

**CS581: Specialization Project**

**Android Medicare Application**

**by**

**Avatansh Awasthi (1947208)**

**Neil D’Souza (1947217)**

**Prakhar M P (1947250)**

**Under the guidance of Dr. Nismon Rio R**

**A Design Document submitted in partial fulfillment of the requirements for the award of the degree of Master of**

**Computer Applications of CHRIST (Deemed to be University) November – 2021**

****

CERTIFICATE

*This is to certify that the report titled* ***Android Medicare Application*** *is a bona fide record of work done by* ***Avatansh Awasthi (1947208), Neil D’Souza (1947217), Prakhar M P (1947250)*** *of CHRIST (Deemed to be University), Bangalore, in partial fulfillment of the requirements of V Semester CS581: Specialization Project during the year 2021.*

**Head of the Department Project Guide**

**Prof. Joy Paulose Prof. Nismon Rio R**

Valued-by:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Name | :Avatansh Awasthi |
| 1. |  | Register Number | :1947208 |
|  |  | Examination Centre | : CHRIST(Deemed to University) |
| 2. |  | Date of Exam | : |

**ACKNOWLEDGEMENTS**

Our project **“Android Medicare Application”** has come to fruition only because of the support and guidance of a lot of people. First and foremost, we would like to thank the Almighty for keeping us fit both physically and mentally, especially during trying times like this. The past few months have tested our endurance on many levels and we’re grateful for all the lessons that we have learned.

We would also like to express our gratitude towards **Prof. Joy Paulose, HOD, computer Science Department**, for giving us the opportunity to propose the idea of this project and supporting us with it. We are greatly indebted to our MCA **Course Coordinator, Prof. Tulasi**, for providing the opportunity to take up this project and for helping us with her valuable suggestions.

We sincerely thank our project guide, **Prof. Nismon Rio R,** for helping us with the project by giving timely suggestions and encouragement. All of our queries were answered without any hesitation or reluctance. we express our sincere thanks to **Dr. Ramesh Chandra Poonia, and Dr. Sarvanakumar**, lecturers of the Department of Computer Science, CHRIST (Deemed to be University), for their valuable suggestions during the course of this project. Their critical suggestions helped us to improve the project work.

We are also thankful to the **Teaching and Non-Teaching Faculty** of the Computer Science Department for their invaluable feedback and encouragement. We convey our sincere gratitude to our family members and our **friends, colleagues, parents** who directly or indirectly helped us in the successful completion of this project.

**ABSTRACT**

The application targets the gap that addresses the ease to access and facilitate various medical facilities. The Medicare application targets several unique features that make it different from existing applications in the market. The Medicare application recommends its users various hospitals and medical facilities and specific requirements according to their location and vicinity. The app stores the user data as his/her medical history for further use. The app will let the users book appointments with doctors as and when they require.

    If a person is new to a place and they are in urgent need of medical help, then a click would help tend to their needs. Thus saving time and effort. Ordering medicines from a clinic/pharmacy with proper prescriptions would bring medicines to the doorstep of the customer.

The major purpose of this application is to tend to the needs of the general public in ways of booking appointments or contacting doctors who might give an insight or a solution or even a procedure to deal with the condition at the moment. To help users who have gone to another location and are new to the area. This would help them in locating nearby hospitals, clinics in case they need one and have no clue as to where to find them

The ability to access the locations of the hospitals or clinics nearby, or be it contacting a doctor or booking an appointment with a specialist doctor will be possible through the application. In order to access the resources of the application, a person will have to create an account and then use the resources available to them. They will be given the freedom to add their medical records and history to the app so that it would be easier for doctors to view their medical history and suggest treatment accordingly.

**Table**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Table Name** | **Page No.** |
| 3.9.1 | User Table | 16 |
| 3.9.2 | Hospital Table | 16 |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Name** | **Page No.** |
| Fig-3.1 | Architecture Diagram | 9 |
| Fig-3.2 | Class Diagram | 10 |
| Fig-3.3 | Use Case Diagram | 11 |
| Fig-3.4 | Deployment Diagram | 12 |
| Fig-3.5.1 | DFD Level 0 | 13 |
| Fig-3.5.2 | DFD Level 1 | 14 |
| Fig-3.5.3 | DFD Level 2 | 15 |
| Fig-5.1 | Home Page | 47 |
| Fig-5.2 | Login | 48 |
| Fig-5.3 | Choose from options available | 49 |
| Fig-5.4 | Booking | 50 |
| Fig-5.5 | Booking Data Check Page | 51 |
| Fig-5.6 | Pharmacy | 52 |
| Fig-5.7 | Pharmacy -2 | 53 |
| Fig-5.8 | Maps Page | 54 |
| Fig-5.9 | Prescription Page | 55 |

**Table of Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| **Acknowledgements** | iii | **Acknowledgements** |  |
| **Abstract** | iv | **Abstract** |  |
| **List of Table** | v | **List of Table** |  |
| **List of Figures** | vii | **List of Figures** |  |

|  |  |
| --- | --- |
| 1. Introduction   1.1 Purpose  1.2 Scope  1.3 Overview | 1  1  2  3 |
| 1. System Requirements   2.1 Product Perspective  2.1.1 System Interfaces  2.1.2 Hardware Interfaces  2.1.3 Software Interfaces  2.1.4 Memory Constraints  2.2 Product Functions  2.3 User Characteristics  2.4 Constraints  2.5 Assumptions and Dependencies  2.6 Specific Requirements  2.6.1 External Interfaces  2.7 Performance Requirements  2.8 Logical Database Requirements | 4  4  4  4  5  6  6  6  6  7  7  7  8  8  8 |

|  |  |
| --- | --- |
| 1. Design Specification    1. Architecture Diagram    2. Class Diagram    3. Use case Diagram    4. Deployment Diagram    5. Data flow diagram       1. DFD Level 0       2. DFD Level 1       3. DFD Level 2    6. Database design       1. User Table       2. Hospital Table | 9  9  10  11  12  13  13  14  15  16  16  16 |
| 1. Implementation Details   4.1 Source Code | 18  18 |
| 1. Result and Discussion | 47 |
| 1. Conclusion   6.1 Advantages  6.2 Limitations  6.3 Future Enhancements | 56  56  56  57 |
| 1. References | 78 |

**1.** **INTRODUCTION**

The application focuses on the hole that tends to the straightforwardness to get to and work with different clinical offices. The Medicare application focuses on few one-of-a-kind highlights that make it not quite the same as existing applications on the lookout. The Medicare application suggests its clients to different emergency clinics and clinical offices and explicit necessities as indicated by their area and area. The application stores the client information as his/her clinical history for additional utilization. The application will allow the clients to book meetings with specialists as and when they require.

If an individual is new to a spot and they are in dire need of clinical assistance, then, at that point, a tick would help watch out for their necessities. In this way saving time and exertion. Requesting medications from a facility/drug store with appropriate remedies would carry meds to the doorstep of the client.

**1.1 PURPOSE**

The major purpose of this application is to tend to the needs of the general public in ways of booking appointments or contacting doctors who might give an insight or a solution or even a procedure to deal with the condition at the moment. To help users who have gone to another location and are new to the area. This would help them in locating nearby hospitals, clinics in case they need one and have no clue as to where to find them

**1.2 SCOPE**

The scope of this project is to provide a software-based solution to the customer patients which is able to clearly identify the need of assistance one wants i.e. whether he/she needs to consult with a doctor and book an appointment for the same, or one wants to search for a specific specialty of doctors or hospitals present in their area then we provide with the necessary details accordingly also any kind of medical facility in the locality could be found. Similarly for the doctors assistance, the customers' clinical history report generator is also included in which the customers history related to his illnesses or diseases or allergies can be kept which in turn will help the doctors to give a personalized suggestion in the treatment of the specific patient.

**1.3 OVERVIEW**

Prior to the Introduction, the table content is listed which shows how the software requirement system document is organized.

This is a working document, such that it is subject to change. In its initial form, it is incomplete by definition, and will require continuous refinement. Requirements may be modified and additional requirements may be added as development progresses and the system description becomes more refined. This information will serve as a framework for the current definition and future evolution of the Medicare app.

**2. SYSTEM REQUIREMENTS**

**2.1 PRODUCT PERSPECTIVE**

The product is a mobile application which is aiming for an easier and portable healthcare system for the user. The system is going to suggest nearby hospitals and medical stores using google maps api. User needs to login in and feed some details initially, and the app will suggest accordingly.

The system will be focusing on users’ present location and their specific requirements. The application will connect with the webservice, which will go through the Firebase database and connect to Google maps API, and will return the request to the user as shown in the diagram above. The database will be storing users’ data and also the previous searches and appointments booked.

**2.1.1 SYSTEM INTERFACE**

The system consists and interacts with the following hardware components and software. The system is made of the following hardware components and the following software platforms are used to create the project.

**2.1.2 HARDWARE INTERFACE**

The system has no hardware interface requirements.

**2.1.3 SOFTWARE INTERFACE**

**2.1.3.1 GOOGLE API**

Image to Audio conversion APIs from Google are available to convert the captured Image into Audio format. Google Cloud APIs are programmatic interfaces to Google Cloud Platform services. They are a key part of Google Cloud Platform, allowing you to easily add the power of everything from computing to networking to storage to machine-learning-based data analysis to your applications.

**2.1.3.2 FIREBASE**

Firebase is a product of Google which helps developers to build, manage, and grow their apps easily. It helps developers to build their apps faster and in a more secure way. No programming is required on the firebase side which makes it easy to use its features more efficiently. It provides services to android, ios, web, and unity. It provides cloud storage. It uses NoSQL for the database for the storage of data.

**2.1.4 MEMORY CONSTRAINTS**

* 512 MB RAM.

