

# **MINOR PROJECT**

## **POCKET SCANNER ANDROID APPLICATION**



**GURU GOBIND SINGH INDRAPIRASTHA UNIVERSITY**

**UNIVERSITY SCHOOL OF INFORMATION  
COMMUNICATION AND TECHNOLOGY**

**Under the supervision of**

**Mentor:**

Prof. Navin Rajpal

**Submitted by:**

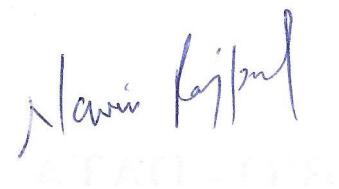
Vikrant Singh Jamwal

B.TECH. - IT

03116401517

## **CERTIFICATE**

This is to certify that the project entitled "**Pocket Scanner Android Application**" submitted by "**Vikrant Singh Jamwal**" to University School of Information, Communication and Technology, GGSIPU in partial fulfilment of the requirement for the award of the degree of B.Tech in Information Technology. This project work carried out by him under my guidance. The project fulfils the requirement as per the regulation of this institute and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university to the best of my knowledge and belief.



**Date:** 06/01/2021

**Prof. Navin Rajpal**  
**USICT, GGSIPU**  
**New Delhi**

## **DECLARATION**

This is to certify that the work being presented in the project entitled "**Pocket Scanner Android Application**" submitted by undersigned student of Bachelors in **Information Technology** to University School of Information Communication and Technology in the fulfilment for award of the degree of Bachelors in **Information Technology** is a record of my own work carried out by me under guidance and supervision of **Professor Navin Rajpal**. I certify the content of this work has not submitted elsewhere for award of any other degree.



**Date:** 05/01/2021

**Signature of Student**

Vikrant Singh Jamwal

B.Tech.-IT, 7<sup>th</sup> semester

(03116401517)

## **ACKNOWLEDGMENT**

I would like to express my sincere gratitude towards **Prof. Pravin Chandra** , Dean, University School of Information & Technology, GGSIPU for giving me this opportunity of working on this project and **Dr. Sartaj Singh Sodhi**, coordinator, who provided a wide vision of scope of the project which was very important to me.

I am also thankful to my mentor, **Prof. Navin Rajpal** for his most valuable and significant guidance throughout the course of the project and in elaborating our view of studying the project's details and getting the right vision for its implementation.

At last, I would like to thank the Department of Information Technology - USICT, GGSIPU, that has played an important role in strengthening my career by providing full cooperation and encouragement throughout the tenure of the project.

## Contents

S. no.	Topics	Page no.
1	<b>Certificate</b>	2
2	<b>Declaration</b>	3
3	<b>Acknowledgement</b>	4
4	<b>Introduction</b>	6
5	<b>Problem statement</b>	9
6	<b>Idea Behind the Project</b>	10
7	<b>Flow Chart</b>	11
8	<b>Use Case</b>	12
9	<b>Requirements</b> <ul style="list-style-type: none"> <li>• Hardware</li> <li>• Software</li> <li>• Libraries</li> </ul>	13
10	<b>Application Codes and UI</b> <ul style="list-style-type: none"> <li>• Splash Screen</li> <li>• Home screen</li> <li>• Adapter</li> <li>• Scan Activity</li> <li>• Manifest Permissions</li> <li>• Image Processing</li> <li>• Advanced REST client</li> </ul>	14
11	<b>References</b>	21

## **Introduction**

- Android Platform

Android is an open source Linux based Operating System. Android offers a unified approach to application development for mobile devices which means developers need to develop for Android, and their applications should be able to run on different devices powered by Android [1]. Using Android Platform will increase the reach of the application making it used by a vast number android users.

For the making of an Android application, Android Studio is a software which helps in creating the backend and the frontend of the desired application.

- Android APIs till date

Android11	11	API level 30
Android10	10	API level 29
Pie	9	API level 28
Oreo	8.1.0	API level 27
Oreo	8.0.0	API level 26
Nougat	7.1	API level 25
Nougat	7.0	API level 24
Marshmallow	6.0	API level 23

Lollipop	5.1	API level 22
Lollipop	5.0	API level 21
KitKat	4.4 - 4.4.4	API level 19
Jelly Bean	4.3.x	API level 18
Jelly Bean	4.2.x	API level 17
Jelly Bean	4.1.x	API level 16
Ice Cream Sandwich	4.0.3 - 4.0.4	API level 15, NDK 8
Ice Cream Sandwich	4.0.1 - 4.0.2	API level 14, NDK 7
Honeycomb	3.2.x	API level 13
Honeycomb	3.1	API level 12, NDK 6
Honeycomb	3.0	API level 11
Gingerbread	2.3.3 - 2.3.7	API level 10
Gingerbread	2.3 - 2.3.2	API level 9, NDK 5
Froyo	2.2.x	API level 8, NDK 4
Eclair	2.1	API level 7, NDK 3
Eclair	2.0.1	API level 6
Eclair	2.0	API level 5
Donut	1.6	API level 4, NDK 2
Cupcake	1.5	API level 3, NDK 1
(no codename)	1.1	API level 2
(no codename)	1.0	API level 1

- Frontend and Backend
  - Frontend is written using XML files. It is used to create the user interface of the application. Features can be written as a code or dragged from the bar.
  - Backend is written using Java, which connects the activities together and provide functions to the object used by the XML files.

- Python Script for image processing

The logic behind this is that the point where an edge exists, there is an abrupt intensity change, which causes a spike in the first derivative's value, hence making that pixel an 'edge pixel'. This is used by the CV2 which is an OpenCV package which helps in image processing by reducing its noise and changing its colour format. After finding an edge the image is cropped and saved.



