DIGITALISED MEDICAL HEALTH RECORD MAINTENANCE APPLICATION

A MINI PROJECT REPORT

Submitted by

NALLA MANU NAGENDRA [191511021]

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

SAVEETHA SCHOOL OF ENGINEERING

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES : CHENNAI 602 105

JUNE 2018

SAVEETHA SCHOOL OF ENGINEERING

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES: CHENNAI 602 105

BONAFIDE CERTIFICATE

Certified that this project report **"Digitalised Medical Health Record Maintenance Application"** is the bonafide work of "**Nalla Manu Nagendra [191511021]"** who carried out the project work under my supervision.

SIGNATURE

PROJECT COORDINATOR

Mrs.V.Banupriya

Assistant professor

Department of CSE

Saveetha School Of Engineering

Chennai 602105

SIGNATURE

HEAD OF THE DEPARTMENT

Dr.S. Anusuya

Professor

Department of CSE

Saveetha School Of Engineering

Chennai 602105

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We here by declare that the mini project entitled "**DIGITALISED MEDICAL HEALTH RECORD MAINTENANCE APPLICATION**"in partial fulfillment of requirement for the award of the degree of Bachelor of Engineering is a record of original project work done by us, during our period of study in **SAVEETHA SCHOOL OF ENGINEERING** and no part of it has been submitted for any degree,association,fellowship or any other similar titles.

SIGNATURE

Name:

Date:

CERTIFICATE BY THE GUIDE

This to certify that the project entitled "**DIGITALISED MEDICAL HEALTH RECORD MAINTENANCE APPLICATION**" is the bonafied work carried out by "**NALLA MANU NAGENDRA**" student of B.E., **SAVEETHA SCHOOL OF ENGINEERING, SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES, CHENNAI 602105,** during the year 2018-2019 under my guidance.

SIGNATURE

Dr.S.Anusuya

Professor

Department of Computer Science and Engineering

Saveetha School Of Engineering

Saveetha Institute of Medical and Technical Sciences

Chennai 602105

Date:

ACKNOWLEDGEMENT

To begin with we thank the Almighty for giving us the strength and knowledge to carry out this project work. At this outset, we take immense pleasure in expressing our sincere gratitude to our honorable **CHANCELLOR, Dr. N.M.VEERAIYAN** for being a source of inspiration.

We sincerely thank our **VICE-CHANCELLOR, Dr.L JAWAHAR NESAN** for providing all the facilities to complete the project successfully.

We extent our heartiest thanks to our **DIRECTOR, Mrs. RAMYA DEEPAK** for providing constant support and encouragement.

We express our gratitude to our beloved **PRINCIPAL Dr. D DHANASEKARAN** for granting permission to undertake the project. We are also thankful to him for his valuable guidance all the way through the course.

We extend our devoted gratitude to our beloved **HEAD OF DEPARTMENT Dr.S.ANUSUYA** Department of Computer Science and Engineering & Information Technology for his consistent encouragement.

We are deeply indebted to our guide and PROJECT **CO-ORDINATOR**, **Mrs.V.BANUPRIYA** for his/her motivation, guidance and valuable suggestions to carry out our project and her consistent encouragement throughout the course of our project work.

We extend our hearty thanks to our PROJECT GUIDE **Dr.S.ANUSUYA** for his/her unstinted support and guidance besides cooperation during the development of this project.

We profusely thank all the faculty members of Department of Computer Science & Engineering, who have been a source of great help in multiple ways in our venture.

Finally, we owe our special thanks to our parents and friends for their moral support and care extended during difficult times.

TABLE OF CONTENTS

ABSTRACT

Despite many innovations in information technology, there is no application to store your medical records People often tend to lose or miss their medical records. Sometimes, they forget to take those records with them when the visit a hospital. To solve all these problems, in this project an application is developed to store available health records of patients such as Xrays, MRI, CT scans etc. securely at one place. This mobile application helps to access patient's medical details from anywhere in the world with just having an internet connection. The user have to first register himself/herself using username, password and email id. User will have permissions to upload and view the records according to the type of the medical record

CHAPTER 1

Introduction:

The Era of mobile technology opens the windows to the Android app. The websites are vanishing and the mobile phones are emerging. Its the time to change from conventional websites to Apps, which has become the part of our daily routine. We are introducing "M-Health.apk" android application software . This application is useful to patient as well as the doctors. This Application will allow the users to maintain their medical records. The user can upload medical records like MRI, ECG,X-RAY etc and store it in this application. They can able to access the uploaded files at anytime ,anywhere This application can also be helpful in some critical or some emergency situations. At that time the user or doctor can able to access their patient details.

1.1 Problem statement:

Due to the growth in technology In this generation, people are making themselves busy in technology 24/7.They are converting each and every seconds of their life into technology. The increase in population of our country its very difficult to lead our normal life style and it is very difficult to remember every task at our hand. At this busy society the rate of patients also increase. In each and every hospital the management has to create the patient profile and they have to maintain their medical records until they discharged from the hospital. After that the job of maintaining the records comes at patients hand. For Every individual, it is not possible to carry each and every report while going to hospital. If we consider a patient with lot of diseases ,it is difficult for him to carry and maintain everything ,when he visits a doctor. Some patient may lose their medical record by mistake . In some cases ,the patients will get confuse with their records.

1.2 Scope of the Project:

- This application allows user to store their medical records.
- This application will let the users to access their stored medical records.
- This application will provide security to the stored medical records by giving right access to right person by providing secure login id and login password.
- It helps in emergency situation.
- The user details are encrypted using SHA-1 Algorithm.

CHAPTER 2

2.SYSTEM ANALYSIS :

2.1 Existing system:

Currently there is no application in the internet to store the medical Records of a patient.

2.2 Literature Review:

Mobile technologies have become ubiquitous among health care consumers and professionals. Simultaneously, the health care industry is moving towards a delivery model that is patient-centered and value-based. Mobile technologies may be able to facilitate the shift in the relationship between clinicians, payers, life sciences companies and patients that will be required to implement this new model.

Report highlights include:

- The health care and life sciences sector is recognized as one of the top three fields (along with consumer products and the financial services industry) likely to experience new mobile business model growth in the next five years, according to the Deloitte Open Mobile Survey 2012.
- After a slow start, the capabilities offered by mobile technologies are fast being appreciated by the health care industry with a raft of devices, sensors, apps and other programs being developed that target chronic conditions, telemedicine and remote monitoring, patient data capture, electronic records, e-prescribing and the parallel industries of fitness and wellness.
- mHealth holds promise for improvement along the value chain – maximizing professionals' time and productivity, improving quality, increasing access and equity, stimulating greater consumer involvement, and potentially decreasing costs, or perhaps equally as importantly, getting better value for the money invested.
- The extent to which stakeholders create user confidence through adequate privacy and security protections will play a key role in accelerating or retarding the adoption of mHealth and the realization of benefits.

2.2.1 Proposed system:

This application is useful to patient as well as the doctors. This Application will allow the users to maintain their medical records. The user can upload medical records like MRI,ECG,X-RAY etc and store it in this application. They can able to access the uploaded files at anytime ,anywhere This application can also be helpful in some critical or some emergency situations. At that time the user or doctor can able to access their patient details. The doctors will ask the past medical details if a patient is admitted in serious condition. At that time the patients relative can able show the doctors the stored medical records in this application. Then the doctor can able to treat the patient according to their medical records from their past. So this application may able to save someone's life.

Advantages:

1. It helps the patients to store the medical record digitally
2. It reduces the stress of carrying the records for each and every visit to doctor
3. It is highly secure, the records are maintained for a lifetime.

2.3. SOFTWARE DESCRIPTION:

ABOUT ANDROID STUDIO

Android Studio is an integrated development environment (IDE) from Google that provides developers with tools needed to build applications for the Android OS platform. Android Studio is available for download on Windows, Mac and Linux. A one-time, $25 developer's license is required to publish apps to Google Play App Store. The foundation for Android Studio is based on IntelliJ IDEA.

The Android Studio IDE is free to download and use. It has a rich UI development environment with templates to give new developers a launching pad into Android development. Developers will find that Studio gives them the tools to build phone and tablet solutions as well as emerging technology solutions for Android TV, Android Wear, Android Auto, Glass and additional contextual models.

Android Studio is intended to be used by development teams as small as one person or as large as global teams. The Android Studio IDE can be linked to larger teams with GIT or similar version control services for larger teams. Mature Android developers will find tools that are necessary for large teams to deliver solutions rapidly to their customers. Android solutions can be developed using either Java or C++ in Android Studio. The workflow for Android Studio is built around the concept of continuous integration. Continuous Integration allows for teams to test their code each and every time a developer checks in their work. Issues can be captured and reported to the team immediately. The concept of continuously checking code provides actionable feedback to the developers with the goal of releasing versions of a mobile solution faster to the Google Play App Store. To this end, there is rigorous support for LINT tools, Pro-Guard and App Signing tools.

