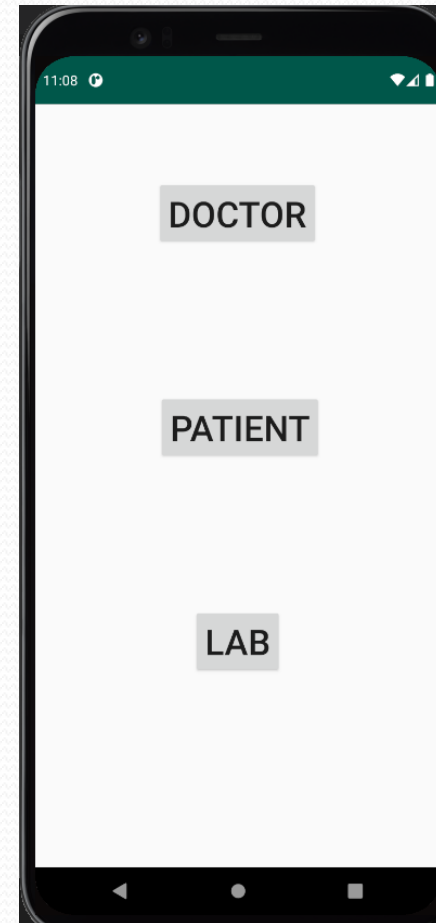
A records management system can also make it easy to transfer or release information between both patients and Doctors as they can access the information in a timely manner without duplicating efforts. Arguably, this type of visibility can increase patient safety, reduce mistakes, and increase confidence in a treatment plan.

Data Flow



Launch

This is the first activity users
See after launching the
application.



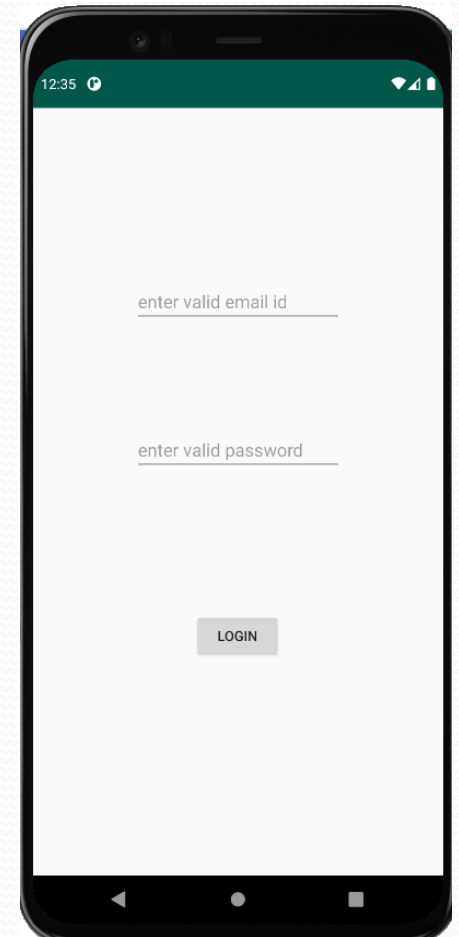
Login/Signup

Existing users can login using relevant credentials. New users need to signup in order to use the application.