## **Problem Statement**

In this fast growing era of technology, developers are focusing on creating things to save time and energy in a significant manner.

Also, analysing the current scenario between India and China, India has banned several applications including Cam Scanner, this project is capable of replacing the widely used cam scanner.

Cam scanner was widely used for creating files of official documents without using the actual scanner. Getting the pictures from any device and after processing converting it to any chosen file format.

Creating an Android based application to convert a picture taken by the android device to the .pdf file after the image is processed using python script and packages like cv2, numPy and Flask.

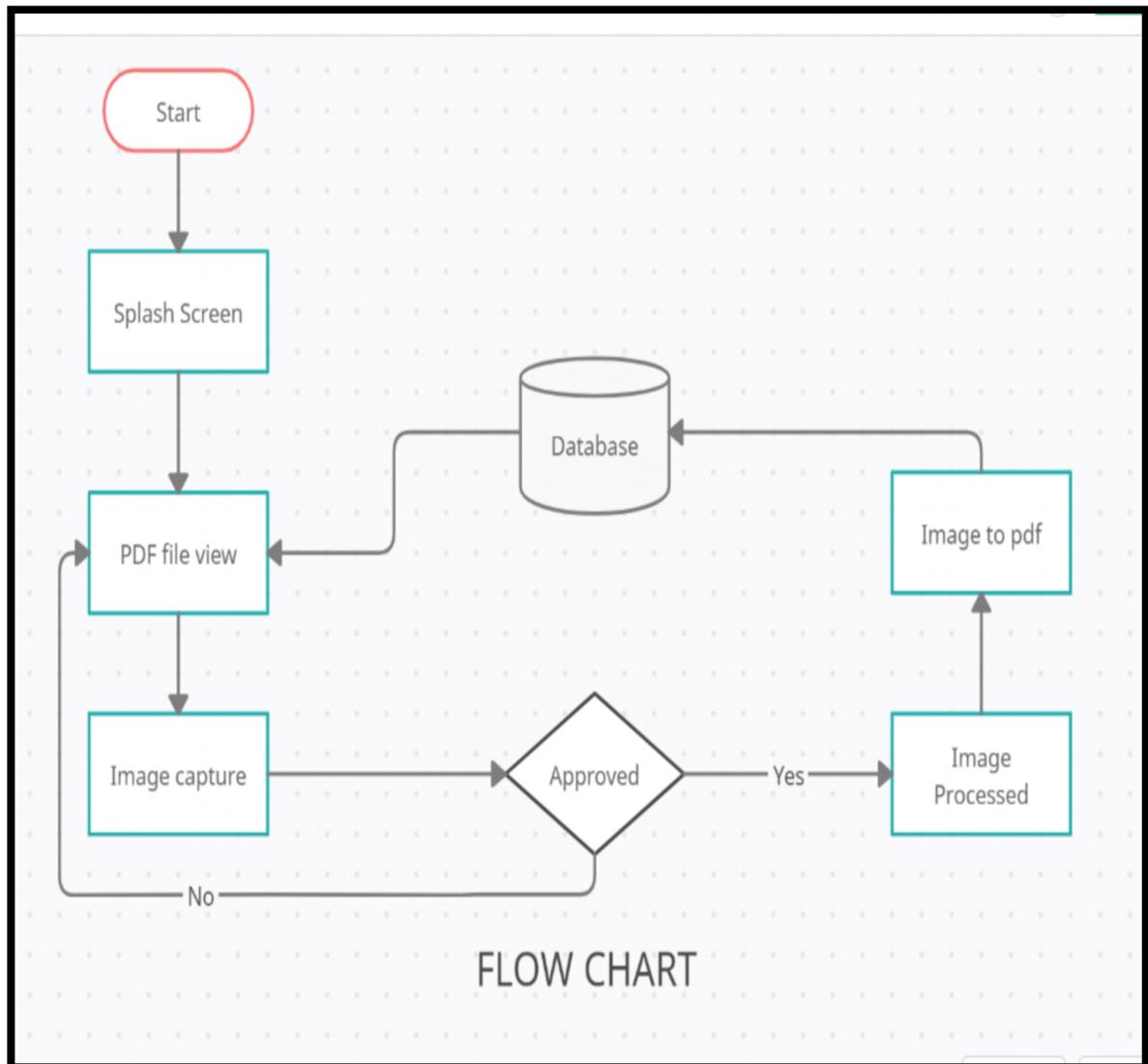
Application should have a pdf viewer activity and an option of taking a picture and converting it to a .pdf file.