**2.2 PRODUCT FUNCTIONS**

The system will enable users to login or create an account where details will be stored in firebase. Therein the user can either search for hospitals in case of an emergency or book an appointment with the hospital or doctor close by or even purchase medicines from a pharmacy all in real time.

**2.3 USER CHARACTERISTICS**

The users are not limited by experience or technical expertise. Anyone with the basic knowledge of how to use a mobile/browser can access the app.

**2.4 CONSTRAINTS**

Listed below are the constraints for the system -

The emergency button can be upgraded in such a way that it can use the sensors like accelerometer, gravity, gyroscope to predict an accident and call the nearest hospital automatically. However, the exception to this is a lot in numbers, in cases such as a user casually throwing his phone on bed, jumping while having phone in pocket and so on. It is not actually possible to work on this feature.

The app needs internet connection and will not be able to function offline as both the Firebase and Google Maps API need internet connection.

Users need to connect with hospitals, doctors and medical stores (which are not registered on the app) manually.

**2.5 ASSUMPTIONS AND DEPENDENCIES**

The first assumption is that most people do have a hardware device with internet connectivity to access the website/application which is mandatory one to use the services provided.

The data provided by the users are valid and appropriately entered in the particular

fields.

The other assumption is that the logged in users will be able to use the app with

ease as the UI is designed to do so.

**2.6 SPECIFIC REQUIREMENTS**

**2.6.1 EXTERNAL INTERFACE**

**2.6.1.1** Basic Functioning

Although the android-based application does not need any external hardware to connect with, it does need some external interfaces as users have to connect with the hospitals, doctors and medical stores near to them.

In case, the hospitals, doctors and medical stores have not registered them on the app, users need to connect with them manually and thus external interfaces are required.

Call - to connect with hospitals, doctors and medical stores

Text Messenger - to send or receive message to/from hospitals, doctors and medical stores

Email - to contact hospitals, doctors and medical stores, usually email responses are slow.

Google Maps - to locate nearby hospitals, doctors and medical stores as per the requirement

**2.6.1.2** Create Account

To create a account that will enable the functionality of the app

The user will have to create an id and password while giving their mail id in order to continue

**2.6.1.3** Book appointment

The user will click the book button which will redirect to the booking page

There they will have to input the details asked and the hospital they choose to book an appointment with

**2.6.1.4** Purchase medicines

The user can purchase medicines from a pharmacy online

They can choose from the options available or they can search for the one they want

**2.6.1.5** Upload prescription

Option to upload their prescription is available

Multiple valid prescriptions can be uploaded at a time

**2.6.1.6** Locate hospitals

The user can search for hospitals around them

The maps API will help in locating hospitals around

**2.7 PERFORMANCE REQUIREMENTS**

**2.7.1** User should be able to operate a mobile

**2.7.2** Should have a stable internet connection

**2.7.3** Should input valid government document details

**2.8 LOGICAL DATABASE REQUIREMENTS**

**2.8.1** User details including authentication and email will be stored in the database

**2.8.2** Booking confirmation state will include

* User Name
* Age
* Gender
* Contact number
* Hospital

**3.0 Design Specification**

**3.1 Architecture Diagram**

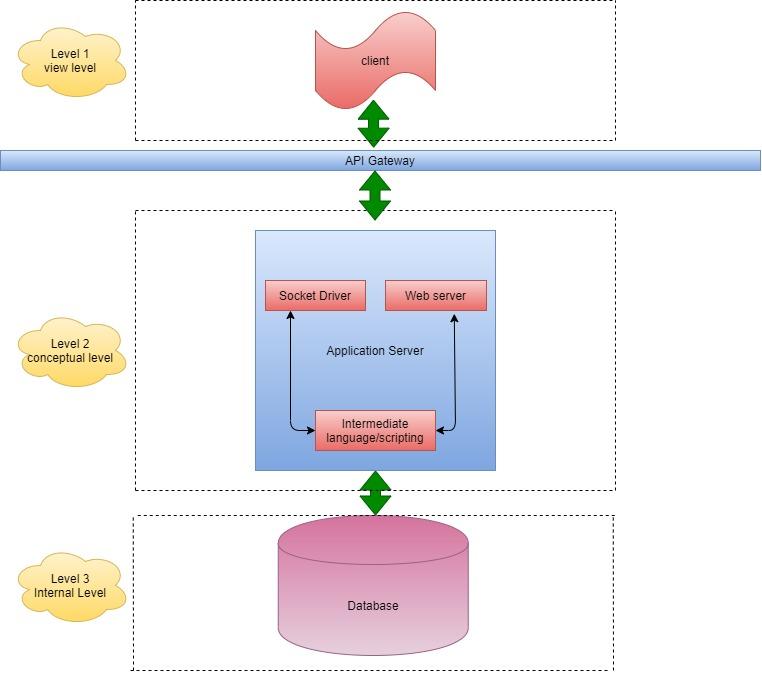
****

Fig-3.1 Architecture Diagram

Medicare app follow a three-tier architecture. The view level, conceptual level, internal level. The view level provides all the controls through which the client can interact with the system. At conceptual level includes all the services that are provided by the application. The Internal Level has the database where all the required information is stored.

**3.2** **Class Diagram**

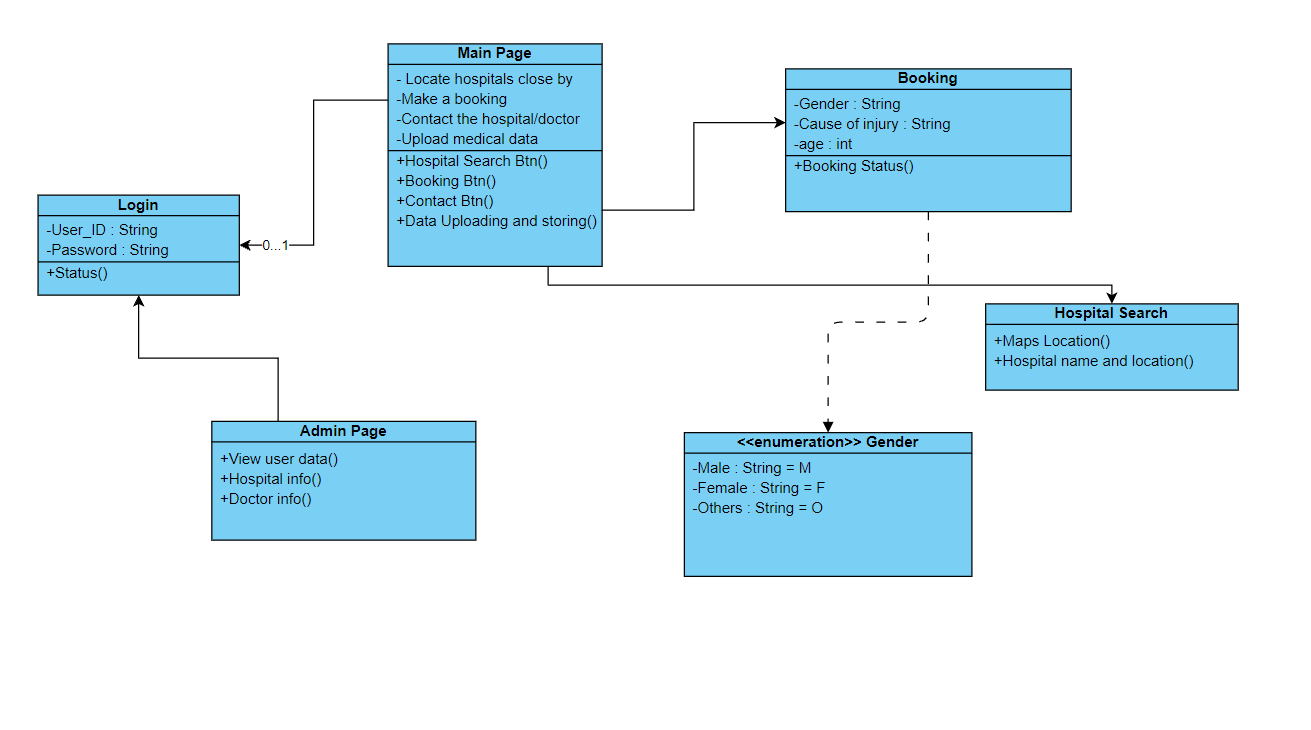
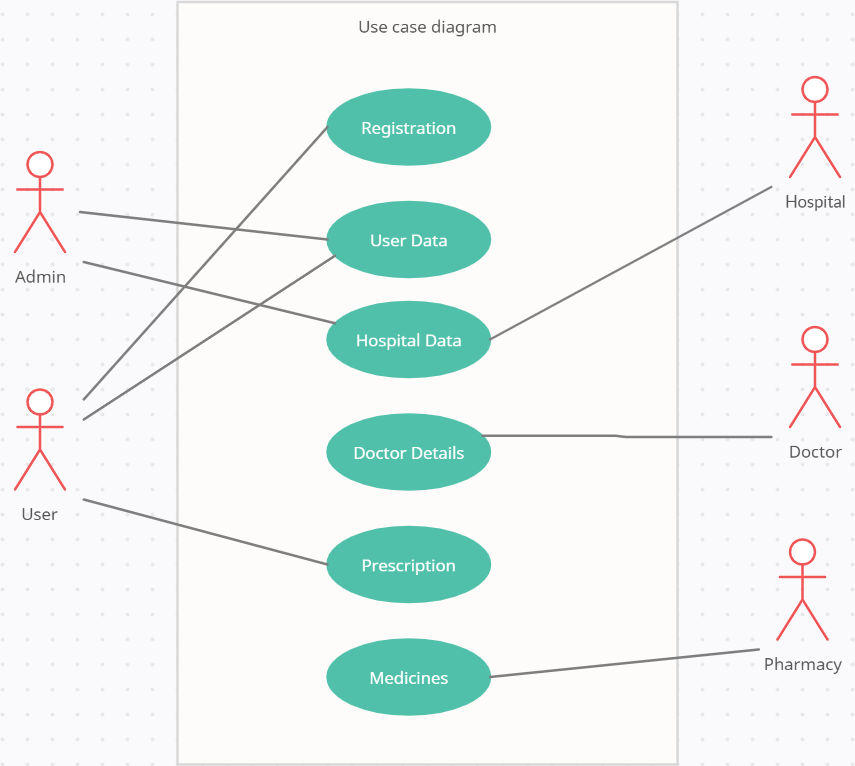
****

Fig-3.2 Class Diagram

The above class diagram describes the aspects of the app. First of all, the app and the user are the actual core two aspects of the application. They have a one-to-many relationship because one user can make more than one booking.

