PUNE INSTITUTE OF COMPUTER TECHNOLOGY, DHANKAWADI,PUNE

A MINI-PROJECT REPORT

ON

HEAL

SUBMITTED BY

NAME: NIRAJ SARODE

ROLL NO: 31456

CLASS: TE4

GUIDED BY

Prof. R.V. Bidwe

COMPUTER ENGINEERING DEPARTMENT

Academic Year: 2019-20

I. ABSTRACT :

The aim of this project is to create a doctor-patient handling management system that will help doctors in their work and will also help patients to book doctor appointments and view medical progress. The system allows doctors to manage their booking slots online. Patients are allowed to book empty slots online and those slots are reserved in their name.The system manages the appointment data for multiple doctors of various dates and times. Each time a user visits a doctor his/her medical entry is stored in the database by the doctor. Next time a user logs in he may view his/her entire medical history as and when needed. At the same time, a doctor may view the patient's previous medical history while the patient visits him. The doctor is able to prescribe the patient online through the portal which will help other doctors for better diagnosis.

II. TABLE OF CONTENTS

III. LIST OF TABLES

IV. LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Motivation

Amit had to travel frequently to different places and often during his visits to different cities he was exposed to a new environment and would often catch a viral infection. Since places were new, it was difficult for him to get his condition diagnosed, he used to spend hours searching for a diagnosis center to get treated. And after so much trouble he used to only find a doctor who didn't have any idea about his previous prescriptions and allergies and used to prescribe him of his condition based on little information, which then worked in favor of treating his condition or the reverse.

All this trouble could have been saved if only he had used the Heal app. Since the heal app helps to find a nearby doctor and also keeps the data of all previous diagnoses handy.

1.2 Scope :

1. There are two basic users - Patients, and Doctors.
2. All users have their own profiles including doctors.
3. Each doctor can have its own timings of the clinic.
4. Patients can search for a doctor and make online appointments.
5. The patient, as well as the doctor, can manage the appointments booked.
6. Doctors can also view their previous health records, reports, and a prescription issued.
7. Doctors can give appointments, e-prescription and can view the patient's health records.
8. A patient can search for doctors near him.

1.3 Importance :

In today's fast-paced era, booking an appointment with the doctor is hectic thing. The long queue adds up to the frustration of the patient so rather than waiting in a long queue the patient can book an appointment to his/her doctor through the portal without any hassle.

Also, it is important for the doctor to know his previous symptoms and allergies to certain medicines and products. Through this portal, the doctor can view the patient's profile as well as previous records and prescriptions prescribed by any other day.

It is difficult in today's world to keep all the prescriptions and report handy all the time. So we had come up with an innovative idea to overcome this problem by storing the doctors' prescriptions and reports on the portal which can be used anywhere at any time for reference.

CHAPTER 2
DESIGN AND IMPLEMENTATION

2.1 Patient’s side :

1. With the help of Firebase Database, the application can maintain patients' data as well as other associated details.
2. The patient can search for various doctors based on location and his choice and book an appointment to the respective doctor.
3. Manage appointments that he has booked to various doctors.
4. A patient can book only one appointment to a doctor at one point of time. Once the doctor has consulted him or he has canceled the appointment with the respective doctor he can book another one.

2.2 Doctor’s side :

1. On the doctor's side, the doctor can manage his/her schedule by checking the appointments and managing them according to his/her convenience.
2. The doctor can prescribe the patient once the doctor has diagnosed the patient.
3. The doctor can also view the patients' profile and previous results and prescriptions of patients that are consulting him/her or consulted him/her.

2.3 Other:

1. If on any given date and time and appointment to a specific doctor already exist with another patient any other patient will not be able to book the same slot with the same doctor.
2. A doctor has to specify his/her clinic timings and slots of 30 minutes are created between those timings.