## **Idea behind the project**

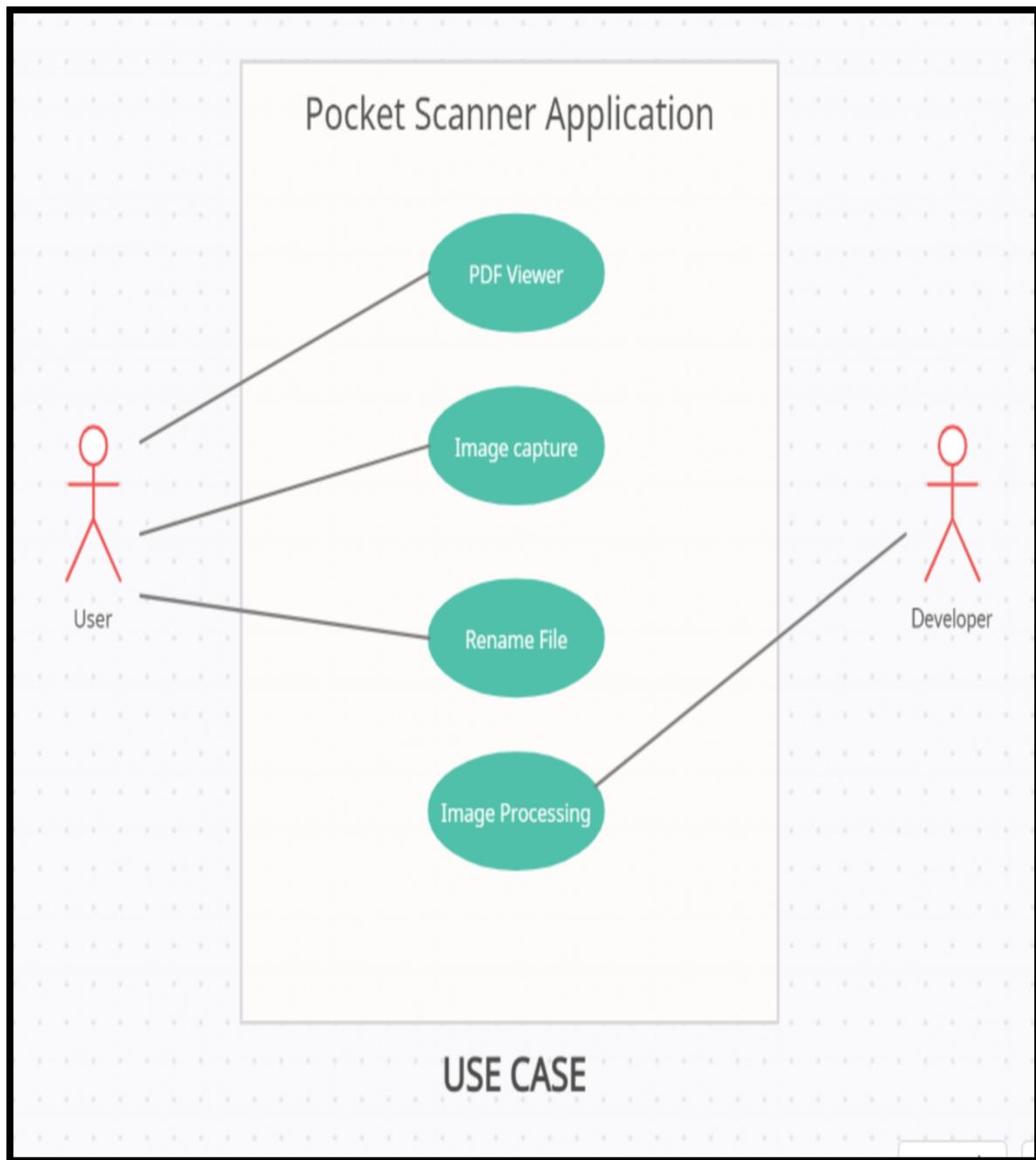
Pocket Scanner is a project which allows the user to scan documents without even using a scanner. Using a scanner was hectic and also not everyone can afford a scanner, this project aims to provide a scanner in the hands of every user through mobile phones.

Just by clicking a picture through the camera or selecting the files from the storage of the phone, any user can easily convert those images to a document with required format. This will save a significant amount of time, which is the basic pillar of technology.

# Flow Chart



## Use Case



## **Requirements**

### Hardware requirement

1. Processor: Intel Atom processor or above [2]
2. Clock speed: 1.2 GHz or faster
3. Storage: 1 GB at least
4. RAM: More than 512 MB

### Software Requirements

1. API level: 19 (Android 4.4 KITKAT) or above
2. Operating System: Android
3. Developing software: Android Studio [3], Python Idle
4. Server Software: Advanced rest Client (localhost)