The User class here is the most important; it has attributes like the userID, the email details, the list of contact details. The user with a userID can log into the app and use the other functions of the app. They can make various bookings and purchase medicines with their ID. DoctosID as well as hospitalID will keep track of the bookings accepted by which hospital.

**3.3 Use Case Diagram**

****  
Fig-3.3 Use Case Diagram

The user first has to register to interact at all with the app. After registration, they are treated as a valid user. Each user can log in using google Identity. The user has the option to either to book or buy medicines depending upon their choice of action. They can either book an appointment with the hospital or doctor or they can purchase medicines online from a pharmacy. The option to upload a prescription is also available to them.

**3.4 Deployment Diagram**

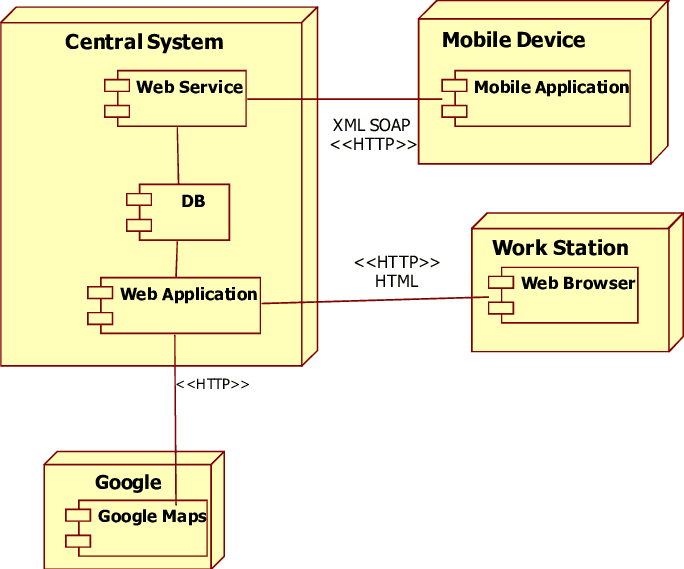


Fig-3.4 Deployment Diagram

The deployment of the application will be done using android studio and Firebase. Here, Android is being used for the frontend. The backend i.e. SQLite will be deployed on Firebase which will communicate with the application hosted on Android and provide quick and efficient results.

**3.5 Data Flow Diagram**

**3.5.1 DFD Level 0**

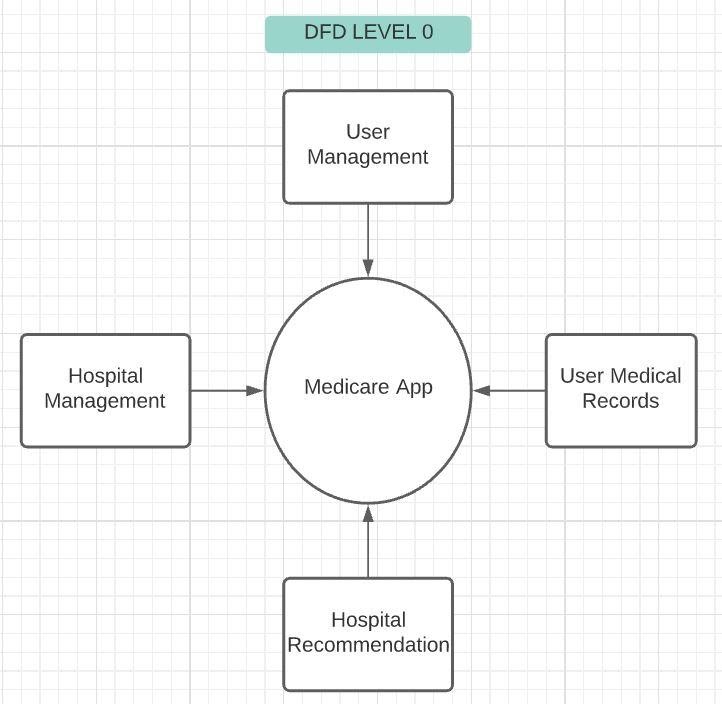


Fig-3.5.1 DFD Level 0

Throughout the application, the user will be able to create or upload medical records. The user will be able to decide to join an existing game or to create a new one. While creating the account, users will be able to modify the data they have entered while it will be cross checked by the admin.

**3.5.2 DFD Level 1**

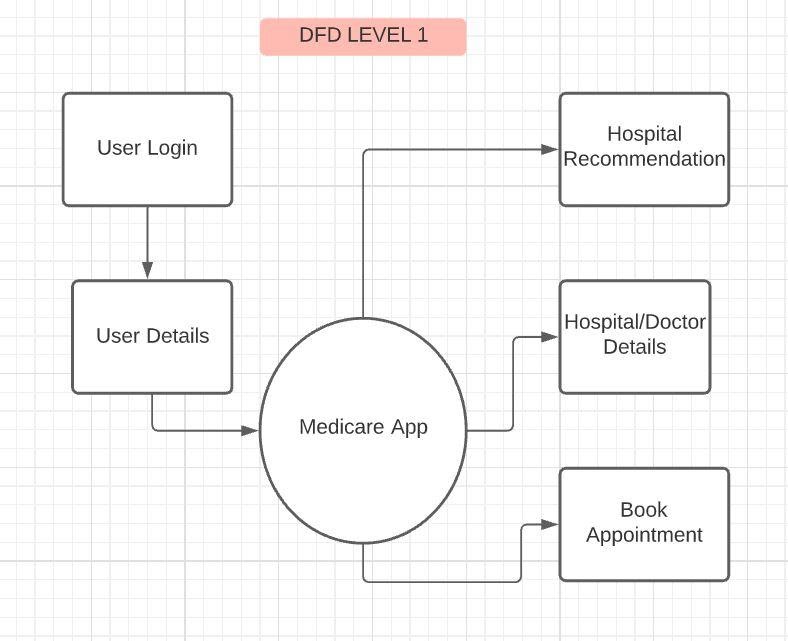


Fig-3.5.2 DFD Level 1

At this level, the user will be able to make bookings for either themselves or for another person say some other family member who doesn’t have the application at the moment. They can purchase medicines as well if they like so. The medical records will be accessed by the pharmacy owner if required before selling the medicines to them.

**3.5.3 DFD Level 2**

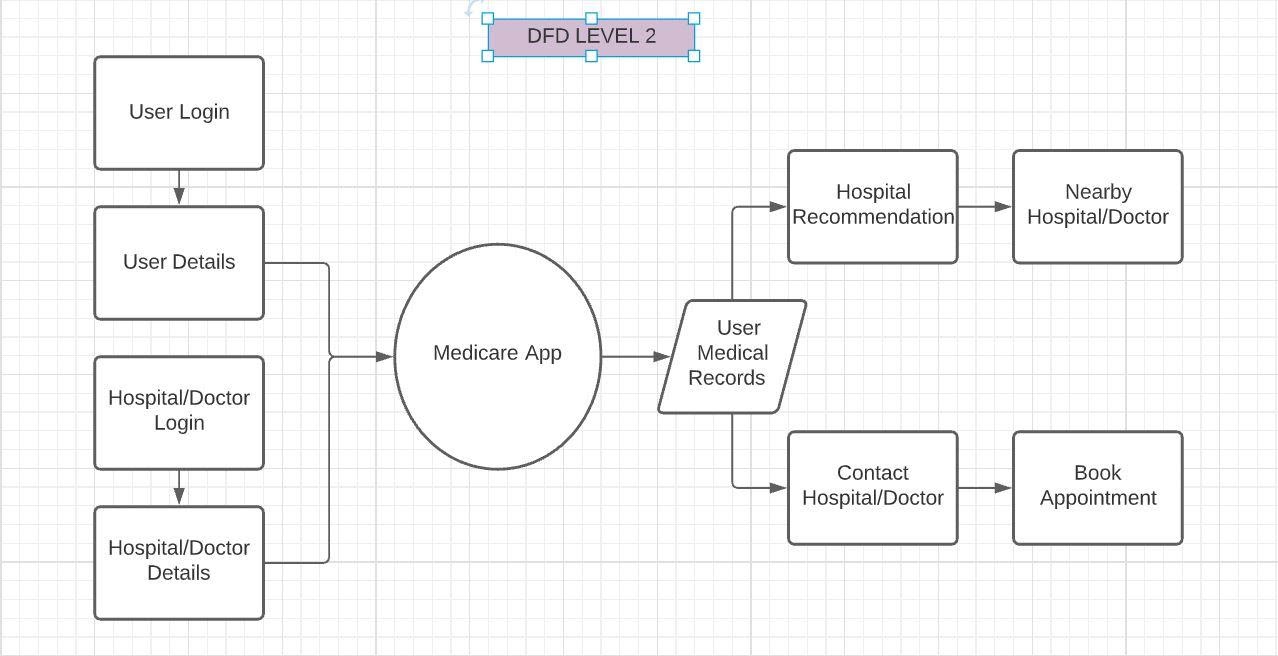


Fig-3.8.3 DFD Level 2

At this level, the admin can set out instructions to the hospitals or other doctors as well as the pharmacy just in case the user doesn’t enter valid details or has entered fake data. All the data will be available to the admin and certain data will be available to the hospitals as well as the doctors. The user can make and confirm their bookings if required even cancel the booking at a later stage in case they don’t want to carry on with the booked appointment.