2.4 Use Case Diagram :

Representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

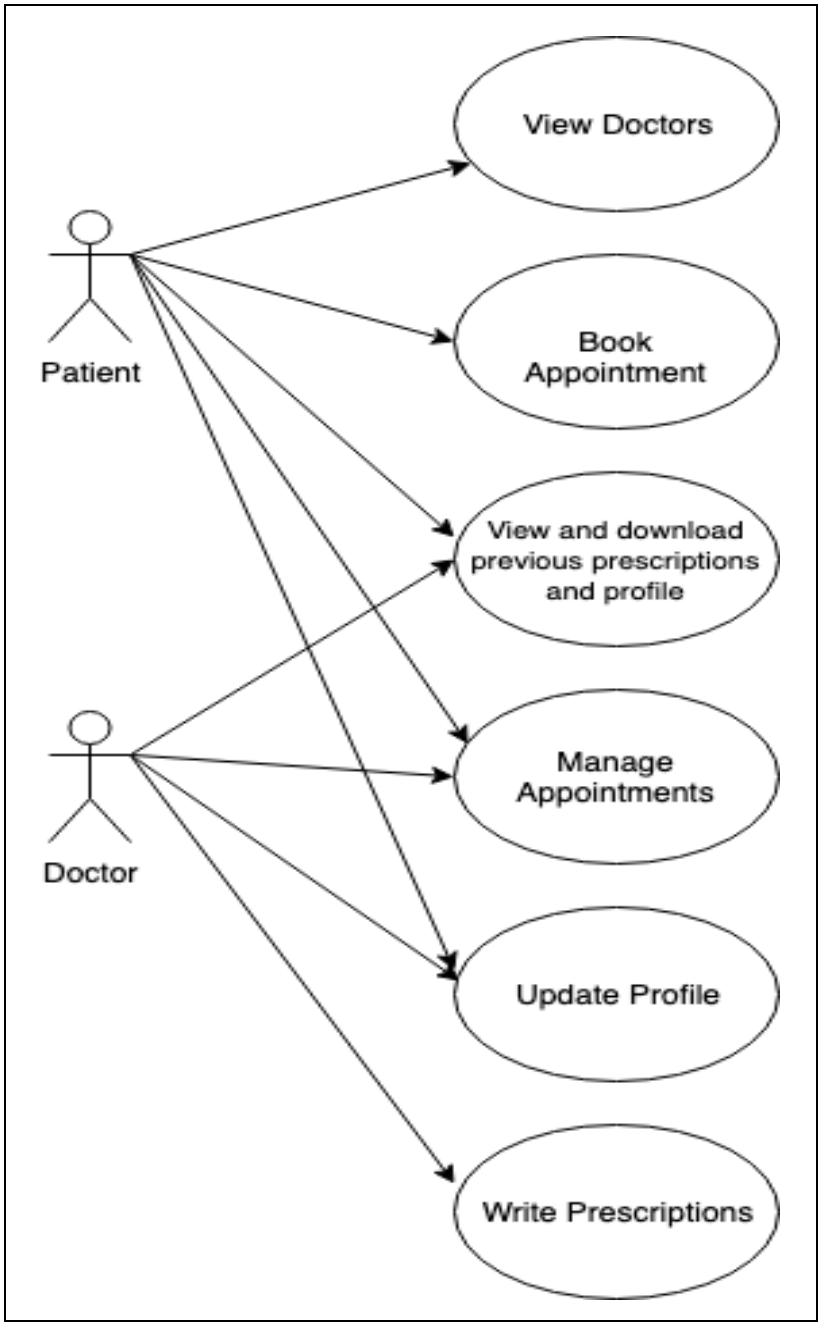


Fig: 2.4 Use Case Diagram

2.5 Workflow Diagram :

Graphic overview of the business process. Using standardized symbols and shapes, the workflow shows step by step how your work is completed from start to finish. It also shows who is responsible for work at what point in the process.