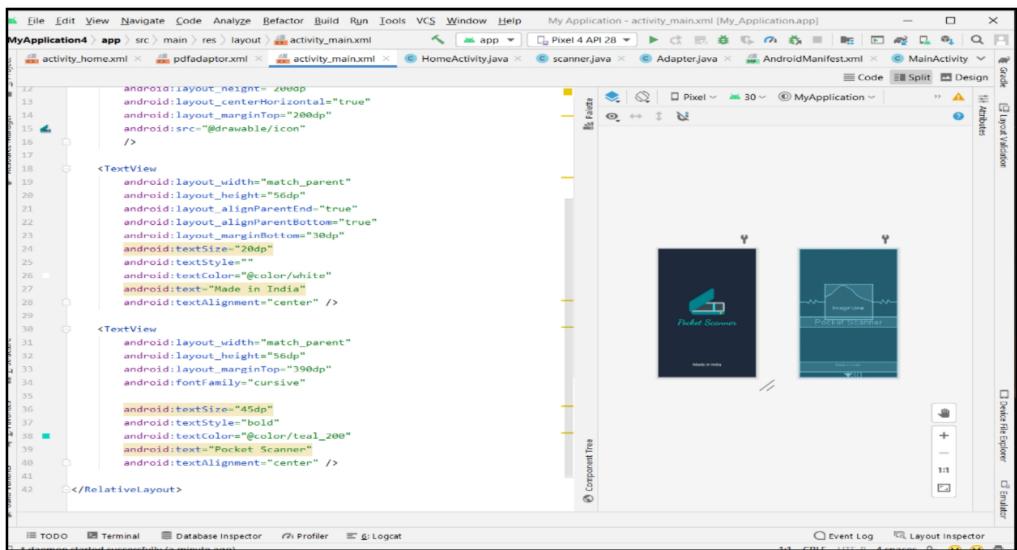
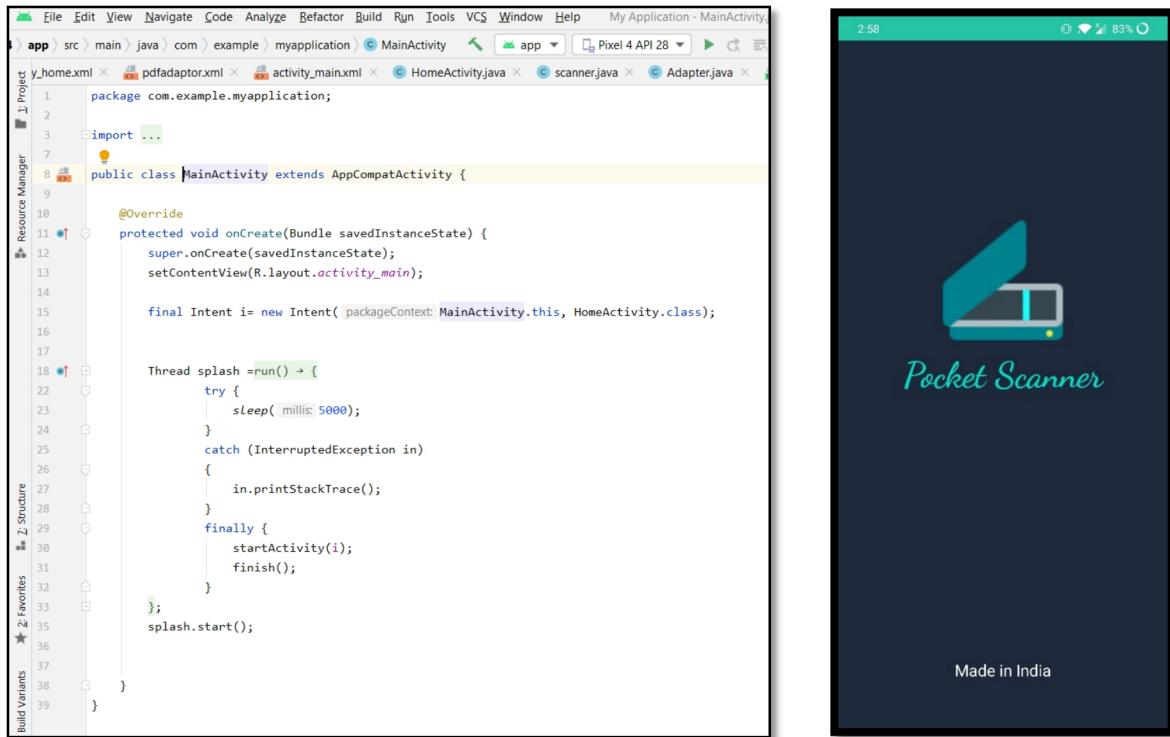
### Libraries Used

1. Flask
2. pdfViewer
3. ArrayList
4. NumPy
5. Cv2

# Application Codes and UI

## 1. Splash Screen

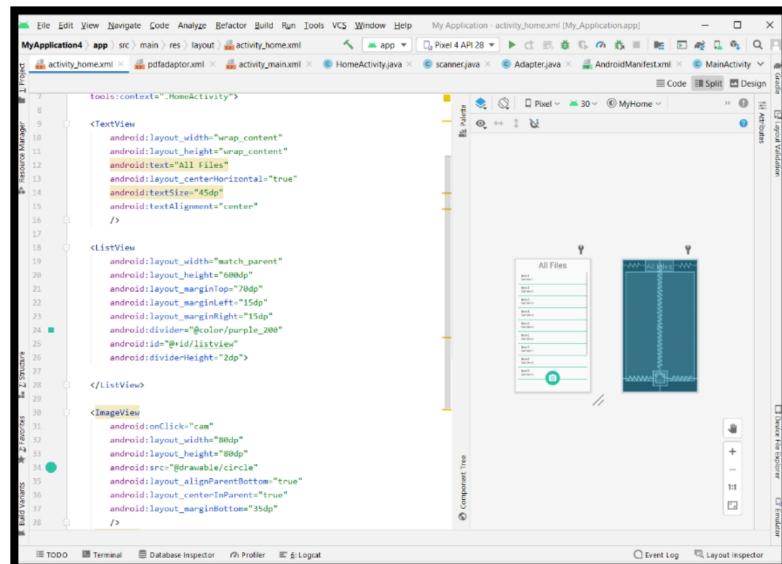
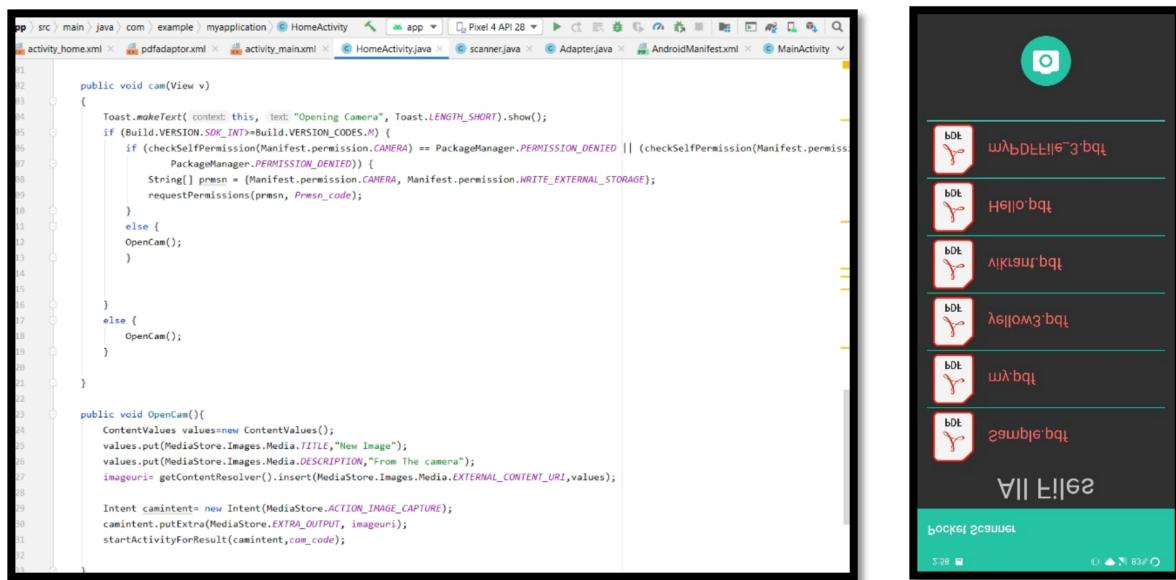
Screen with Logo and title of the application stays for 5 sec.