Performance tools provide access to view how well an Android application package file (APK) is going. The performance and profiling tools display a color-coded image to show how often the same pixel is drawn on a screen to reduce rendering overhead. The GPU rendering shows how well your app does in maintaining Google's 16-ms-per-frame benchmark. Memory tools visualize where and when your app will use too much system RAM and when Garbage collection occurs, Battery Analysis tools present how much drain you're placing on a device.

Android Studio supports Google App Engine for quick cloud integration of new APIs and features. You will find support for many APIs directly in Android Studio such as Google Play, Android Pay and Health.

There is support for all platforms of Android starting with Android 1.6 and later. There are variants of Android that are significantly different to the Google Android version. The most popular is Amazon's Fire OS. Android Studio can be used to build Amazon Fire OS APKs using these guidelines. Android Studio is replacing Google's support for Eclipse ADT.

ABOUT JAVA

Java is a programming language originally developed by Sun Microsystems and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code that can run on any Java virtual machine (JVM) regardless of computer architecture.

One characteristic of Java is portability, which means that computer programs written in the Java language must run similarly on any supported hardware/operating-system platform. One should be able to write a program once, compile it once, and run it anywhere.

This is achieved by compiling the Java language code, not to machine code but to Java byte code – instructions analogous to machine code but intended to be interpreted by a virtual machine (VM) written specifically for the host hardware. End-users commonly use a JRE installed on their own machine, or in a Web browser.

Standardized libraries provide a generic way to access host specific features such as graphics, threading and networking. In some JVM versions, byte code can be compiled to native code, either before or during program execution, resulting in faster execution.

A major benefit of using byte code is porting. However, the overhead of interpretation means that interpreted programs almost always run more slowly than programs compiled to native executables would, and Java suffered a reputation for poor performance. This gap has been

narrowed by a number of optimization techniques introduced in the more recent JVM implementations.

Another technique, commonly known as *static compilation*, or ahead-of-time (AOT) compilation, is to compile directly into native code like a more traditional compiler. Static Java compilers translate the Java source or byte code to native object code. This achieves good performance compared to interpretation, at the expense of portability; the output of these compilers can only be run on a single architecture. AOT could give Java something like performance, yet it is still not portable since there are no compiler directives, and all the pointers are indirect with no way to micro manage garbage collection.

Java's performance has improved substantially since the early versions, and performance of JIT compilers relative to native compilers has in some tests been shown to be quite similar. The performance of the compilers does not necessarily indicate the performance of the compiled code; only careful testing can reveal the true performance issues in any system.

One of the unique advantages of the concept of a runtime engine is that errors (exceptions) should not 'crash' the system. Moreover, in runtime engine environments such as Java there exist tools that attach to the runtime engine and every time that an exception of interest occurs they record debugging information that existed in memory at the time the exception was thrown (stack and heap values). These Automated Exception Handling tools provide 'root-cause' information for exceptions in Java programs that run in production, testing or development environments.

CHAPTER 3

3. SYSTEM REQUIREMENTS:

3.1 SOFTWARE REQUIREMENTS(Minimum):

Front End	:	Java
Environment	:	Android Studio
Back End	:	My Sql
Operating System	:	Windows 7
Operating system(Mobile)	:	Android 4.0

3.1.1HARDWARE REQUIREMENTS(Minimum):