2.5.1 Patients' flow :

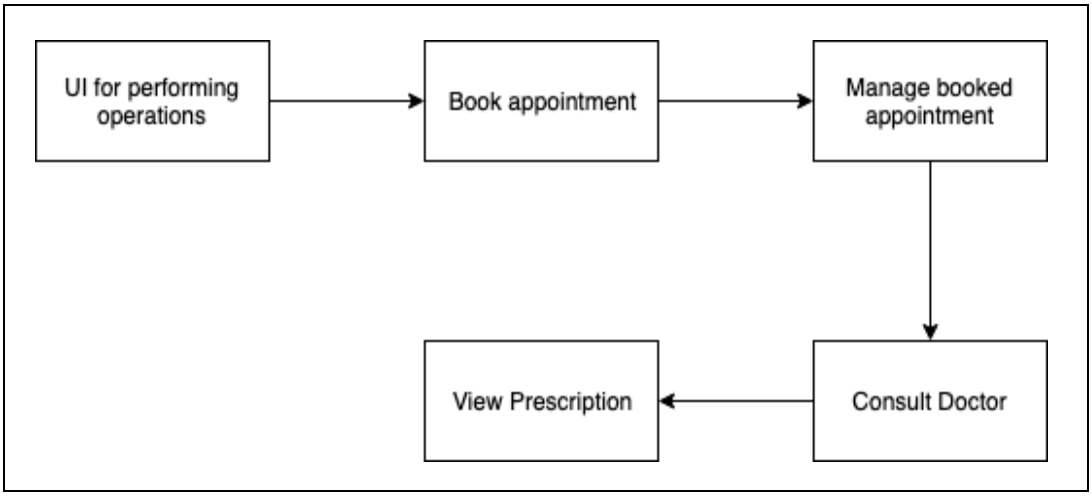


Fig: 2.5.1 Patients' flow

2.5.2 Doctors' flow :

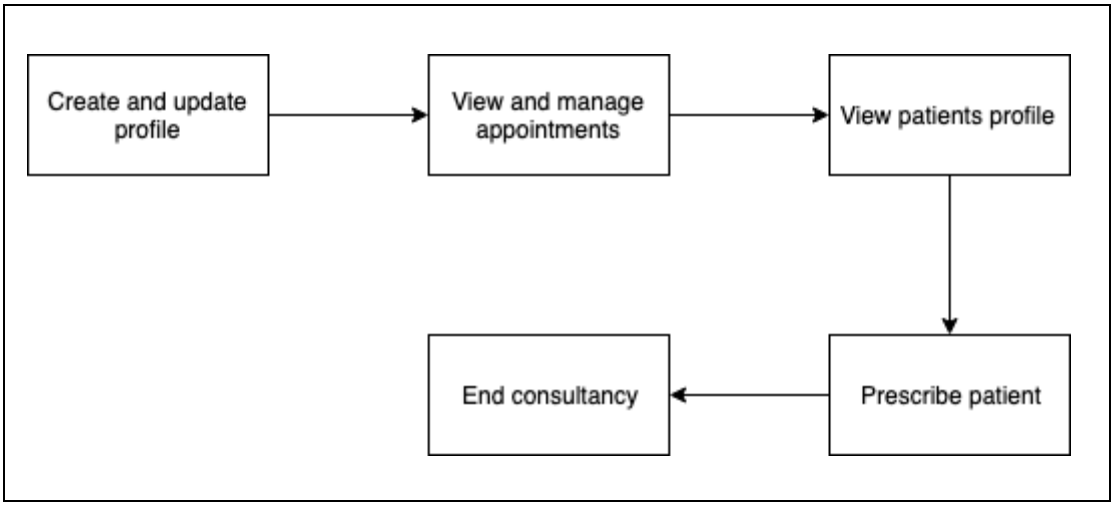


Fig: 2.5.2 Doctors' flow

2.6 System Architecture :

This gives a high level view of the new system with the main components of the system and the service they provide and how they communicate. The system is implemented using a three-tier architecture that comprises of our interface, process management and DBMS as illustrated below.