## 2. Home Screen

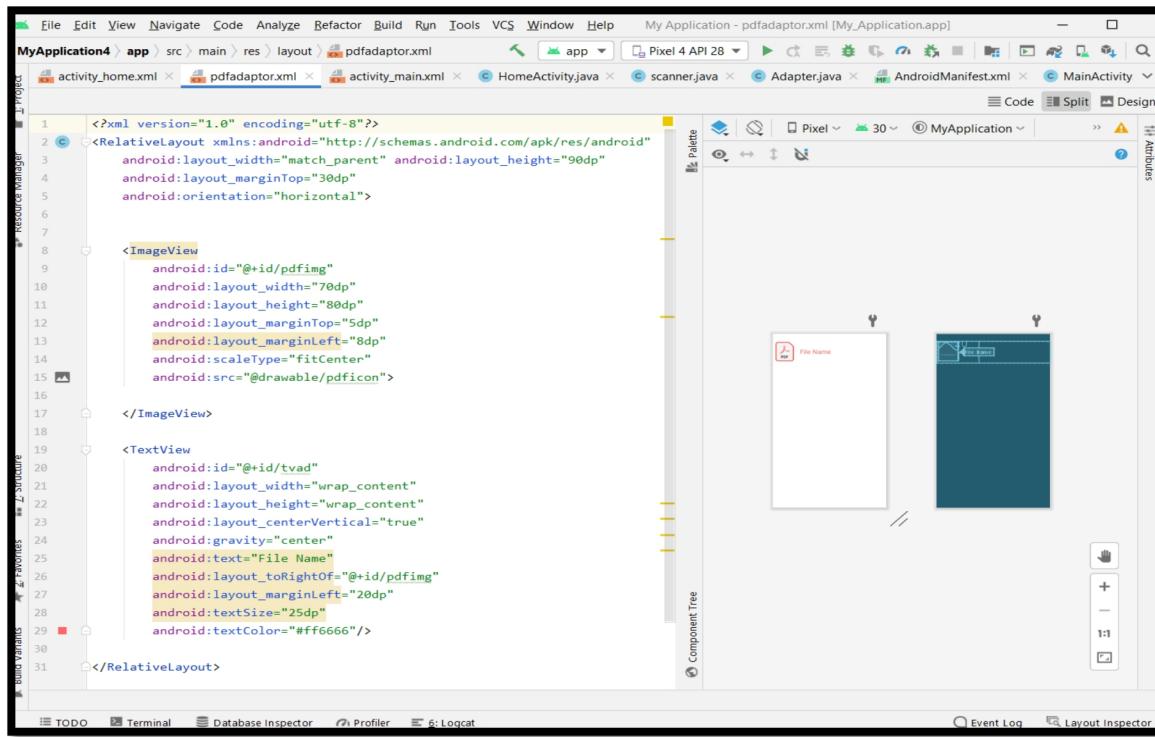
Contains list of all the pdf files made by the application. All the files are kept in the Pocket Scanner folder and is called as an array list to the home screen using an Adapter and Listview tag.

Camera button is also placed at the bottom to click a new picture. Asks for the permission of writing and reading external storage data and use of camera by the application.



### 3. Adaptor Class

Adaptor class extends Array List and collects the pdf files created by the application and then used by the Listview to show the files on the home screen.