**3.9 Database Design**

**3.9.1 User Table**

|  |  |  |
| --- | --- | --- |
| **Column** | **Datatype** | **Constraint** |
| UserID | Varchar(50) | PrimaryKey |
| Name | Varchar(50) | Not Null |
| emailID | Varchar(50) | Not Null |
| Contact | Nchar(10) | Not Null |

Table-3.9.1 Player Table

The user table contains the data of each user that joins the application. The user table has user details like the user’s unique id, name, the email id, and the contact. The table is used to make a list of all the users registered with the app. Using this table the number of users is monitored and other details are kept track of.

**3.9.2 Hospital Table**

|  |  |  |
| --- | --- | --- |
| **Column** | **Datatype** | **Constraint** |
| HospitalID | Varchar(50) | PrimaryKey |
| HospAdminTag | Varchar(50) | Not Null |
| DoctorID | Varchar(50) | Not Null |
| Hospital Contact | Nchar(10) | Not Null |

Table-3.9.2 Game Table

The hospital table contains attributes like the hospital id, the id of the admin of hospital, the list of doctors in the hospital with their respective tag/ID, and the contact number of the head of the hospital in charge.

**4.0 Implementation Details**

**4.1 Main**

**4.1.1 Main Page**

|  |
| --- |
| package com.example.medicare\_application;  import androidx.appcompat.app.AppCompatActivity;  import android.content.Intent; import android.os.Bundle; import android.view.View; import android.widget.Button; import android.widget.Toast;  public class MainActivity extends AppCompatActivity {  private Button button\_open;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_main*);  button\_open = (Button) findViewById(R.id.*button*);   button\_open.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Toast.*makeText*(getApplicationContext(), "Welcome", Toast.*LENGTH\_SHORT*).show();  nextPage();  }  });  }  private void nextPage(){  Intent intent = new Intent(this,Login.class);  startActivity(intent);  } } |

**4.1.2 MainPage.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"  android:background="@drawable/logo\_bg"  tools:context=".MainActivity">   <ImageView  android:id="@+id/imageView"  android:layout\_width="117dp"  android:layout\_height="84dp"  android:layout\_marginStart="1dp"  android:layout\_marginTop="2dp"  app:layout\_constraintEnd\_toStartOf="@+id/textView"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent"  app:srcCompat="@drawable/logo" />   <TextView  android:id="@+id/textView"  android:layout\_width="222dp"  android:layout\_height="49dp"  android:layout\_marginTop="63dp"  android:layout\_marginEnd="71dp"  android:text="Medicare App"  android:textAlignment="textStart"  android:textColor="@color/white"  android:textSize="32sp"  android:textStyle="bold|italic"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toEndOf="@+id/imageView"  app:layout\_constraintTop\_toTopOf="parent" />   <Button  android:id="@+id/button"  android:layout\_width="216dp"  android:layout\_height="47dp"  android:layout\_marginStart="96dp"  android:layout\_marginTop="128dp"  android:layout\_marginEnd="99dp"  android:text="Proceed"  android:textColor="@android:color/holo\_orange\_light"  android:textSize="20sp"  android:textStyle="bold"  app:backgroundTint="@color/material\_on\_background\_disabled"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="0.0"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/textView" />  </androidx.constraintlayout.widget.ConstraintLayout> |

**4.1.3 Login**

|  |
| --- |
| package com.example.medicare\_application;  import androidx.appcompat.app.AppCompatActivity;  import android.content.Intent; import android.os.Bundle; import android.view.View; import android.widget.Button; import android.widget.EditText; import android.widget.TextView; import android.widget.Toast;  public class Login extends AppCompatActivity {  private EditText mEdit, pEdit;  private Button loginButton;  TextView t;  int count = 5;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_login*);  mEdit = (EditText)findViewById(R.id.*user\_input*);  pEdit = (EditText)findViewById(R.id.*pass\_input*);  loginButton = (Button)findViewById(R.id.*button*);  t = (TextView)findViewById(R.id.*message*);  t.setVisibility(View.*GONE*);   loginButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  if(mEdit.getText().toString().equals("neildsouza98@gmail.com") &&  pEdit.getText().toString().equals("kingfisher.1")) {  Toast.*makeText*(getApplicationContext(), "Congrats, you hit the mark", Toast.*LENGTH\_SHORT*).show();  openSecondActivity();  }  else {  Toast.*makeText*(getApplicationContext(),"Wrong Input Buddy..", Toast.*LENGTH\_SHORT*).show();  t.setVisibility(View.*VISIBLE*);  count--;  t.setText(Integer.*toString*(count));   if(count == 0){  loginButton.setEnabled(false);  }  }  }  });   }   private void openSecondActivity() {  Intent intent = new Intent(this, MainPage.class);  startActivity(intent);  } } |

**4.1.4 Login.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=".Login">   <TextView  android:id="@+id/header"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginStart="158dp"  android:layout\_marginTop="88dp"  android:layout\_marginEnd="165dp"  android:text="Welcome"  android:textColor="@color/teal\_700"  android:textSize="30sp"  android:textStyle="bold"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="0.466"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent" />   <TextView  android:id="@+id/user\_name"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginStart="78dp"  android:layout\_marginTop="192dp"  android:layout\_marginEnd="275dp"  android:text="Username"  android:textColor="@android:color/holo\_orange\_dark"  android:textSize="20sp"  android:textStyle="bold"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="0.09"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent" />   <EditText  android:id="@+id/user\_input"  android:layout\_width="318dp"  android:layout\_height="39dp"  android:layout\_marginStart="77dp"  android:layout\_marginTop="20dp"  android:autofillHints=""  android:textSize="20sp"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/user\_name" />   <TextView  android:id="@+id/passwd"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginStart="76dp"  android:layout\_marginTop="36dp"  android:text="Password"  android:textColor="@android:color/holo\_orange\_dark"  android:textSize="20sp"  android:textStyle="bold"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/user\_input" />   <EditText  android:id="@+id/pass\_input"  android:layout\_width="318dp"  android:layout\_height="39dp"  android:layout\_marginStart="77dp"  android:layout\_marginTop="32dp"  android:layout\_marginEnd="16dp"  android:importantForAutofill="no"  android:textSize="20sp"  android:inputType="textPassword"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="0.0"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/passwd" />   <Button  android:id="@+id/button"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginStart="79dp"  android:layout\_marginTop="36dp"  android:layout\_marginEnd="216dp"  android:text="Submit"  android:textColor="@android:color/black"  android:textSize="20sp"  app:backgroundTint="@color/purple\_200"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/pass\_input" />   <TextView  android:id="@+id/message"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginStart="79dp"  android:layout\_marginTop="45dp"  android:layout\_marginEnd="79dp"  android:textSize="20sp"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toEndOf="@+id/button"  app:layout\_constraintTop\_toBottomOf="@+id/pass\_input" />  </androidx.constraintlayout.widget.ConstraintLayout> |

**4.1.5 App main page**

|  |
| --- |
| package com.example.medicare\_application;  import androidx.appcompat.app.AppCompatActivity;  import android.content.ActivityNotFoundException; import android.content.Intent; import android.net.Uri; import android.os.Bundle; import android.view.MotionEvent; import android.view.View; import android.widget.ArrayAdapter; import android.widget.Button; import android.widget.ImageButton; import android.widget.Spinner; import android.widget.Toast;  import com.google.android.material.floatingactionbutton.FloatingActionButton;  public class MainPage extends AppCompatActivity {  private Button bButton, hButton, mButton;    @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_main\_page*);  bButton = (Button) findViewById(R.id.*bookButton*);  hButton = (Button) findViewById(R.id.*hospital*);  mButton = (Button) findViewById(R.id.*medicineBtn*);  FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.*fab1*);  fab.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Intent intent = new Intent(MainPage.this, Prescription.class);  startActivity(intent);  }  });    bButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  bookingPage();   }  });   hButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  hospitalSearch();  }  });   mButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  medicinePage();  }  });  }   private void bookingPage(){  Intent intent = new Intent(this, Booking\_Page.class);  startActivity(intent);  }   private void medicinePage(){  Intent intent = new Intent(this, Medicines.class);  startActivity(intent);  }   private void hospitalSearch(){  Intent i = new Intent(Intent.*ACTION\_VIEW*, Uri.*parse*("geo:0,0?q=\" + \"12.929951,77.609036"));  startActivity(i);   } } |

**4.1.6 AppMainPage.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"  android:background="@drawable/logo\_background"  tools:context=".MainPage">   <Button  android:id="@+id/medicineBtn"  android:layout\_width="341dp"  android:layout\_height="66dp"  android:layout\_marginStart="40dp"  android:layout\_marginTop="26dp"  android:layout\_marginEnd="30dp"  android:text="Purchase Medicines"  android:textSize="20sp"  android:textStyle="bold"  app:backgroundTint="#972F2F"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="1.0"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/bookButton" />   <Button  android:id="@+id/hospital"  android:layout\_width="341dp"  android:layout\_height="66dp"  android:layout\_marginStart="40dp"  android:layout\_marginTop="42dp"  android:layout\_marginEnd="30dp"  android:text="Locate Hospitals"  android:textSize="20sp"  android:textStyle="bold"  app:backgroundTint="#972F2F"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="1.0"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/medicineBtn" />   <ImageView  android:id="@+id/imageView2"  android:layout\_width="191dp"  android:layout\_height="136dp"  android:layout\_marginStart="1dp"  android:layout\_marginTop="4dp"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent"  app:srcCompat="@drawable/images" />   <TextView  android:id="@+id/textView4"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginStart="21dp"  android:layout\_marginTop="56dp"  android:layout\_marginEnd="28dp"  android:text="Medicare App"  android:textColor="@color/design\_default\_color\_background"  android:textSize="28sp"  android:textStyle="bold|italic"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="0.0"  app:layout\_constraintStart\_toEndOf="@+id/imageView2"  app:layout\_constraintTop\_toTopOf="parent" />   <com.google.android.material.floatingactionbutton.FloatingActionButton  android:id="@+id/fab1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginStart="327dp"  android:layout\_marginTop="540dp"  android:layout\_marginEnd="28dp"  android:clickable="true"  app:backgroundTint="#DAAD86"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="0.0"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent"  app:srcCompat="@android:drawable/btn\_star\_big\_on" />   <Button  android:id="@+id/bookButton"  android:layout\_width="341dp"  android:layout\_height="66dp"  android:layout\_marginStart="40dp"  android:layout\_marginTop="72dp"  android:layout\_marginEnd="30dp"  android:text="Make Booking"  android:textSize="20sp"  android:textStyle="bold"  app:backgroundTint="#972F2F"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/imageView2" /> </androidx.constraintlayout.widget.ConstraintLayout> |