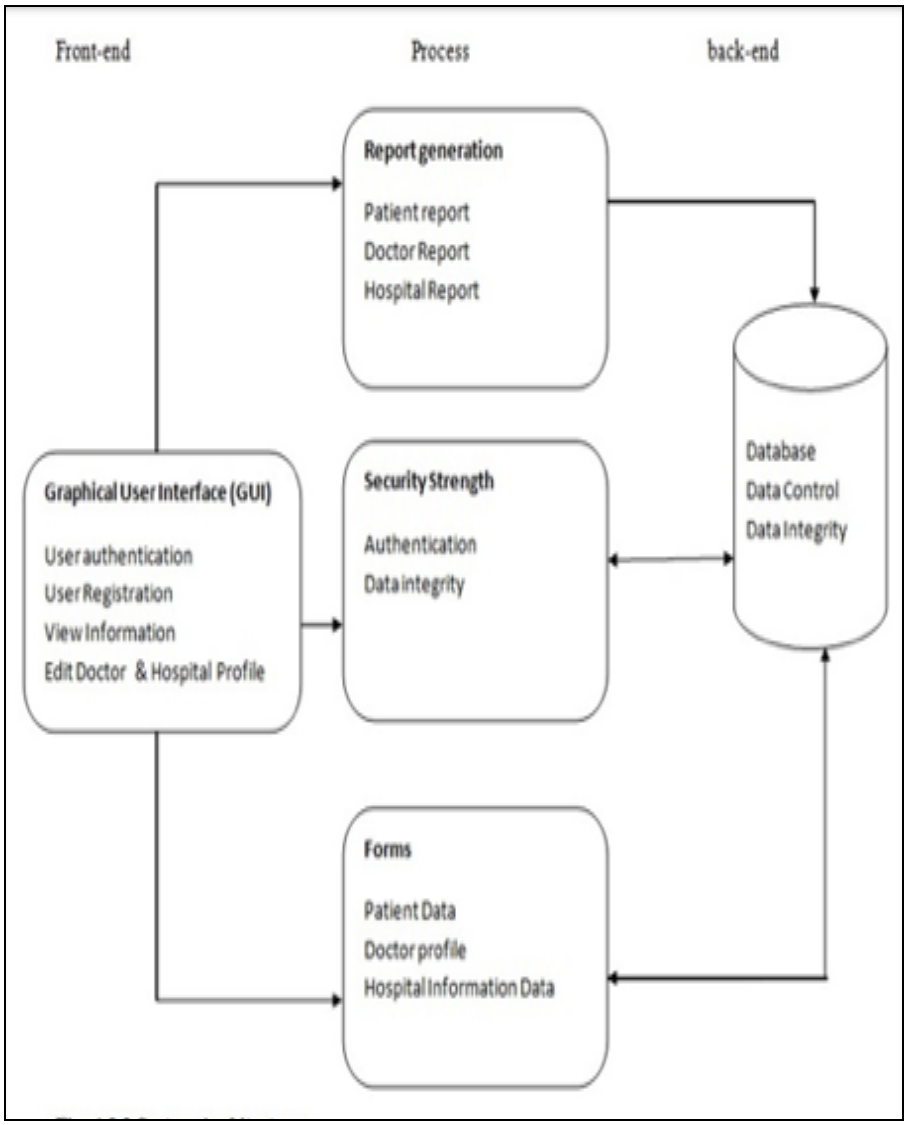


Fig: 2.6 System Architecture

CHAPTER 3
SYSTEM REQUIREMENTS

3.1 Android Studio Requirements :

- Linux OS requirements
    1. GNOME or KDE desktop. Tested on Ubuntu 14.04 LTS, Trusty Tahr (64-bit distribution capable of running 32-bit applications)
    2. 64-bit distribution capable of running 32-bit applications
    3. GNU C Library (glibc) 2.19 or later
    4. 3 GB RAM minimum, 8 GB RAM recommended (plus 1 GB for the Android Emulator)
    5. 2 GB of available disk space minimum, 4 GB recommended (500 MB for IDE plus 1.5 GB for Android SDK and emulator system image)
    6. 1280 x 800 minimum screen resolution

- Windows requirements
    1. Microsoft Windows 7/8/10 (32-bit or 64-bit)
    2. 3 GB RAM minimum, 8 GB RAM recommended (plus 1 GB for the Android Emulator)
    3. 2 GB of available disk space minimum, 4 GB recommended (500 MB for IDE plus 1.5 GB for Android SDK and emulator system image)
    4. 1280 x 800 minimum screen resolution

- The Android Emulator has additional requirements beyond the basic system requirements for Android Studio:
    1. SDK Tools 26.1.1 or higher
    2. 64-bit processor
    3. Windows: CPU with UG (unrestricted guest) support
    4. HAXM 6.2.1 or later (HAXM 7.2.0 or later recommended)

3.2 Hardware Requirements :

Sr. No.	Hardware	Minimum System Requirement
1	Memory	1 GB RAM
2	Processor	1 GHz
3	Disk Space	500 MB

Table 3.1 Hardware Requirements

3.3 Software Requirements :

Sr. No.	Software	Minimum System Requirement
1	Operating System	Android higher than 4.0.3
2	Wifi Support	802.11 a/b/g/n/ac
3	Wireless Mobile Telecommunication	3G or 4G/LTE