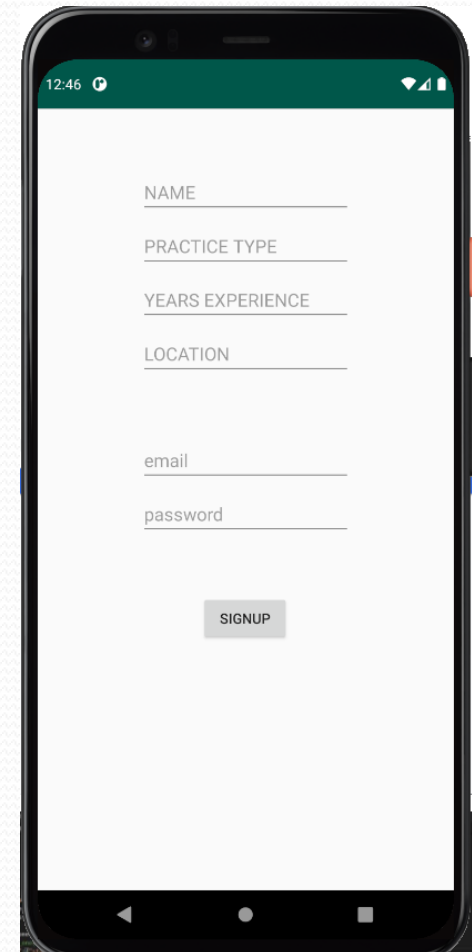
Login

This is the activity which an existing user has to go through in order to access his account.



Doctor Signup

This is the activity which a doctor sees when he signs up to the application.

A smartphone displaying a doctor signup form. The form is titled "Doctor Signup" and contains the following fields: NAME, PRACTICE TYPE, YEARS EXPERIENCE, LOCATION, email, and password. A SIGNUP button is located at the bottom of the form. The phone's status bar at the top shows the time 12:46 and various icons. The phone's navigation bar at the bottom shows the back, home, and recent apps buttons.