**4.1.7 Booking Page**

|  |
| --- |
| package com.example.medicare\_application;  import androidx.appcompat.app.AppCompatActivity;  import android.content.Intent; import android.os.Bundle; import android.view.View; import android.widget.AdapterView; import android.widget.ArrayAdapter; import android.widget.Button; import android.widget.EditText; import android.widget.RadioButton; import android.widget.RadioGroup; import android.widget.Spinner; import android.widget.SpinnerAdapter; import android.widget.Toast;  import kotlin.collections.IndexedValue;  public class Booking\_Page extends AppCompatActivity {  private Button bookButton, checkButton;  private DBHelper dbHelper;  private EditText pName, pContact, pAge;  private RadioButton rb1, rb2, rb3;  RadioGroup rg;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_booking\_\_page*);  Spinner dropdown = (Spinner) findViewById(R.id.*spinner*);  String[] items = new String[]{"Choose Hospital", "Hospital 1", "Hospital 2", "Hospital 3"};  ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.*simple\_spinner\_dropdown\_item*, items);  dropdown.setAdapter(adapter);   pName = (EditText) findViewById(R.id.*b\_name*);  pContact = (EditText) findViewById(R.id.*b\_contact*);  pAge = (EditText) findViewById(R.id.*b\_age*);  bookButton = (Button) findViewById(R.id.*bookBtn*);  checkButton = (Button) findViewById(R.id.*loadBtn*);  rg =(RadioGroup) findViewById(R.id.*radioGroup*);    dbHelper = new DBHelper(Booking\_Page.this);   dropdown.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {  @Override  public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {  String hospitalName = (String) dropdown.getItemAtPosition(dropdown.getSelectedItemPosition());  *//Toast.makeText(Booking\_Page.this, "Selected Item "+hospitalName, Toast.LENGTH\_SHORT).show();* }   @Override  public void onNothingSelected(AdapterView<?> parent) {   }  });   bookButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  *// below line is to get data from all edit text fields.* int sel = rg.getCheckedRadioButtonId();  rb1 = (RadioButton)findViewById(sel);  *//Toast.makeText(Booking\_Page.this,rb1.getText().toString(),Toast.LENGTH\_SHORT).show();* String hospitalName = dropdown.getSelectedItem().toString();  String patientName = pName.getText().toString();  String patientContact = pContact.getText().toString();  String patientAge = pAge.getText().toString();  String patientGender = rb1.getText().toString();  *//RadioGroup patientGender = option.getCheckedRadioButtonId()   // validating if the text fields are empty or not.* if (patientName.isEmpty() && patientContact.isEmpty() && patientAge.isEmpty()) {  Toast.*makeText*(Booking\_Page.this, "Please enter all the data..", Toast.*LENGTH\_SHORT*).show();  return;  }  else if(patientContact.length() != 10) {  Toast.*makeText*(getApplicationContext(), "Enter a valid phone number", Toast.*LENGTH\_SHORT*).show();  }  else if(patientAge.equals(0) | patientAge.equals(102)){  Toast.*makeText*(getApplicationContext(), "Enter a valid age", Toast.*LENGTH\_SHORT*).show();  }   else {  *// on below line we are calling a method to add new  // course to sqlite data and pass all our values to it.* dbHelper.addNewBooking(hospitalName, patientName, patientContact, patientAge, patientGender);   *// after adding the data we are displaying a toast message.* Toast.*makeText*(Booking\_Page.this, "Booking Suucessful Done.", Toast.*LENGTH\_SHORT*).show();  pName.setText("");  pContact.setText("");  pAge.setText("");  }  }  });   checkButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Intent i = new Intent(Booking\_Page.this, ViewBookingData.class);  startActivity(i);  }  });  } } |

**4.1.8 Booking.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"  android:background="#E7717D"  tools:context=".Booking\_Page">   <EditText  android:id="@+id/b\_age"  android:layout\_width="350dp"  android:layout\_height="43dp"  android:layout\_marginStart="20dp"  android:layout\_marginTop="20dp"  android:layout\_marginEnd="41dp"  android:ems="10"  android:hint="Age of Patient"  android:inputType="number"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/b\_contact" />   <EditText  android:id="@+id/b\_contact"  android:layout\_width="350dp"  android:layout\_height="43dp"  android:layout\_marginStart="20dp"  android:layout\_marginTop="16dp"  android:layout\_marginEnd="41dp"  android:ems="10"  android:hint="Contact Number"  android:inputType="number"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/b\_name" />   <ImageView  android:id="@+id/imageView3"  android:layout\_width="0dp"  android:layout\_height="118dp"  android:layout\_marginStart="54dp"  android:layout\_marginTop="16dp"  android:layout\_marginEnd="66dp"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent"  app:srcCompat="@drawable/booking\_page" />   <Spinner  android:id="@+id/spinner"  android:layout\_width="234dp"  android:layout\_height="39dp"  android:layout\_marginStart="16dp"  android:layout\_marginTop="30dp"  android:spinnerMode="dropdown"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/imageView3" />   <EditText  android:id="@+id/b\_name"  android:layout\_width="350dp"  android:layout\_height="43dp"  android:layout\_marginStart="16dp"  android:layout\_marginTop="23dp"  android:layout\_marginEnd="45dp"  android:ems="10"  android:hint="Patient's Name"  android:inputType="textPersonName"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/spinner" />   <RadioGroup  android:id="@+id/radioGroup"  android:layout\_width="258dp"  android:layout\_height="62dp"  android:layout\_marginStart="20dp"  android:layout\_marginTop="15dp"  android:orientation="horizontal"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/textView5">   <RadioButton  android:id="@+id/radioButton"  android:layout\_width="82dp"  android:layout\_height="wrap\_content"  android:text="Male" />   <RadioButton  android:id="@+id/radioButton2"  android:layout\_width="82dp"  android:layout\_height="wrap\_content"  android:text="Female" />   <RadioButton  android:id="@+id/radioButton3"  android:layout\_width="82dp"  android:layout\_height="wrap\_content"  android:text="Other" />  </RadioGroup>   <TextView  android:id="@+id/textView5"  android:layout\_width="350dp"  android:layout\_height="43dp"  android:layout\_marginStart="20dp"  android:layout\_marginTop="22dp"  android:text="Gender"  android:textColor="@color/white"  android:textSize="20sp"  android:textStyle="bold"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/b\_age" />   <Button  android:id="@+id/bookBtn"  android:layout\_width="164dp"  android:layout\_height="39dp"  android:layout\_marginStart="110dp"  android:layout\_marginTop="24dp"  android:layout\_marginEnd="110dp"  android:text="Book"  android:textSize="18sp"  android:textStyle="bold"  app:backgroundTint="#BC986A"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="0.233"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/radioGroup" />   <Button  android:id="@+id/loadBtn"  android:layout\_width="153dp"  android:layout\_height="62dp"  android:layout\_marginStart="116dp"  android:layout\_marginTop="16dp"  android:layout\_marginEnd="106dp"  android:text="Check bookings"  android:textSize="16sp"  android:textStyle="bold"  app:backgroundTint="#BC986A"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="0.0"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/bookBtn" />   </androidx.constraintlayout.widget.ConstraintLayout> |

**4.1.9 Medicines Page**

|  |
| --- |
| package com.example.medicare\_application;  import androidx.appcompat.app.AppCompatActivity;  import android.content.Intent; import android.os.Bundle; import android.view.View; import android.widget.Button; import android.widget.EditText; import android.widget.ImageButton; import android.widget.TextView; import android.widget.Toast;  public class Medicines extends AppCompatActivity {  Button btn1,btn2;  ImageButton img;  EditText text;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_medicines*);  btn1 = (Button) findViewById(R.id.*migrane*);  btn2 = (Button) findViewById(R.id.*cold*);  img = (ImageButton) findViewById(R.id.*imgButton*);  text = (EditText) findViewById(R.id.*textV*);   btn1.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  loadOne();  }  });   btn2.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  loadTwo();  }  });   img.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  if(text.getText().toString() == "headaches" || text.getText().toString() == "fever"){  loadOne();  }  else if(text.getText().toString() == "Cold syrup" || text.getText().toString() == "cough"){  loadTwo();  }  else{  Toast.*makeText*(getApplicationContext(),"Search not availale", Toast.*LENGTH\_SHORT*).show();  }  }  });  }  private void loadOne(){  Intent intent = new Intent(this,Migranes.class);  startActivity(intent);  }   private void loadTwo(){  Intent in = new Intent(this,Cold.class);  startActivity(in);  } } |

