Digitalization And Integration of Healthcare Systems in India

Nafees

Dedicated to my parents, family and Friends

Table of Content

List of Figures

List of Listings

List of Abbreviations

NHS	National Health Service
UK	United Kingdom
App	Application
IDE	Integrated Development Environment
API	Application Programming Interface
IT	Information Technology
AI	Artificial Intelligence
HIPAA	Health Insurance Portability and Accountability Act Rules
FDA	Food and Drug Administration
APK	Android Package Kit
UI	User Interface
XML	Extensible Markup Language
OS	Operating System

Abstract

Most of the hospitals in India are not integrated or not accountable for their operations or serviced due to massive expansion of people across the country. sometimes the poor does not get the pure medical services by the health systems in India due to discriminated nature of their services and operations in terms of public and private spaces. Therefore, this work aims to design a mobile application that integrates the overall hospitals at one place and allows to create an interaction platform from which the patients and doctors can interact easily with the help of this application with minimum efforts. This application is built on Android Studio with Java and used the external APIs (application programming interfaces) for connecting various buttons and retrieve data from the databases and display the same on the mobile screen. In this process, the SQL database has been implemented to store and retrieve data from the database. The overall performance of the application seem to deliver the objectives of listing out the hospitals, doctors, and services provided by each hospital with great accuracy.

1 Introduction

1.1 Overview

India's healthcare system has been facing a crisis for a long time with the significant impact felt by its poor citizens (Gauttam et al., 2021). There is a serious challenge which is the lack of proper equipment, adequate infrastructure, and the number of medical staff. The government hospitals which are supposed to provide the best healthcare services to the poor are affected by many issues which leave many patients missing the proper medical attention. Another big problem every government hospital face is the long waiting hours for consultation and patients left unattended for extended periods (Kadravello et al., 2021). These hospitals are also not following hygiene and safety standards. Often the maintenance of these hospitals is substandard. A maximum of the hospitals in India are also not adapted to use the technology and doctors are not updated with the latest medical procedures adequately. The risk of infections within the hospitals is also questionable due to the lack of hygiene. The lack of proper research and development is another major concern to count on for the poor medical system in India. Patients will do good only when there is a proper monitoring system after their discharge and the absence of this is leading to an increase in the rate of readmission.

These problems need to be addressed using a comprehensive solution. Advanced and updated technologies can be implemented within healthcare to improve the overall systems. The introduction of an electronic health record system will make it easy to monitor the progress of the patients. It could have been highly helpful, especially in situations like COVID where millions of people are hospitalized. The development of a mobile application could be a better solution where patients can make appointments, check the availability of hospitals, minimizing of waiting time and improve consultations.

1.2 Purpose of Study

The purpose of this study is to discover the potential advantages of implementing an integrated central system to administer healthcare in hospitals in India funded by the government. The present system of healthcare in India is in many ways and mainly in terms of technology. There is an immediate necessity of developing a centralized system which could be possible using an app. The author here proposes developing an app that could help the patients in selecting the best hospital based on their illness and requirement. The proposed plan would involve state governments working in collaboration with each district to form an integrated care system. This system would take collective responsibility for managing resources, delivering standards, and

improving the health of the population they serve. Through this association, local services can provide better care and treatment. It is a practice in India that people tend to visit hospitals for every other small issue due to a lack of knowledge. The app will also help to educate them about regular illnesses and make them understand about the dealing with that situations.

1.3 Aim and Objectives

Aim: The study aims to highlight the importance and advantages of having a centralized healthcare system in India which is possible via the development of an app.

Objectives:

The objective of this study includes:

- To understand the current healthcare system in India and the challenges faced by the hospitals and especially the government funded.
- The benefits and impact on the healthcare system after implementing the centralized governing system.
- Using digital transformation, it is possible to implement a unified system for healthcare.
- To assess the feasibility of developing an application that can integrate with existing healthcare systems and infrastructure in India.

1.4 Technologies / Software Used

There will be a requirement for various technologies to be involved in developing a healthcare app. This app was developed using Java, IDEs like Android Studio and APIs to access and integrate with other healthcare systems. Java is an object-oriented programming language widely used for developing mobile apps. Android Studio is the preferred IDE for developing Android-based applications, which can be used to create high-quality user interfaces, test the app's functionality, and optimize the code.

1.5 Outline of the Thesis

The author discussed the overview of the study including the problem statement and proposed solution in Chapter 1. Chapter 2 provides a review of the literature related to the healthcare system in India. The research methodologies used in the study are explained in Chapter 3. Implementation is discussed in Chapter 4 followed by analysis and findings where the research methods are studied. Discussions, conclusions, and Future Scope can be observed in Chapter 5 and Chapter 6 respectively.

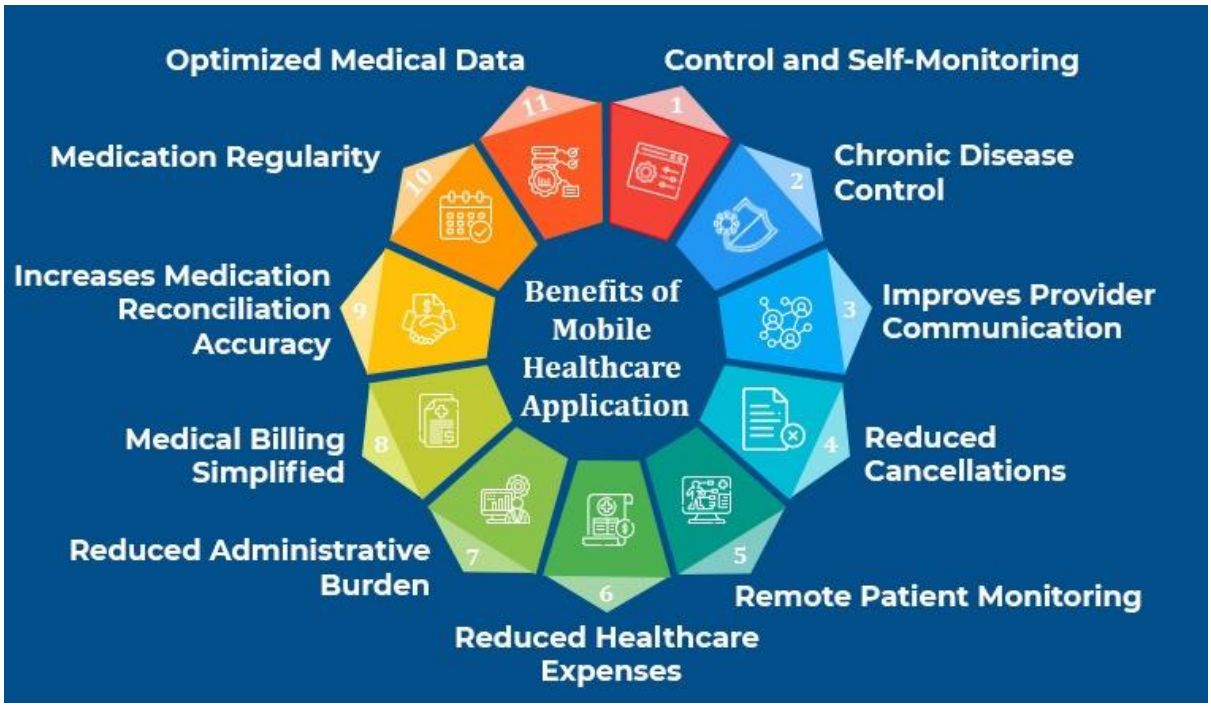
2 Literature Review

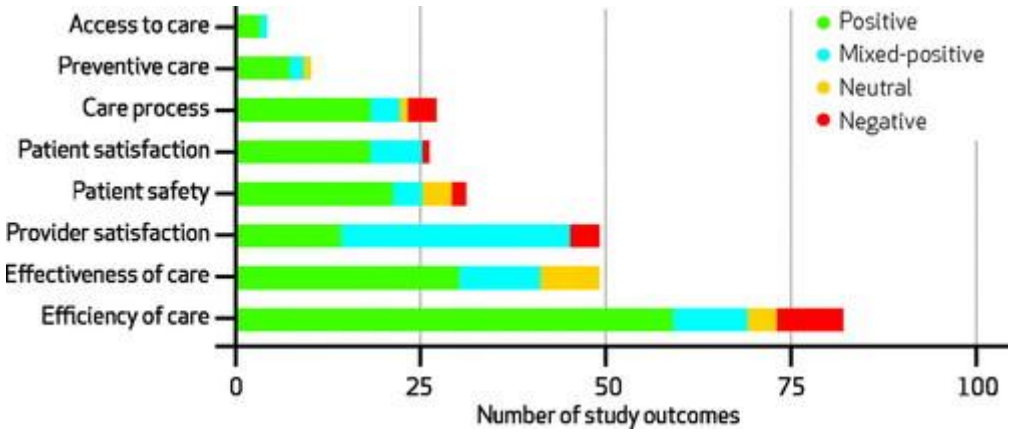
2.1 Overview

Every day, more and more medication is being delivered digitally. In particular, the prevalence of information technology (IT) has helped the development of multinational health systems. Over the past ten years, medical expenditure has grown globally and has become a significant portion of the ordinary man's salary. However, the overall healthcare spending per individual in India increased from $ 38 in 2009 to $ 64 in 2019. The method of documenting consumer permission, which would be increasingly digitalized by IT companies in the healthcare industry, serves as one of the most crucial relationships involving medical systems and digitalization. Dwivedi et al. (2020) explained that using tools to promote consciousness gives the individual relevant information to make educated decisions. Moreover, the lack of smooth contact between medical staff and patients is also filled through this digitized authorization from hospitals. Unless it's integral to its success, cryptocurrency digital records, or Automation diagnostic implants, there are many unmistakable signs that virtual artificial intelligence is rapidly permeating healthcare systems. This modernization will affect judgment calls, potential treatments, objectives, and the way medical data is being stored and distributed, how service users connect with their medical providers, and then how service users engage in conversation with one another. Blockchains, for example, could provide improved electronic health records much like big data, smart wearable technologies, fully immersive, and treatment would help alter India's medical system (Haleem *et al*. 2022).

2.2 Impact of digital technology on Healthcare in India

The environment of something like the Indian health sector is evolving in response to the use of contemporary patient portals including smart wearables, telecare, genotyping, augmented worlds, bots, and artificial intelligence (AI). According to Dang et al. (2020), India is getting


Benefits of Mobile Healthcare Application
1
Control and Self-Monitoring
2
Chronic Disease Control
3
Improves Provider Communication
4
Reduced Cancellations
5
Remote Patient Monitoring
6
Reduced Healthcare Expenses
7
Reduced Administrative Burden
8
Medical Billing Simplified
9
Increases Medication Reconciliation Accuracy
10
Medication Regularity
11
Optimized Medical Data


Access to care
Preventive care
Care process
Patient satisfaction
Patient safety
Provider satisfaction
Effectiveness of care
Efficiency of care
Positive
Mixed-positive
Neutral
Negative
0
25
50
75
100
Number of study outcomes

2. of the OS and varying screen sizes and resolutions. This can make it challenging to develop apps that work seamlessly across all devices.
3. Ensuring that an app works across different devices can be challenging, as different devices have different hardware capabilities and configurations (Tóth and Lovászová, 2021).
4. Mobile apps are vulnerable to security threats such as hacking, data breaches and malware. Developers need to take appropriate measures to ensure that their apps are secure and protect user data.
5. App performance is critical to user satisfaction. Developing apps that are responsive, fast and reliable can be challenging, especially when dealing with intensive features such as graphics and multimedia (Hort *et al*., 2021).
6. Developing a user friendly and visually appealing user interface (UI) is critical to the success of many mobile apps. However, designing a UI that works across different devices and screen sizes can be challenging.
7. Thorough testing is essential to ensure that an app works as intended and is free from bugs and errors. However, testing can be time consuming and intensive and it may be challenging to test an app across all the devices and OS versions as well. (Alshayban *et al*., 2020).

2.7 Summary

In summary, the impact of digital technology on healthcare in India has been significant with the use of contemporary patient portals, smart wearables, telecare, genotyping, augmented

worlds, bots, and artificial intelligence (AI) improving the accessibility and quality of care. Information technology has also provided benefits such as pharmacological warnings, professional alerts, increased documentation and reporting of appointments and medical tests, clinical assistance, and accessibility of entire health information, all improving patient safety. However, there are also challenges such as the need for integrating healthcare institutions, ensuring privacy and security of patient data, and addressing the potential flaws and inaccuracies of connected health tools. The implementation of mobile apps using Java and Android Studio has several advantages, including cost effectiveness, cross platform compatibility and ease of development. However, challenges such as hardware fragmentation, security concerns and market saturation need to be addressed. Overall, the technology is transforming the healthcare industry and providing opportunities for better patient outcomes and more efficient care delivery.

3 Methodology

3.1 Overview

This chapter explains the research method selected for this project work and the flow of work process to achieve the overall objectives. In the last Chapter 2 of Literature Review the author considered a critical review of the existing research articles and found some of the gaps from the existing research. After that an appropriate method is need to the find the solutions for the defined problem statement in the research. Therefore, this chapter consists a basic overview of the research methods in Section 3.2 followed by the research design, its perspectives and creation of a mobile app procedures. In the Section 3.3, the author considered to declare the ethics and legal issues that generally considered while designing a project work in real-time environments and in academic studies. Finally, the chapter is summarized in Section 3.4 with overall highlights of the studies in this chapter to introduce the next chapter.

3.2 Research Methods

The research methods help to understand different strategies, processes, and techniques to be implemented in conducting research or creating a product (Mekonnen, 2020). There are different types of research methods to provide existing ideas and allows to uncover the information for creating a better approach for the topic understanding and creating the new approach in a scientific manner. These methods include interview, surveys, observations, focus groups, and field experiments based on the type of research type.

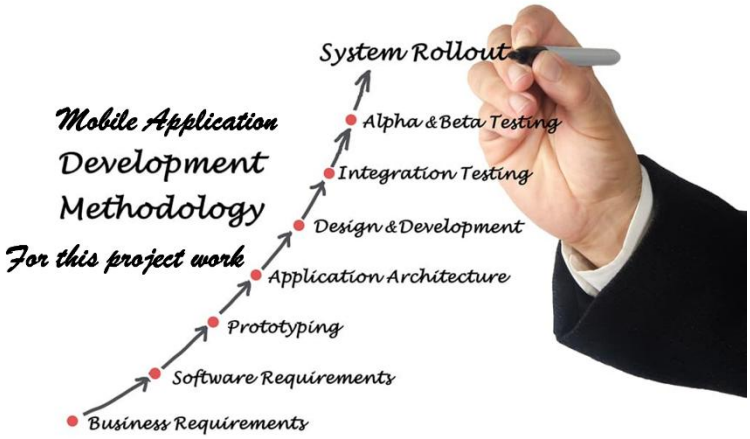
On the other hand, the research methods in general are also classified as qualitative, quantitative, mixed research and secondary data analysis (Taherdoost, 2022). These methods are extensively used in deriving certain parameters based on the existing information, knowledge, assumptions, opinions, etc. For example, the qualitative method is used to obtain information about theories, functionalities, processes, and obtain information from the subject area experts (Flemming and Noyes, 2021). This is very much specific and participants in this research are limited. They are expected to share quality information with their experience and exposure to certain backgrounds of research scopes. In contrary, the participants in the quantitative research tend to be more in number and the collection of data will be massive with wide range of opinions, thoughts, experiences, and feedbacks (Lee et al., 2019). These two methods are used in some cases are considered to be the mixed research and these types of researches are considered when there are multi-dimensional studies and research studies.

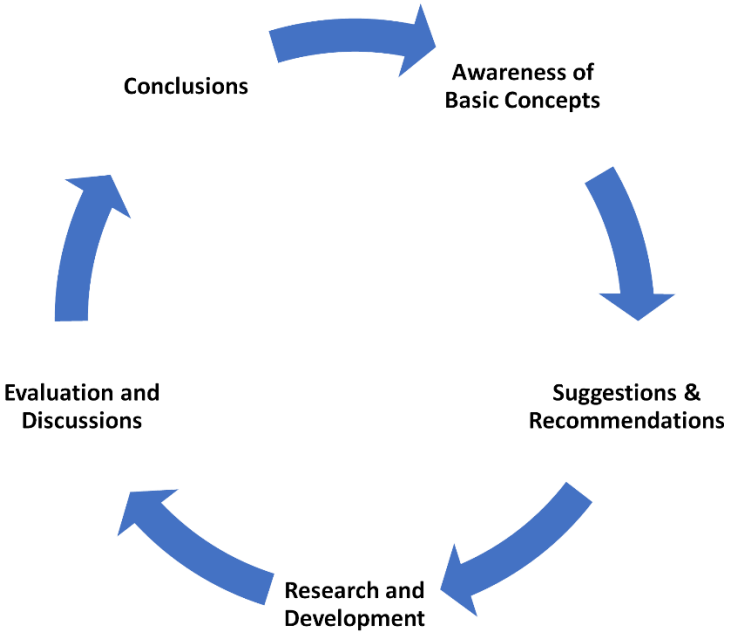
Finally, the secondary data analysis is considered mostly in studying the concepts on existing research outcomes, Sensex reports, etc. These kind of research activities takes place in the research organization and educational institutions to make some of the conclusive summaries for further studies or research programs.