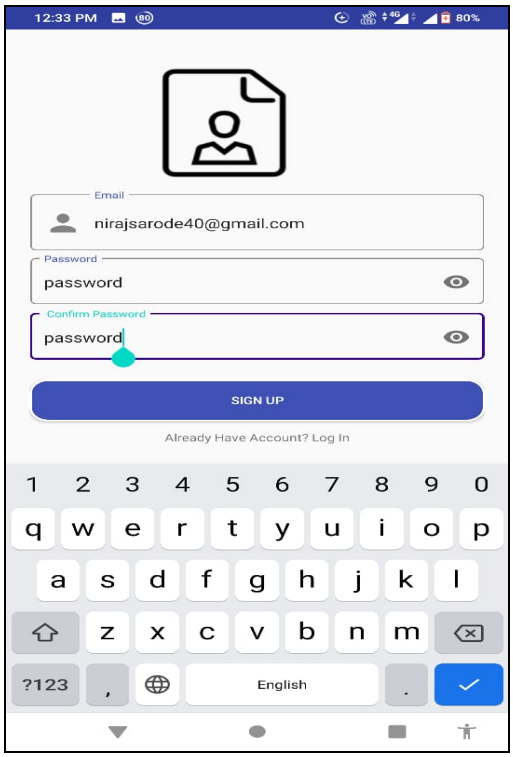
Table 3.2 Software Requirements

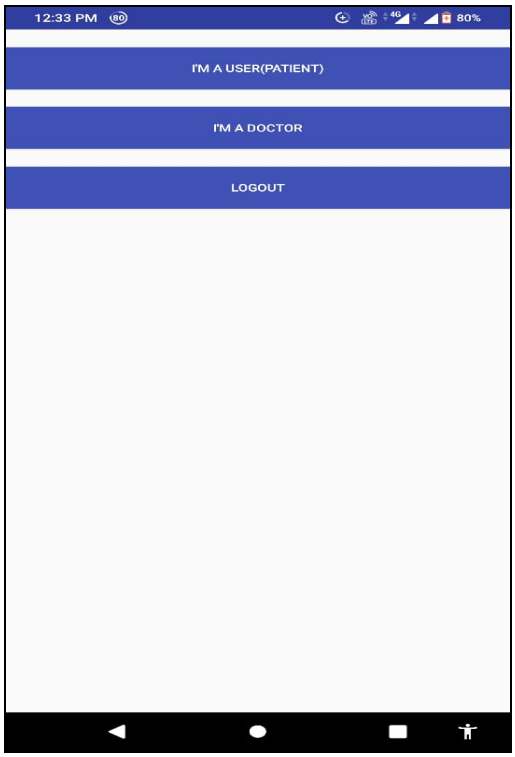
3.4 API's and Libraries :

1. Library adds support for the Action Bar user interface design pattern: implementation 'com.android.support:appcompat-v7:28.0.0'
2. Design dependencies : implementation 'com.android.support:design:28.0.0'
3. Support for the CardView widget: implementation 'com.android.support:cardview-v7:28.0.0'
4. Support for the RecyclerView widget: implementation 'com.android.support:recyclerview-v7:28.0.0'
5. FirebaseUI is an open-source library for Android that allows you to quickly connect common UI elements to Firebase APIs: implementation 'com.firebaseui:firebase-ui-firestore:4.3.1'
6. Dependency for the Firebase Authentication Android library: implementation 'com.google.firebase:firebase-auth:16.2.0'
7. Cloud Firestore Android library: implementation 'com.google.firebase:firebase-firestore:18.1.0'
8. Google maps Dependency: implementation 'com.google.android.gms:play-services-maps:16.1.0'
9. Google maps location Dependency: implementation 'com.google.android.gms:play-services-location:16.0.0'

CHAPTER 4
RESULT AND ANALYSIS

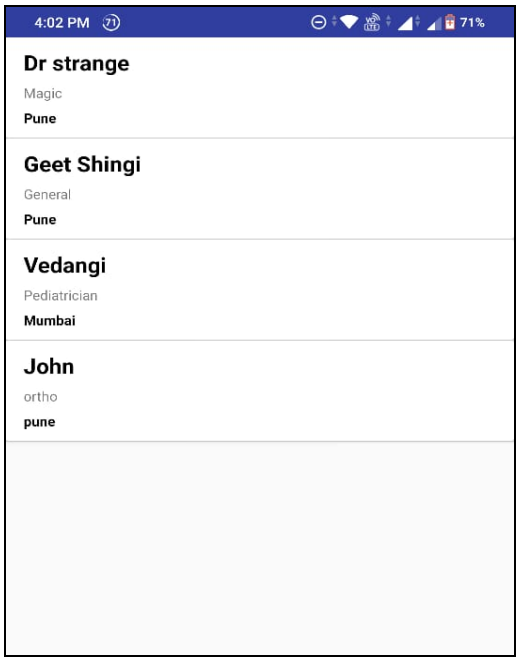
4.1 Login/Signup:

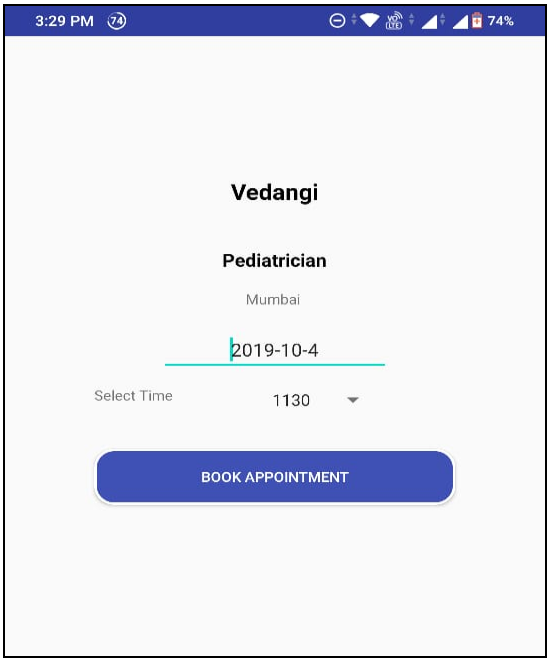




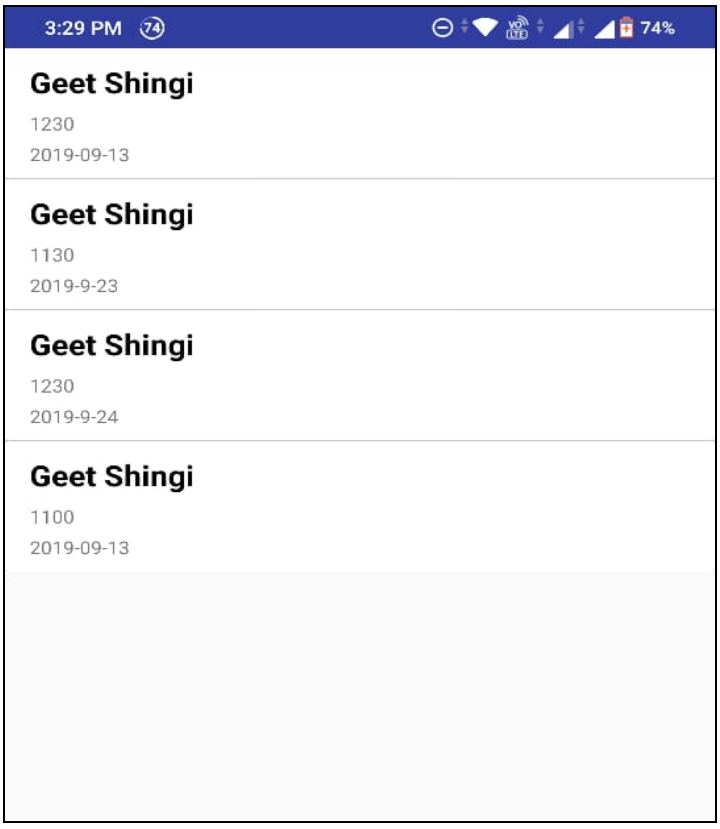
4.2 Patient:

4.2.1 Appointment Booking :

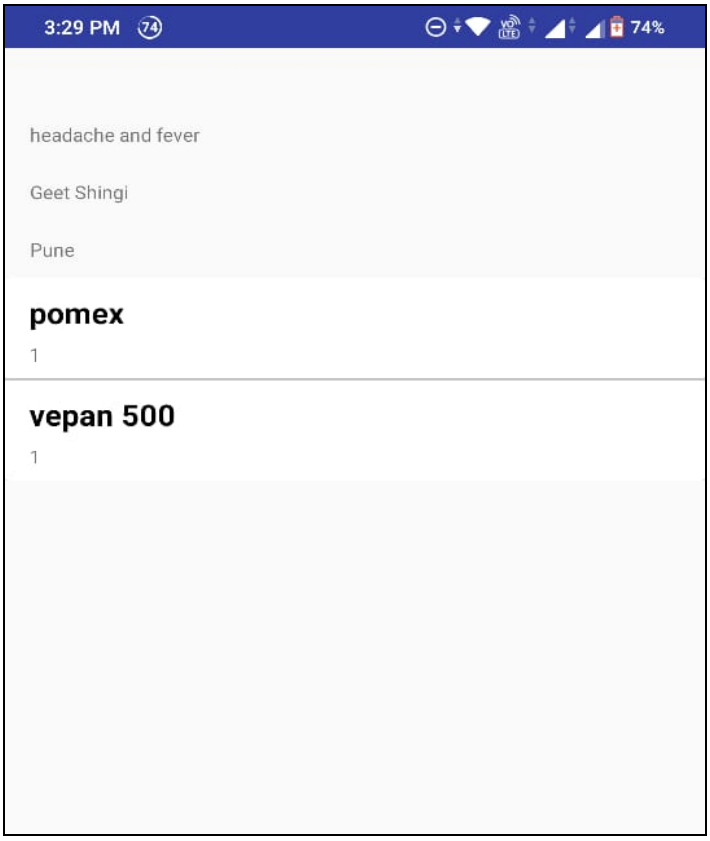




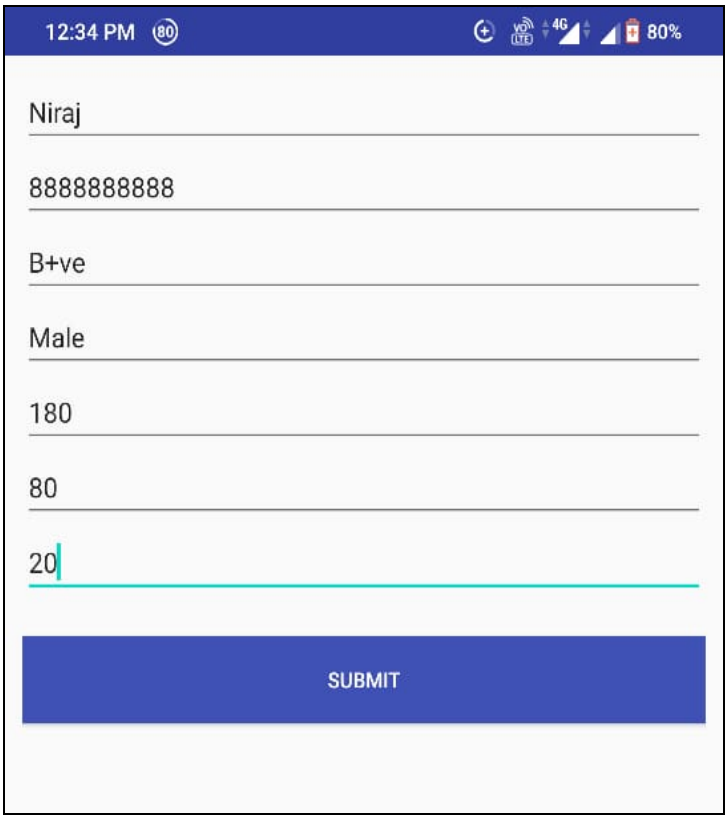
4.2.2 My Appointments :