**4.2.1 MedicinesPage.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"  android:background="@drawable/med"  tools:context=".Medicines">   <Button  android:id="@+id/cold"  android:layout\_width="311dp"  android:layout\_height="46dp"  android:layout\_marginStart="51dp"  android:layout\_marginTop="42dp"  android:layout\_marginEnd="49dp"  android:text="Cold/Cough"  android:textSize="16sp"  android:textStyle="bold"  app:backgroundTint="#4AB669"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="0.0"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/migrane" />   <Button  android:id="@+id/calc"  android:layout\_width="311dp"  android:layout\_height="46dp"  android:layout\_marginStart="51dp"  android:layout\_marginTop="42dp"  android:layout\_marginEnd="49dp"  android:text="Calcium/Zinc Tabs"  android:textSize="16sp"  android:textStyle="bold"  app:backgroundTint="#4AB669"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="0.0"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/cold" />   <Button  android:id="@+id/pain"  android:layout\_width="311dp"  android:layout\_height="46dp"  android:layout\_marginStart="51dp"  android:layout\_marginTop="44dp"  android:layout\_marginEnd="49dp"  android:text="Pain Killers"  android:textSize="16sp"  android:textStyle="bold"  app:backgroundTint="#4AB669"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="1.0"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/calc" />   <TextView  android:id="@+id/textView2"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginStart="127dp"  android:layout\_marginTop="43dp"  android:layout\_marginEnd="124dp"  android:text="Select a Category"  android:textAlignment="center"  android:textColor="#F6F4F4"  android:textSize="25sp"  android:textStyle="bold"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent" />   <Button  android:id="@+id/migrane"  android:layout\_width="311dp"  android:layout\_height="46dp"  android:layout\_marginStart="51dp"  android:layout\_marginTop="152dp"  android:layout\_marginEnd="49dp"  android:text="Migranes"  android:textSize="16sp"  android:textStyle="bold"  app:backgroundTint="#4AB669"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="0.0"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/textView2" />   <ImageView  android:id="@+id/imageView9"  android:layout\_width="311dp"  android:layout\_height="55dp"  android:layout\_marginStart="51dp"  android:layout\_marginTop="19dp"  android:layout\_marginEnd="49dp"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/textView2"  app:srcCompat="@drawable/searchbar" />   <EditText  android:id="@+id/textV"  android:layout\_width="235dp"  android:layout\_height="45dp"  android:layout\_marginTop="35dp"  android:textSize="16sp"  android:textStyle="bold"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="0.4"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/textView2" />   <ImageButton  android:id="@+id/imgButton"  android:layout\_width="46dp"  android:layout\_height="50dp"  android:layout\_marginStart="11dp"  android:layout\_marginTop="24dp"  android:layout\_marginEnd="48dp"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toEndOf="@+id/textV"  app:layout\_constraintTop\_toBottomOf="@+id/textView2"  app:srcCompat="@android:drawable/ic\_search\_category\_default" />  </androidx.constraintlayout.widget.ConstraintLayout> |

**4.2.2 Prescription Page**

|  |
| --- |
| package com.example.medicare\_application;  import androidx.appcompat.app.AppCompatActivity;  import android.app.Activity; import android.content.ContentValues; import android.content.Intent; import android.database.Cursor; import android.database.sqlite.SQLiteDatabase; import android.graphics.Bitmap; import android.graphics.BitmapFactory; import android.net.Uri; import android.os.Bundle; import android.provider.MediaStore; import android.view.View; import android.widget.ImageView; import android.widget.Toast;  import java.io.ByteArrayOutputStream; import java.io.FileNotFoundException; import java.io.IOException;  public class Prescription extends AppCompatActivity {  ImageView imageView,imageview2;   Bitmap bitmap = null;   byte img[];   private MyDataBase mdb=null;   private SQLiteDatabase db=null;   private Cursor c=null;   private static final String *DATABASE\_NAME* = "ImageDb.db";   public static final int *DATABASE\_VERSION* = 1;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_prescription*);  imageView = (ImageView) findViewById(R.id.*imageView4*);  imageview2 = (ImageView) findViewById(R.id.*imageView5*);   mdb=new MyDataBase(getApplicationContext(), *DATABASE\_NAME*,null,     *DATABASE\_VERSION*);     }   public void selectImage(View view)   {   Intent intent = new Intent(Intent.*ACTION\_PICK*,     MediaStore.Images.Media.*EXTERNAL\_CONTENT\_URI*);     startActivityForResult(intent,0);   }     @Override     protected void onActivityResult(int requestCode, int resultCode, Intent data) {   if (resultCode == Activity.*RESULT\_OK* && data != null) {   Uri selectedImage = data.getData();   try {   bitmap = MediaStore.Images.Media.*getBitmap*(this.getContentResolver(),    selectedImage);   ByteArrayOutputStream bos = new ByteArrayOutputStream();   bitmap.compress(Bitmap.CompressFormat.*PNG*, 100, bos);   img = bos.toByteArray();   imageView.setImageBitmap(bitmap);   } catch (FileNotFoundException e) {   e.printStackTrace();   } catch (IOException e) {   e.printStackTrace();   }   }    }     @SuppressWarnings("deprecation")   public void uploadImage(View view)   {   db=mdb.getWritableDatabase();   ContentValues cv = new ContentValues();   cv.put("image", img);   db.insert("tableimage", null, cv);   imageView.setAlpha(0);   Toast.*makeText*(this, "inserted successfully", Toast.*LENGTH\_SHORT*).show();     }   public void sendImage(View view)   {   String[] col={"image"};   db=mdb.getReadableDatabase();   c=db.query("tableimage", col, null, null, null, null, null);   if(c!=null){   c.moveToFirst();   do{   img=c.getBlob(c.getColumnIndex("image"));   }while(c.moveToNext());   }   Bitmap b1= BitmapFactory.*decodeByteArray*(img, 0, img.length);     imageview2.setImageBitmap(b1);   Toast.*makeText*(this, "Retrive successfully", Toast.*LENGTH\_SHORT*).show();   }  } |

**4.2.3 Prescription.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"  android:background="#EDC7B7"  tools:context=".Prescription">   <Button  android:id="@+id/imgGet"  android:layout\_width="0dp"  android:layout\_height="65dp"  android:layout\_marginStart="124dp"  android:layout\_marginTop="10dp"  android:layout\_marginEnd="116dp"  android:text="Load Precription"  android:textSize="16sp"  android:textStyle="bold"  app:backgroundTint="#116466"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/imgUpload" />   <Button  android:id="@+id/imgUpload"  android:layout\_width="0dp"  android:layout\_height="47dp"  android:layout\_marginTop="35dp"  android:layout\_marginEnd="16dp"  android:onClick="uploadImage"  android:text="Upload Image"  android:textSize="16sp"  android:textStyle="bold"  app:backgroundTint="#116466"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/imageView4" />   <TextView  android:id="@+id/textView6"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginStart="145dp"  android:layout\_marginTop="49dp"  android:layout\_marginEnd="146dp"  android:onClick="sendImage"  android:text="Upload Precription "  android:textColor="#AC3B61"  android:textSize="25sp"  android:textStyle="bold|italic"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent" />   <ImageView  android:id="@+id/imageView4"  android:layout\_width="158dp"  android:layout\_height="140dp"  android:layout\_marginStart="126dp"  android:layout\_marginTop="24dp"  android:layout\_marginEnd="127dp"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/textView6"  app:srcCompat="@android:drawable/ic\_input\_add" />   <Button  android:id="@+id/imgSelect"  android:layout\_width="0dp"  android:layout\_height="47dp"  android:layout\_marginStart="34dp"  android:layout\_marginTop="35dp"  android:onClick="selectImage"  android:text="Choose Image"  android:textSize="16sp"  android:textStyle="bold"  app:backgroundTint="#116466"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/imageView4" />   <ImageView  android:id="@+id/imageView5"  android:layout\_width="309dp"  android:layout\_height="243dp"  android:layout\_marginStart="49dp"  android:layout\_marginTop="9dp"  android:layout\_marginEnd="49dp"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/imgGet" />  </androidx.constraintlayout.widget.ConstraintLayout> |