Processor	:	Core i3
RAM	:	4GB
Hard Disk	:	20 GB

CHAPTER 4

4. SYSTEM DESIGN

4.1 Architecture diagram

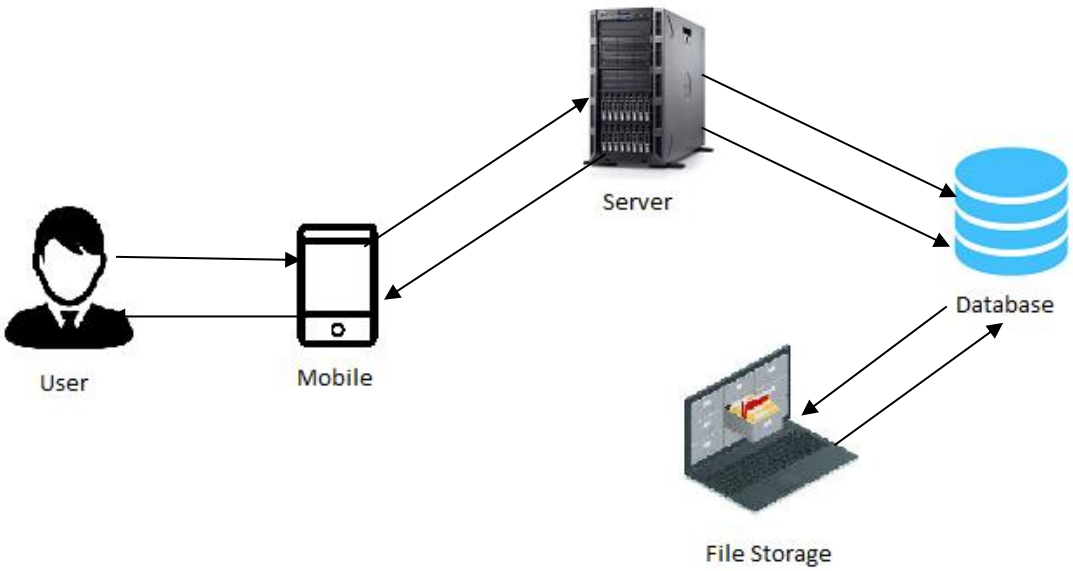


Fig.4.1 Architecture Diagram

4.2 Data Flow Diagram:

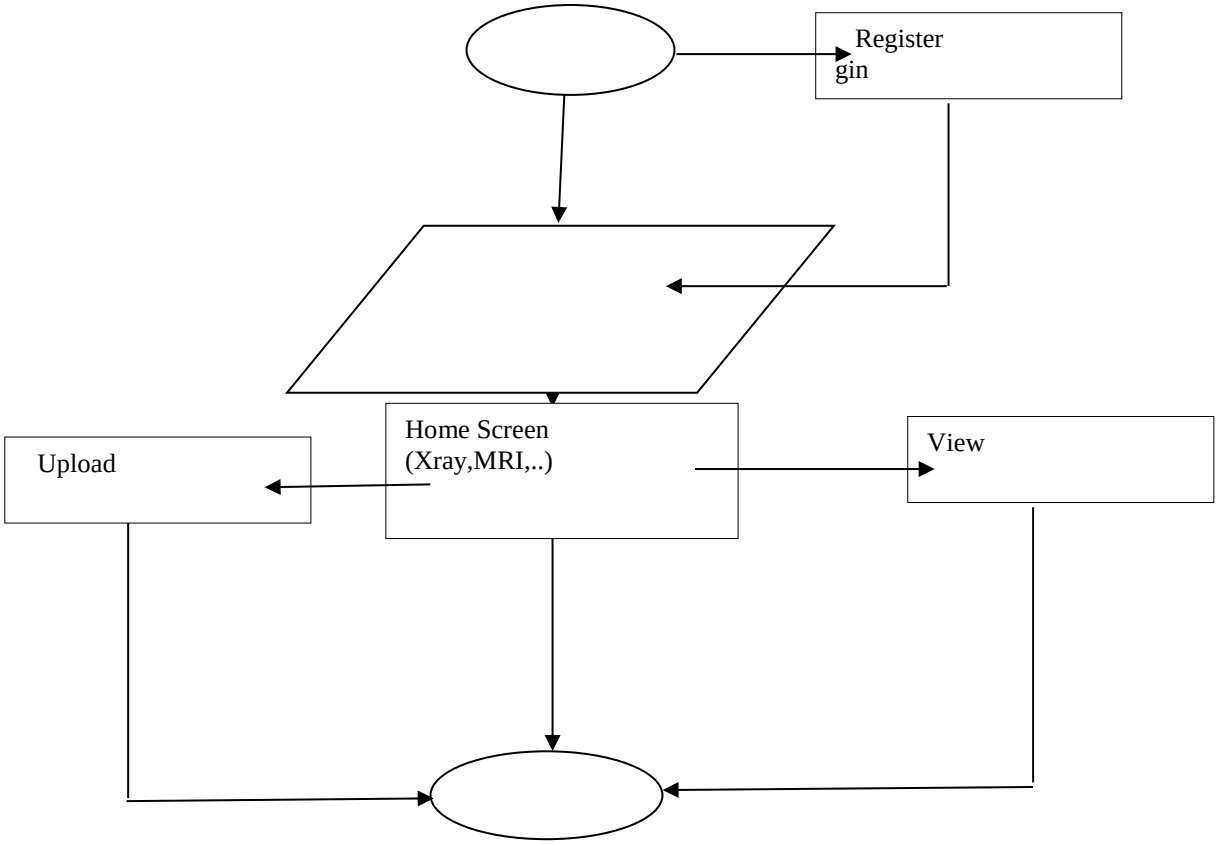


Fig.4.2 Data Flow Diagram

4.3 UML Diagrams:

4.3.1 Use Case Diagram:

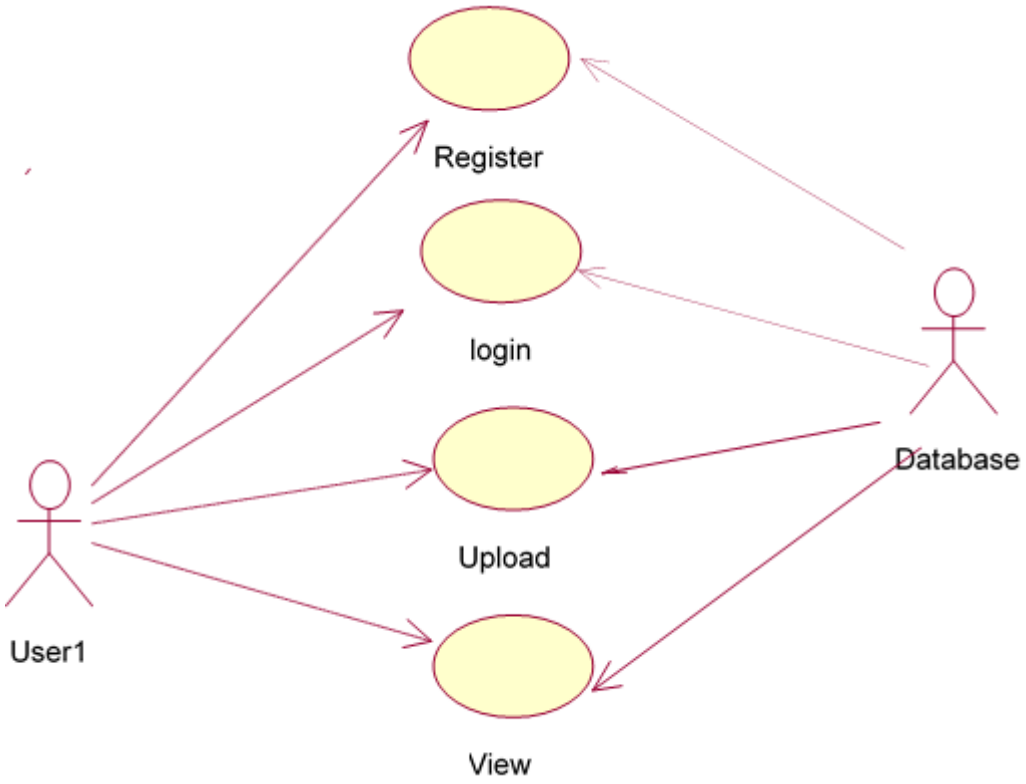


Fig 4.3.1 Use Case Diagram

4.3.2 Activity Diagram:

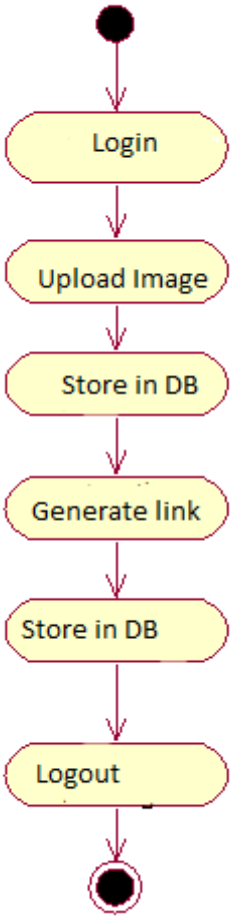


Fig 4.3.2 Activity Diagram

CHAPTER 5

SYSTEM IMPLEMENTATION

ANDROID STUDIO:

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on windows , macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. The current stable version is 3.1 released in March 2018.

Features

The following features are provided in the current stable version:

- Gradle-based build support
- Android-specific refactoring and quick fixes
- Lint tools to catch performance, usability, version compatibility and other problems
- ProGuard integration and app-signing capabilities
- Template-based wizards to create common Android designs and components
- A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations
- Support for building Android Wear apps
- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine
- Android Virtual Device (Emulator) to run and debug apps in the Android studio.

Android Studio supports all the same programming languages of IntelliJ, and PyCharm e.g. Python, and Kotlin; and Android Studio 3.0 supports "Java 7 language features and a subset of Java 8 language features that vary by platform version." External projects backport some Java 9 features.

Creating an Android project

First, be sure you have installed the latest version of Android Studio. .

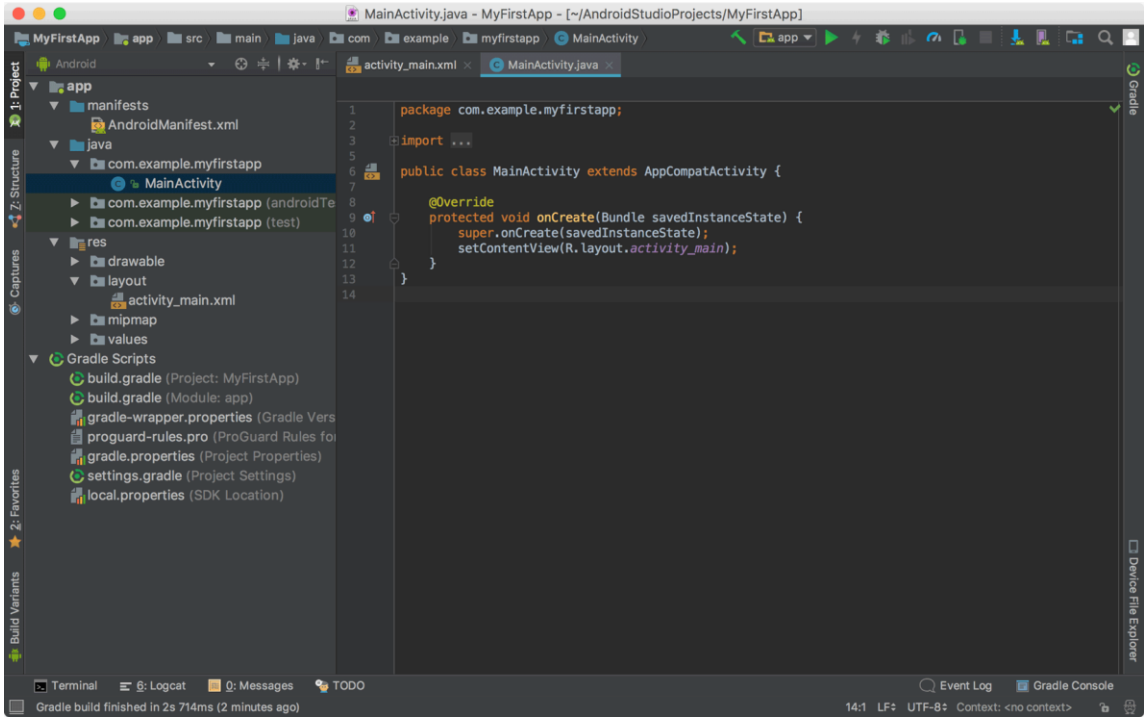
1. In the **Welcome to Android Studio** window, click **Start a new Android Studio project**.

   Or if you have a project opened, select **File > New Project**.

2. In the **Create New Project** window, enter the following values:
   - **Application Name**: "My First App"
   - **Company Domain**: "example.com"

     You might want to change the project location. Also, if you want to write a Kotlin app, check the **Include Kotlin support** checkbox. Leave the other options as they are.
3. Click **Next**.
4. In the **Target Android Devices** screen, keep the default values and click **Next**.
5. In the **Add an Activity to Mobile** screen, select **Empty Activity** and click **Next**.
6. In the **Configure Activity** screen, keep the default values and click **Finish**.

   After some processing, Android Studio opens the IDE.



Now take a moment to review the most important files.

First, be sure the **Project** window is open (select **View > Tool Windows > Project**) and the **Android** view is selected from the drop-down list at the top of that window. You can then see the following files:

app > java > com.example.myfirstapp > MainActivity

This is the main activity (the entry point for your app). When you build and run the app, the system launches an instance of this Activity and loads its layout.

app > res > layout > activity_main.xml

This XML file defines the layout for the activity's UI. It contains a TextView element with the text "Hello world!".

app > manifests > AndroidManifest.xml

The manifest file describes the fundamental characteristics of the app and defines each of its components.