The screenshot shows the Java code for the 'Adapter.java' class. It extends 'ArrayAdapter<File>' and overrides methods like 'getCount()', 'getItemType(int position)', and 'getView(int position, View view, ViewGroup parent)'. The code uses 'R.layout.pdfadaptor' as the layout resource for each item.

```
package com.example.myapplication;

import ...

public class Adapter extends ArrayAdapter<File> {

    Context context;
    ArrayList<File> all_pdf;
    ViewHolder viewHolder;

    public Adapter(Context context, ArrayList<File> all_pdf)
    {
        super(context,R.layout.pdfadaptor,all_pdf);
        this.context=context;
        this.all_pdf=all_pdf;
    }

    @Override
    public int getItemViewType(int position) { return position; }

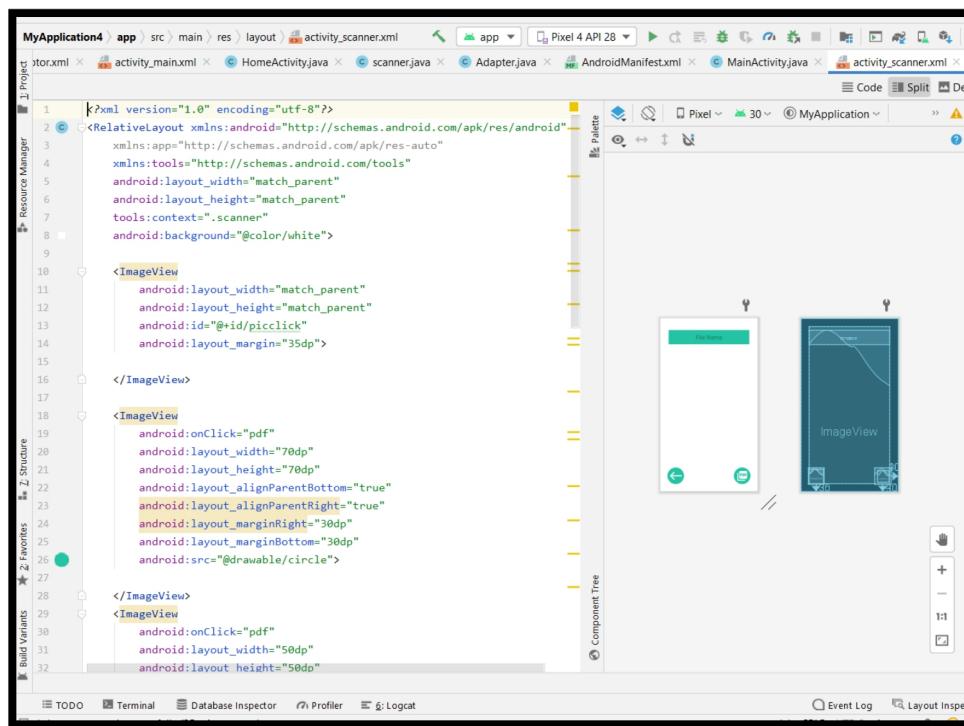
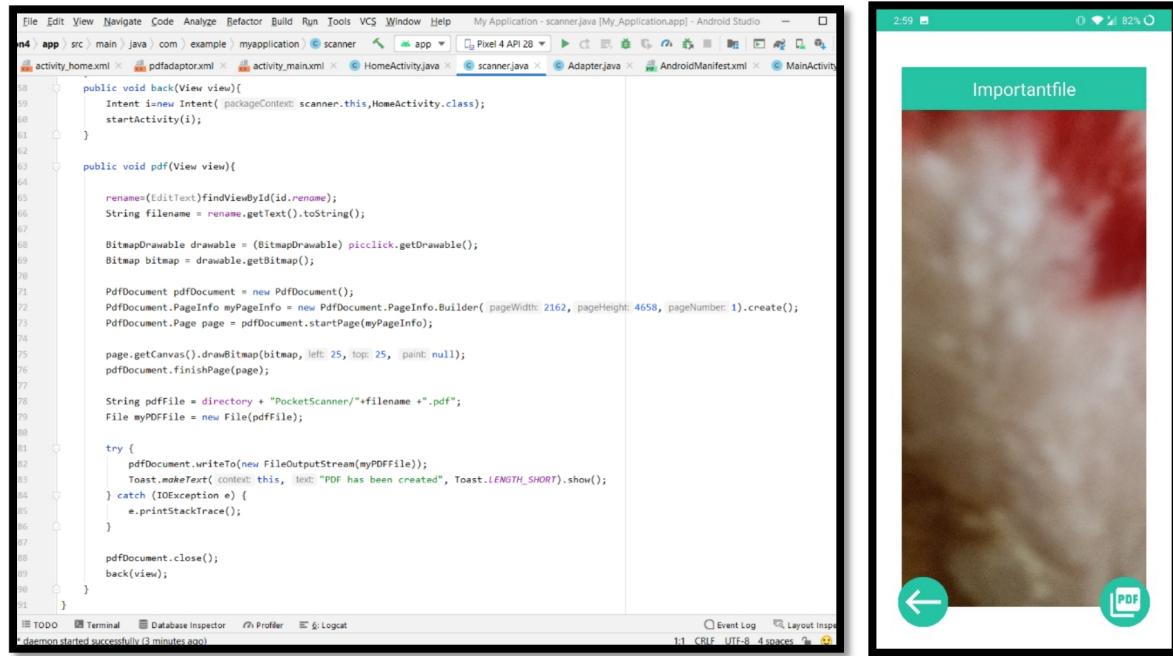
    @Override
    public int getCount() {
        if (all_pdf.size()>0) {
            return all_pdf.size();
        }else {
            return 1;
        }
    }

    @NonNull
    @Override
    public View getView(int position, @Nullable View view, @NonNull ViewGroup parent) {
```

## 4. Scan Activity

This activity comprises of nomenclature of the pdf file to be created and confirming the image taken by the camera.

After confirming and giving a name file can be converted using the pdf button below. Option for going back to the home screen is also present in the bottom.