Here in this research field experiments method is used to create a new mobile application that suits to integrate all the hospitals across India to meet the requirements of the population across the country. Therefore, the next section is going to explain the process of research design and creation of a mobile application in real-time environments.

3.3 Research Design and Creation

The mobile application design requires some of the important steps and they include to understand the




Awareness of
Basic Concepts
Suggestions &
Recommendations
Research and
Development
Evaluation and
Discussions
Conclusions

standards and academic standards to avoid any kind of ethical issues.

3.6 Summary

In this chapter author summarized the complete chapter. Author discussed about how java is far better than the other languages, and the benefits of the java and workflow of java in mobile application. Author explained how API's play a crucial role in mobile app development using

Android Studio by enabling developers to integrate third-party services, access device functionalities, improve app performance, create a better user experience, and develop apps more rapidly. API's can save time, resources, and provide more options for developers and users alike. Therefore, APIs are an essential component of modern mobile app development. According to author the android studio is why good for making mobile app, and how it worked and advantages of the android studio.

4 Implementation/Architecture Design/ App Design

4.1 Overview

This chapter discussed the overview of the study how the application developed using various software such as Java based Android Studio. Below Authors discussed about each and every step from finding the nearest and best hospitals and what are the services provides to the best treatment. In this chapter various Figure are given below which is discuss briefly like how the application and each steps Implemented using the java and android Studio. Mobile health technologies such as mobile apps and wearable devices can help patients manage their health and enable remote monitoring of patients. This can lead to improved patient engagement and better management of chronic conditions. Digitalization and integration of healthcare systems in India refers to the use of digital technologies to improve access to healthcare services and make healthcare delivery more efficient. This involves the use of electronic health records telemedicine, health information exchange mobile health, health analytics, and policies and regulations to ensure the effective implementation of digital healthcare technologies.

4.2 Architecture flow diagram

The architecture flow helps to understand the overall workflow of the healthcare application proposed in this research is shown in Fig. 4.1 and followed by the home page design using Android Studio is shown in Fig. 4.2.

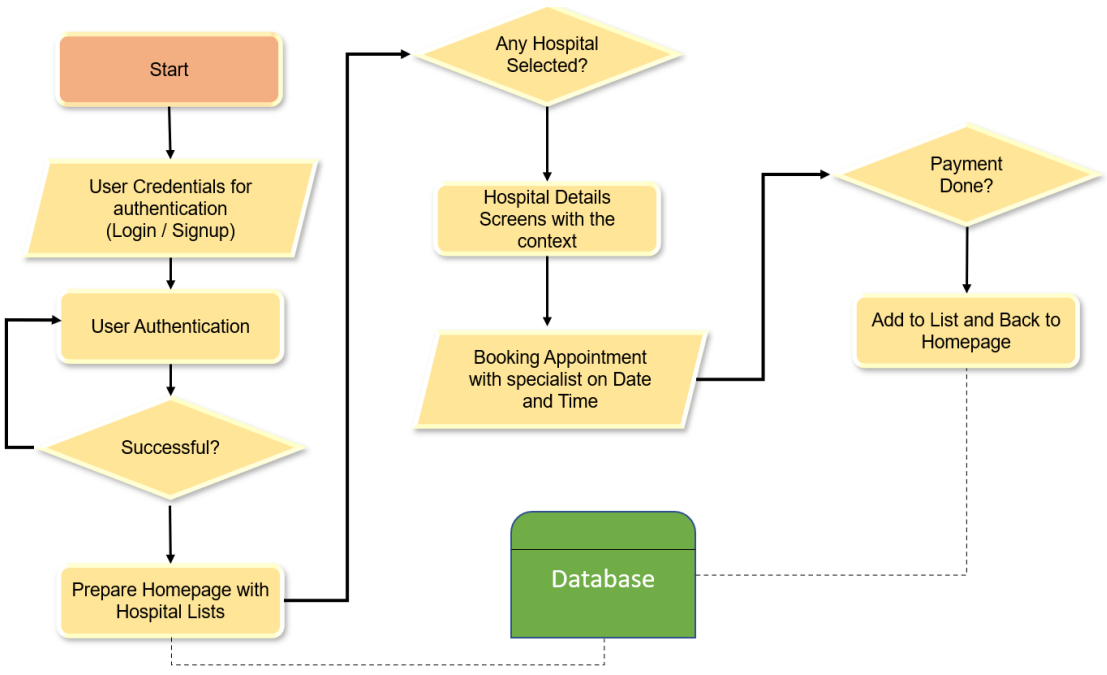


Figure 4.1 Architecture Flow Chart

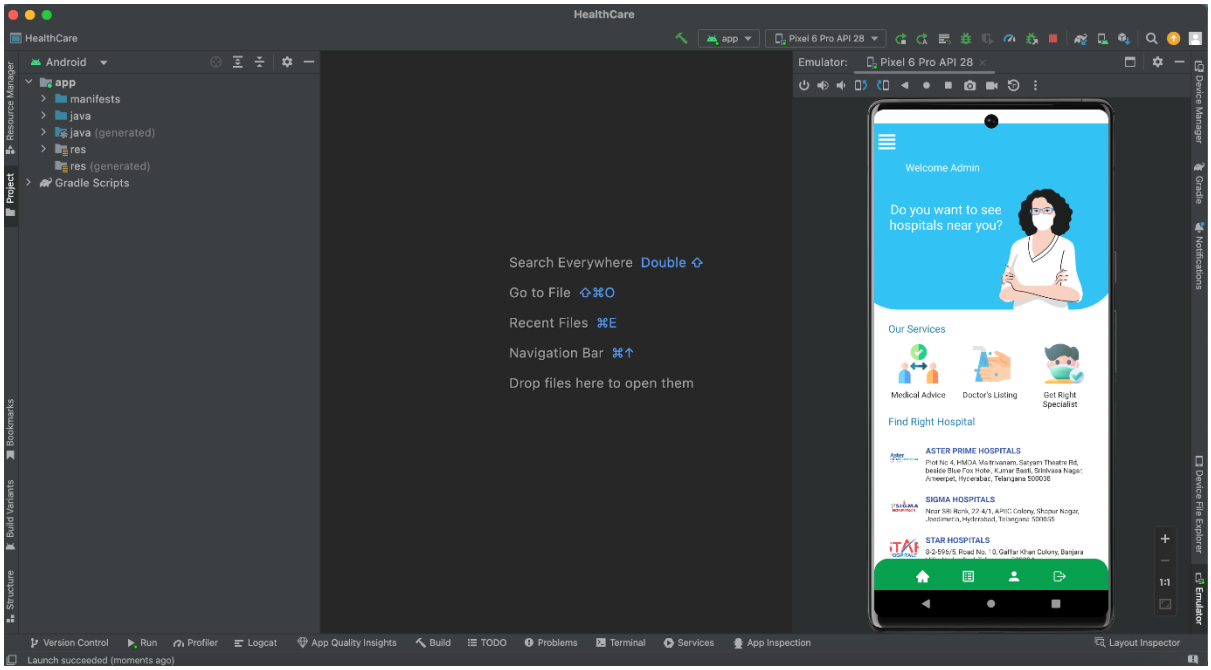


Figure 4.2 Home Page

```
public class MainActivity extends BaseActivity implements IHomeView, MenuAdapter.ItemClickListener,HospitalAdapter.ItemClickListner{


    public void onCreate(Bundle savedInstanteState) {
        super.onCreate(savedInstanteState);
        DataBindingUtil.setContentView( activity: this, R.layout.activity_main);

        presenter = new HomePresenter();
        presenter.setView(this);

        txtUserName = findViewById(R.id.txtUserName);
        imgProfile = findViewById(R.id.imgProfile);
        readyHospitals = findViewById(R.id.reacyHospitals);
        rvList = findViewById(R.id.rvItemList);
        drawerLayout = findViewById(R.id.drawer_layout);

        updateProfileInfo();

        menuAdapter = new MenuAdapter( context: this,navItems, listener: this);
        LinearLayoutManager linearLayoutManager = new LinearLayoutManager( context: this, LinearLayoutManager.VERTICAL, reverseLayout: false);
        rvList.setLayoutManager(linearLayoutManager);
        rvList.setItemAnimator(new DefaultItemAnimator());
        rvList.setAdapter(menuAdapter);
```

Listing 4.1 Code for Home Page Created for the Proposed Application using Java based Android Studio

In the Fig. 4.2, Home page will suggest the nearest hospital to users so that in emergency case it shouldn't create threat to life and get treat with better services. While visiting on the home

page this application suggests and the advice the various Services as best as it can provide, including the various medical advice, it will suggest the right Specialist based on the User the health condition where this proposed application is created using java and Android Studio. application for finding the nearest hospitals is a mobile app that uses location-based services to identify the user's current location and provide a list of nearby hospitals. An app for finding the nearest hospitals could be a useful tool for anyone seeking medical care or assistance, particularly in emergency situations. By providing a list of nearby hospitals with ratings, reviews, and other information, the app could help users make informed decisions about where to seek medical care. The app could use GPS to determine the user's location, and then search a database of hospitals to find the closest ones.

The app could provide a variety of information about each hospital, such as the hospital's name, address, phone number, website, and hours of operation. It could also provide ratings and reviews from other users, as well as information about the hospital's specialties, services, and facilities. Telemedicine allows patients to access healthcare services remotely through video conferencing, which can be particularly beneficial for patients in rural areas who may not have easy access to healthcare providers. Telemedicine can also help reduce the burden on healthcare providers and improve the efficiency of healthcare delivery. The flow diagram for login page is shown in Fig. 4.3 along with login page design in Fig. 4.4.

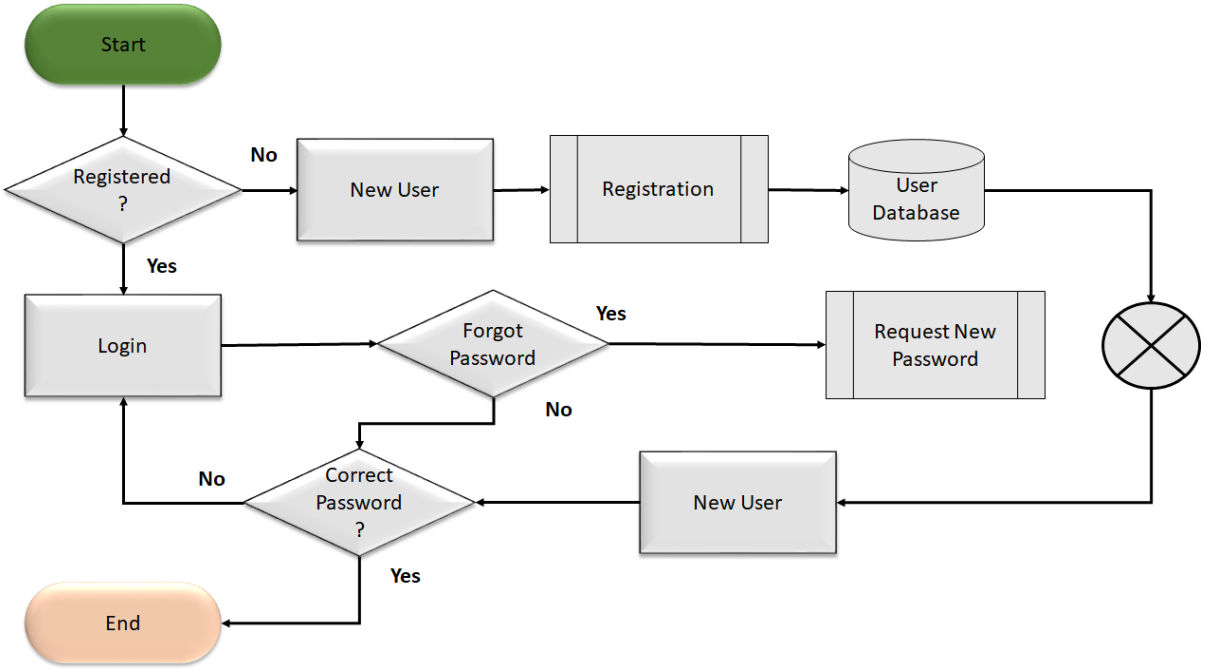


Figure 4.3 Flow Diagram for the Registration and Login Page

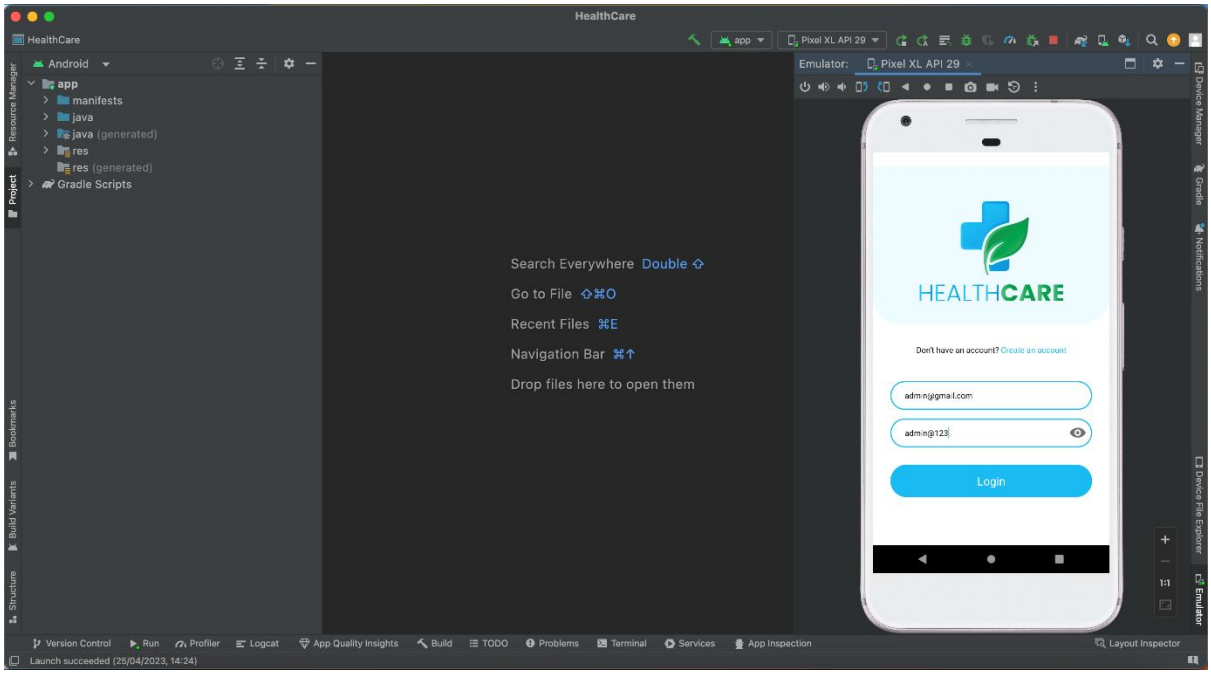


Figure 4.4 Login Page Design using Android Studio

```
public class LoginActivity extends BaseActivity implements ILoginView {

    @Override
    public void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        presenter = new LoginPresenter();
        presenter.setView(this);

        edtEmail = findViewById(R.id.edtEmail);
        edtPass = findViewById(R.id.edtPass);

        imgCardEmail = findViewById(R.id.imgCardEmail);
        imgCardPassword = findViewById(R.id.imgCardPassword);

        edtEmail.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence s, int start, int count, int after) {

            }

            @Override
            public void onTextChanged(CharSequence s, int start, int before, int count) {
                if (s.length() > 0) {
```

Listing 4.2 Code for Login Page Created for the Proposed Application

In the Fig. 4.4, after visiting the home page it will come to next page as Login page, where the user needs to enter personal details like Gmail ID and Password, after entering the user details, the enter on the given Blue Login Button.

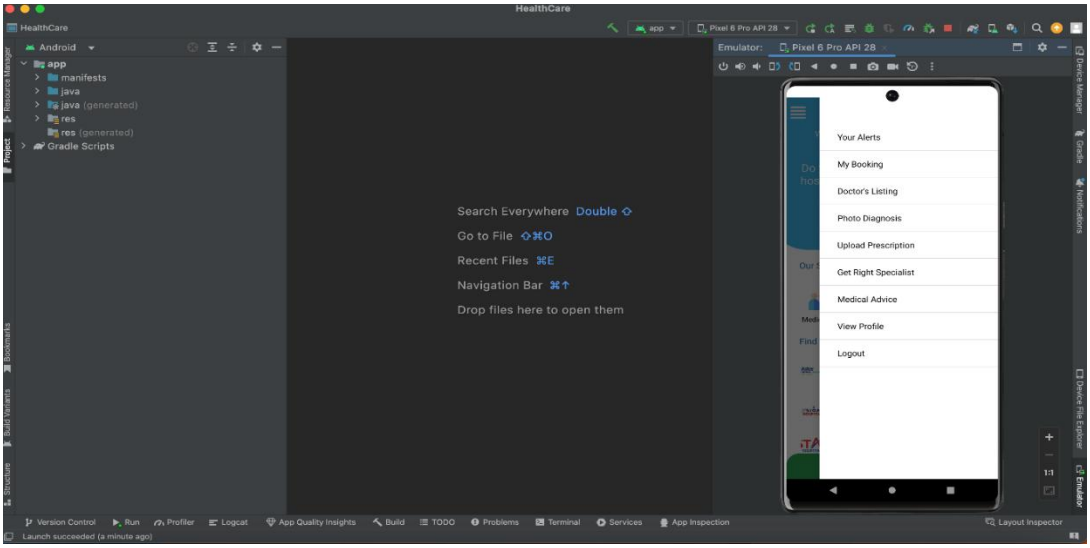


Figure 4.5 Menu Bar at the Top Left Corner of the Page

In the Fig. 4.5 After the User Login, it will show the Menu Bar at Top left Corner of the current page, it will show the various details such as Bookings, Doctor's listing they have suggest by this application, photo diagnosis where the user need to upload injury photos, test reports, MRI reports, X rays etc. So that based on the user health and uploaded photos, this application can suggest the right Doctor and best medicines. On the Bottom of the menu bar, it will show the User's Profile and Finally Logout Option. On the other hand, the Edit profile page is also important to update the personal details and flow diagram is shown in Fig. 4.6 along with the page created for this in Fig. 4.7.