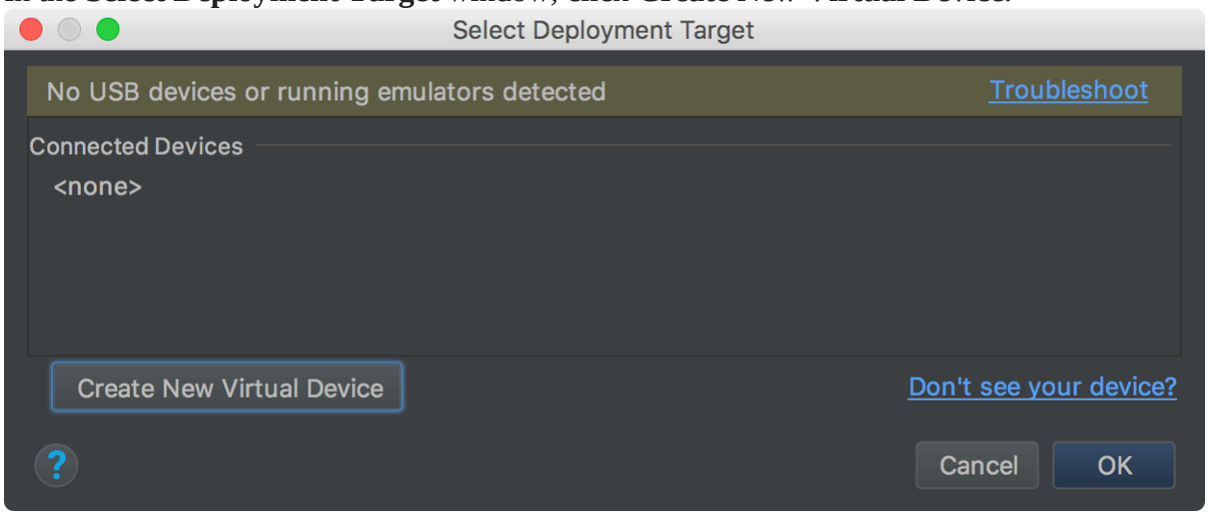
Gradle Scripts > build.gradle

You'll see two files with this name: one for the project and one for the "app" module. Each module has its own build.gradle file, but this project currently has just one module. You'll mostly work with the module's build.gradle file to configure how the Gradle tools compile and build your app.

Running the created Application on an Emulator

Run the app on an emulator as follows:

1. In Android Studio, click the **app** module in the **Project** window and then select **Run** > **Run** (or click **Run**  in the toolbar).
2. In the **Select Deployment Target** window, click **Create New Virtual Device**.



3. In the **Select Hardware** screen, select a phone device, such as Pixel, and then click **Next**.
4. In the **System Image** screen, select the version with the highest API level. If you don't have that version installed, a **Download** link is shown, so click that and complete the download.
5. Click **Next**.
6. On the **Android Virtual Device (AVD)** screen, leave all the settings alone and click **Finish**.
7. Back in the **Select Deployment Target** dialog, select the device you just created and click **OK**.

Android Studio installs the app on the emulator and starts it. You should now see "Hello World!" displayed in the app running on the emulator.

Congratulations! You created your first Android project, a package, a Java class and you ran this app in an Emulator

FEATURES OF MySQL Database:

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons:

- MySQL is released under an open-source license. So you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).

CONNECTING TO AND DISCONNECTING FROM THE SERVER:

To connect to the server, you will usually need to provide a MySQL user name when you invoke mysql and, most likely, a password. If the server runs on a machine other than the one where you log in, you will also need to specify a host name. Contact your administrator to find out what connection parameters you should use to connect (that is host, user name, and password to use). Once you know the proper parameters, you should be able to connect like this:

shell> mysql -h host -u user -p

Enter password: ********

QUERY:

A query is a question that has to be asked the data. Access gathers data that answers the question from one or more table. The data that make up the answer is either dynaset (if you edit it) or a snapshot(it cannot be edited).Each time we run query, we get latest information in the dynaset. Access either displays the dynaset or snapshot for us to view or perform an action on it ,such as deleting or updating.

Sample Coding:

Main Activity :

```
package com.example.manun.loginregdb;

import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import org.apache.http.message.BasicNameValuePair;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {


    EditText editEmail, editPassword, editName;
    Button btnSignIn, btnRegister;

    String URL= "https://mhealthmanu.000webhostapp.com/index1.php";
    public static String usn;
    String act="manuuuuuu";

    JSONParser jsonParser=new JSONParser();

    int i=0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        editEmail=(EditText)findViewById(R.id.editEmail);
        editName=(EditText)findViewById(R.id.editName);
        //Toast.makeText(this, editName, Toast.LENGTH_SHORT).show();
        editPassword=(EditText)findViewById(R.id.editPassword);

        btnSignIn=(Button)findViewById(R.id.btnSignIn);
        btnRegister=(Button)findViewById(R.id.btnRegister);

        btnSignIn.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
        public void onClick(View view) {


            AttemptLogin attemptLogin= new AttemptLogin();

attemptLogin.execute(editName.getText().toString(),editPassword.getText().toString(),"");
        }
    });

    btnRegister.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

            if(i==0)
            {
                i=1;
                editEmail.setVisibility(View.VISIBLE);
                btnSignIn.setVisibility(View.GONE);
                btnRegister.setText("CREATE ACCOUNT");
            }
            else{

                btnRegister.setText("REGISTER");
                editEmail.setVisibility(View.GONE);
                btnSignIn.setVisibility(View.VISIBLE);
                i=0;

                AttemptLogin attemptLogin= new AttemptLogin();

attemptLogin.execute(editName.getText().toString(),editPassword.getText().toString(),editE
mail.getText().toString());

            }

        }
    });


}

private class AttemptLogin extends AsyncTask<String, Void, JSONObject> {

    @Override
    protected void onPreExecute() {

        super.onPreExecute();

    }

    @Override
    protected JSONObject doInBackground(String... args) {
```

```
        String email = args[2];
        String password = args[1];
        String name= args[0];

        ArrayList params = new ArrayList();
        params.add(new BasicNameValuePair("username", name));
        params.add(new BasicNameValuePair("password", password));
        if(email.length()>0)
           params.add(new BasicNameValuePair("email",email));

        JSONObject json = jsonParser.makeHttpRequest(URL, "POST", params);


        return json;

    }

    protected void onPostExecute(JSONObject result) {


         String a= result.toString();


        try {
           if (result != null) {

Toast.makeText(getApplicationContext(),result.getString("message"),Toast.LENGTH_SHOR
T).show();

                if(result.getString("message").equals("Successfully logged in"))
                {
                   usn=editName.getText().toString();
                  startActivity(new Intent(MainActivity.this,listview.class));

                }



           } else {

                Toast.makeText(getApplicationContext(), "Unable to retrieve any data from
server", Toast.LENGTH_LONG).show();
           }
        } catch (JSONException e) {
           e.printStackTrace();
        }

    }
```

```
    }
}
```