12:46

NAME

PRACTICE TYPE

YEARS EXPERIENCE

LOCATION

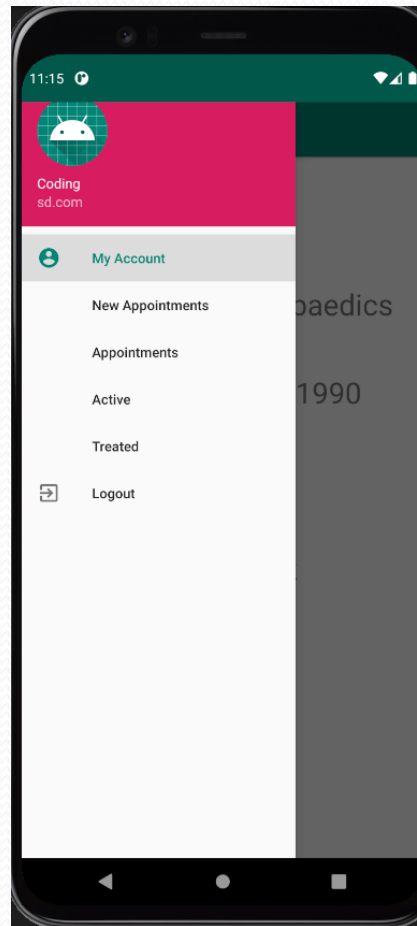
email

password

SIGNUP

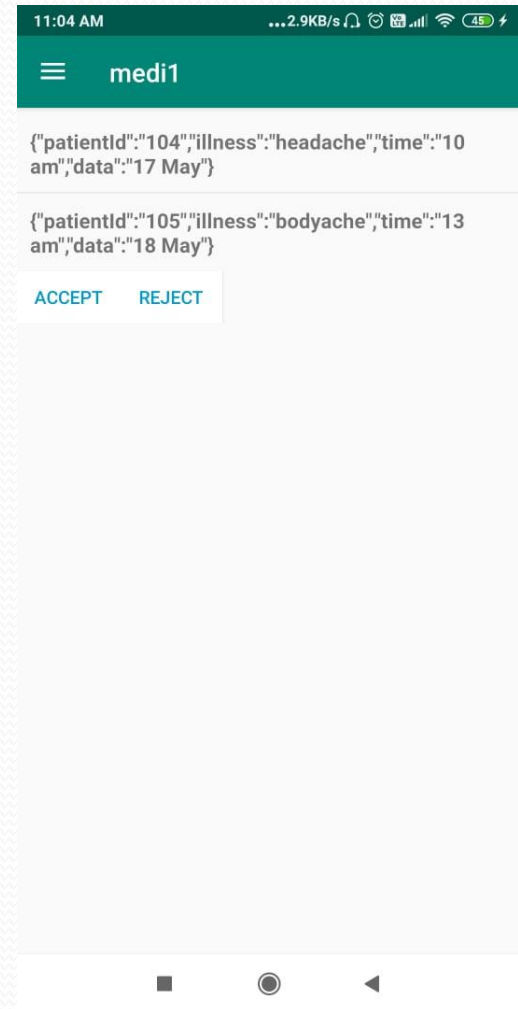
Doctor Launch

These are the activity where the patient lands post logging in and authenticating himself.



Appointment request

This is the activity where a doctor sees appointment requests sent by a patient. He has the option to accept/reject it.



Appointment list

This is the activity where a doctor sees a list of all appointments concerned with him/her. On clicking a Item prescription page opens. In prescription page doctor Has to enter medicine names and reports required.

The screenshot displays a mobile application interface with a dark green header bar containing the text "medi1" and a hamburger menu icon. Below the header, the left panel shows a list of appointments under the heading "Appointments". The list contains two items: {"patientId": "107", "Name": "Fracture"} and {"patientId": "108", "Name": "Diarrhoea"}. The right panel shows a prescription form under the heading "Prescriptions" with a "SUBMIT" button. The form includes a section for "Enter Medicines" with the text "paracetamol, fabiflu" and a section for "Reports:" with the text "Sugar, LHT".