## 5. Manifest Permissions

Four permissions are asked from the device by the application:

- Using Camera
- Using Internet
- Read External Storage
- Write External storage



The screenshot shows the AndroidManifest.xml file in the Android Studio editor. The manifest file contains the following XML code:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.myapplication">

    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
        tools:ignore="ScopedStorage" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:requestLegacyExternalStorage="true"
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Pocket Scanner"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.MyApplication">
        <activity android:name=".scanner"></activity>
        <activity
            android:name=".HomeActivity"
            android:theme="@style/Theme.MyHome"></activity>
        <activity
            android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

```

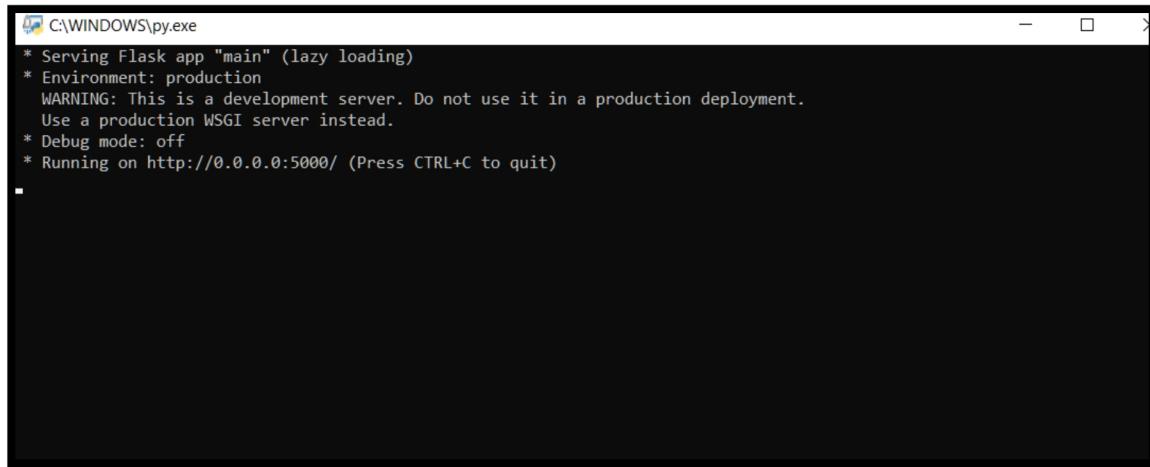
The permissions section is highlighted in blue, and the activity definitions for HomeActivity and MainActivity are highlighted in yellow.

## 6. Image Processing

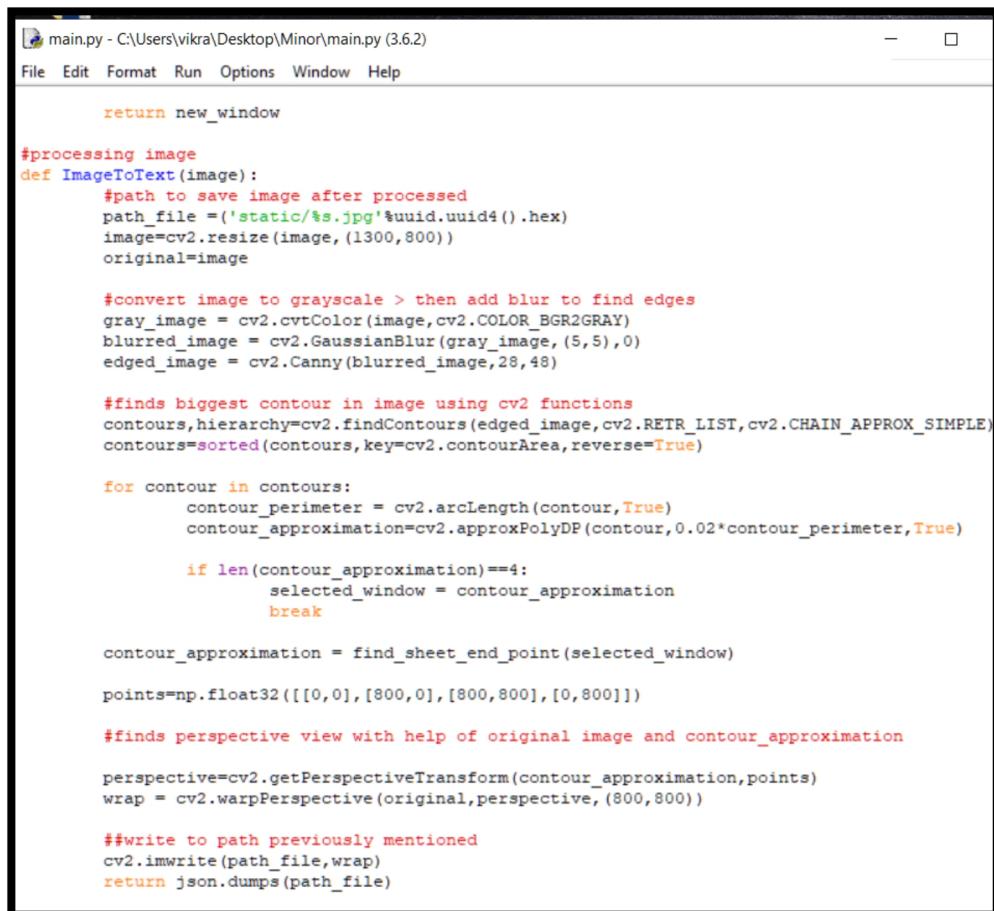
Image is processed by a python script which uses packages like:

Flask: Creating web instances for the server.

CV2: To convert the image to the grey scale and using blur to identify the edges of the image to clear out the unwanted part.