**4.2.4 DBHelper Database**

|  |
| --- |
| package com.example.medicare\_application;  import android.content.ContentValues; import android.content.Context; import android.database.Cursor; import android.database.sqlite.SQLiteDatabase; import android.database.sqlite.SQLiteOpenHelper;  import androidx.annotation.Nullable;  import java.util.ArrayList;  public class DBHelper extends SQLiteOpenHelper {  *// creating a constant variables for our database.  // below variable is for our database name.* private static final String *DB\_NAME* = "bookingdb";   *// below int is our database version* private static final int *DB\_VERSION* = 1;   *// below variable is for our table name.* private static final String *TABLE\_NAME* = "mybookings";   *// below variable is for our id column.* private static final String *ID\_COL* = "id";   *// below variable is for our course name column* private static final String *HOSPITAL\_NAME\_COL* = "hospital\_name";   *// below variable id for our course duration column.* private static final String *NAME\_COL* = "name";   *// below variable for our course description column.* private static final String *CONTACT\_COL* = "contact";   *// below variable is for our course tracks column.* private static final String *AGE\_COL* = "age";   *// below variable is for gender column.* private static final String *GENDER\_COL* = "gender";   *// creating a constructor for our database handler.* public DBHelper(Context context) {  super(context, *DB\_NAME*, null, *DB\_VERSION*);  }   @Override  public void onCreate(SQLiteDatabase db) {  *// on below line we are creating  // an sqlite query and we are  // setting our column names  // along with their data types.* String query = "CREATE TABLE " + *TABLE\_NAME* + " ("  + *ID\_COL* + " INTEGER PRIMARY KEY AUTOINCREMENT, "  + *HOSPITAL\_NAME\_COL* + " TEXT,"  + *NAME\_COL* + " TEXT,"  + *CONTACT\_COL* + " INTEGER,"  + *AGE\_COL* + " INTEGER,"  + *GENDER\_COL* + " TEXT)";   *// at last we are calling a exec sql  // method to execute above sql query* db.execSQL(query);  }   public void addNewBooking(String hospitalName, String patientName, String patientContact, String patientAge, String patientGender) {   *// on below line we are creating a variable for  // our sqlite database and calling writable method  // as we are writing data in our database.* SQLiteDatabase db = this.getWritableDatabase();   *// on below line we are creating a  // variable for content values.* ContentValues values = new ContentValues();   *// on below line we are passing all values  // along with its key and value pair.* values.put(*HOSPITAL\_NAME\_COL*, hospitalName);  values.put(*NAME\_COL*, patientName);  values.put(*CONTACT\_COL*, patientContact);  values.put(*AGE\_COL*, patientAge);  values.put(*GENDER\_COL*, patientGender);   *// after adding all values we are passing  // content values to our table.* db.insert(*TABLE\_NAME*, null, values);   *// at last we are closing our  // database after adding database.* db.close();  }   public ArrayList<BookingData> readCourses() {  *// on below line we are creating a  // database for reading our database.* SQLiteDatabase db = this.getReadableDatabase();   *// on below line we are creating a cursor with query to read data from database.* Cursor cursorCourses = db.rawQuery("SELECT \* FROM " + *TABLE\_NAME*, null);   *// on below line we are creating a new array list.* ArrayList<BookingData> courseModalArrayList = new ArrayList<>();   *// moving our cursor to first position.* if (cursorCourses.moveToFirst()) {  do {  *// on below line we are adding the data from cursor to our array list.* courseModalArrayList.add(new BookingData(cursorCourses.getString(1),  cursorCourses.getString(2),  cursorCourses.getString(3),  cursorCourses.getString(4),  cursorCourses.getString(5)));  } while (cursorCourses.moveToNext());  *// moving our cursor to next.* }  *// at last closing our cursor  // and returning our array list.* cursorCourses.close();  return courseModalArrayList;  }   @Override  public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  *// this method is called to check if the table exists already.* db.execSQL("DROP TABLE IF EXISTS " + *TABLE\_NAME*);  onCreate(db);  } } |

**4.2.5 ViewBookingData**

|  |
| --- |
| package com.example.medicare\_application;  import androidx.appcompat.app.AppCompatActivity; import androidx.recyclerview.widget.LinearLayoutManager; import androidx.recyclerview.widget.RecyclerView;  import android.os.Bundle;  import java.util.ArrayList;  public class ViewBookingData extends AppCompatActivity {   *// creating variables for our array list,  // dbhandler, adapter and recycler view.* private ArrayList<BookingData> courseModalArrayList;  private DBHelper dbHelper;  private BookingRVAdapter bookingRVAdapter;  private RecyclerView bookingRV;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_view\_booking\_data*);   *// initializing our all variables.* courseModalArrayList = new ArrayList<>();  dbHelper = new DBHelper(ViewBookingData.this);   *// getting our course array  // list from db handler class.* courseModalArrayList = dbHelper.readCourses();   *// on below line passing our array lost to our adapter class.* bookingRVAdapter = new BookingRVAdapter(courseModalArrayList, ViewBookingData.this);  bookingRV = findViewById(R.id.*idRVCourses*);   *// setting layout manager for our recycler view.* LinearLayoutManager linearLayoutManager = new LinearLayoutManager(ViewBookingData.this, RecyclerView.*VERTICAL*, false);  bookingRV.setLayoutManager(linearLayoutManager);   *// setting our adapter to recycler view.* bookingRV.setAdapter(bookingRVAdapter);  } } |

**4.2.6 ViewBookingData.xml**

|  |
| --- |
| *<?*xml version="1.0" encoding="utf-8"*?>* <RelativeLayout  xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".ViewBookingData">   *<!--recycler view for displaying our courses-->* <androidx.recyclerview.widget.RecyclerView  android:id="@+id/idRVCourses"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent" />  </RelativeLayout> |

**4.2.7 BookingRVAdapter**

|  |
| --- |
| package com.example.medicare\_application;  import android.content.Context; import android.view.LayoutInflater; import android.view.View; import android.view.ViewGroup; import android.widget.TextView;  import androidx.annotation.NonNull; import androidx.recyclerview.widget.RecyclerView;  import java.util.ArrayList;  public class BookingRVAdapter extends RecyclerView.Adapter<BookingRVAdapter.ViewHolder> {  *// variable for our array list and context* private ArrayList<BookingData> courseModalArrayList;  private Context context;   *// constructor* public BookingRVAdapter(ArrayList<BookingData> courseModalArrayList, Context context) {  this.courseModalArrayList = courseModalArrayList;  this.context = context;  }   @NonNull  @Override  public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  *// on below line we are inflating our layout  // file for our recycler view items.* View view = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*booking\_rv\_item*, parent, false);  return new ViewHolder(view);  }   @Override  public void onBindViewHolder(@NonNull ViewHolder holder, int position) {  *// on below line we are setting data  // to our views of recycler view item.* BookingData modal = courseModalArrayList.get(position);  holder.hospitalName.setText(modal.getHospitalName());  holder.patientName.setText(modal.getPatientName());  holder.patientContact.setText(modal.getPatientContact());  holder.patientAge.setText(modal.getPatientAge());  holder.patientGender.setText(modal.getPatientGender());  }   @Override  public int getItemCount() {  *// returning the size of our array list* return courseModalArrayList.size();  }   public class ViewHolder extends RecyclerView.ViewHolder {   *// creating variables for our text views.* private TextView hospitalName, patientName, patientContact, patientAge, patientGender;   public ViewHolder(@NonNull View itemView) {  super(itemView);  *// initializing our text views* hospitalName = itemView.findViewById(R.id.*HospitalName*);  patientName = itemView.findViewById(R.id.*PatientName*);  patientContact = itemView.findViewById(R.id.*PatientContact*);  patientAge = itemView.findViewById(R.id.*PatientAge*);  patientGender = itemView.findViewById(R.id.*PatientGender*);  }  } } |

**5.0 Result and Discussion**

The Application works as a service that is made up of various attribute making sure it is efficient and smooth functioning at the same time. It has a lot of other features too. The app is mainly divided into different modules which are shown below.

**Home Page**

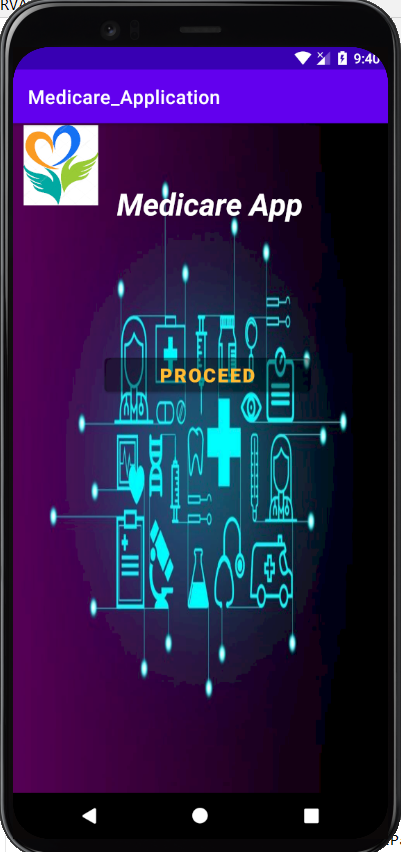
****

Fig-5.1 Home Page

This is the main page or the home page of the application. The user is taken to the home page first when they open the app and then from there they can navigate to other pages like login or register. The main page gives a general look of the app. This page works as a bridge between the user and the resources as well as provides functionality which the user can use. This page is accessible to anyone who accesses the application through a device.

**Login Page**

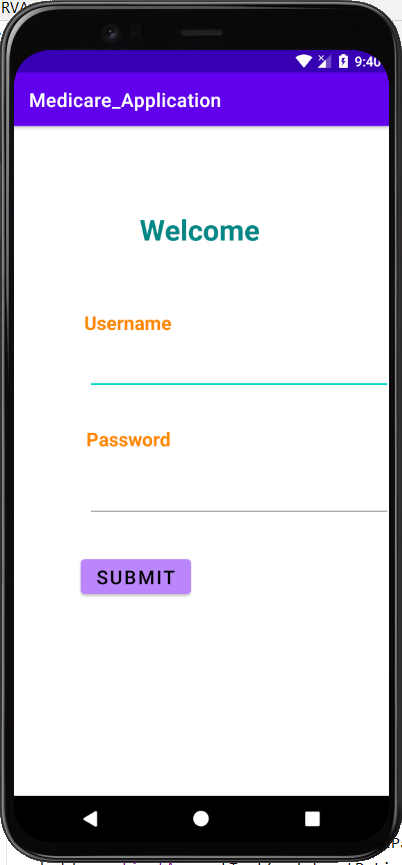
****

Fig-5.2 Login Page

The site uses Google firebase for user Authentication. Because of that the user can login and use the resources with a simple click. Google firebase provides a layer of security along with fast login service.