11:04 AM ...2.8KB/s 11:05 AM ...1.5KB/s

medi1 medi1

Appointments

{"patientId": "107", "Name": "Fracture"}

{"patientId": "108", "Name": "Diarrhoea"}

Prescriptions SUBMIT

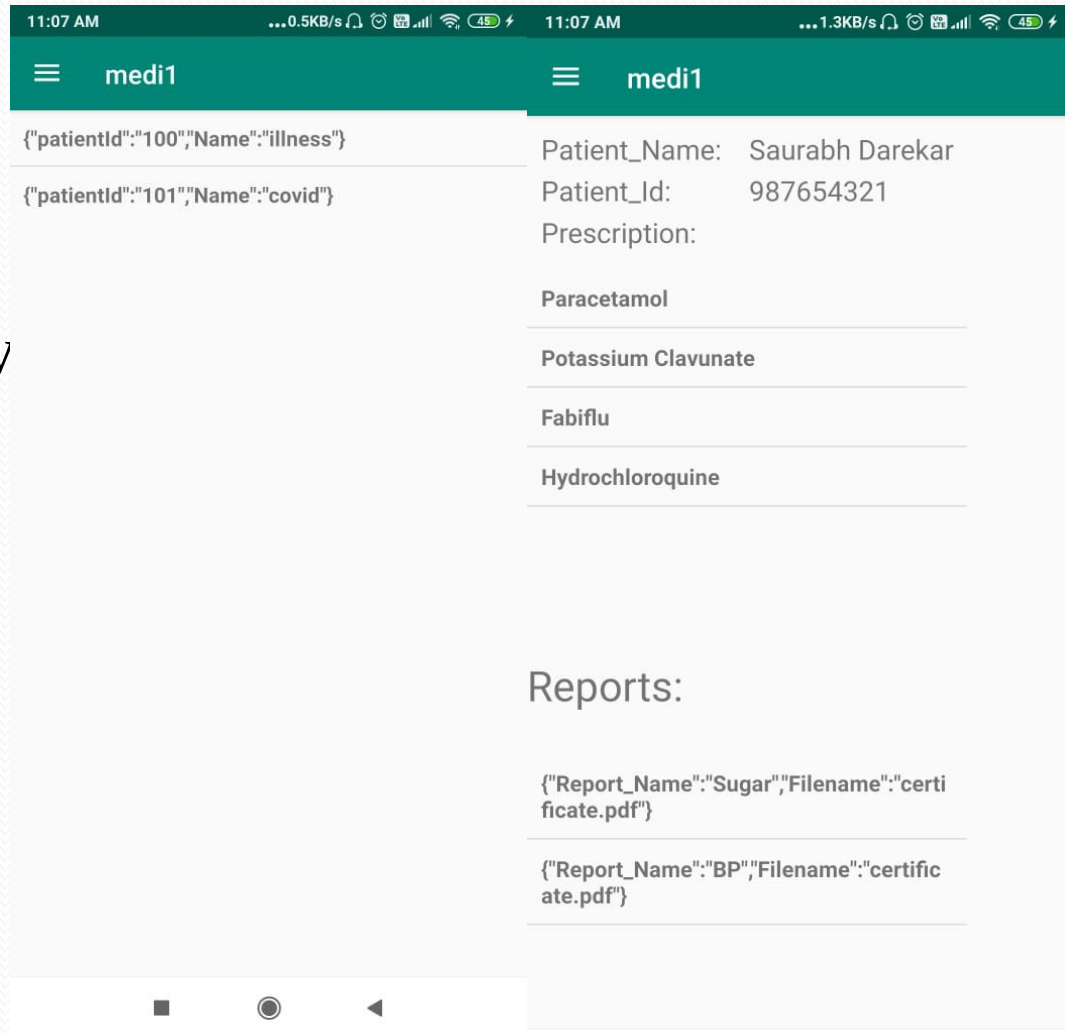
Enter Medicines

paracetamol, fabiflu

Reports: Sugar, LHT

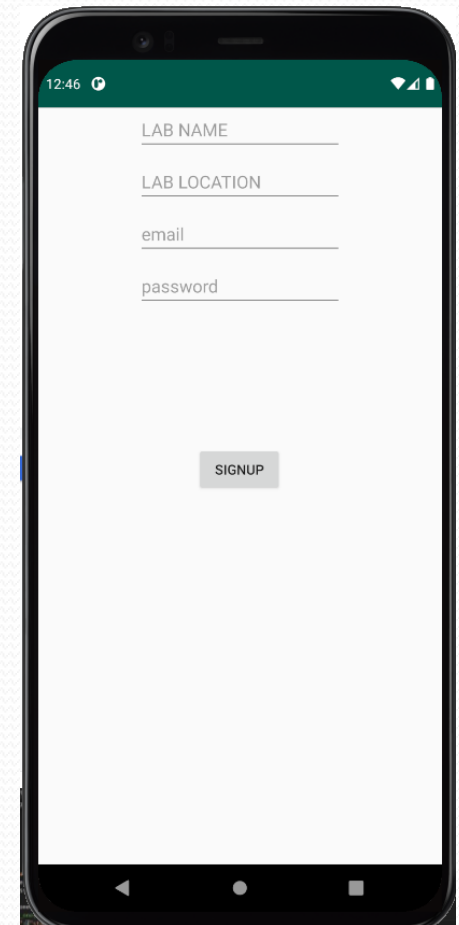
Treated Patients

- List of treated Patients.
- Onclick gets prescription given that patient.