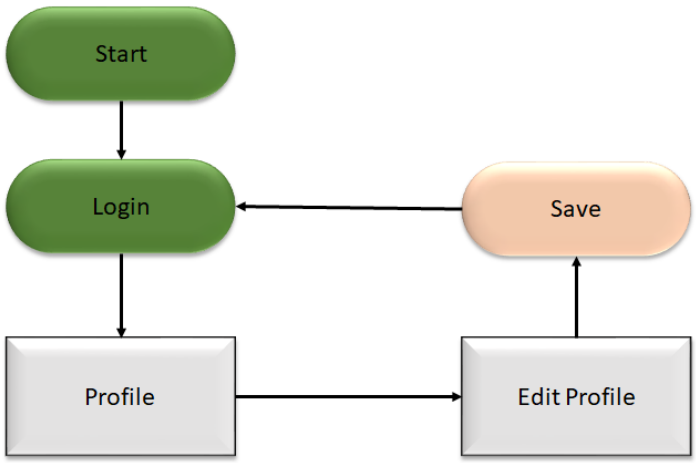


Figure 4.6 Flow Diagram for Edit Profile Page

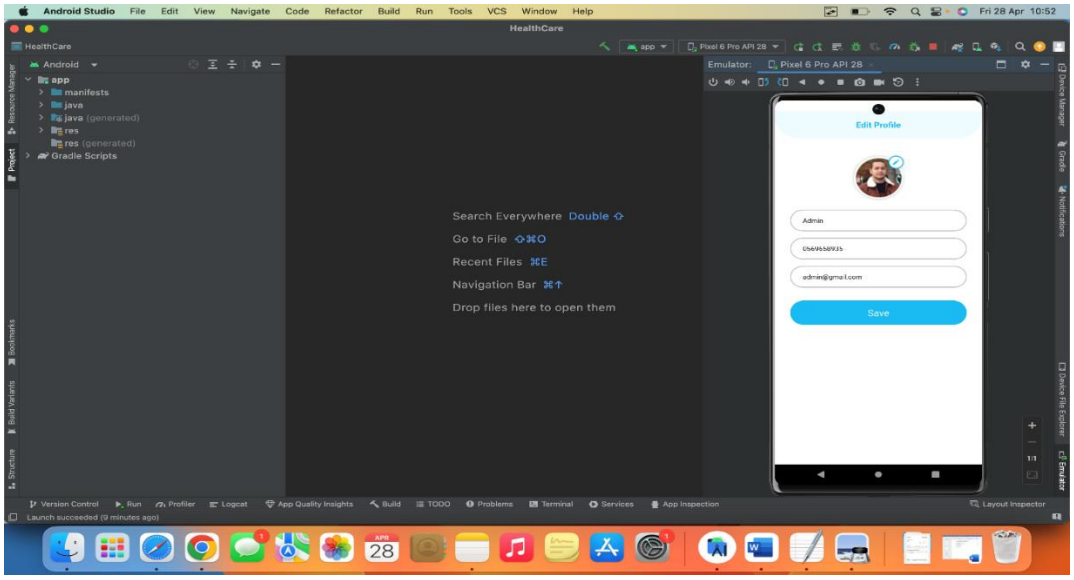


Figure 4.7 Edit Profile Page Created for the APP

```
public class EditProfileActivity extends BaseActivity implements IEditProfileView {


    @Override
    public void onCreate(Bundle savedInstanceState){...}

    public void onClick(View view){
        switch (view.getId()){
            case R.id.txtSave:
                if (TextUtils.isEmpty(edtName.getText().toString())){
                    toast("Please enter your name");
                }else if (TextUtils.isEmpty(edtEmail.getText().toString().trim())
                        || !isValidEmail(edtEmail.getText())) {
                    toast("Please enter valid email");
                }else if (TextUtils.isEmpty(edtMobile.getText().toString().trim())||
                        !isValidPhoneNumber(edtMobile.getText())) {
                    toast("Please enter valid phone number");
                }else {
                    if (NetworkCheck.isConnected( context: this)) {
                        MultipartBody.Part user_image = null;
                        if (uriProfile != null) {
                            if (isimageFromGallery) {
                                String selectedPath = FileUtils.getPath( context: this, uriProfile);
                                File file = new File(selectedPath);
                                RequestBody requestFile = RequestBody.create(MediaType.parse( $this$toMediaTypeOrNull: "image/*"), file);
```

Listing 4.3 Profile Page of the Patient in the Application

In the Fig. 4.7 the user can edit personal information such as name, mobile number and Gmail so that they may save their profile for the further process for finding the right hospitals and best doctor. So that based on their reports and health conditions Doctor may give the right medicines with respect to their heath condition and the problem that user may facing. At last User needs to save their personal details by hit on the Save button. This application also come up with feature like user can Upload their profile Photo. Implementing health analytics can help

healthcare providers make better-informed decisions by analysing data on patient health outcomes, treatment effectiveness, and healthcare costs. This can help identify areas for improvement and optimize healthcare delivery. Policies and regulations need to be in place to ensure that digital healthcare technologies are safe, secure, and effective. These policies and regulations should also address issues related to data privacy and security.

4.3 Testing the Offline Access of the Application

The mobile app for the healthcare was designed and shown some of the functionalities in a most significant way. During the offline mode the app is opening the home screen but not able go for the further operations as shown in Fig. 4.8. At the same time the application is showing some of the features during offline include the list of hospitals to limited extent. Otherwise, more information is only available after login with the application that happens only after getting into online mode. This app was tested with different android mobile set ups with different versions used by companies like Samsung, MI, Oppo, etc.

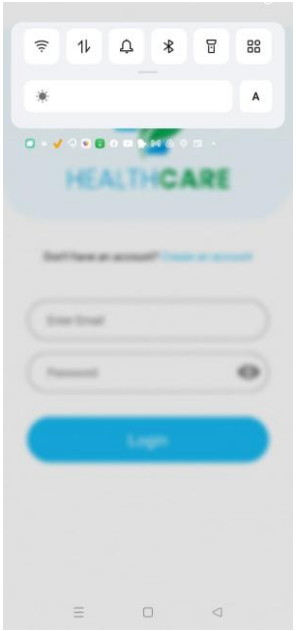


Figure 4.8 Testing Results of APP during Offline Mode

The performance testing was carried out by using the third-party support from the lambdatest.com website towards testing the code and developed screens of the applications. This code seems to express better results and performance as expected from the initial design objectives. It measured some of the important features of the application towards routing the data, storing and downloading the data, and some other features as well.

5 Analysis, Findings and Validating

5.1 Overview

Chapter 5 consists of Analysis and Findings related to the proposed Healthcare app and the internal Sections provide a detain description of all the features of the app which makes the users accessible to immediate consultations of doctors in healthcare industry. Section 5.2 describes about the purpose of the study; Section 5.3 displays the Doctors Listing and the Home page of the healthcare app and the complete appointment process of the users. Finally, Section 5.4 details about the loading and Feature Detection using the Camera Option in the App and accessing the pictures through media from one's mobiles.

5.2 Purpose of Study

Healthcare apps serve various purposes, including tracking and monitoring health conditions, providing access to medical information and resources, facilitating communication between patients and healthcare providers, enabling remote consultations and telemedicine, promoting healthy lifestyle behaviours, and improving medication management and adherence. They aim to enhance the overall quality of healthcare delivery and empower individuals to take a more proactive role in managing their health.

5.3 Implementation of Appointment Booking Using Proposed Application

The ways to take the appointments of doctors vary according to multiple hospitals, including like making a phone call to the respective hospital, or taking a live appointment by entering a respective hospital of customer's choice. But the newly innovated healthcare system provides an easy option for customers to take the appointments of doctors by sitting at home through their proposed app. The below figures display all the implemented steps for the patients to take the appointments at their own convenience. The flow diagram for the list of doctors is shown in Fig. 5.1 along with the screen of doctors listing page in Fig. 5.2.

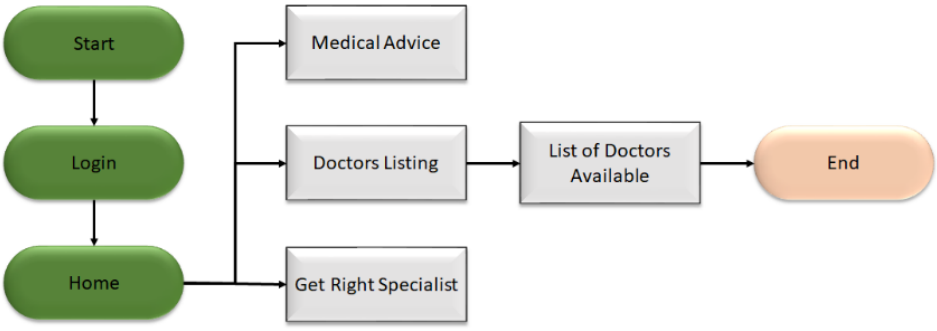


Figure 5.1 Flow Diagram for Doctors listing

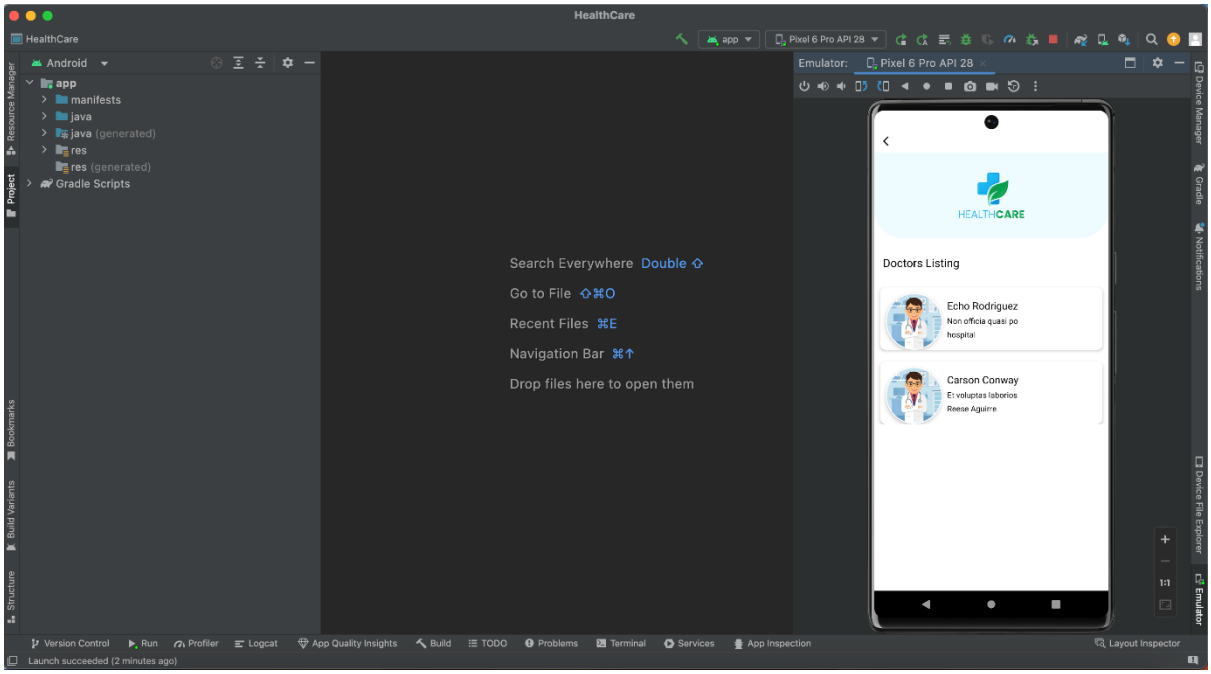
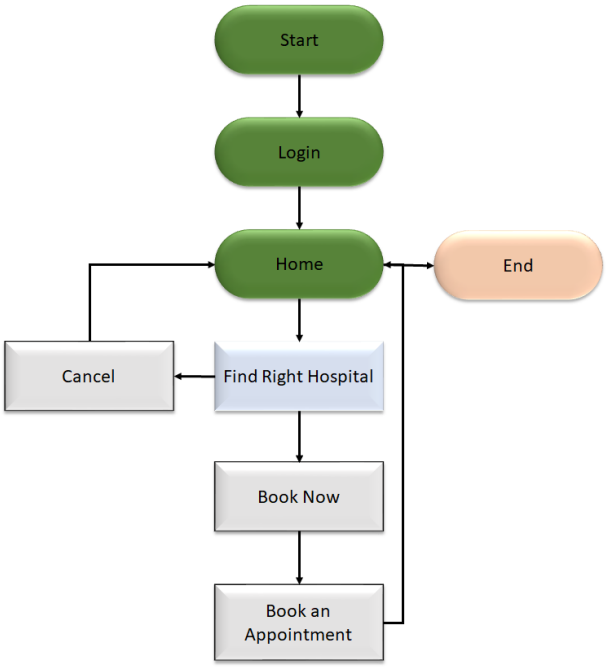


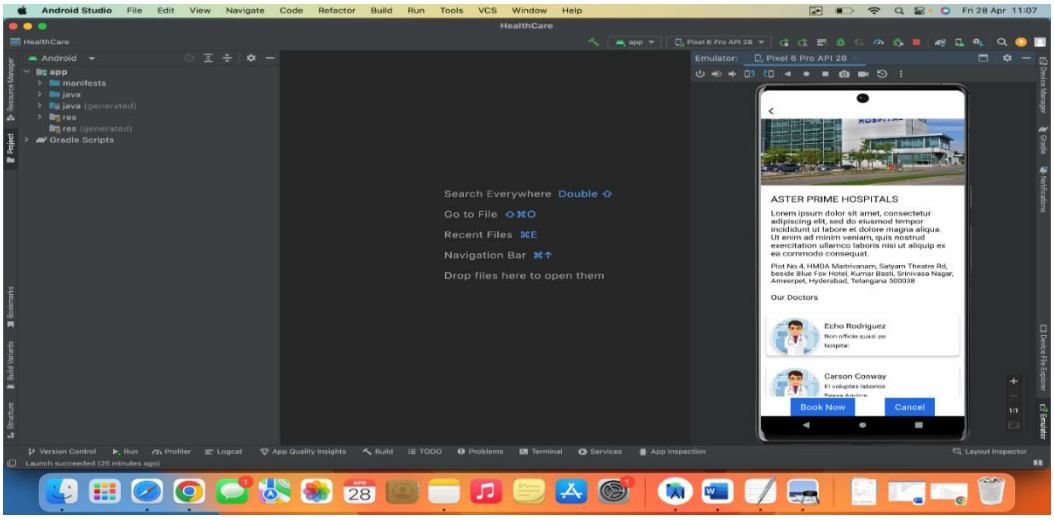
Figure 5.2 Doctors Listing Page

```
public class DoctorListing extends BaseActivity implements IDoctorView, DoctorAdapter.ItemClickListner {

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_doctorlisting);

        fsProductPresenter = new DoctorLPresenter();
        fsProductPresenter.setView(this);

        userData = new SharedPreferencesData(getContext());

        recydoc= findViewById(R.id.recyDoctors);

        fsProdAdapter = new DoctorAdapter(itemList, getContext(), listner: this);
        LinearLayoutManager gridLayoutManager = new LinearLayoutManager(getContext(), RecyclerView.VERTICAL, reverseLayout: false);
        recydoc.setLayoutManager(gridLayoutManager);
        recydoc.setAdapter(fsProdAdapter);

        fsProductPresenter.Doctor(getContext(), userData.getACCESS_TOKEN());
    }

    public void onClick(View view){
        switch (view.getId()){
            case R.id.imgback:
                startActivity(new Intent( packageContext: this, MainActivity.class));
                break;
```