IMAGE UPLOAD:

```
package com.example.manun.loginregdb;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.AsyncTask;
import android.provider.MediaStore;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Base64;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;

import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.ResponseHandler;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.BasicResponseHandler;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;

import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class imageupload extends AppCompatActivity implements View.OnClickListener {

    public static final String UPLOAD_URL =
"https://mhealthmanu.000webhostapp.com/upload.php";
    public static final String UPLOAD_KEY = "image";
    public static final String x="mn";
```

```
    private int PICK_IMAGE_REQUEST = 1;

    private Button buttonChoose;
    private Button buttonUpload;


    private ImageView imageView;

    private Bitmap bitmap;

    private Uri filePath;
       String name;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        Intent x = getIntent();

        name = x.getStringExtra("name");

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_imageupload);

        buttonChoose = (Button) findViewById(R.id.buttonChoose);
        buttonUpload = (Button) findViewById(R.id.buttonUpload);


        imageView = (ImageView) findViewById(R.id.imageView);

        buttonChoose.setOnClickListener(this);
        buttonUpload.setOnClickListener(this);




    }

    private void showFileChooser() {
        Intent intent = new Intent();
        intent.setType("image/*");
        intent.setAction(Intent.ACTION_GET_CONTENT);
        startActivityForResult(Intent.createChooser(intent, "Select Picture"),
PICK_IMAGE_REQUEST);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if (requestCode == PICK_IMAGE_REQUEST && resultCode == RESULT_OK &&
```

```
data != null && data.getData() != null) {

            filePath = data.getData();
            try {
                bitmap = MediaStore.Images.Media.getBitmap(getContentResolver(), filePath);
                imageView.setImageBitmap(bitmap);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

    public String getStringImage(Bitmap bmp) {
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        bmp.compress(Bitmap.CompressFormat.JPEG, 100, baos);
        byte[] imageBytes = baos.toByteArray();
        String encodedImage = Base64.encodeToString(imageBytes, Base64.DEFAULT);
        return encodedImage;
    }

    private void uploadImage() {
        class UploadImage extends AsyncTask<Bitmap, Void, String> {

            ProgressDialog loading;
            RequestHandler rh = new RequestHandler();

            @Override
            protected void onPreExecute() {
                super.onPreExecute();
                loading = ProgressDialog.show(imageupload.this, "Uploading...", null, true, true);
            }

            @Override
            protected void onPostExecute(String s) {
                super.onPostExecute(s);
                loading.dismiss();
                Toast.makeText(getApplicationContext(), s, Toast.LENGTH_LONG).show();
            }

            @Override
            protected String doInBackground(Bitmap... params) {
                Bitmap bitmap = params[0];
                String uploadImage = getStringImage(bitmap);

                HashMap<String, String> data = new HashMap<>();

                data.put(UPLOAD_KEY, uploadImage);
                data.put("action",MainActivity.usn);
                data.put("fname",name);

                String result = rh.sendPostRequest(UPLOAD_URL, data);
```

```
            return result;
        }
    }


    UploadImage ui = new UploadImage();
    ui.execute(bitmap);
  }

  @Override
  public void onClick(View v) {
    if (v == buttonChoose) {
      showFileChooser();
    }

    if (v == buttonUpload) {
      uploadImage();
    }


  }


}
```

UPD_OR_VIEW:

```
package com.example.manun.loginregdb;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class upd_or_view extends AppCompatActivity implements View.OnClickListener{
  String name;

    Button upld;
    Button vw;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
      Intent x = getIntent();

      name = x.getStringExtra("name");


      super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_upd_or_view);
        upld = findViewById(R.id.upd);
        vw=findViewById(R.id.vw);

       upld.setOnClickListener(upd_or_view.this);
       vw.setOnClickListener(upd_or_view.this);

    }

    @Override
    public void onClick(View v) {
        if (v == upld) {
            Intent x = new Intent(upd_or_view.this, imageupload.class);
            x.putExtra("name", name);

            startActivity(x);
        }

        if (v == vw) {
            Intent x = new Intent(upd_or_view.this, MainActivity1.class);
            x.putExtra("name", name);
            x.putExtra("usr",MainActivity.usn);

            startActivity(x);;

        }

    }
}
```

MAIN ACTIVITY 1:

```
package com.example.manun.loginregdb;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ListView;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONException;
```

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class MainActivity1 extends AppCompatActivity {
    ListView listView;

    String newurl = "https://mhealthmanu.000webhostapp.com/os.php?";


    private String json;
    private JSONArray urls;
    public  Getjson getjsonobj;
    Customadapter customadapter;
    Customadapter aptr;
    String name;
    String user;
    String URLL;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        Intent x = getIntent();
        String a="fname=";
        String b="&usr=";

        name = x.getStringExtra("name");
        user=x.getStringExtra("usr");
        URLL=newurl+a+name+b+user;
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main1);
        listView = (ListView) findViewById(R.id.lst);
        getURLs();
        Toast.makeText(this, URLL, Toast.LENGTH_LONG).show();
    }


    private void getImages() {
        class GetImages extends AsyncTask<Void, Void, Void> {
            ProgressDialog loading;

            @Override
            protected void onPreExecute() {
                super.onPreExecute();
                loading = ProgressDialog.show(MainActivity1.this, "Loading Menu", "Please
wait...", false, false);
            }

            @Override
```

```
        protected void onPostExecute(Void v) {
            super.onPostExecute(v);
            loading.dismiss();


              customadapter = new Customadapter(MainActivity1.this,
getjsonobj.Android_Name ,getjsonobj.bitmaps );


            listView.setAdapter(customadapter);
            listView.setOnItemClickListener(new AdapterView.OnItemClickListener(){
               public void onItemClick(AdapterView<?> adapter, View v, int position, long
id){
                  String x= Integer.toString(position);
                   int i=position;

                   if (position==i)
                   {


                      Intent z = new Intent(MainActivity1.this, webviewimg.class);
                      z.putExtra("url", Getjson.Image_Url[i]);
                      startActivity(z);

                      Toast.makeText(MainActivity1.this,Getjson.Image_Url[i],
Toast.LENGTH_SHORT).show();


                   }
               }
            });
        }

        @Override
        protected Void doInBackground(Void... voids) {
            try {
               getjsonobj.getAllImages();

            } catch (JSONException e) {
               e.printStackTrace();
            }
            return null;
        }
    }
    GetImages getImages = new GetImages();
    getImages.execute();
  }

  private void getURLs() {

    class GetURLs extends AsyncTask<String, Void, String > {
```

```
        ProgressDialog loading;

        @Override
        protected void onPreExecute() {
            super.onPreExecute();
            loading = ProgressDialog.show(MainActivity1.this, "Loading...", "Please Wait...",
true, true);
        }

        @Override
        protected void onPostExecute(String s) {
            super.onPostExecute(s);
            loading.dismiss();
            getjsonobj = new Getjson(s);
            getImages();
        }

        @Override
        protected String doInBackground(String... strings) {
            BufferedReader bufferedReader = null;
            try {

                URL url = new URL(strings[0]);
                HttpURLConnection con = (HttpURLConnection) url.openConnection();
                StringBuilder sb = new StringBuilder();

                bufferedReader = new BufferedReader(new
InputStreamReader(con.getInputStream()));

                String json;
                while ((json = bufferedReader.readLine()) != null) {
                    sb.append(json + "\n");
                }

                return sb.toString().trim();