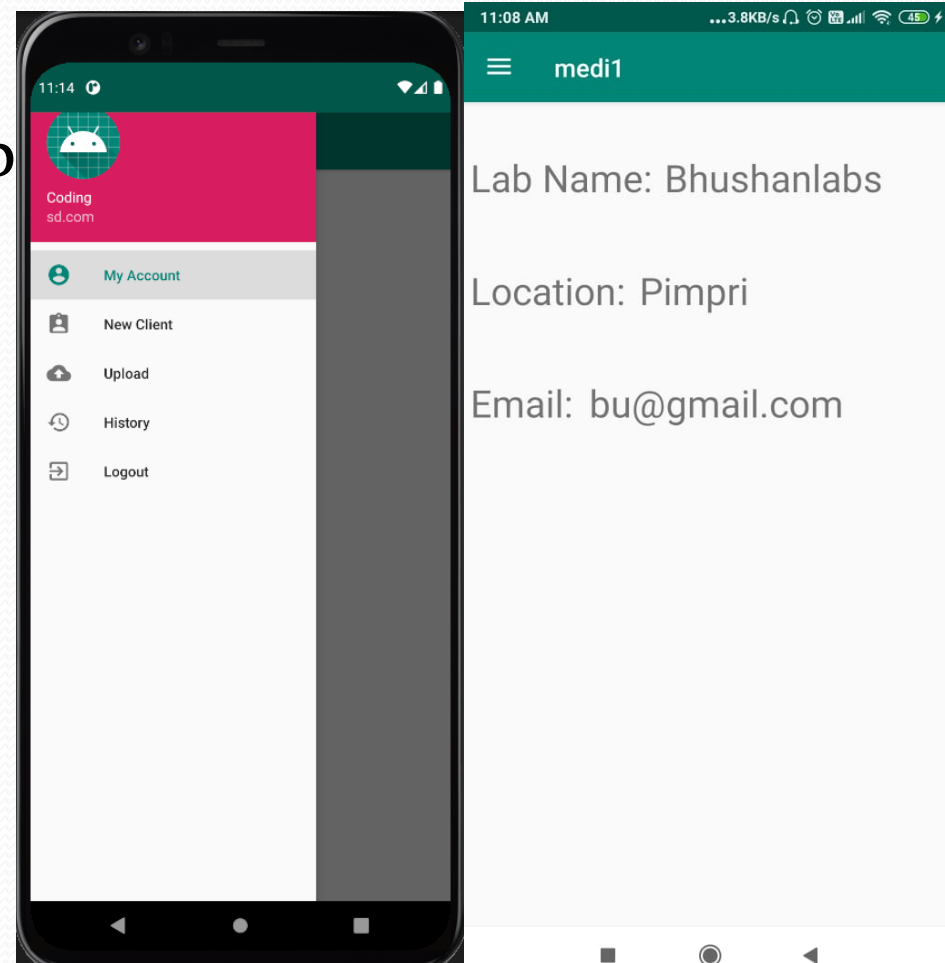
Lab Signup

This is the activity which a lab owner sees when he signs up to the application.



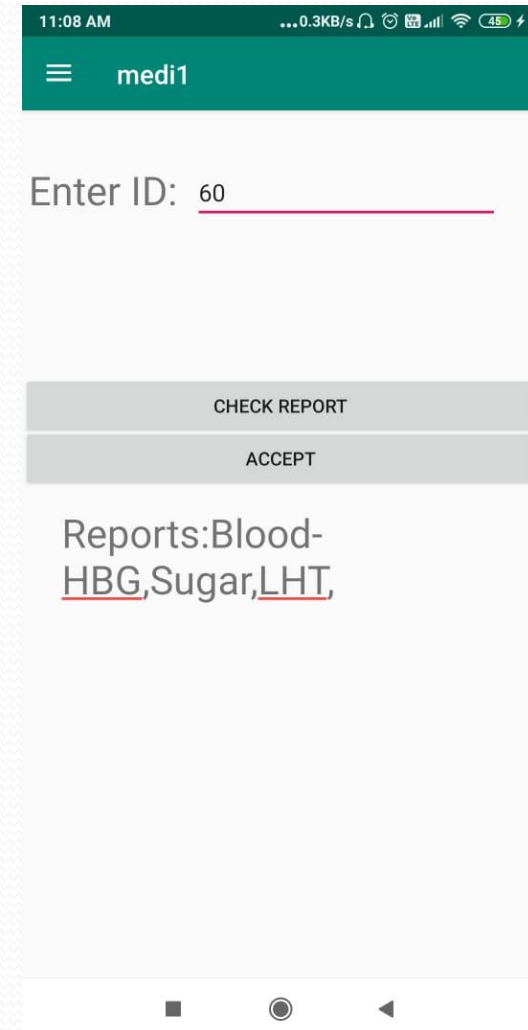
Lab launch

This is the activity which lab users see post logging in.