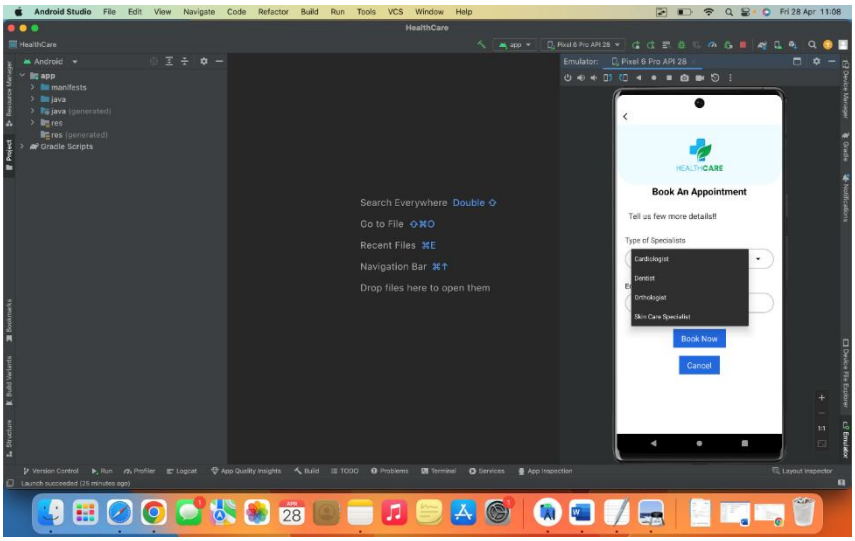
Listing 5.1 Doctors Listing – Helps to find the overall doctors list for any particular area of medical problem

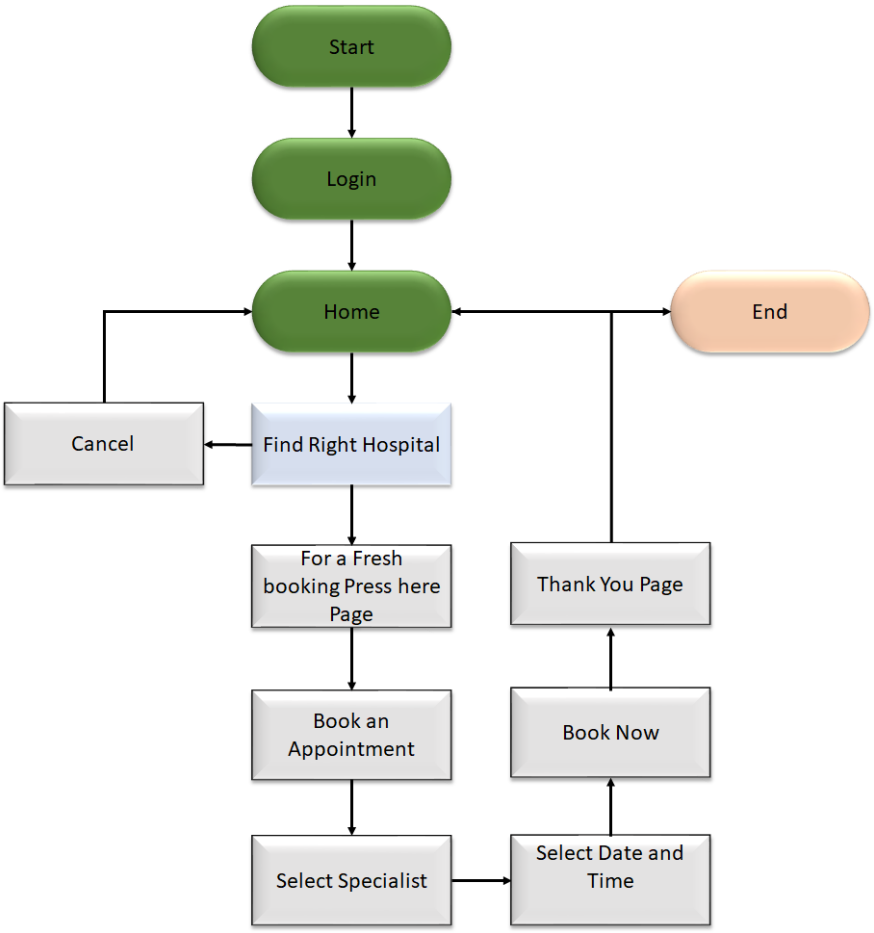
The first step in taking a doctor's appointment involves identification of the doctors according to their specialisations, and the hospitals they currently belong to. Additionally, it also provides information on the hospitals which are opened nearby the patient's premises. It is provided as an option making the patients to choose a doctor of a respective hospital of their own choice. And they can also know various other doctors who are diagnosing the same injury or disease.





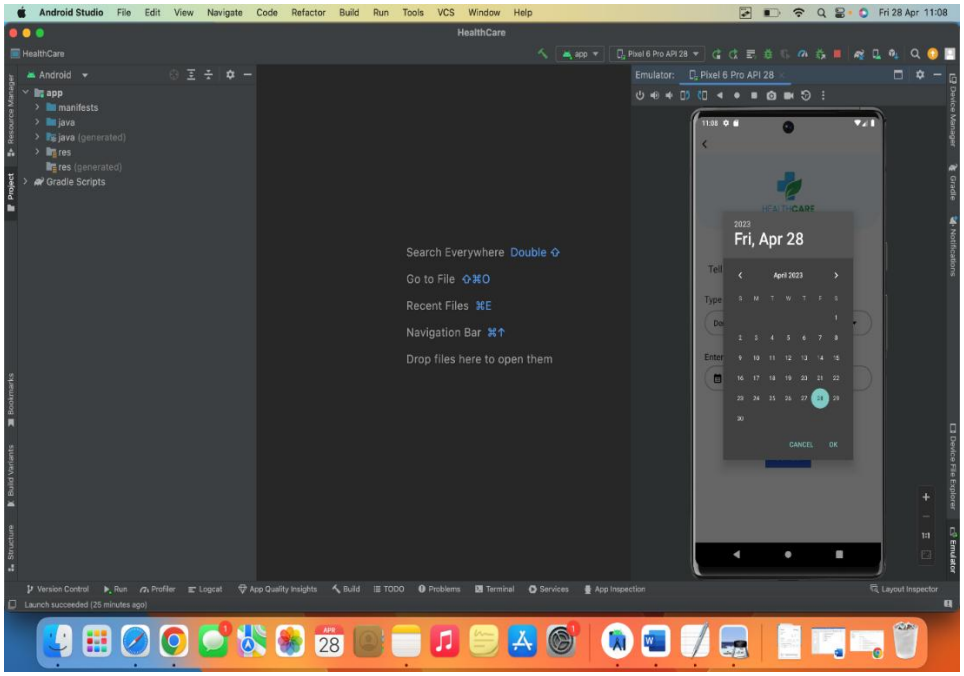
```
public class HospitalDetails extends BaseActivity implements IHomeView, View.OnClickListener, DoctorAdapter.ItemClickListner{

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hospital_details);
        product_id = getIntent().getStringExtra( name: "product_id");
        presenter = new HomePresenter();
        presenter.setView(this);
        profiledata = new SharedPreferencesData( context: this);
        imgback = findViewById(R.id.imgback);
        imgProd = findViewById(R.id.imgProd);
        consRoot = findViewById(R.id.consroot);
        txtcat = findViewById(R.id.txtCategory);
        txttitle = findViewById(R.id.txtTitle);
        txtdesc = findViewById(R.id.txtDes);
        txtcart = findViewById(R.id.txtcart);

        recydoc= findViewById(R.id.recyBook);

        fsProdAdapter = new DoctorAdapter(itemList, getContext(), listner: this);
        LinearLayoutManager gridLayoutManager = new LinearLayoutManager(getContext(), RecyclerView.VERTICAL, reverseLayout: false);
        recydoc.setLayoutManager(gridLayoutManager);
        recydoc.setAdapter(fsProdAdapter);

        imgback.setOnClickListener(this);
```



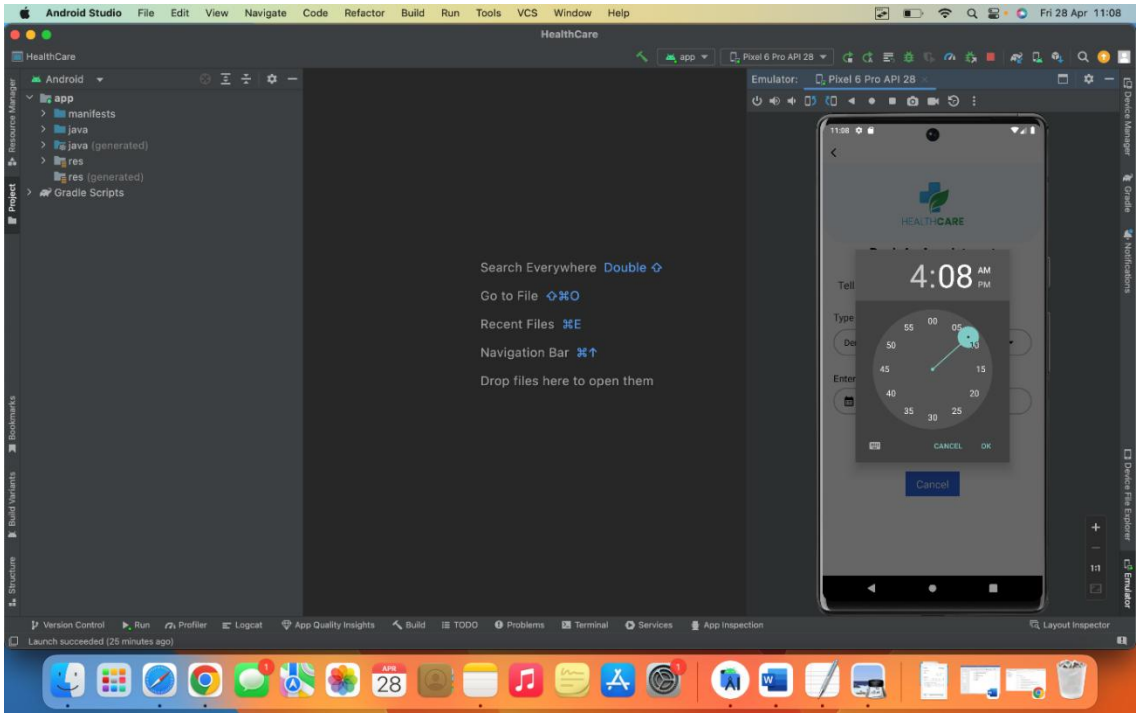



```
ArrayAdapter<CharSequence> adapterpeople=ArrayAdapter.createFromResource( context: this, R.array.specialist,
        R.layout.spin_drop_down_view);
adapterpeople.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
txtspin.setAdapter(adapterpeople);
txtspin.setOnItemSelectedListener(this);
special = txtspin.getSelectedItem().toString();
```

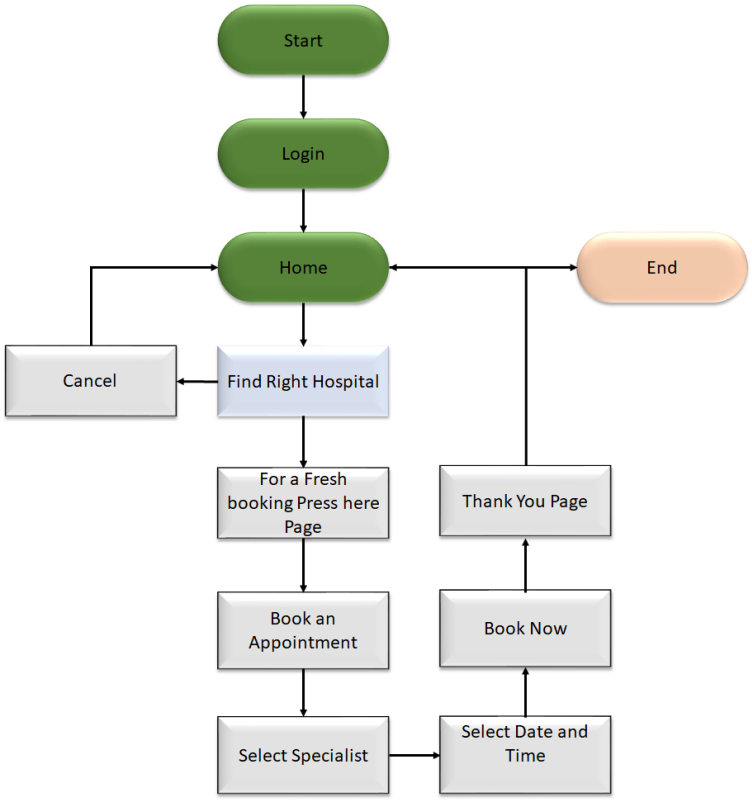
```
<string-array name="specialist">
    <item>Cardiologist</item>
    <item>Dentist</item>
    <item>Orthologist</item>
    <item>Skin Care Specialist</item>
</string-array>
```

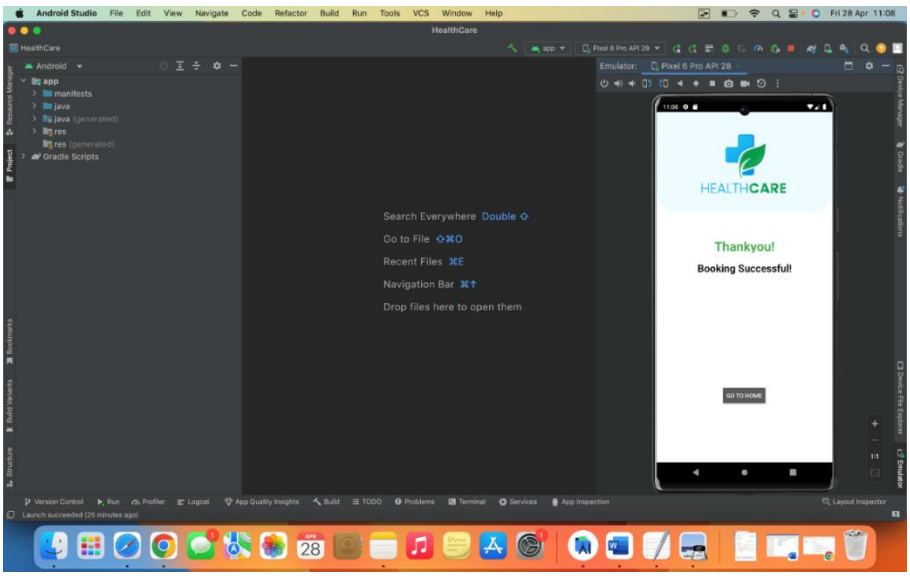



```
case R.id.consDate:
    final Calendar c = Calendar.getInstance();
    mYear = c.get(Calendar.YEAR);
    mMonth = c.get(Calendar.MONTH);
    mDay = c.get(Calendar.DAY_OF_MONTH);
    DatePickerDialog datePickerDialog = new DatePickerDialog( context: this,
            new DatePickerDialog.OnDateSetListener() {
                @Override
                public void onDateSet(DatePicker view, int year,
                                      int monthOfYear, int dayOfMonth) {
                    edtdate.setText(dayOfMonth + "-" + (monthOfYear + 1) + "-" + year);
                }
            }, mYear, mMonth, mDay);
    datePickerDialog.show();
    break;
```

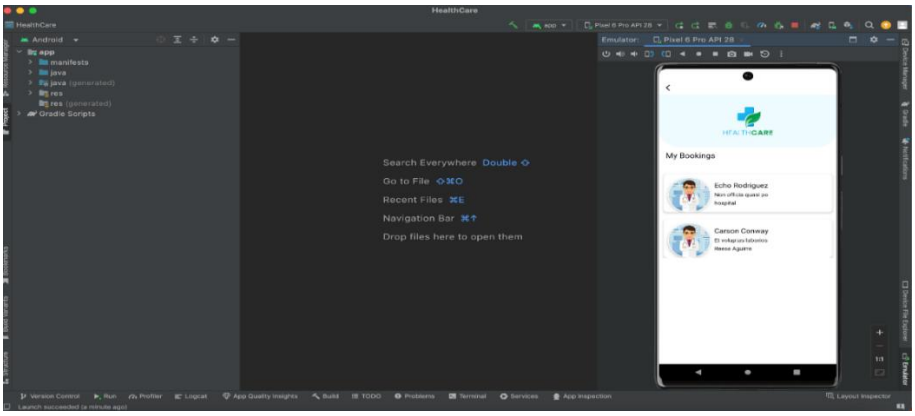



```
case R.id.consTime:
    final Calendar ca = Calendar.getInstance();
    mHour = ca.get(Calendar.HOUR_OF_DAY);
    mMinute = ca.get(Calendar.MINUTE);

    // Launch Time Picker Dialog
    TimePickerDialog timePickerDialog = new TimePickerDialog( context: this,
            new TimePickerDialog.OnTimeSetListener() {

                @Override
                public void onTimeSet(TimePicker view, int hourOfDay,
                                      int minute) {

                    edttime.setText(hourOfDay + ":" + minute);
                }
            }, mHour, mMinute, is24HourView: false);
    timePickerDialog.show();
    break;
```

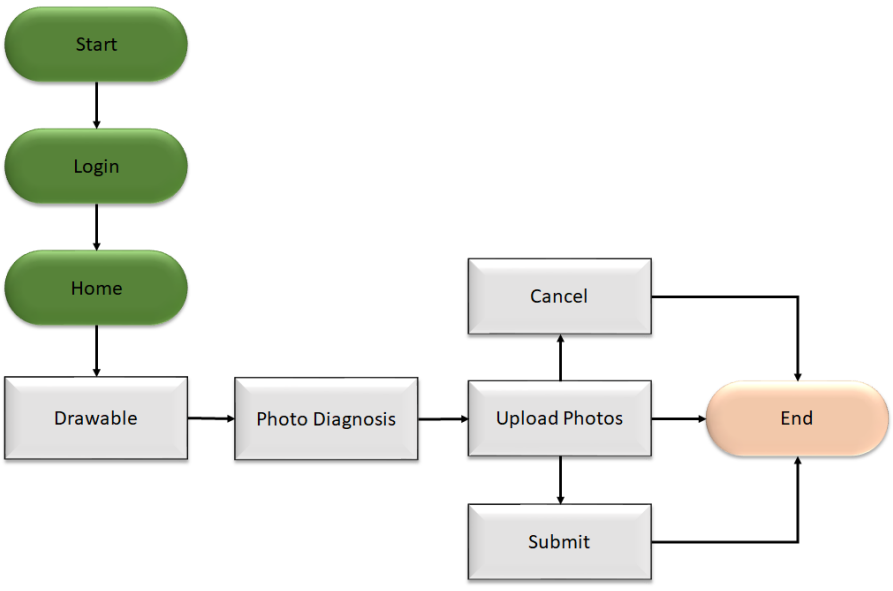

Start
Login
Home
End
Cancel
Find Right Hospital
For a Fresh booking Press here Page
Thank You Page
Book an Appointment
Book Now
Select Specialist
Select Date and Time

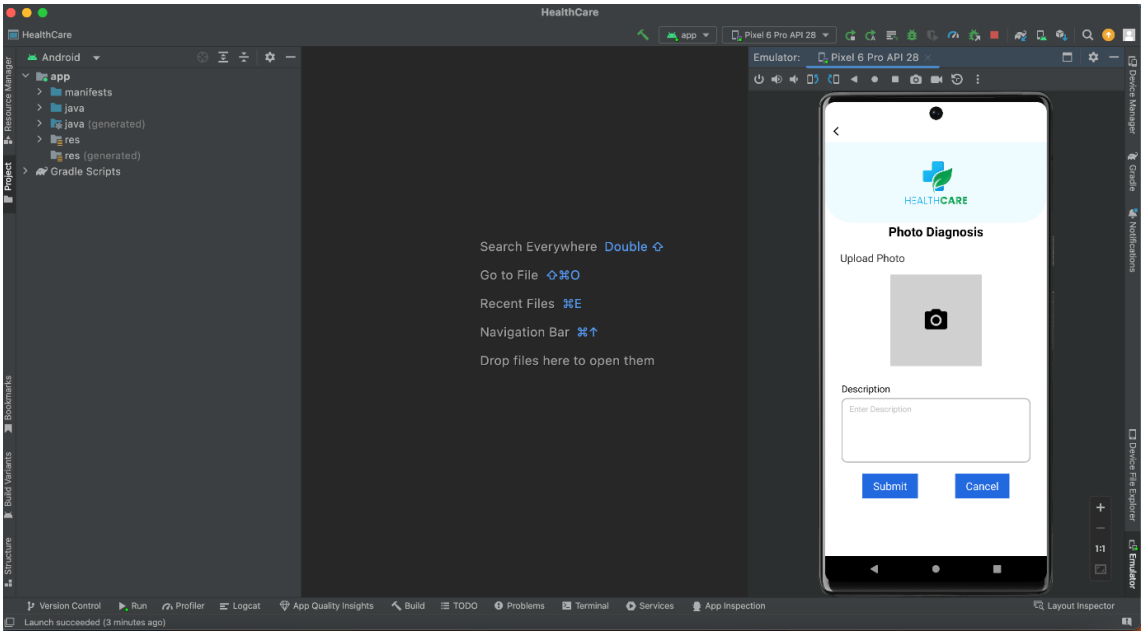

Android Studio
HealthCare
Search Everywhere Double
Go to File
Recent Files
Navigation Bar
Drop files here to open them
HEALTHCARE
Thankyou!
Booking Successful!
GO TO HOME

```java
public class PayemntSuccessfull extends AppCompatActivity {
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_payment);

        @SuppressLint({"MissingInflatedId", "LocalSuppress"}) Button btn = findViewById(R.id.home_btn);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                finishAffinity();
                change_home();
            }
        });
    }

    1 usage
    private void change_home() {
        Intent intent = new Intent( packageContext: this, MainActivity.class);
        startActivity(intent);
    }
}
```

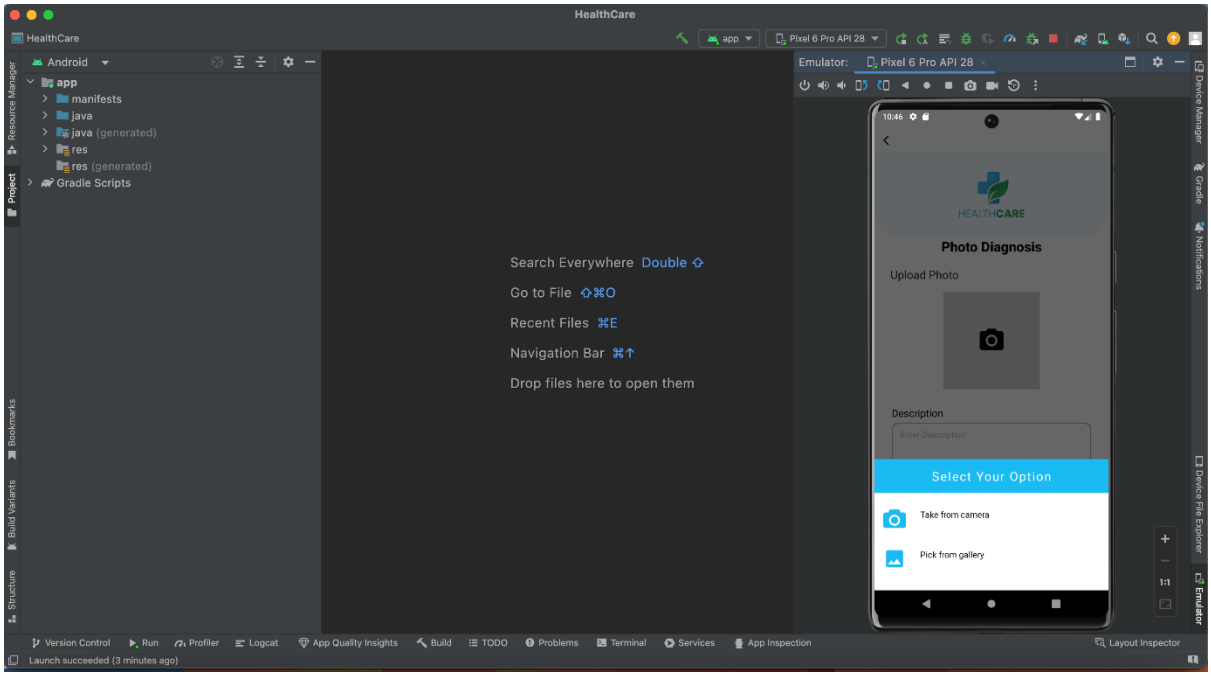


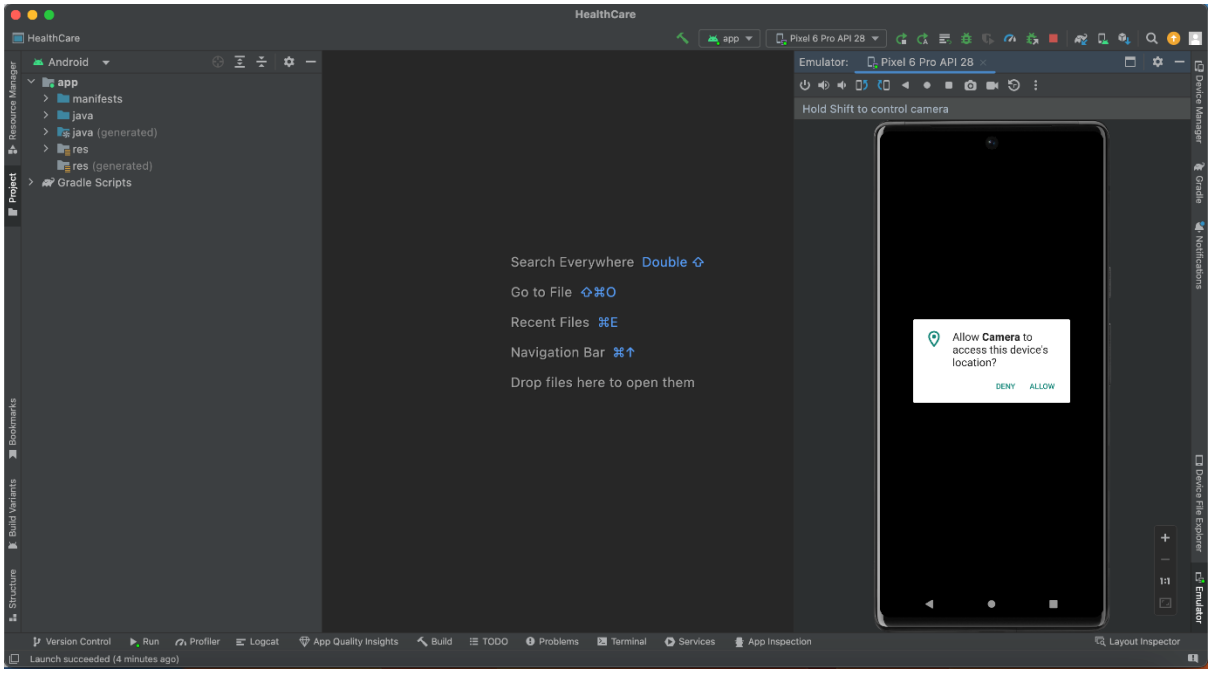
```java
public class Mybookings extends BaseActivity implements IDoctorView, DoctorAdapter.ItemClickListner {

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_mybookinglisting);

        fsProductPresenter = new DoctorLPresenter();
        fsProductPresenter.setView(this);

        userData = new SharedPreferencesData(getContext());

        recydoc= findViewById(R.id.recyBook);

        fsProdAdapter = new DoctorAdapter(itemList, getContext(),  listner: this);
        LinearLayoutManager gridLayoutManager = new LinearLayoutManager(getContext(), RecyclerView.VERTICAL,  reverseLayout: false);
        recydoc.setLayoutManager(gridLayoutManager);
        recydoc.setAdapter(fsProdAdapter);

        fsProductPresenter.Doctor(getContext(), userData.getACCESS_TOKEN());
    }
```

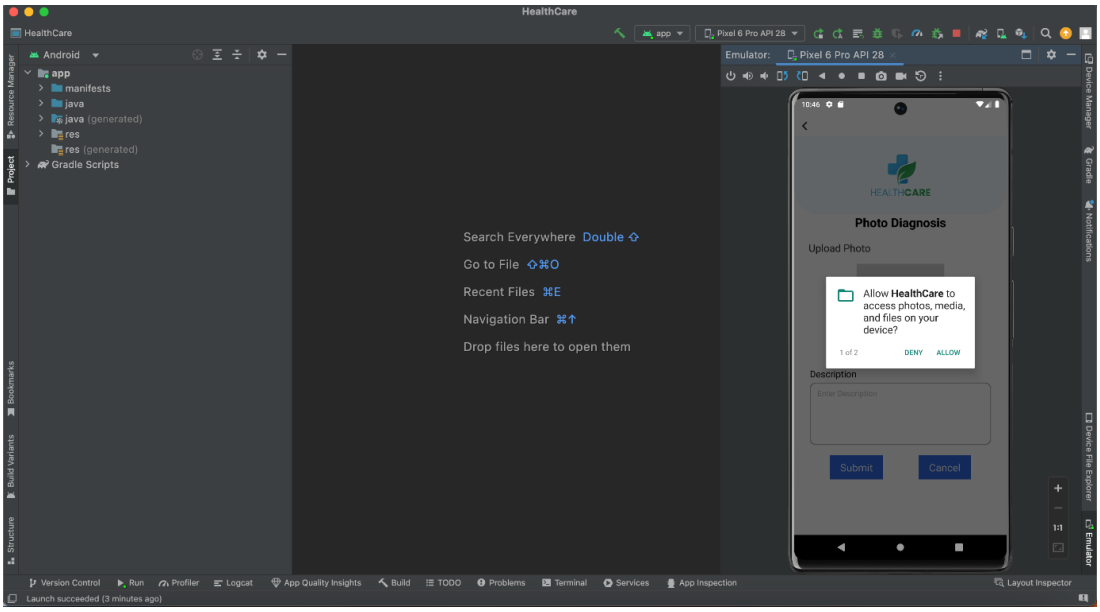


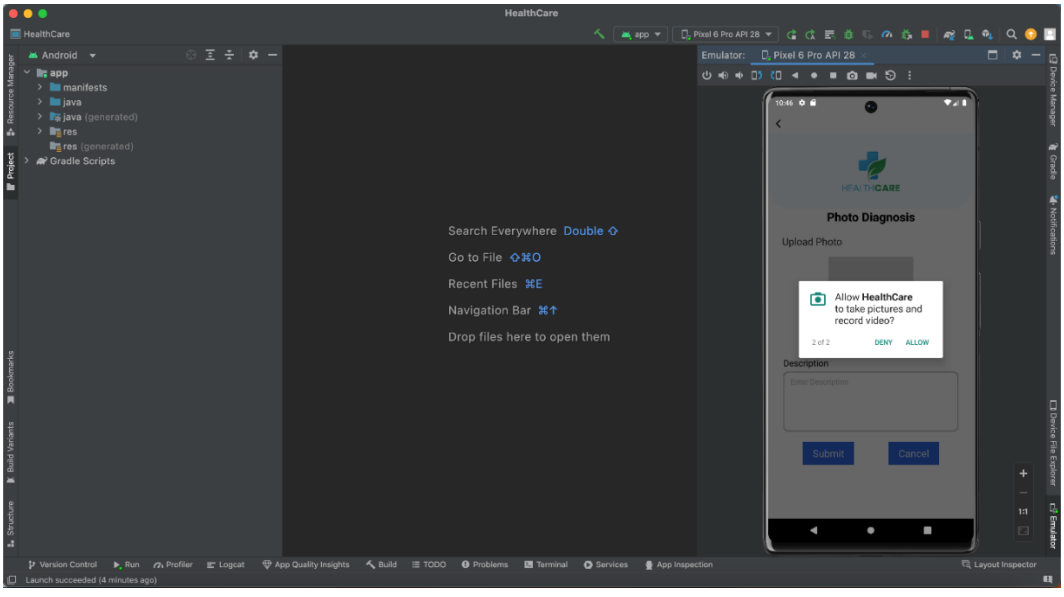


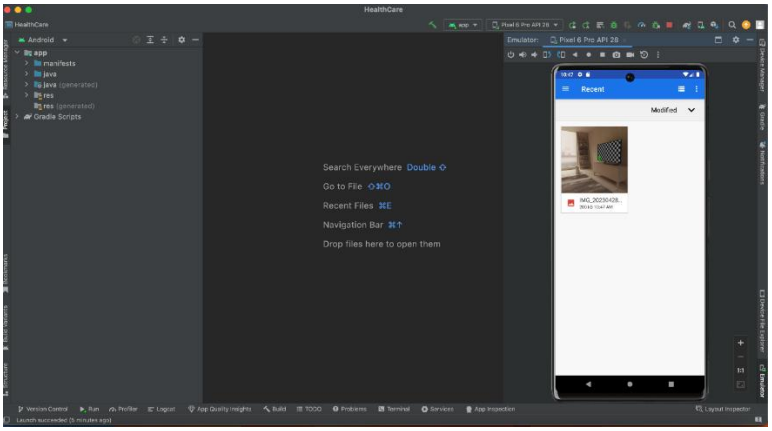

```
public void onClick(View view){
    switch (view.getId()){
        case R.id.submit:
            if (TextUtils.isEmpty(edtdes.getText().toString())){
                toast("Please enter description");
            }else {
                if (NetworkCheck.isConnected( context: this)) {
                    MultipartBody.Part user_image = null;
                    if (uriProfile != null) {
                        if (isimageFromGallery) {
                            String selectedPath = FileUtils.getPath( context: this, uriProfile);
                            File file = new File(selectedPath);
                            RequestBody requestFile = RequestBody.create(MediaType.parse( $this$toMediaTypeOrNull: "image/*"), file);
                            user_image = MultipartBody.Part.createFormData( name: "image", file.getName(), requestFile);
                        } else {
                            String fileName = new File(uriProfile.getPath()).getName();
                            File actualFile = new File(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES), new File(uriProfile.getPath()).getName());
                            if (actualFile != null) {
                                user_image = MultipartBody.Part.createFormData( name: "image", fileName, RequestBody.create(MediaType.parse( $this$toMediaTypeOrNull: "multipart/form-data"), actualFile));
                            } else {
                                user_image = MultipartBody.Part.createFormData( name: "image", filename: "", RequestBody.create(MediaType.parse( $this$toMediaTypeOrNull: "multipart/form-data"), content: ""));
                            }
                        }
                    }
                    presenter.PhotoD( context: this, user_image,edtdes.getText().toString(), userData.getACCESS_TOKEN());
                }
            }
            break;
```

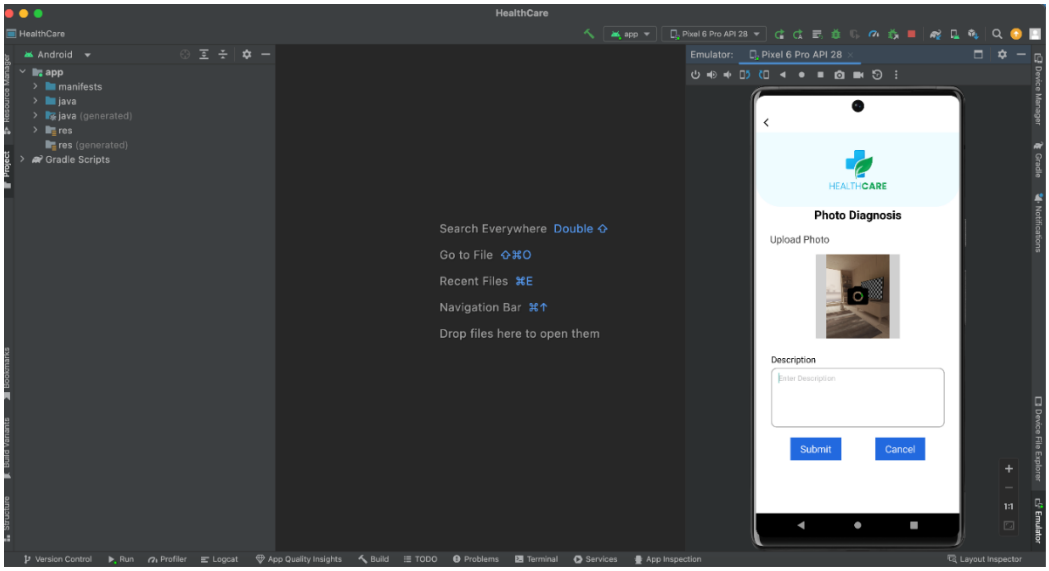

HealthCare
Search Everywhere Double
Go to File
Recent Files
Navigation Bar
Drop files here to open them
Photo Diagnosis
Upload Photo
Description
Select Your Option
Take from camera
Pick from gallery

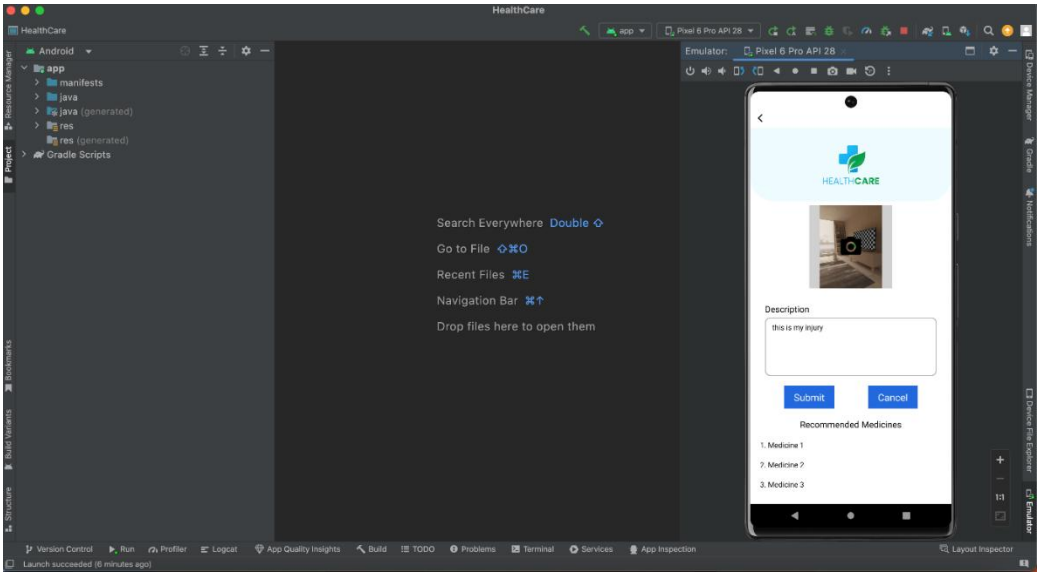

HealthCare
Search Everywhere Double
Go to File
Recent Files
Navigation Bar
Drop files here to open them
Hold Shift to control camera
Allow Camera to access this device's location?
DENY
ALLOW

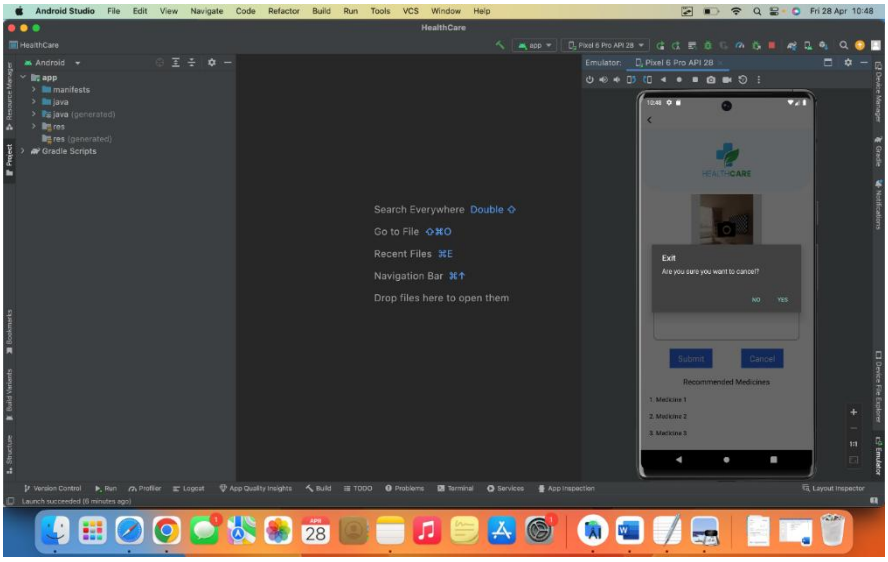

HealthCare
Search Everywhere Double ⇧
Go to File ⇧⌘O
Recent Files ⌘E
Navigation Bar ⌘↑
Drop files here to open them
Photo Diagnosis
Upload Photo
Allow HealthCare to access photos, media, and files on your device?
DENY
ALLOW
Description
Submit
Cancel


HealthCare
Search Everywhere Double ⇧
Go to File ⇧⌘O
Recent Files ⌘E
Navigation Bar ⌘↑
Drop files here to open them
Photo Diagnosis
Upload Photo
Allow HealthCare to take pictures and record video?
DENY
ALLOW
Description
Submit
Cancel


HealthCare
Search Everywhere Double
Go to File
Recent Files
Navigation Bar
Drop files here to open them
Recent
Modified


HealthCare
Android
app
manifests
java
res
Gradle Scripts
Search Everywhere Double
Go to File
Recent Files
Navigation Bar
Drop files here to open them
Emulator: Pixel 6 Pro API 28
HEALTHCARE
Photo Diagnosis
Upload Photo
Description
Submit
Cancel


HealthCare
Search Everywhere Double
Go to File
Recent Files
Navigation Bar
Drop files here to open them
HEALTHCARE
Description
this is my injury
Submit
Cancel
Recommended Medicines
1. Medicine 1
2. Medicine 2
3. Medicine 3


Android Studio
HealthCare
Search Everywhere Double
Go to File
Recent Files
Navigation Bar
Drop files here to open them
Exit
Are you sure you want to cancel?
NO
YES
Submit
Cancel
Recommended Medicines

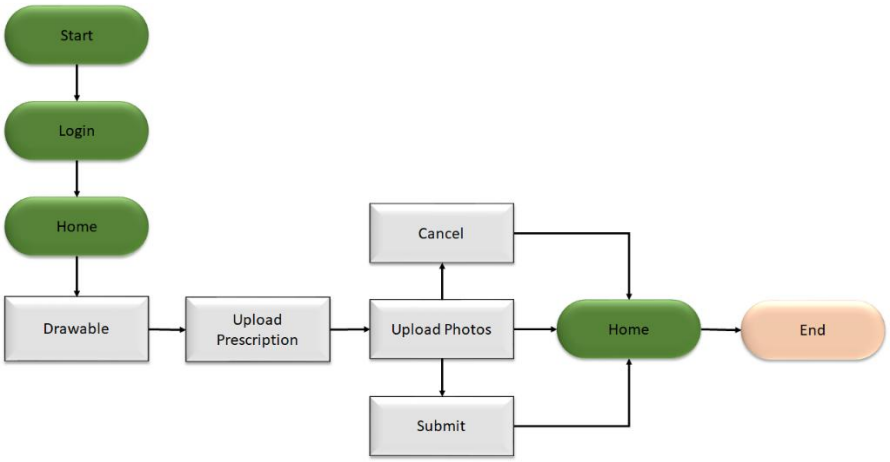


Figure 5.22 Flow Diagram for Upload Prescription

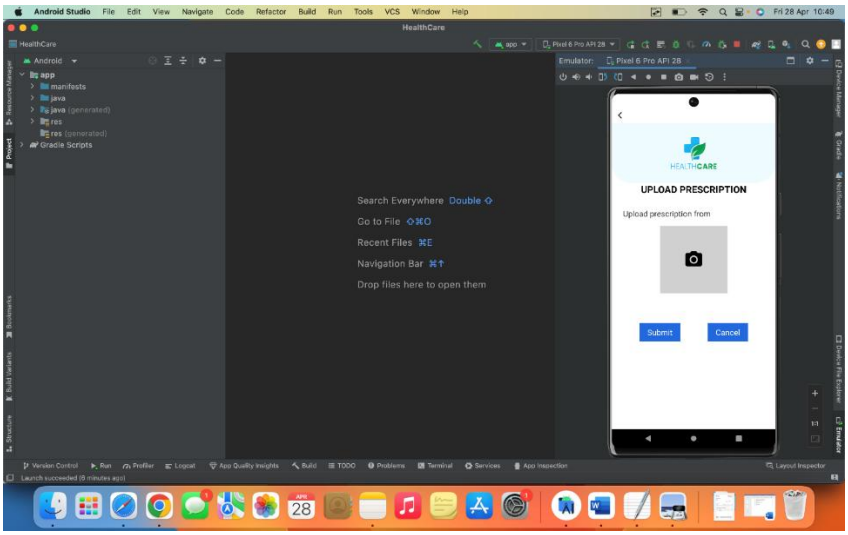


Figure 5.23 Upload Prescription for Second Opinion/Medical Progress

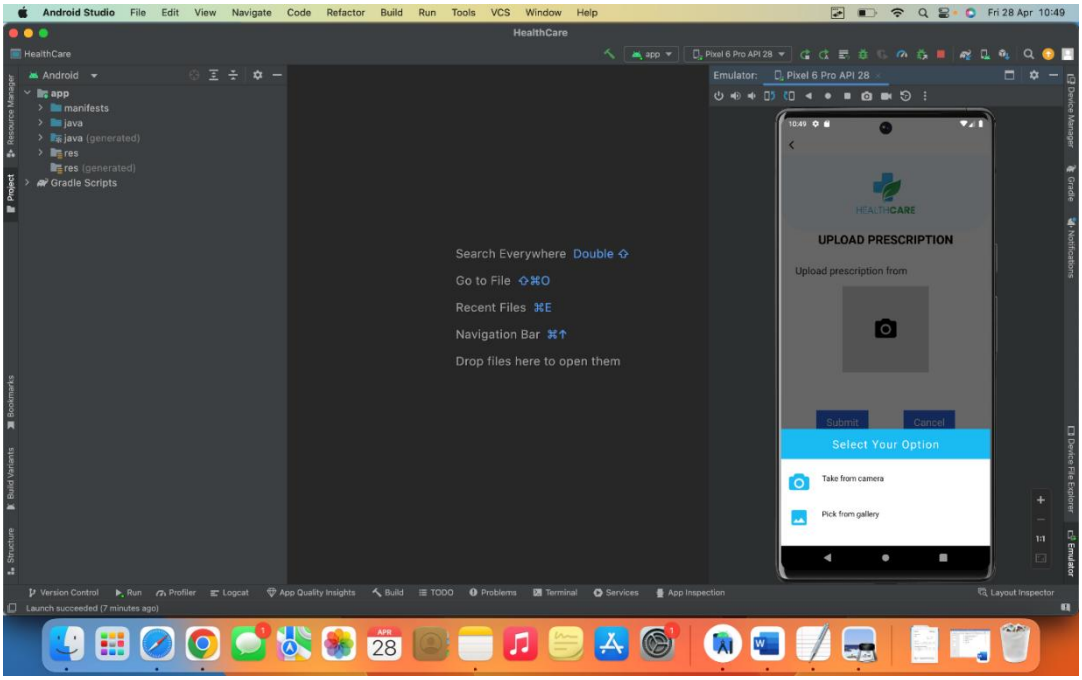


Figure 5.24 Uploading the Images of Prescriptions or to take a Picture

In Figure 5.25, for patient second opinion need to upload the past prescription for that first user open the camera on his mobile or pc and take a picture of the required prescription and after take the pic the next part is upload the pic into the upload prescription box and press okay button. The prescription is uploaded automatically.

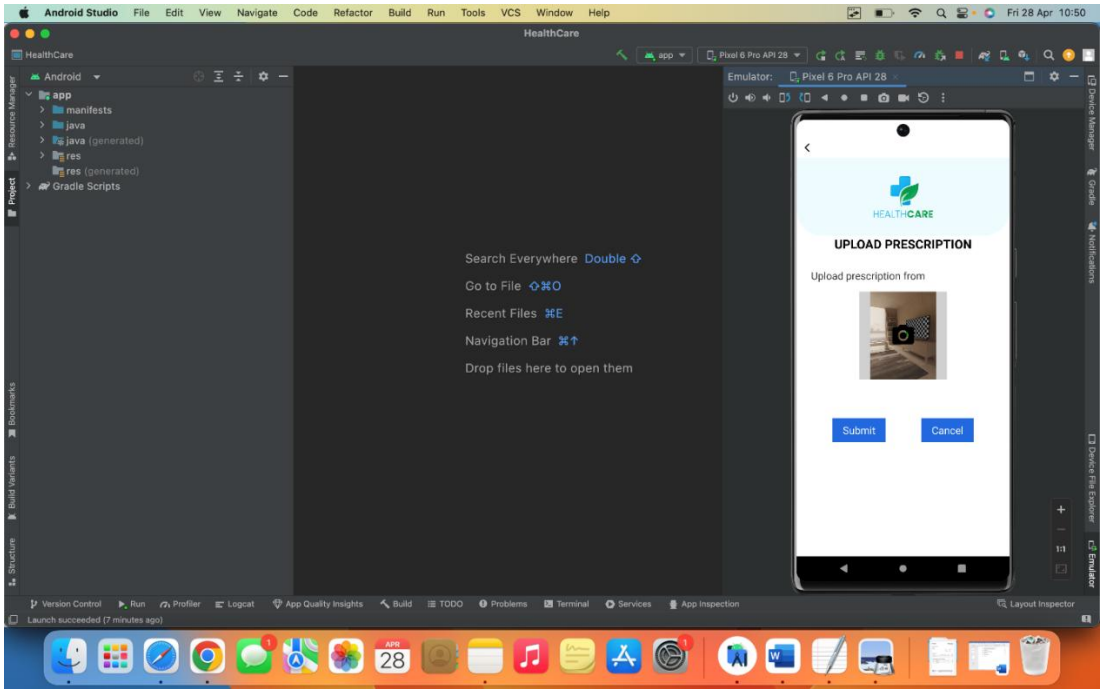


Figure 5.25 Uploaded Image of Prescription

Fig. 5.25 explains how to upload the image of prescription. First user takes the prescription picture and upload it and at last click on the submit button. Image will be uploaded in the required Section and the flow diagram confirmation page is shown in Fig. 5.26.

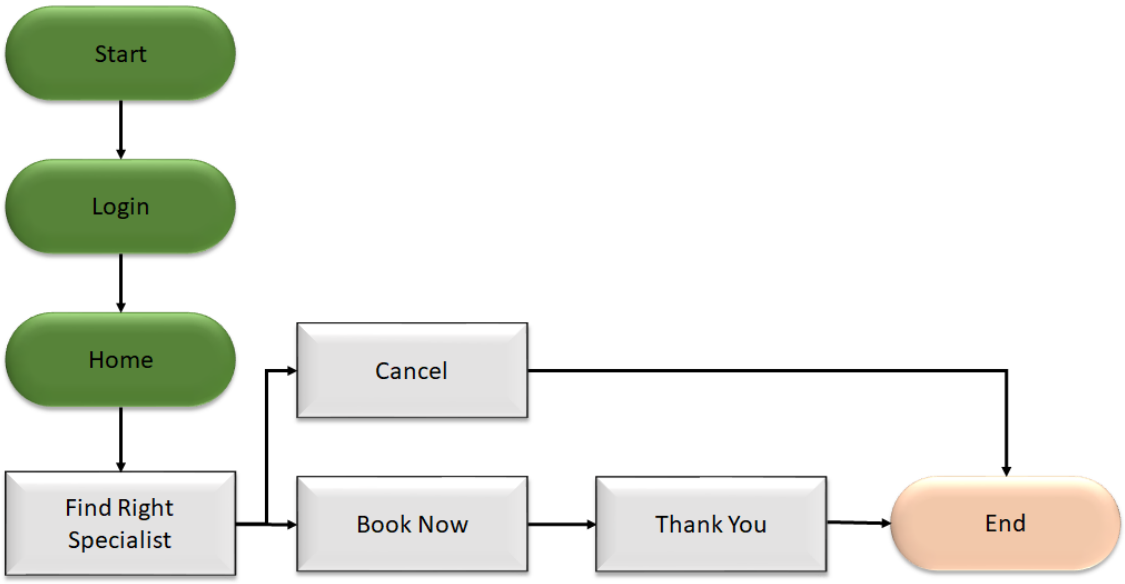


Figure 5.26 Flow process for Confirmation Page

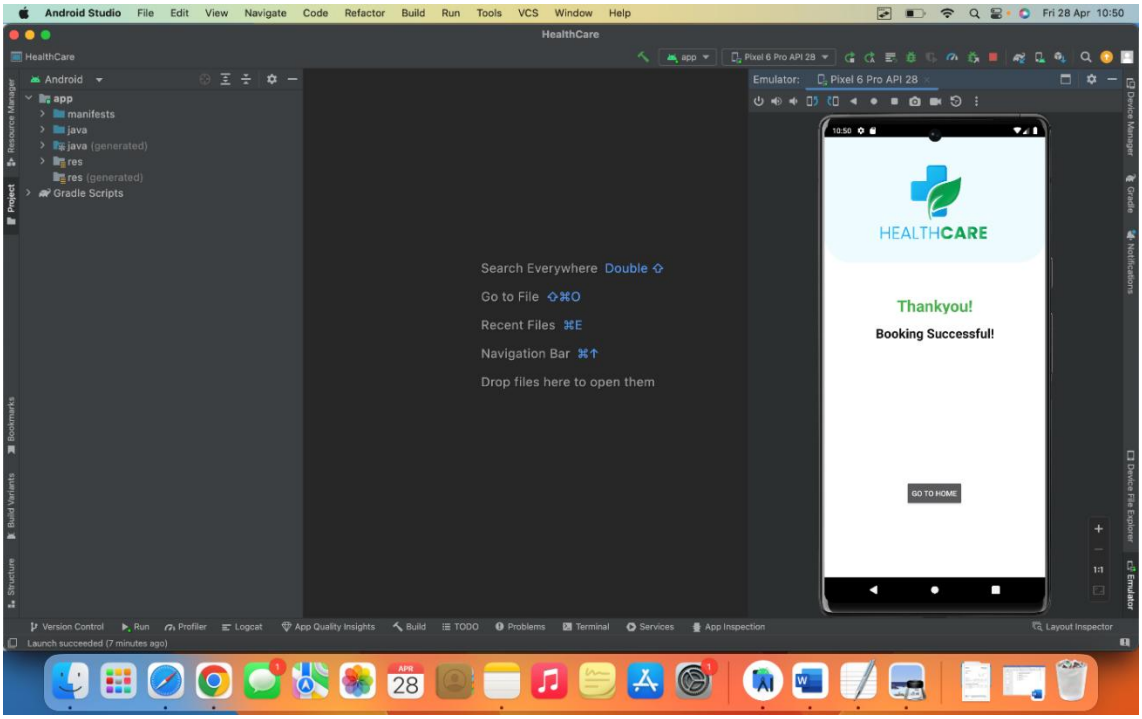


Figure 5.27 Confirmation Page of Prescription

Fig. 5.27 is having the information regarding booking of a user in healthcare. For conformation first patient need to search for a required hospital based on the injury after searching need to

check the timing and book the required slot. If the doctor is available at that slot, then user got the message like booking successful. The flow diagram of current activity for prescription module is shown in Fig. 5.28 along with screen design is shown in Fig. 5.29.

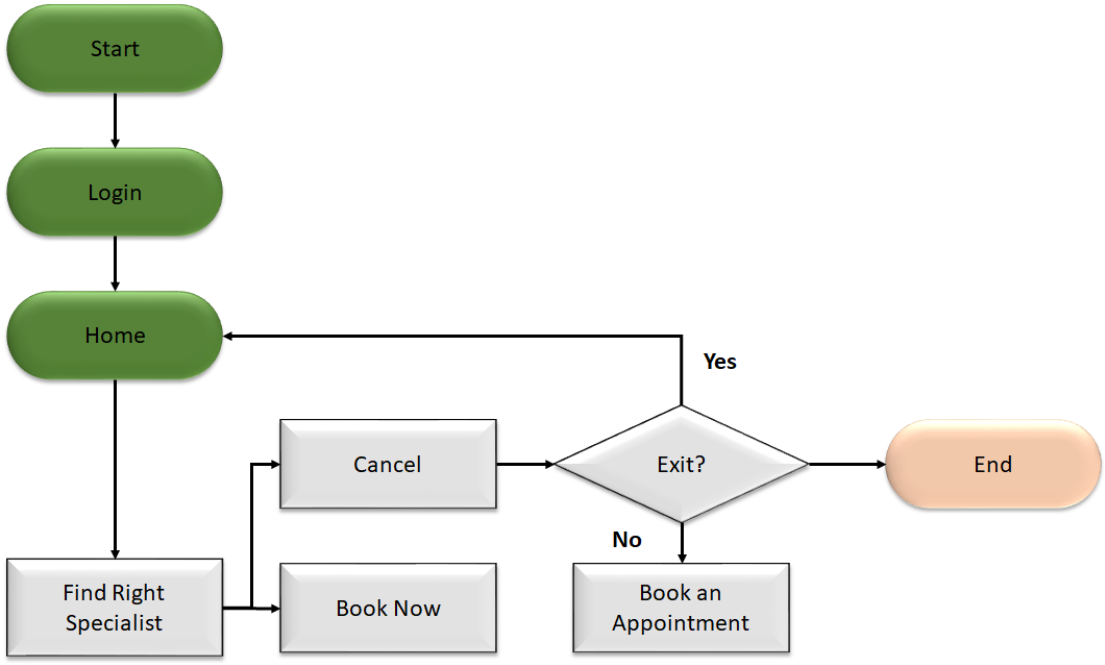


Figure 5.28 Exiting from the Current Activity of Prescription Module

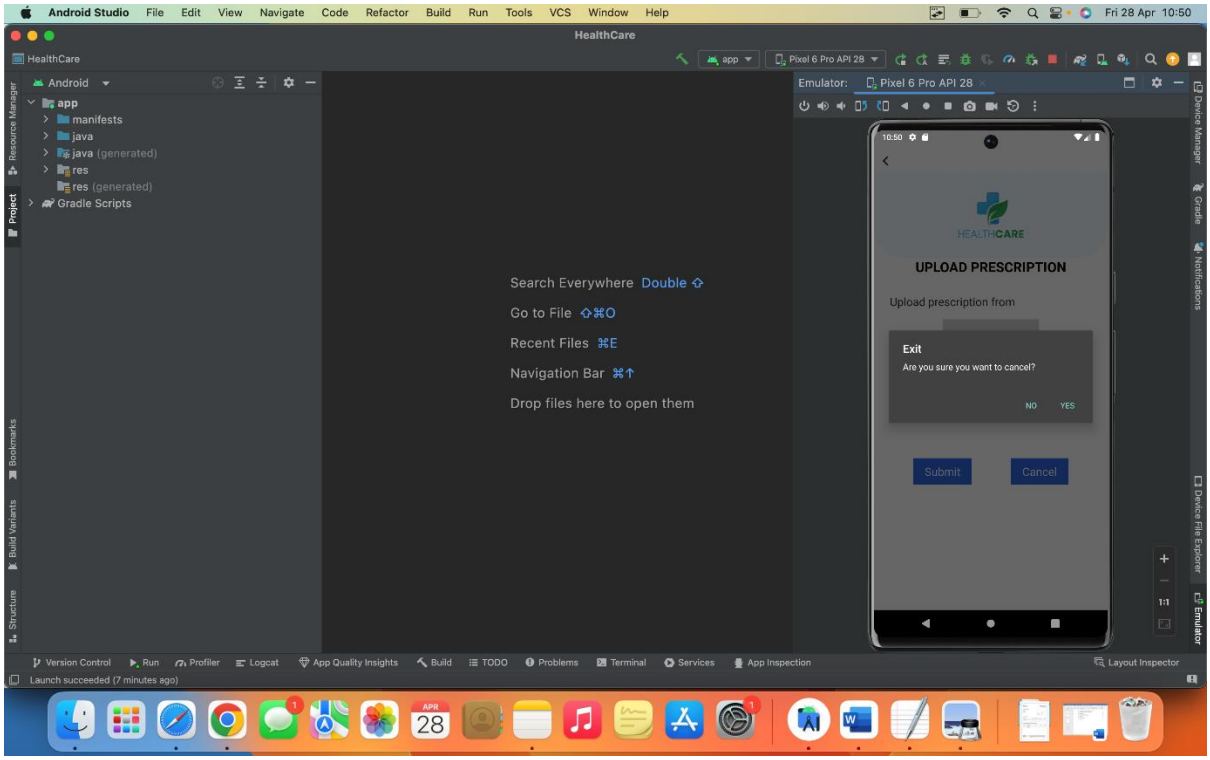


Figure 5.29 Exiting from the Current Activity of Prescription Module

Fig. 5.29 explains the patient upload the required prescription in the upload prescription box after that press the submit button once patient select the submit button automatically patient exiting from the current activity.

5.6 Feature to Get the Right Specialist

In this Section detail information how to contact a right specialist based on the injury of the patient. For finding right specialist first patient need to be observe which part is affected in the body of the patient and its effected what are the changes in the injury day wise all these can be observed by patient. Once the patient observed all these features those are very helpful to find the right Specialist for the right treatment. The advantage of identifying the right specialist is to recover the patient health quickly.

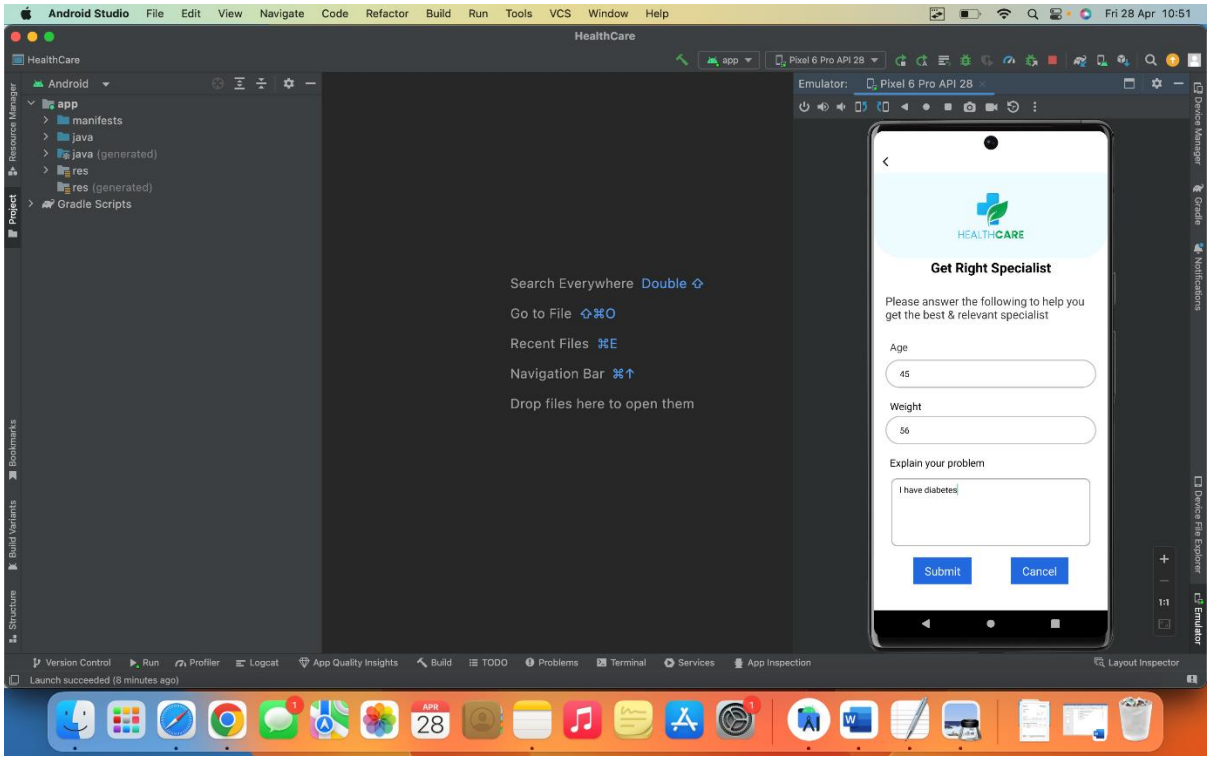


Figure 5.30 Providing the Personal Details towards Getting a Right Specialist

Fig. 5.30 provides the screen that allows to enter the personal details towards getting the right specialist. In this, patients need to observe the changes in the injury and how it effected to the body parts all these are observed clearly after that only patient find the right specialist which causes the fast recovery of the patient's health condition. The suitable doctors available for a particular search is provides as shown in Fig. 5.31 along with the Listing 5.10 that explains the code related to this act.

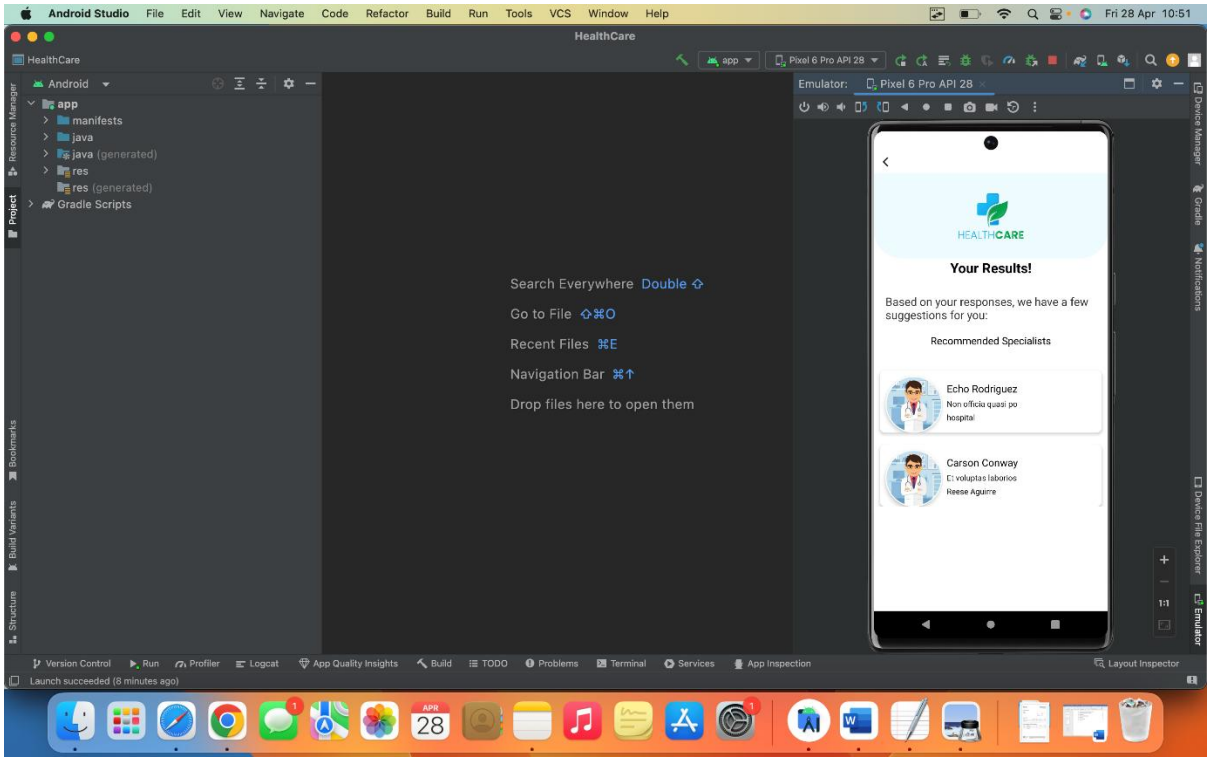


Figure 5.31 Providing the Personal Details towards Getting a Right Specialist

```
public class ResultsRightS extends BaseActivity implements IDoctorView, DoctorAdapter.ItemClickListner{


    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_results);

        fsProductPresenter = new DoctorLPresenter();
        fsProductPresenter.setView(this);

        userData = new SharedPreferencesData(getContext());

        recydoc= findViewById(R.id.recyBook);

        fsProdAdapter = new DoctorAdapter(itemList, getContext(),  listner: this);
        LinearLayoutManager gridLayoutManager = new LinearLayoutManager(getContext(), RecyclerView.VERTICAL,  reverseLayout: false);
        recydoc.setLayoutManager(gridLayoutManager);
        recydoc.setAdapter(fsProdAdapter);

        fsProductPresenter.Doctor(getContext(), userData.getACCESS_TOKEN());
    }
```

Listing 5.10 Recommended list of doctors based on the Personal Details of a Patient

Fig. 5.31 explains the recommended list of doctors based on the personal details of the patient. In this patient have to enter his details and his injury details and his personal information like how the injury happens and how it affects to the remaining parts of the body all the options need to be entered by the patient after entering all the details then the system will display the recommended list of all the doctors based on the details of the patient what he entered.