            } catch (Exception e) {
                return null;
            }
        }
    }
    GetURLs gu = new GetURLs();
        gu.execute(URLL);
  }
}
```

Screenshots:

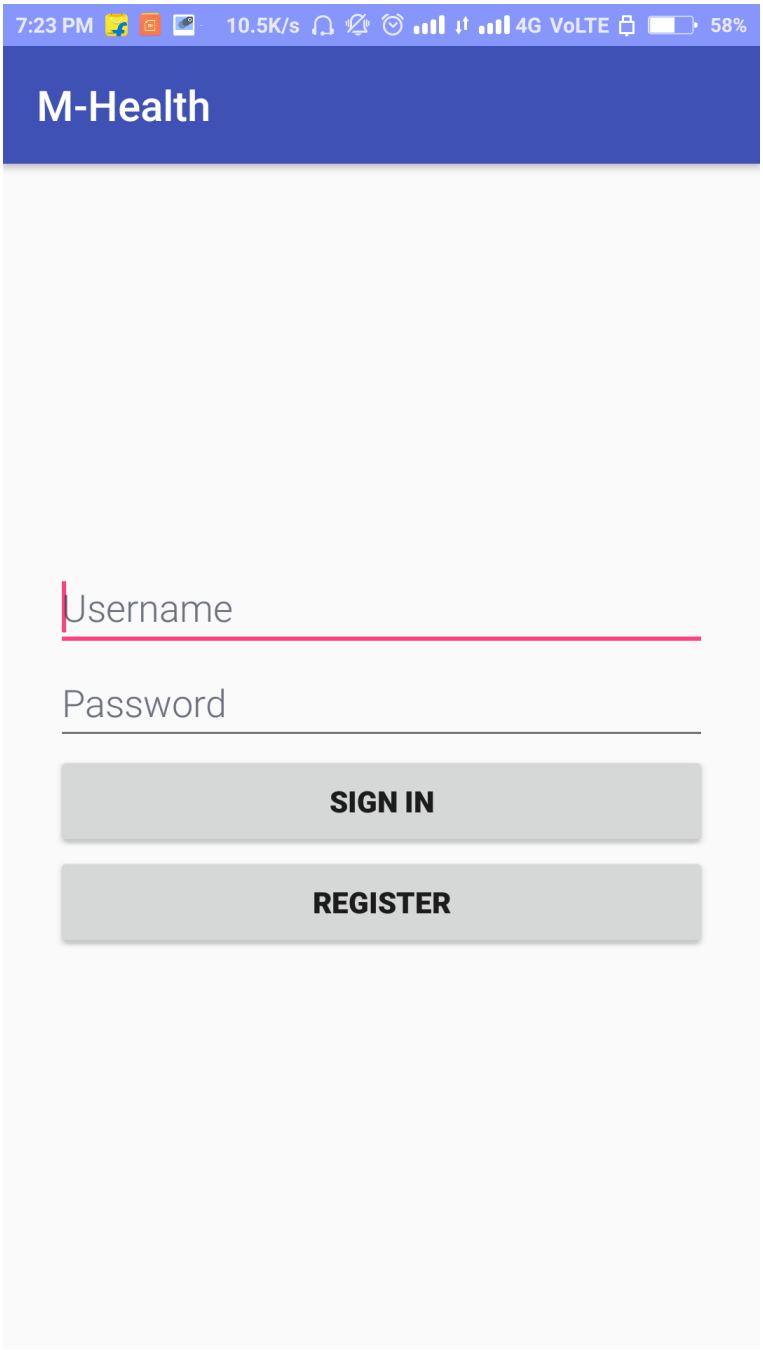


Fig.1 Login Page

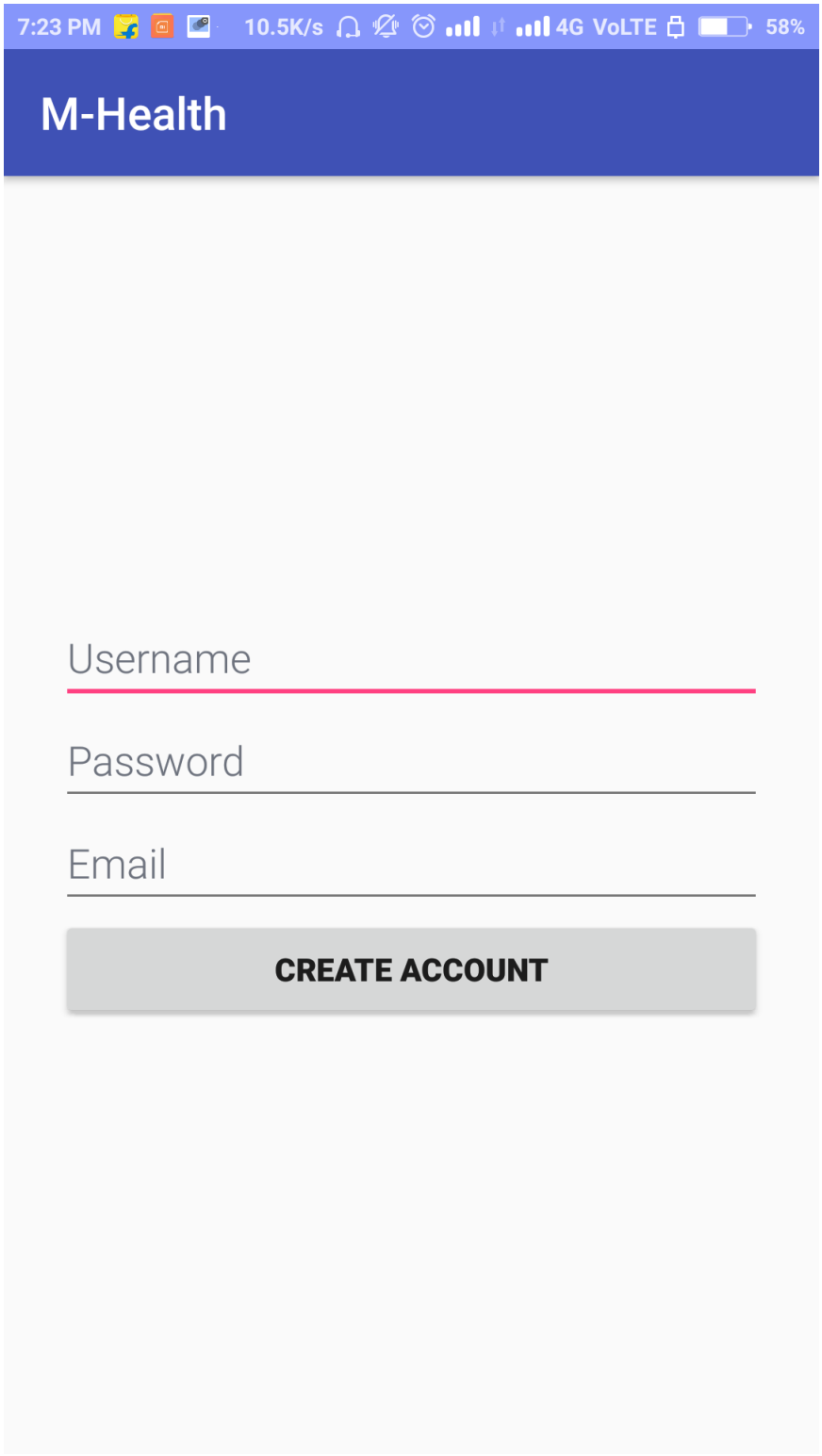


Fig.2 Registration Screen

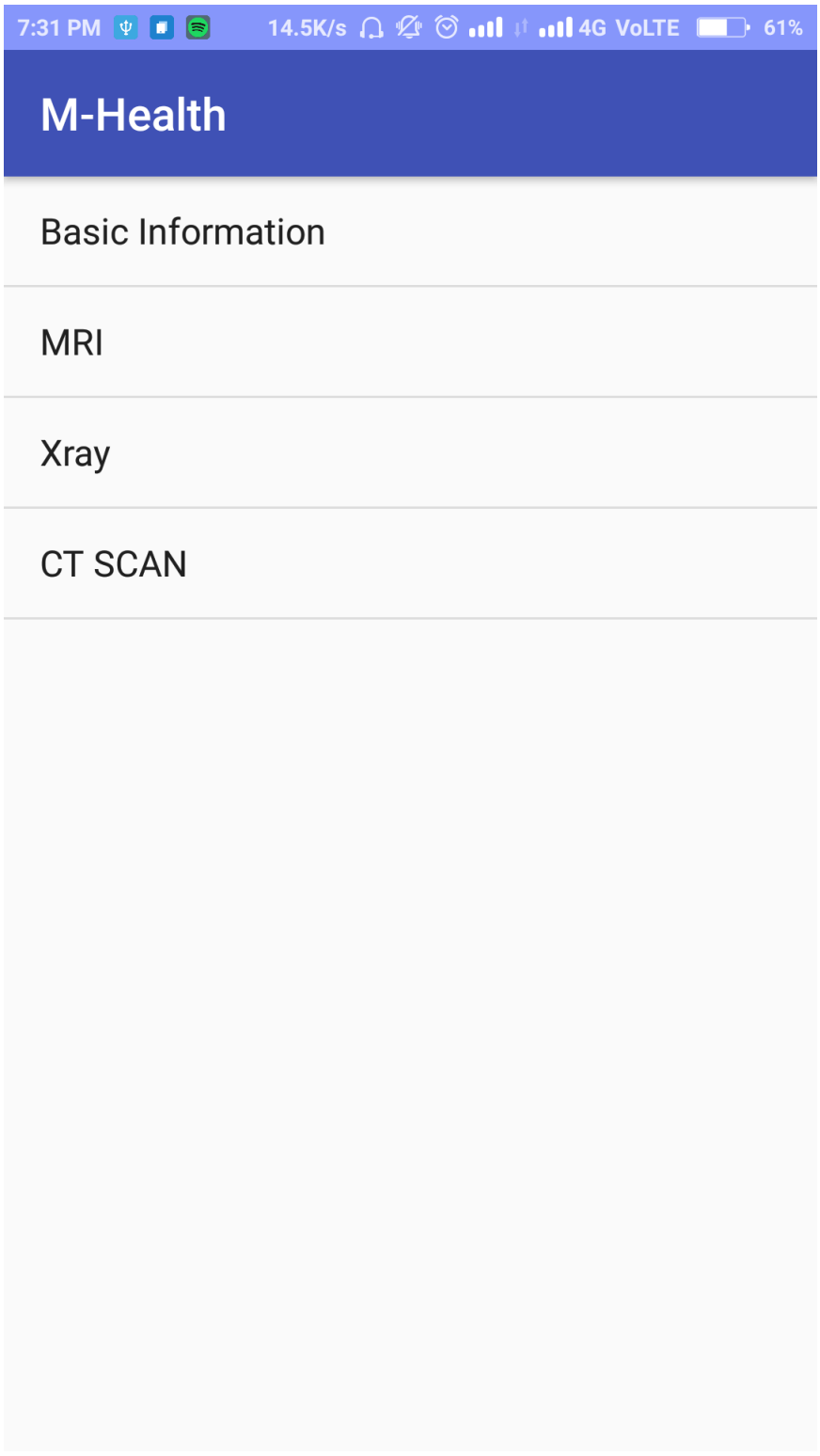


Fig.3 Home Screen

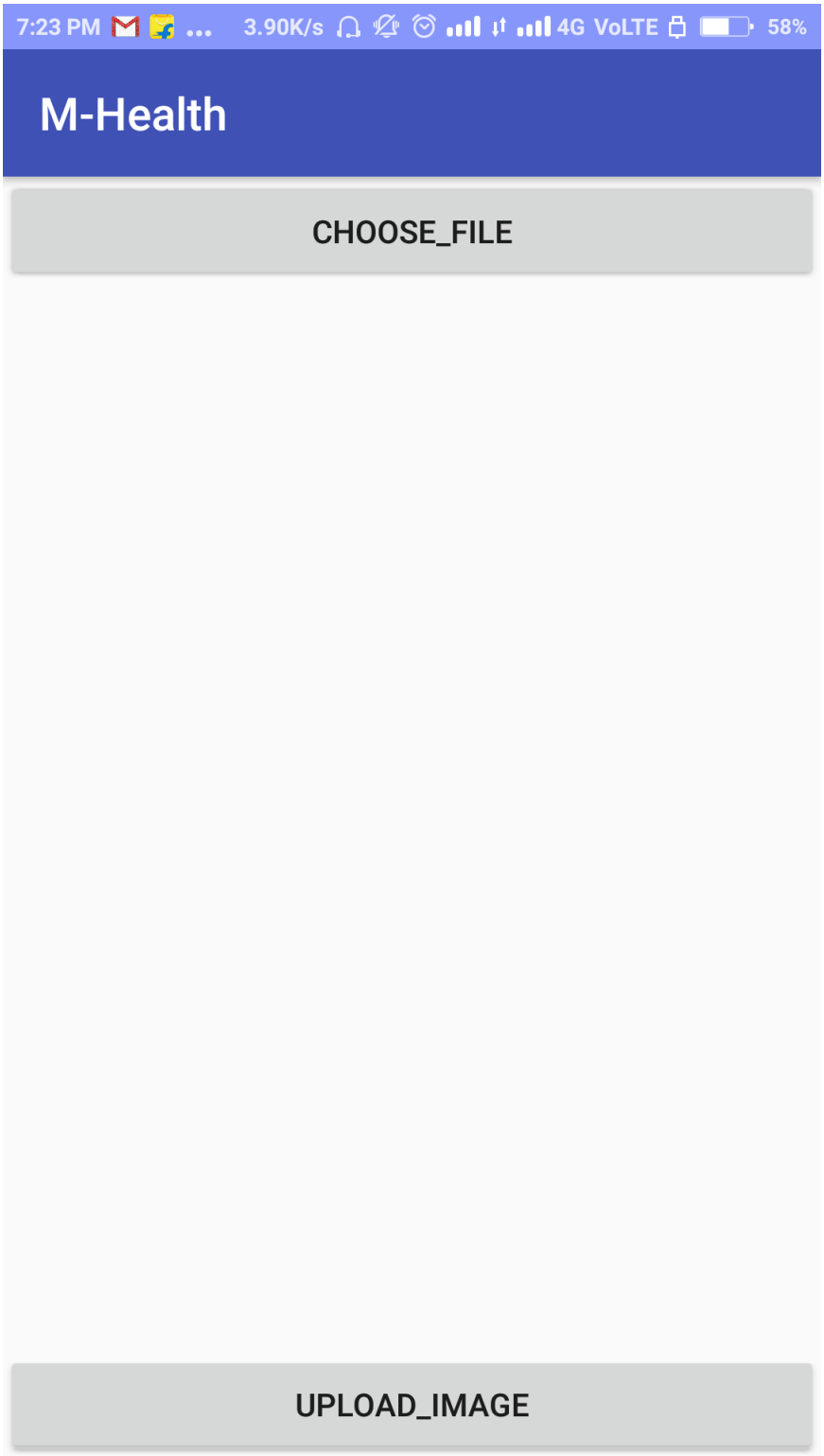


Fig 4 Upload and View Screen

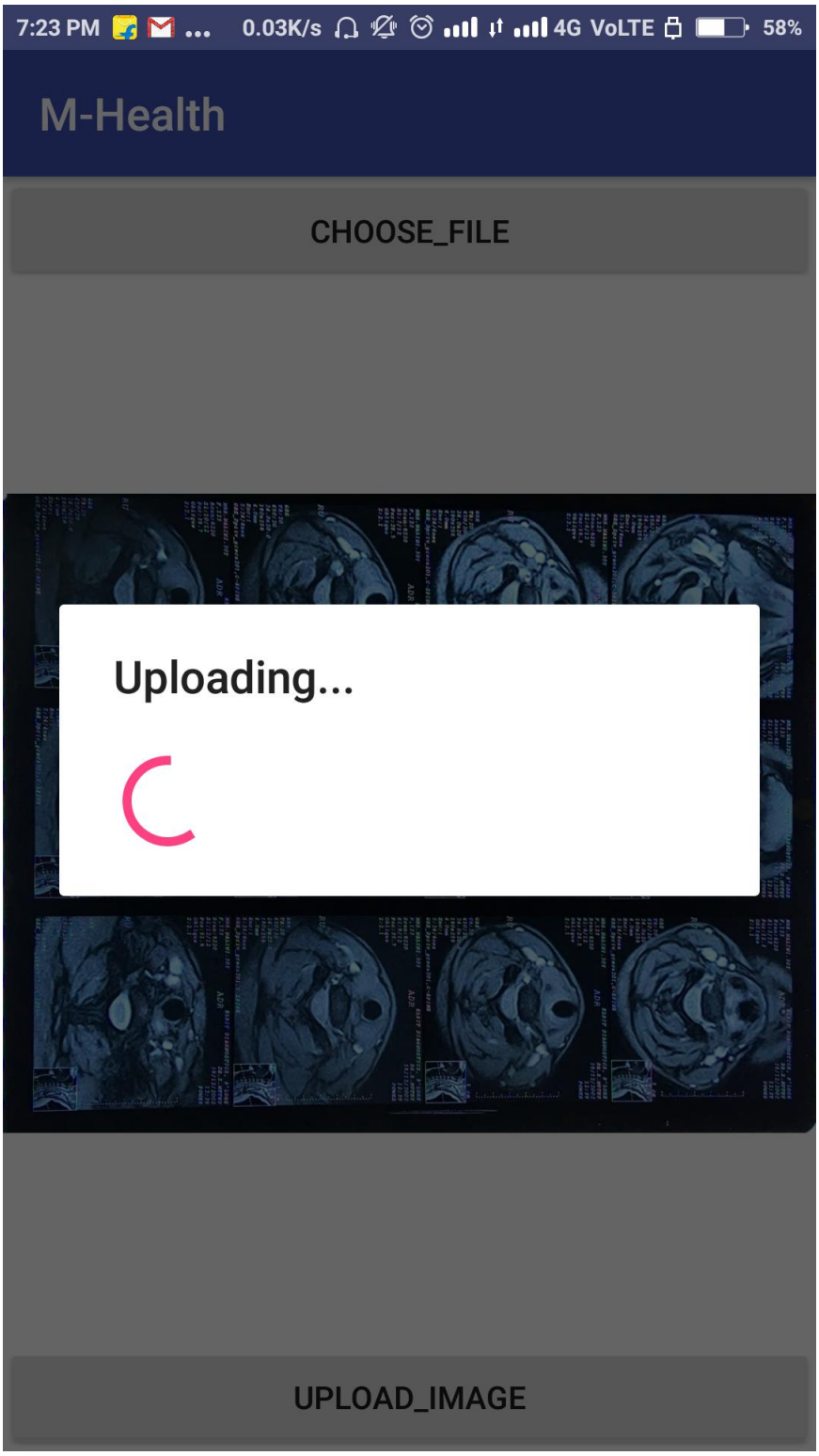


Fig 5 Uploading an X RAY

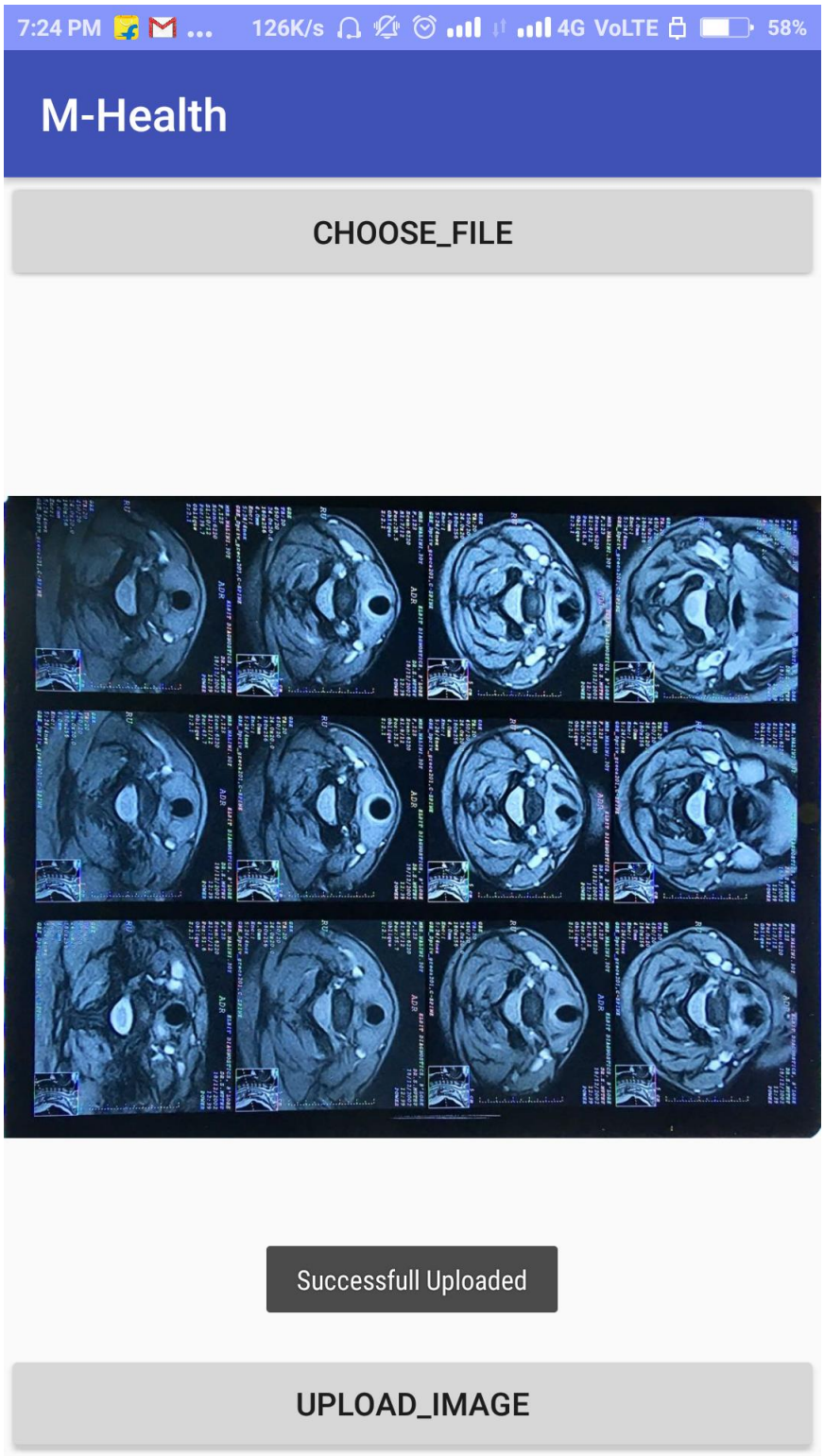


Fig 7 Successfully Uploaded

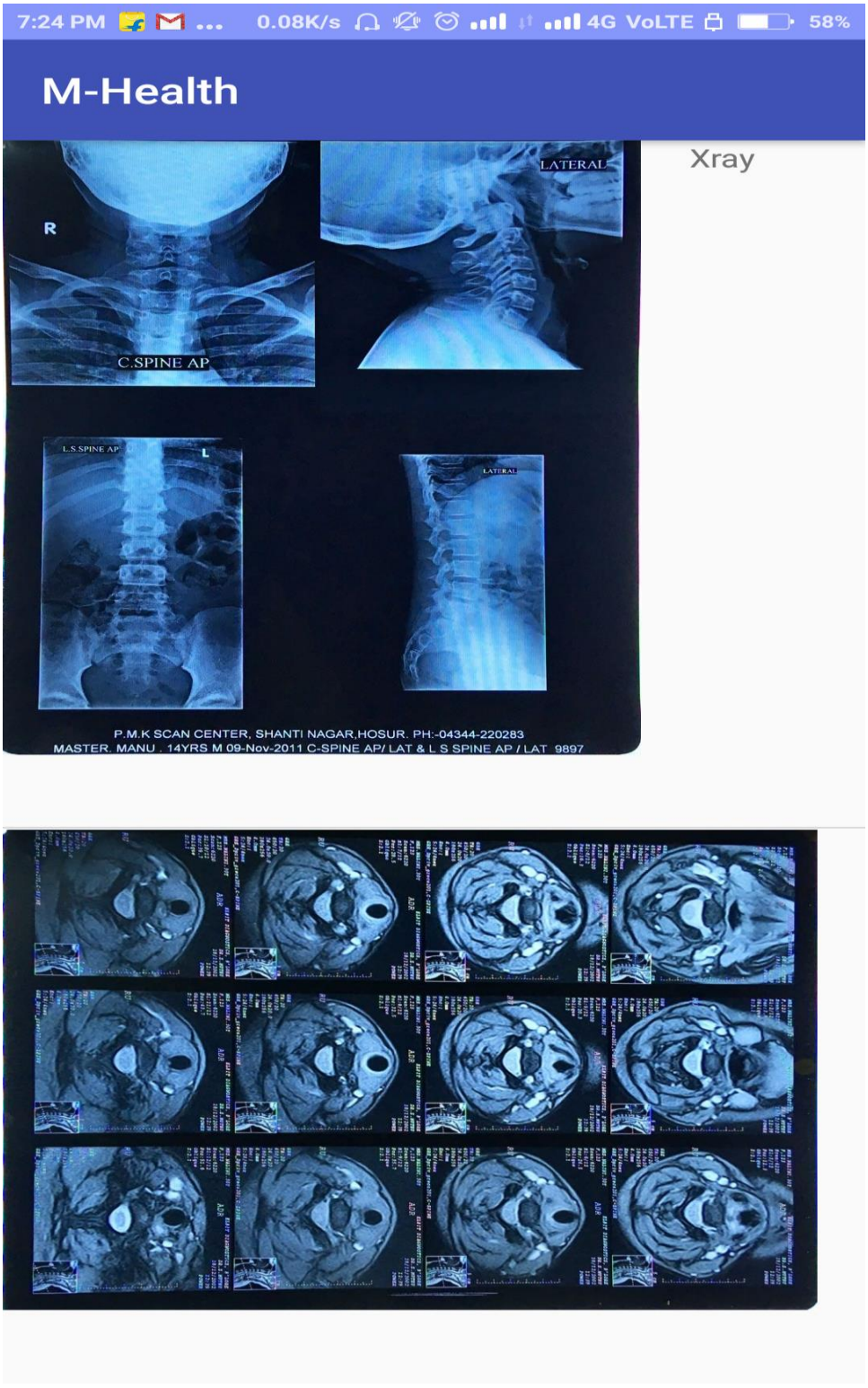


Fig 8 Viewing uploaded X RAY

References:

[1] R. Sathiyavathi, A Survey: Big Data Analytics on Healthcare System, Contemporary Engineering Sciences, Vol. 8, no. 3, 121 - 125 HIKARI Ltd, www.m-hikari.com, 2015.

[2] Christopher Pearce (2009), Electronic medical records - where to from here ? Melbourne: Professional Practice.

[3] GSMA/McKinsey&Company Report, "mHealth: A new vision for healthcare",2015.

[4] Singer Stacey(2010). Electronic Medical Records May Cause Patient Care Errors, Florida Medical Board Says.

[5] Deloitte Development LLC report, "mHealth in an mWorld: How mobile technology is transforming health care" Deloitte Center for Health Solutions, 2012.

[6] M V Ramana Murthy, "Mobile based Primary Health Care System for Rural India," CDAC report,2016.

[7] Leon N, Schneider H, Daviaud E. Applying a framework for assessing the health system challenges to scaling up mHealth in South Africa. *BMC Medical Informatics and Decision Making,* 2012; 12:123 doi:10.1186/1472-6947-12-123

[8] Gurman TA, Rubin SE, Roess AA. Effectiveness of mHealth behavior change communication interventions in developing countries: A systematic review of the literature. *Journal of Health Communication*, 2012; 17 (sup1): 82-104.

[9] Fanning J, Mullen SP, McAuley E. Increasing physical activity with mobile devices: A meta-analysis. *Journal of Medical Internet Research*, 2012; 14(6):e161

[10] de Jongh T, Gurol-Urganci I, Vodopivec-Jamsek V, Car J, Atun R. Mobile phone messaging for facilitating self-management of long-term illnesses. *Cochrane Database Systematic Review*, 2012;12: