

Smart Home Automation using 2-Way Switching

Syeda Maheen Hashmi,
Bachelor of Computer science
Engineering, I SL Engineering
College ,Hyd, TS, India.,

Ayesha Saba,
Bachelor of Computer science
Engineering, I SL Engineering
College ,Hyd, TS, India.,

Husna Anjum
Bachelor of Computer science
Engineering, I SL Engineering
College ,Hyd, TS, India.,

Mohammed Liaqat Ali Khan,
Bachelor of Computer science
Engineering, I SL Engineering
College ,Hyd, TS, India.,

Abstract—Our Smart Home Automation provides a low cost effective and flexible home control and monitoring system using QR code. As Home Automation has become more and more commercial and marketable in recent years. The goal of our project is to provide comfort to old and disabled people as scanning of QR code is a one time process and can be accessed by anyone easily. Our proposed system required ESP32 and ESP8266 microcontroller to provide wireless connections to electrical appliances. We introduce smartphone application as smartphones are commonly used now-a-days and our application provides easy access to our home by using wifi module, bluetooth, scanning of QR code. The wastage of electricity can be known by notification on smartphone application. Application can be log in by using QR code or SSID.

Keywords— QR Code, SSID, 2-way switching, Smartphone application.

1. INTRODUCTION

1.1 OVERVIEW

Classic smart home, internet of things and using QR codes for security are the building blocks of our proposed advanced home automation. Each component contributes its core attributes and technologies to the proposed composition. IOT contributes the internet connection and remote management of mobile appliances, incorporated with a variety of wireless connections. Connections may be attached to home related

appliances, such as air conditioning, lights, fans, and other devices. And so, it embeds software intelligence into home devices to provide ways to measure home conditions and monitor home appliances functionality.

As QR stands for Quick Response (they can be read quickly by hand sets). It works in the same way as a barcode at the super market. QR code makes work easy and they will be no pressure of remembering IDs and passwords combining softwares, hardware, technologies, in order to generate the best product. In this project we explain the integration of IOT based Home Automation.

From the connectivity perspective, the added Electrical appliances are connected to internet in the basic components of our project are:- 1. We connect the electrical appliance to the hardware. 2. Then the mobile application is created for easy control. 3. We create the QR code for each and every room. 4. Then the appliances are connected to Internet. We describe the composition of an advanced smart home, incorporating these four components for building a practical advanced smart home automation. Finally, we identify open issues and future directions on the future of advanced smart home components and applications.

1.2 RELATED WORK

Smart devices remain the easiest and flexible way to a smart home. Smart things are now-a-days most commonly used by smartphones (or) smart handsets. They are many benefits of using smart technology for home automation as it insures

convenience, sustainability, security, efficiency, save time etc. A smart home is a home in which all the electrical appliances are connected to a control application. So they can either be switched on and switch off at any certain time (or) any certain place. As Home automation is more secure and saves a lot of time even though many people do not use it because they can't afford it the main objective of our project is to provide smart home automation which can be affordable by every individual and by introducing QR codes it became more flexible and can be managed by anyone. By using 2-way switching techniques even old people can use transitional switching method. Given below is 2-way switching technique.

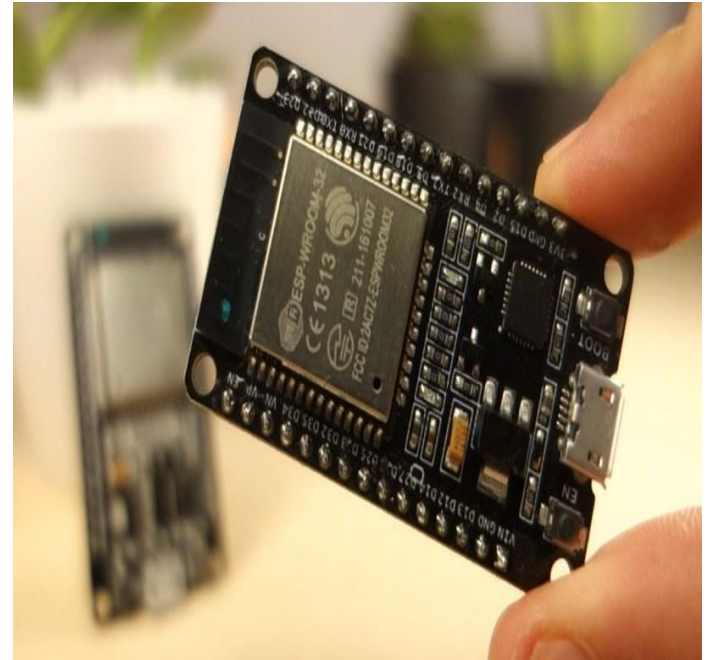
2.HOME AUTOMATION DEVICES

(REQUIREMENTS)

2.1 HARDWARE

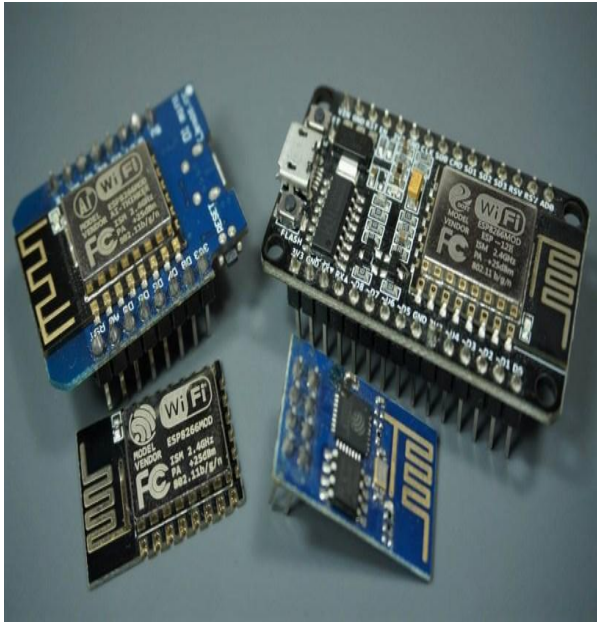
2.1.1 ESP32

- ESP32 is a low cost, low-power system on a chip microcontrollers with integrated wifi & dual-mode bluetooth.
- It is capable of functioning reliably in industrial environments, with an operating temperature ranging from -40°C to +125°C. It acts as an intriguing solution for IOT.
- The ESP32 peripherals include
 - *18 Analog-to-Digital Converter (ADC) channels*
 - *3 SPI interfaces*
 - *3 UART interfaces*
 - *2 I2C interfaces*
 - *16 PWM output channels*
 - *2 Digital-to-Analog Converters (DAC)*
 - *2 I2S interfaces*
 - *10 Capacitive sensing GPIOs*



2.1.2 ESP8266

- ESP8266 is a low-cost wifi microchip with a full TCP/IP stack & microcontroller capability.
- This chip first came to attention of western makers with ESP-01 module made by a third party manufacturer Ai-Thinker.
- It is capable of functioning consistently in industrial environments due to its wide operating temperature range.
- The ESP8266 peripherals include:
 - 17 GPIOs
 - SPI
 - I2C (implemented on software)
 - I2S interfaces with DMA
 - UART
 - 10-bit ADC



2.1.3 ELECTRICAL RELAY

- A relay is an electrically operated switch.
- It consists of a set of input terminals for a single or multiple control signals & a set of operating contact terminals.
- Switch may have any no of contacts in multiple contact forms.
- The electronic relay is a type of an electronic switch that opens or close the circuit contacts by using electronic component without any mechanical operation.
- In this relay, the current carrier pilot relaying scheme is used for the protection of the transmission line.
- The electronic relay has several advantages likes
 - They require low maintenance.
 - The relay has fast response time.
 - They decrease the burden on the instrument transformer.
- Disadvantages of Electronic Relay
 - The high tension supply is required for their operation.
 - The power consumption in the electronic relay is very high.
 - The relay has the short lifespan.
 - The electronic relay is practically not used in the power system

2.2 SOFTWARE

2.2.1 JAVA

- In our project, we are using JAVA as our backend process.
- Java is a general purpose programming language ie classbased, object-oriented & designed to have as few implementation dependencies as possible.
- This language produces software for multiple platforms.
- Types of Applications that Run on Java
 - Desktop GUI Applications
 - Mobile Applications
 - Embedded Systems
 - Web Applications
 - Web Servers and Application Servers
 - Enterprise Applications
 - Scientific Applications

2.2.2 XML

- XML is used as frontend process in our project.
- XML is Extensible Markup Language that defines a set of rules for encoding documents in a format ie both human-readable & machine-readable.
- It is designed to store & transport data.
- It is designed to be self-descriptive.
- By using **XML**, Web agents and robots (programs that automate Web searches or other tasks) are more efficient and produce more useful results.
- **XML** provides a standard method to access information, making it easier for **applications** and devices of all kinds to **use**, store, transmit, and display data.

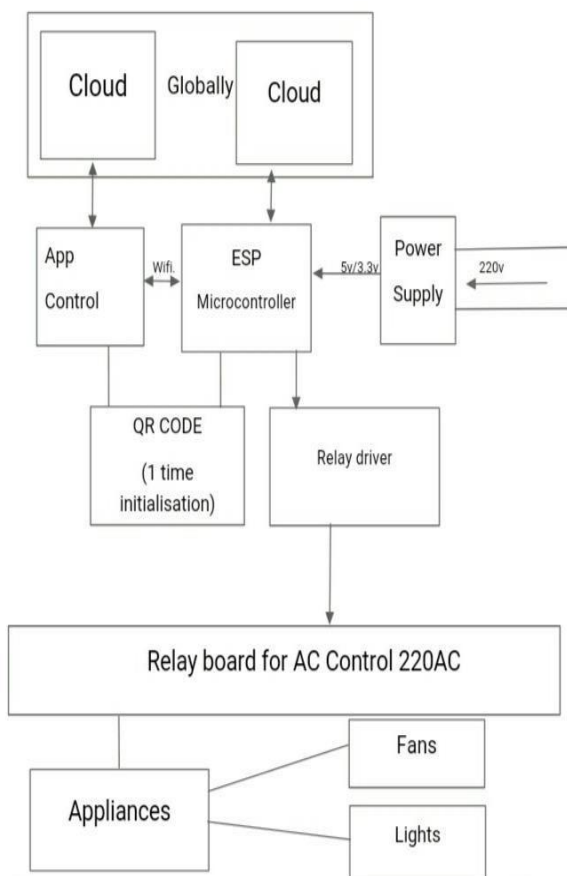
2.2.3 ANDROID STUDIO

- Android Studio is official integrated development environment for Google's Android operating system, built on JetBrainsIntelli J IDEA software & designed specifically for Android development.

- It offers tools custom-tailored for Android developers including rich code editing, debugging, testing, & profiling tools.
- It is purpose built for **Android** to accelerate your development and help you build the highest-quality apps for every **Android** device
- . It offer tools custom-tailored for **Android** developers, including rich code editing, debugging, testing, and profiling tools.

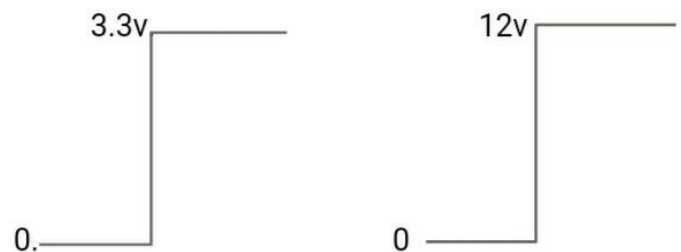
3. PROPOSED SYSTEM

App is a way to interface between user & devices. App scans QR code & save connection of setting into phone. App connects the device via local station. User is physically present near the devices, if user is not near the devices it moves to cloud to connect to devices & the devices will always be connected to cloud via local routes.

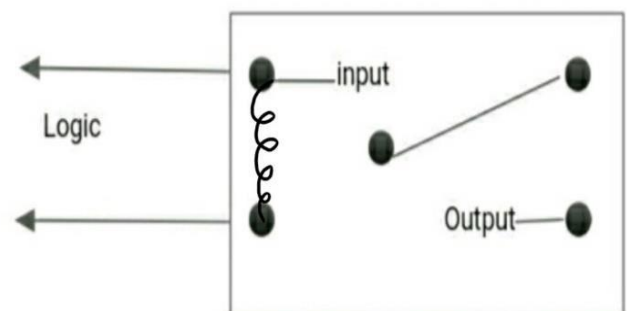


Block Diagram Of Smart Home Automation

Main board consist of Esp micro controller, relay drivers, 5v supply, and relay board's. Esp checks for any update being send via cloud (or) local station & act accordingly. If user sends a signal to switch any appliances, the Esp saves the data & set (or) reset the relay accordingly & then update this data to user & cloud. The app saves the acknowledged data sent from devices. Relay driver is logic shifter since Esp cannot drive relay directly & has a logic high of 3.3v & relay needs logic high of 12v.



We use a buffer logic shifter when ever Esp sets a relay, it creates a magnetic field in the coil inside it. The input string connect's to output terminal (similar to a switch we use whenever we apply physical press on a traditional switch it connect'stherwo terminals) & current flows through the input to output & appliances turn's on. Currently we are using a local host as our cloud.



4. WORKING PRINCIPLE

4.1 USE-CASE DIAGRAM

4.1.1 GENERAL DEFINATION

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

Use case diagrams consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system. Hence to model the entire system, a number of use case diagrams are used.

4.1.2 ADVANTAGES

- Use case help to capture the functional requirements of a system.
- Use cases are traceable.
- Use cases can serve as the basis for the estimating, scheduling, and validating effort.
- Use case can evolve at each iteration from a method of capturing requirements, to development guidelines to programmers, to a test case and finally into user documentation.
- Use case alternative paths capture additional behavior that can improve system robustness.
- Use cases have proved to be easily understandable by business users, and so have proven an excellent bridge between software developers and end users.

4.1.3 DESIGN

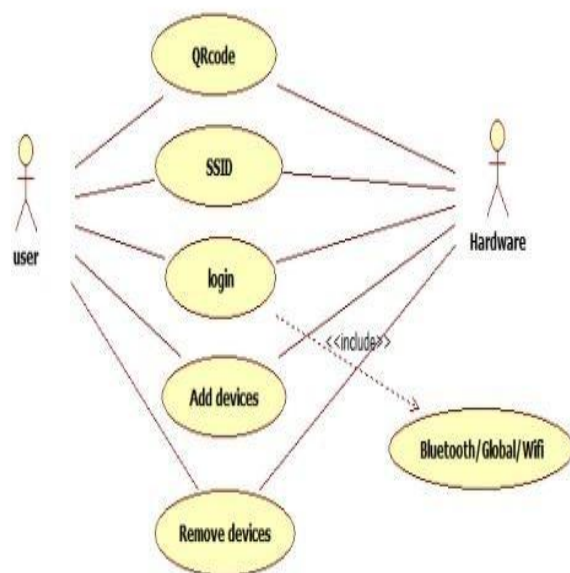
Our home automation system has the following use-cases

- QR code
- SSID
- Login
- Add devices
- Remove devices

The actors involved in the system are

- User
- Hardware

By using include relationship we can login the application and control it globally (or) by wifi (or) bluetooth.



4.2 CLASS DIAGRAM

4.2.1 GENERAL DEFINITION

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

An attribute is named property of a class which describes the object being modeled. In the class diagram, this component is placed just below the name-compartment. A derived attribute is computed from other attributes.

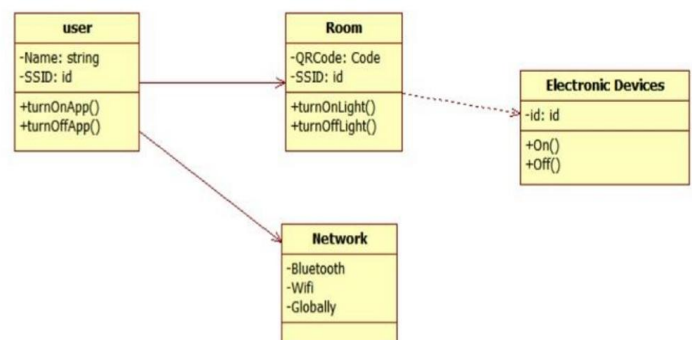
4.2.2 ADVANTAGES

- Class diagrams give you a sense of orientation.
- They provide detailed insight into the structure of your systems.
- At the same time they offer a quick overview of the synergy happening among the different system elements as well as their properties and relationships.

4.2.3 DESIGN

The home automation system makes use of following classes

- User :- User involves attributes such as Name, SSID and operations like turn on app and turn off app.
- Room:- Involves attributes such as QR code and SSID and operation turn on light and turn off light.
- Network :- Involves attributes like Bluetooth, Wifi, and Globally.
- Electronic devices :- Involves attributes I'd and operations on and off. Association relationship is given between users and Room and also Network user and dependency is given between Room and electronic devices.
-



4.3 SEQUENCE DIAGRAM

4.3.1 GENERAL DEFINITION

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function.

In a UML sequence diagram: lifeline, execution specification, message, combined fragment, interaction use, state invariant, continuation, destruction occurrence. Major elements of the sequence diagram are shown on the picture below.

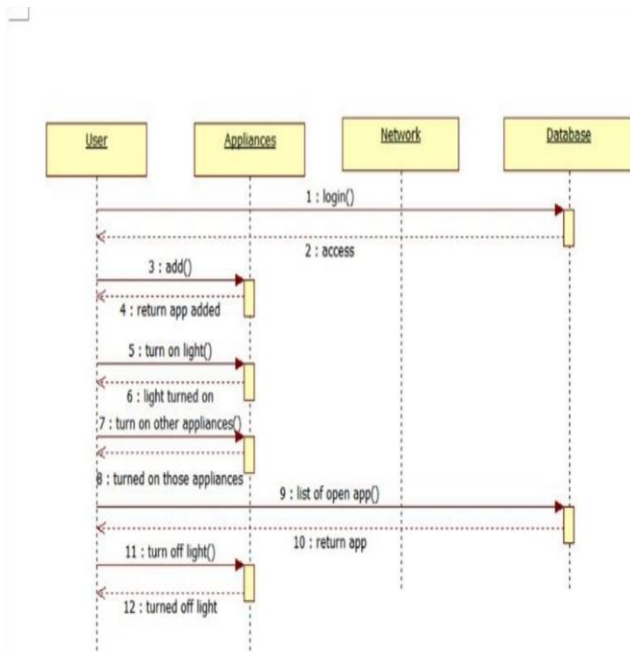
4.3.2 ADVANTAGES

- Sequence diagrams are used to represent message flow from one object to another object. Sequence diagrams are easier to maintain.
- Sequence diagrams are easier to generate.
- Sequence diagrams can be easily updated according to the changes within a system.

4.3.3 DESIGN

It represent sequence and interaction of a given use case. This can show the most of information about the system. It shows the events which includes signals inputs, decisions, interrupts and actions to (or) from users (or) external devices. A sequence diagram showing smart home automation system which involves the following objects.

- User
- Appliances
- Network
- Database



5. SOURCE CODE

#AddKey.java

```
package com.example.ibrahim.firebase;
import android.content.Context;
import android.net.Uri;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
public class AddKey extends Fragment
{
    Button button;
    public static AddKey newInstance() {
        return null;
    }
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_add_key, container, false);
    }
}
```

#AddInfo.java

```
package com.example.ibrahim.firebase;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import java.util.List;
public class AppInfo extends AppCompatActivity {
    EditText name, reg;
    Button pair;
    SharedPreferences sharedPreferences;
    private String vehicle_name, reg_no,
```

```
unique_no;
    RecyclerView adapter;
    List<CardViewData> list;
    // public CardViewData data;
    int FLAG=1;
    public AppInfo(){
    }
    public AppInfo (RecyclerView
adapter, List<CardViewData> list){
    this.adapter=adapter;
    this.list=list;
    }
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.info_layout);
        name = findViewById(R.id.info_name);
        reg = findViewById(R.id.info_reg);
        pair = findViewById(R.id.info_pair);
        //
        sharedPreferences=getSharedPreferences("DATA"
,Context.MODE_PRIVATE);
        pair.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                vehicle_name =
name.getText().toString().trim();
                reg_no =
reg.getText().toString().trim();
                //unique_no = "5000";
                // SharedPreferences.Editor
editor=sharedPreferences.edit();

                //editor.putString("Name",vehicle_name);
                Intent intent=new
Intent(getApplicationContext(),Home.class);
                intent.putExtra("name",vehicle_name);
                intent.putExtra("status","Paired");
                intent.putExtra("flag",FLAG);
                startActivity(intent);
                finish();
                // new
ConnectTask().execute(vehicle_name,reg_no,uni
que_no);
            }
        });
    }
    /* class ConnectTask extends
AsyncTask<String,String,String>{
        String vehicle_name, reg_no,
unique_no;
        @Override
        protected String
doInBackground(String... strings) {
            vehicle_name=strings[0];
            reg_no=strings[1];
            unique_no=strings[2];
```

```
        return null;
    }
    }*/
}
```

Bluetooth.java

```
package com.example.ibrahim.firebase;

import android.bluetooth.BluetoothA2dp;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import
android.bluetooth.BluetoothServerSocket;
import android.bluetooth.BluetoothSocket;
import android.content.Intent;
import android.os.Bundle;
import android.os.ParcelUuid;
import
android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.lang.invoke.MethodHandle;
import
java.lang.reflect.InvocationTargetException;
import java.lang.reflect.Method;
import java.util.ArrayList;
import java.util.Set;

import static android.os.SystemClock.sleep;
import static
com.example.ibrahim.firebase.R.layout.bluetooth;

public class Bluetooth extends
AppCompatActivity
{
    TextView b1,b3,b4;
    private BluetoothAdapter BA;
    private
Set<BluetoothDevice>pairedDevices;
    ListView lv;
    private OutputStream outputStream;
    private InputStream inputStream;
    BluetoothSocket socket = null;
    ParcelUuid[] uuids=null;
    BluetoothDevice result = null;
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
setContentView(bluefoot);

b1 = (Button)
findViewById(R.id.button);
b3=(Button)findViewById(R.id.send);
b4=(Button)findViewById(R.id.button4);

BA =
BluetoothAdapter.getDefaultAdapter();
lv =
(ListView)findViewById(R.id.listView);
if (!BA.isEnabled()) {
    Intent turnOn = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE
);
    startActivityForResult(turnOn,
0);

Toast.makeText(getApplicationContext(),
"Turned on",Toast.LENGTH_LONG).show();
    sleep(100);
} else {
Toast.makeText(getApplicationContext(),
"Already on", Toast.LENGTH_LONG).show();
}
try {
    Method getUuidsMethod =
BluetoothAdapter.class.getDeclaredMethod("get
Uuids", null);
    ParcelUuid[] uuids =
(ParcelUuid[]) getUuidsMethod.invoke(BA,
null);

    if(uuids != null) {
        for (ParcelUuid uuid : uuids)
        {
            Log.d("asad", "UUID: " +
uuid.getUuid().toString());
        }
    }else{
        Log.d("asad", "Uuids not
found, be sure to enable Bluetooth!");
    }

} catch (NoSuchMethodException e) {
    e.printStackTrace();
} catch (IllegalAccessException e) {
    e.printStackTrace();
} catch (InvocationTargetException e)
{
    e.printStackTrace();
}

}
```

```
public void off(View v){
    BA.disable();

    Toast.makeText(getApplicationContext(),
    "Turned off" ,Toast.LENGTH_LONG).show();
}

public void visible(View v){
    Intent getVisible = new
Intent(BluetoothAdapter.ACTION_REQUEST_DISCOV
ERABLE);
    startActivityForResult(getVisible,
0);
}

public void send(View v){
    String tosend="I am android app by
adil\n";
    try {
        socket =
result.createRfcommSocketToServiceRecord(uuid
s[0].getUuid());
    } catch (Exception e)
{Log.e("error:", "Error creating socket");}

    try {
        socket.connect();
        Log.e("1", "Connected");
        outputStream =
socket.getOutputStream();

        outputStream.write(tosend.getBytes());
        inputStream =
socket.getInputStream();
        //new Toast(context, "Already
Registered", Toast.LENGTH_SHORT).send();
        Log.w("try1:", "sent");
    } catch (IOException e) {
        Log.e("error:", e.getMessage());
        try {
            Log.e("2", "trying
fallback...");

            socket =(BluetoothSocket)
result.getClass().getMethod("createRfcommSock
et", new Class[]
{int.class}).invoke(result,1);
            socket.connect();
            Log.e("", "Connected");
            outputStream =
socket.getOutputStream();

            outputStream.write(tosend.getBytes());
            inputStream =
socket.getInputStream();
            //new Toast(context, "Already
Registered", Toast.LENGTH_SHORT).send();
            Log.w("try2:", "sent");
```

```
        }
        catch (Exception e2) {
            Log.e(":error:", "Couldn't
establish Bluetooth connection!");
        }
    }
}

public void list(View v){
    String deviceName = "ESP32test";
    pairedDevices =
BA.getBondedDevices();
    ArrayList list = new ArrayList();
    if (pairedDevices != null) {
        for (BluetoothDevice device :
pairedDevices) {
            if
(deviceName.equals(device.getName())) {
                uuids= device.getUuids();
                result = device;
                break;
            }
        }
    }
    //for(BluetoothDevice bt :
pairedDevices) list.add(bt.getName());

Toast.makeText(getApplicationContext(),
"connecting to"+deviceName+"
Devices",Toast.LENGTH_SHORT).show();
    //connect.invoke(proxy, result);
    //final ArrayAdapter adapter = new
ArrayAdapter(this,android.R.layout.simple_lis
t_item_1, list);
    //lv.setAdapter(adapter);
}
/*
private class ConnectThread extends
Thread {
    private final BluetoothSocket
mmSocket;
    private final BluetoothDevice
mmDevice;

    public ConnectThread(BluetoothDevice
device) {
        // Use a temporary object that is
later assigned to mmSocket
        // because mmSocket is final.
        BluetoothSocket tmp = null;
        mmDevice = device;

        try {
            // Get a BluetoothSocket to
connect with the given BluetoothDevice.
            // MY_UUID is the app's UUID
string, also used in the server code.
            tmp =
device.createRfcommSocketToServiceRecord(MY_U
UID);
```

```
        } catch (IOException e) {
            Log.e(TAG, "Socket's create()
method failed", e);
        }
        mmSocket = tmp;
    }

    public void run() {
        // Cancel discovery because it
otherwise slows down the connection.
        bluetoothAdapter.cancelDiscovery();

        try {
            // Connect to the remote
device through the socket. This call blocks
// until it succeeds or
throws an exception.
            mmSocket.connect();
        } catch (IOException
connectException) {
            // Unable to connect; close
the socket and return.
            try {
                mmSocket.close();
            } catch (IOException
closeException) {
                Log.e(TAG, "Could not
close the client socket", closeException);
            }
            return;
        }

        // The connection attempt
succeeded. Perform work associated with
// the connection in a separate
thread.
        manageMyConnectedSocket(mmSocket);
    }

    // Closes the client socket and
causes the thread to finish.
    public void cancel() {
        try {
            mmSocket.close();
        } catch (IOException e) {
            Log.e(TAG, "Could not close
the client socket", e);
        }
    }
}*/
}
```

CardViewData.java

```
package com.example.ibrahim.firebase;

import android.widget.ImageView;

public class CardViewData {
    String name, ssid, status, image_name_v;
    int id;

    public CardViewData(String ssid, String status,int id) {
        this.ssid = ssid;
        this.status = status;
        this.image_name_v = "@drawable/abc";
        this.id=id;
    }

    public CardViewData(String ssid, String name, String status) {
        this.ssid = ssid;
        this.name = name;
        this.status = status;
        this.image_name_v =
"@drawable/setting";
    }

    public void setSsid(String ssid) {
        this.ssid = ssid;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public String getSsid() {
        return ssid;
    }

    public String getStatus() {
        return status;
    }

    public int getId() {
        return id;
    }

    public String getimage_name() {
        return image_name_v;
    }

    public void setStatus(String status) {
        this.status = status;
    }
}
```

```
}
}
```

control_switch.java

```
package com.example.ibrahim.firebase;

public class control_switch {
    private String name;
    private int imageResource,color;
    private String serialnumber;
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getImageResource() {
        return imageResource;
    }

    public void setImageResource(int imageResource) {
        this.imageResource = imageResource;
    }

    public String getserialnumber() {
        return serialnumber;
    }

    public int getcolor(){
        return color;
    }

    public void setserialnumber(String serialnumber) {
        this.serialnumber = serialnumber;
    }

    public control_switch(String name, int imageResource, String serialnumber,int color) {
        this.name = name;
        this.imageResource = imageResource;
        this.serialnumber = serialnumber;
        this.color=color;
    }
}
```

control_switch_Adapter.java

```
package com.example.ibrahim.firebase;

import android.content.Context;
import android.graphics.Color;
import android.net.wifi.WifiManager;
import android.os.AsyncTask;
import android.os.Build;
import
```

```
android.support.annotation.RequiresApi;
import android.support.v7.widget.PopupMenu;
import
android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.List;
import java.util.stream.Stream;

import static android.os.SystemClock.sleep;

public class control_switch_Adapter extends
RecyclerView.Adapter<control_switch_Adapter.C
ustomViewHolder> {

    private Context mContext;
    private List<control_switch>
control_switch;
    int[] pinvalue;
    //final
control_switch_Adapter.CustomViewHolder
holder;
    public
control_switch_Adapter(List<control_switch>
control_switch, Context mContext) {

        this.mContext = mContext;
        this.control_switch = control_switch;
        //holder = null;
    }

    @Override
    public CustomViewHolder
onCreateViewHolder(ViewGroup parent, int
viewType) {
        View itemView =
LayoutInflater.from(parent.getContext())
.inflate(R.layout.scard,
parent, false);

        return new
CustomViewHolder(itemView);
```

```
    }

    @Override
    public void onBindViewHolder(final
control_switch_Adapter.CustomViewHolder
holder ,int position) {
        final control_switch
control_switch_var =
control_switch.get(position);

        holder.name.setText(control_switch_var.getNam
e());

        holder.serialnumber.setText(control_switch_va
r.getserialnumber());

        holder.image.setImageResource(control_switch_
var.getImageResource());

        holder.image.setBackgroundColor(control_switc
h_var.getcolor());
        holder.image.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast.makeText(mContext,
"clicked"+control_switch_var.getName(),
Toast.LENGTH_SHORT).show();

                funbacktask(holder,control_switch_var.getseri
alnumber());

                sleep(100);

                //holder.image.setBackgroundColor(Color.rgb(2
55,235,97));
            }
        });
    }

    private void funbacktask(final
control_switch_Adapter.CustomViewHolder
holder, final String pinnum) {
        class BackTask extends
AsyncTask<String, Void, String> {
            String msg, value, pin;
            int pin1 = 10, value1 = 0;
            StringBuilder response = new
StringBuilder();
            @Override
            protected void
onPostExecute(String s) {
                super.onPostExecute(s);
                if (pin1 ==
Integer.parseInt(pin)) {
                    control_switch
control_switch_var =
control_switch.get(pin1);

                    holder.image.setImageResource(control_switch_
var.getImageResource());
```

```
        if (value1 == 1) {
holder.image.setBackgroundColor(Color.rgb(76,
76, 76));
        Log.w("Signal
class:", "pin:is ON");
        } else {
holder.image.setBackgroundColor(Color.rgb(248
, 223, 95));
        Log.w("Signal
class:", "pin:is OFF");
        }
    }
}

@Override
protected String
doInBackground(String... params) {
    pin = pinnum;

//control_switch_Adapter.CustomViewHolder
holder = null;

    try {

        URL url = new
URL("http://192.168.4.1:1500/");
        HttpURLConnection
connection = (HttpURLConnection)
url.openConnection();

connection.setDoOutput(true);

connection.setInstanceFollowRedirects(false);

connection.setRequestMethod("POST");

connection.setRequestProperty("Content-Type",
"text/plain");

connection.setRequestProperty("raw", "utf-
8");

        connection.connect();

        BufferedWriter out = new
BufferedWriter(new
OutputStreamWriter(connection.getOutputStream
()));

        out.write("{ \"pin\": \"\" +
pin + \"\", \"id\": \"12356\" }");
        out.flush();
        BufferedReader in = new
BufferedReader(new
InputStreamReader(connection.getInputStream()
));

        String strLine = null;
        while ((strLine =
in.readLine()) != null) {
```

```
response.append(strLine);
        }
        in.close();
        msg =
connection.getResponseMessage();
        Log.w("control_switc",
"doInBackground: " + response.toString());
        String[] all_val =
response.toString().split(";");
        int i;
        for (i = 0; i <
all_val.length; i++) {
            String[] val =
all_val[i].split(":");
            if
(val[0].equals("pin")) {
                pin1 =
Integer.parseInt(val[1]);
            } else if
(val[0].equals("value")) {
                value1 =
Integer.parseInt(val[1]);
            }
        }
    } catch (Exception e) {
        Log.w("control_switc
error", "doInBackground: " + e.toString());
    }
    return response.toString();
}

new BackTask().execute();
}

@Override
/*public void csprocessFinish(String
output){
    Log.w("Signal class:", "processFinish
output :"+output);
    int pin1;
    String value1;
    String[] all_val=output.split(";");
    int i;
    for(i=0;i<all_val.length;i++){
        String[]
val=all_val[i].split(":");
        if(val[0].equals("pin")){
pin1=Integer.parseInt(val[1]);
        }
        else if(val[0].equals("value")){
            value1=val[1];
        }
    }
    if(value1.equals("1")){
        Log.w("Signal class:", "pin:is ON");
    }
    else{
        Log.w("Signal class:", "pin:is
OFF");
    }
}
```

```
    }*/
    /*private void funbacktask(final
CustomViewHolder holder, String
getserialnumber) {
        class BackTask extends
AsyncTask<String, void, String> {
            String msg, value, pin;

            StringBuilder response = new
StringBuilder();

            @Override
            protected void
onPostExecute(String s) {
                super.onPostExecute(s);

            }

            @Override
            protected String
doInBackground(String... strings) {
                value = strings[0];
                pin = strings[1];

//control_switch_Adapter.CustomViewHolder
holder = null;
                try {
                    int pin1 = 10, value1 =
0;
                    URL url = new
URL("http://192.168.4.1:1500/");
                    HttpURLConnection
connection = (HttpURLConnection)
url.openConnection();

                    connection.setDoOutput(true);
                    connection.setInstanceFollowRedirects(false);
                    connection.setRequestMethod("POST");
                    connection.setRequestProperty("Content-Type",
"text/plain");
                    connection.setRequestProperty("raw", "utf-
8");
                    connection.connect();

                    BufferedWriter out = new
BufferedWriter(new
OutputStreamWriter(connection.getOutputStream
()));
                    out.write("{\"pin\":\"" +
pin + "\",\"id\":\"12356\"}");
                    out.flush();
                    BufferedReader in = new
BufferedReader(new
InputStreamReader(connection.getInputStream()
));
```

```
                String strLine = null;
                while ((strLine =
in.readLine()) != null) {
                    response.append(strLine);
                }
                in.close();
                msg =
connection.getResponseMessage();
                Log.w("control_switc",
"doInBackground: " + msg);
                String[] all_val =
response.toString().split(";");
                int i;
                for (i = 0; i <
all_val.length; i++) {
                    String[] val =
all_val[i].split(":");
                    if
(val[0].equals("pin")) {
                        pin1 =
Integer.parseInt(val[1]);
                    } else if
(val[0].equals("value")) {
                        value1 =
Integer.parseInt(val[1]);
                    }
                    if (pin1 ==
Integer.parseInt(pin)) {
                        control_switch
control_switch_var =
control_switch.get(pin1);
                        holder.image.setImageResource(control_switch_
var.getImageResource());
                        if (value1 == 1) {
                            holder.image.setBackgroundColor(Color.rgb(255
, 235, 97));
                            Log.w("Signal
class:", "pin:is ON");
                        } else {
                            holder.image.setBackgroundColor(Color.rgb(0,
235, 97));
                            Log.w("Signal
class:", "pin:is OFF");
                        }
                    }
                } catch (Exception e) {
                    Log.w("control_switc
error", "doInBackground: " + e.toString());
                }
                return response.toString();
            }
        }
    }*/
```

```
@Override
public int getItemCount() {
    return control_switch.size();
}

public class CustomViewHolder extends
RecyclerView.ViewHolder {
    public TextView name, serialnumber;
    public ImageView image, menu;

    public CustomViewHolder(View
itemView) {
        super(itemView);
        name = (TextView)
itemView.findViewById(R.id.swName);
        serialnumber = (TextView)
itemView.findViewById(R.id.swNumber);
        image = (ImageView)
itemView.findViewById(R.id.swImage);
    }
}
```

CyclerAdapter.java

```
package com.example.ibrahim.firebase;
import android.content.Context;
import android.content.Intent;
import
android.support.annotation.NonNull;
import
android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import java.util.List;
public class CyclerAdapter extends
RecyclerView.Adapter<CyclerAdapter.ViewHolder
> {
    private static
CyclerAdapter.onItemClickListener
onItemClickListener;
    List<CardViewData> list;
    public Context context;
    //public onItemClickListener
onItemClickListener;
    public interface
onItemClickListener{
        void onItemClick(int position);
    }
    public void
setOnItemClickListener(RecyclerView.onItem
ClickListener onItemClickListener) {
        this.onItemClickListener =
(CyclerAdapter.onItemClickListener)
onItemClickListener;
    }
}
```

```
public CyclerAdapter(Context context,
List<CardViewData> list){
    this.list=list;
    this.context=context;
}
@Override
public int getItemCount() {
    return list.size();
}
@NonNull
@Override
public ViewHolder
onCreateViewHolder(@NonNull ViewGroup
viewGroup, int i) {
    LayoutInflater
layoutInflater=LayoutInflater.from(viewGroup.
getContext());
    View
view=layoutInflater.inflate(R.layout.card_lay
out,viewGroup,false);
    return new
ViewHolder(view,com.example.ibrahim.firebase.
CyclerAdapter.onItemClickListener);
}
@Override
public void onBindViewHolder(@NonNull
ViewHolder holderView, int i) {
    CardViewData
cardViewData=list.get(i);

    holderView.vehicle_name.setText(cardViewData.
getName());

    holderView.ssid.setText(cardViewData.getSsid(
));

    holderView.status.setText(cardViewData.getSta
tus());
}
public static class ViewHolder
extends RecyclerView.ViewHolder {
    private TextView
vehicle_name,ssid,status;
    public ViewHolder(@NonNull View
itemView,final onItemClickListener listener)
{
        super(itemView);

        vehicle_name=itemView.findViewById(R.id.vehic
le_name);

        ssid=itemView.findViewById(R.id.ssid_name);

        status=itemView.findViewById(R.id.status);
        itemView.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                int id=getAdapterPosition();
```

```
        listener.onClick(id);
    }
});
/* itemView.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent=new
Intent(v.getContext(),AppInfo.class);
    }
});*/
}
```

Dialog.java

```
package com.example.ibrahim.firebase;
import
android.content.DialogInterface;
import
android.support.v7.app.AlertDialog;
import
android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class Dialog extends
AppCompatActivity {
    AlertDialog.Builder builder;
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_dialog);
        builder=new
AlertDialog.Builder(getApplicationContext());
        builder.setPositiveButton("Agree",
new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface
dialog, int which) {
            }
        }).setNegativeButton("Decline", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface
dialog, int which) {
                finish();
            }
        });
    }
}
```

globally.java

```
package com.example.ibrahim.firebase;

import android.content.Intent;
import android.graphics.Typeface;
import android.net.Uri;
import
```

```
android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import
android.text.method.LinkMovementMethod;
import android.view.View;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

public class globelly extends YourActivity {

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.globelly);
        TextView txt2= (TextView)
findViewById(R.id.textView7);

        txt2.setMovementMethod(LinkMovementMethod.get
Instance());
        txt2.setOnClickListener(new
View.OnClickListener() {
            public void onClick(View v) {
                //Intent intent = new
Intent(Intent.ACTION_VIEW);
                //startActivity(intent);

                Toast.makeText(getBaseContext(), "Smart Home
is up to Date!",
                    Toast.LENGTH_SHORT).show();
            }
        });
        TextView txt1= (TextView)
findViewById(R.id.textView8); //txt is object
of TextView

        txt1.setMovementMethod(LinkMovementMethod.get
Instance());
        txt1.setOnClickListener(new
View.OnClickListener() {
            public void onClick(View v) {
                Intent browserIntent = new
Intent(Intent.ACTION_VIEW);

                browserIntent.setData(Uri.parse("https://app.
termly.io/document/terms-and-
conditions/62239a63-b665-4a7e-9c50-
bd030cbb4218"));
                startActivity(browserIntent);
            }
        });
    }
}
```

Home.java

```
package com.example.ibrahim.firebase;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.net.wifi.WifiConfiguration;

import android.os.AsyncTask;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.NavigationView;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentTransaction;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBar;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;
import android.util.Log;

import android.net.wifi.WifiManager;
import android.view.Menu;

import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.google.zxing.integration.android.IntentIntegrator;
import com.google.zxing.integration.android.IntentResult;

import android.support.design.widget.BottomNavigationView;
```

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.util.ArrayList;
import java.util.List;

public class Home extends AppCompatActivity
implements View.OnClickListener {
    public static int GETSTATUS = 1;
    EditText textmsg;

    static final int READ_BLOCK_SIZE = 100;
    private RecyclerView recyclerView;
    private WifiManager wifiManager;
    private WifiConfiguration
wifiConfiguration;
    private RecyclerView.Adapter adapter;
    List<CardViewData> list;
    String ssid, password;
    public int FLAG = 0;
    private Button scanBtn;
    private ActionBar toolbar;
    int nnumber;

    //public Home home;
    @Override

    protected void onActivityResult(int
requestCode, int resultCode, Intent data) {
        /*super.onActivityResult(requestCode,
resultCode, data);
        if (resultCode == RESULT_OK &&
requestCode == GETSTATUS && data != null) {
            if (data.hasExtra("SSID") &&
data.hasExtra("password")) {
                ssid =
data.getStringExtra("SSID");
                list.add(new
CardViewData(ssid, "Connected"));
                adapter.notifyDataSetChanged();
            }
        }*/
        IntentResult result =
IntentIntegrator.parseActivityResult(requestCode,
resultCode, data);
        if (result != null) {
            if (result.getContents() == null)
{
                Toast.makeText(this,
"Cancelled", Toast.LENGTH_LONG).show();
            } else {
                //Toast.makeText(this, "msg"
+ result, Toast.LENGTH_LONG).show();
                if
(!result.getContents().isEmpty()) {
                    String qr_data =
```

```
result.getContents());
        String[] qr_val =
qr_data.split("\n");
        int i;
        int count =
qr_val.length;
        if (count == 2) {
            for (i = 0; i <
qr_val.length; i++) {
                String[] qr_para
= qr_val[i].split(":");
                if
(qr_para[0].equals("ssid")) {
                    ssid =
qr_para[1];
                } else if
(qr_para[0].equals("password")) {
                    password =
qr_para[1];
                }
            }
            make_wifi_ready(1);
        } else {
            Toast.makeText(getBaseContext(), "Invalid QR
Code",
            Toast.LENGTH_SHORT).show();
        }
    } else {
        Toast.makeText(getBaseContext(), "Empty Qr
code",
        Toast.LENGTH_SHORT).show();
    }
}
} else {
    super.onActivityResult(requestCode,
resultCode, data);
    //Toast.makeText(this,
"msg"+resultCode, Toast.LENGTH_LONG).show();
}
}

    public void make_wifi_ready(int j) {
        if (!ssid.isEmpty()) {
            wifiManager = (WifiManager)
getApplicationContext().getSystemService(Cont
ext.WIFI_SERVICE);
            if ((wifiManager.isWifiEnabled())
== false) {
                wifiManager.setWifiEnabled(true);
            }
            wifiConfiguration = new
WifiConfiguration();
            wifiConfiguration.SSID = "\"" +
```

```
ssid + "\"";
            wifiConfiguration.preSharedKey =
 "\"" + password + "\"";
            wifiManager.addNetwork(wifiConfiguration);
            //BackgroundTask backgroundTask =
new BackgroundTask(wifiManager,
wifiConfiguration);
            //backgroundTask.execute(ssid,
password);
            List<WifiConfiguration> list =
wifiManager.getConfiguredNetworks();
            // WifiInfo
wifiInfo=wifiManager.getConnectionInfo();
            for (WifiConfiguration i : list)
            {
                if (i.SSID != null &&
i.SSID.equals("\"" + ssid + "\"")) {
                    wifiManager.disconnect();
                }
                wifiManager.enableNetwork(i.networkId, true);
                wifiManager.reconnect();
            }
            Writessid(1);
        } else {
            Toast.makeText(getBaseContext(),
"Empty Qr code",
            Toast.LENGTH_SHORT).show();
        }
    }

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_home);
        Toolbar toolbar = (Toolbar)
findViewById(R.id.toolbar);
        setListItem();
        String[][] all =
read_all_wifi_conn();
        if (all != null) {
            int i = 0;
            Log.w("hi", "value length " +
all.length);
            for (i = 0; i < all.length; i++)
            {
                if (all[i][0] == null) break;
                list.add(new
CardViewData(all[i][0], "is " + i, i));
            }
            nnumber = i;
        }
        adapter.notifyDataSetChanged();
        buildRecyclerView();
        //nav();
    }
}
```

```
        BottomNavigationView navigation =
        (BottomNavigationView)
        findViewById(R.id.navigationView);

        navigation.setOnNavigationItemSelectedListener(new
        BottomNavigationView.OnNavigationItemSelectedListener() {
            @Override
            public boolean
            onNavigationItemSelectedListener(@NonNull MenuItem
            item) {
                switch (item.getItemId()) {
                    case R.id.home:
                        return true;
                    case R.id.scan_button:
                        scan_qr_code(1);
                        return true;
                    case R.id.add:
                        Intent s1 = new
                        Intent(Home.this, WifiConnection.class);
                        startActivity(s1);
                        return true;
                    case R.id.settings:
                        Intent s = new
                        Intent(Home.this, YourActivity.class);
                        startActivity(s);
                        return true;
                }

                return false;
            }
        });

        //toolbar.setTitle("Home");

        /*FloatingActionButton fab =
        (FloatingActionButton)
        findViewById(R.id.fab);
        fab.setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new
                Intent(getApplicationContext(),
                WifiConnection.class);

                startActivityForResult(intent, GETSTATUS);
            }
        });

        fab.setImageResource(R.drawable.ic_add_black_
        24dp);*/

        /*DrawerLayout drawer =
        (DrawerLayout)
        findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new
```

```
ActionBarDrawerToggle(
        this, drawer, toolbar,
        R.string.navigation_drawer_open,
        R.string.navigation_drawer_close);
        drawer.addDrawerListener(toggle);
        toggle.syncState();*/
        //toolbar = supportActionBar!!

        //NavigationView navigationView =
        (NavigationView) findViewById(R.id.nav_view);
        //scanBtn = (Button)
        findViewById(R.id.scan_button);

        //scanBtn.setOnClickListener(this);

        //navigationView.setNavigationItemSelectedListener(this);
    }

    /*private void loadFragment(Fragment
    fragment) {
        // load fragment
        FragmentTransaction transaction =
        getSupportFragmentManager().beginTransaction();

        transaction.replace(R.id.frame_container,
        fragment);
        transaction.addToBackStack(null);
        transaction.commit();
    }*/
    /*@Override
    public boolean onCreateOptionsMenu(Menu
    menu) {
        getMenuInflater().inflate(R.menu.navigation,
        menu);

        return true;
    }

    public void nav() {
        BottomNavigationView bottomNavigation
        = findViewById(R.id.navigationView);

        bottomNavigation.setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View v) {
                scan_qr_code(1);
            }
        });
    }*/

    /*@Override
    public boolean
```

```
onOptionsItemSelected(MenuItem item) {
    // Handle item selection
    switch (item.getItemId()) {
        case R.id.home:
            //home();
            return true;
        case R.id.scan_button:
            scan_qr_code(1);
            return true;
        case R.id.add:
            //scan();
            return true;
        case R.id.settings:
            //scan();
            return true;
        default:
            return
    }
}
super.onOptionsItemSelected(item);
}*/
public void scan_qr_code(int i) {
    IntentIntegrator integrator = new
IntentIntegrator(this);
    integrator.setPrompt("Scan a
QRcode");

integrator.setOrientationLocked(false);
    integrator.initiateScan();
}

private String[][] read_all_wifi_conn() {
    String s = read_all_recent_wifi(1);
    if (s != null) {
        String all_wifi[] =
s.split("\n");
        //list.add(new
CardViewData("Syed", "ATIF"));
        int j = 0;
        String[][] val = new
String[all_wifi.length][2];
        for (int i = 0; i <
all_wifi.length; i++) {
            if (!all_wifi[i].isEmpty()) {
                val[j] =
all_wifi[i].split(":");
                j++;
            }
        }
        Log.w("read_all_wifi_conn",
"value length " + all_wifi.length);
        return val;
    }
    return null;
}

private void buildRecyclerView() {
    recyclerView =
findViewById(R.id.recyclerView);
    recyclerView.setHasFixedSize(true);
```

```
LinearLayoutManager layoutManager =
new
LinearLayoutManager(getApplicationContext())
);

recyclerView.setLayoutManager(layoutManager);
    recyclerView.setAdapter(adapter);
}

private void setListItem() {
    list = new ArrayList<>();
    adapter = new
RecyclerViewAdapter(getApplicationContext(),
list);
    Bundle bundle =
getIntent().getExtras();
    if (bundle != null) {
        String s =
bundle.getString("name", "name");
        String sta =
bundle.getString("status", "status");
        FLAG = bundle.getInt("flag");
        list.add(new CardViewData(ssid,
s, sta));
        adapter.notifyDataSetChanged();
    }
    ((RecyclerViewAdapter)
adapter).setOnItemClickListener(new
RecyclerViewAdapter.OnItemClickListener() {
        @Override
        public void onItemClick(int
position) {
            Intent intent = new
Intent(getApplicationContext(),
Signal.class);
            startActivity(intent);
            /*if (FLAG == 1) {
                Intent intent = new
Intent(getApplicationContext(),
AppInfo.class);
                startActivity(intent);
            }
            adapter.notifyDataSetChanged();
        }
    });
}

@Override
public void onBackPressed() {
    DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer_layout);
    if
(drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    } else {
```

```
        super.onBackPressed();
    }
}

/*@Override
public boolean onCreateOptionsMenu(Menu
menu) {
    // Inflate the menu; this adds items
to the action bar if it is present.

getMenuInflater().inflate(R.menu.home, menu);
    return true;
}*/

@Override
public void onClick(View view) {
    switch (view.getId()) {

    }
}

private void delete(Home F) {
    read_all_recent_wifi(1);
}

public void Writessid(int v) {
    String pre = read_all_recent_wifi(1);
    int sta = check_wifi_data(pre, ssid,
password);
    if (sta == 1) {
        try {
            if ((!ssid.isEmpty()) &
(!password.isEmpty())) {
                FileOutputStream fileout
= openFileOutput("allwificonnections.txt",
MODE_PRIVATE);
                OutputStreamWriter
outputWriter = new
OutputStreamWriter(fileout);
                outputWriter.write(pre +
"\n" + ssid + ":" + password);
                outputWriter.close();
                list.add(new
CardViewData(ssid, "NEW", nnumber));
            }
        }
    }
    adapter.notifyDataSetChanged();

    Toast.makeText(getBaseContext(), "File saved
successfully!",
    Toast.LENGTH_SHORT).show();
} else {

    Toast.makeText(getBaseContext(), "fill all
parameter",
    Toast.LENGTH_SHORT).show();
}
```

```
    } catch (Exception e) {
        e.printStackTrace();
    }
} else {
    Toast.makeText(getBaseContext(),
"Already Saved \nplz check in home page",
    Toast.LENGTH_SHORT).show();
}

public int check_wifi_data(String all,
String s, String p) {
    String[] wifi_list = all.split("\n");
    int i;
    for (i = 0; i < wifi_list.length;
i++) {
        String[] val =
wifi_list[i].split(":");
        if (val[0].equals(s)) {
            if (val[1].equals(p)) {
                return 0;
            }
        }
    }
    return 1;
}

public void WriteBtn(View v, String val)
{
    // add-write text into file
    String pre = read_all_recent_wifi(1);
    try {
        FileOutputStream fileout =
openFileOutput("allwificonnections.txt",
MODE_PRIVATE);
        OutputStreamWriter outputWriter =
new OutputStreamWriter(fileout);
        outputWriter.write(pre +
textmsg.getText().toString());
        outputWriter.close();
        //display file saved message

        //Toast.makeText(getBaseContext(), "File
saved successfully!",
        //Toast.LENGTH_SHORT).show();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

// Read text from file
public String read_all_recent_wifi(int i)
{
    try {
        FileInputStream fileIn =
openFileInput("allwificonnections.txt");
        InputStreamReader InputRead = new
InputStreamReader(fileIn);
    }
```

```
        char[] inputBuffer = new
char[READ_BLOCK_SIZE];
        String s = "";
        int charRead;
        while ((charRead =
InputRead.read(inputBuffer)) > 0) {
            String readstring =
String.valueOf(inputBuffer, 0, charRead);
            s += readstring;
        }
        InputRead.close();
        return s;
    } catch (Exception e) {
        e.printStackTrace();
        return "";
    }
}

/* public class BackgroundTask extends
AsyncTask<String, Integer, String> {
    String sid, pass;
    WifiConfiguration wifiConfiguration;
    WifiManager wifiManager;

    public BackgroundTask(WifiManager
wifiManager, WifiConfiguration
wifiConfiguration) {
        this.wifiManager = wifiManager;
        this.wifiConfiguration =
wifiConfiguration;
    }

    @Override
    protected void onPostExecute(String
s) {
        super.onPostExecute(s);
        Intent intent = new Intent();
        intent.putExtra("SSID", sid);
        intent.putExtra("password",
pass);
        setResult(Activity.RESULT_OK,
intent);
        finish();
    }

    @Override
    protected String
doInBackground(String... strings) {
        sid = strings[0];
        pass = strings[1];

        return "connected";
    }
}*/
}
```

MainActivity.java

```
package com.example.ibrahim.firebase;

/*import android.content.Intent;
import android.os.Bundle;
import
android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;
//import
android.support.test.InstrumentationRegistry;
import
com.google.zxing.integration.android.IntentIn
tegrator;
import
com.google.zxing.integration.android.IntentRe
sult;

//import
android.support.test.runner.AndroidJUnit4;
public class MainActivity extends
AppCompatActivity implements
View.OnClickListener {
    private Button scanBtn;

    //new
IntentIntegrator(this).initiateScan();
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);
        scanBtn = (Button)
findViewById(R.id.scan_button);

        //scanBtn.setOnClickListener(this);
        move_to_home(this);
    }

    private static void
move_to_home(MainActivity mainActivity) {
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        Intent intent = new
Intent(mainActivity, Home.class);

        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK
| Intent.FLAG_ACTIVITY_CLEAR_TASK);
        mainActivity.startActivity(intent);
    }
}
```

```
private static void
userLogin(MainActivity mainActivity) {
    // String email="";
    //String email =
    editTextEmail.getText().toString().trim();
    //String email=
    mainActivity.sharedPreferences.getString("ema
il","hello");
    //if(arrayDeque!=null)
    //email= (String) arrayDeque.peek();
    String password =
    mainActivity.editTextPassword.getText().toStr
ing().trim();
    if (password.length() < 4 ) {
mainActivity.editTextPassword.setError("Minim
um length of password should be 4");
mainActivity.editTextPassword.requestFocus();
return;
    }
    if (password.length() > 6) {
mainActivity.editTextPassword.setError("Maxim
um length of password should be 6");
mainActivity.editTextPassword.requestFocus();
return;
    }
    int result = Integer.parseInt(password);
    if(result==5498) {
        Intent intent = new Intent(mainActivity,
Home.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK
| Intent.FLAG_ACTIVITY_CLEAR_TASK);
        mainActivity.startActivity(intent);
    }
    else{
mainActivity.editTextPassword.setError("Incor
rect Password");
mainActivity.editTextPassword.requestFocus();
return;
    }

//progressBar.setVisibility(View.VISIBLE);
}
@Override
public void onClick(View view) {
    //llSearch.setVisibility(View.GONE);
    IntentIntegrator integrator = new
IntentIntegrator(this);
    integrator.setPrompt("Scan a
QRcode");
    integrator.setOrientationLocked(false);
    integrator.initiateScan();
}
```

```
/* switch(view.getId()) {
    case R.id.textViewSignup:
        Intent intent=new
Intent(getApplicationContext(),SignUpActivity
.class);
        startActivity(intent);
        break;
    case R.id.buttonLogin:
        userLogin(this);
        break;
    }
}

@Override
protected void onActivityResult(int
requestCode, int resultCode, Intent data) {
    IntentResult result =
IntentIntegrator.parseActivityResult(requestC
ode, resultCode, data);
    if (result != null) {
        if (result.getContents() == null)
{
            Toast.makeText(this,
"Cancelled", Toast.LENGTH_LONG).show();
        } else {
            move_to_home(this);
            //Toast.makeText(this,
"msg"+result, Toast.LENGTH_LONG).show();
        }
    } else {
super.onActivityResult(requestCode,
resultCode, data);
        //Toast.makeText(this,
"msg"+resultCode, Toast.LENGTH_LONG).show();
    }
}
}*/

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;

public class MainActivity extends Activity {

    // Splash screen timer
    private static int SPLASH_TIME_OUT =
2000;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);
        new Handler().postDelayed(new
Runnable() {
```

```
        @Override
        public void run() {
            Intent i = new
Intent(MainActivity.this, Home.class);
            startActivity(i);
            finish();
        }
    }, SPLASH_TIME_OUT);
}
```

policy.java

```
package com.example.ibrahim.firebase;

import android.os.Bundle;
import
android.support.v7.app.AppCompatActivity;
import android.text.Html;
import android.text.Spanned;
import android.widget.TextView;

public class policy extends AppCompatActivity
{
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.policy);
        String htmlAsString =
getString(R.string.policy);
        Spanned htmlAsSpanned =
Html.fromHtml(htmlAsString); // used by
TextView
        TextView textView = (TextView)
findViewById(R.id.textView6);
        textView.setText(htmlAsSpanned);
    }
}
```

RecyclerView.java

```
package com.example.ibrahim.firebase;

import android.content.Context;
import android.content.Intent;
import android.graphics.Color;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.net.wifi.WifiConfiguration;
import android.net.wifi.WifiInfo;
import android.net.wifi.WifiManager;
import android.support.annotation.NonNull;
import
android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
```

```
import java.io.FileInputStream;
import android.content.Context;
import java.io.FileOutputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import android.widget.Toast;
```

```
import java.io.FileInputStream;
import java.io.InputStreamReader;
import java.util.List;
```

```
public class RecyclerViewAdapter extends
RecyclerView.Adapter<RecyclerViewAdapter.ViewHolder> {
    private static
RecyclerViewAdapter.OnItemClickListener
onItemClickListener;
    List<CardViewData> list;
    public Context context;

    //public onItemClickListener
onItemClickListener;
    public interface onItemClickListener {
        void onItemClick(int position);
    }

    public void
setOnItemClickListener(RecyclerViewAdapter.OnItemClickListener
onItemClickListener) {
        this.onItemClickListener =
onItemClickListener;
    }

    public RecyclerViewAdapter(Context context,
List<CardViewData> list) {
        this.list = list;
        this.context = context;
    }

    @Override
    public int getItemCount() {
        return list.size();
    }

    @NonNull
    @Override
    public ViewHolder
onCreateViewHolder(@NonNull ViewGroup
viewGroup, int i) {
        LayoutInflater inflater =
LayoutInflater.from(viewGroup.getContext());
        View view =
inflater.inflate(R.layout.card_layout,
viewGroup, false);
        return new ViewHolder(view,
com.example.ibrahim.firebase.RecyclerViewAdapter.
```

```
onItemClickListener,this.context);
    }

    @Override
    public void onBindViewHolder(@NonNull
ViewHolder holderView, int i) {
        final CardViewData cardViewData =
list.get(i);

holderView.vehicle_name.setText(cardViewData.
getName());

holderView.ssid.setText(cardViewData.getSsid(
));

holderView.status.setText(cardViewData.getSta
tus());

//holderView.id.setText(cardViewData.getId())
;
        /*cardViewData.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast.makeText(context,
"clicked"+cardViewData.getId(),
Toast.LENGTH_SHORT).show();

//showOptionsMenu(holderView.menu);

//holder.image.setBackgroundColor(Color.rgb(2
55,235,97));
            }
        });*/

//holderView.image_name.setText(cardViewData.
getImage_name());
    }

    public static class ViewHolder extends
RecyclerView.ViewHolder {
        private TextView vehicle_name, ssid,
status, image_name;
        private WifiManager wifiManager;
        private WifiConfiguration
wifiConfiguration;
        static final int READ_BLOCK_SIZE =
100;
        public Context context;

        public ViewHolder(@NonNull View
itemView, final onItemClickListener listener,
final Context val) {
            super(itemView);
            vehicle_name =
itemView.findViewById(R.id.vehicle_name);
            ssid =
itemView.findViewById(R.id.ssid_name);
            status =
```

```
itemView.findViewById(R.id.status);
            itemView.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    int id =
getAdapterPosition();
                    listener.onItemClick(id);
                    Toast.makeText(val, "Room
Number : " + id, Toast.LENGTH_SHORT).show();

make_wifi_ready(ssid.getText().toString(),
val);

                    Toast.makeText(val,
"Connecting", Toast.LENGTH_SHORT).show();
                }
            }
        }
        /*public static boolean
isConnected(Context context) {
            ConnectivityManager
connectivityManager = (ConnectivityManager)

context.getSystemService(Context.CONNECTIVITY
_SERVICE);
            NetworkInfo networkInfo = null;
            if (connectivityManager != null)
            {
                networkInfo =
connectivityManager.getActiveNetworkInfo();
            }

            return networkInfo != null &&
networkInfo.getState() ==
NetworkInfo.State.CONNECTED;
        }*/
        public void make_wifi_ready(String
ssid, Context val) {
            String pre =
read_all_recent_wifi(1, val);
            String password =
check_wifi_data(pre, ssid);
            if (!password.equals("no")) {
                wifiManager = (WifiManager)
val.getSystemService(Context.WIFI_SERVICE);
                if
((wifiManager.isWifiEnabled()) == false) {
                    wifiManager.setWifiEnabled(true);
                }
                wifiConfiguration = new
WifiConfiguration();
                wifiConfiguration.SSID = "\""
+ ssid + "\"";

                wifiConfiguration.preSharedKey = "\""
+ password + "\"";

                wifiManager.addNetwork(wifiConfiguration);
```

```
        WifiInfo wifiInfo =
wifiManager.getConnectionInfo();
        String info_ssid =
wifiInfo.getSSID().replaceAll("^\"(.*)\"$",
"$1");;
        if(!info_ssid.equals(ssid)) {
            List<WifiConfiguration>
list = wifiManager.getConfiguredNetworks();
            // WifiInfo
wifiInfo=wifiManager.getConnectionInfo();
            for (WifiConfiguration i
: list) {
                if (i.SSID != null &&
i.SSID.equals("\"" + ssid + "\"")) {
                    wifiManager.disconnect();
                    wifiManager.enableNetwork(i.networkId, true);
                    wifiManager.reconnect();
                }
            }
        }
    }
}

    public String
read_all_recent_wifi(int i, Context val) {
        try {
            FileInputStream fileIn =
val.openFileInput("allwificonnections.txt");
            InputStreamReader InputRead =
new InputStreamReader(fileIn);
            char[] inputBuffer = new
char[READ_BLOCK_SIZE];
            String s = "";
            int charRead;
            while ((charRead =
InputRead.read(inputBuffer)) > 0) {
                String readstring =
String.valueOf(inputBuffer, 0, charRead);
                s += readstring;
            }
            InputRead.close();
            return s;
        } catch (Exception e) {
            e.printStackTrace();
            return "";
        }
    }

    public String check_wifi_data(String
all, String s) {
        String[] wifi_list =
all.split("\n");
        int i;
        for (i = 0; i < wifi_list.length;
i++) {
            String[] val =
```

```
wifi_list[i].split(":");
            if (val[0].equals(s)) {
                return val[1];
            }
        }
        return "no";
    }
}
```

SharedKeys.java

```
package com.example.ibrahim.firebase;
import android.os.Bundle;
import
android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
/**
 * A simple {@link Fragment} subclass.
 */
public class SharedKeys extends
Fragment {
    public SharedKeys() {
        // Required empty public constructor
    }
    @Override
    public View
onCreateView(LayoutInflater inflater,
ViewGroup container,
Bundle savedInstanceState) {
        // Inflate the layout for this
fragment
        return
inflater.inflate(R.layout.fragment_shared_key
s, container, false);
    }
}
```

ShareKey.java

```
package com.example.ibrahim.firebase;
import
android.support.design.widget.TabLayout;
import
android.support.design.widget.FloatingActionB
utton;
import
android.support.design.widget.Snackbar;
import
android.support.v7.app.AppCompatActivity;
import
android.support.v7.widget.Toolbar;
import
android.support.v4.app.Fragment;
import
android.support.v4.app.FragmentManager;
import
android.support.v4.app.FragmentPagerAdapter;
```

```
import
android.support.v4.view.ViewPager;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
public class ShareKey extends
AppCompatActivity {
    /**
     * The {@link
    android.support.v4.view.PagerAdapter} that
    will provide
     * fragments for each of the sections.
    We use a
     * {@link FragmentPagerAdapter}
    derivative, which will keep every
     * loaded fragment in memory. If this
    becomes too memory intensive, it
     * may be best to switch to a
     * {@link
    android.support.v4.app.FragmentStatePagerAdap
    ter}.
     */
    private SectionsPagerAdapter
    mSectionsPagerAdapter;
    /**
     * The {@link ViewPager} that will
    host the section contents.
     */
    private ViewPager mViewPager;
    @Override
    protected void onCreate(Bundle
    savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_share_key);
        Toolbar toolbar = (Toolbar)
        findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        // Create the adapter that will
        return a fragment for each of the three
        // primary sections of the activity.
        mSectionsPagerAdapter = new
        SectionsPagerAdapter(getSupportFragmentManager());
        // Set up the ViewPager with the
        sections adapter.
        mViewPager = (ViewPager)
        findViewById(R.id.container);
        mViewPager.setAdapter(mSectionsPagerAdapter);
        TabLayout tabLayout = (TabLayout)
        findViewById(R.id.tabs);
        mViewPager.addOnPageChangeListener(new
        TabLayout.TabLayoutOnPageChangeListener(tabLa
```

```
yout));
tabLayout.addTabSelectedListener(new
TabLayout.ViewPagerOnTabSelectedListener(mVie
wPager));
        FloatingActionButton fab =
        (FloatingActionButton)
        findViewById(R.id.fab);
        fab.setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with
                your own action", Snackbar.LENGTH_LONG)
                .setAction("Action", null).show();
            }
        });
        @Override
        public boolean
        onCreateOptionsMenu(Menu menu) {
            // Inflate the menu; this adds items
            to the action bar if it is present.
            getMenuInflater().inflate(R.menu.menu_share_k
            ey, menu);
            return true;
        }
        @Override
        public boolean
        onOptionsItemSelected(MenuItem item) {
            // Handle action bar item clicks
            here. The action bar will
            // automatically handle clicks on the
            Home/Up button, so long
            // as you specify a parent activity
            in AndroidManifest.xml.
            int id = item.getItemId();
            //noinspection
            SimplifiableIfStatement
            if (id == R.id.action_settings) {
                return true;
            }
            return
            super.onOptionsItemSelected(item);
        }
        /**
         * A placeholder fragment containing a
        simple view.
         */
        public static class
        PlaceholderFragment extends Fragment {
            /**
             * The fragment argument representing
            the section number for this
             * fragment.
             */
            private static final String
            ARG_SECTION_NUMBER = "section_number";
```

```
public PlaceholderFragment() {
}
/**
 * Returns a new instance of this
 * fragment for the given section
 * number.
 */
public static PlaceholderFragment
newInstance(int sectionNumber) {
    PlaceholderFragment fragment = new
    PlaceholderFragment();
    Bundle args = new Bundle();
    args.putInt(ARG_SECTION_NUMBER,
    sectionNumber);
    fragment.setArguments(args);
    return fragment;
}
@Override
public View
onCreateView(LayoutInflater inflater,
    ViewGroup container,
    Bundle savedInstanceState) {
    View rootView =
    inflater.inflate(R.layout.fragment_share_key,
    container, false);
    TextView textView = (TextView)
    rootView.findViewById(R.id.section_label);

    textView.setText(getString(R.string.section_f
    ormat,
    getArguments().getInt(ARG_SECTION_NUMBER)));
    return rootView;
}
/**
 * A {@link FragmentPagerAdapter} that
 * returns a fragment corresponding to
 * one of the sections/tabs/pages.
 */
public class SectionsPagerAdapter
extends FragmentPagerAdapter {
    public
    SectionsPagerAdapter(FragmentManager fm) {
        super(fm);
    }
    @Override
    public Fragment getItem(int position)
    {
        // getItem is called to instantiate
        the fragment for the given page.
        // Return a PlaceholderFragment
        (defined as a static inner class below).
        return
        PlaceholderFragment.newInstance(position +
        1);
    }
    @Override
    public int getCount() {
        // Show 3 total pages.

```

```
return 3;
}
}
}
```

Signal.java

```
package com.example.ibrahim.firebase;

import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.graphics.Color;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.net.wifi.WifiManager;
import android.os.AsyncTask;
import
    android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import
    android.support.v7.widget.DefaultItemAnimator
    ;
import
    android.support.v7.widget.GridLayoutManager;
import
    android.support.v7.widget.RecyclerView;

import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ProgressBar;
import android.widget.Toast;

import java.io.StringReader;
import java.util.ArrayList;
import java.util.List;

import java.io.BufferedReader;
import java.io.BufferedWriter;

import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;

interface BackTaskResponse {
    void processFinish(String output);
}

public class Signal extends AppCompatActivity
implements BackTaskResponse{
    List<control_switch> c_switch = new
    ArrayList<>();
    private RecyclerView recyclerView;
    private control_switch_Adapter

```

```
c_s_Adapter;  
    private ProgressBar spinner;  
    @Override  
    protected void onCreate(Bundle  
savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        setContentView(R.layout.activity_signal);  
  
        spinner=(ProgressBar)findViewById(R.id.progre  
ssBar);  
  
        spinner.setVisibility(View.VISIBLE);  
        BackTask backtask =new BackTask();  
        backtask.delegate = this;  
        backtask.execute("1","1");  
  
        recyclerView = (RecyclerView)  
findViewById(R.id.recycler_view);  
        c_s_Adapter = new  
control_switch_Adapter(c_switch, this);  
  
        // Create grids with 2 items in a row  
        RecyclerView.LayoutManager  
mLayoutManager = new GridLayoutManager(this,  
2);  
  
        recyclerView.setLayoutManager(mLayoutManager)  
;  
        recyclerView.setItemAnimator(new  
DefaultItemAnimator());  
        recyclerView.setAdapter(c_s_Adapter);  
  
    }  
    @Override  
    public void processFinish(String output){  
        if(!output.isEmpty()) {  
            Log.w("Signal class:",  
"processFinish output :"+ output);  
            String[] all_val =  
output.split(";");  
            int i;  
            control_switch c_switch_obj;  
            if (all_val.length > 1) {  
  
spinner.setVisibility(View.GONE);  
            }  
            for (i = 0; i < all_val.length;  
i++) {  
                Log.w("Signal class:", "line"  
+ all_val[i]);  
                String[] value =  
all_val[i].split(",");  
                if  
(value[1].equals("value:1")) {  
                    c_switch_obj = new  
control_switch("Switch", R.drawable.light,  
Integer.toString(i), Color.rgb(76, 76, 76));  
                } else {
```

```
                    c_switch_obj = new  
control_switch("Switch", R.drawable.light,  
Integer.toString(i), Color.rgb(248, 223,  
95));  
                }  
                c_switch.add(c_switch_obj);  
            }  
            c_s_Adapter.notifyDataSetChanged();  
        }  
    }  
  
    class BackTask extends AsyncTask<String,  
String, String>{  
        String msg;  
  
        public BackTaskResponse delegate =  
null;  
  
        StringBuilder response = new  
StringBuilder();  
        @Override  
        protected void onPostExecute(String  
s) {  
            super.onPostExecute(s);  
  
            delegate.processFinish(response.toString());  
        }  
  
        @Override  
        protected String  
doInBackground(String... strings) {  
            try {  
                URL url = new  
URL("http://192.168.4.1:1500/status");  
                HttpURLConnection connection  
= (HttpURLConnection) url.openConnection();  
                connection.setDoOutput(true);  
  
connection.setInstanceFollowRedirects(false);  
connection.setRequestMethod("POST");  
connection.setRequestProperty("Content-Type",  
"text/plain");  
connection.setRequestProperty("raw", "utf-  
8");  
  
//connection.getResponseCode();  
                connection.connect();  
                BufferedWriter out = new  
BufferedWriter(new  
OutputStreamWriter(connection.getOutputStream  
()));  
  
                out.write("{\"id\":\"12356\"}");  
            }  
        }  
    }  
}
```

```
        out.flush();
        BufferedReader in = new
BufferedReader(new
InputStreamReader(connection.getInputStream()
));
        String strLine = null;
        while ((strLine =
in.readLine()) != null)
        {
            response.append(strLine);
        }
        in.close();
        msg =
connection.getResponseMessage();
    } catch (Exception e) {

    }
    return response.toString();
}

}
```

SignUpActivity.java

```
/*package com.example.ibrahim.firebase;
import android.content.Context;
import android.content.Intent;
import
android.content.SharedPreferences;
import
android.support.annotation.NonNull;
import
android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Patterns;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;
import
com.google.android.gms.tasks.OnCompleteListener;
import
com.google.android.gms.tasks.Task;
//import
com.google.firebase.auth.AuthResult;
//import
com.google.firebase.auth.FirebaseAuth;
//import
com.google.firebase.auth.FirebaseAuthUserColl
isionException;
import java.util.ArrayDeque;
public class SignUpActivity extends
AppCompatActivity implements
View.OnClickListener {
```

```
    ProgressBar progressBar;
    EditText editTextEmail,
    editTextPassword;
    //private FirebaseAuth mAuth;
    SharedPreferences sharedPreferences;
    static public ArrayDeque<String>
arrayDeque;
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_sign_up);
        editTextEmail = (EditText)
findViewById(R.id.editTextEmail);
        editTextPassword = (EditText)
findViewById(R.id.editTextPassword);
        progressBar = (ProgressBar)
findViewById(R.id.progressBar);
        //mAuth = FirebaseAuth.getInstance();
        arrayDeque=new ArrayDeque<>();
        Button
button=findViewById(R.id.buttonSignUp);
        TextView
textView=findViewById(R.id.textViewLogin);
        button.setOnClickListener(this);
        textView.setOnClickListener(this);

        sharedPreferences=getSharedPreferences("Email
",Context.MODE_PRIVATE);
    }
    private void registerUser() {
        String email =
editTextEmail.getText().toString().trim();
        String password =
editTextPassword.getText().toString().trim();
        if (email.isEmpty()) {
            editTextEmail.setError("Email is
required");
            editTextEmail.requestFocus();
            return;
        }
        if
(!Patterns.EMAIL_ADDRESS.matcher(email).match
es()) {
            editTextEmail.setError("Please enter
a valid email");
            editTextEmail.requestFocus();
            //arrayDeque.push(email);
            return;
        }
        /*if (password.isEmpty()) {
            editTextPassword.setError("Password
is required");
            editTextPassword.requestFocus();
            return;
        }
        if (password.length() < 4 ) {
            editTextPassword.setError("Minimum
```

```
length of password should be 6");
    editTextPassword.requestFocus();
    return;
}
if (password.length() > 6) {
    editTextPassword.setError("Maximum
length of password should be 6");
    editTextPassword.requestFocus();
    return;
}*/
Intent intent = new
Intent(SignUpActivity.this, Control.class);

intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK
| Intent.FLAG_ACTIVITY_CLEAR_TASK);
startActivity(intent);

//progressBar.setVisibility(View.VISIBLE);
//arrayDeque.push(email);
/* *SharedPreferences.Editor
editor=sharedPreferences.edit();
editor.putString("email",email);
editor.commit();/
/*

mAuth.createUserWithEmailAndPassword(email,
password).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull
Task<AuthResult> task) {
        progressBar.setVisibility(View.GONE);
        if (task.isSuccessful()) {
            Intent intent = new
Intent(SignUpActivity.this, Control.class);

            intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK
| Intent.FLAG_ACTIVITY_CLEAR_TASK);
            startActivity(intent);
        } else {
            if (task.getException() instanceof
FirebaseAuthUserCollisionException) {

                Toast.makeText(getApplicationContext(),
                "Already Registered",
                Toast.LENGTH_SHORT).show();
            } else {

                Toast.makeText(getApplicationContext(),task.g
etException().getMessage(),Toast.LENGTH_SHORT
).show();
            }
        }
    }
});
/* /
}
@Override
public void onClick(View view) {/*
```

```
switch (view.getId()) {
    case R.id.buttonSignUp:
        registerUser();
        break;
    case R.id.textViewLogin:
        Intent intent=new
Intent(getApplicationContext(),MainActivity.c
lass);

        startActivity(intent);
        break;
    }
    /* /
    Intent intent=new
Intent(getApplicationContext(),MainActivity.c
lass);

    startActivity(intent);
}
}*/
```

WifiConnection.java

```
package com.example.ibrahim.firebase;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.net.wifi.WifiConfiguration;
import android.net.wifi.WifiInfo;
import android.net.wifi.WifiManager;
import android.os.AsyncTask;
import
android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.util.List;

public class WifiConnection extends
AppCompatActivity {
    private WifiManager wifiManager;
    private String SSID, password;
    private TextView ssid, pass;
    private Button connect;
    private WifiConfiguration
wifiConfiguration;
    static final int READ_BLOCK_SIZE = 100;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_wifi_connect
```

```
ion);
    ssid = findViewById(R.id.c_ssid);
    pass = findViewById(R.id.c_password);
    connect = findViewById(R.id.add);
    wifiManager = (WifiManager)
getApplicationContext().getSystemService(Context.WIFI_SERVICE);
    if ((wifiManager.isWifiEnabled()) == false) {
        wifiManager.setWifiEnabled(true);
    }
    /*connect.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (ssid.getText().equals(""))
|| pass.getText().equals("")) {

Toast.makeText(getApplicationContext(),
"Enter SSID and Password",
Toast.LENGTH_LONG).show();

        } else {
            SSID =
ssid.getText().toString().trim();
            password =
pass.getText().toString().trim();
            wifiConfiguration = new
WifiConfiguration();
            wifiConfiguration.SSID =
"\\" + SSID + "\\";

wifiConfiguration.preSharedKey = "\\" +
password + "\\";

wifiManager.addNetwork(wifiConfiguration);
            BackgroundTask
backgroundTask = new
BackgroundTask(wifiManager,
wifiConfiguration);

backgroundTask.execute(SSID, password);
            Writessid(1);

        }
    }
});*/
}

    public String read_all_recent_wifi(int i)
{
    try {

        FileInputStream fileIn =
openFileInput("allwificonnections.txt");
        InputStreamReader InputRead = new
InputStreamReader(fileIn);
        char[] inputBuffer = new
char[READ_BLOCK_SIZE];
```

```
String s = "";
        int charRead;
        while ((charRead =
InputRead.read(inputBuffer)) > 0) {
            // char to string conversion
            String readstring =
String.valueOf(inputBuffer, 0, charRead);
            s += readstring;
        }
        InputRead.close();
        //textmsg.setText(s);
        return s;
    } catch (Exception e) {
        e.printStackTrace();
        return "";
    }
}

    public void Writessid(int v) {
        String pre = read_all_recent_wifi(1);
        try {
            if ((!SSID.isEmpty()) &
(!password.isEmpty())) {
                FileOutputStream fileout =
openFileOutput("allwificonnections.txt",
MODE_PRIVATE);
                OutputStreamWriter
outputWriter = new
OutputStreamWriter(fileout);
                outputWriter.write(pre + "\n"
+ SSID + ":" + password);
                outputWriter.close();
                //display file saved message

Toast.makeText(getBaseContext(), "File saved
successfully!",
Toast.LENGTH_SHORT).show();
            } else {

Toast.makeText(getBaseContext(), "fill all
parameter",
Toast.LENGTH_SHORT).show();
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public class BackgroundTask extends
AsyncTask<String, Integer, String> {
        String sid, pass;
        WifiConfiguration wifiConfiguration;
        WifiManager wifiManager;

        public BackgroundTask(WifiManager
wifiManager, WifiConfiguration
wifiConfiguration) {
```

```
        this.wifiManager = wifiManager;
        this.wifiConfiguration =
wifiConfiguration;
    }

    @Override
    protected void onPostExecute(String
s) {

        super.onPostExecute(s);
        Intent intent = new Intent();
        intent.putExtra("SSID", sid);
        intent.putExtra("password",
pass);
        setResult(Activity.RESULT_OK,
intent);
        finish();
    }

    @Override
    protected String
doInBackground(String... strings) {
        sid = strings[0];
        pass = strings[1];
        List<WifiConfiguration> list =
wifiManager.getConfiguredNetworks();
        // WifiInfo
        wifiInfo=wifiManager.getConnectionInfo();
        for (WifiConfiguration i : list)
        {
            if (i.SSID != null &&
i.SSID.equals("\"" + sid + "\"")) {
                wifiManager.disconnect();

                wifiManager.enableNetwork(i.networkId, true);
                wifiManager.reconnect();
            }
        }
        return "connected";
    }
}
```

YourActivity.java

```
package com.example.ibrahim.firebase;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.support.annotation.NonNull;
import
android.support.design.widget.BottomNavigatio
nView;
import
android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.CardView;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import
```

```
android.support.v7.widget.RecyclerView;
import android.widget.GridLayout;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.Toast;

public class YourActivity extends
AppCompatActivity {

    GridLayout gridLayout;
    ImageButton myImageButton;
    // @SuppressWarnings("WrongViewCast")
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.youractivity);
        BottomNavigationView navigation =
(BottomNavigationView)
findViewById(R.id.navigationView);

        gridLayout=(GridLayout)findViewById(R.id.main
Grid);

        setSingleEvent(gridLayout);

    }

    // we are setting onClickListener for
each element
    private void setSingleEvent(GridLayout
gridLayout) {
        for(int i = 0;
i<gridLayout.getChildCount();i++){
            CardView
cardView=(CardView)gridLayout.getChildAt(i);
            final int finalI= i;
            cardView.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View
view) {

                    if(finalI==3){
                        Intent intent=new
Intent(getApplicationContext(),
globelly.class);

                        startActivity(intent);
                    }
                    if(finalI==4){
                        Intent intent=new
Intent(getApplicationContext(),
policy.class);

                        startActivity(intent);
                    }
                }
            });
        }
    }
}
```

```
        }
        if(finalI==2){
            Intent intent=new
Intent(getApplicationContext(),
Bluetooth.class);

startActivity(intent);
        }
    }
});
    }
}

/*
navigation.setOnNavigationItemSelectedListener(new
BottomNavigationView.OnNavigationItemSelectedListener() {
    @Override
    public boolean
onNavigationItemSelectedListener(@NonNull MenuItem
item) {
        //Fragment fragment;
        switch (item.getItemId()) {
            case R.id.home:
                Intent s2 = new
Intent(YourActivity.this, Home.class);
                startActivity(s2);
                return true;
            case R.id.scan_button:

Toast.makeText(getApplicationContext(), "To Scan
make Sure You are in Home Page!",

Toast.LENGTH_SHORT).show();
                return true;
            case R.id.add:

Toast.makeText(getApplicationContext(), "To Add make
Sure You are in Home Page!",

Toast.LENGTH_SHORT).show();
                return true;
            case R.id.settings:
                Intent s = new
Intent(YourActivity.this,
YourActivity.class);
                startActivity(s);
                return true;
        }

        return false;
    }
});*/
}
```

6. ADVANTAGES

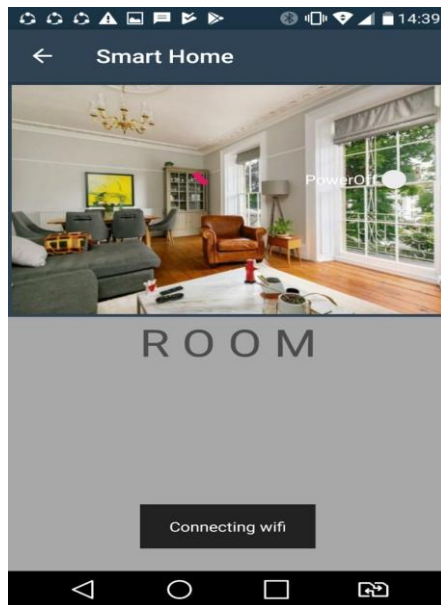
- All electronic devices can be controlled and monitored from one place.
- As QR code is a one time process and can be accessed by anyone easily.
- Home security can be improved using Smart Home automation.
- QR code is more flexible for scanning than compared to other technologies.
- Encoding of all types of data like numeric, alphabetic, special and binary is not required as it can scan directly.
- Maintenance of this software is easier than compared to other softwares.
- As smart home automation is for new generation & old generation people would find it difficult to connect the app's/QR codes/phones, we have implemented 2-way switching technique which helps the old generation to use the traditional method of operation and by time they will also start using the app method for connecting.

7. RESULT

- Following are the screenshots of our project.







8.CONCLUSION

This paper presents the new circuit topology for monitoring and controlling home electrical devices by using the flexible home-based Android smart phone at a reasonable price and implemented by using ESP as well as using android app for system control configuration. The proposed new circuit topology is used in a quiet based web services in an interoperable application layer for communication between the remote user and the home device. All Android-based smart phone, the Wi-Fi connection is the support built; the home access device to control can use the phone, Wi-Fi, 3G or 4G to access the Web page on hosting server using Android App.

REFERENCES

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6]

- [1] Ravi Kishore Kodali, Vishal Jain, Swadeep Bose & Lakshmi Boppana, "IOT Based smart security & home automation system". International Conference on Computing Communication & Automation (ICCCA2016)
- [2] K.Satteesh Kumar, N.Ajith Kumar, P.A Gopinath,S.Ranjith Kumar, J.Chndramohan,"Implementation of smart home automation & security system using Arduino &wi-fi International journal of engineering research & technology. Vol-5, Issue 13
- [3] Mohammed S.Soliman, MajedO.Dwairi, ImanI.M.AbuSulayman, Sami H.A.Almalki "Towards the design & implementation a smart home automation system based on Internet of things approach" International journal of applied engineering research ISSN, Volume 12, Number 11(2017)

-
- [4] Ma Naing, Ni Ni San Hlaing "Arduino based smart home automation system" International journal of Trend in scientific research & Development (IJTSRD) Volume 3 Issue 4
- [5] A.R.ALAli, Imran A.Zualkeman, Mohammed Rashid, Ragini Gupta, Maxim Ali Karar "A smart home energy management system using IOT & Big Data Analytics Approach" IEEE Transaction on consumer electronics, Vol 63, No 4, November 2017
- [6] Amit Kumar Dwivedi, Mrs.Shimi S.L "Home Automation & energy management using Android App" International Journal of Engineering Research & Technology (IJERT) Vol 4, Issue 12 Dec 2015.
- [7] Varun Kaushik, Shahbaaz Ahmed, Sufia Begum Shaikh, "Smart Home Automation and security system using GSM and ARM7", International Journal of Engineering Research & Technology (IJERT) Vol.5, Issue 18.
- [8] Md.Rakib Ahsan, Sheikh Zarif Ahmed, Mohammed ShamsulArefin, Md.Aminul Bari, "Implementation of IOT based smart security & home automation system" IJERT Vol 8 Issue 06.
- [9] K.S.Panchpor, Dr.C.N.Deshmukh "Home automation & surveillance : A review" International Journal of science & engineering application Volume 8, Issue 4.
- [10] VaishnaviS.Gunge, PratibhaS.Yalagi "Smart Home Automation : A Literature Review" International Journal of Computer Application. 11. Basil Hamed "Design & Implementation of Smart House control using LABVIEW" International Journal of Soft Computing & Engineering (IJSCE) Volume-1, Issue-6, January 2012.