**Choose from the options available**

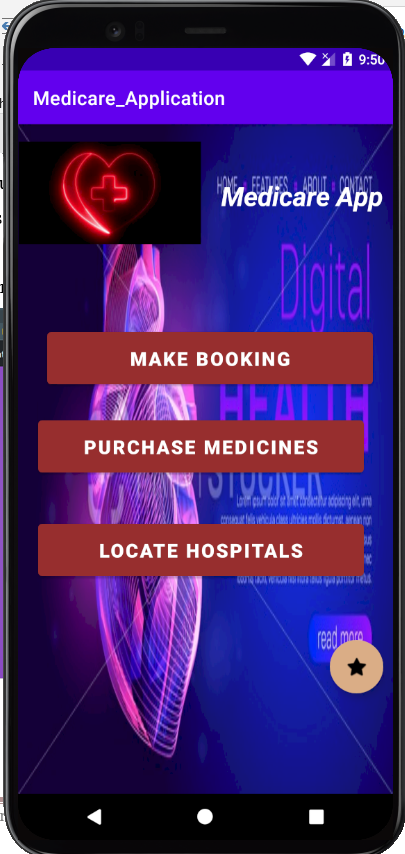
****

Fig-5.3 Choose from the options available

The app at this stage allows the user to book an appointment, purchase medicines from a pharmacy online, locate hospitals near by and upload prescriptions in case they want to purchase medicines which require doctors prescription in order to purchase them

**Booking**

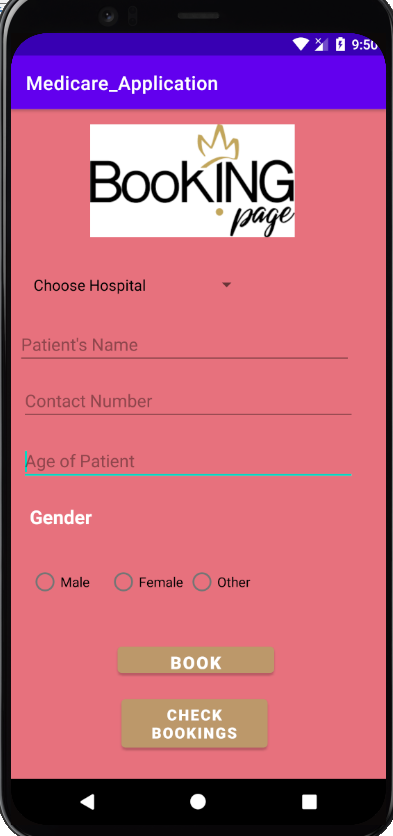
****

Fig-5.4 Booking

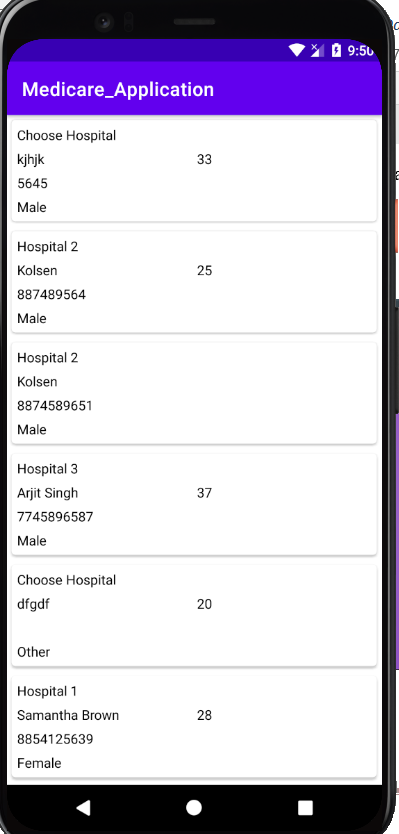
****

Fig-5.5 Booking Data Check Page

When the user selects the booking button, they will be redirected to the booking page where they can book an appointment with the doctor or hospital. They will have to fill the form and enter valid details as the form has validation checks in it. They can then check the booking which has been done by them.

**Pharmacy Page**

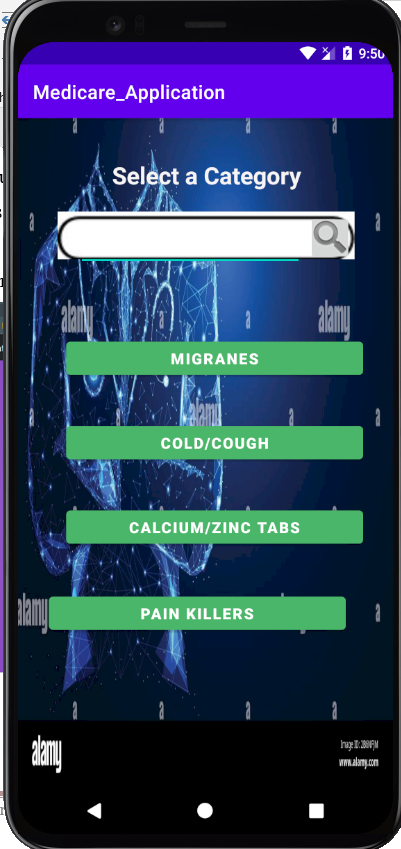
****

Fig-5.6 Pharmacy Page

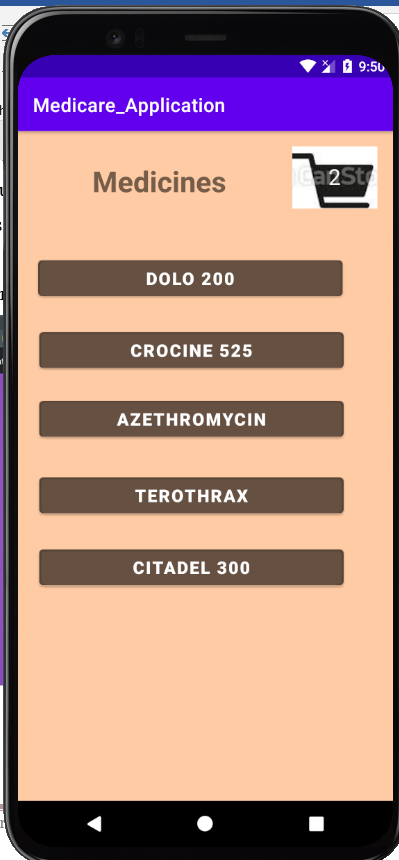
****

Fig-5.7 Pharmacy Page

**Maps Page**

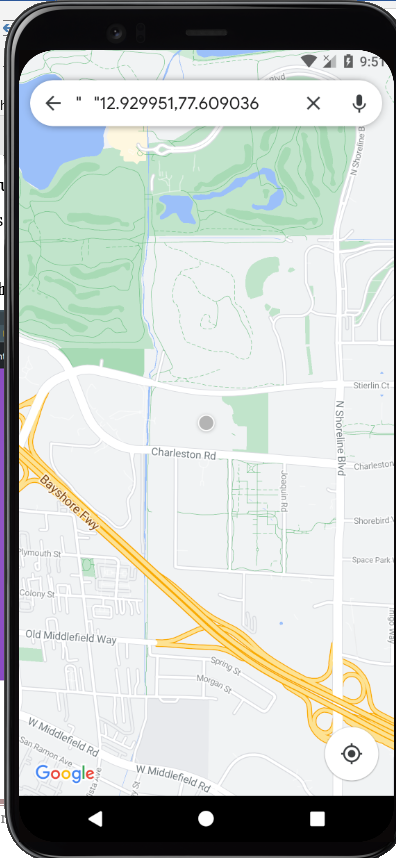
****

Fig-5.8 Maps Page

**Prescription Page**

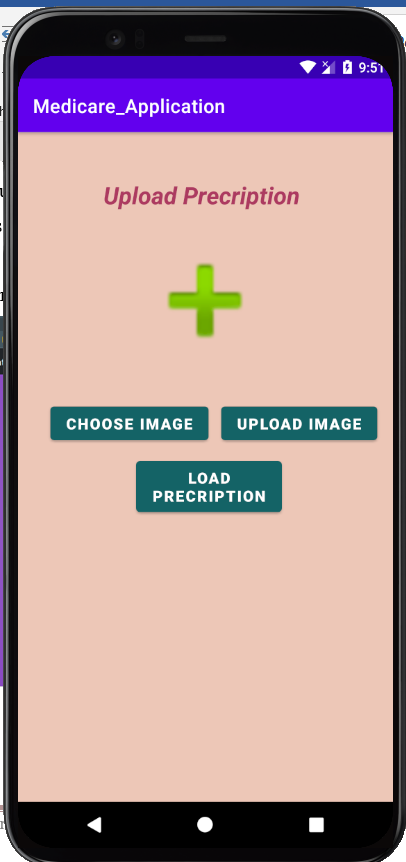
****

Fig-5.9 Prescription Page

**6.0 Conclusion**

**6.1 ADVANTAGES**

The application has a lot of advantages as mentioned below

* Instant location information of nearby hospitals is provided
* Booking can be done for more than one individual
* Medicines can be purchases online without the need for personally going to the store and picking it up
* Unlike other booking apps, various options are available here such as booking appointments with hospitals and doctors, purchasing medicines online and locating hospitals in case of emergency
* The user is allowed to upload prescriptions in order to purchase medicines only which are prescribed by the doctor
* User data is safely stored on firebase with the appropriate security measures in place to protect the data.

**6.2 Limitations**

One of the few limitations of the app is the efficiency and its functioning, as the app has limited number of modules and functionality. The app depends on the internet in order to function, so if the user’s network isn’t good enough, it won’t work. The maps API can cause a few problems in places where google maps doesn’t work properly, in the sense that the geographic location won’t be exact which might affect the locator functioning of the app.

**6.3 FUTURE ENHANCEMENT**

* The app lacks in connecting doctors who are not part of a hospital, this makes it difficult for doctors to connect with patient accordingly. Unless they have contact with the hospital, connecting to them personally is not possible. To handle this, special bookings will have to be made and registration of such doctors is required.
* In the long run, to increase the apps functionality, more modules and operability will be added, ease of access and smooth functionality will be looked into. Uniqueness of the app will also be dealt with, i.e. making it different from other applications that are already available in the market.
* The apps UI can be enhanced to look more appealing and pleasing to the eyes, as well as working in a smooth and efficient way.
  1. **REFERENCES**

1. <https://firebase.google.com/docs/android/setup#:~:text=Open%20the%20Firebase%20Assistant%3A%20Tools,your%20Android%20project%20with%20Firebase>.
2. <https://www.geeksforgeeks.org/how-to-read-data-from-sqlite-database-in-android/>
3. <https://stackoverflow.com/questions/7510219/deleting-row-in-sqlite-in-android>
4. <https://www.studentstutorial.com/android/sql_lite_example.php>