Faculty of Computers &
Artificial Intelligence

Benha University

Smart Glasses for Visually Impaired People

Computer Science Departement,

Project Team

1- Abd ElGawad Elsayed Abd ElGawad Abdrabo
2- Doaa Mohammed Mohammed Mohammed
3- Mohammed Mahmoud Shehata Abdelghafar
4- Mohammed Ali Abd ElDaym Ali
5- Nermeen Nesem Samir Aziz
6- Rana Mohammed Mahmoud Ali

Under Supervision of

Dr. Mohamed Taha
Eng.Mohamed Ibrahim

Benha, August 2020

ACKNOWLEDGMENT

We offer our sincere thanks and appreciation to Dr. Mohamed Taha and Eng. Mohamed Ibrahim for their efforts and assistance to us in our graduation project.

DECLARATION

We hereby certify that this material, which we now submit for assessment on the program of study leading to the award of Bachelor of Computers and Informatics in computer science is entirely our own work, that we have exercised reasonable care to ensure that the work is original, and does not to the best of our knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of our work.

Signed: ______________________________

Signed: ______________________________

Signed: ______________________________

Signed: ______________________________

Signed: ______________________________

Signed: ______________________________

Date: Tuesday, 18 August 2020.

ABSTRACT

The idea of our project is based on helping the blind person to know the things and objects that are around him through glasses with a camera on it, the person wears the glasses and the camera capture the frames and analyze them and extract the objects from them, then the result of analyzing the images is transformed into sound via an earphone tells him the name of the thing in front of him.

- The application also recognizes the names of people familiar to him.

✓ The image is analyzed and processed by applying various filters on it to extract the objects from it.

✓ The application is trained on a dataset for many objects that may be exist in front of the blind person such as chairs, tables, fruits, cars, buses, etc., and also people familiar to him such as his relatives or friends

✓ CNN is applied to identify objects by their color and shape

✓ we use SIFT algorithm for matching similar faces for better image matching and also to create a descriptor that was robust to the variations corresponding to typical viewing conditions. Also, to detect distinctive, invariant image feature points, which easily can be matched between images to perform tasks such as object detection and recognition, or to compute geometrical transformations between images

✓ We use tensor flow framework to link between Android and Python

❖ the object name that converts to sound to the blind person

❖ the familiar name that converts to the sound to the blind person

TABLE OF CONTENTS

Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

Table 1: abbreviations

Abbreviation	Full Meaning
TF	Tensor flow
MS COCO	Common Objects in Context
API	Application Programming Interface
TTS	Text To Speech

Introduction

Chapter One

1 INTRODUCTION

On an approximation ,285 million people are visually impaired across the globe, among which 39 million are blind, and 246 have low vision according to WHO statistics of 2011. About 90% of the world's visually impaired live in low-income settings, whereas 82% of people living with blindness are aged 50 and above. Could you imagine how the life of a person who is blind could be? Blind people are usually dependent on assistance from others. The assistance is from human beings, dogs, or some special electronic devices. The increasing number of people with disabilities in the world attracts the concern of researchers to invent various technologies, hoping that these technologies can assist the disabled people in carrying out their tasks in everyday life like normal people. So, we want to make something for them that would help them become independent. Therefore, the solution that has been portrayed in this paper is cost-effective, reliable, robust, and portable Smart glasses, which would help a blind person to walk on the streets almost like any other pedestrian. This smart glass can assist them while walking alone in new environments. by taking inputs through a camera and providing feedback to the person through headphones. So, people blind can be trained to visualize objects.

1.1 OVERVIEW

The aims of the project of Blind assistance is promoting a wide challenge in computer vision, such as recognition of objects of the surrounding objects practiced by the blind on a daily basis. The camera placed on blind person's glasses, MS COCO is large-scale object detection, segmentation, is employed to provide the necessary information about the surrounding environment. Objects and faces detection are used to find objects and faces in the real world from an image of the world such as father, mother, bicycles, chairs, doors, or tables that are common in the scenes of a blind. Based on their locations, and the camera is used to detect any objects and familiar faces. The proposed method for the blind aims at expanding possibilities to people with vision loss to achieve their full potential. The main object of the project is to design and implement a real time object and face recognition using blind glass.

1.2 PROJECT MOTIVATION

In our lives, there are many people who are suffering from different diseases or handicaps. According to WHO 2019, approximately 1 million people blind and 3 million visually impaired. Percentage of persons with disabilities in Egyptian society. Especially those with visual disabilities (blind), which is estimated [60 %] It is not simple. From here the idea of our project begins where it aims. The aims of the project of Blind assistance is promoting a wide challenge in computer vision, such as recognition of objects of the surrounding objects practiced by the blind on a daily basis.

1.3 PROJECT AIMS

The main objective of the project is to design and implement a smart glass for blind people using a special mini camera.

- Programing the camera Wi-Fi using the Android language, its powerful for processing.
- Process and analyze the camera records using WebView in real-time.
- Detect and recognize objects and faces in front of the blind.
- Design and build a headphone system to notify the user about the recognized objects and faces using voice messages.

1.4 PROJECT IDEA AND IMPORTANCE

- This project is mainly aimed at helping people who are blind and who suffer from a total lack of vision.
- Due to the development of technology, we must be tapped to help blind people.
- Due to the large number of blind people in Egypt.
- The next future and the future of technology is to serve people and help them in life.

1.5 PROBLEM STATEMENT

Ideal:

Ideally, we want to make something for blind people that would help them to become independent so our smart glasses can assist them while walking alone in new environments and knowing objects and the familiar people to him without helping of anyone

Reality:

In our lives, there are many people who are suffering from different diseases or handicaps. According to WHO 2019, approximately 1 million people blind and 3 million visually impaired. Nearly 60% of the visually impaired in Egypt have cataracts. These people need some help to make their life more-comfortable and better without any help from anyone

, hoping that these technologies can assist blind people in carrying out their tasks in everyday life like normal ordinary.

Consequences:

the visually impaired people and blind cannot see the obstacles that face them so the solution of their problems using the camera, its holder can recognize the things around him as if it were seen cause using the Android we would tell them what they saw around them So he is free to move just like he sees because he will get to know the scenes and the objects around him.

literature review

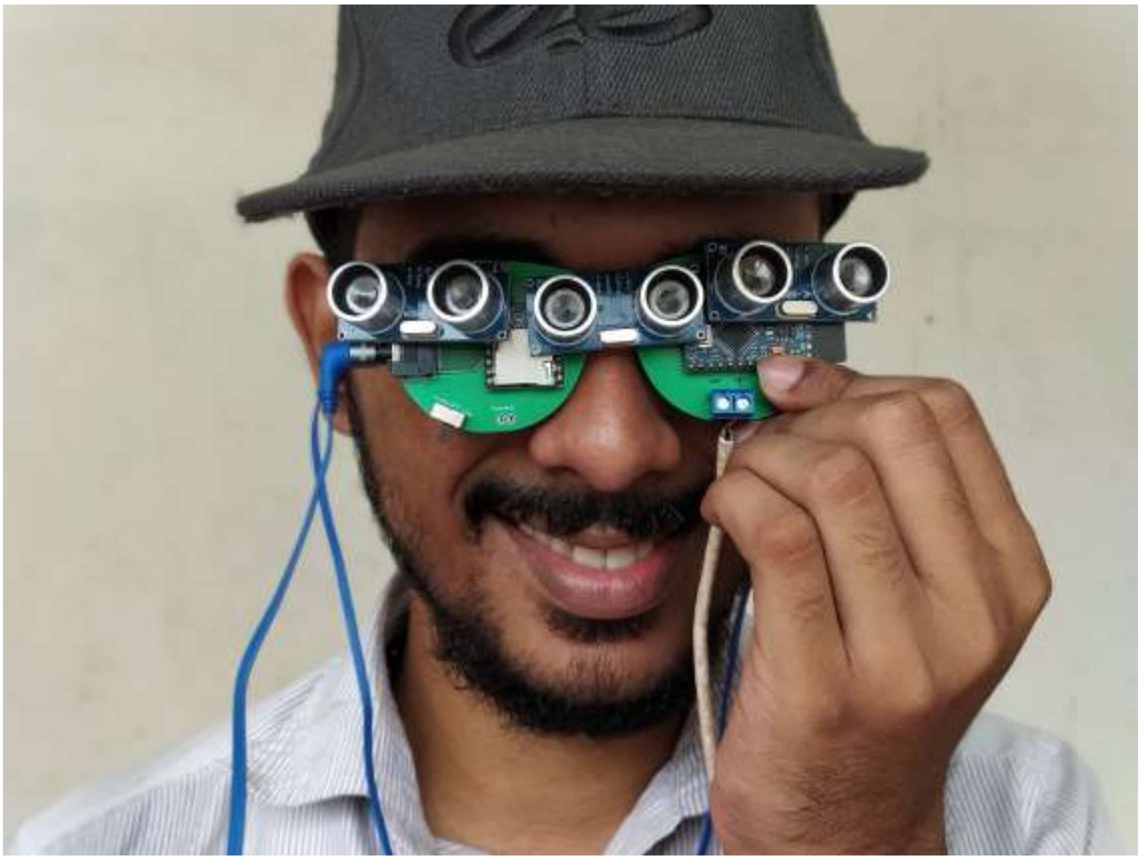
Chapter two

2 LITERATURE REVIEW

In this chapter, a review of past work and research papers related to the project is examined. The project looks at many areas, at first the needs and expectation of the visually impaired people are reviewed, followed by a discussion about some examples of the available and suggested aids and technologies to assist the visually impaired in their lives, after that a technical section that is important for the scope of the project is reviewed which is object recognition.

- Vocal vision for visually impaired [The International Journal of Issn: 2319 – 1813 Isbn: 2319 – 1805 Engineering and Science (Ijes)-01-07||2013|| Shrilekha Banger, Preetam Narkhede, Rajashree Parajape.] This project is a vision substitute system designed to assist blind people in autonomous navigation. Its working concept is based on 'image to sound' conversion. The vision sensor captures the image in front of the blind user. This image is then fed to MATLAB for processing. Process intuit processes the captured image and enhances the significant vision data. This processed image is then compared with the data-base kept in the microcontroller. The processed information is then presented as a structured form of the acoustic signal, and it is conveyed to the blind user using a set of ear phones. Color information from the interested objects evaluated to determine the color of the object. The colour output is informed to the blind user through headphones.

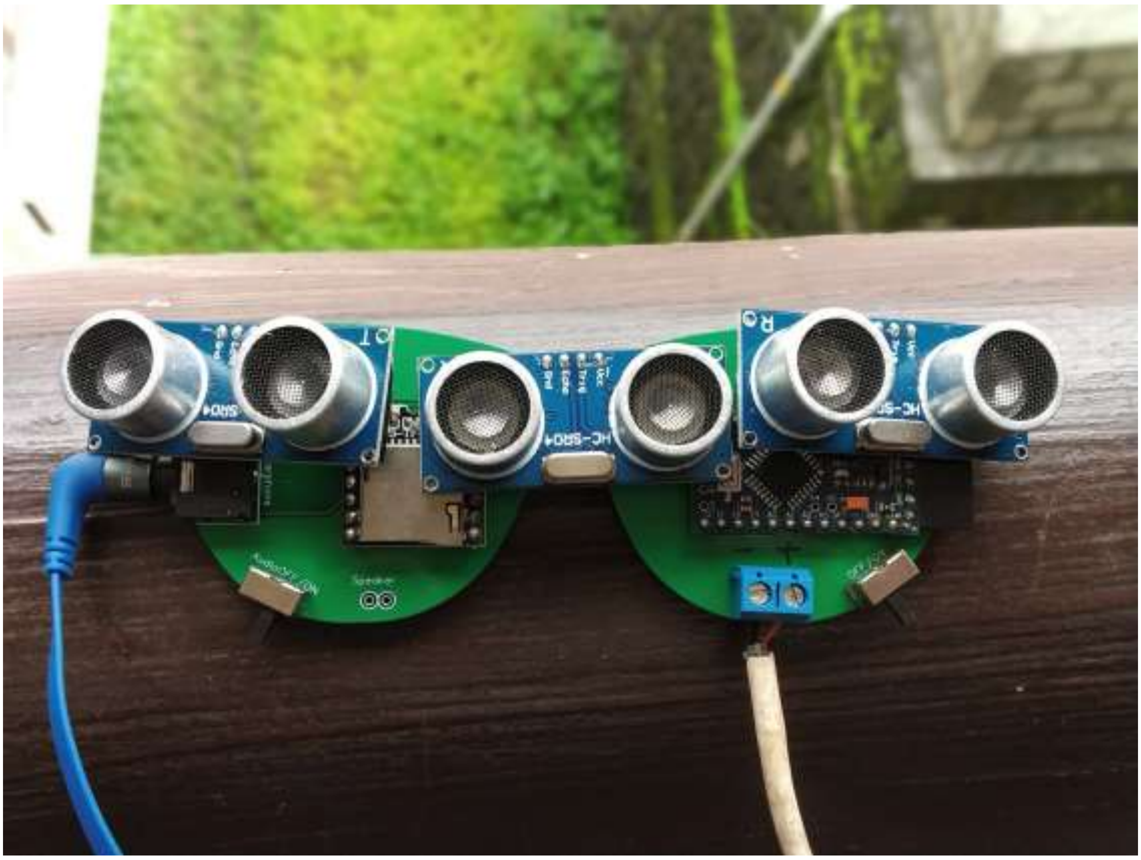
- **Talking Smart Glass for Blind** [1]



There are multiple smart accessories such as smart glasses, smart watches etc. available in the market. But all of them are built for us. There is a significant lack of technology to aid the physically challenged. I wanted to build something that is useful for the visually challenged people. So, I designed low cost smart glass that can be used to help the visually impaired.

This project uses few Ultrasonic distance sensors, an Arduino pro mini, mp3 Player module and some vibration motors.

Circuit board used in this project is designed in the form of a spectacle, that can be worn by visually impaired person. Arduino mounted on the spectacle will detect the obstacle with the help of the sensors and notify the user through headphones and vibration motors.



Parts needed

1. 1 * Arduino pro mini 5v 16Mhz (Any version of Arduino like nano or uno cane be used, but pro mini is easier to mount on the spectacles).
2. 3 * HC SR04 Ultrasonic distance sensor.
3. 1 * DF Player mini.
4. 3 * Vibration motors.
5. 3.5mm audio jack.
6. 2 * Slide switches.
7. 1 * SD card

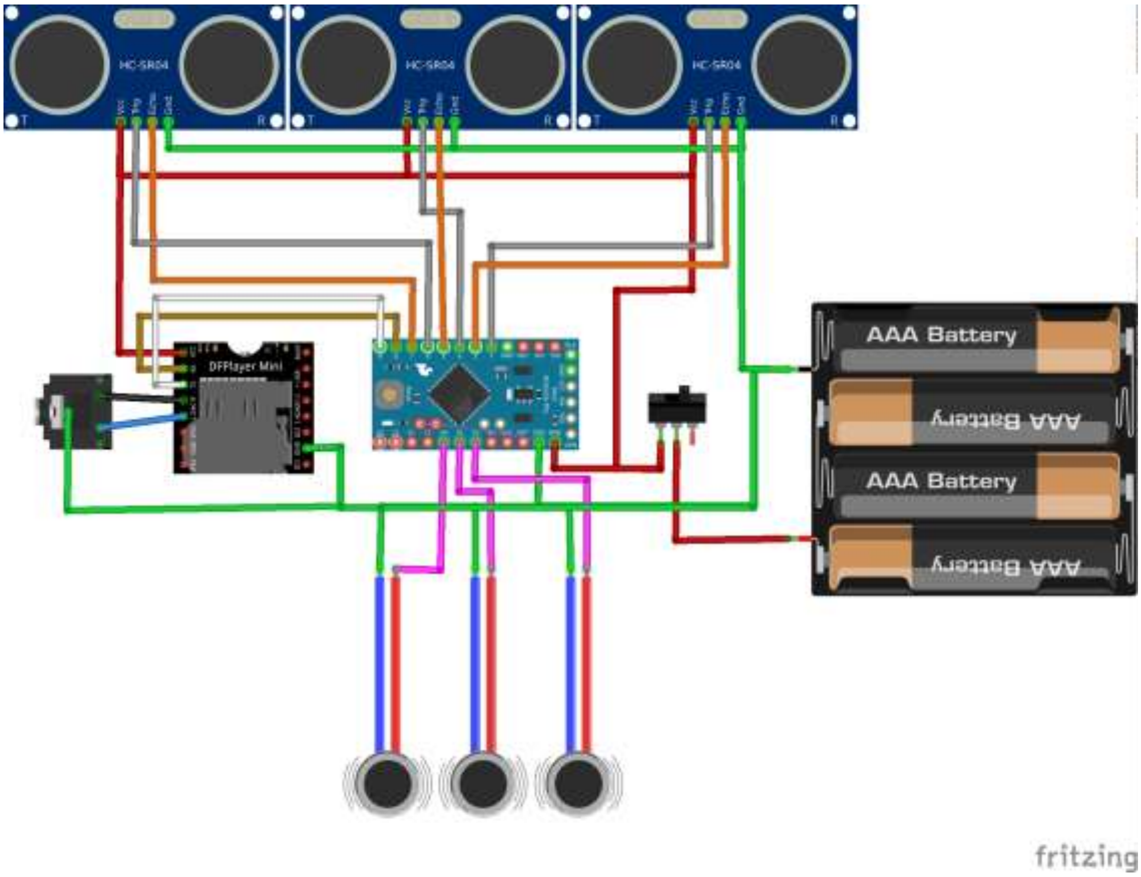
Optional

1. FTDI Breakout cable (only if you are using the pro mini)

2. My custom PCB

3. 3W Speaker

The Circuit Diagram



Proposed framework

3 PROPOSED FRAMEWORK

3.1 USE CASE DIAGRAM

The user has to create e-mail, and the login then checks the camera Wi-Fi and detects what is in the image if it is an object or person.

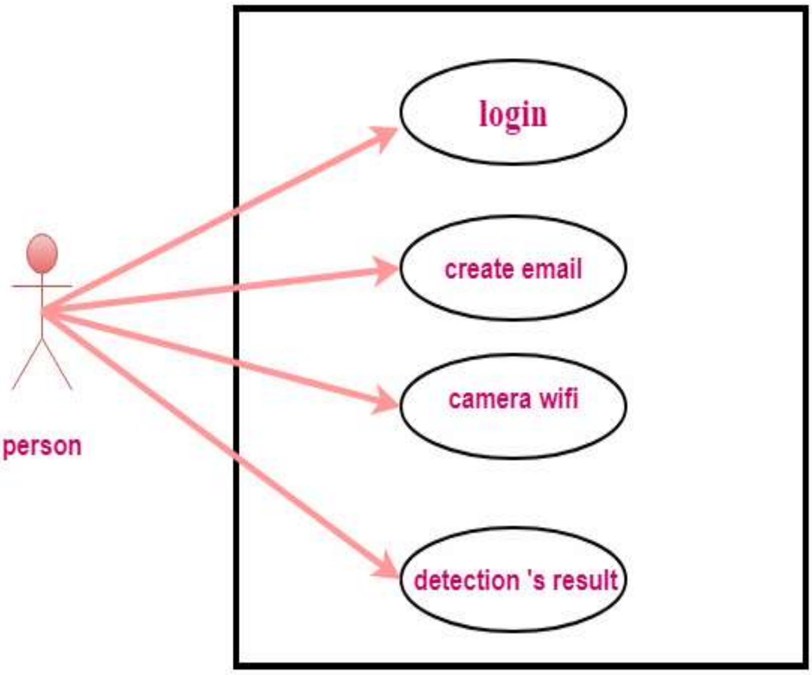


Figure 3- 1 : Use case diagram

3.2 ACTIVITY DIAGRAMS

The users have to make pre-configuration before using the application. The user must begin with the Login page. He must enter the E-mail and password to login to the page.

The user must have not an account consists of E-mail and password to login to the application .the user have to register to create an account .In the registration page, the user have to enter the E-mail and password and confirm password then press on Register button to save data in database SQL to facilitate the login page .

The user has to open the camera and check that camera is opened, and the Wi-Fi of the camera is connected to the smartphone.

After getting an image from the camera and knowing what in the image if it is an object or person and if the person is familiar or not. If the person is familiar, receives the name and type of relation, then converts that to speech. If what in the image is object, send it to the model, then gets the result from the model and converts to speech.

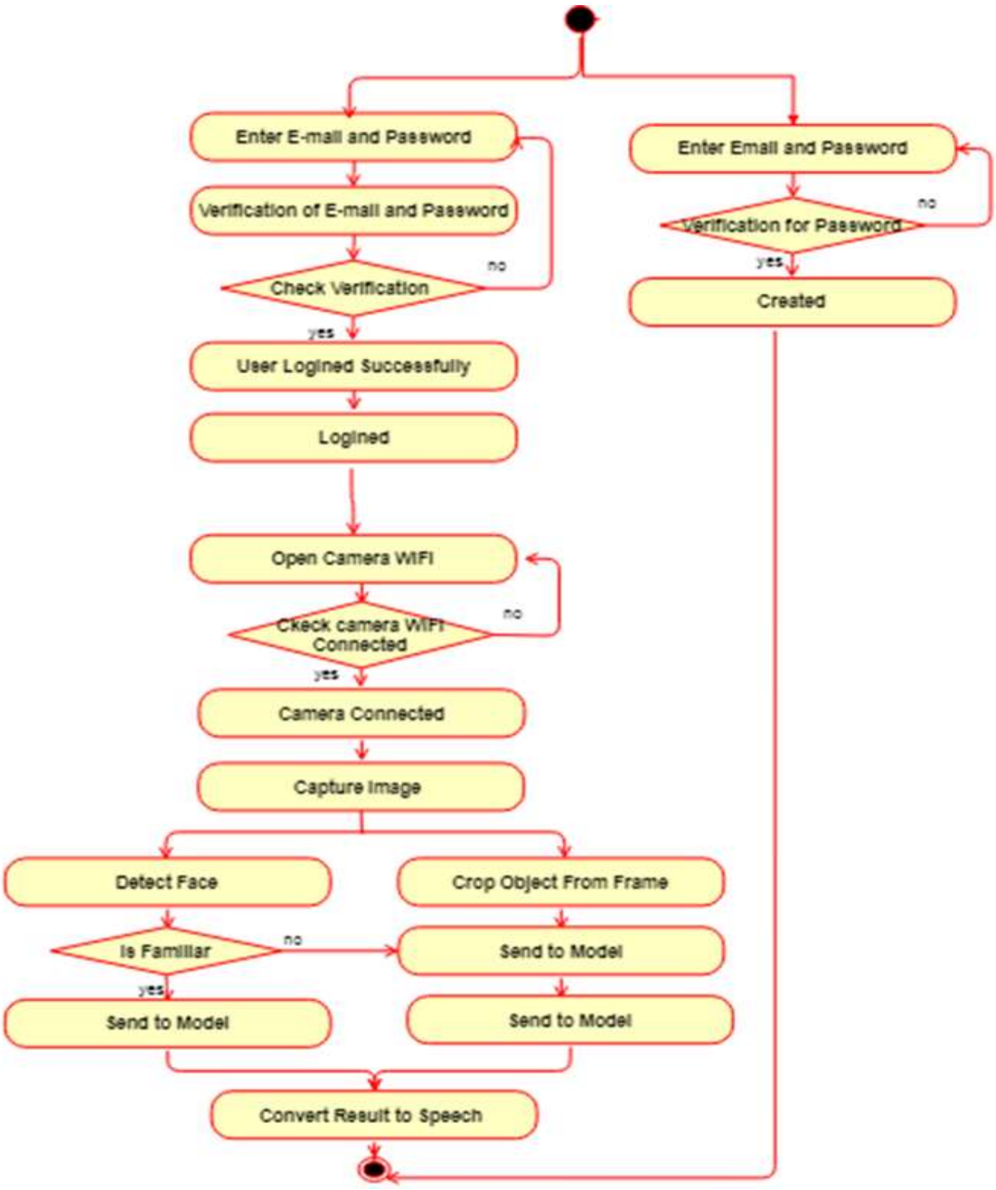


Figure 3- 2: activity diagram

3.3 ERD DIAGRAM

The user has an ID, E-mail, and password and the user have to enter the relatives name, image and Type of relation to the user.

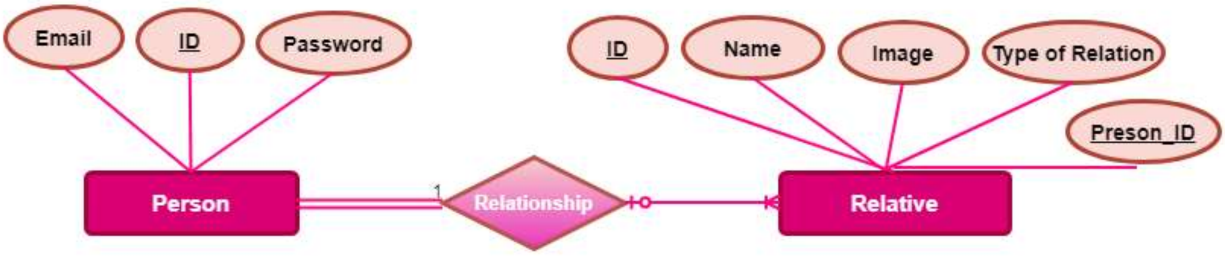


Figure 3- 3 : ERD diagram

3.4 CLASS DIAGRAM

This diagram shows how the function used to set and get from the database and the relation between the user and relative person is 1:M that the user may have more than one person.

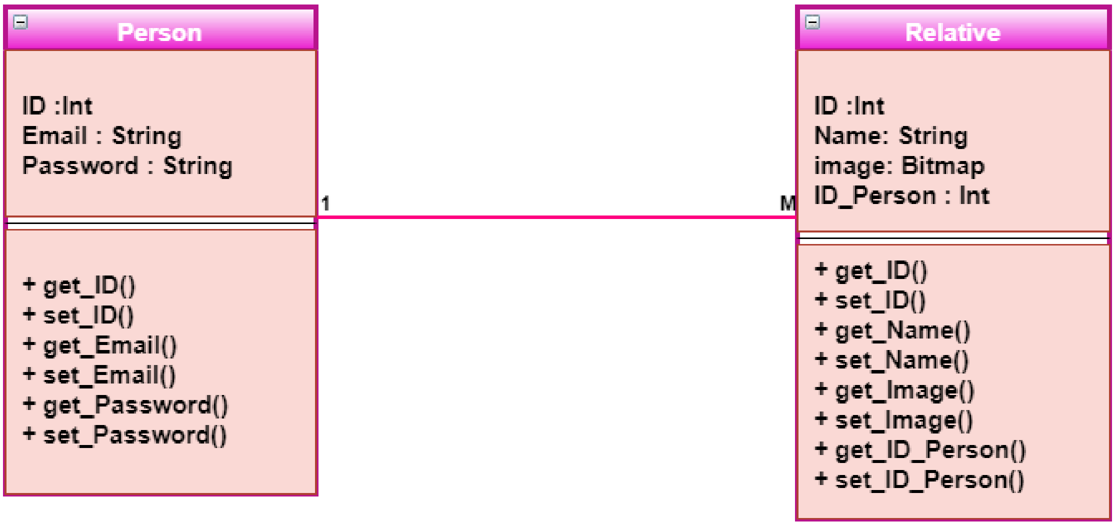


Figure 3- 4 : class diagram

3.5 DFD

Data flow diagram shows how data goes from processes to another and describe what happened in the system

3.5.1 Level 0

In the first:

The person must register to have an account. The user has to make a login. The user has to enter the familiar people and the system sends the speech about objects and faces.

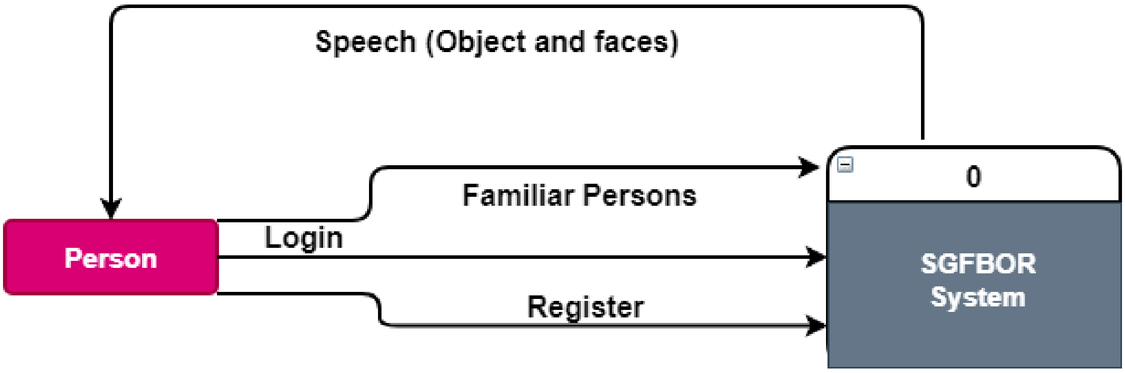


Figure 3- 5 :DFD diagram (level 0)

3.5.2 Level 1

In the second:

The user must enter the data to make registration and save data in the person database. The user must enter the data to login and get the data from database that search for it to find it to login ,then open the camera and take frames the detect what in the image is object or faces in the image if in the image is object converts it to speech. If the image is facing. If he is familiar or not. It must be check familiar by search in the relative database then converts it to speech.

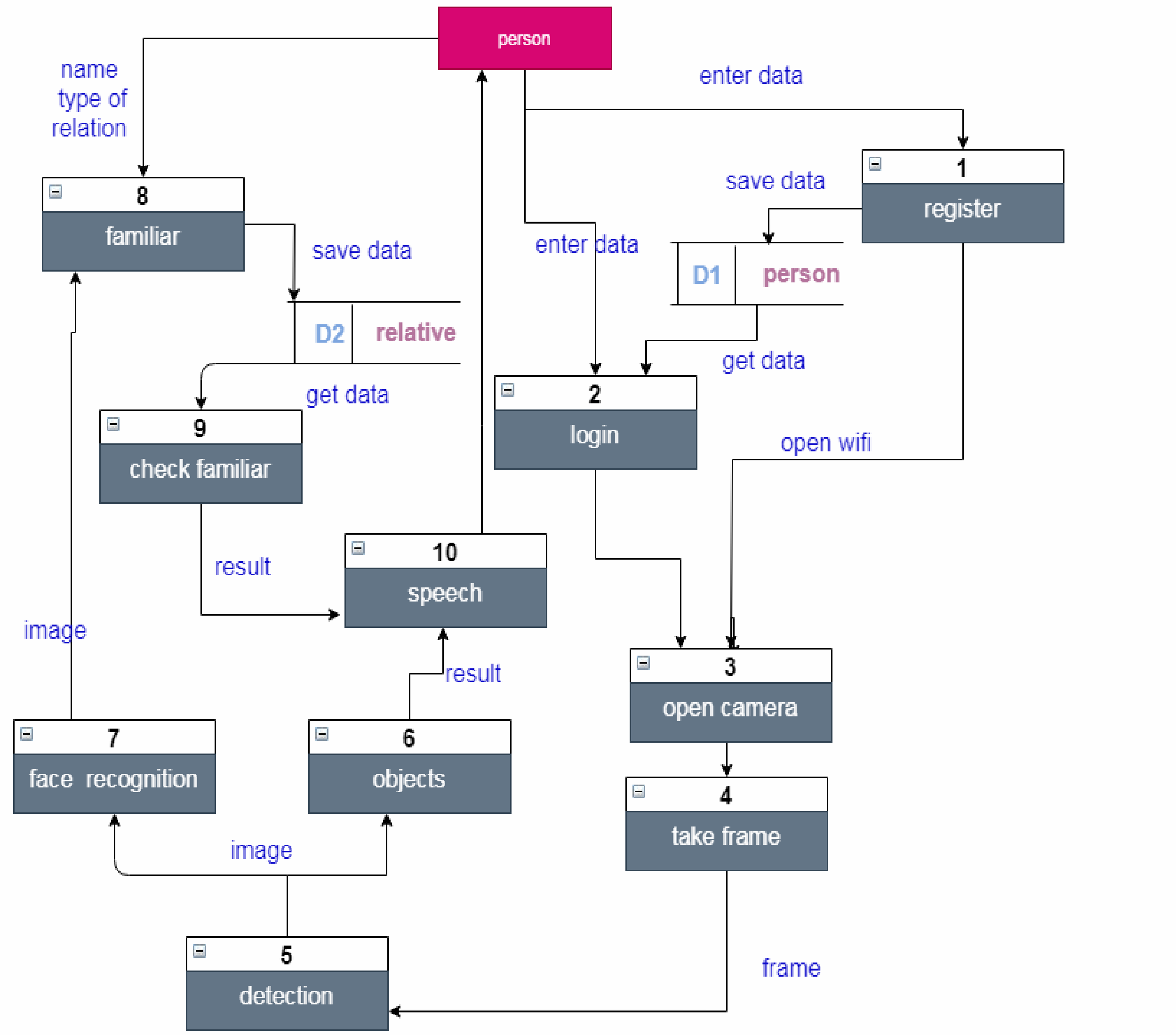


Figure 3- 6 :DFD diagram (level 1)

3.6 SEQUENCE

The user sends e-mail and password to the system, and the system must respond. If the camera opened, it sends frames. If the camera is not opened close to the camera. If the relative is familiar or object converts it to speech

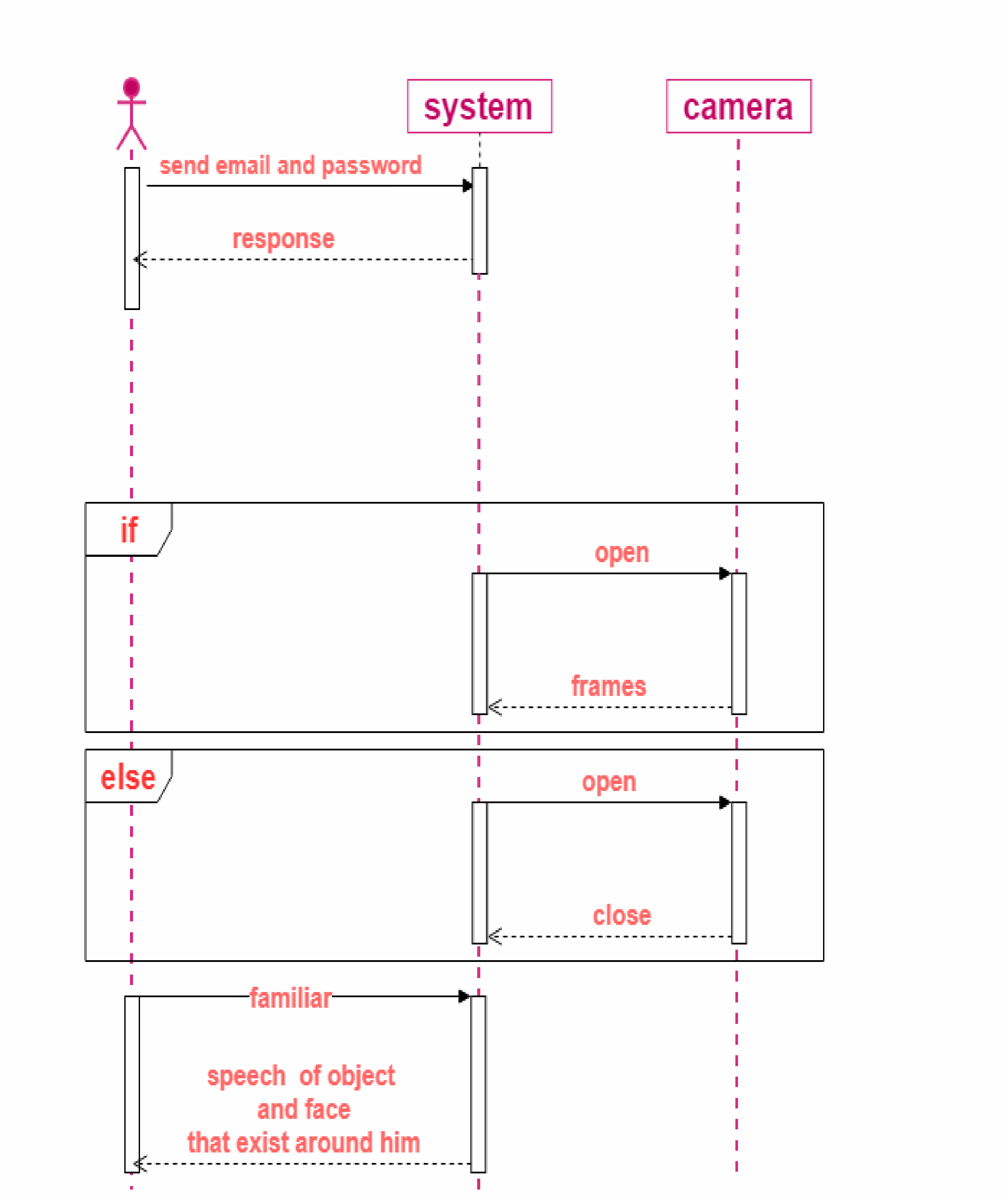


Figure 3- 7 :sequence diagram

Result and Discussion

Chapter four

4 RESULT AND DISCUSSION

4.1 CAMERA

This Mini Spy Camera is smaller than a thumbnail with a high enough resolution to see people, or ghosts, or whatever it is you're looking for. It's fairly high resolution (1080p video and 1920*1080 photo) module, with a driver board that is about 1 square inch in size, with a microSD card holder.

The image module is about the size of a cell phone camera - the body being just 6.2mm x 6.2mm and has a stick-on back so it's easy to mount in a doorbell or behind a teddy bear's eye (might as well be creative). There's a power LED and an 'activity' LED that lets you know what it's doing.