4.2.3 My Prescriptions :

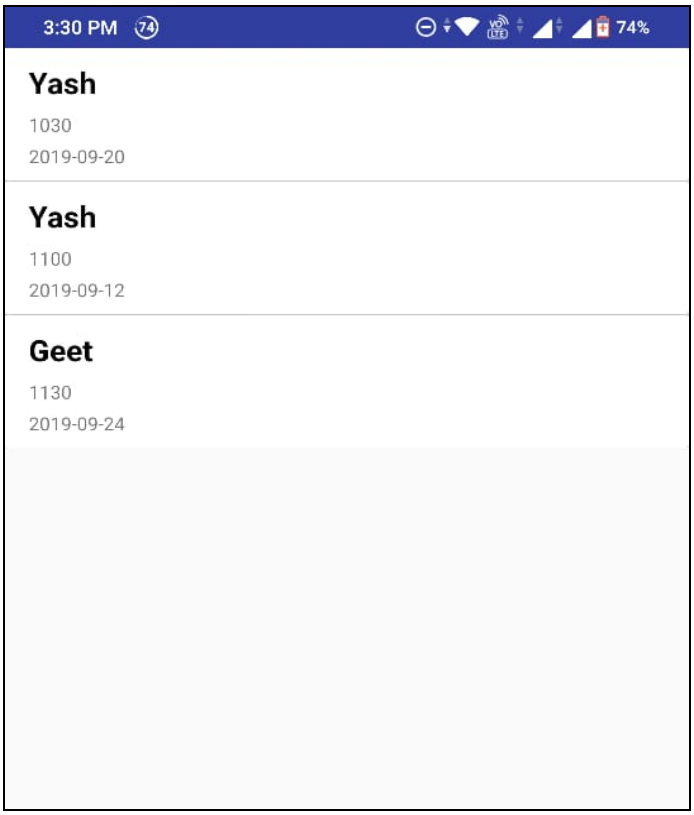


4.2.4 My Profile :

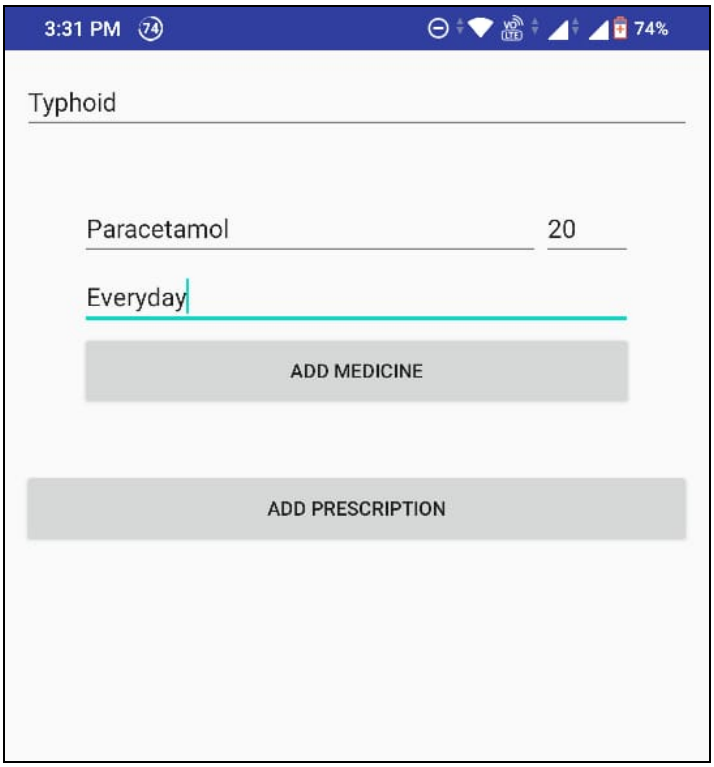


4.3 Doctor :

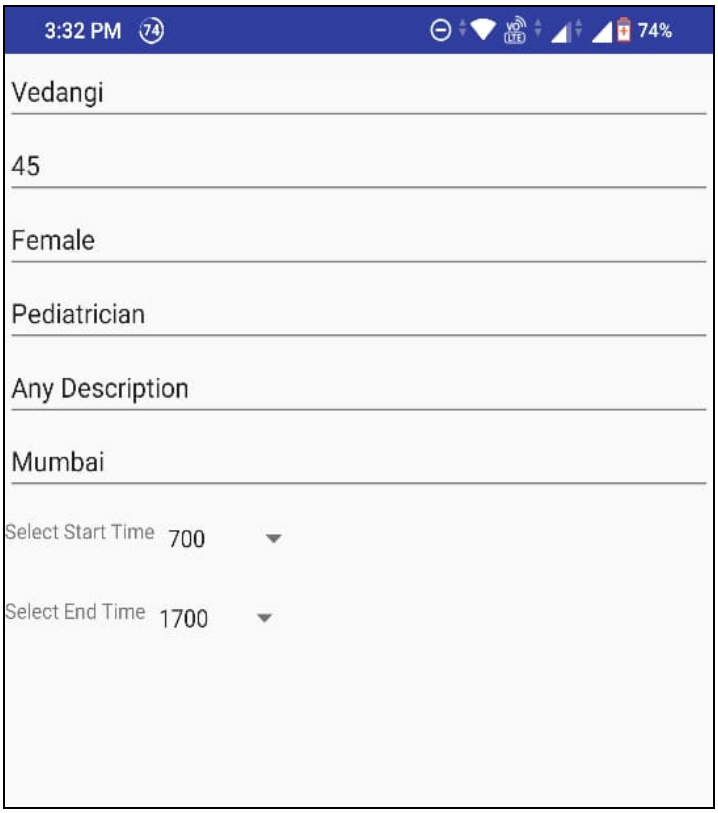
4.3.1 My Appointments :



4.3.2 Prescribe :



4.3.3 My Profile :



CHAPTER 5
CONCLUSION AND FUTURE ENHANCEMENTS

5.1 Future Enhancements :

1. Improvisation in User Interface
2. Integration of medical stores to buy medicines.
3. Chatbot to predict symptoms of the patient.
4. Rescheduling of appointments based on patients' location.

5.2 Conclusion :

The core reason for the establishment of computerizing Online Doctor Appointment System is to enable the hospital administrators in a convenient, fair and timely manner.

Therefore the IT used should support the core objective of the system if it is to remain relevant to the hospital. A lot still needs to be done in the IT department in order to make available technology effective. This may involve training of the hospital staff on how to enter data in the right and relevant data in the system and the management to keep updating the hardware and software requirements of the system. IT and computer systems need to be kept being upgraded as more and more IT facilities software are introduced in today's IT market. The researcher acknowledges the fact that this system does not handle all patient doctor and hospital. The researcher therefore suggests that for further research, the following can be researched on. The most cost effective way of handling all Hospital Patient management system process.

CHAPTER 6
REFERENCES

1. **Youtube. “Android Firestore Basics”. [Online] Available: https://www.youtube.com/playlist?list=PLGCjwl1RrtcRB0hvGQ_DixFoW0zSNf6O6**
2. **Youtube. “ Custom Buttons Design - Android Studio Tutorial”. [Online] Available: https://www.youtube.com/watch?v=nlPtfncjOWA**
3. **Cloud Firestore Documentation [Online] Available: https://cloud.google.com/firestore/docs/**
4. **Android Documentation [Online] Available: https://developer.android.com/docs**