5.7 Feature-based Medical Advice on the Application

The feature-based medical advice on the application based on the features of the patient that he entered into the healthcare app will display the information regarding the advice of the injury and advice of the patient's health condition etc. The flow diagram for medical advice activity is shown in Fig. 5.32 along with the screen in Fig. 5.33.

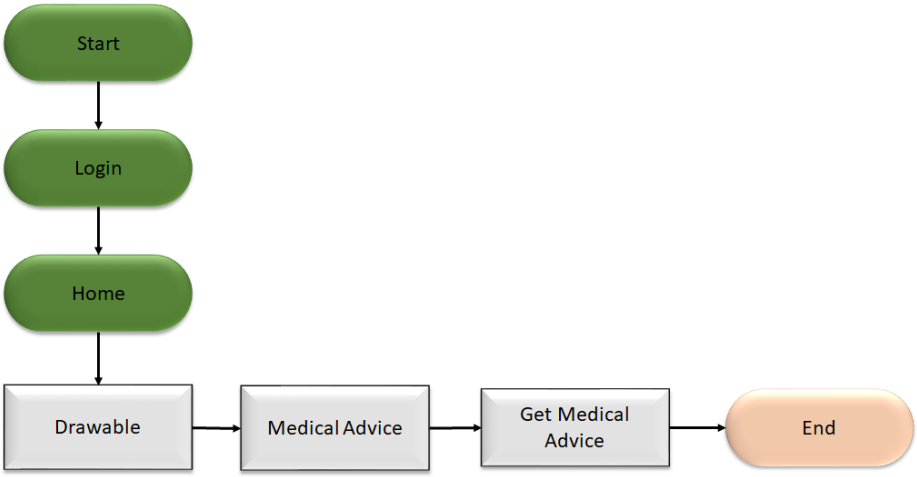


Figure 5.32 Flow Diagram for Medical Advice

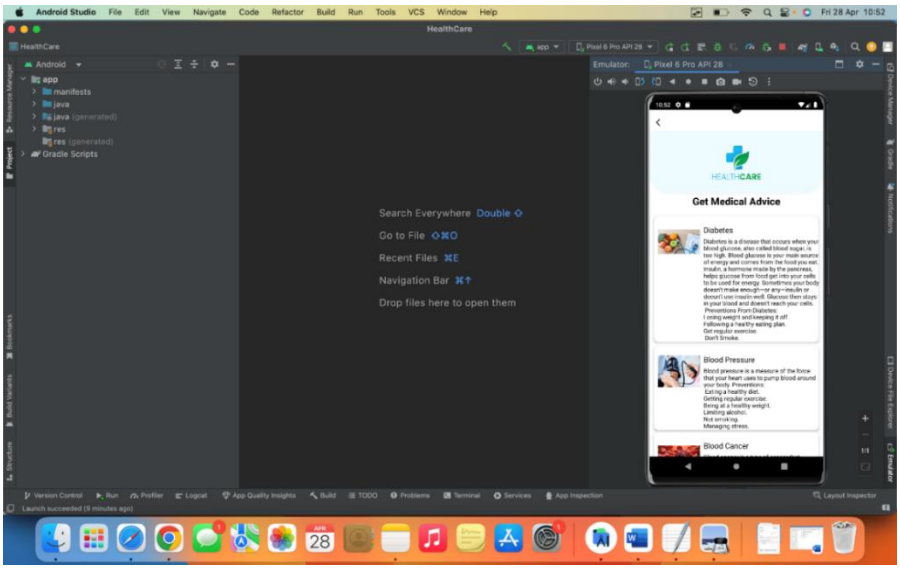


Figure 5.33 Medical Advice Page

```
3 usages
public class MedicalAdvice extends AppCompatActivity {
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_medical_advice);

    }

    public void onClick(View view){
        switch (view.getId()){
            case R.id.imgback:
                startActivity(new Intent( packageContext: this, MainActivity.class));
                break;
            default:
                break;
        }
    }
}
```

Listing 5.11 Advice or FAQs based on the Symptoms Expressed by the Patients

Fig. 5.33 explains the advice page for medicine based on the patient injury that is based on the symptoms. By using this page, patient gets some idea about his injury and this idea will be helpful to consult a required doctor for required treatment which causes the recovery of injury.

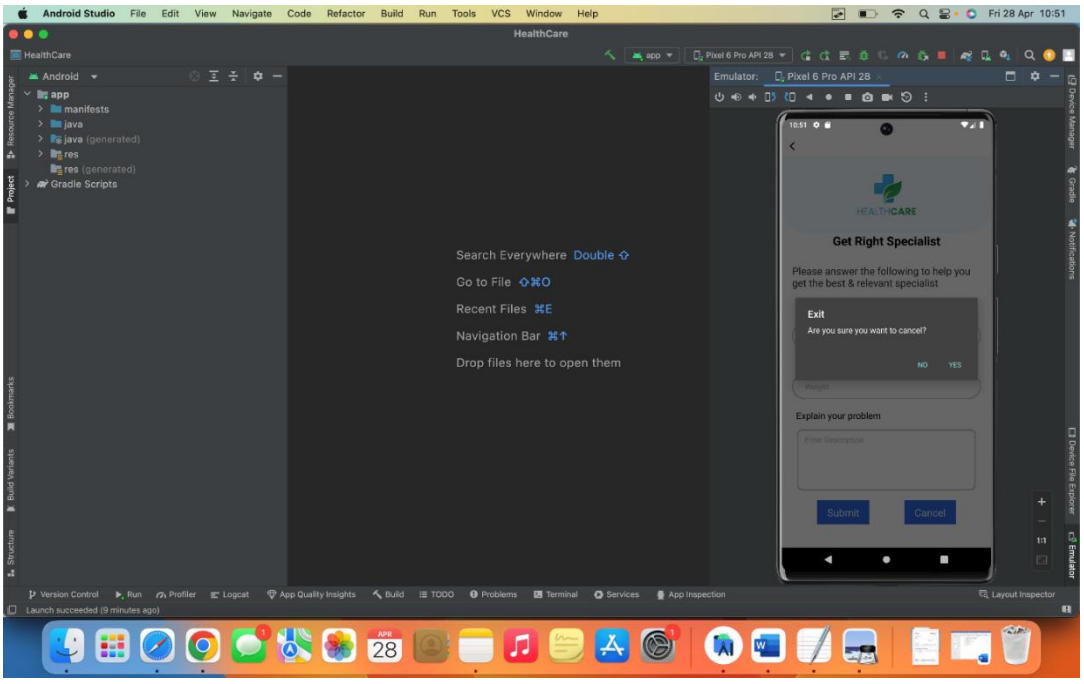


Figure 5.34 Exiting from the Current Activity of Right Specialist

Fig. 5.34 shows the exit page for patient when submits the features of injury after submitting the injury details, it automatically display the right specialist details out of the list choose the correct one once the patient selects the right person automatically exiting from the current page.

6 Conclusions and Limitations

6.1 Conclusions

Digitalization and integration of healthcare systems have become increasingly important in India, as there is a growing demand for quality healthcare and rising healthcare costs. The Indian government has recognized this and taken several measures to promote digitalization and integration in the healthcare sector. One of the key initiatives launched by the government is the National Health Stack. It is a centralized digital platform that aims to create a comprehensive healthcare system in India. The platform includes several components, such as a National Health ID, a personal health record, a health analytics platform, and a telemedicine platform. The main goal of these components is to improve the efficiency and quality of healthcare services by providing a unified digital platform for patients, healthcare providers, and government agencies.

The National Health ID is a unique digital identity assigned to every individual in India, which will be linked to their health records. This will enable patients to access their medical records from anywhere in the country, facilitating better management of their health and medical advice. The health analytics platform will provide data analysis and insights to healthcare providers and government agencies, aiding them in making informed decisions and enhancing healthcare services. The telemedicine platform is another crucial aspect of the National Health Stack that will facilitate remote consultations and medical advice. This is particularly crucial for rural areas with limited access to healthcare services. The telemedicine platform will also enable healthcare providers to collaborate and share information, thereby improving the quality of care for patients.

Taherdoost, H., 2022. What are different research approaches? Comprehensive review of qualitative, quantitative, and mixed method research, their applications, types, and limitations. *Journal of Management Science & Engineering Research*, *5*(1), pp.53-63.

TÓTH, T. and LOVÁSZOVÁ, G., 2021. VISUAL VS. TEXTUAL PROGRAMMING: A CASE STUDY ON MOBILE APPLICATION PROGRAMMING BY TEENAGERS. *Ad Alta: Journal of Interdisciplinary Research*.

Tran, A.D., Nguyen, M.Q., Phan, G.H. and Tran, M.T., 2021. Security Issues in Android Application Development and Plug-in for Android Studio to Support Secure Programming. In *Future Data and Security Engineering. Big Data, Security and Privacy, Smart City and Industry 4.0 Applications: 8th International Conference, FDSE 2021, Virtual Event, November 24–26, 2021, Proceedings 8* (pp. 105-122). Springer Singapore.

Velez, M., Jamshidi, P., Siegmund, N., Apel, S. and Kästner, C., 2022, May. On debugging the performance of configurable software systems: Developer needs and tailored tool support. In *Proceedings of the 44th International Conference on Software Engineering* (pp. 1571-1583).

Yelamarthi, K., Aman, M.S. and Abdelgawad, A., 2017. An application-driven modular IoT architecture. *Wireless Communications and Mobile Computing, 2017.*

Zhao, Y., Jiang, S., Zhong, W., Wang, L. and Li, X.F., 2023, March. NEOP: A Framework for Distributed Mobile Apps on Heterogeneous Devices. In *2023 IEEE 15th International Symposium on Autonomous Decentralized System (ISADS)* (pp. 1-8). IEEE.

Zhao, Y., Li, L., Wang, H., He, Q. and Grundy, J., 2022. APIMatchmaker: Matching the Right APIs for Supporting the Development of Android Apps. IEEE Transactions on Software Engineering.

8 Appendix

Agile as Design and Creation Methodology

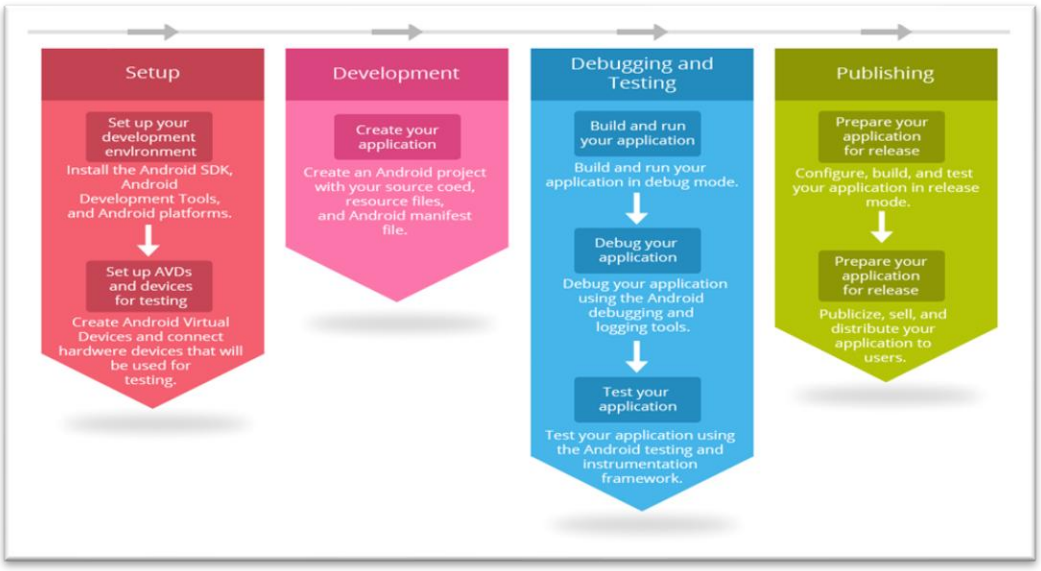
Agile methodology is one of the most popular project management approaches in mobile app development. The main principle of this methodology is having flexibility, collaboration, and iterative development.

The main process of this methodology is to increment the functionality of the app step by step by taking feedback from the clients and users. It follows an iterative approach. This will help in ensuring a rapid development and greater responsiveness to updating requirements. The whole development process also looks encouraging by following this method. Agile method makes everyone to stay on same page by making the teams work closely and communicate regularly. The problems can be fixed quickly, and the risks of delays and other issues can be mitigated to minimum. Agile mainly focus on delivering the mobile app in user friendly manner and meeting the clients requirements without missing on any important features.

Implementation of Mobile Apps using Android Studio

Android studio is Integrated Development Environment or IDE is software which is widely used in development of android mobile applications (Sulistyo et al., 2022). This IDE was released by the Google on 16th May, 2013 and since then this has been used as official IDE for android application development during the Google I/O conference. This software contains different set of tools for developing and testing the software (Syaifudin et al., 2021). Like other IDE, the android studio also contains some of the common feature like code editor, compiler/interpreter and debugger that helps the developers in creating and developing the application with little difficulty. The code editor is the place for not only writing but also the user can edit and debug the program codes (Tran et al., 2021). Apart from these features in the code editor, the other features include highlighting the syntax, helping in completing the code. Highlighting the syntax feature highlights the different keywords of a programming language using different colours like green, red etc and it greatly help to read the code easily. Code completion helps the developer by providing the code suggestions which helps to complete the code in short duration of time. Debugging is where the codes are debugged for errors. It helps in tracking and identifying the bugs or generally the errors in the program code (Velez et al., 2022). In the process of debugging the programmer usually sets the breakpoints to execute the code till that point and verify the results and also inspects or prints the variables to review the

outputs thereby helping the developers to track the bugs and errors. Therefore, debugging plays a very important role in the software development and often consumes a lot of time because this helps in developing the clean codes.


Setup
Set up your development environment
Install the Android SDK, Android Development Tools, and Android platforms.
Set up AVDs and devices for testing
Create Android Virtual Devices and connect hardwere devices that will be used for testing.
Development
Create your application
Create an Android project with your source coed, resource files, and Android manifest file.
Debugging and Testing
Build and run your application
Build and run your application in debug mode.
Debug your application
Debug your application using the Android debugging and logging tools.
Test your application
Test your application using the Android testing and instrumentation framework.
Publishing
Prepare your application for release
Configure, build, and test your application in release mode.
Prepare your application for release
Publicize, sell, and distribute your application to users.

Once the functionality is added, the mobile app must be tested to ensure it works as expected. This can be done using the built-in emulator provided by the IDE or by connecting a physical device to the development computer. Finally, when the mobile app is fully tested and functional, it can be deployed on the Google Play Store or other app marketplaces. This involves packaging the app into an APK file and submitting it to the app store for review and approval. Once approved, the mobile app will be available to users to download and use (Cheon., 2019).

Importance of APIs in Mobile App Development using Android Studio

API's sets of routines, protocols, and tools that allow different software applications to communicate with each other. Jung *et al.,* (2021) said in today's world, mobile app development has become an essential part of business strategy. Zhao et al., (2023) said that mobile apps provide a way for businesses to reach a wider audience, increase customer engagement, and improve brand loyalty. However, creating a mobile app requires a lot of time, effort, and resources. This is where APIs (Application Programming Interfaces) come into play.

Shama (2023) explained that API is very important in today's world while dealing most of the complicated mobile apps and some of the important points author mentioned are listed below:

- Integration of third-party services: Usually coders are doing very hard coding to include the important features in their mobile app, meanwhile API's wipe out those coding tasks (Luo et al., 2021); and offers that only functionalities effortlessly and with the help of

this third-party services (Sharma, 2023). For example, API's can be used to integrate payment gateways, social media platforms, or mapping services into a mobile app. This provides users with more options and convenience, and also saves time and resources for the developers.

- Access to device functionalities: API's provide developers with access to various device functionalities such as the camera, microphone, and GPS (Zhao et al., 2023); and Sharma (2023) strongly opined that the developers create more engaging and interactive mobile apps with all these features. For example, a camera API can be used to create a photo-sharing app that allows users to take pictures and share them with their friends.
- Improved app performance: API's can improve app performance by enabling developers to offload some computing tasks to remote servers, reducing the processing load on the mobile device (Sharma, 2023). For example, a weather app can use an API to fetch weather data from a remote server instead of processing the data locally on the device.
- Better user experience: API's can be used to create a better user experience by providing real-time updates, personalized content, and interactive features (Sharma, 2023). For example, a news app can use an API to provide personalized news articles based on the user's interests.
- Rapid app development: API's can help developers build mobile apps quickly by providing pre-built components and functionalities (Sharma, 2023). This saves development time and resources, allowing developers to focus on other aspects of the app. For example, a chat app can use a messaging API to quickly add messaging functionality to the app.