```
C:\WINDOWS\py.exe
 * Serving Flask app "main" (lazy loading)
 * Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```



```
main.py - C:\Users\vikra\Desktop\Minor\main.py (3.6.2)
File Edit Format Run Options Window Help

    return new_window

#processing image
def ImageToText(image):
    #path to save image after processed
    path_file =('static/%s.jpg'%uuid.uuid4().hex)
    image=cv2.resize(image,(1300,800))
    original=image

    #convert image to grayscale > then add blur to find edges
    gray_image = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
    blurred_image = cv2.GaussianBlur(gray_image,(5,5),0)
    edged_image = cv2.Canny(blurred_image,28,48)

    #finds biggest contour in image using cv2 functions
    contours,hierarchy=cv2.findContours(edged_image,cv2.RETR_LIST,cv2.CHAIN_APPROX_SIMPLE)
    contours=sorted(contours, key=cv2.contourArea, reverse=True)

    for contour in contours:
        contour_perimeter = cv2.arcLength(contour,True)
        contour_approximation=cv2.approxPolyDP(contour,0.02*contour_perimeter,True)

        if len(contour_approximation)==4:
            selected_window = contour_approximation
            break

    contour_approximation = find_sheet_end_point(selected_window)
    points=np.float32([[0,0],[800,0],[800,800],[0,800]])

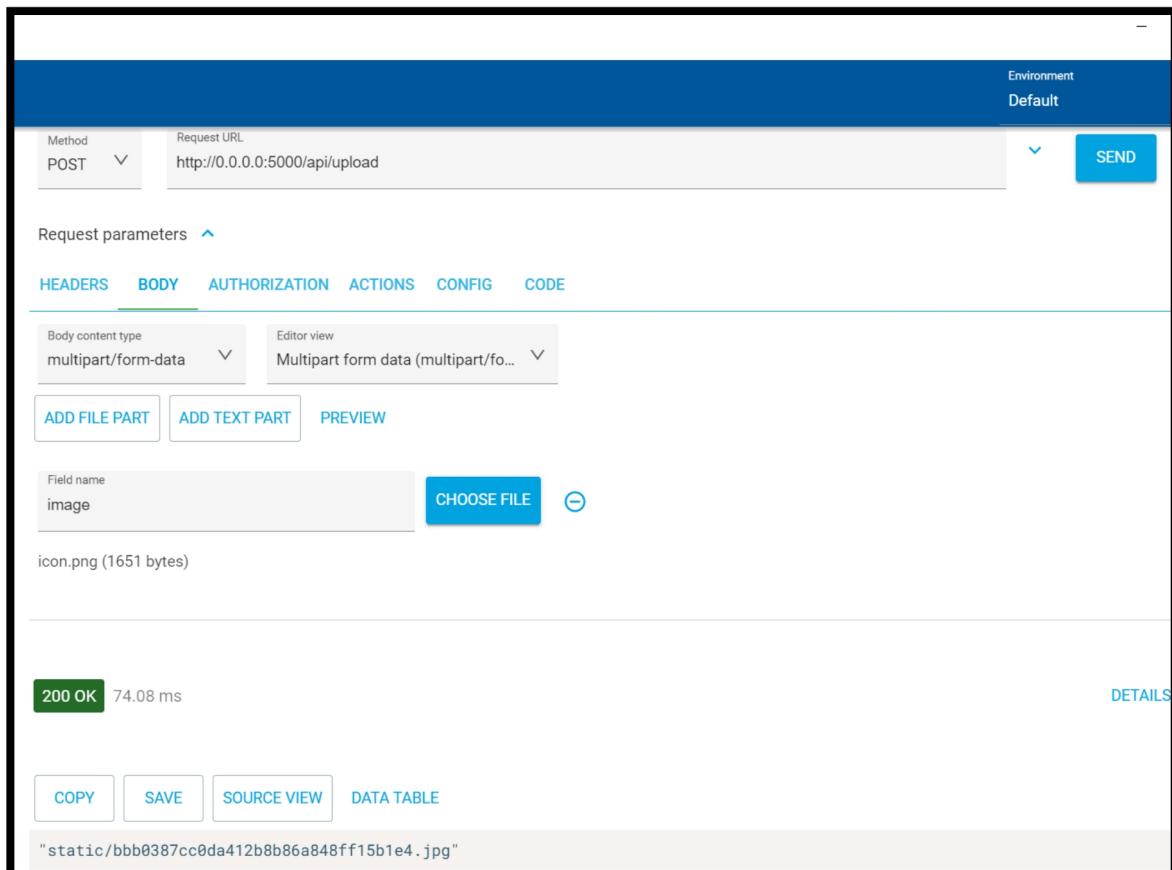
    #finds perspective view with help of original image and contour_approximation
    perspective=cv2.getPerspectiveTransform(contour_approximation,points)
    wrap = cv2.warpPerspective(original,perspective,(800,800))

    ##write to path previously mentioned
    cv2.imwrite(path_file,wrap)
    return json.dumps(path_file)
```

## 7. Using ARC (Advanced Rest Client)

This is used to test the python script by executing it in the local server created.

Amazon's EC2 can be used as the server for this application.



As we uploaded the image we can see that it runs the python script and downloaded the processed image in the static folder.

This image is then read by the application and shown in the Scan activity which then converts it into PDF file format.

## **References**

1. <https://www.tutorialspoint.com/android/androidoverview.htm>
2. [www.intel.in](http://www.intel.in)
3. [www.developer.android.com](http://www.developer.android.com)
4. <https://chrome.google.com/webstore/detail/advanced-rest-client>
5. <https://www.youtube.com/>
6. <https://www.w3schools.com/>
7. <https://stackoverflow.com/>

## **Books**

1. Android Application Development by Hewlett Packard enterprise
2. Java: A Beginner's Guide by Herbert Schildt