Accept report

This is the activity through which labs can accept appointments for diagnostics.



The screenshot shows a mobile application interface with a dark green header bar. The header bar contains a hamburger menu icon on the left and the text "medi1" on the right. Below the header bar, there is a text input field labeled "Enter ID:" with the value "60" entered. Below the input field, there are two buttons: "CHECK REPORT" and "ACCEPT". Below the buttons, there is a text label "Reports:Blood-HBG,Sugar,LHT". The bottom of the screen shows the Android navigation bar with three icons: a square, a circle, and a triangle.

11:08 AM ...0.3KB/s

medi1

Enter ID: 60

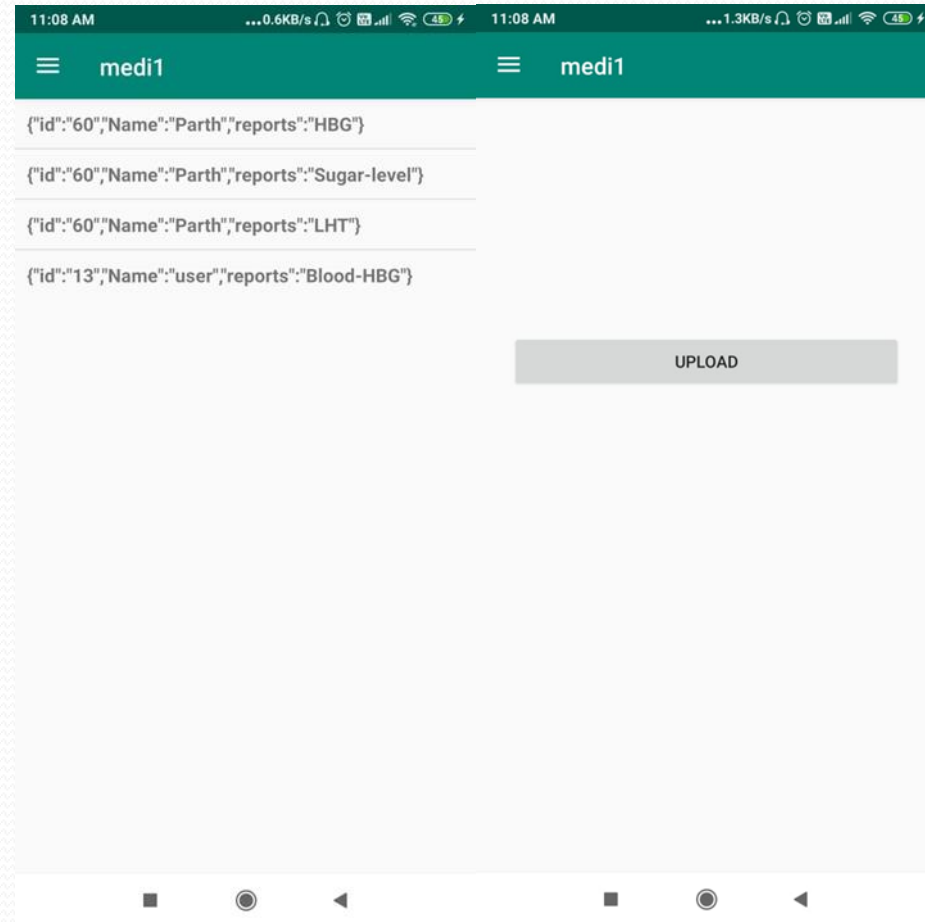
CHECK REPORT

ACCEPT

Reports:Blood-HBG,Sugar,LHT

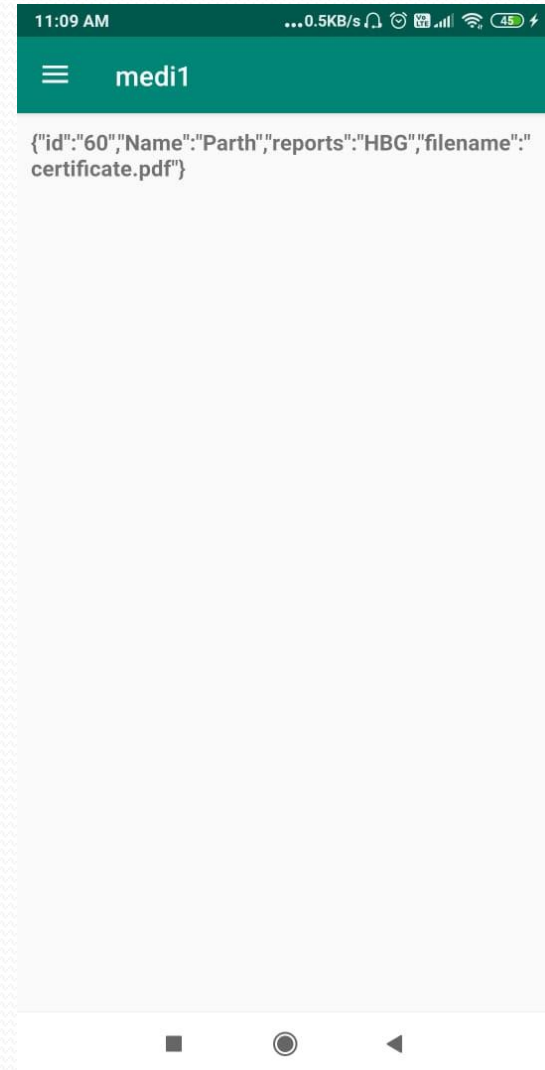
Upload Reports

- In this section there is list of patients whose report the lab user has to upload.
- On click next page opens where report is to uploaded.



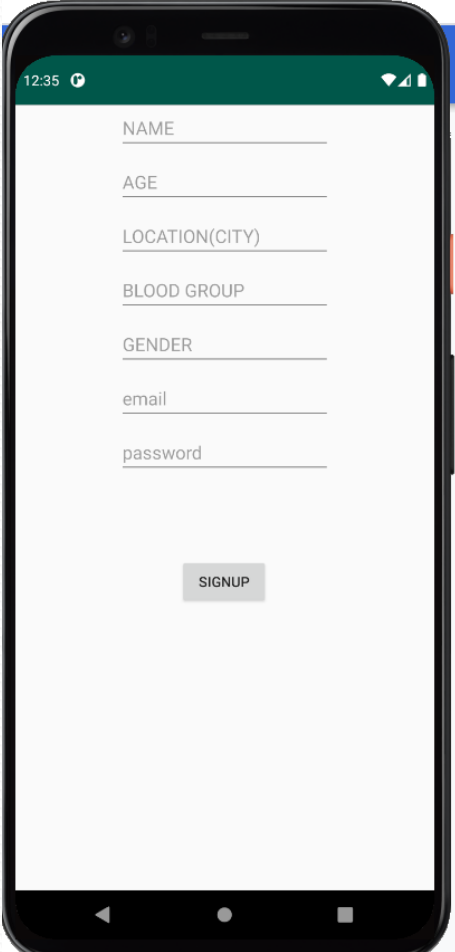
History

- In this section there is a list of all previous reports



Patient Signup

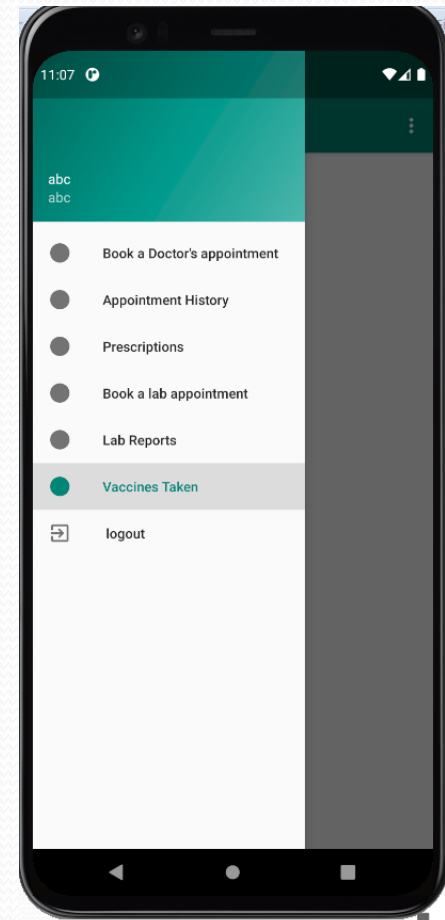
This is the activity which a patient sees when he signs up to the application. Similar activities are present for labs and doctors.



A smartphone displaying a patient signup form. The form has a green header bar with the time 12:35 and status icons. The form fields are labeled: NAME, AGE, LOCATION(CITY), BLOOD GROUP, GENDER, email, and password. A grey button labeled SIGNUP is at the bottom.

Patient Landing

This is the activity where the patient lands post logging in and authenticating himself.



Book Doctor's appointment

This is the activity where the Books/requests an appointment with a doctor.

11:07

DOCTOR'S ID

DOCTOR'S NAME

DATE(DD/MM/YYYY)

PREFERRED TIME(eg 16:)

Describe your illness/
problem

BOOK APPOINTMENT

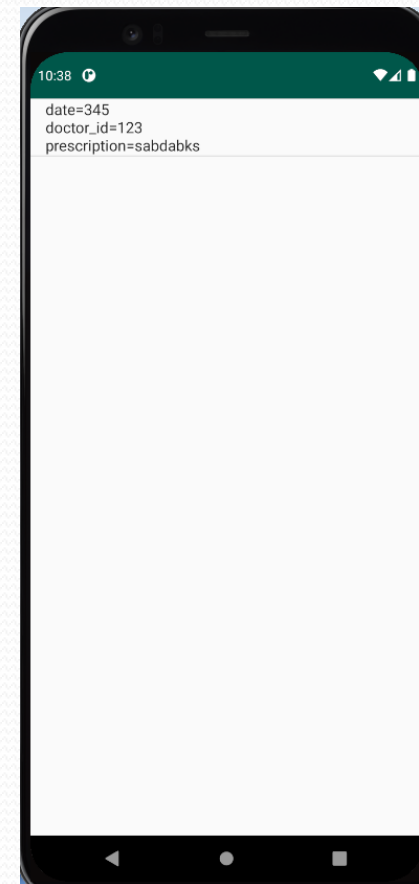
Vaccine list

In this activity, a patient can see the list of vaccines given to him.



Prescription list

In this activity, a patient can see the list of medicinal drugs prescribed to him



Scope for improvement

If given more time, we would like to improve the User Interface of the project because it lacks attractiveness. We would also like to refine the backend code so that the application runs smoother.

Lastly, we would also like to add a video calling feature with the help of which the patient can take consultation from the doctor without meeting him in person.

Conclusion

Practical implementation is lot more challenging compared to theorotical knowledge when it comes to application development. We were able to develop a near full stack application containing a server, a backend database and a frontend application helping us gain a lot of experience during the development of the project.

References

Android Documentation-

<https://developer.android.com/>

Mongodb documentation-

<https://docs.mongodb.com/>

Flask documentation-

<https://flask.palletsprojects.com/en/1.1.x/>