We don't have a real datasheet; we think these are just 'guts' of some low-cost keychain camera.

However, it's not too hard to use. Connect the red wire to 3.7V-5VDC and the black wire to ground.

Then use the white wire to control the camera. When touched to ground for under half a second, the camera will take a photo and stick it in a PHOTO folder on the SD card, numbered from PIC000.jpg up to PIC999.jpg. The red LED will blink briefly. When the trigger wire is touched for over a half second, it will start taking a video clip with audio from the mini microphone. The LED will stay on during the entire recording. Touch it again for half a second+ to stop.

What we like about this module is its simplicity and good quality. Best of all, there's no overlay text like most other cheap modules. It's really easy to make a micro-time-lapse device (see the video above for a time-lapse we made with a simple

Arduino pulsing the trigger pin every 30 seconds), or make little videos from the point of view of your robot! Note that there is a microphone but the audio quality is a little buzzy, it works but we're not going to replace our goPro or cellphone camera here.

In terms of still images, the camera is capable of 1920*1080-pixel static images, and 640*480 video. A microSD slot on the board allows video recording to either a microSD or microSDHC memory card. Comes with a USB cable for power and data transfer (plug it in and it will show up as a disk drive).

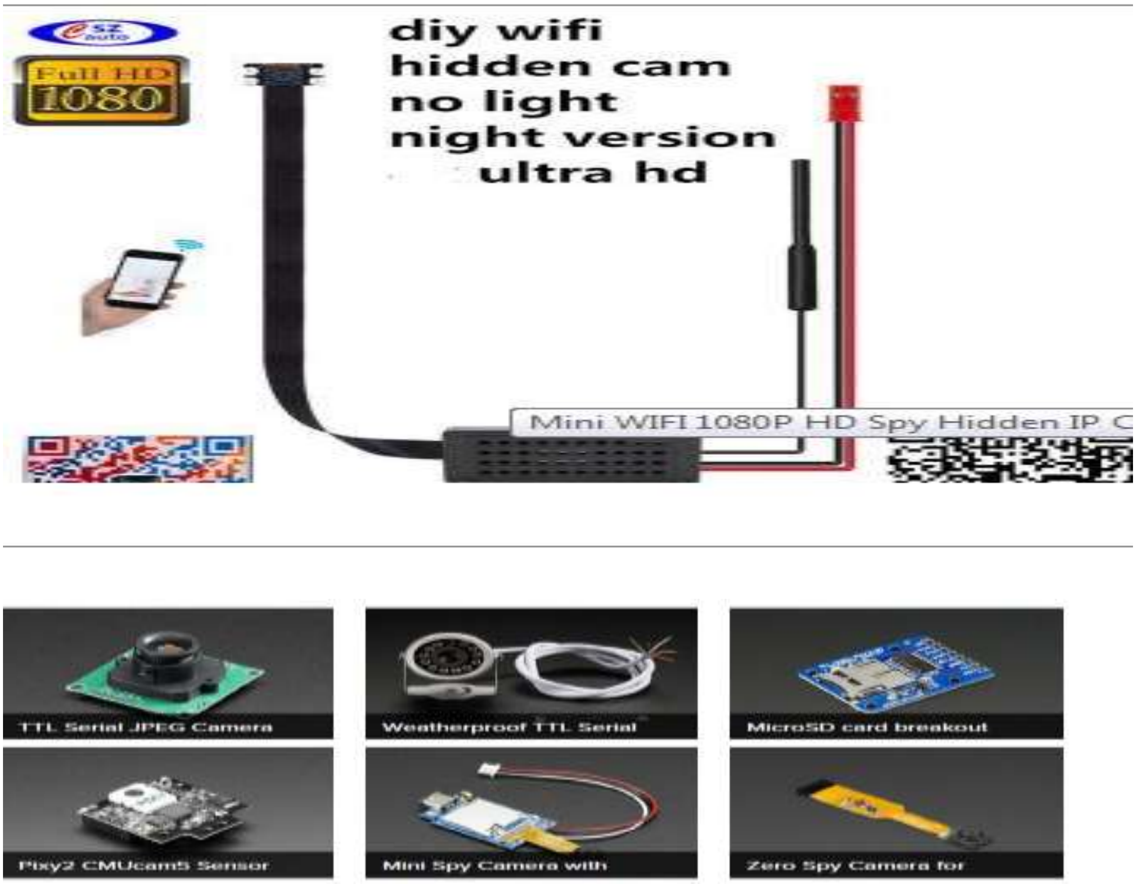


Figure 4- 1:Camera WIFI

Table 4- 1 :Camera specifications

Pixel	**1200 Mega CMOS**
Video Format	**ASF**
Compressed Format	**H.264**
Frame Number 25 **Visual Angle**	**140 degree**
Motion Detection Camera Shooting	**Straight line 6 meters**
Video Duration	**Over 4 Hours**
Battery Capacity	**2200MA**
Web browser	**IE7 and above, chrome, Firefox safari. Etc**

4.2 ANDROID

Our android application starts with an image that shows the application open then wait 4 seconds for WIFI to open and try to connect with our camera WIFI if it open then our application goes to pre-configuration:

- ✓ login page that asks the user the e-mail and password that he registered on our application if he has e-mail, he will go to the WebView page that shows the frame of the camera and does some processing on this frame like detect objects, faces by using OpenCV Library to do it like using:

- sift[2] (is a feature detection algorithm in computer vision to detect and describe local features in images. SIFT key points of objects are first extracted from a set of reference images[1] and stored in a database. An object is recognized in a new image by individually comparing each feature from the new image to this database and finding candidate matching features based on Euclidean distance of their feature vectors. From the full set of matches, subsets of key points that agree on the object and its location, scale, and orientation in the new image are identified to filter out good matches.)

- Gaussian filter.
- Canny [4] (Canny edge detection is a multi-step algorithm that can detect edges with noise suppressed at the same time.

1. Smooth the image with a Gaussian filter to reduce noise and unwanted details and textures.

$$g\ (m,\ n) = G\ (m,\ n) * f\ (m,\ n)$$

where

$$G = \frac{1}{\sqrt{2\pi\sigma}} e^{-(\frac{m^2+n^2}{2\sigma^2})}$$

2.Compute gradient of g (m, n) using any of the gradient operatiors (Roberts, Sobel, Prewitt, etc) to get:

$$M\ (n,\ n) = \sqrt{g^2(m,n) + g^2(m,n)}$$

and

$$\theta(m,n) = tan^{-1}[g_n(m,n)/g_m(m,n)]$$

3.Threshold M:

$$M_T(m,n) = \begin{cases} M(m,n) & \text{if } M(m,n) > T \\ 0 & \text{otherwise} \end{cases}$$

Where is so chosen that all edge elements are kept while most of the noise is suppressed.

4.Suppress non-maxima pixels in the edges in M_t obtained above to thin the edge ridges (as the edges might have been broadened in step 1). To do so, check to see whether each non-zero is greater than its two neighbors along the gradient direction $\theta(m,n)$. If so, keep M_t(m,n) unchanged, otherwise, set it to 0.

5.Threshold the previous result by two different thresholds T_1 and T_2 (where T_1<T_2) to obtain two binary images T_1 and T_2. Note that T_2 with greater T_2 has less noise and fewer false edges but greater gaps between edge segments, when compared to T_1 with smaller T_1.

6.Link edge segments in T_2 to form continuous edges. To do so, trace each segment in T_2 to its end and then search its neighbors in T_1 to find any edge segment in T_1 to bridge the gap until reaching another edge segment in T_2.)

then take these objects and faces to the TensorFlow(model) to predicted what is it and know the familiar persons by searching on database if the person is familiar to user or not and if the person is familiar getting his name and the type of relationship between them   then calculate the distance for each object and face then convert the prediction to sound by using text to speech(TTS)library in the language of the application that the user selected it (English or Arabic).

- ✓ Register page if he hasn’t e-mailed, he will go to register to make a new

  e-mail by put the information of him then he goes to the familiar page he put the image of the familiar person and then put his name and type of relation.

After finishing this step, he will go to WebView page automatically.

- ✓ The WebView represent the live video captured with the camera, show toast that contain what's image contain if it's an object or familiar person then the application make sound about this toast.

  But actual this process of what's image contain is very difficult because we using the algorithm of sift[2] to compare between two or more images (images that get from camera, and images that exists in data base) then if the image matches the image in database and ratio is greater than 90% that returns the result is the same person then returns the name of the person if not returns the gender of this person that appears in toast then converts to sound to the blind person to hear it , and using deep learning to know what objects exist in image then returns the name of objects that appears in toast then converts to sound to the blind person to hear it.

We will take about deep leaning and get more information about it.

4.3 DEEP LEARNING

4.3.1 CONVOLUTIONAL neural network

A neural network in which at least one layer is a convolution filter. A typical convolutional neural network consists of some combination of the following layers:

- Convolution Layer
- Batch Normalization Layer
- Activation Layer
- MaxPooling2D Layer
- Flatten Layer
- Dropout Layer
- Dense Layer

A layer of a deep neural network in which a convolution filter passes along an input matrix. For example, consider the following 3x3 convolution filter.

Table 4- 2 : convolution filter

1	**0**	**-1**
2	0	-2
1	0	-1

The following two-step mathematical operation:

- Element-wise multiplication of the convolution filter and a slice of an input matrix

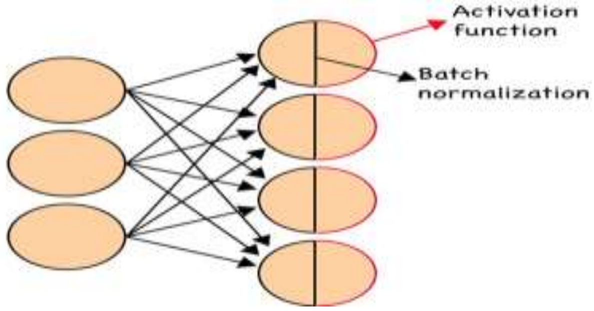
- Summation of all the values in the resulting product matrix.

A convolution layer [2] consists of a series of convolutional operations, each acting on a different slice of the input matrix. (The slice of the input matrix has the same rank and size as the convolutional filter.)

BATCH Normalization Layer

Normalizing the input or output of the activation functions in a hidden Layer. Batch normalization can provide the following benefits:

- Make a neural network more stable by protecting against outlier weights.
- Enable higher learning rates.
- Reduce overfitting.



$$\widehat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$
$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i)$$

Figure 4- 2 :Batch normalization

Activation function

A function [5] (for example REIU) that takes in the weighted sum of all of the inputs from the previous layer and then generates and passes an output value (typically nonlinear) to the next layer. In the input layer and hidden layer, we used the Relu Activation and in the output Layer we will used the SoftMax Activation

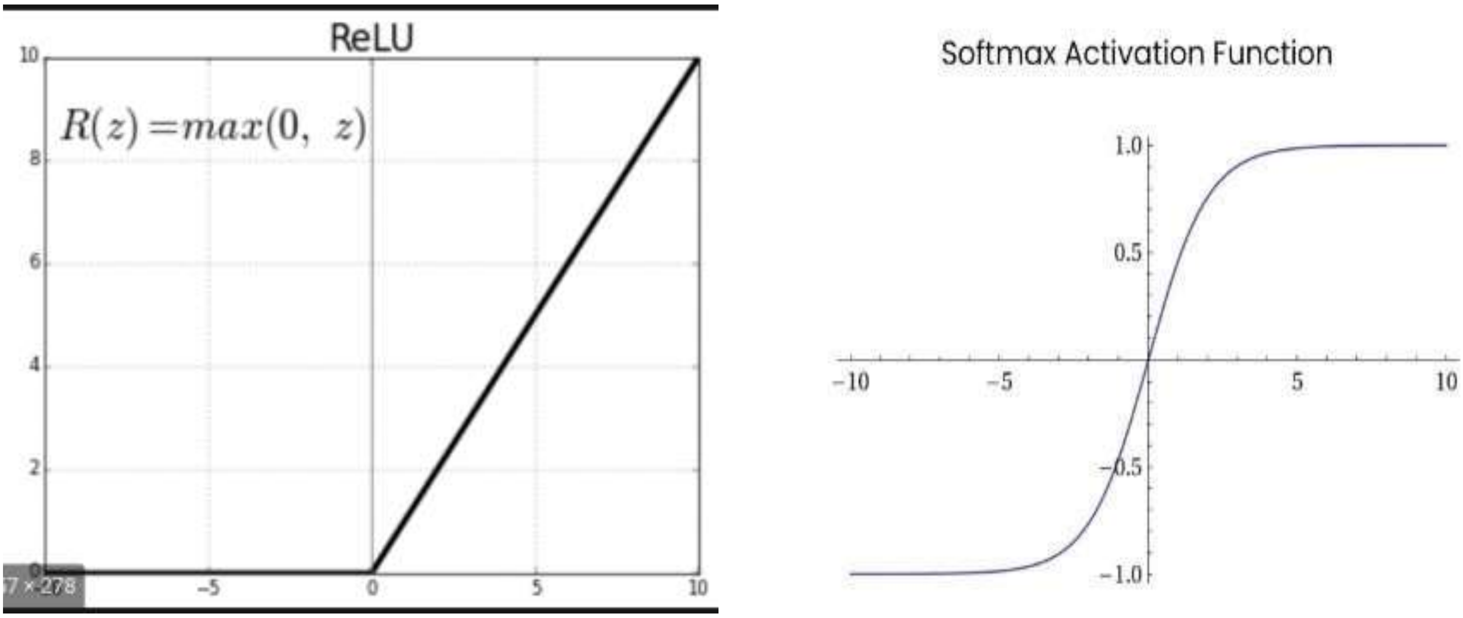


Figure 4- 3:Activation function types

<u>SoftMax</u> used in the Output Layer with Multi-Class

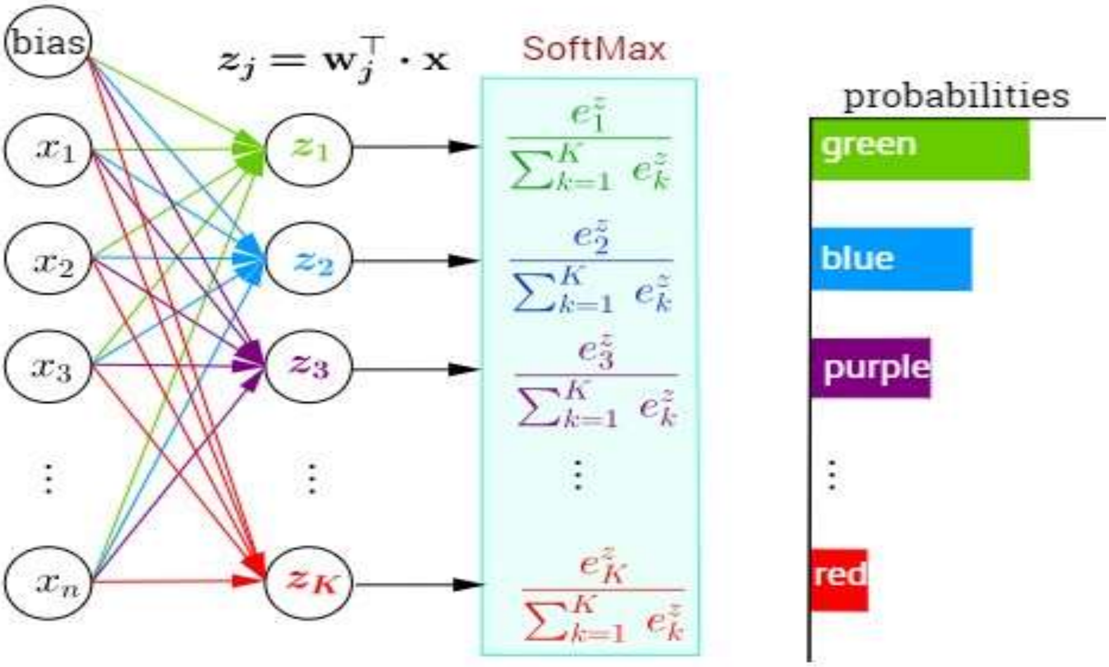

bias
x_1
x_2
x_3
⋮
x_n
$z_j = \mathbf{w}_j^\top \cdot \mathbf{x}$
z_1
z_2
z_3
⋮
z_K
SoftMax
$\frac{e^{z}_{1}}{\sum_{k=1}^{K} e^{z}_{k}}$
$\frac{e^{z}_{2}}{\sum_{k=1}^{K} e^{z}_{k}}$
$\frac{e^{z}_{3}}{\sum_{k=1}^{K} e^{z}_{k}}$
⋮
$\frac{e^{z}_{K}}{\sum_{k=1}^{K} e^{z}_{k}}$
probabilities
green
blue
purple
⋮
red

Figure 4- 4:SoftMax function

MaxPooling2D Layer

Reducing a matrix (or matrices) created by an earlier convolution layer to a smaller matrix. Pooling usually involves taking either the maximum or average value across the pooled area. For example, suppose we have the following 3x3 matrix.

Table 4- 3:Max Pooling filter

5	**3**	**1**
8	2	5
9	4	3

A pooling operation [3]: just like a convolutional operation, divides that matrix into slices and then slides that convolutional operation by **strides**. For example, suppose the pooling operation divides the convolutional matrix into 2x2 slices with a 1x1 stride. As the following diagram illustrates, four pooling operations take place. Imagine that each pooling operation picks the maximum value of the four in that slice:

MaxPooling2D helps enforce translational invariance in the input matrix.

Dropout Layer

Dropout [6] regularization works by removing a random selection of a fixed number of the units in a network layer for a single gradient step. The more units dropped out, the stronger the regularization. This is analogous to training the network to emulate an exponentially large ensemble of smaller networks. For full details see Dropout a simple way to Prevent Neural Networks from Overfitting.

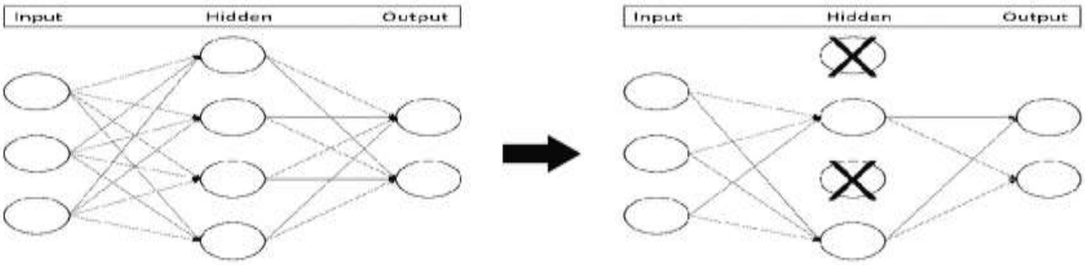


Figure 4- 5: Dropout layer

Flatten Layer [7]

• transforms a two-dimensional matrix of features into a vector before a fully connected neural network classifier.

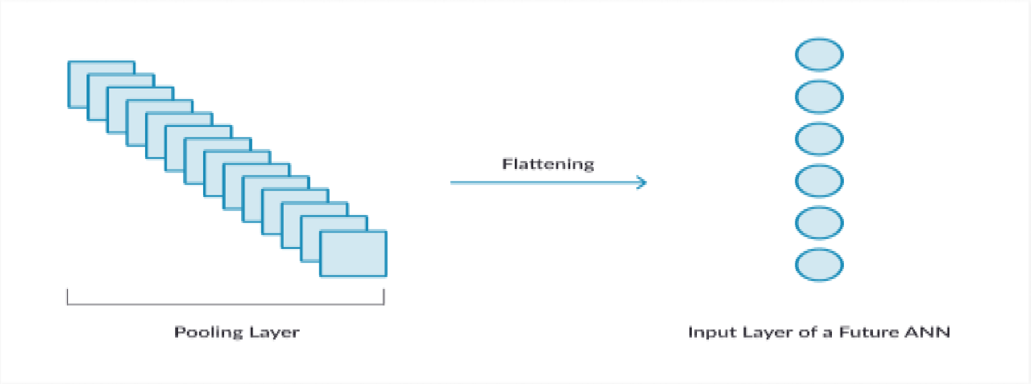


Figure 4- 6:Flatten layer

Example

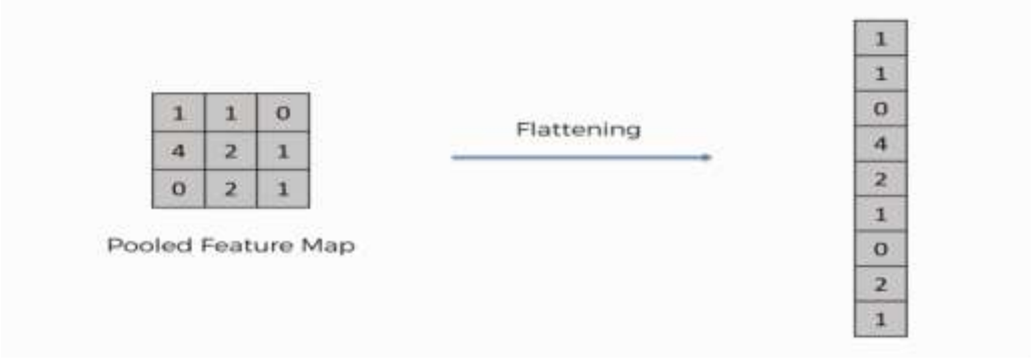


Figure 4- 7:Flatten layer (example)

Dense Layer

Synonym for fully connected layer

A hidden layer in which each node is connected to every node in the subsequent hidden layer.

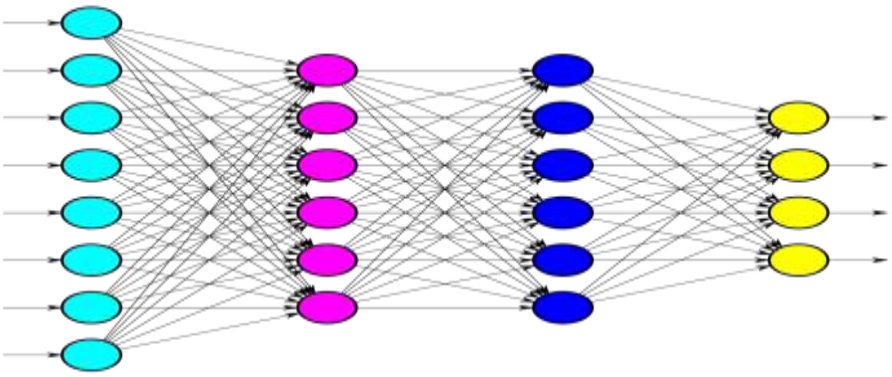


Figure 4- 8:Dense layer

After Build Layers

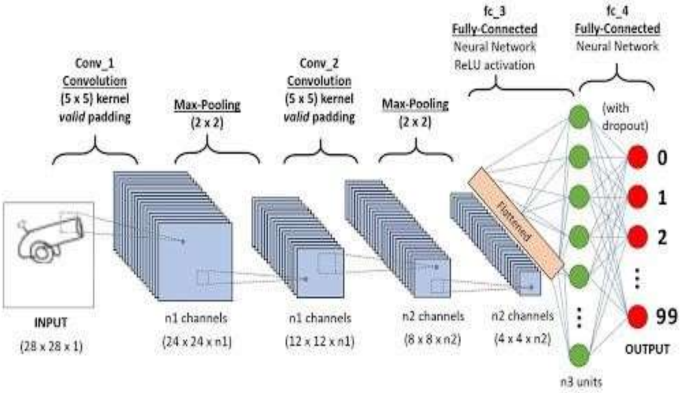


Figure 4- 9:CNN structure

4.3.2 Compiling the Model

compile the model you specified earlier. To compile the model, you need to specify the optimizer and loss function to use.

optimizer

- we will use Adam optimizer. Adam is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based on training data.

- reduce overfitting and make better predictions with deep learning models
- Different optimizers may leverage one or more of the following concepts to enhance the effectiveness of gradient descent on a given training set.

Gradient descent

- A technique to minimize loss by computing the gradients of loss with respect to the model's parameters conditioned on training data.
- Informally gradient descent iteratively adjusts parameters gradually finding the best combination of weights and bias to minimize loss.
- A gradient descent algorithm that uses minibatches. In other words, mini-batch SGD estimates the gradient-based on a small subset of the training data

Loss Function

A measure of how far a model's predictions are from its label. Or to phrase it more pessimistically a measure of how bad the model is. To determine this value a model must define a loss function. For example, linear regression models typically use mean squared error for a loss function while logistic regression models use log loss.

4.3.3 Fitting the Model

When we call the Model.fit_generator () function it makes assumptions:

➢Keras is first calling the generator function (data augmentation)

➢Generator function (data Augmentation) provides a batch size of 32 to us. fit generator () function.

➢our Model.fit_generator () function first accepts a batch of the dataset then performs backpropagation on it and then updates the weights in our model.

➢For the number of epochs specified (10 in our case) the process is repeated.

4.3.4 Project data set

We collected a dataset from The Kaggle website ,and it contains 24 object We divided the group data into two parts, which are training and testing, the training data includes 80 percent of the data, and the test is 20 percent. And the size of this data is equal to 1.95 GB , Some of the data is Specific to the clothes, And it includes the Five object and they are [black-jeans, blue-dress, blue-shirt, red-dress, red-shirt] , data about transports and they are [Car ,Bus]And data on fruits , and includes The four object [banana, lemon, orange, tomato] And things used in the kitchen And they are[Knife, Fork, Glass , bottle] And data to detect difference between gender [men, women] And data for some furniture [Table, Sofa, Chair] My data includes four object that make up some computer components and Phone [Mouse, Keyboard, Laptop Screen , Smart Phone]. The total number of images used in the data set for all items equals 23500 images.

4.4 INTEGRATION

We integration android application with hardware and software.

4.4.1 integration with hardware

We use camera Wi-Fi that it 's view angle 90 degree and video frame rate is 25fps that we connect the android application with it by using the Wi-Fi of the camera using the IP address (192.168.10.1) of it and getting frames by WebView in android studio .

4.4.2 integration with python

We use the TensorFlow (that is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks.[4] It is used for both research and production at Google.) , Specially TensorFlow Lite (is the mobile GPU inference engine with OpenGL ES 3.1 Compute Shaders on Android devices and Metal Compute Shaders on iOS devices) to integrate the android application with deep learning application .

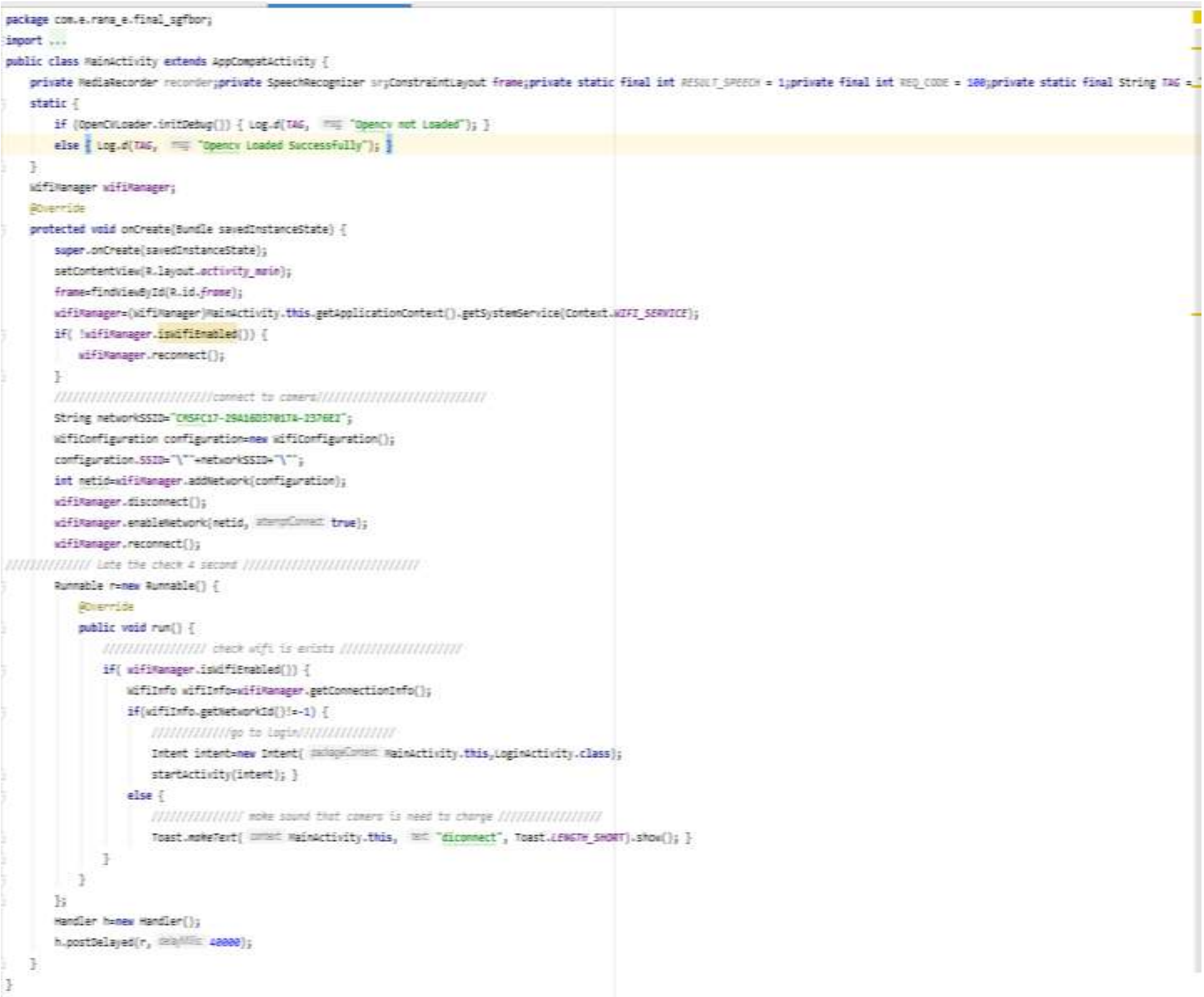
Appendix

5 APPENDIX

5.1 ANDROID

Main activity

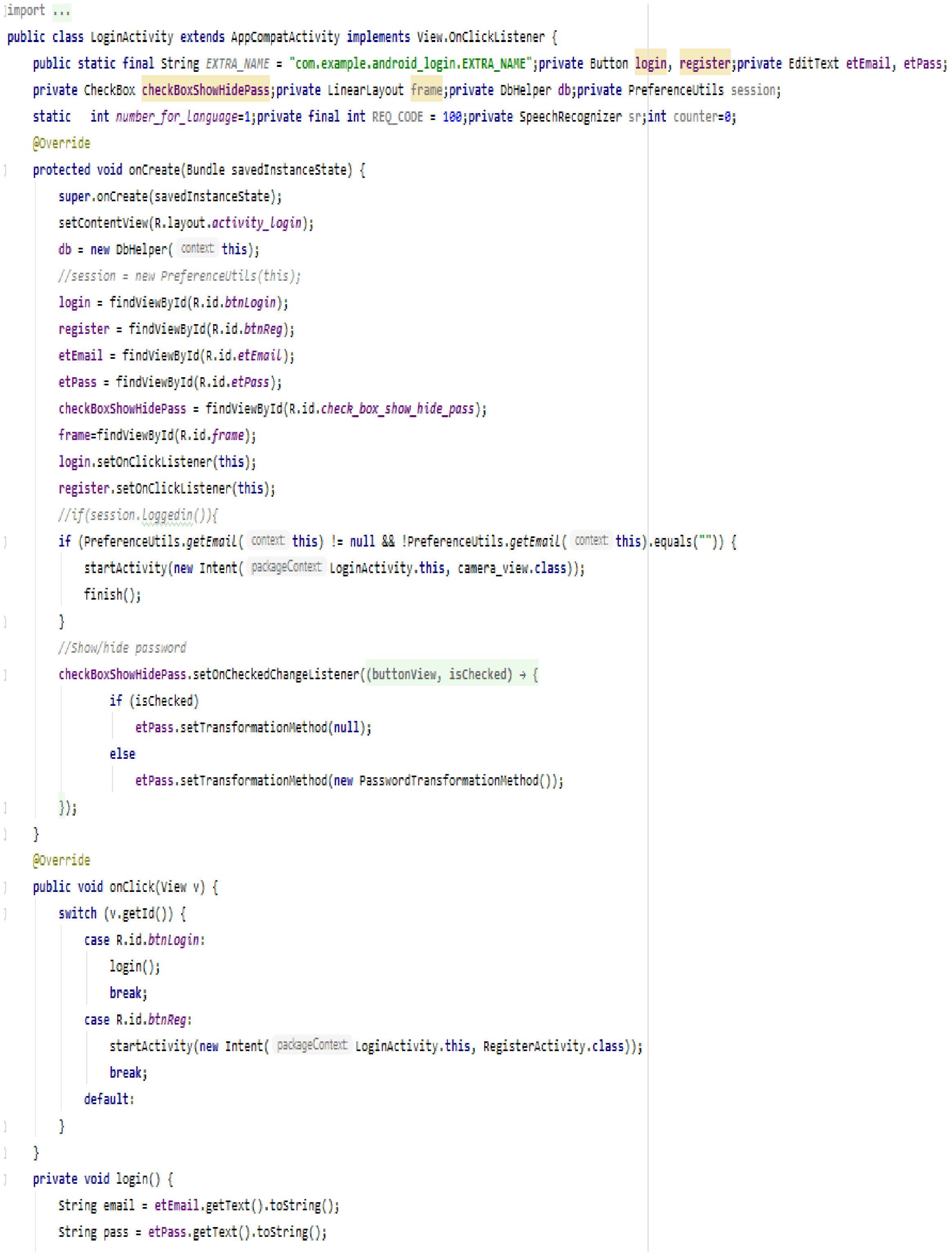
In this activity is the code of home page that have the declaration of Wi-Fi that is the way we connect camera by using Wi-Fi manger in android studio that gave us the ability to connect it's SSID , and check if it connect then go to login activity if not show toast , and wait 4s and reload the request again.



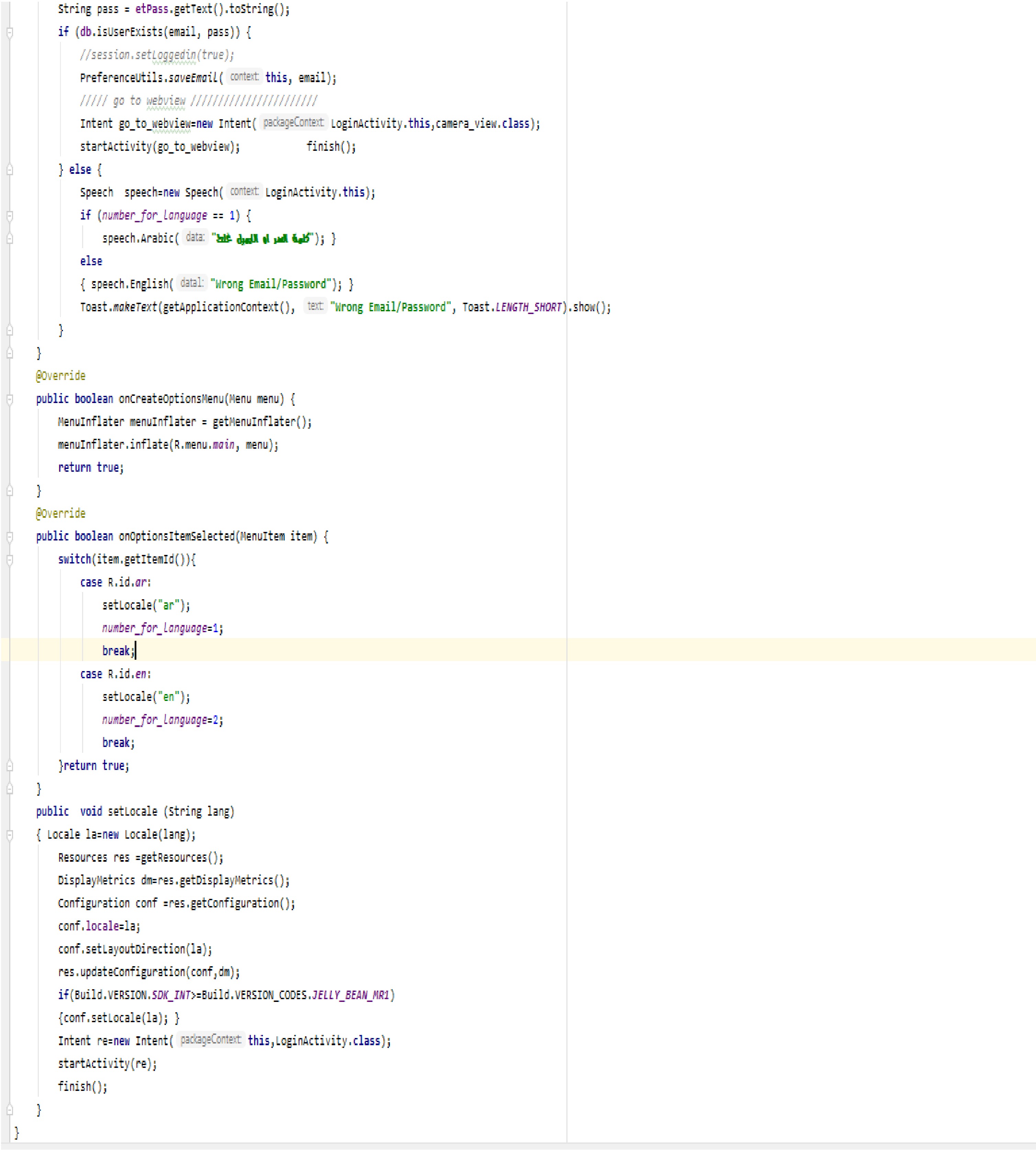
```
package com.e.rana_e.final_sgfbor;
import ...
public class MainActivity extends AppCompatActivity {
    private MediaRecorder recorder;private SpeechRecognizer sr;ConstraintLayout frame;private static final int RESULT_SPEECH = 1;private final int REQ_CODE = 100;private static final String TAG =_
    static {
        if (OpenCVLoader.initDebug()) { Log.d(TAG,  msg: "Opencv not Loaded"); }
        else { Log.d(TAG,  msg: "Opencv Loaded Successfully"); }
    }
    WifiManager wifiManager;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        frame=findViewById(R.id.frame);
        wifiManager=(WifiManager)MainActivity.this.getApplicationContext().getSystemService(Context.WIFI_SERVICE);
        if( !wifiManager.isWifiEnabled()) {
            wifiManager.reconnect();
        }
        ///////////////////////////////connect to camera//////////////////////////////
        String networkSSID="CMSFC17-29A1603701 7A-2376EI";
        WifiConfiguration configuration=new WifiConfiguration();
        configuration.SSID="\""+networkSSID+"\"";
        int netid=wifiManager.addNetwork(configuration);
        wifiManager.disconnect();
        wifiManager.enableNetwork(netid,  attemptConnect: true);
        wifiManager.reconnect();
/////////////// late the check 4 second /////////////////////////////
        Runnable r=new Runnable() {
            @Override
            public void run() {
                ////////////////// check wifi is exists ///////////////////
                if( wifiManager.isWifiEnabled()) {
                    WifiInfo wifiInfo=wifiManager.getConnectionInfo();
                    if(wifiInfo.getNetworkId()!=-1) {
                        /////////////go to login////////////////
                        Intent intent=new Intent( packageContext: MainActivity.this,LoginActivity.class);
                        startActivity(intent); }
                    else {
                        ////////////// make sound that camera is need to charge /////////////////
                        Toast.makeText( context: MainActivity.this,  text: "diconnect", Toast.LENGTH_SHORT).show(); }
                }
            }
        };
        Handler h=new Handler();
        h.postDelayed(r,  delayMillis: 40000);
    }
}
```

Login activity

In this activity is the code of login page that have the declaration of Text Views and Edit Text to enter the email and password and checkbox to hide or show the password and save this data in the database and choose the language that the text appear in it if the language is Arabic or English.



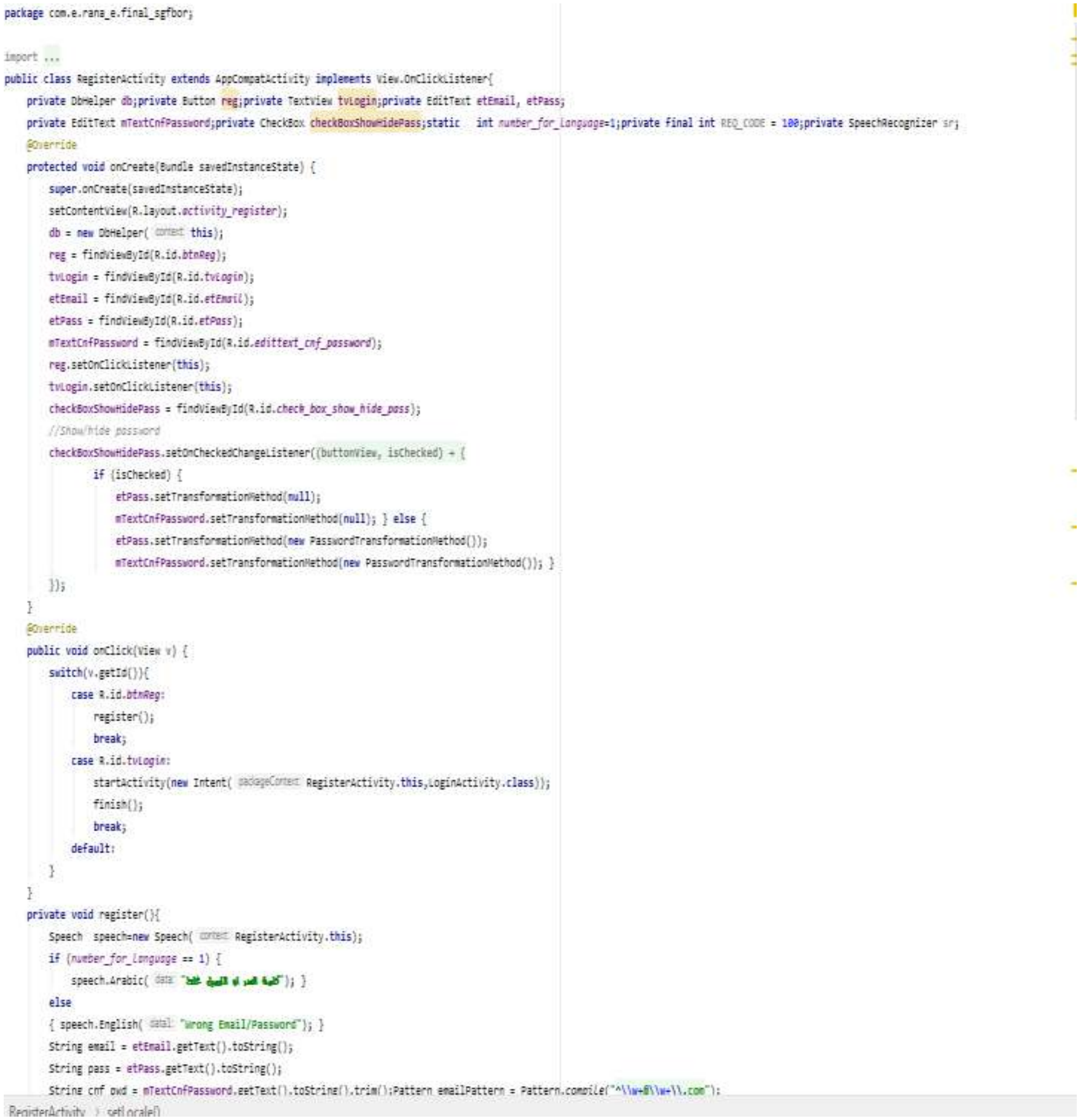
```
import ...
public class LoginActivity extends AppCompatActivity implements View.OnClickListener {
    public static final String EXTRA_NAME = "com.example.android_login.EXTRA_NAME";private Button login, register;private EditText etEmail, etPass;
    private CheckBox checkBoxShowHidePass;private LinearLayout frame;private DbHelper db;private PreferenceUtils session;
    static  int number_for_language=1;private final int REQ_CODE = 100;private SpeechRecognizer sr;int counter=0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        db = new DbHelper( context: this);
        //session = new PreferenceUtils(this);
        login = findViewById(R.id.btnLogin);
        register = findViewById(R.id.btnReg);
        etEmail = findViewById(R.id.etEmail);
        etPass = findViewById(R.id.etPass);
        checkBoxShowHidePass = findViewById(R.id.check_box_show_hide_pass);
        frame=findViewById(R.id.frame);
        login.setOnClickListener(this);
        register.setOnClickListener(this);
        //if(session.loggedin()){
        if (PreferenceUtils.getEmail( context: this) != null && !PreferenceUtils.getEmail( context: this).equals("")) {
            startActivity(new Intent( packageContext: LoginActivity.this, camera_view.class));
            finish();
        }
        //Show/hide password
        checkBoxShowHidePass.setOnCheckedChangeListener((buttonView, isChecked) → {
                if (isChecked)
                    etPass.setTransformationMethod(null);
                else
                    etPass.setTransformationMethod(new PasswordTransformationMethod());
        });
    }
    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.btnLogin:
                login();
                break;
            case R.id.btnReg:
                startActivity(new Intent( packageContext: LoginActivity.this, RegisterActivity.class));
                break;
            default:
        }
    }
    private void login() {
        String email = etEmail.getText().toString();
        String pass = etPass.getText().toString();
```



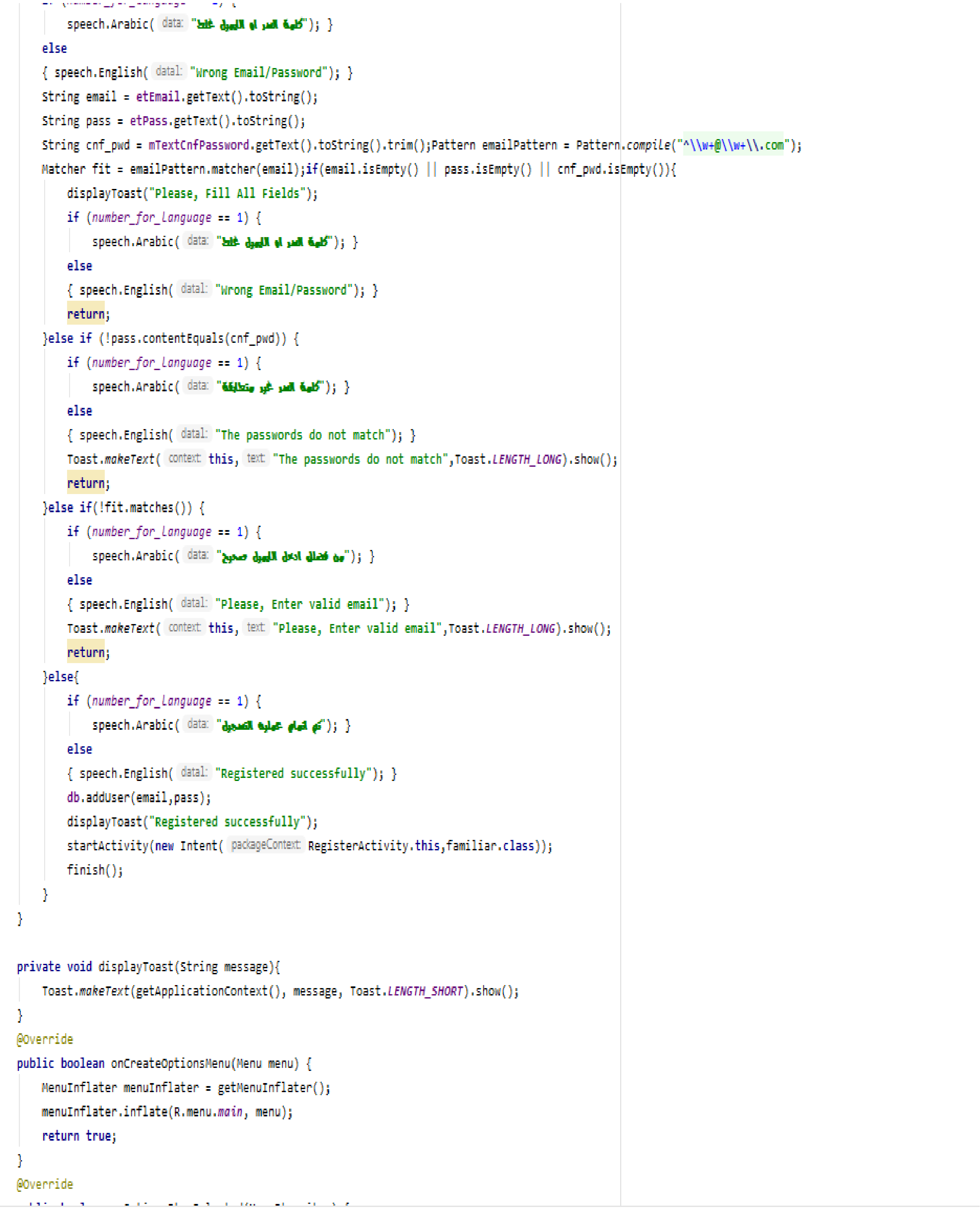
```
        String pass = etPass.getText().toString();
        if (db.isUserExists(email, pass)) {
            //session.setLoggedin(true);
            PreferenceUtils.saveEmail( context: this, email);
            ///// go to webview /////////////////////
            Intent go_to_webview=new Intent( packageContext: LoginActivity.this,camera_view.class);
            startActivity(go_to_webview);           finish();
        } else {
            Speech  speech=new Speech( context: LoginActivity.this);
            if (number_for_language == 1) {
                speech.Arabic( data: "كلمة السر او الايميل خطأ"); }
            else
            { speech.English( data1: "Wrong Email/Password"); }
            Toast.makeText(getApplicationContext(),  text: "Wrong Email/Password", Toast.LENGTH_SHORT).show();
        }
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater menuInflater = getMenuInflater();
        menuInflater.inflate(R.menu.main, menu);
        return true;
    }
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch(item.getItemId()){
            case R.id.ar:
                setLocale("ar");
                number_for_language=1;
                break;
            case R.id.en:
                setLocale("en");
                number_for_language=2;
                break;
        }return true;
    }
    public  void setLocale (String lang)
    { Locale la=new Locale(lang);
        Resources res =getResources();
        DisplayMetrics dm=res.getDisplayMetrics();
        Configuration conf =res.getConfiguration();
        conf.locale=la;
        conf.setLayoutDirection(la);
        res.updateConfiguration(conf,dm);
        if(Build.VERSION.SDK_INT>=Build.VERSION_CODES.JELLY_BEAN_MR1)
        {conf.setLocale(la); }
        Intent re=new Intent( packageContext: this,LoginActivity.class);
        startActivity(re);
        finish();
    }
}
```

Register activity

In this activity is the code of Registration page that have the declaration of Text Views and Edit Text to enter the email, password ,confirm password and checkbox to hide or show the password and save this data in the database by pressing on register button and choose the language that the text appear in it if the language is Arabic or English.



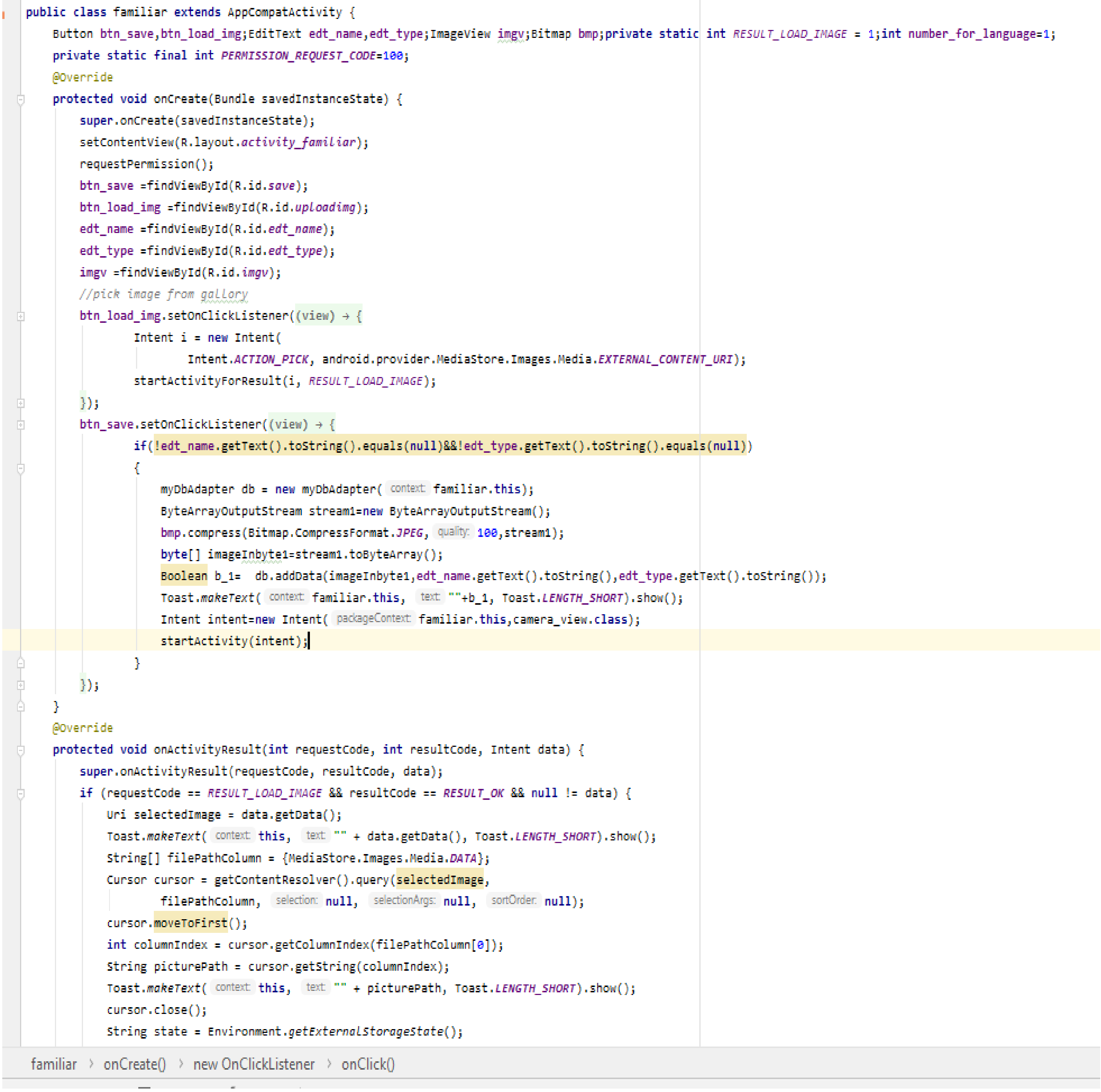
```
package com.e.rana_e.final_sgfbor;

import ...
public class RegisterActivity extends AppCompatActivity implements View.OnClickListener{
    private DbHelper db;private Button reg;private TextView tvLogin;private EditText etEmail, etPass;
    private EditText mTextCnfPassword;private CheckBox checkBoxShowHidePass;static   int number_for_longuage=1;private final int REQ_CODE = 100;private SpeechRecognizer sr;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);
        db = new DbHelper( context: this);
        reg = findViewById(R.id.btnReg);
        tvLogin = findViewById(R.id.tvLogin);
        etEmail = findViewById(R.id.etEmail);
        etPass = findViewById(R.id.etPass);
        mTextCnfPassword = findViewById(R.id.edittext_cnf_password);
        reg.setOnClickListener(this);
        tvLogin.setOnClickListener(this);
        checkBoxShowHidePass = findViewById(R.id.check_box_show_hide_pass);
        //Show/hide password
        checkBoxShowHidePass.setOnCheckedChangeListener((buttonView, isChecked) -> {
                if (isChecked) {
                    etPass.setTransformationMethod(null);
                    mTextCnfPassword.setTransformationMethod(null); } else {
                    etPass.setTransformationMethod(new PasswordTransformationMethod());
                    mTextCnfPassword.setTransformationMethod(new PasswordTransformationMethod()); }
        });
    }
    @Override
    public void onClick(View v) {
        switch(v.getId()){
            case R.id.btnReg:
                register();
                break;
            case R.id.tvLogin:
                startActivity(new Intent( packageContext: RegisterActivity.this,LoginActivity.class));
                finish();
                break;
            default:
        }
    }
    private void register(){
        Speech  speech=new Speech( context: RegisterActivity.this);
        if (number_for_longuage == 1) {
            speech.Arabic( data: "كلمة السر او البريد خطأ"); }
        else
        { speech.English( data: "Wrong Email/Password"); }
        String email = etEmail.getText().toString();
        String pass = etPass.getText().toString();
        String cnf_pwd = mTextCnfPassword.getText().toString().trim();Pattern emailPattern = Pattern.compile("^\\w+@\\w+\\.com");
```



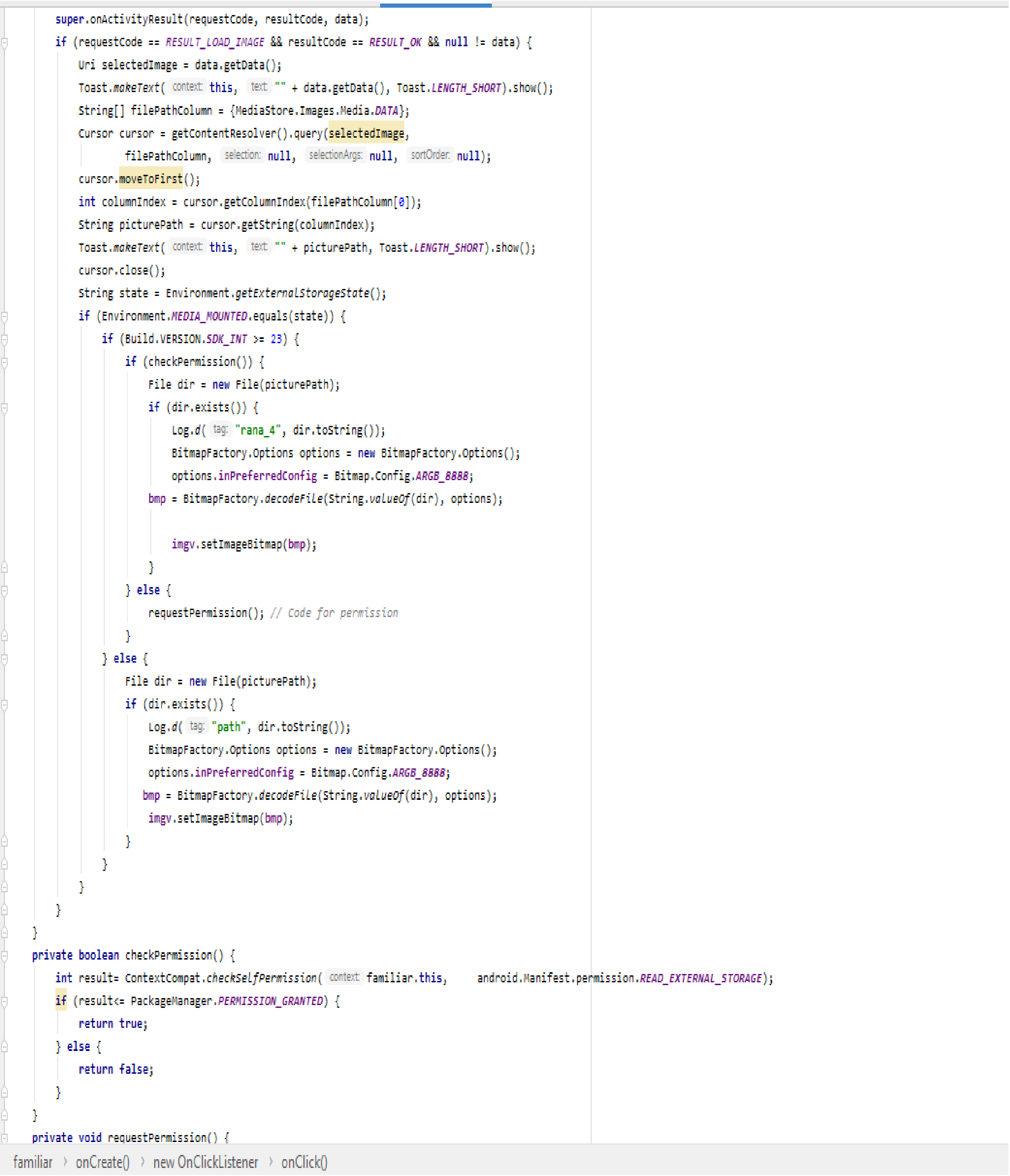
```
    if (number_for_language == 1) {
        speech.Arabic( data: "كلمة السر او البريد خاطئ"); }
    else
    { speech.English( data1: "Wrong Email/Password"); }
    String email = etEmail.getText().toString();
    String pass = etPass.getText().toString();
    String cnf_pwd = mTextCnfPassword.getText().toString().trim();Pattern emailPattern = Pattern.compile("^\\w+@\\w+\\.com");
    Matcher fit = emailPattern.matcher(email);if(email.isEmpty() || pass.isEmpty() || cnf_pwd.isEmpty()){
        displayToast("Please, Fill All Fields");
        if (number_for_language == 1) {
            speech.Arabic( data: "كلمة السر او البريد خاطئ"); }
        else
        { speech.English( data1: "Wrong Email/Password"); }
        return;
    }else if (!pass.contentEquals(cnf_pwd)) {
        if (number_for_language == 1) {
            speech.Arabic( data: "كلمة السر غير متطابقة"); }
        else
        { speech.English( data1: "The passwords do not match"); }
        Toast.makeText( context: this, text: "The passwords do not match",Toast.LENGTH_LONG).show();
        return;
    }else if(!fit.matches()) {
        if (number_for_language == 1) {
            speech.Arabic( data: "من فضلك ادخل البريد صحيح"); }
        else
        { speech.English( data1: "Please, Enter valid email"); }
        Toast.makeText( context: this, text: "Please, Enter valid email",Toast.LENGTH_LONG).show();
        return;
    }else{
        if (number_for_language == 1) {
            speech.Arabic( data: "تم اتمام عملية التسجيل"); }
        else
        { speech.English( data1: "Registered successfully"); }
        db.addUser(email,pass);
        displayToast("Registered successfully");
        startActivity(new Intent( packageContext: RegisterActivity.this,familiar.class));
        finish();
    }
}

private void displayToast(String message){
    Toast.makeText(getApplicationContext(), message, Toast.LENGTH_SHORT).show();
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater menuInflater = getMenuInflater();
    menuInflater.inflate(R.menu.main, menu);
    return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
```

Familiar activity

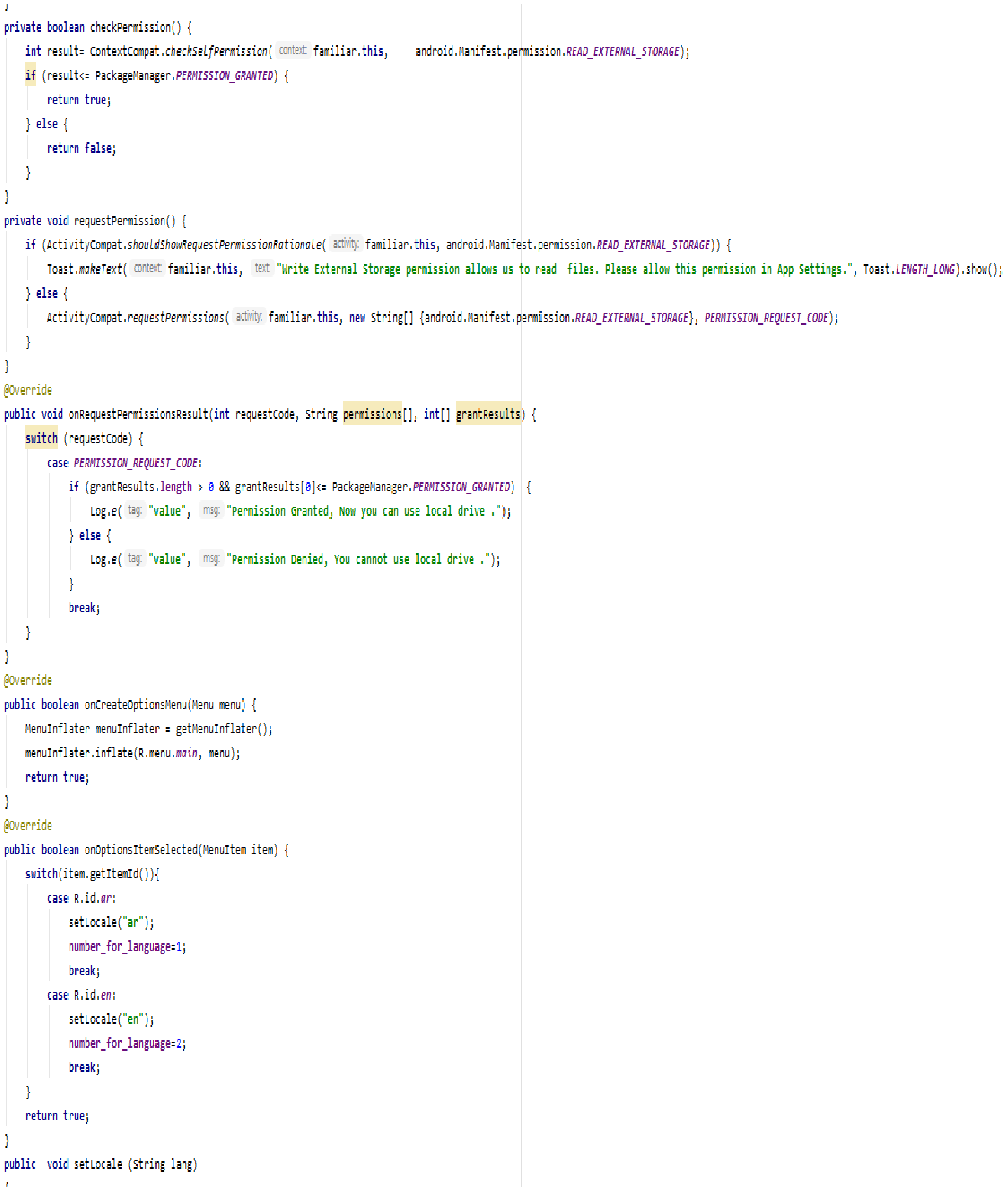
In this activity is the code of familiar page that have the declaration of Text Views and Edit Text to enter the email, password ,confirm password and Image View to get the image from gallery , and choose the image to appear the image and save this data in the database by pressing on save button , and choose the language that the text appear in it if the language is Arabic or English.



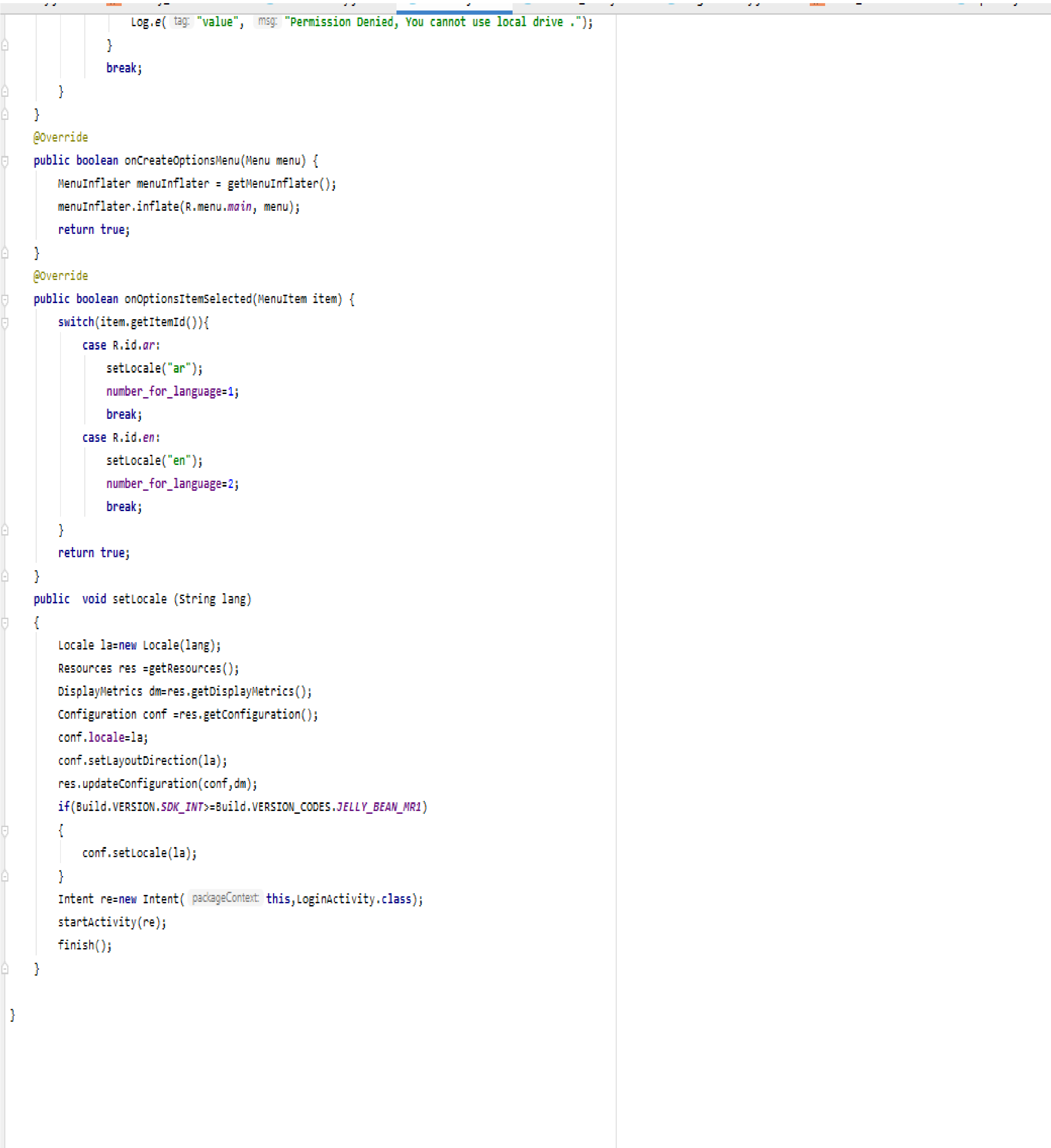
```
public class familiar extends AppCompatActivity {
    Button btn_save,btn_load_img;EditText edt_name,edt_type;ImageView imgv;Bitmap bmp;private static int RESULT_LOAD_IMAGE = 1;int number_for_language=1;
    private static final int PERMISSION_REQUEST_CODE=100;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_familiar);
        requestPermission();
        btn_save =findViewById(R.id.save);
        btn_load_img =findViewById(R.id.uploadimg);
        edt_name =findViewById(R.id.edt_name);
        edt_type =findViewById(R.id.edt_type);
        imgv =findViewById(R.id.imgv);
        //pick image from gallery
        btn_load_img.setOnClickListener((view) → {
                Intent i = new Intent(
                        Intent.ACTION_PICK, android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
                startActivityForResult(i, RESULT_LOAD_IMAGE);
        });
        btn_save.setOnClickListener((view) → {
                if(!edt_name.getText().toString().equals(null)&&!edt_type.getText().toString().equals(null))
                {
                    myDbAdapter db = new myDbAdapter( context: familiar.this);
                    ByteArrayOutputStream stream1=new ByteArrayOutputStream();
                    bmp.compress(Bitmap.CompressFormat.JPEG, quality: 100,stream1);
                    byte[] imageInbyte1=stream1.toByteArray();
                    Boolean b_1=  db.addData(imageInbyte1,edt_name.getText().toString(),edt_type.getText().toString());
                    Toast.makeText( context: familiar.this, text: ""+b_1, Toast.LENGTH_SHORT).show();
                    Intent intent=new Intent( packageContext: familiar.this,camera_view.class);
                    startActivity(intent);
                }
        });
    }
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == RESULT_LOAD_IMAGE && resultCode == RESULT_OK && null != data) {
            Uri selectedImage = data.getData();
            Toast.makeText( context: this, text: "" + data.getData(), Toast.LENGTH_SHORT).show();
            String[] filePathColumn = {MediaStore.Images.Media.DATA};
            Cursor cursor = getContentResolver().query(selectedImage,
                    filePathColumn, selection: null, selectionArgs: null, sortOrder: null);
            cursor.moveToFirst();
            int columnIndex = cursor.getColumnIndex(filePathColumn[0]);
            String picturePath = cursor.getString(columnIndex);
            Toast.makeText( context: this, text: "" + picturePath, Toast.LENGTH_SHORT).show();
            cursor.close();
            String state = Environment.getExternalStorageState();
```



```
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == RESULT_LOAD_IMAGE && resultCode == RESULT_OK && null != data) {
        Uri selectedImage = data.getData();
        Toast.makeText( context: this,  text: "" + data.getData(), Toast.LENGTH_SHORT).show();
        String[] filePathColumn = {MediaStore.Images.Media.DATA};
        Cursor cursor = getContentResolver().query(selectedImage,
                filePathColumn,  selection: null,  selectionArgs: null,  sortOrder: null);
        cursor.moveToFirst();
        int columnIndex = cursor.getColumnIndex(filePathColumn[0]);
        String picturePath = cursor.getString(columnIndex);
        Toast.makeText( context: this,  text: "" + picturePath, Toast.LENGTH_SHORT).show();
        cursor.close();
        String state = Environment.getExternalStorageState();
        if (Environment.MEDIA_MOUNTED.equals(state)) {
            if (Build.VERSION.SDK_INT >= 23) {
                if (checkPermission()) {
                    File dir = new File(picturePath);
                    if (dir.exists()) {
                        Log.d( tag: "rana_4", dir.toString());
                        BitmapFactory.Options options = new BitmapFactory.Options();
                        options.inPreferredConfig = Bitmap.Config.ARGB_8888;
                    bmp = BitmapFactory.decodeFile(String.valueOf(dir), options);

                        imgv.setImageBitmap(bmp);
                    }
                } else {
                    requestPermission(); // Code for permission
                }
            } else {
                File dir = new File(picturePath);
                if (dir.exists()) {
                    Log.d( tag: "path", dir.toString());
                    BitmapFactory.Options options = new BitmapFactory.Options();
                    options.inPreferredConfig = Bitmap.Config.ARGB_8888;
                   bmp = BitmapFactory.decodeFile(String.valueOf(dir), options);
                    imgv.setImageBitmap(bmp);
                }
            }
        }
    }
}
private boolean checkPermission() {
    int result= ContextCompat.checkSelfPermission( context: familiar.this,     android.Manifest.permission.READ_EXTERNAL_STORAGE);
    if (result<= PackageManager.PERMISSION_GRANTED) {
        return true;
    } else {
        return false;
    }
}
private void requestPermission() {
```



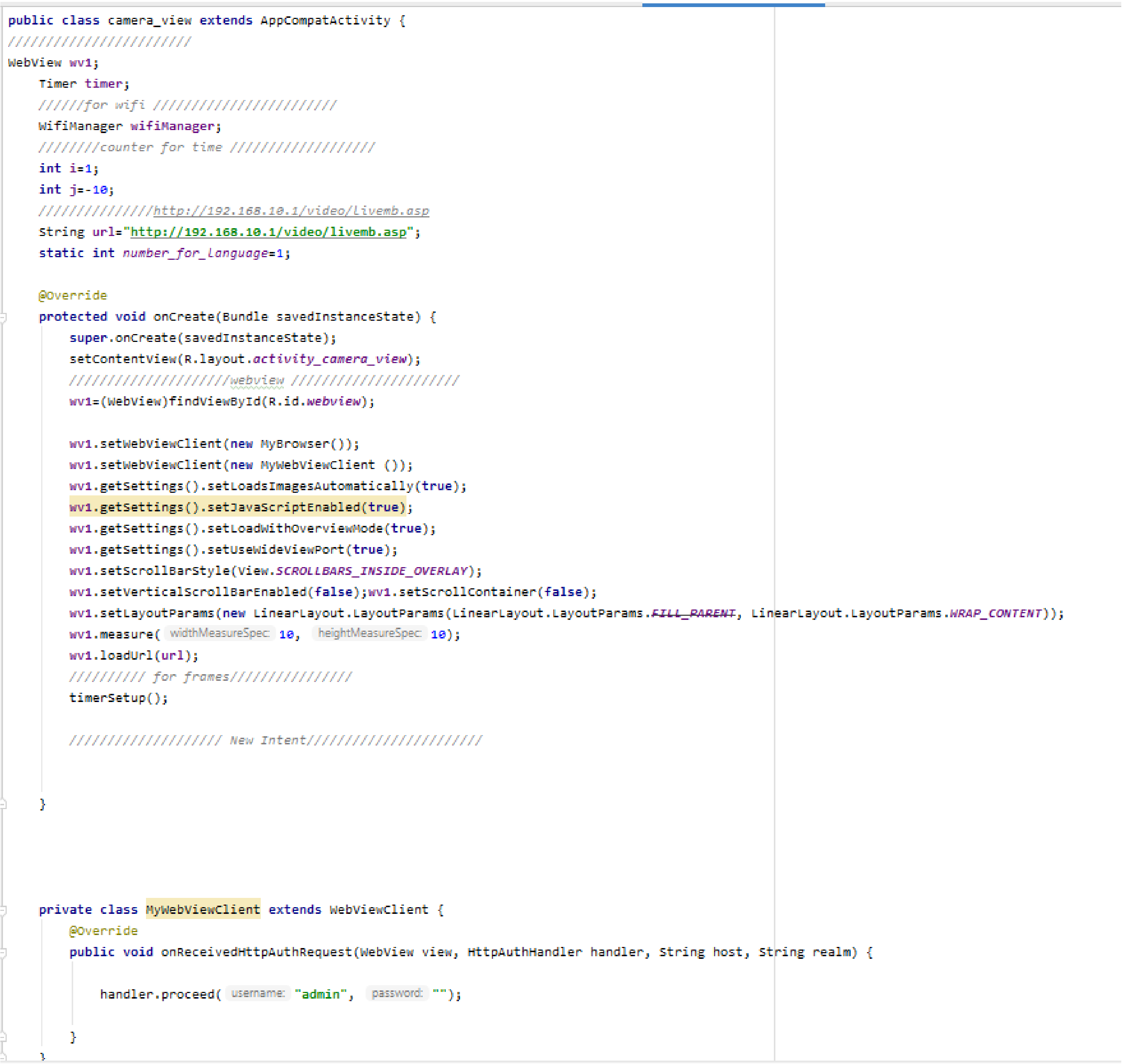
```
}
private boolean checkPermission() {
    int result= ContextCompat.checkSelfPermission( context: familiar.this,     android.Manifest.permission.READ_EXTERNAL_STORAGE);
    if (result<= PackageManager.PERMISSION_GRANTED) {
        return true;
    } else {
        return false;
    }
}
private void requestPermission() {
    if (ActivityCompat.shouldShowRequestPermissionRationale( activity: familiar.this, android.Manifest.permission.READ_EXTERNAL_STORAGE)) {
        Toast.makeText( context: familiar.this,  text: "Write External Storage permission allows us to read  files. Please allow this permission in App Settings.", Toast.LENGTH_LONG).show();
    } else {
        ActivityCompat.requestPermissions( activity: familiar.this, new String[] {android.Manifest.permission.READ_EXTERNAL_STORAGE}, PERMISSION_REQUEST_CODE);
    }
}
@Override
public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {
    switch (requestCode) {
        case PERMISSION_REQUEST_CODE:
            if (grantResults.length > 0 && grantResults[0]<= PackageManager.PERMISSION_GRANTED)  {
                Log.e( tag: "value",  msg: "Permission Granted, Now you can use local drive .");
            } else {
                Log.e( tag: "value",  msg: "Permission Denied, You cannot use local drive .");
            }
            break;
    }
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater menuInflater = getMenuInflater();
    menuInflater.inflate(R.menu.main, menu);
    return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch(item.getItemId()){
        case R.id.ar:
            setLocale("ar");
            number_for_language=1;
            break;
        case R.id.en:
            setLocale("en");
            number_for_language=2;
            break;
    }
    return true;
}
public  void setLocale (String lang)
{
```



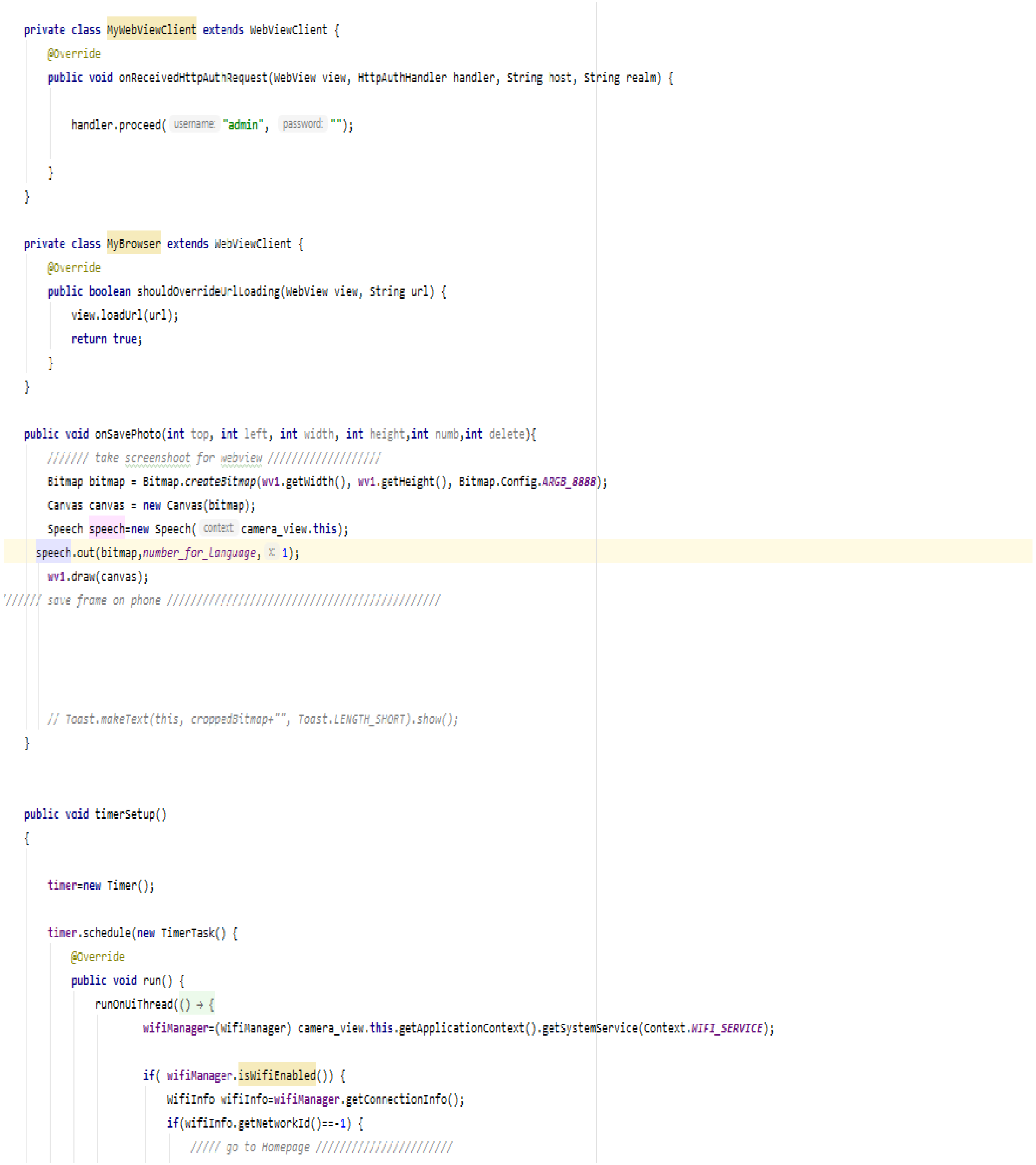
```
                    Log.e( tag: "value",  msg: "Permission Denied, You cannot use local drive .");
                }
                break;
        }
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater menuInflater = getMenuInflater();
        menuInflater.inflate(R.menu.main, menu);
        return true;
    }
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch(item.getItemId()){
            case R.id.ar:
                setLocale("ar");
                number_for_language=1;
                break;
            case R.id.en:
                setLocale("en");
                number_for_language=2;
                break;
        }
        return true;
    }
    public  void setLocale (String lang)
    {
        Locale la=new Locale(lang);
        Resources res =getResources();
        DisplayMetrics dm=res.getDisplayMetrics();
        Configuration conf =res.getConfiguration();
        conf.locale=la;
        conf.setLayoutDirection(la);
        res.updateConfiguration(conf,dm);
        if(Build.VERSION.SDK_INT>=Build.VERSION_CODES.JELLY_BEAN_MR1)
        {
            conf.setLocale(la);
        }
        Intent re=new Intent( packageContext: this,LoginActivity.class);
        startActivity(re);
        finish();
    }

}
```

camera activity

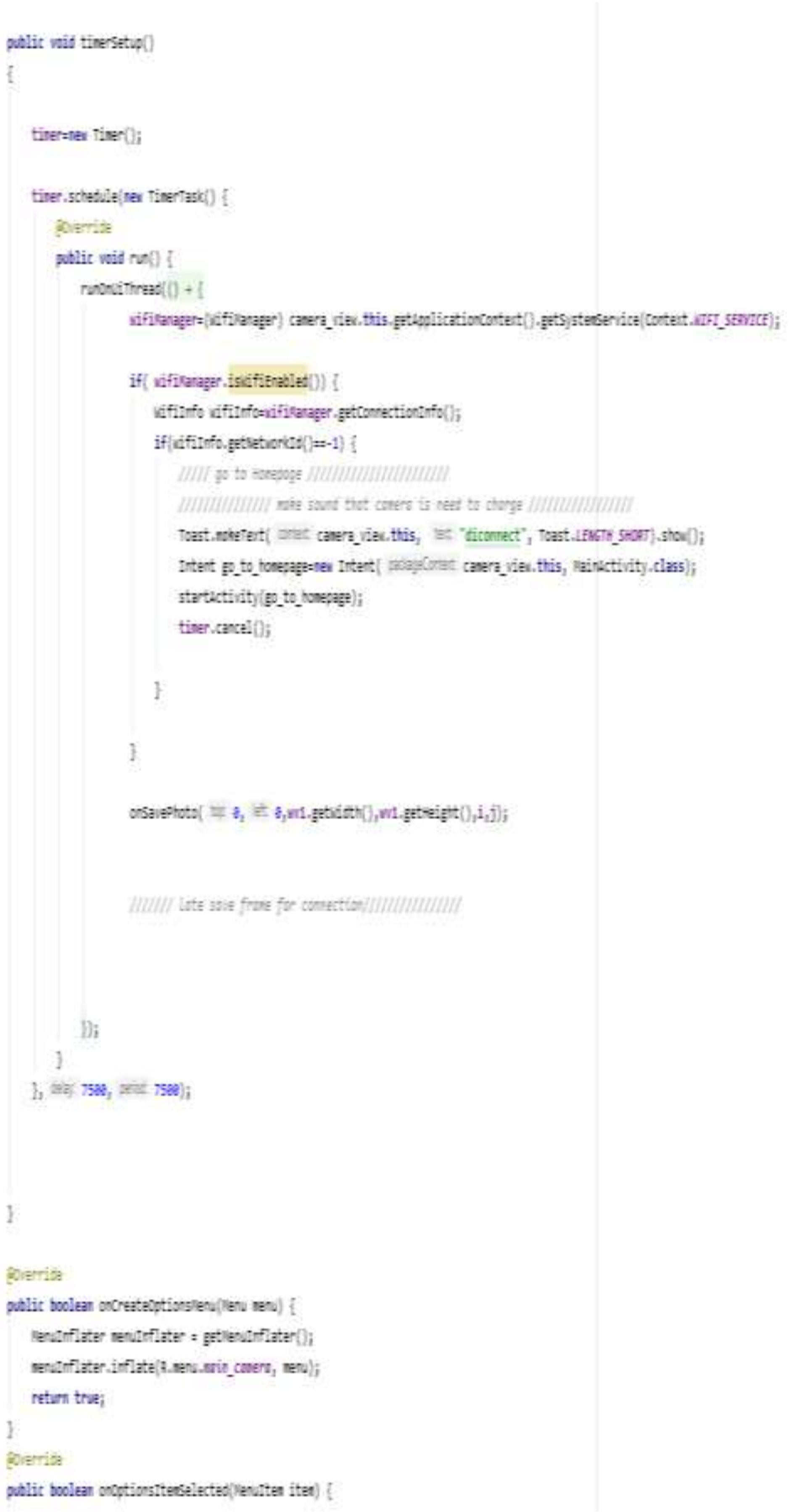
In this activity is the code[8] of camera connectivity by the WebView by using the URL of the camera to represent the video captured by camera, and apply some settings to setup , and using timer to control of the number of frames captured depending on the number of seconds that used to be analyzing , and returns by the output of the image , and choose the language that the text appear in it if the language is Arabic or English.



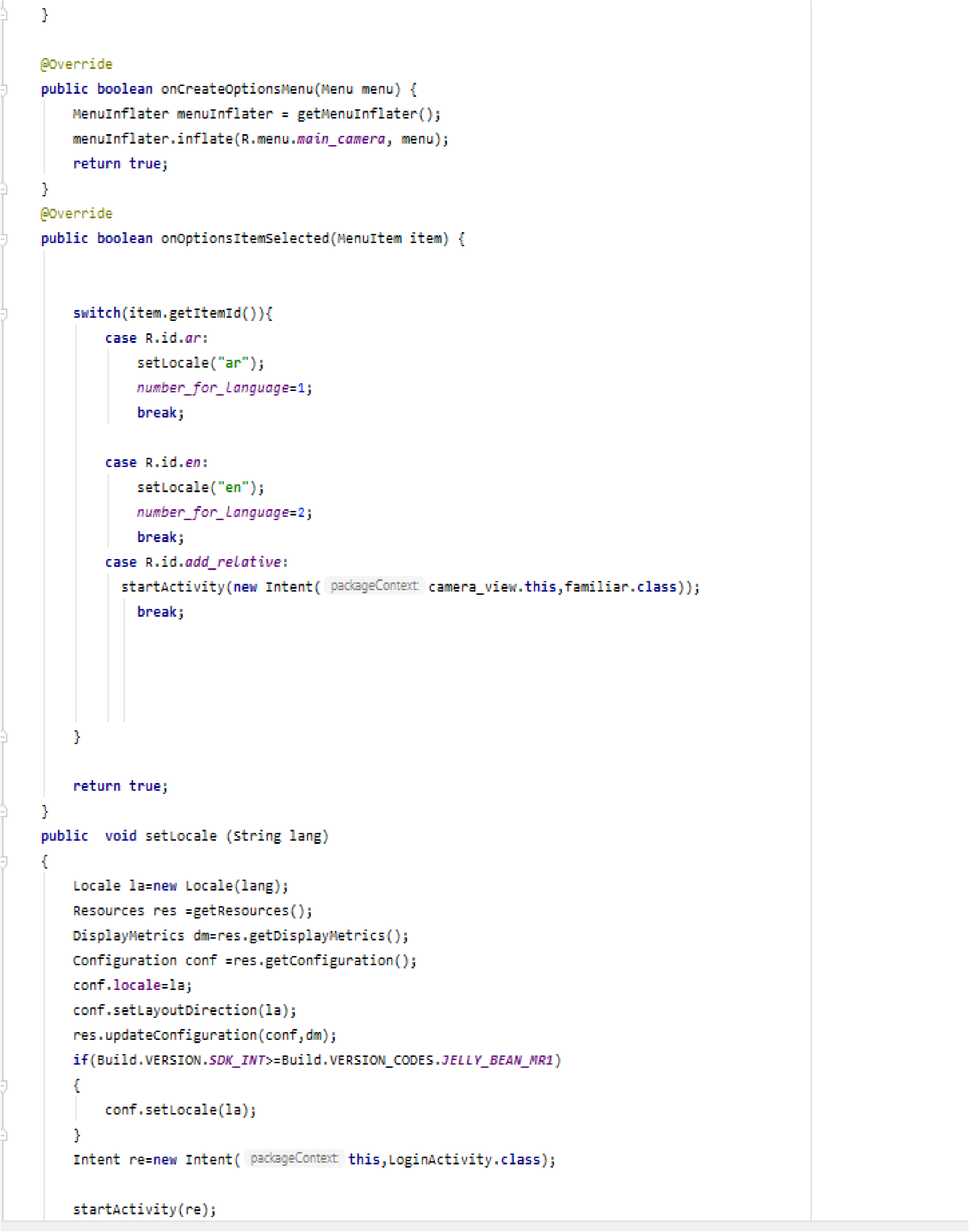
```
public class camera_view extends AppCompatActivity {
////////////////////////
WebView wv1;
    Timer timer;
    //////for wifi ////////////////////////
    WifiManager wifiManager;
    ////////counter for time ///////////////////
    int i=1;
    int j=-10;
    ///////////////http://192.168.10.1/video/livemb.asp
    String url="http://192.168.10.1/video/livemb.asp";
    static int number_for_language=1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_camera_view);
        /////////////////////webview //////////////////////
        wv1=(WebView)findViewById(R.id.webview);

        wv1.setWebViewClient(new MyBrowser());
        wv1.setWebViewClient(new MyWebViewClient ());
        wv1.getSettings().setLoadsImagesAutomatically(true);
        wv1.getSettings().setJavaScriptEnabled(true);
        wv1.getSettings().setLoadWithOverviewMode(true);
        wv1.getSettings().setUseWideViewPort(true);
        wv1.setScrollBarStyle(View.SCROLLBARS_INSIDE_OVERLAY);
        wv1.setVerticalScrollBarEnabled(false);wv1.setScrollContainer(false);
        wv1.setLayoutParams(new LinearLayout.LayoutParams(LinearLayout.LayoutParams.~~FILL_PARENT~~, LinearLayout.LayoutParams.WRAP_CONTENT));
        wv1.measure( widthMeasureSpec: 10,  heightMeasureSpec: 10);
        wv1.loadUrl(url);
        ////////// for frames////////////////
        timerSetup();

        //////////////////// New Intent///////////////////////


    }



    private class MyWebViewClient extends WebViewClient {
        @Override
        public void onReceivedHttpAuthRequest(WebView view, HttpAuthHandler handler, String host, String realm) {

            handler.proceed( username: "admin",  password: "");

        }
    }
```



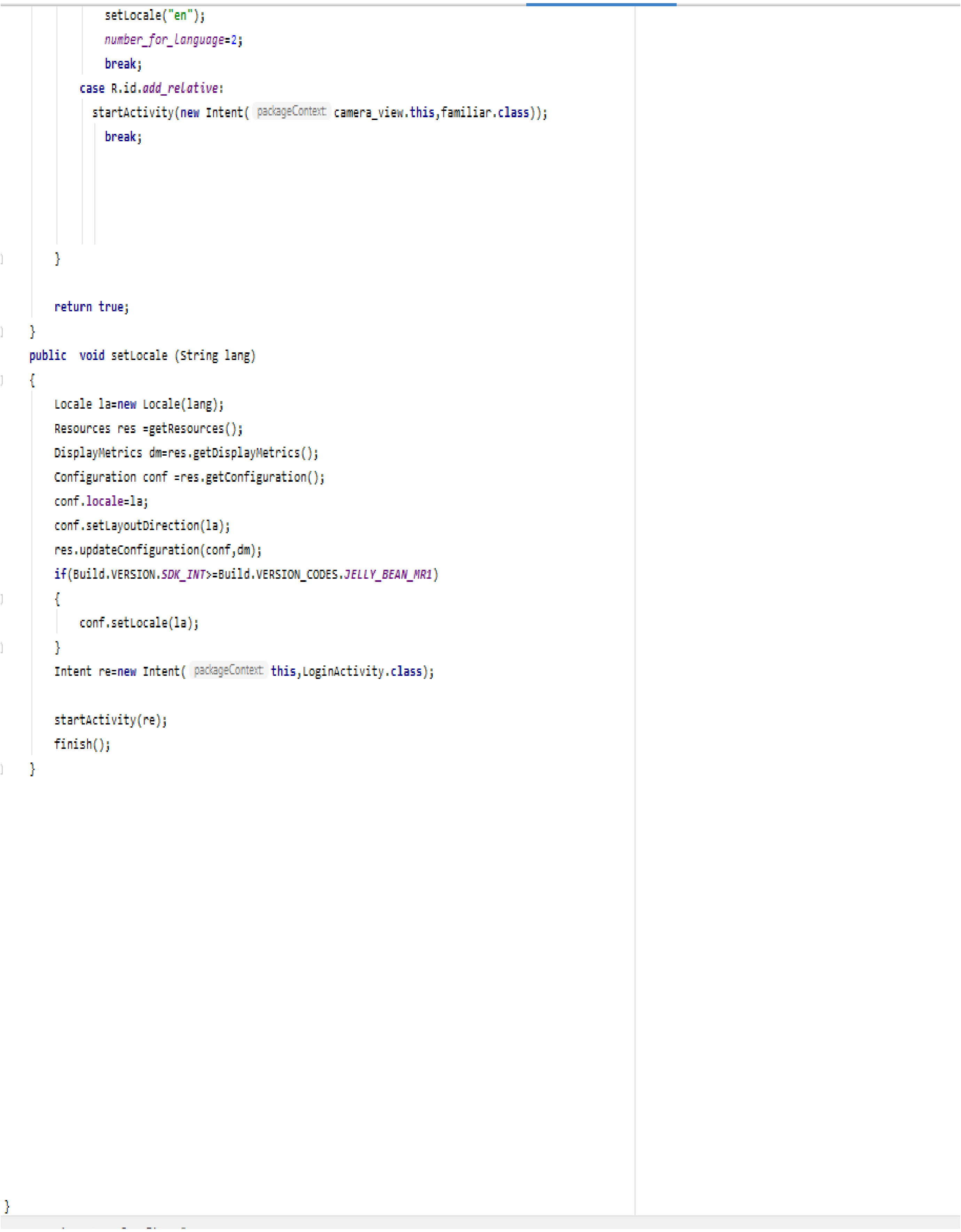
```java
private class MyWebViewClient extends WebViewClient {
    @Override
    public void onReceivedHttpAuthRequest(WebView view, HttpAuthHandler handler, String host, String realm) {

        handler.proceed( username: "admin",  password: "");

    }
}

private class MyBrowser extends WebViewClient {
    @Override
    public boolean shouldOverrideUrlLoading(WebView view, String url) {
        view.loadUrl(url);
        return true;
    }
}

public void onSavePhoto(int top, int left, int width, int height,int numb,int delete){
    /////// take screenshoot for webview ///////////////////
    Bitmap bitmap = Bitmap.createBitmap(wv1.getWidth(), wv1.getHeight(), Bitmap.Config.ARGB_8888);
    Canvas canvas = new Canvas(bitmap);
    Speech speech=new Speech( context: camera_view.this);
  speech.out(bitmap,number_for_language, x: 1);
    wv1.draw(canvas);
'////// save frame on phone /////////////////////////////////////////////////




    // Toast.makeText(this, croppedBitmap+"", Toast.LENGTH_SHORT).show();
}


public void timerSetup()
{

    timer=new Timer();

    timer.schedule(new TimerTask() {
        @Override
        public void run() {
            runOnUiThread(() → {
                    wifiManager=(WifiManager) camera_view.this.getApplicationContext().getSystemService(Context.WIFI_SERVICE);

                    if( wifiManager.isWifiEnabled()) {
                        WifiInfo wifiInfo=wifiManager.getConnectionInfo();
                        if(wifiInfo.getNetworkId()==-1) {
                            ///// go to Homepage /////////////////////
```



```java
public void timerSetup()
{

    timer=new Timer();

    timer.schedule(new TimerTask() {
        @Override
        public void run() {
            runOnUiThread(() -> {
                    wifiManager=(WifiManager) camera_view.this.getApplicationContext().getSystemService(Context.WIFI_SERVICE);

                    if( wifiManager.isWifiEnabled()) {
                        WifiInfo wifiInfo=wifiManager.getConnectionInfo();
                        if(wifiInfo.getNetworkId()==-1) {
                            ////// go to Homepage ////////////////////////
                            //////////////// make sound that camera is need to charge //////////////////
                            Toast.makeText( context: camera_view.this,  text: "diconnect", Toast.LENGTH_SHORT).show();
                            Intent go_to_homepage=new Intent( packageContext: camera_view.this, MainActivity.class);
                            startActivity(go_to_homepage);
                            timer.cancel();

                        }

                    }

                    onSavePhoto( x: 0, y: 0,wv1.getWidth(),wv1.getHeight(),i,j);



                    /////// late save frame for connection/////////////////



            });
        }
    }, delay: 7500, period: 7500);



}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater menuInflater = getMenuInflater();
    menuInflater.inflate(R.menu.main_camera, menu);
    return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
```



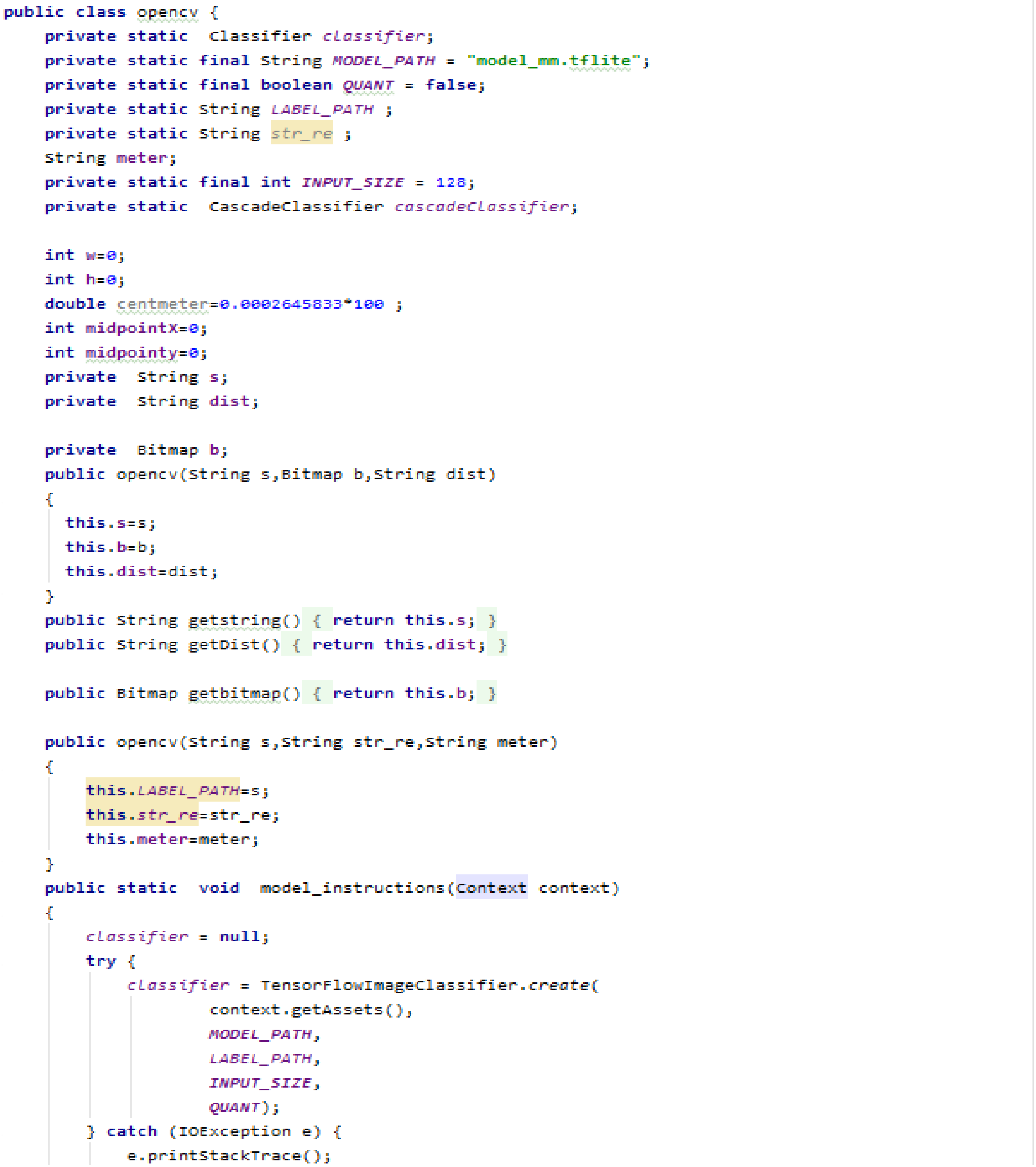
```
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater menuInflater = getMenuInflater();
    menuInflater.inflate(R.menu.main_camera, menu);
    return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {


    switch(item.getItemId()){
        case R.id.ar:
            setLocale("ar");
            number_for_language=1;
            break;

        case R.id.en:
            setLocale("en");
            number_for_language=2;
            break;
        case R.id.add_relative:
          startActivity(new Intent( packageContext: camera_view.this,familiar.class));
            break;




    }

    return true;
}
public  void setLocale (String lang)
{
    Locale la=new Locale(lang);
    Resources res =getResources();
    DisplayMetrics dm=res.getDisplayMetrics();
    Configuration conf =res.getConfiguration();
    conf.locale=la;
    conf.setLayoutDirection(la);
    res.updateConfiguration(conf,dm);
    if(Build.VERSION.SDK_INT>=Build.VERSION_CODES.JELLY_BEAN_MR1)
    {
        conf.setLocale(la);
    }
    Intent re=new Intent( packageContext: this,LoginActivity.class);

    startActivity(re);
```



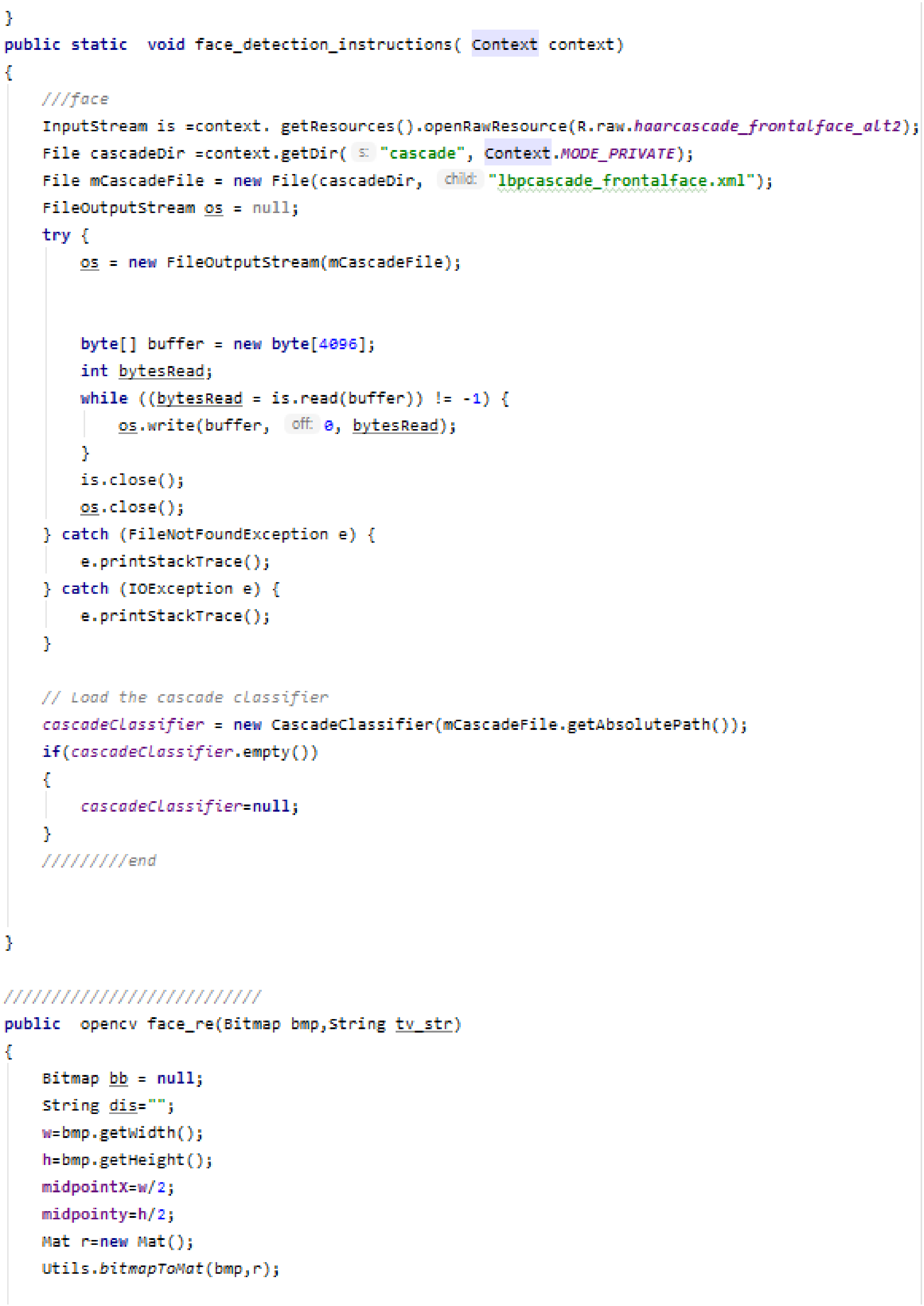
```
                setLocale("en");
                number_for_language=2;
                break;
            case R.id.add_relative:
              startActivity(new Intent( packageContext: camera_view.this,familiar.class));
                break;




        }

        return true;
    }
    public  void setLocale (String lang)
    {
        Locale la=new Locale(lang);
        Resources res =getResources();
        DisplayMetrics dm=res.getDisplayMetrics();
        Configuration conf =res.getConfiguration();
        conf.locale=la;
        conf.setLayoutDirection(la);
        res.updateConfiguration(conf,dm);
        if(Build.VERSION.SDK_INT>=Build.VERSION_CODES.JELLY_BEAN_MR1)
        {
            conf.setLocale(la);
        }
        Intent re=new Intent( packageContext: this,LoginActivity.class);

        startActivity(re);
        finish();
    }










}
```

OpenCV

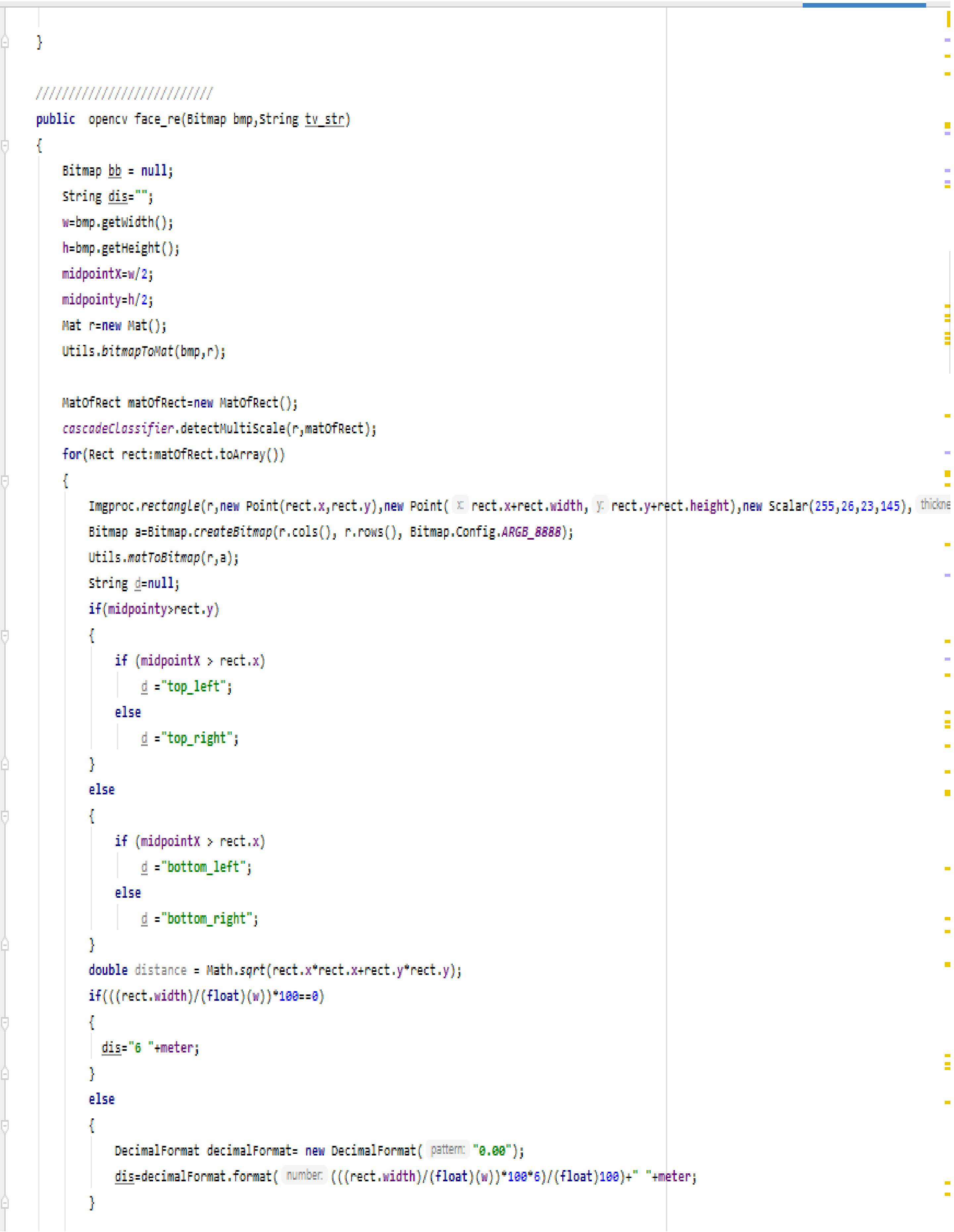
In this activity is the code of OpenCV that is used to detect faces, and comparing these images[9] to know if they are familiar or not , or detect objects[9] by cropping[11] the image into small of pieces then send each of piece to model by using TensorFlow [10]as platform to connect the application and model then send result to speech activity .



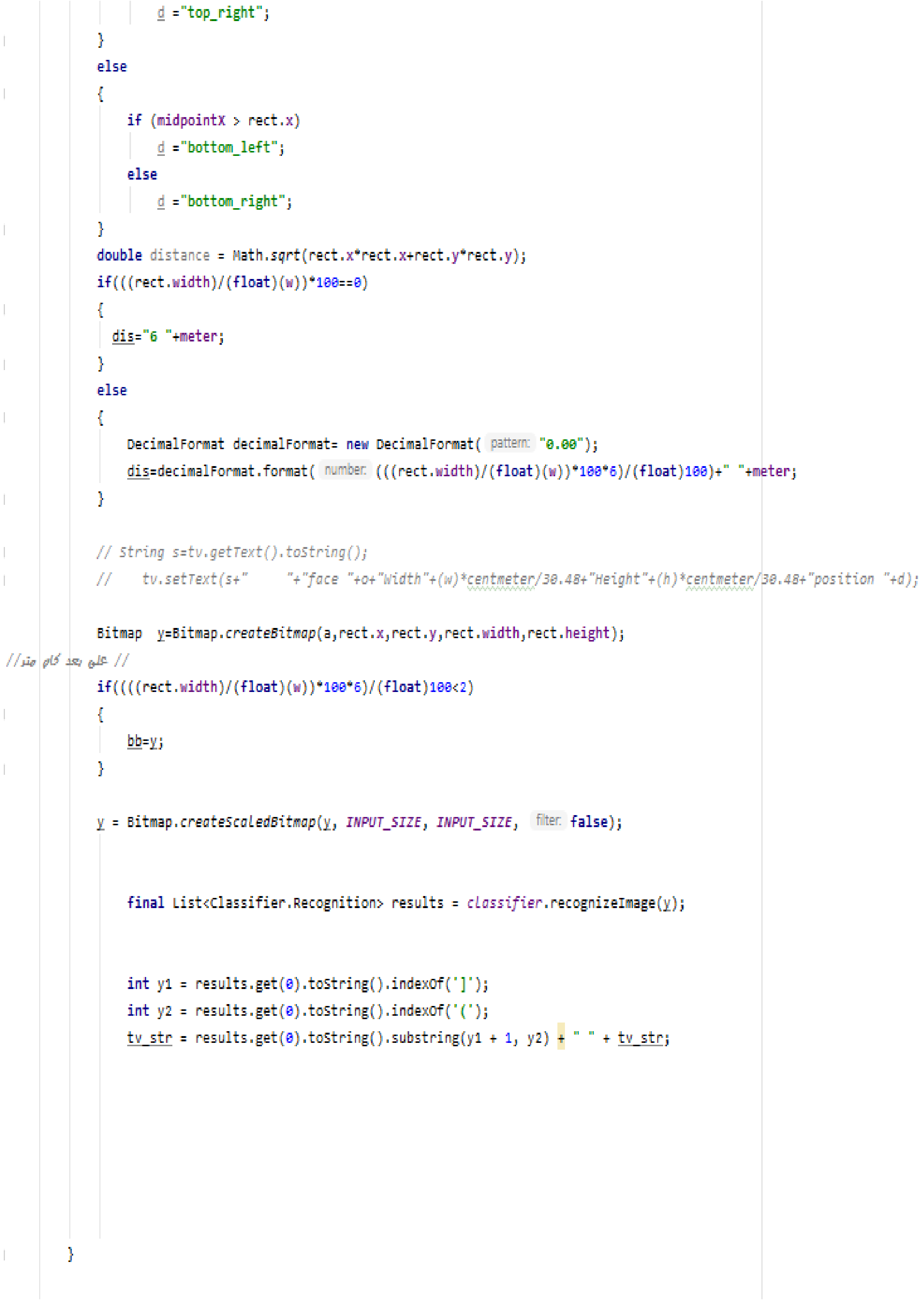
```
public class opencv {
    private static  Classifier classifier;
    private static final String MODEL_PATH = "model_mm.tflite";
    private static final boolean QUANT = false;
    private static String LABEL_PATH ;
    private static String str_re ;
    String meter;
    private static final int INPUT_SIZE = 128;
    private static  CascadeClassifier cascadeClassifier;

    int w=0;
    int h=0;
    double centmeter=0.0002645833*100 ;
    int midpointX=0;
    int midpointy=0;
    private  String s;
    private  String dist;

    private  Bitmap b;
    public opencv(String s,Bitmap b,String dist)
    {
      this.s=s;
      this.b=b;
      this.dist=dist;
    }
    public String getstring() { return this.s; }
    public String getDist() { return this.dist; }

    public Bitmap getbitmap() { return this.b; }

    public opencv(String s,String str_re,String meter)
    {
        this.LABEL_PATH=s;
        this.str_re=str_re;
        this.meter=meter;
    }
    public static  void  model_instructions(Context context)
    {
        classifier = null;
        try {
            classifier = TensorFlowImageClassifier.create(
                    context.getAssets(),
                    MODEL_PATH,
                    LABEL_PATH,
                    INPUT_SIZE,
                    QUANT);
        } catch (IOException e) {
            e.printStackTrace();
```



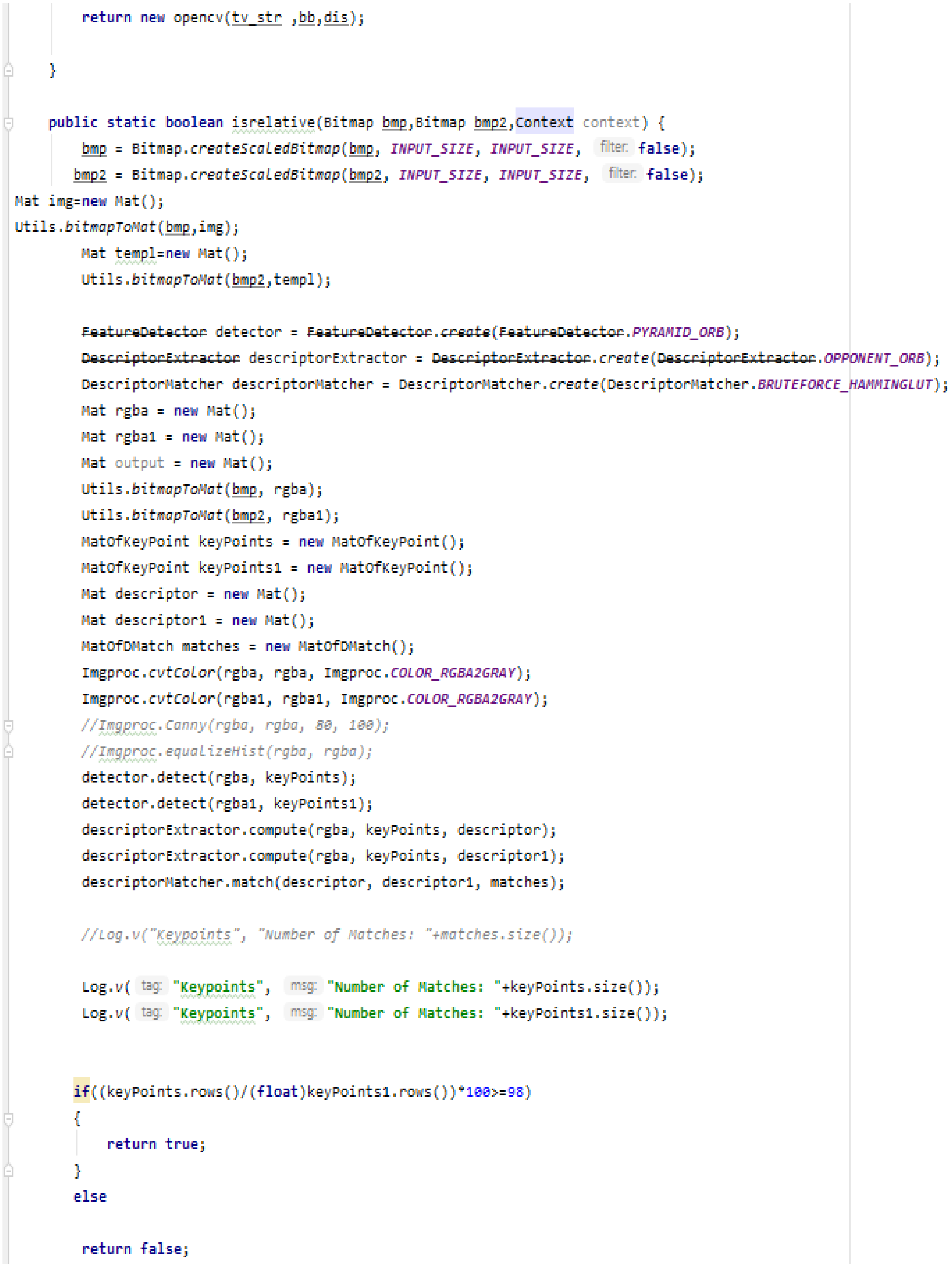
```
}
public static  void face_detection_instructions( Context context)
{
    ///face
    InputStream is =context. getResources().openRawResource(R.raw.haarcascade_frontalface_alt2);
    File cascadeDir =context.getDir( s: "cascade", Context.MODE_PRIVATE);
    File mCascadeFile = new File(cascadeDir,  child: "lbpcascade_frontalface.xml");
    FileOutputStream os = null;
    try {
        os = new FileOutputStream(mCascadeFile);


        byte[] buffer = new byte[4096];
        int bytesRead;
        while ((bytesRead = is.read(buffer)) != -1) {
            os.write(buffer,  off: 0, bytesRead);
        }
        is.close();
        os.close();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }

    // Load the cascade classifier
    cascadeClassifier = new CascadeClassifier(mCascadeFile.getAbsolutePath());
    if(cascadeClassifier.empty())
    {
        cascadeClassifier=null;
    }
    /////////end


}

///////////////////////////
public  opencv face_re(Bitmap bmp,String tv_str)
{
    Bitmap bb = null;
    String dis="";
    w=bmp.getWidth();
    h=bmp.getHeight();
    midpointX=w/2;
    midpointy=h/2;
    Mat r=new Mat();
    Utils.bitmapToMat(bmp,r);
```



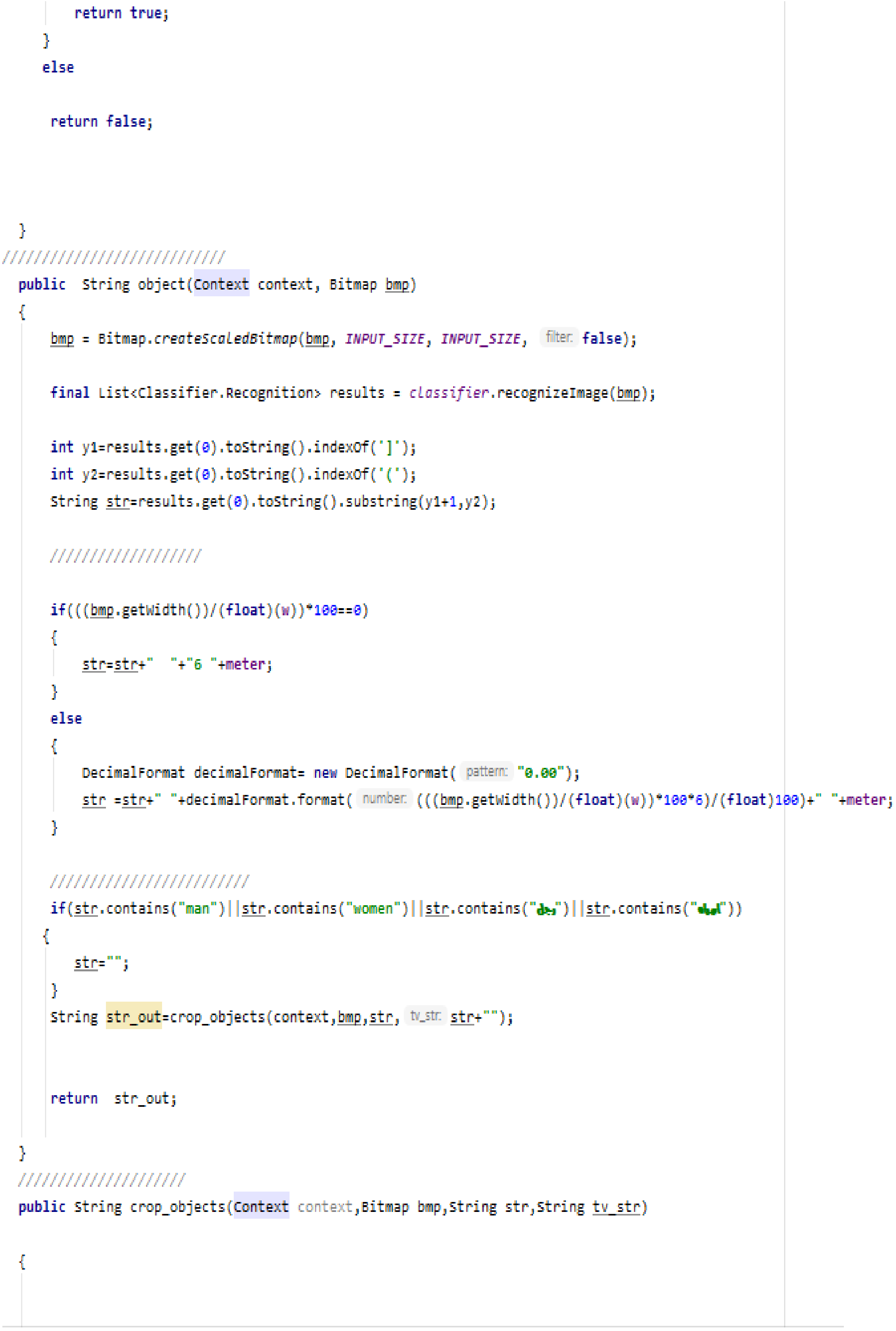
```
}

//////////////////////////
public  opencv face_re(Bitmap bmp,String tv_str)
{
    Bitmap bb = null;
    String dis="";
    w=bmp.getWidth();
    h=bmp.getHeight();
    midpointX=w/2;
    midpointy=h/2;
    Mat r=new Mat();
    Utils.bitmapToMat(bmp,r);

    MatOfRect matOfRect=new MatOfRect();
    cascadeClassifier.detectMultiScale(r,matOfRect);
    for(Rect rect:matOfRect.toArray())
    {
        Imgproc.rectangle(r,new Point(rect.x,rect.y),new Point( x: rect.x+rect.width, y: rect.y+rect.height),new Scalar(255,26,23,145), thickne
        Bitmap a=Bitmap.createBitmap(r.cols(), r.rows(), Bitmap.Config.ARGB_8888);
        Utils.matToBitmap(r,a);
        String d=null;
        if(midpointy>rect.y)
        {
            if (midpointX > rect.x)
                d ="top_left";
            else
                d ="top_right";
        }
        else
        {
            if (midpointX > rect.x)
                d ="bottom_left";
            else
                d ="bottom_right";
        }
        double distance = Math.sqrt(rect.x*rect.x+rect.y*rect.y);
        if(((rect.width)/(float)(w))*100==0)
        {
          dis="6 "+meter;
        }
        else
        {
            DecimalFormat decimalFormat= new DecimalFormat( pattern: "0.00");
            dis=decimalFormat.format( number: (((rect.width)/(float)(w))*100*6)/(float)100)+" "+meter;
        }
```



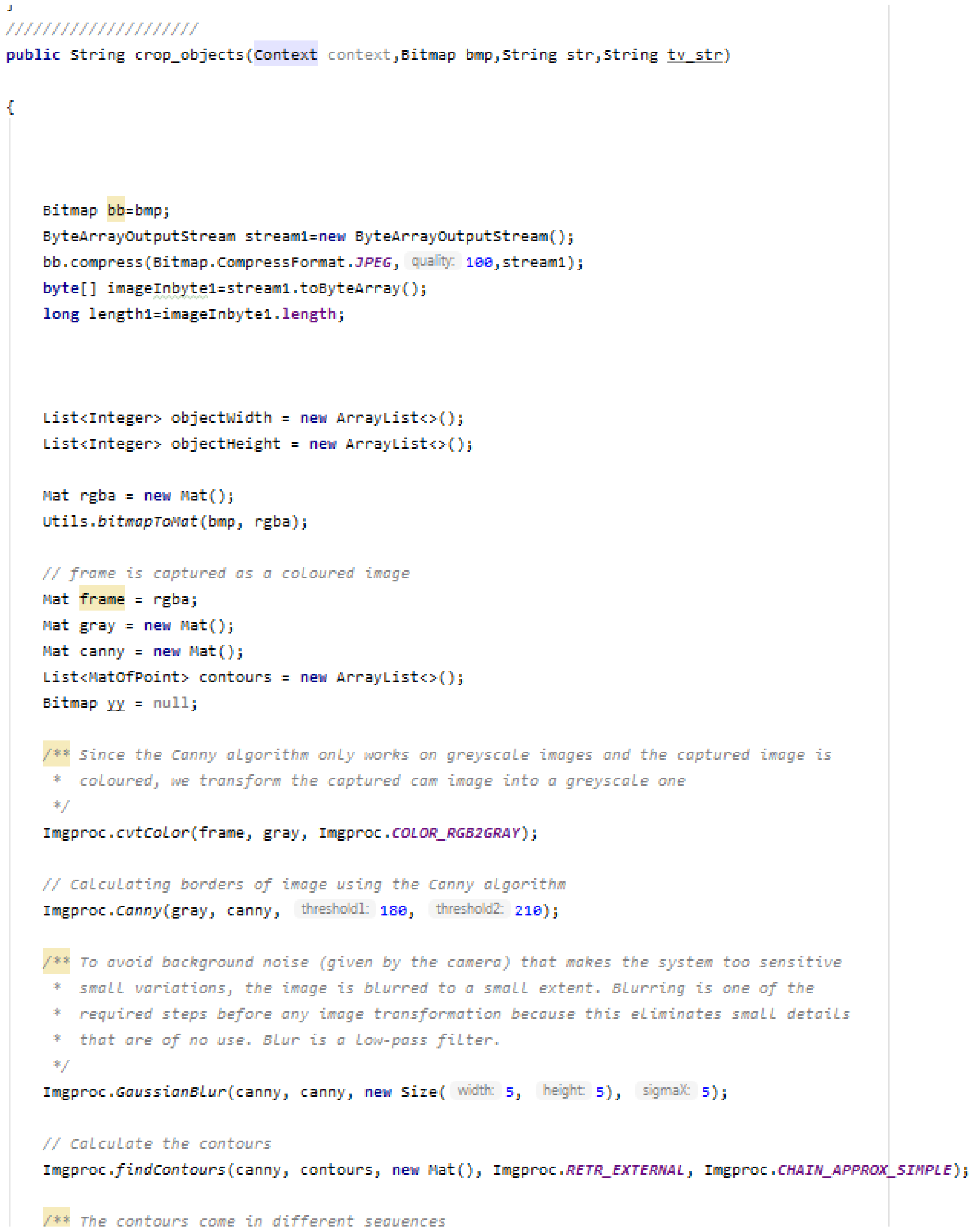
```
                d ="top_right";
        }
        else
        {
            if (midpointX > rect.x)
                d ="bottom_left";
            else
                d ="bottom_right";
        }
        double distance = Math.sqrt(rect.x*rect.x+rect.y*rect.y);
        if(((rect.width)/(float)(w))*100==0)
        {
          dis="6 "+meter;
        }
        else
        {
            DecimalFormat decimalFormat= new DecimalFormat( pattern: "0.00");
            dis=decimalFormat.format( number: (((rect.width)/(float)(w))*100*6)/(float)100)+" "+meter;
        }

        // String s=tv.getText().toString();
        //    tv.setText(s+"      "+"face "+o+"Width"+(w)*centmeter/30.48+"Height"+(h)*centmeter/30.48+"position "+d);

        Bitmap  y=Bitmap.createBitmap(a,rect.x,rect.y,rect.width,rect.height);
//متر كام بعد علي //
        if((((rect.width)/(float)(w))*100*6)/(float)100<2)
        {
            bb=y;
        }

        y = Bitmap.createScaledBitmap(y, INPUT_SIZE, INPUT_SIZE,  filter: false);


            final List<Classifier.Recognition> results = classifier.recognizeImage(y);


            int y1 = results.get(0).toString().indexOf(']');
            int y2 = results.get(0).toString().indexOf('(');
            tv_str = results.get(0).toString().substring(y1 + 1, y2) + " " + tv_str;










    }
```



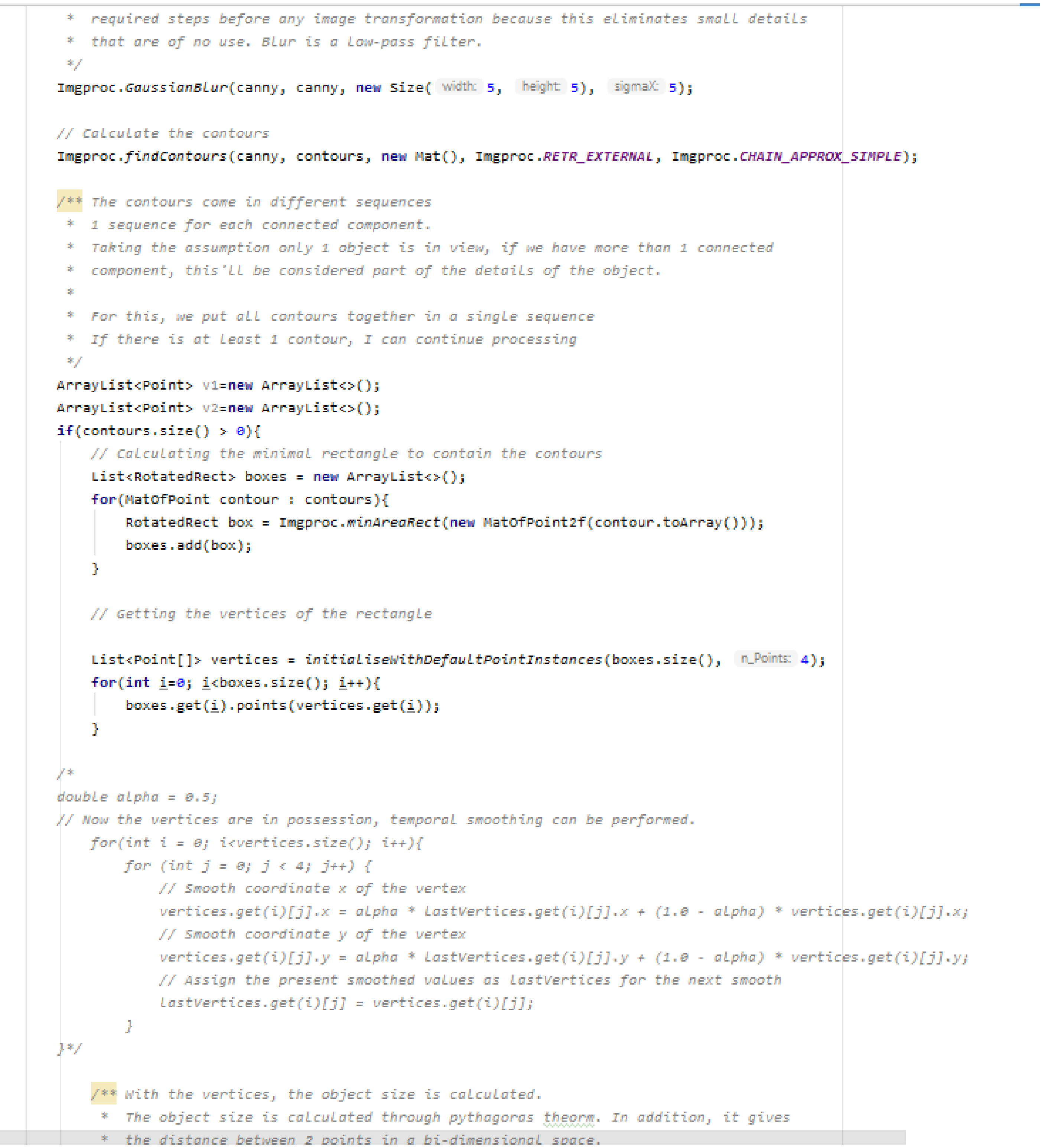
```
        return new opencv(tv_str ,bb,dis);

    }

    public static boolean isrelative(Bitmap bmp,Bitmap bmp2,Context context) {
        bmp = Bitmap.createScaledBitmap(bmp, INPUT_SIZE, INPUT_SIZE,  filter: false);
       bmp2 = Bitmap.createScaledBitmap(bmp2, INPUT_SIZE, INPUT_SIZE,  filter: false);
Mat img=new Mat();
Utils.bitmapToMat(bmp,img);
        Mat templ=new Mat();
        Utils.bitmapToMat(bmp2,templ);

        ~~FeatureDetector~~ detector = ~~FeatureDetector.create~~(~~FeatureDetector~~.PYRAMID_ORB);
        ~~DescriptorExtractor~~ descriptorExtractor = ~~DescriptorExtractor~~.create(~~DescriptorExtractor~~.OPPONENT_ORB);
        DescriptorMatcher descriptorMatcher = DescriptorMatcher.create(DescriptorMatcher.BRUTEFORCE_HAMMINGLUT);
        Mat rgba = new Mat();
        Mat rgba1 = new Mat();
        Mat output = new Mat();
        Utils.bitmapToMat(bmp, rgba);
        Utils.bitmapToMat(bmp2, rgba1);
        MatOfKeyPoint keyPoints = new MatOfKeyPoint();
        MatOfKeyPoint keyPoints1 = new MatOfKeyPoint();
        Mat descriptor = new Mat();
        Mat descriptor1 = new Mat();
        MatOfDMatch matches = new MatOfDMatch();
        Imgproc.cvtColor(rgba, rgba, Imgproc.COLOR_RGBA2GRAY);
        Imgproc.cvtColor(rgba1, rgba1, Imgproc.COLOR_RGBA2GRAY);
        //Imgproc.Canny(rgba, rgba, 80, 100);
        //Imgproc.equalizeHist(rgba, rgba);
        detector.detect(rgba, keyPoints);
        detector.detect(rgba1, keyPoints1);
        descriptorExtractor.compute(rgba, keyPoints, descriptor);
        descriptorExtractor.compute(rgba, keyPoints, descriptor1);
        descriptorMatcher.match(descriptor, descriptor1, matches);

        //Log.v("Keypoints", "Number of Matches: "+matches.size());

        Log.v( tag: "Keypoints",  msg: "Number of Matches: "+keyPoints.size());
        Log.v( tag: "Keypoints",  msg: "Number of Matches: "+keyPoints1.size());


       if((keyPoints.rows()/(float)keyPoints1.rows())*100>=98)
       {
           return true;
       }
       else

        return false;
```



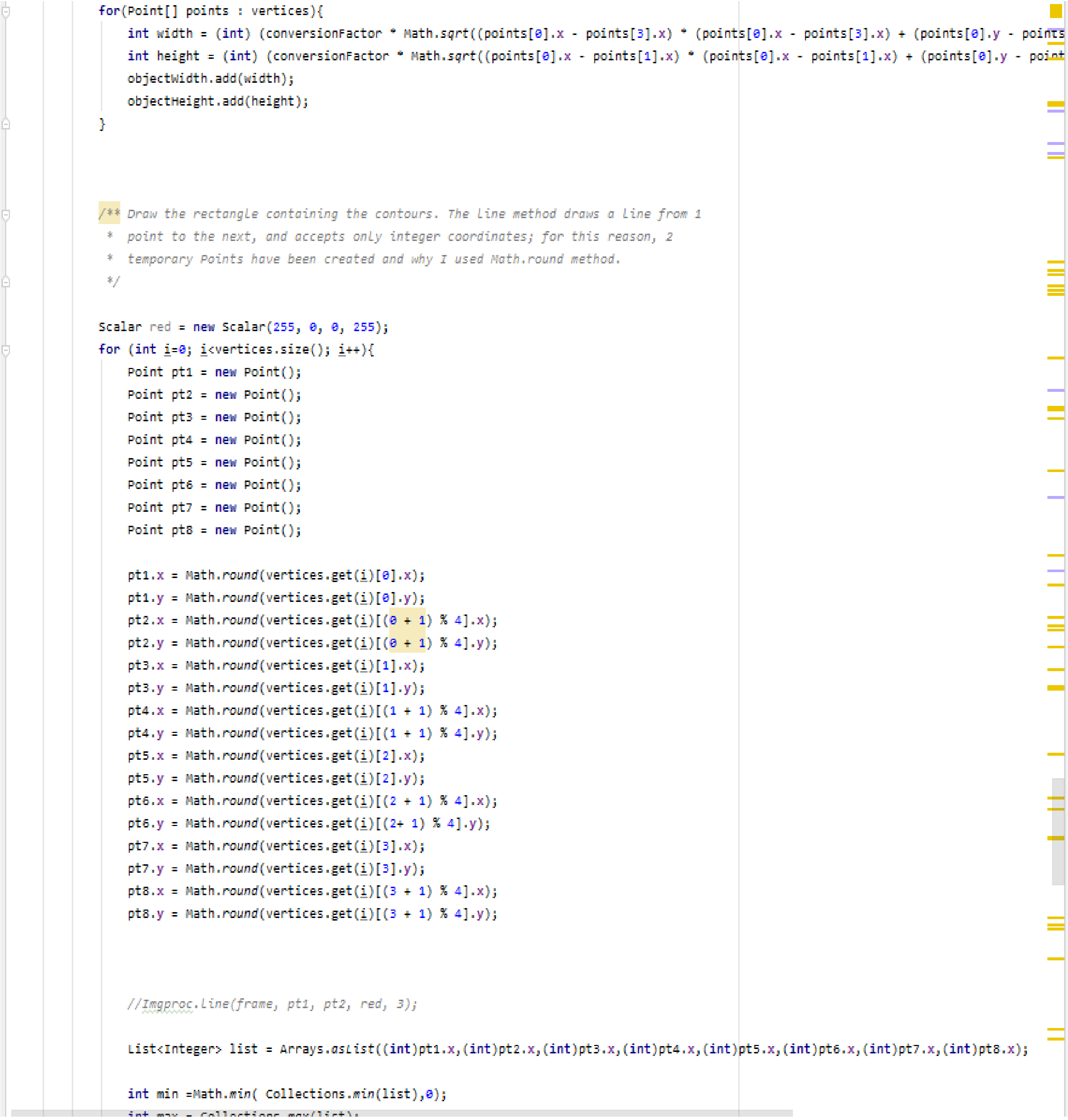
```
        return true;
    }
    else

     return false;



}
////////////////////////////
  public  String object(Context context, Bitmap bmp)
  {
      bmp = Bitmap.createScaledBitmap(bmp, INPUT_SIZE, INPUT_SIZE,  filter: false);

      final List<Classifier.Recognition> results = classifier.recognizeImage(bmp);

      int y1=results.get(0).toString().indexOf(']');
      int y2=results.get(0).toString().indexOf('(');
      String str=results.get(0).toString().substring(y1+1,y2);

      ///////////////////

      if(((bmp.getWidth())/(float)(w))*100==0)
      {
          str=str+"  "+"6 "+meter;
      }
      else
      {
          DecimalFormat decimalFormat= new DecimalFormat( pattern: "0.00");
          str =str+" "+decimalFormat.format( number: (((bmp.getWidth())/(float)(w))*100*6)/(float)100)+" "+meter;
      }

      /////////////////////////
      if(str.contains("man")||str.contains("women")||str.contains("رجل")||str.contains("طفل"))
     {
         str="";
      }
      String str_out=crop_objects(context,bmp,str, tv_str: str+"");


      return  str_out;

  }
  ////////////////////
  public String crop_objects(Context context,Bitmap bmp,String str,String tv_str)

  {
```



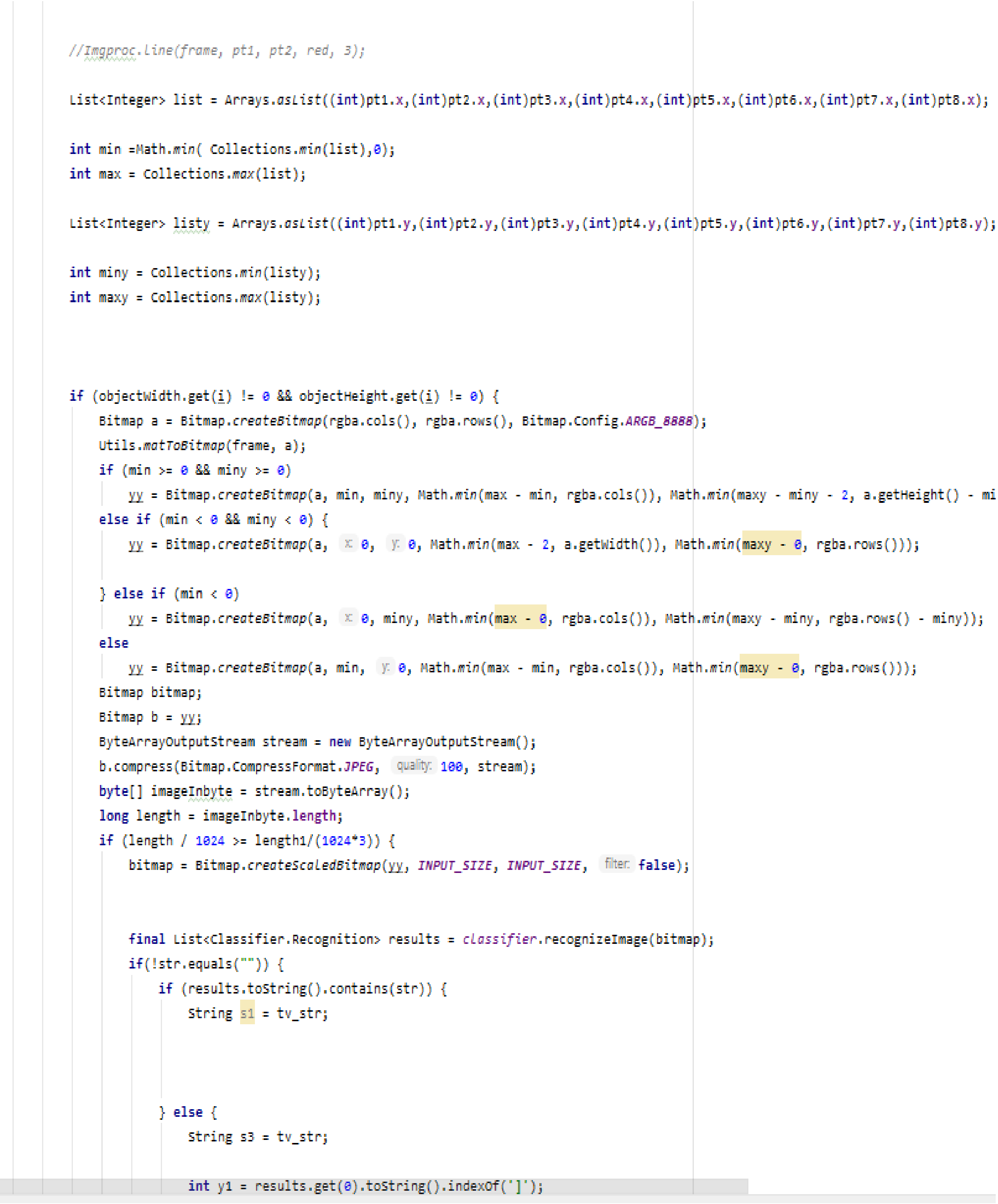
```
////////////////////
public String crop_objects(Context context,Bitmap bmp,String str,String tv_str)

{



    Bitmap bb=bmp;
    ByteArrayOutputStream stream1=new ByteArrayOutputStream();
    bb.compress(Bitmap.CompressFormat.JPEG,  quality: 100,stream1);
    byte[] imageInbyte1=stream1.toByteArray();
    long length1=imageInbyte1.length;



    List<Integer> objectWidth = new ArrayList<>();
    List<Integer> objectHeight = new ArrayList<>();

    Mat rgba = new Mat();
    Utils.bitmapToMat(bmp, rgba);

    // frame is captured as a coloured image
    Mat frame = rgba;
    Mat gray = new Mat();
    Mat canny = new Mat();
    List<MatOfPoint> contours = new ArrayList<>();
    Bitmap yy = null;

    /** Since the Canny algorithm only works on greyscale images and the captured image is
     *  coloured, we transform the captured cam image into a greyscale one
     */
    Imgproc.cvtColor(frame, gray, Imgproc.COLOR_RGB2GRAY);

    // Calculating borders of image using the Canny algorithm
    Imgproc.Canny(gray, canny,  threshold1: 180,  threshold2: 210);

    /** To avoid background noise (given by the camera) that makes the system too sensitive
     *  small variations, the image is blurred to a small extent. Blurring is one of the
     *  required steps before any image transformation because this eliminates small details
     *  that are of no use. Blur is a low-pass filter.
     */
    Imgproc.GaussianBlur(canny, canny, new Size( width: 5,  height: 5),  sigmaX: 5);

    // Calculate the contours
    Imgproc.findContours(canny, contours, new Mat(), Imgproc.RETR_EXTERNAL, Imgproc.CHAIN_APPROX_SIMPLE);

    /** The contours come in different sequences
```



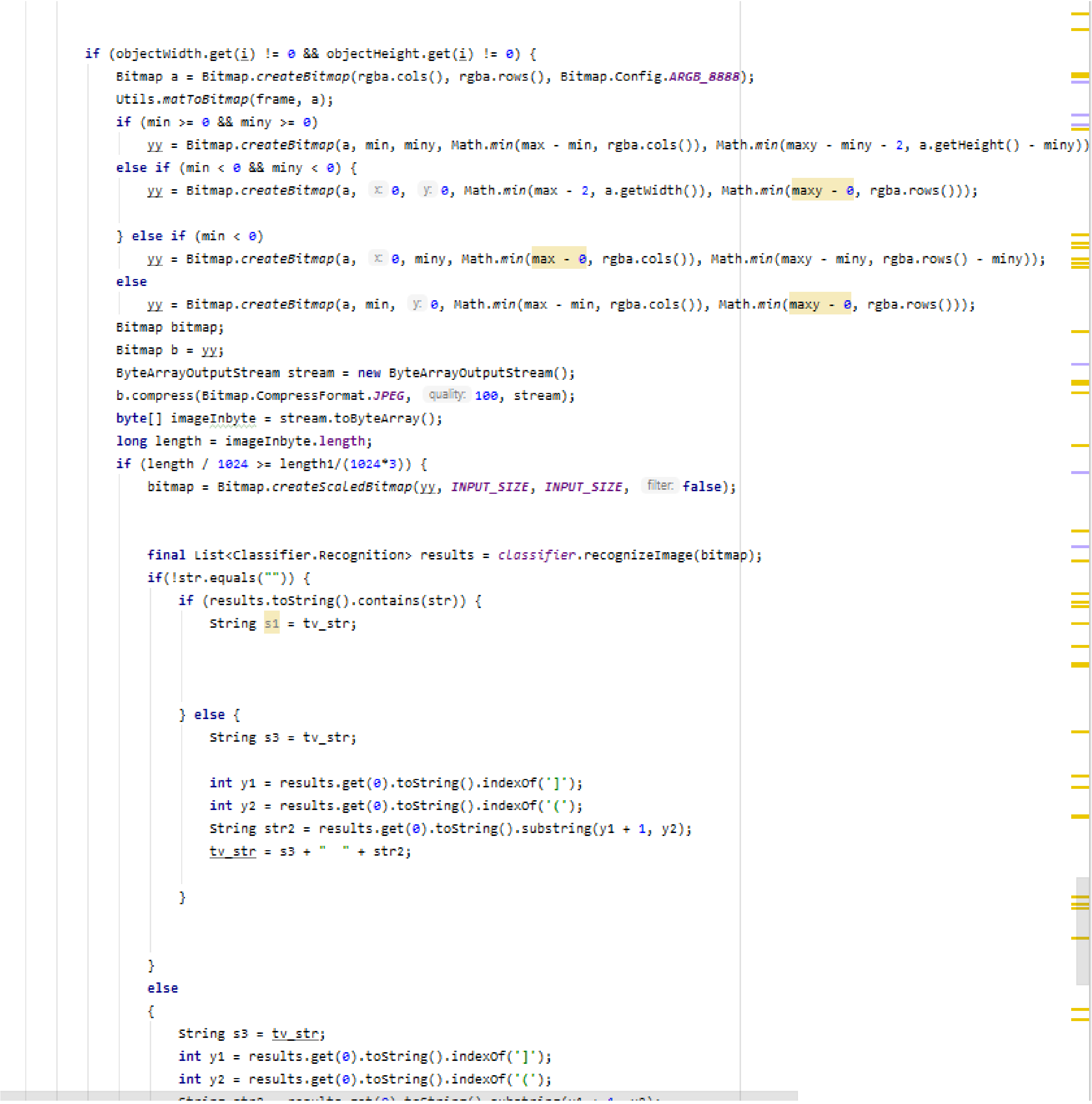
```
 *  required steps before any image transformation because this eliminates small details
 *  that are of no use. Blur is a low-pass filter.
 */
Imgproc.GaussianBlur(canny, canny, new Size( width: 5,  height: 5),  sigmaX: 5);

// Calculate the contours
Imgproc.findContours(canny, contours, new Mat(), Imgproc.RETR_EXTERNAL, Imgproc.CHAIN_APPROX_SIMPLE);

/** The contours come in different sequences
 *  1 sequence for each connected component.
 *  Taking the assumption only 1 object is in view, if we have more than 1 connected
 *  component, this'll be considered part of the details of the object.
 *
 *  For this, we put all contours together in a single sequence
 *  If there is at least 1 contour, I can continue processing
 */
ArrayList<Point> v1=new ArrayList<>();
ArrayList<Point> v2=new ArrayList<>();
if(contours.size() > 0){
    // Calculating the minimal rectangle to contain the contours
    List<RotatedRect> boxes = new ArrayList<>();
    for(MatOfPoint contour : contours){
        RotatedRect box = Imgproc.minAreaRect(new MatOfPoint2f(contour.toArray()));
        boxes.add(box);
    }

    // Getting the vertices of the rectangle

    List<Point[]> vertices = initialiseWithDefaultPointInstances(boxes.size(),  n_Points: 4);
    for(int i=0; i<boxes.size(); i++){
        boxes.get(i).points(vertices.get(i));
    }

/*
double alpha = 0.5;
// Now the vertices are in possession, temporal smoothing can be performed.
    for(int i = 0; i<vertices.size(); i++){
        for (int j = 0; j < 4; j++) {
            // Smooth coordinate x of the vertex
            vertices.get(i)[j].x = alpha * lastVertices.get(i)[j].x + (1.0 - alpha) * vertices.get(i)[j].x;
            // Smooth coordinate y of the vertex
            vertices.get(i)[j].y = alpha * lastVertices.get(i)[j].y + (1.0 - alpha) * vertices.get(i)[j].y;
            // Assign the present smoothed values as lastVertices for the next smooth
            lastVertices.get(i)[j] = vertices.get(i)[j];
        }
}*/

    /** With the vertices, the object size is calculated.
     *  The object size is calculated through pythagoras theorm. In addition, it gives
     *  the distance between 2 points in a bi-dimensional space.
```



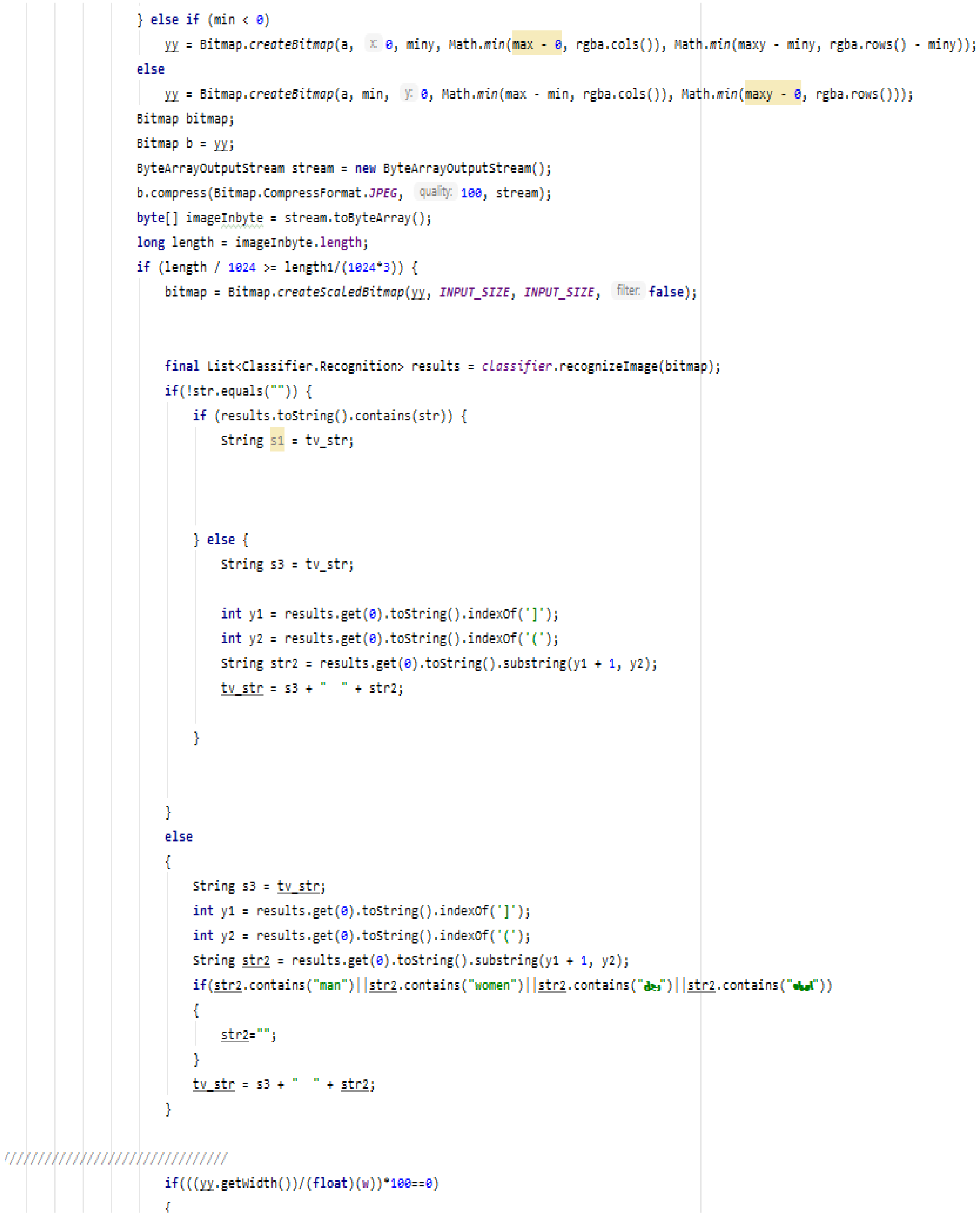
```
for(Point[] points : vertices){
    int width = (int) (conversionFactor * Math.sqrt((points[0].x - points[3].x) * (points[0].x - points[3].x) + (points[0].y - points
    int height = (int) (conversionFactor * Math.sqrt((points[0].x - points[1].x) * (points[0].x - points[1].x) + (points[0].y - point
    objectWidth.add(width);
    objectHeight.add(height);
}



/** Draw the rectangle containing the contours. The Line method draws a line from 1
 * point to the next, and accepts only integer coordinates; for this reason, 2
 * temporary Points have been created and why I used Math.round method.
 */

Scalar red = new Scalar(255, 0, 0, 255);
for (int i=0; i<vertices.size(); i++){
    Point pt1 = new Point();
    Point pt2 = new Point();
    Point pt3 = new Point();
    Point pt4 = new Point();
    Point pt5 = new Point();
    Point pt6 = new Point();
    Point pt7 = new Point();
    Point pt8 = new Point();

    pt1.x = Math.round(vertices.get(i)[0].x);
    pt1.y = Math.round(vertices.get(i)[0].y);
    pt2.x = Math.round(vertices.get(i)[(0 + 1) % 4].x);
    pt2.y = Math.round(vertices.get(i)[(0 + 1) % 4].y);
    pt3.x = Math.round(vertices.get(i)[1].x);
    pt3.y = Math.round(vertices.get(i)[1].y);
    pt4.x = Math.round(vertices.get(i)[(1 + 1) % 4].x);
    pt4.y = Math.round(vertices.get(i)[(1 + 1) % 4].y);
    pt5.x = Math.round(vertices.get(i)[2].x);
    pt5.y = Math.round(vertices.get(i)[2].y);
    pt6.x = Math.round(vertices.get(i)[(2 + 1) % 4].x);
    pt6.y = Math.round(vertices.get(i)[(2+ 1) % 4].y);
    pt7.x = Math.round(vertices.get(i)[3].x);
    pt7.y = Math.round(vertices.get(i)[3].y);
    pt8.x = Math.round(vertices.get(i)[(3 + 1) % 4].x);
    pt8.y = Math.round(vertices.get(i)[(3 + 1) % 4].y);



    //Imgproc.line(frame, pt1, pt2, red, 3);

    List<Integer> list = Arrays.asList((int)pt1.x,(int)pt2.x,(int)pt3.x,(int)pt4.x,(int)pt5.x,(int)pt6.x,(int)pt7.x,(int)pt8.x);

    int min =Math.min( Collections.min(list),0);
    int max = Collections.max(list);
```



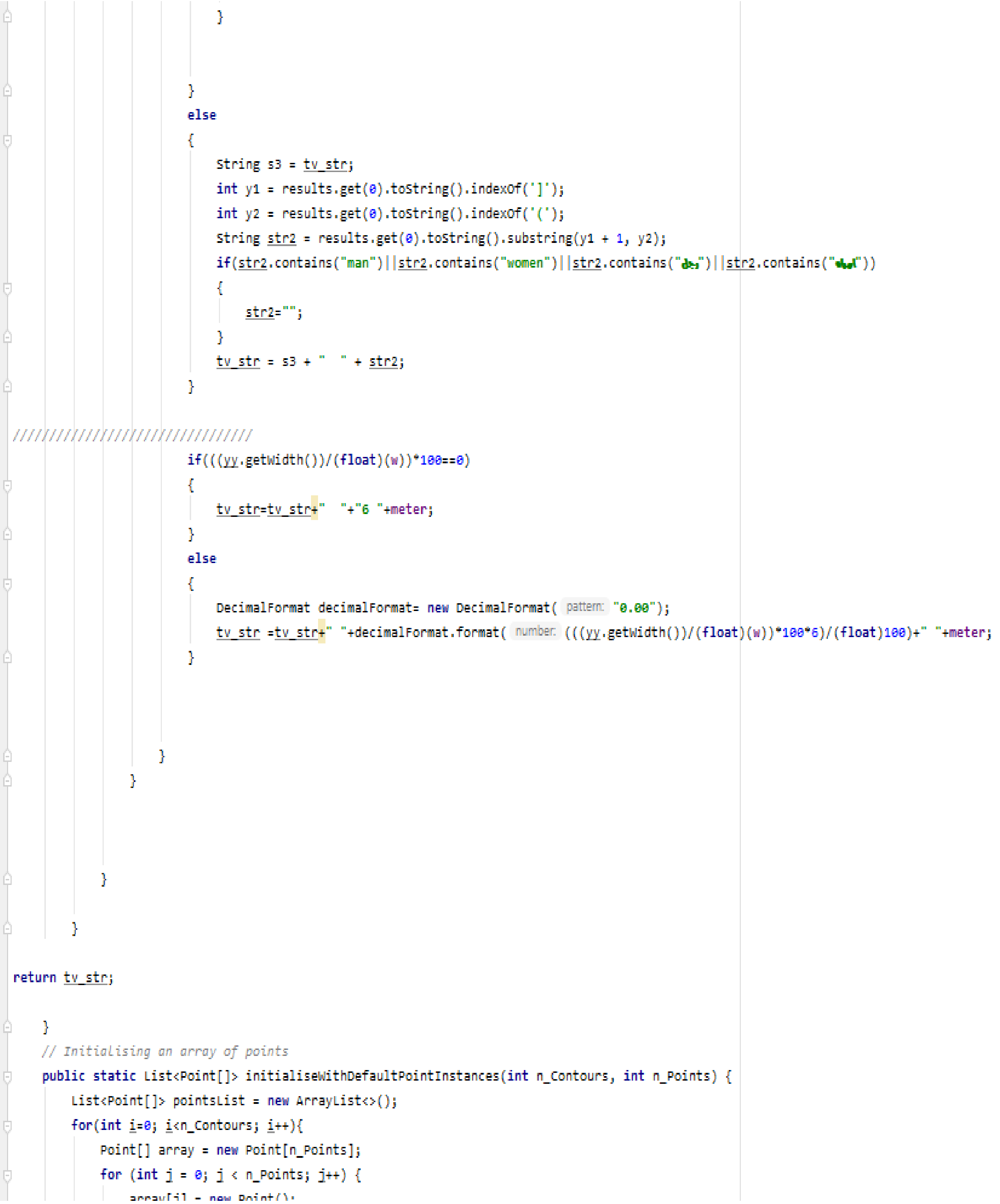
```
//Imgproc.line(frame, pt1, pt2, red, 3);

List<Integer> list = Arrays.asList((int)pt1.x,(int)pt2.x,(int)pt3.x,(int)pt4.x,(int)pt5.x,(int)pt6.x,(int)pt7.x,(int)pt8.x);

int min =Math.min( Collections.min(list),0);
int max = Collections.max(list);

List<Integer> listy = Arrays.asList((int)pt1.y,(int)pt2.y,(int)pt3.y,(int)pt4.y,(int)pt5.y,(int)pt6.y,(int)pt7.y,(int)pt8.y);

int miny = Collections.min(listy);
int maxy = Collections.max(listy);



if (objectWidth.get(i) != 0 && objectHeight.get(i) != 0) {
    Bitmap a = Bitmap.createBitmap(rgba.cols(), rgba.rows(), Bitmap.Config.ARGB_8888);
    Utils.matToBitmap(frame, a);
    if (min >= 0 && miny >= 0)
        yy = Bitmap.createBitmap(a, min, miny, Math.min(max - min, rgba.cols()), Math.min(maxy - miny - 2, a.getHeight() - mi
    else if (min < 0 && miny < 0) {
        yy = Bitmap.createBitmap(a,  x: 0,  y: 0, Math.min(max - 2, a.getWidth()), Math.min(maxy - 0, rgba.rows()));

    } else if (min < 0)
        yy = Bitmap.createBitmap(a,  x: 0, miny, Math.min(max - 0, rgba.cols()), Math.min(maxy - miny, rgba.rows() - miny));
    else
        yy = Bitmap.createBitmap(a, min,  y: 0, Math.min(max - min, rgba.cols()), Math.min(maxy - 0, rgba.rows()));
    Bitmap bitmap;
    Bitmap b = yy;
    ByteArrayOutputStream stream = new ByteArrayOutputStream();
    b.compress(Bitmap.CompressFormat.JPEG,  quality: 100, stream);
    byte[] imageInbyte = stream.toByteArray();
    long length = imageInbyte.length;
    if (length / 1024 >= length1/(1024*3)) {
        bitmap = Bitmap.createScaledBitmap(yy, INPUT_SIZE, INPUT_SIZE,  filter: false);


        final List<Classifier.Recognition> results = classifier.recognizeImage(bitmap);
        if(!str.equals("")) {
            if (results.toString().contains(str)) {
                String s1 = tv_str;



            } else {
                String s3 = tv_str;

                int y1 = results.get(0).toString().indexOf(']');
```



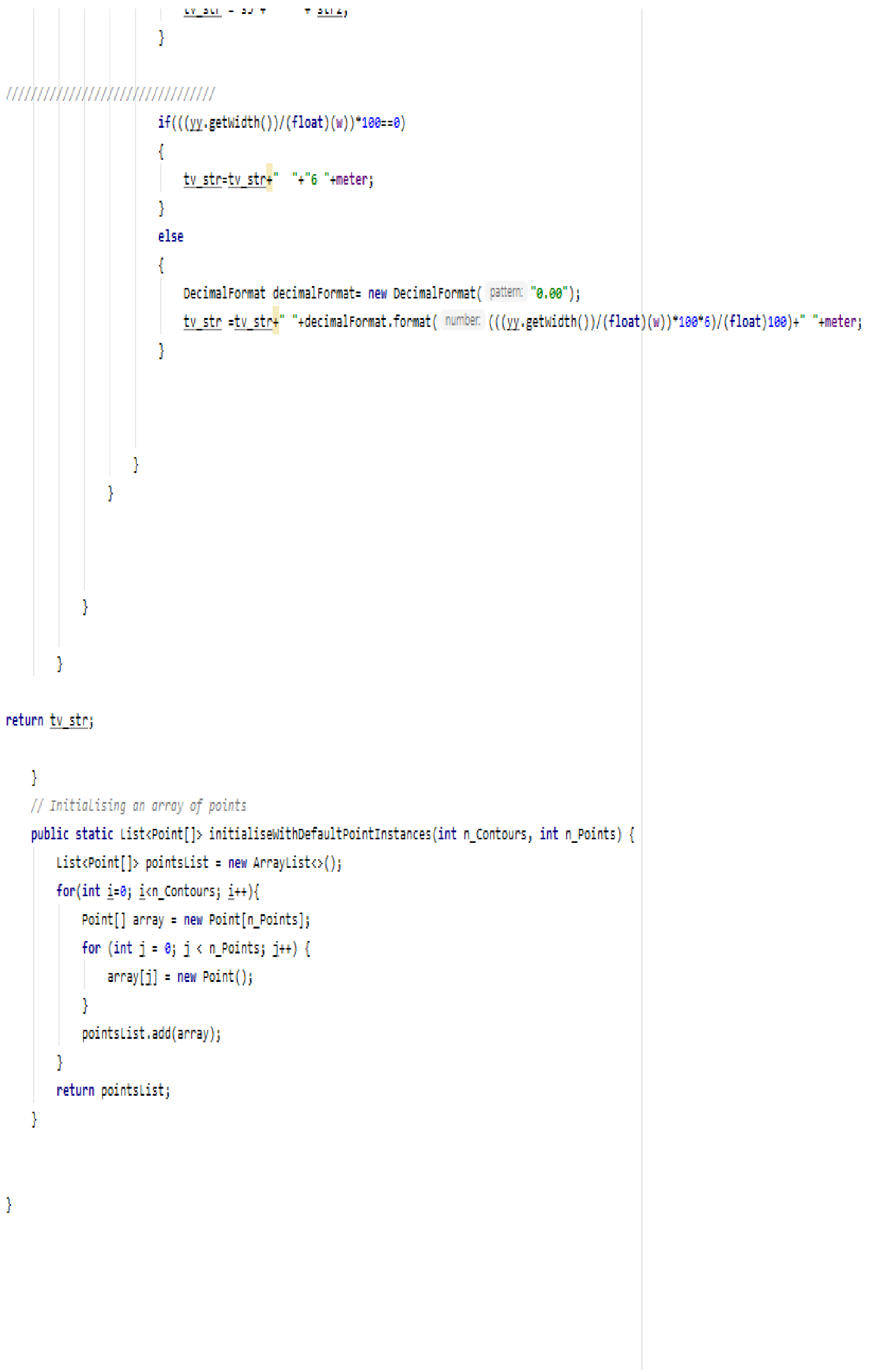
```
if (objectWidth.get(i) != 0 && objectHeight.get(i) != 0) {
    Bitmap a = Bitmap.createBitmap(rgba.cols(), rgba.rows(), Bitmap.Config.ARGB_8888);
    Utils.matToBitmap(frame, a);
    if (min >= 0 && miny >= 0)
        yy = Bitmap.createBitmap(a, min, miny, Math.min(max - min, rgba.cols()), Math.min(maxy - miny - 2, a.getHeight() - miny))
    else if (min < 0 && miny < 0) {
        yy = Bitmap.createBitmap(a,  x: 0,  y: 0, Math.min(max - 2, a.getWidth()), Math.min(maxy - 0, rgba.rows()));

    } else if (min < 0)
        yy = Bitmap.createBitmap(a,  x: 0, miny, Math.min(max - 0, rgba.cols()), Math.min(maxy - miny, rgba.rows() - miny));
    else
        yy = Bitmap.createBitmap(a, min,  y: 0, Math.min(max - min, rgba.cols()), Math.min(maxy - 0, rgba.rows()));
    Bitmap bitmap;
    Bitmap b = yy;
    ByteArrayOutputStream stream = new ByteArrayOutputStream();
    b.compress(Bitmap.CompressFormat.JPEG,  quality: 100, stream);
    byte[] imageInbyte = stream.toByteArray();
    long length = imageInbyte.length;
    if (length / 1024 >= length1/(1024*3)) {
        bitmap = Bitmap.createScaledBitmap(yy, INPUT_SIZE, INPUT_SIZE,  filter: false);


        final List<Classifier.Recognition> results = classifier.recognizeImage(bitmap);
        if(!str.equals("")) {
            if (results.toString().contains(str)) {
                String s1 = tv_str;



            } else {
                String s3 = tv_str;

                int y1 = results.get(0).toString().indexOf(']');
                int y2 = results.get(0).toString().indexOf('(');
                String str2 = results.get(0).toString().substring(y1 + 1, y2);
                tv_str = s3 + "  " + str2;

            }


        }
        else
        {
            String s3 = tv_str;
            int y1 = results.get(0).toString().indexOf(']');
            int y2 = results.get(0).toString().indexOf('(');
            String str2 = results.get(0).toString().substring(y1 + 1, y2);
```



```
} else if (min < 0)
    yy = Bitmap.createBitmap(a,  x: 0, miny, Math.min(max - 0, rgba.cols()), Math.min(maxy - miny, rgba.rows() - miny));
else
    yy = Bitmap.createBitmap(a, min,  y: 0, Math.min(max - min, rgba.cols()), Math.min(maxy - 0, rgba.rows()));
Bitmap bitmap;
Bitmap b = yy;
ByteArrayOutputStream stream = new ByteArrayOutputStream();
b.compress(Bitmap.CompressFormat.JPEG,  quality: 100, stream);
byte[] imageInbyte = stream.toByteArray();
long length = imageInbyte.length;
if (length / 1024 >= length1/(1024*3)) {
    bitmap = Bitmap.createScaledBitmap(yy, INPUT_SIZE, INPUT_SIZE,  filter: false);


    final List<Classifier.Recognition> results = classifier.recognizeImage(bitmap);
    if(!str.equals("")) {
        if (results.toString().contains(str)) {
            String s1 = tv_str;



        } else {
            String s3 = tv_str;

            int y1 = results.get(0).toString().indexOf(']');
            int y2 = results.get(0).toString().indexOf('(');
            String str2 = results.get(0).toString().substring(y1 + 1, y2);
            tv_str = s3 + "  " + str2;

        }


    }
    else
    {
        String s3 = tv_str;
        int y1 = results.get(0).toString().indexOf(']');
        int y2 = results.get(0).toString().indexOf('(');
        String str2 = results.get(0).toString().substring(y1 + 1, y2);
        if(str2.contains("man")||str2.contains("women")||str2.contains("رجل")||str2.contains("طفل"))
        {
            str2="";
        }
        tv_str = s3 + "  " + str2;
    }

////////////////////////////////
    if(((yy.getWidth())/(float)(w))*100==0)
    {
```



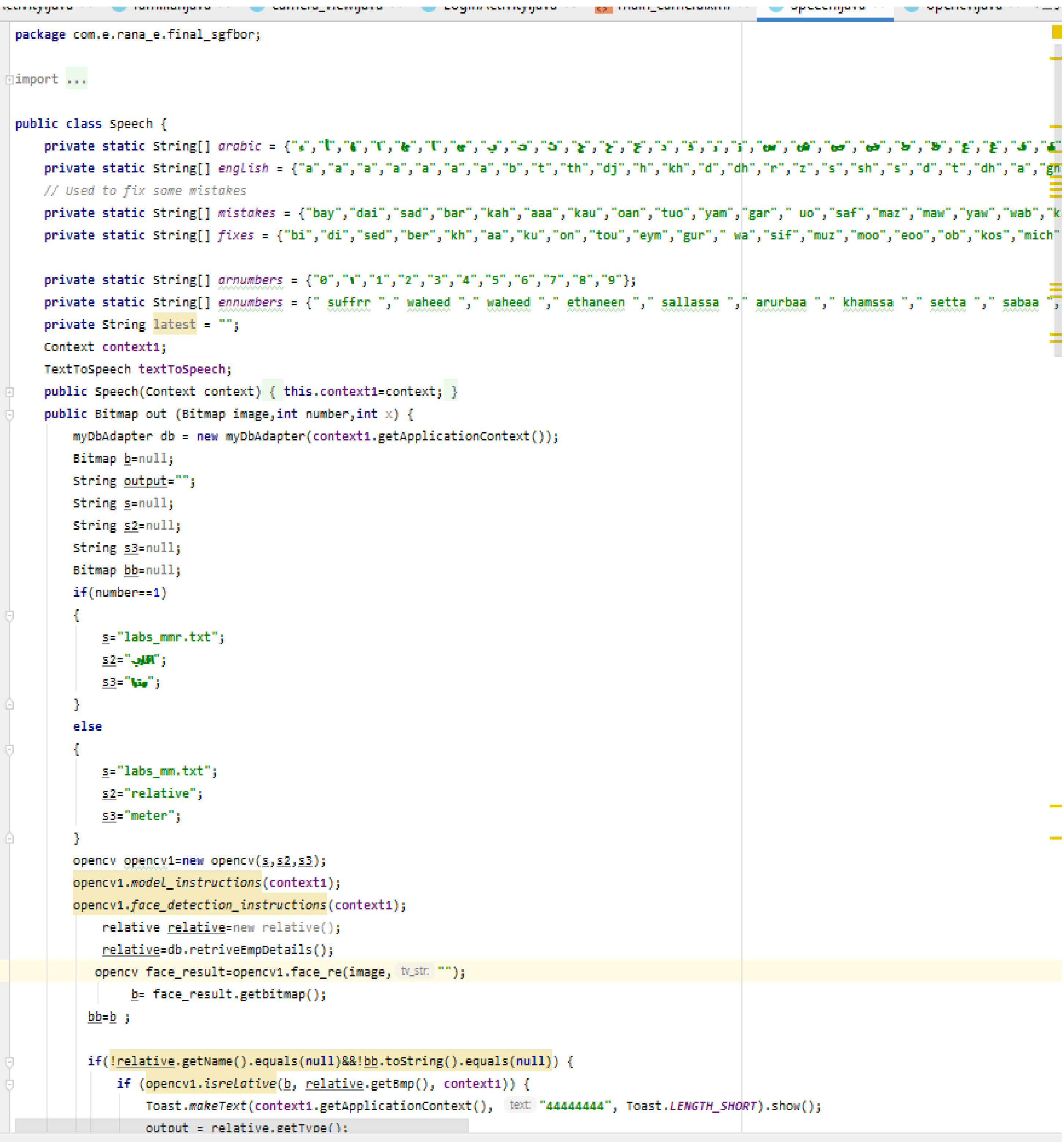
```
                            }


                        }
                        else
                        {
                            String s3 = tv_str;
                            int y1 = results.get(0).toString().indexOf(']');
                            int y2 = results.get(0).toString().indexOf('(');
                            String str2 = results.get(0).toString().substring(y1 + 1, y2);
                            if(str2.contains("man")||str2.contains("women")||str2.contains("رجل")||str2.contains("امرأة"))
                            {
                                str2="";
                            }
                            tv_str = s3 + "  " + str2;
                        }

/////////////////////////////////
                        if(((yy.getWidth())/(float)(w))*100==0)
                        {
                            tv_str=tv_str+"  "+"6 "+meter;
                        }
                        else
                        {
                            DecimalFormat decimalFormat= new DecimalFormat( pattern: "0.00");
                            tv_str =tv_str+" "+decimalFormat.format( number: (((yy.getWidth())/(float)(w))*100*6)/(float)100)+" "+meter;
                        }



                    }
                }



            }

        }

return tv_str;

    }
    // Initialising an array of points
    public static List<Point[]> initialiseWithDefaultPointInstances(int n_Contours, int n_Points) {
        List<Point[]> pointsList = new ArrayList<>();
        for(int i=0; i<n_Contours; i++){
            Point[] array = new Point[n_Points];
            for (int j = 0; j < n_Points; j++) {
                array[j] = new Point();
```



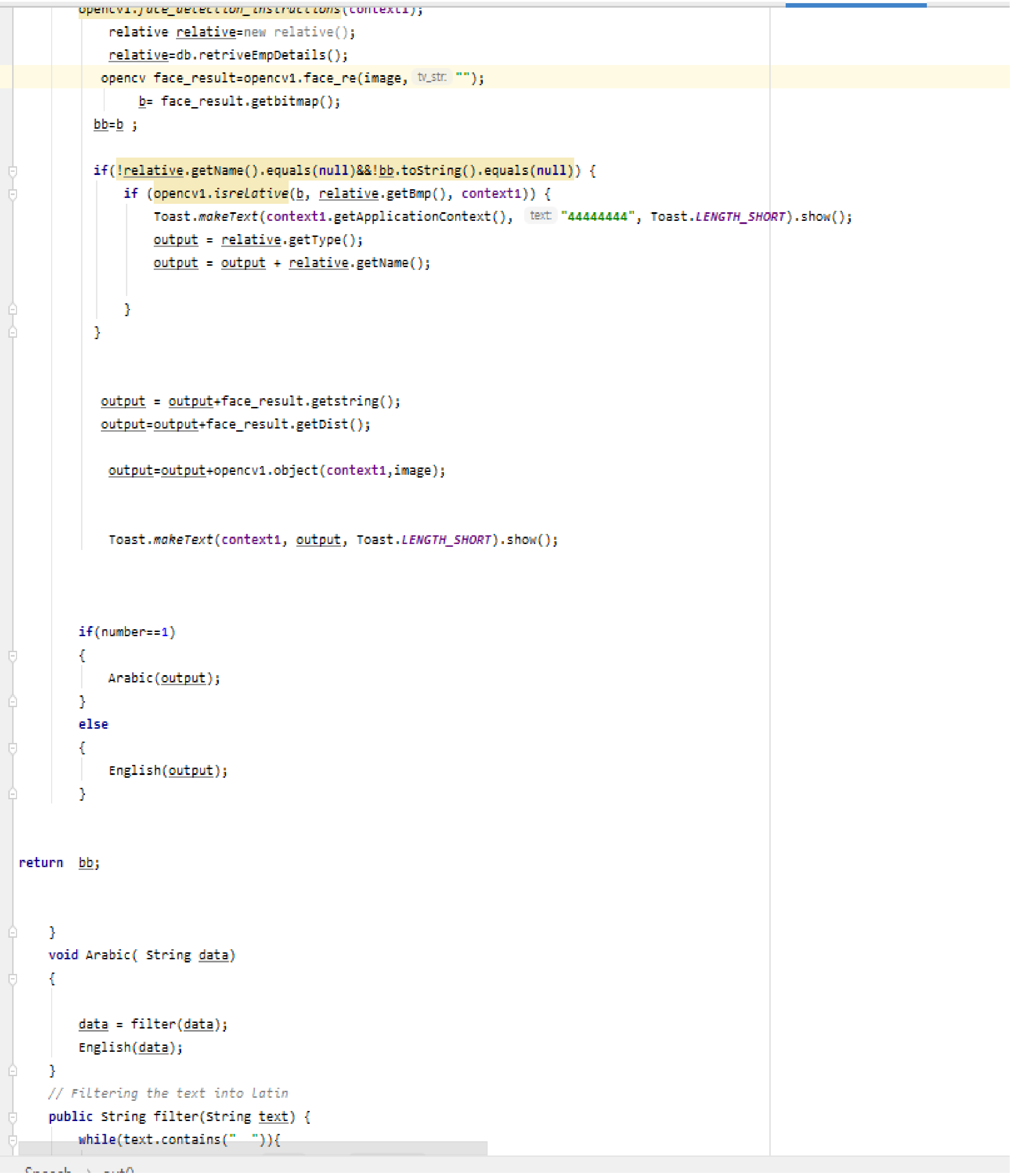
```
                            tv_str = s5 +      + str2;
                        }

////////////////////////////////
                        if(((yy.getWidth())/(float)(w))*100==0)
                        {
                            tv_str=tv_str+"  "+"6 "+meter;
                        }
                        else
                        {
                            DecimalFormat decimalFormat= new DecimalFormat( pattern: "0.00");
                            tv_str =tv_str+" "+decimalFormat.format( number: (((yy.getWidth())/(float)(w))*100*6)/(float)100)+" "+meter;
                        }


                    }
                }


            }

        }

return tv_str;

    }
    // Initialising an array of points
    public static List<Point[]> initialiseWithDefaultPointInstances(int n_Contours, int n_Points) {
        List<Point[]> pointsList = new ArrayList<>();
        for(int i=0; i<n_Contours; i++){
            Point[] array = new Point[n_Points];
            for (int j = 0; j < n_Points; j++) {
                array[j] = new Point();
            }
            pointsList.add(array);
        }
        return pointsList;
    }


}
```

Speech

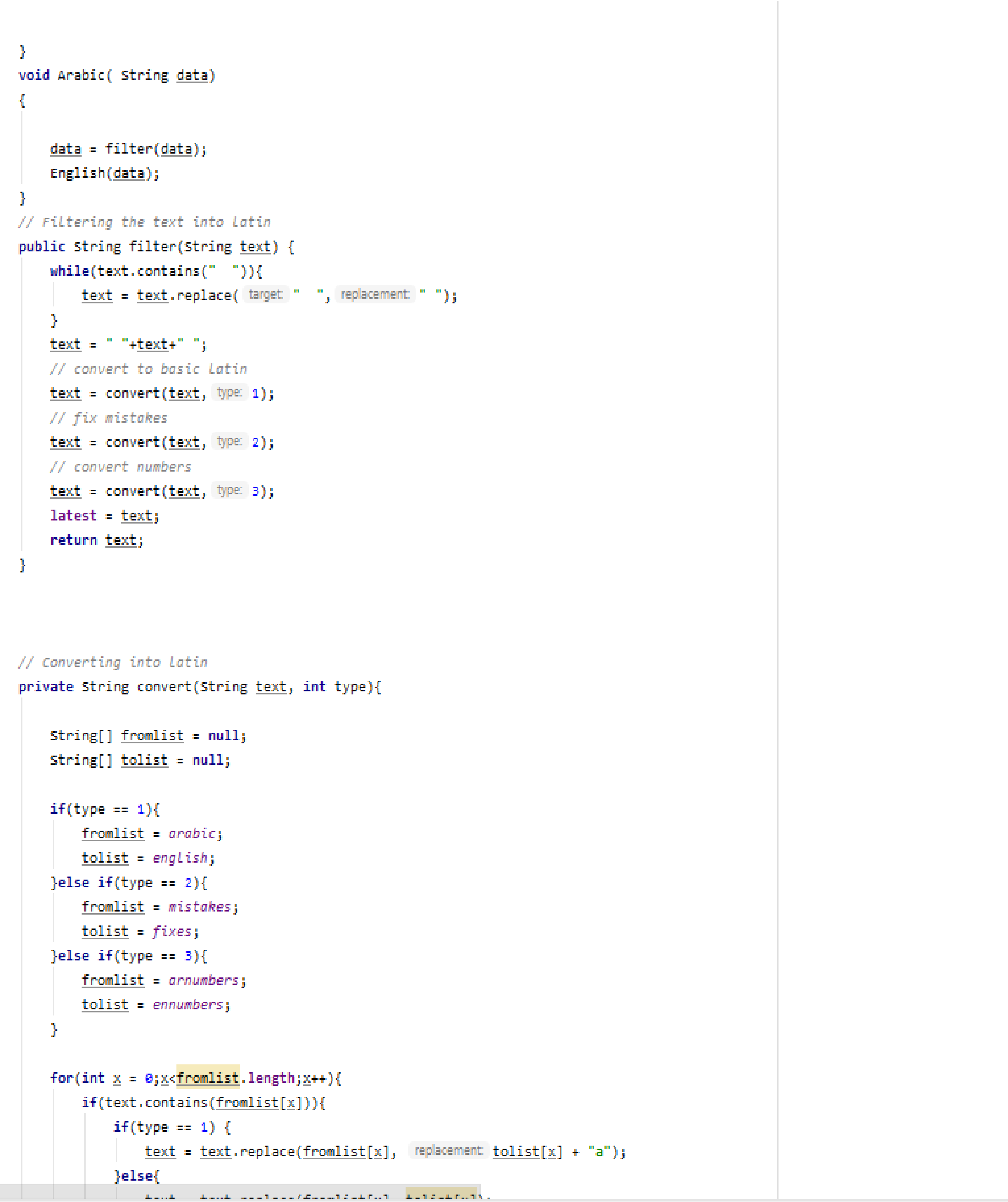
In this activity is the code of speech that converts the text to speech by using the algorithm TTS [12] that take the result of the analyzed image, and convert it to text in Arabic or English depending on the language chooses.



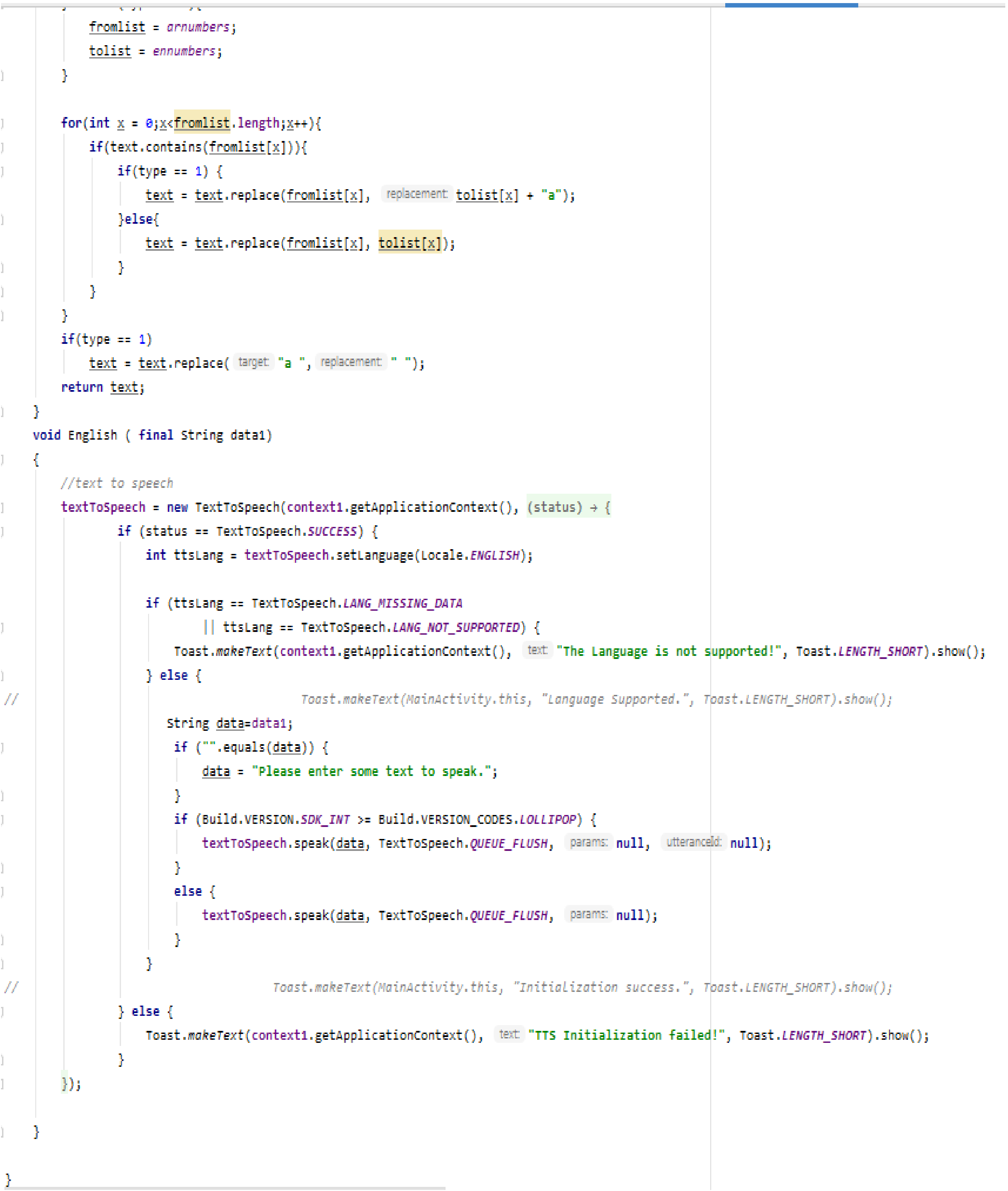
```
package com.e.rana_e.final_sgfbor;

import ...

public class Speech {
    private static String[] arabic = {"ء","آ","أ","ؤ","إ","ئ","ا","ب","ت","ث","ج","ح","خ","د","ذ","ر","ز","س","ش","ص","ض","ط","ظ","ع","غ","ف
    private static String[] english = {"a","a","a","a","a","a","a","b","t","th","dj","h","kh","d","dh","r","z","s","sh","s","d","t","dh","a","gn
    // Used to fix some mistakes
    private static String[] mistakes = {"bay","dai","sad","bar","kah","aaa","kau","oan","tuo","yam","gar"," uo","saf","maz","maw","yaw","wab","k
    private static String[] fixes = {"bi","di","sed","ber","kh","aa","ku","on","tou","eym","gur"," wa","sif","muz","moo","eoo","ob","kos","mich"

    private static String[] arnumbers = {"0","١","1","2","3","4","5","6","7","8","9"};
    private static String[] ennumbers = {" suffrr "," waheed "," waheed "," ethaneen "," sallassa "," arurbaa "," khamssa "," setta "," sabaa ",
    private String latest = "";
    Context context1;
    TextToSpeech textToSpeech;
    public Speech(Context context) { this.context1=context; }
    public Bitmap out (Bitmap image,int number,int x) {
        myDbAdapter db = new myDbAdapter(context1.getApplicationContext());
        Bitmap b=null;
        String output="";
        String s=null;
        String s2=null;
        String s3=null;
        Bitmap bb=null;
        if(number==1)
        {
            s="labs_mmr.txt";
            s2="القريب";
            s3="متر";
        }
        else
        {
            s="labs_mm.txt";
            s2="relative";
            s3="meter";
        }
        opencv opencv1=new opencv(s,s2,s3);
        opencv1.model_instructions(context1);
        opencv1.face_detection_instructions(context1);
            relative relative=new relative();
            relative=db.retriveEmpDetails();
           opencv face_result=opencv1.face_re(image, tv_str: "");
               b= face_result.getbitmap();
          bb=b ;

          if(!relative.getName().equals(null)&&!bb.toString().equals(null)) {
              if (opencv1.isrelative(b, relative.getBmp(), context1)) {
                  Toast.makeText(context1.getApplicationContext(), text: "44444444", Toast.LENGTH_SHORT).show();
                  output = relative.getType();
```



```
opencv1.face_detection_instructions(context1);
    relative relative=new relative();
    relative=db.retriveEmpDetails();
   opencv face_result=opencv1.face_re(image, tv_str: "");
        b= face_result.getbitmap();
  bb=b ;

  if(!relative.getName().equals(null)&&!bb.toString().equals(null)) {
      if (opencv1.isrelative(b, relative.getBmp(), context1)) {
          Toast.makeText(context1.getApplicationContext(), text: "44444444", Toast.LENGTH_SHORT).show();
          output = relative.getType();
          output = output + relative.getName();

      }
  }


   output = output+face_result.getstring();
   output=output+face_result.getDist();

    output=output+opencv1.object(context1,image);


    Toast.makeText(context1, output, Toast.LENGTH_SHORT).show();



        if(number==1)
        {
            Arabic(output);
        }
        else
        {
            English(output);
        }


return  bb;


    }
    void Arabic( String data)
    {

        data = filter(data);
        English(data);
    }
    // Filtering the text into latin
    public String filter(String text) {
        while(text.contains("  ")){
```



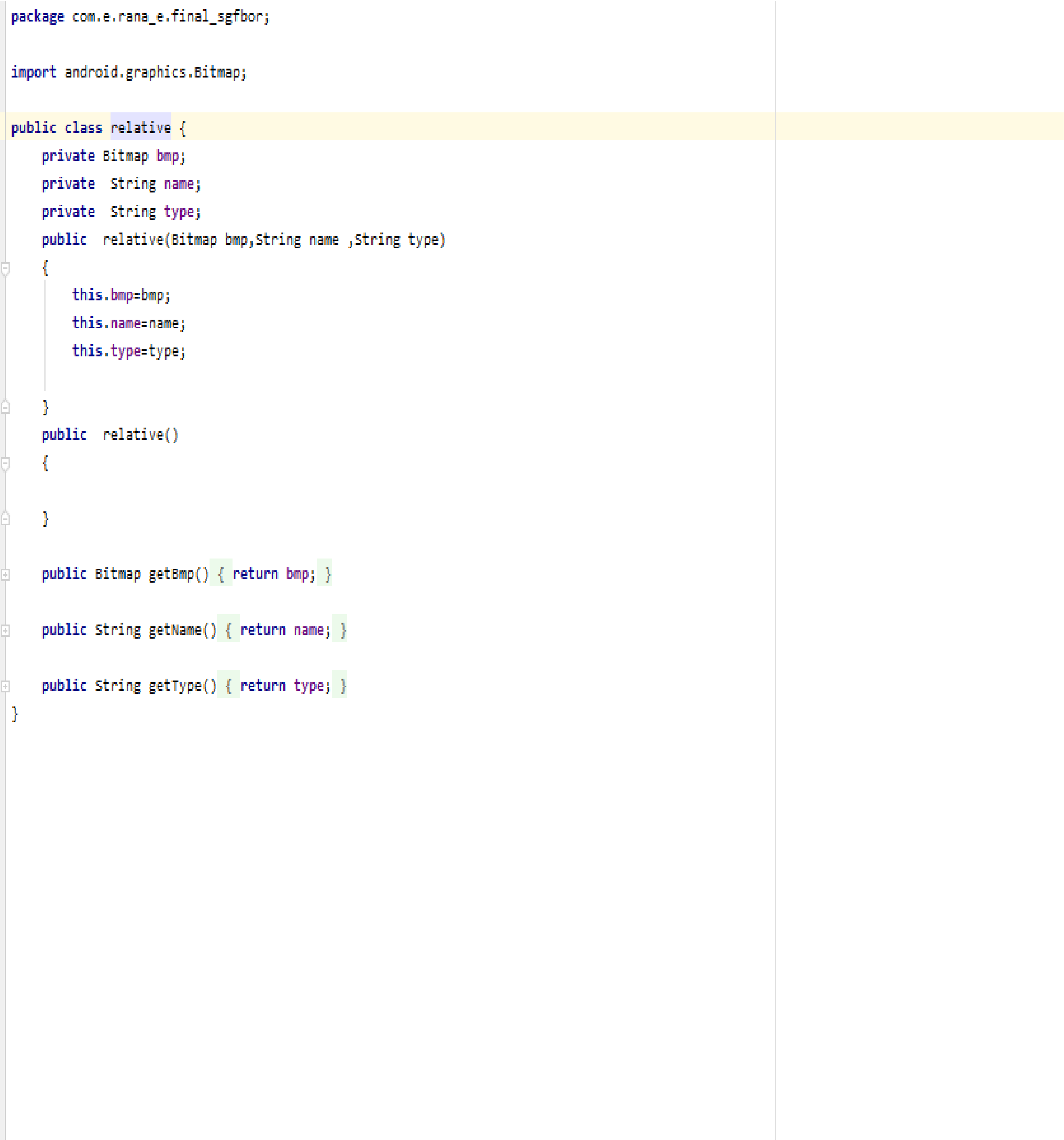
```
}
void Arabic( String data)
{

    data = filter(data);
    English(data);
}
// Filtering the text into latin
public String filter(String text) {
    while(text.contains("  ")){
        text = text.replace( target: "  ", replacement: " ");
    }
    text = " "+text+" ";
    // convert to basic latin
    text = convert(text, type: 1);
    // fix mistakes
    text = convert(text, type: 2);
    // convert numbers
    text = convert(text, type: 3);
    latest = text;
    return text;
}



// Converting into latin
private String convert(String text, int type){

    String[] fromlist = null;
    String[] tolist = null;

    if(type == 1){
        fromlist = arabic;
        tolist = english;
    }else if(type == 2){
        fromlist = mistakes;
        tolist = fixes;
    }else if(type == 3){
        fromlist = arnumbers;
        tolist = ennumbers;
    }

    for(int x = 0;x<fromlist.length;x++){
        if(text.contains(fromlist[x])){
            if(type == 1) {
                text = text.replace(fromlist[x], replacement: tolist[x] + "a");
            }else{
                text = text.replace(fromlist[x], tolist[x]);
```



```java
            fromlist = arnumbers;
            tolist = ennumbers;
        }

        for(int x = 0;x<fromlist.length;x++){
            if(text.contains(fromlist[x])){
                if(type == 1) {
                    text = text.replace(fromlist[x],  replacement: tolist[x] + "a");
                }else{
                    text = text.replace(fromlist[x], tolist[x]);
                }
            }
        }
        if(type == 1)
            text = text.replace( target: "a ",  replacement: " ");
        return text;
    }
    void English ( final String data1)
    {
        //text to speech
        textToSpeech = new TextToSpeech(context1.getApplicationContext(), (status) → {
                if (status == TextToSpeech.SUCCESS) {
                    int ttsLang = textToSpeech.setLanguage(Locale.ENGLISH);

                    if (ttsLang == TextToSpeech.LANG_MISSING_DATA
                            || ttsLang == TextToSpeech.LANG_NOT_SUPPORTED) {
                        Toast.makeText(context1.getApplicationContext(),  text: "The Language is not supported!", Toast.LENGTH_SHORT).show();
                    } else {
//                                          Toast.makeText(MainActivity.this, "Language Supported.", Toast.LENGTH_SHORT).show();
                        String data=data1;
                        if ("".equals(data)) {
                            data = "Please enter some text to speak.";
                        }
                        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
                            textToSpeech.speak(data, TextToSpeech.QUEUE_FLUSH,  params: null,  utteranceId: null);
                        }
                        else {
                            textToSpeech.speak(data, TextToSpeech.QUEUE_FLUSH,  params: null);
                        }
                    }
//                                      Toast.makeText(MainActivity.this, "Initialization success.", Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(context1.getApplicationContext(),  text: "TTS Initialization failed!", Toast.LENGTH_SHORT).show();
                }
        });

    }

}
```

Relative

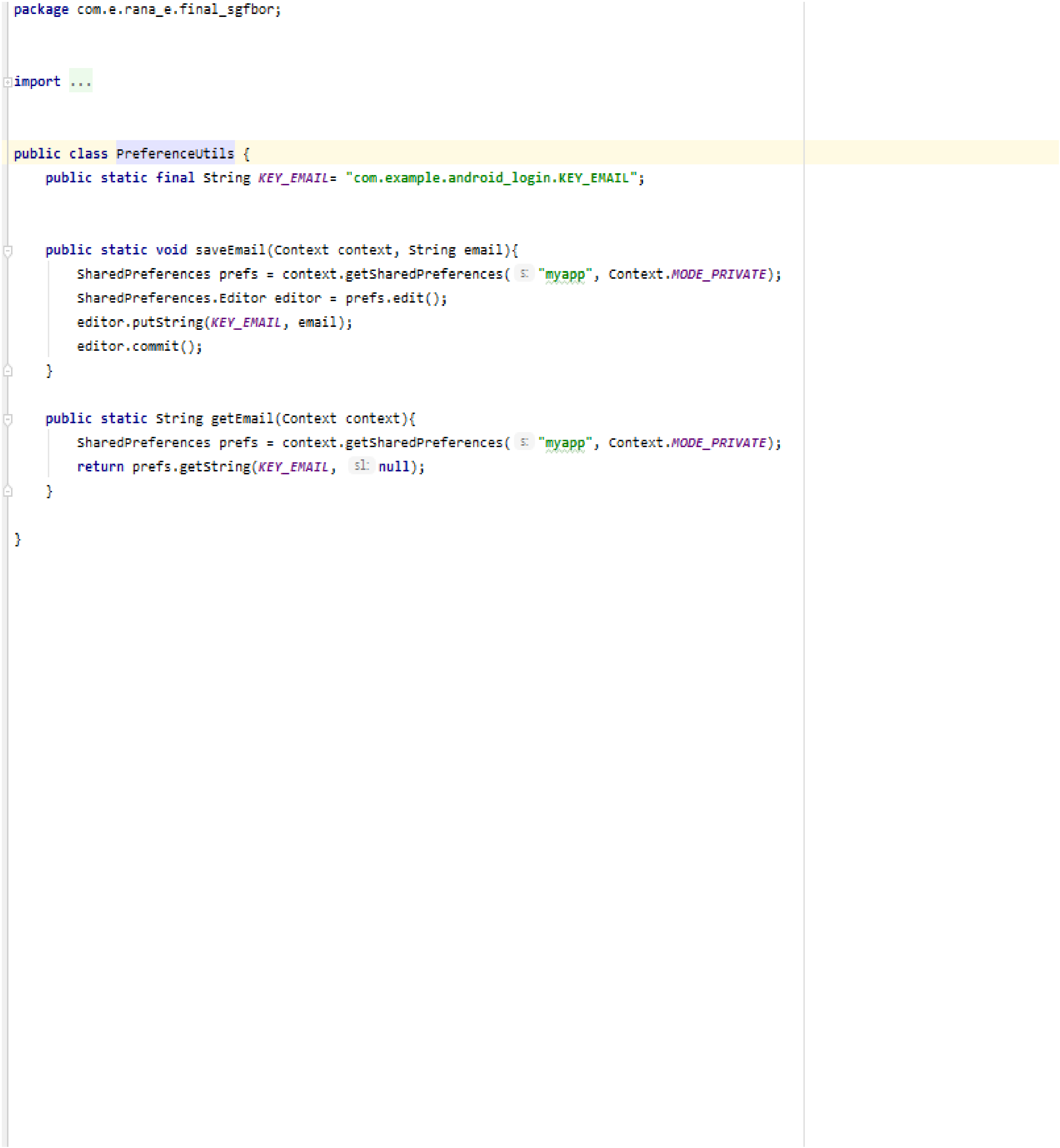
In this activity is the code of relative page to detect if the person is familiar after check in the database exist in it or not.



```
package com.e.rana_e.final_sgfbor;

import android.graphics.Bitmap;

public class relative {
    private Bitmap bmp;
    private  String name;
    private  String type;
    public  relative(Bitmap bmp,String name ,String type)
    {
        this.bmp=bmp;
        this.name=name;
        this.type=type;

    }
    public  relative()
    {

    }

    public Bitmap getBmp() { return bmp; }

    public String getName() { return name; }

    public String getType() { return type; }
}
```

Preference Utils

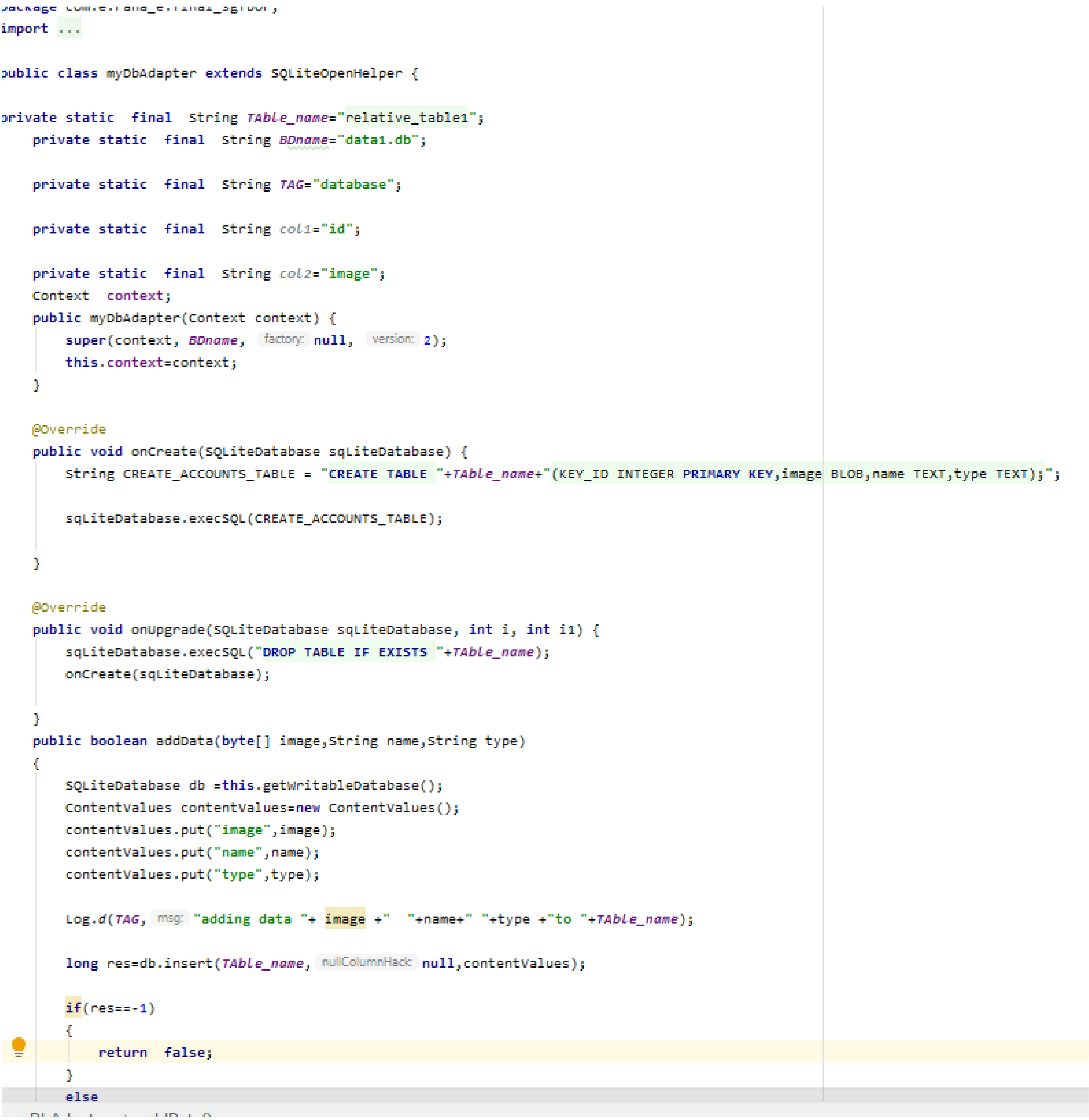
In this activity is preference utils that used to make email, password is saved to login page and the user don't need the enter the email, password every time.



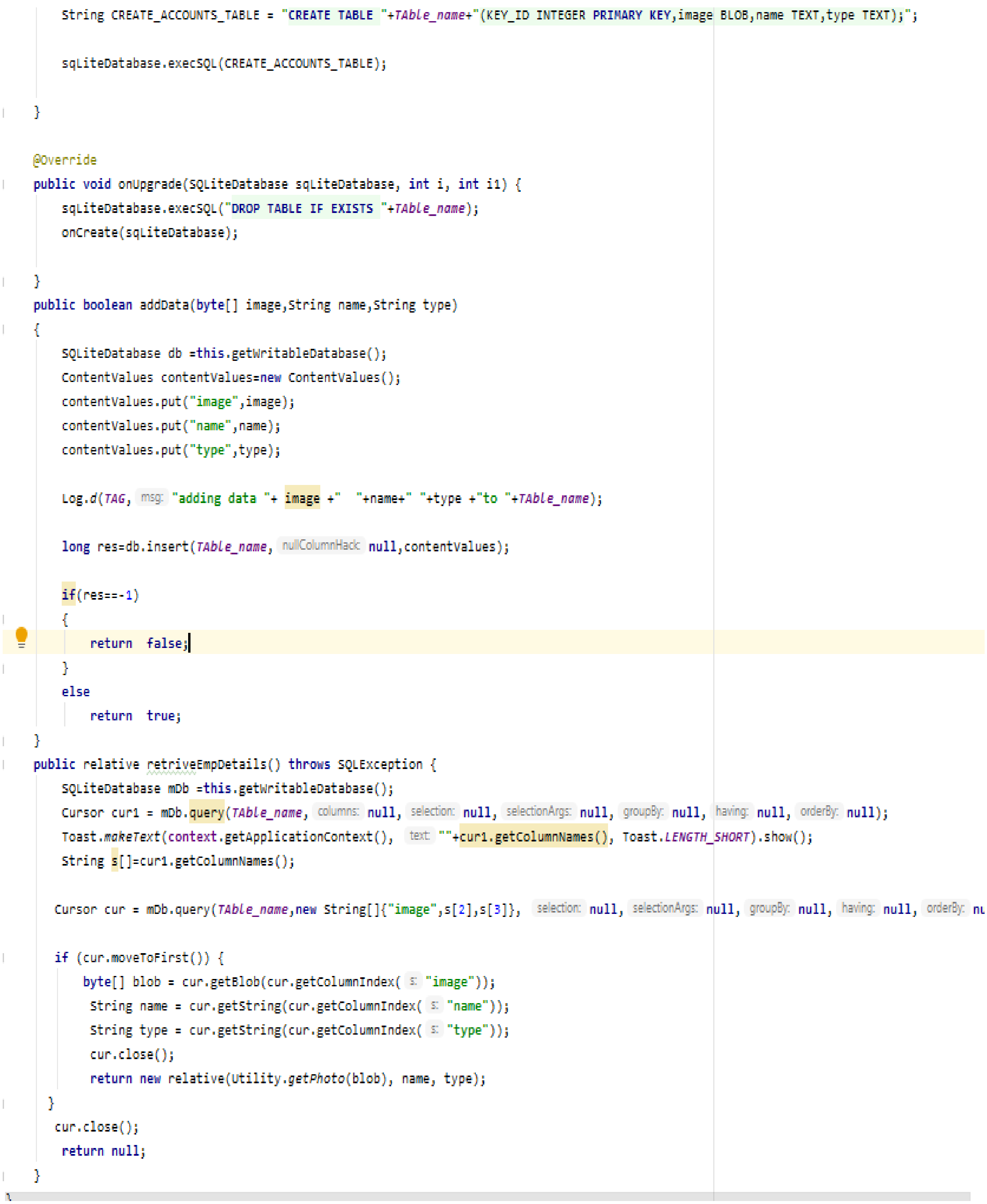
```
package com.e.rana_e.final_sgfbor;


import ...


public class PreferenceUtils {
    public static final String KEY_EMAIL= "com.example.android_login.KEY_EMAIL";


    public static void saveEmail(Context context, String email){
        SharedPreferences prefs = context.getSharedPreferences( s: "myapp", Context.MODE_PRIVATE);
        SharedPreferences.Editor editor = prefs.edit();
        editor.putString(KEY_EMAIL, email);
        editor.commit();
    }

    public static String getEmail(Context context){
        SharedPreferences prefs = context.getSharedPreferences( s: "myapp", Context.MODE_PRIVATE);
        return prefs.getString(KEY_EMAIL,  s1: null);
    }

}
```

Database

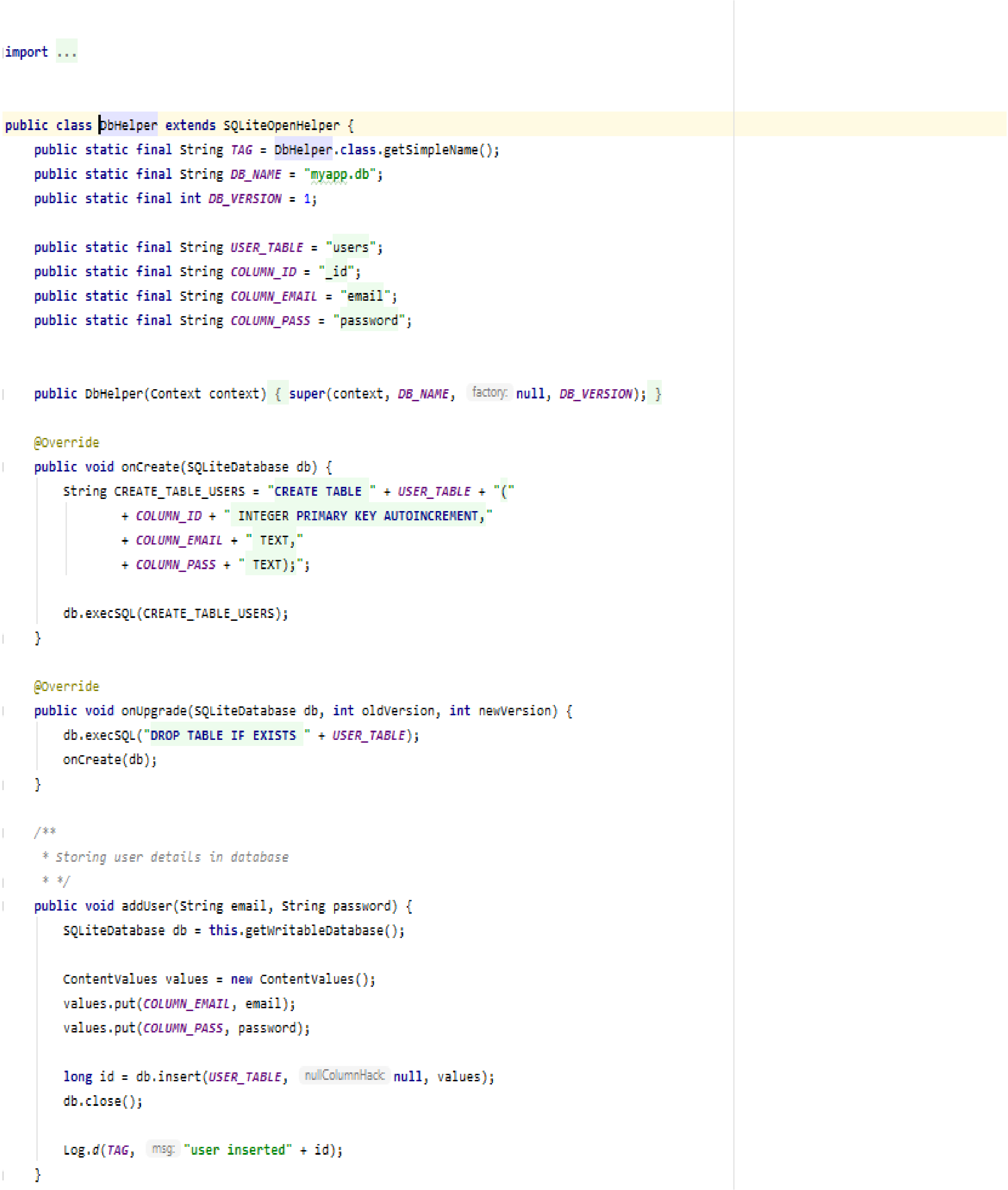
In this activity is the code of database. The database is SQLite Relative database used to save name, type of relation and the image of the familiar person add user used to save E-mail and password for the user.



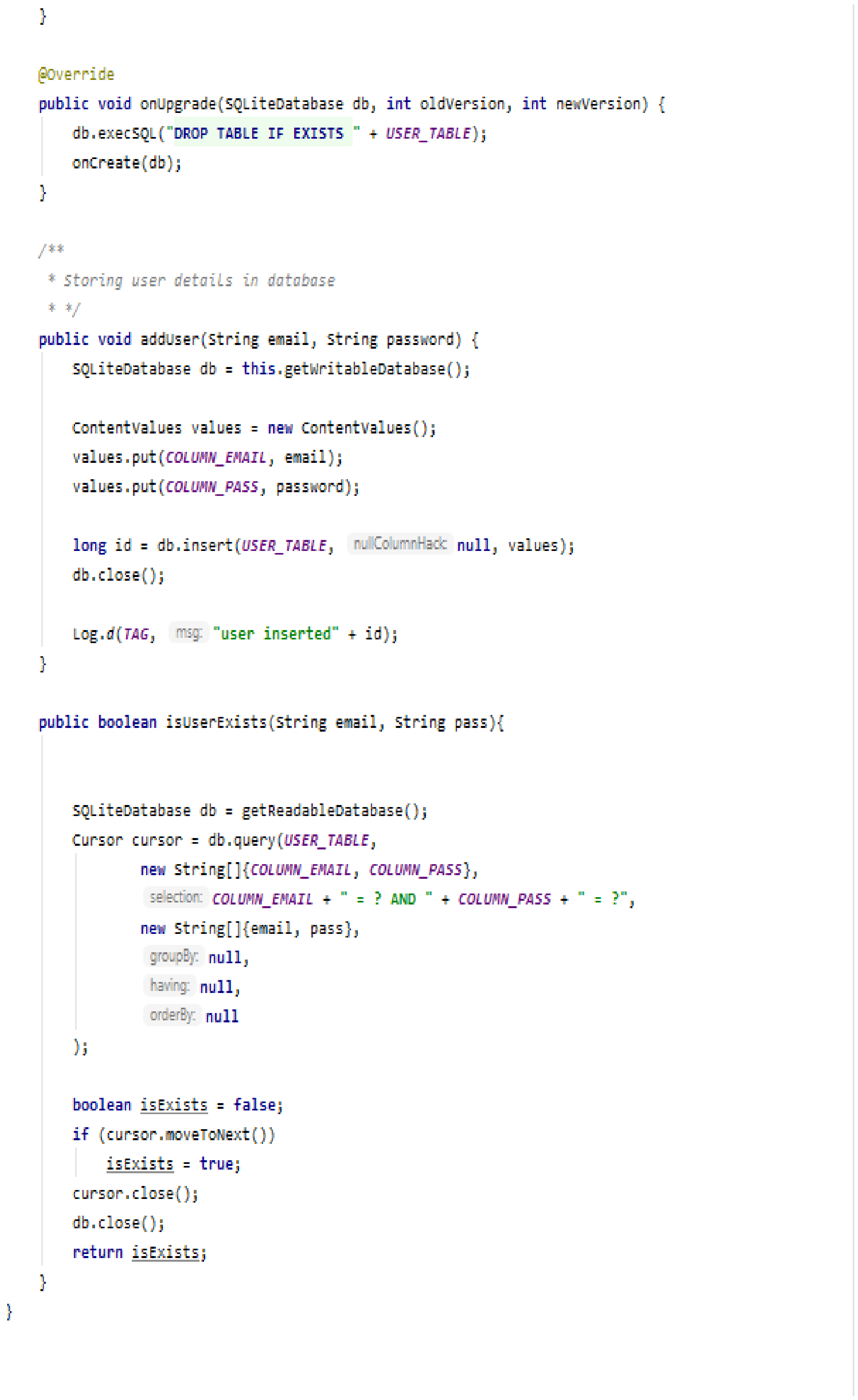
```
package com.e.rana_e.final_sgroup;
import ...

public class myDbAdapter extends SQLiteOpenHelper {

private static  final  String TAble_name="relative_table1";
    private static  final  String BDname="data1.db";

    private static  final  String TAG="database";

    private static  final  String col1="id";

    private static  final  String col2="image";
    Context  context;
    public myDbAdapter(Context context) {
        super(context, BDname,  factory: null,  version: 2);
        this.context=context;
    }

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase) {
        String CREATE_ACCOUNTS_TABLE = "CREATE TABLE "+TAble_name+"(KEY_ID INTEGER PRIMARY KEY,image BLOB,name TEXT,type TEXT);";

        sqLiteDatabase.execSQL(CREATE_ACCOUNTS_TABLE);

    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {
        sqLiteDatabase.execSQL("DROP TABLE IF EXISTS "+TAble_name);
        onCreate(sqLiteDatabase);

    }
    public boolean addData(byte[] image,String name,String type)
    {
        SQLiteDatabase db =this.getWritableDatabase();
        ContentValues contentValues=new ContentValues();
        contentValues.put("image",image);
        contentValues.put("name",name);
        contentValues.put("type",type);

        Log.d(TAG,  msg: "adding data "+ image +"  "+name+" "+type +"to "+TAble_name);

        long res=db.insert(TAble_name,  nullColumnHack: null,contentValues);

        if(res==-1)
        {
            return  false;
        }
        else
```



```
        String CREATE_ACCOUNTS_TABLE = "CREATE TABLE "+TAble_name+"(KEY_ID INTEGER PRIMARY KEY,image BLOB,name TEXT,type TEXT);";

        sqLiteDatabase.execSQL(CREATE_ACCOUNTS_TABLE);

    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {
        sqLiteDatabase.execSQL("DROP TABLE IF EXISTS "+TAble_name);
        onCreate(sqLiteDatabase);

    }
    public boolean addData(byte[] image,String name,String type)
    {
        SQLiteDatabase db =this.getWritableDatabase();
        ContentValues contentValues=new ContentValues();
        contentValues.put("image",image);
        contentValues.put("name",name);
        contentValues.put("type",type);

        Log.d(TAG, msg: "adding data "+ image +"  "+name+" "+type +"to "+TAble_name);

        long res=db.insert(TAble_name, nullColumnHack: null,contentValues);

        if(res==-1)
        {
            return  false;
        }
        else
            return  true;
    }
    public relative retriveEmpDetails() throws SQLException {
        SQLiteDatabase mDb =this.getWritableDatabase();
        Cursor cur1 = mDb.query(TAble_name, columns: null, selection: null, selectionArgs: null, groupBy: null, having: null, orderBy: null);
        Toast.makeText(context.getApplicationContext(), text: ""+cur1.getColumnNames(), Toast.LENGTH_SHORT).show();
        String s[]=cur1.getColumnNames();

       Cursor cur = mDb.query(TAble_name,new String[]{"image",s[2],s[3]}, selection: null, selectionArgs: null, groupBy: null, having: null, orderBy: nu

       if (cur.moveToFirst()) {
           byte[] blob = cur.getBlob(cur.getColumnIndex( s: "image"));
            String name = cur.getString(cur.getColumnIndex( s: "name"));
            String type = cur.getString(cur.getColumnIndex( s: "type"));
            cur.close();
            return new relative(Utility.getPhoto(blob), name, type);
      }
       cur.close();
        return null;
    }
```



```
import ...

public class DbHelper extends SQLiteOpenHelper {
    public static final String TAG = DbHelper.class.getSimpleName();
    public static final String DB_NAME = "myapp.db";
    public static final int DB_VERSION = 1;

    public static final String USER_TABLE = "users";
    public static final String COLUMN_ID = "_id";
    public static final String COLUMN_EMAIL = "email";
    public static final String COLUMN_PASS = "password";


    public DbHelper(Context context) { super(context, DB_NAME,  factory: null, DB_VERSION); }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String CREATE_TABLE_USERS = "CREATE TABLE " + USER_TABLE + "("
                + COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT,"
                + COLUMN_EMAIL + " TEXT,"
                + COLUMN_PASS + " TEXT);";

        db.execSQL(CREATE_TABLE_USERS);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + USER_TABLE);
        onCreate(db);
    }

    /**
     * Storing user details in database
     * */
    public void addUser(String email, String password) {
        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues values = new ContentValues();
        values.put(COLUMN_EMAIL, email);
        values.put(COLUMN_PASS, password);

        long id = db.insert(USER_TABLE,  nullColumnHack: null, values);
        db.close();

        Log.d(TAG,  msg: "user inserted" + id);
    }
```

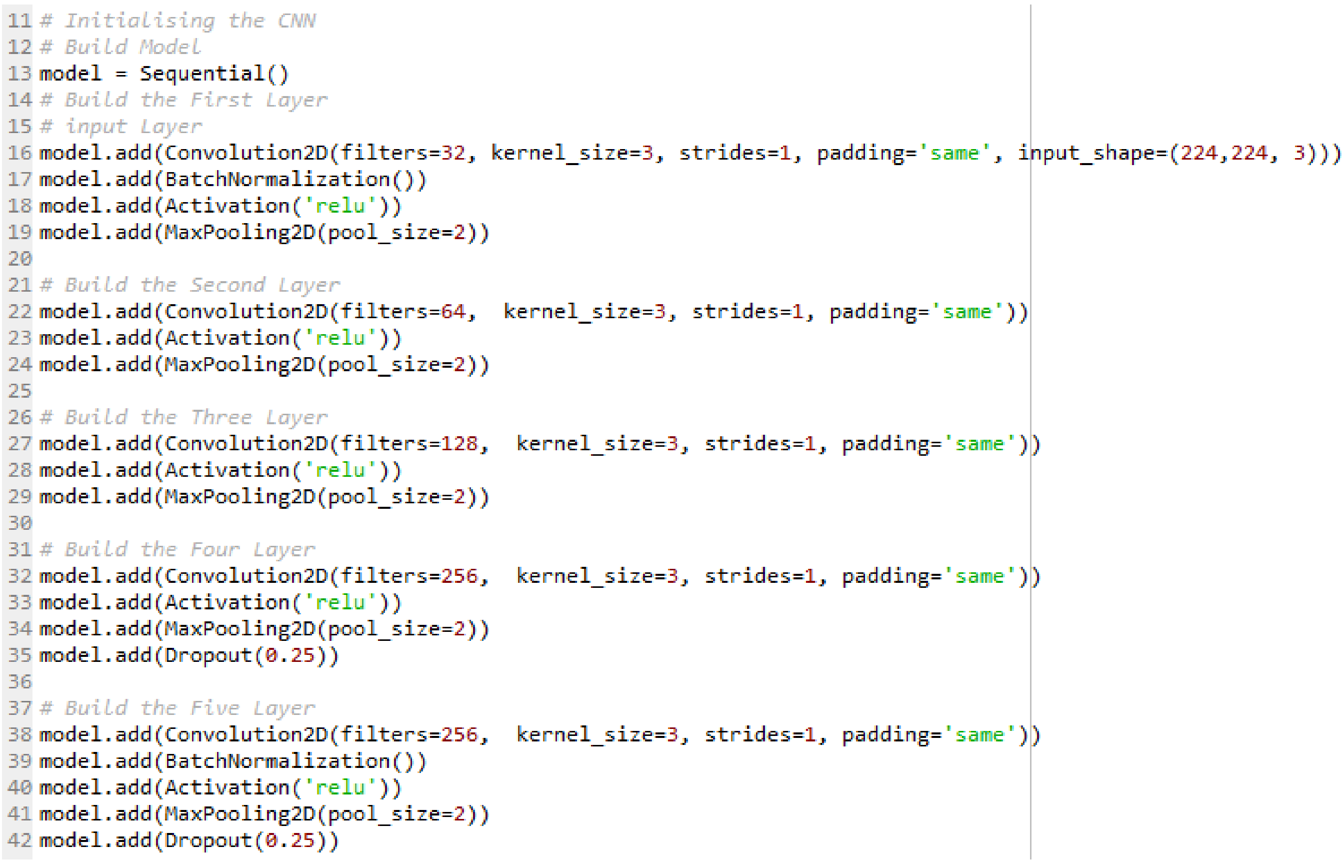


```
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + USER_TABLE);
        onCreate(db);
    }

    /**
     * Storing user details in database
     * */
    public void addUser(String email, String password) {
        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues values = new ContentValues();
        values.put(COLUMN_EMAIL, email);
        values.put(COLUMN_PASS, password);

        long id = db.insert(USER_TABLE,  nullColumnHack: null, values);
        db.close();

        Log.d(TAG,  msg: "user inserted" + id);
    }

    public boolean isUserExists(String email, String pass){


        SQLiteDatabase db = getReadableDatabase();
        Cursor cursor = db.query(USER_TABLE,
                new String[]{COLUMN_EMAIL, COLUMN_PASS},
                 selection: COLUMN_EMAIL + " = ? AND " + COLUMN_PASS + " = ?",
                new String[]{email, pass},
                 groupBy: null,
                 having: null,
                 orderBy: null
        );

        boolean isExists = false;
        if (cursor.moveToNext())
            isExists = true;
        cursor.close();
        db.close();
        return isExists;
    }
}
```

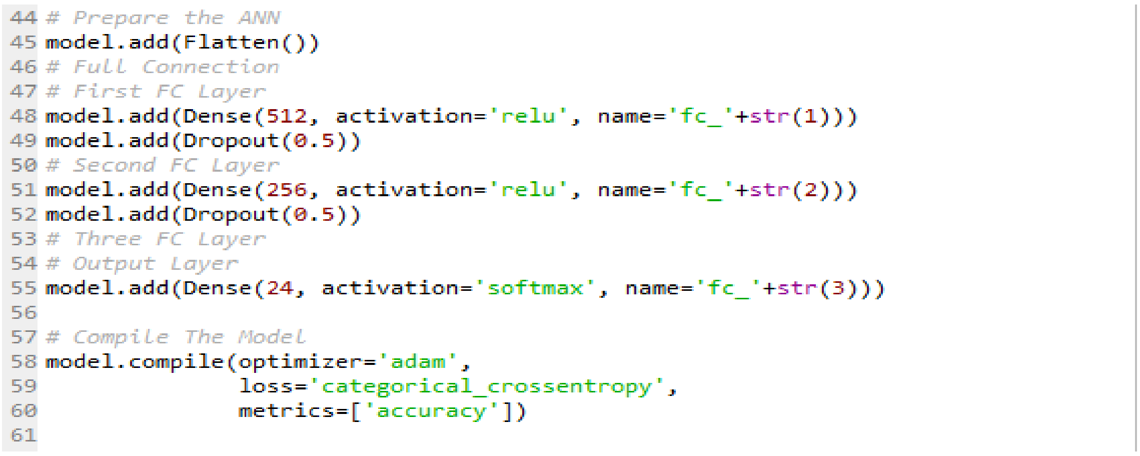
5.2 PYTHON

In the deep learning model [14] use convolution layers, activation functions and max pooling used in four layers that is hidden layers then flatten data and converting it to TensorFlow lite [13].

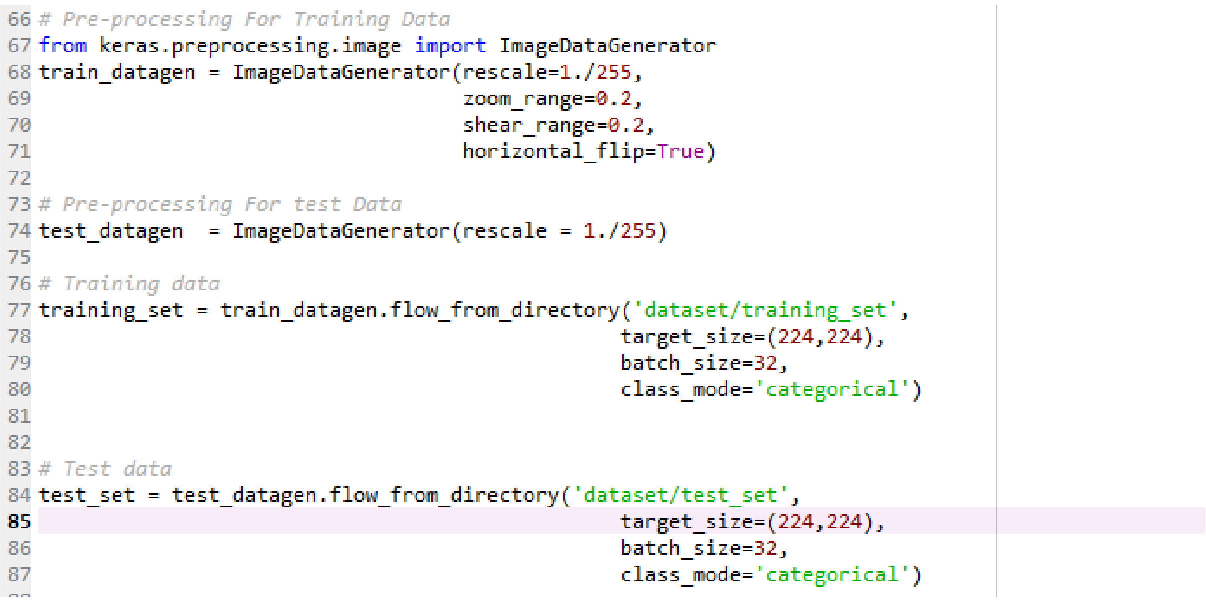
Model Structure



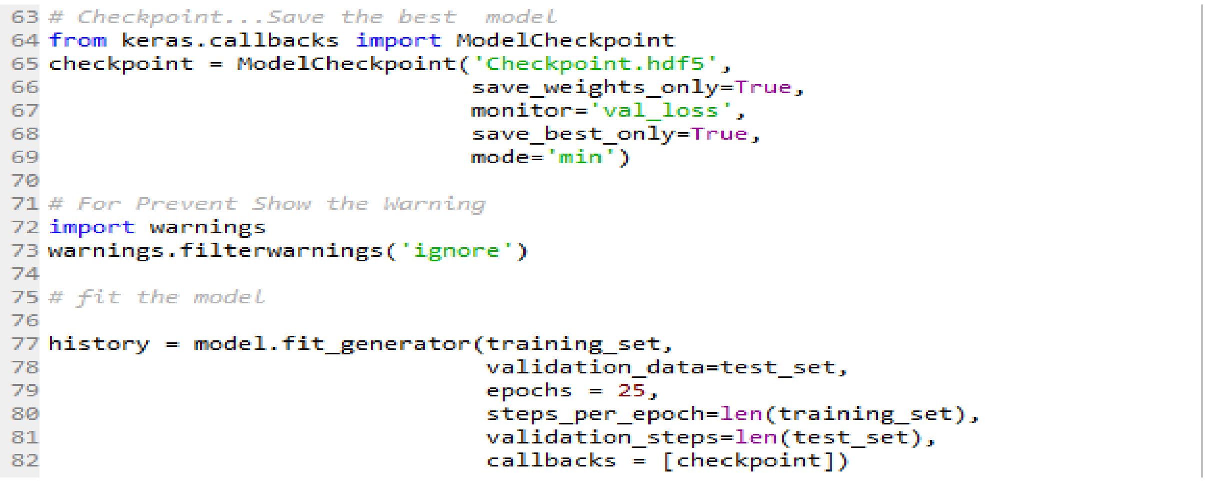
```
# Initialising the CNN
# Build Model
model = Sequential()
# Build the First Layer
# input Layer
model.add(Convolution2D(filters=32, kernel_size=3, strides=1, padding='same', input_shape=(224,224, 3)))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=2))

# Build the Second Layer
model.add(Convolution2D(filters=64,  kernel_size=3, strides=1, padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=2))

# Build the Three Layer
model.add(Convolution2D(filters=128,  kernel_size=3, strides=1, padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=2))

# Build the Four Layer
model.add(Convolution2D(filters=256,  kernel_size=3, strides=1, padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(Dropout(0.25))

# Build the Five Layer
model.add(Convolution2D(filters=256,  kernel_size=3, strides=1, padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(Dropout(0.25))
```



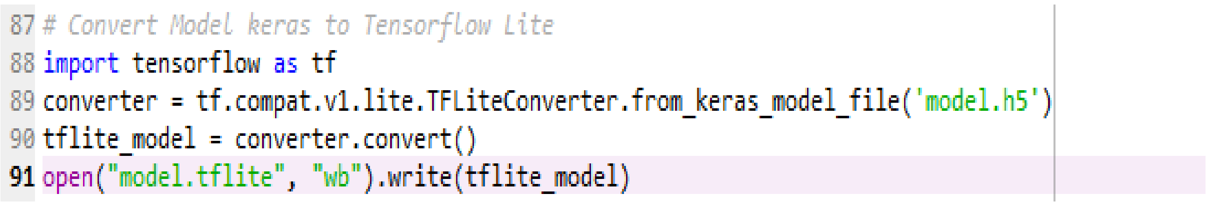
```
# Prepare the ANN
model.add(Flatten())
# Full Connection
# First FC Layer
model.add(Dense(512, activation='relu', name='fc_'+str(1)))
model.add(Dropout(0.5))
# Second FC Layer
model.add(Dense(256, activation='relu', name='fc_'+str(2)))
model.add(Dropout(0.5))
# Three FC Layer
# Output Layer
model.add(Dense(24, activation='softmax', name='fc_'+str(3)))

# Compile The Model
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

```



```python
# Pre-processing For Training Data
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale=1./255,
                                   zoom_range=0.2,
                                   shear_range=0.2,
                                   horizontal_flip=True)

# Pre-processing For test Data
test_datagen  = ImageDataGenerator(rescale = 1./255)

# Training data
training_set = train_datagen.flow_from_directory('dataset/training_set',
                                                 target_size=(224,224),
                                                 batch_size=32,
                                                 class_mode='categorical')


# Test data
test_set = test_datagen.flow_from_directory('dataset/test_set',
                                                 target_size=(224,224),
                                                 batch_size=32,
                                                 class_mode='categorical')
```



```python
# Checkpoint...Save the best  model
from keras.callbacks import ModelCheckpoint
checkpoint = ModelCheckpoint('Checkpoint.hdf5',
                             save_weights_only=True,
                             monitor='val_loss',
                             save_best_only=True,
                             mode='min')

# For Prevent Show the Warning
import warnings
warnings.filterwarnings('ignore')

# fit the model

history = model.fit_generator(training_set,
                              validation_data=test_set,
                              epochs = 25,
                              steps_per_epoch=len(training_set),
                              validation_steps=len(test_set),
                              callbacks = [checkpoint])
```



```python
# Convert Model keras to Tensorflow Lite
import tensorflow as tf
converter = tf.compat.v1.lite.TFLiteConverter.from_keras_model_file('model.h5')
tflite_model = converter.convert()
open("model.tflite", "wb").write(tflite_model)
```

TESTING &Design

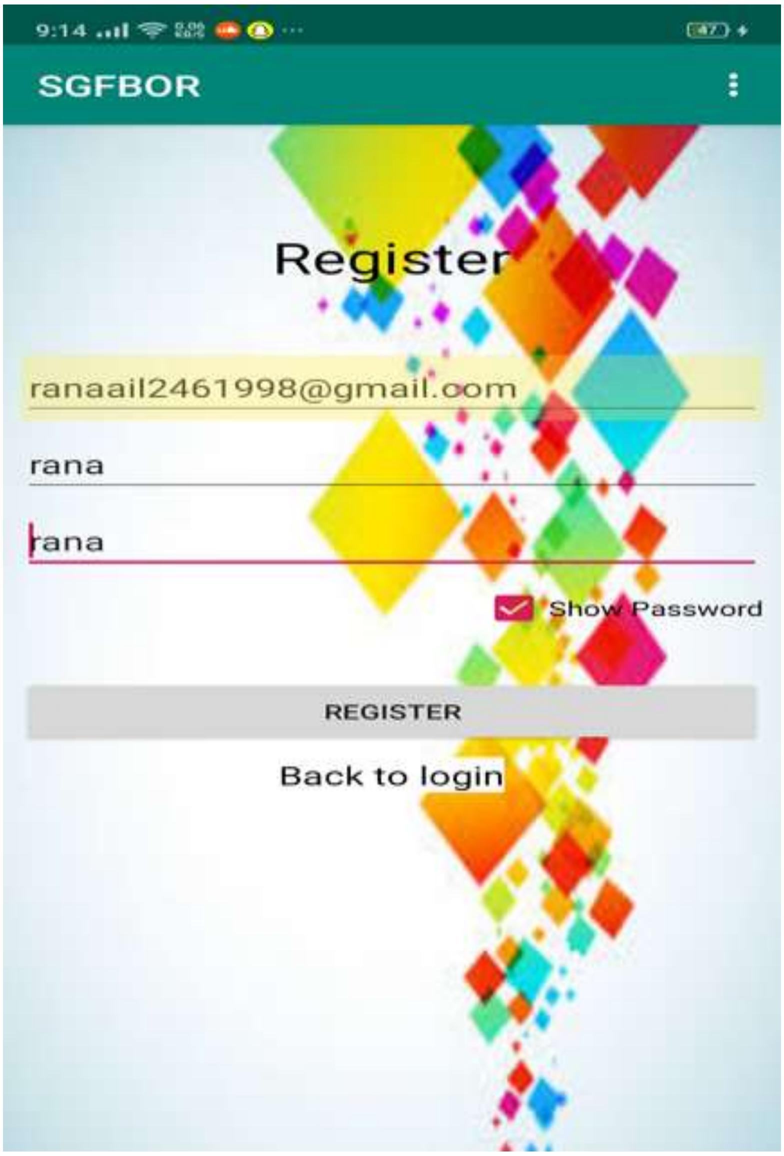
Chapter six

6 TESTING & DESIGN

This page is home page that the start of the application that appears 4s then Login page appears

Figure 6- 1 :Home page

This page is registration page that used to create an account to enter the Email and password and confirming password then press on button register to help the user to login to the application.



ranaail2461998@gmail.com

rana

rana

☑ Show Password

Figure 6- 2:Register page

This page is familiar page that the user must enter the name and the type of relation of that familiar person and he must choose the image of the familiar person from the gallery by pressing on the upload image button then press on save button to save data in database.

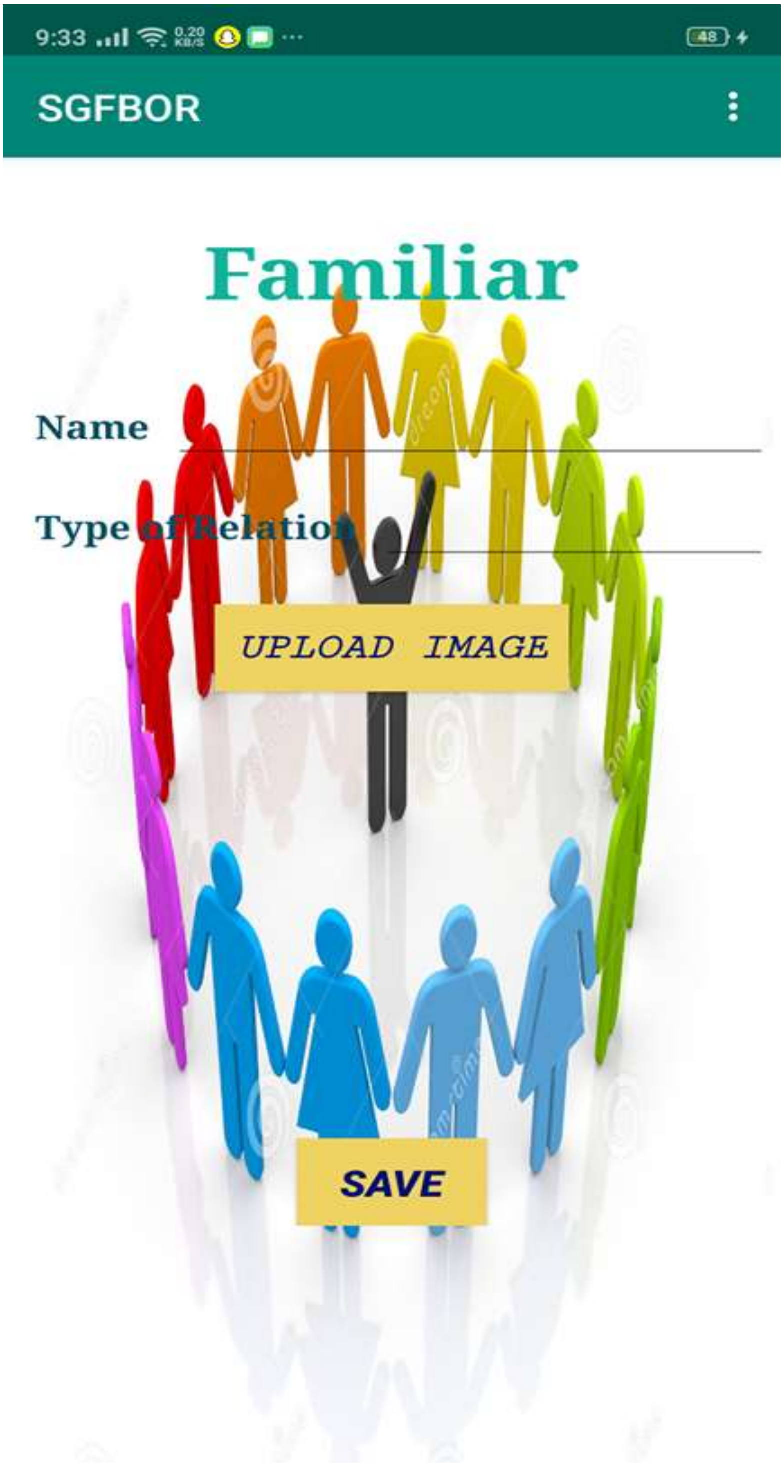


Figure 6- 3 :Familiar page

This page is Login page used that enter the E-mail and password and if the email or password is wrong it will appear that the email or password is wrong that the user must be sure of email and password to leads to WebView page.

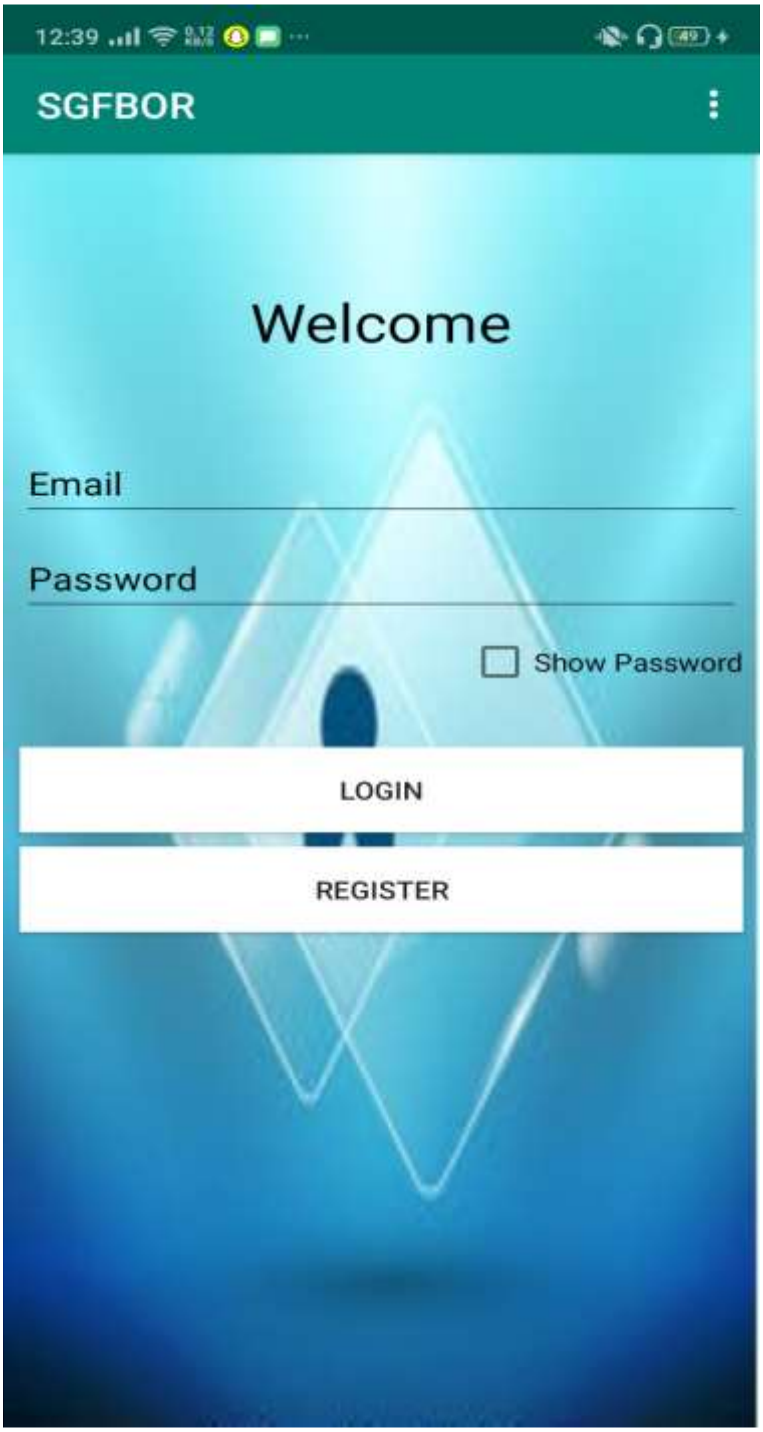


Figure 6- 4 :Login page

This page is WebView page that the camera captures the frames to analyses the content of the image then the result appears in toast that converts to sound that the blind person can hear.

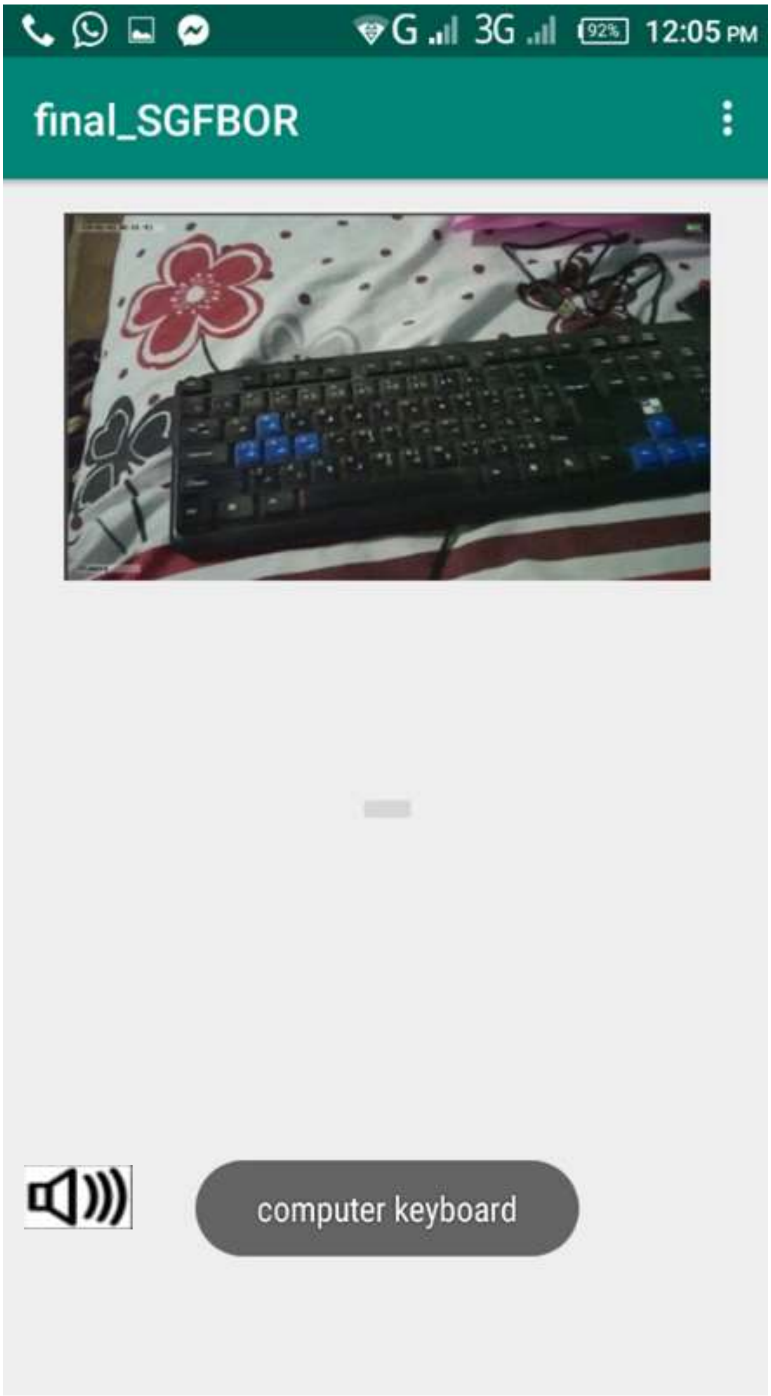


Figure 6- 5:result

This is the hardware used in the project (glasses) mini camera That the user must open the camera placed on the glasses to capture the live video.

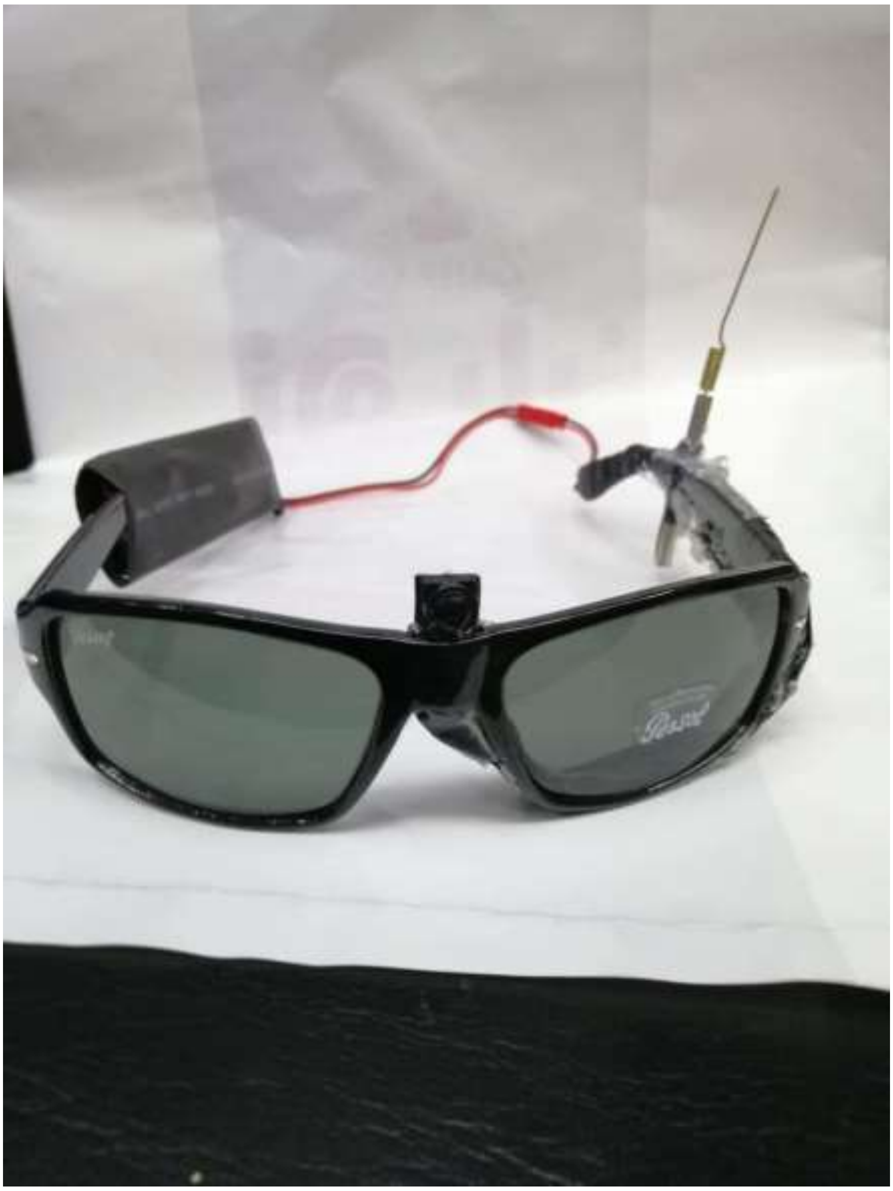


Figure 6- 6:hardware

CONCLUSION AND FUTURE WORK

Chapter SIVEN

7 CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

We use Android Studio and OpenCV in Android to make detections in a frame that taken by Wi-Fi in the application ,and we use deep learning to classify objects and determine how far of the object from the user and make the user know the relatives and familiar people he knows and converts the result of classification and the relatives into sound to the user can hear.

7.2 FUTURE WORK

- Increase accuracy of model deep learning.
- Increase objects in model.
- Increase favourite places in database.
- Increase Warning signs on the roads in the model.
- Increase Traffic signs in the model.

REFERENCE

8 REFERENCE

[1] https://www.pcbway.com/project/shareproject/Talking_Smart_Glass_For_Blind.html

[2] https://towardsdatascience.com/sift-scale-invariant-feature-transform-c7233dc60f37

[3] https://cs231n.github.io/convolutional-networks/

[4] https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123

[5] https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/

https://www.tensorflow.org/tutorials/keras/classification

[6] https://www.oreilly.com/library/view/machine-learning-for/9781786469878/252b7560-e262-49c4-9c8f-5b78d2eec420.xhtml

[7] https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-3-flattening

[8] https://android.developreference.com/article/15103443/Android+webview+shows+%E2%80%9Copen+this+website+in+chrome+or+mozilla%E2%80%9D

[9] https://www.programcreek.com/java-api-examples/?class=org.opencv.core.MatOfRect&method=toArray

[10] https://github.com/amitshekhariitbhu/Android-TensorFlow-Lite-Example/tree/master/app/src/main/java/com/amitshekhar/tflite

[11] https://stackoverflow.com/questions/48235495/how-to-draw-a-rectangle-containing-an-object-in-android-java-opencv

[12] https://www.codota.com/code/java/methods/android.speech.tts.TextToSpeech/setLanguage

[13] https://www.tensorflow.org/tutorials/keras/classification

[14] https://www.tensorflow.org/api_docs/python/tf/keras/Sequential