**CONTROLLER INTERFACE**

package com.eca;

importjava.io.IOException;

importjava.io.InputStream;

importjava.io.OutputStream;

importjava.sql.ResultSet;

importjava.util.Enumeration;

importjavax.swing.\*;

importjavax.comm.\*;

importcom.commondb.Common\_DB;

public class Main implements Runnable {

CommDriver driver;

CommPortIdentifierpid;

SerialPort port;

Enumeration ports = null;

InputStreamips = null;

OutputStream ops = null;

String commid="";

public Main() {

try {

commid = JOptionPane.showInputDialog("Enter your Comm Port ID","COM");

Common\_DB cd = new Common\_DB();

driver = (CommDriver)Class.forName("com.sun.comm.Win32Driver").newInstance();

} catch (InstantiationException | IllegalAccessException

| ClassNotFoundException e) {

e.printStackTrace();

}

driver.initialize();

}

public void run() {

try {

ports = CommPortIdentifier.getPortIdentifiers();

while(ports.hasMoreElements()) {

pid = (CommPortIdentifier)ports.nextElement();

if(pid.getPortType()==CommPortIdentifier.PORT\_SERIAL) {

if(pid.getName().equalsIgnoreCase(commid)) {

System.out.println("\nRunning..\n");

port = (SerialPort)pid.open("ECA Thread", 2000); ips = port.getInputStream();

ops = port.getOutputStream();

while(true) {

ResultSetrs = Common\_DB.ViewParticularData("GPRSECA", "instructions", “passkey", "MySecret");

if(rs.next()) {

String instrction = rs.getString("ins"); ops.write(instrction.getBytes());

}

Thread.sleep(1000);

}

}

}

}

} catch (InstantiationException | IllegalAccessException

| ClassNotFoundException | PortInUseException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

} catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\*

\* main program called at the beginning of code execution

\*

\*/

public static void main(String[] args) {

System.out.println("Initializing Home Automation System\n");

System.out.println("Scanning for COM ports... Please Wait..\n\n");

new Thread(new Main()).start();

}

}

**ANDROIDAPPLICATION**

packagecom.voicebasedeca;

importjava.sql.Connection;

importjava.sql.DriverManager;

importjava.sql.Statement;

importjava.util.ArrayList;

importjava.util.Locale;

importandroid.app.Activity;

importandroid.content.ActivityNotFoundException;

importandroid.content.Intent;

importandroid.os.AsyncTask;

importandroid.os.Bundle;

importandroid.speech.RecognizerIntent;

importandroid.speech.tts.TextToSpeech;

importandroid.util.Log;

importandroid.view.Menu;

importandroid.view.View;

importandroid.widget.EditText;

importandroid.widget.ImageButton;

importandroid.widget.TextView;

importandroid.widget.Toast;

importandroid.widget.ToggleButton;

public class MainActivity extends Activity implements

TextToSpeech.OnInitListener {

protected static final int RESULT\_SPEECH = 1;

privateImageButtonbtnSpeak;

ToggleButtonlighttoggle, fantoggle;

privateTextViewtxtText;

EditTextipaddress;

public String voicedata = "", dispdata = "";

privateTextToSpeechtts;

Connection con = null;

Statement ps = null;

privateintdev = 0;

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

tts = new TextToSpeech(MainActivity.this, MainActivity.this);

txtText = (TextView) findViewById(R.id.txtText);

ipaddress = (EditText) findViewById(R.id.ipaddr123);

lighttoggle = (ToggleButton) findViewById(R.id.lighttoggle);

fantoggle = (ToggleButton) findViewById(R.id.fantoggle);

btnSpeak = (ImageButton) findViewById(R.id.btnSpeak);

btnSpeak.setOnClickListener(new View.OnClickListener() {

/\*\*

\*

\* onclick event listener

\*

\*/

@Override

public void onClick(View v) {

Intent intent = new Intent(

RecognizerIntent.ACTION\_RECOGNIZE\_SPEECH);

intent.putExtra(RecognizerIntent.EXTRA\_LANGUAGE\_MODEL, "en-US");

try {

startActivityForResult(intent, RESULT\_SPEECH);

txtText.setText("");

} catch (ActivityNotFoundException a) {

Toast t = Toast.makeText(getApplicationContext(),

"Ops! Your device doesn't support Speech to Text",

Toast.LENGTH\_SHORT);

t.show();

}

}

});

/\*\*

\*

\* onclick listener for light toggle button

\*

\*/

lighttoggle.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if (lighttoggle.getText().toString().equals("light on")) {

voicedata = "3";

dispdata = "light on";

}

if (lighttoggle.getText().toString().equals("light off")) {

voicedata = "4";

dispdata = "light off";

}

SendDatasdata = new SendData();

sdata.execute();

}

});

/\*\*

\*

\* onclick listener for fan toggle button

\*

\*/

fantoggle.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if (fantoggle.getText().toString().equals("fan on")) {

voicedata = "1";

dispdata = "fan on";

}

if (fantoggle.getText().toString().equals("fan off")) {

voicedata = "2";

dispdata = "fan off";

}

SendDatasdata = new SendData();

sdata.execute();

}

});

}

/\*\*

\*

\* onclick listener for menu creation

\*

\*/

@Override

Public booleanonCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.main, menu);

return true;

}

/\*\*

\*

\* called when an activity is invoked

\*

\*/

@Override

protected void onActivityResult(intrequestCode, intresultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

switch (requestCode) {

case RESULT\_SPEECH: {

if (resultCode == RESULT\_OK && null != data) {

ArrayList<String> text = data

.getStringArrayListExtra(RecognizerIntent.EXTRA\_RESULTS);

voicedata = text.get(0);

if (voicedata.equals("fan start")) {

dispdata = "Fan Start";

voicedata = "1";

}

if (voicedata.equals("fan stop")) {

dispdata = "Fan Stop";

voicedata = "2";

}

if (voicedata.equals("lamp on")) {

dispdata = "Lamp on";

voicedata = "3";

}

if (voicedata.equals("lamp off")) {

dispdata = "Lamp off";

voicedata = "4";

}

if (voicedata.matches("(.\*)switch(.\*)on(.\*)lamp(.\*)")

|| voicedata.matches("(.\*)turn(.\*)on(.\*)lamp(.\*)"))

{

dispdata = "Lamp on";

voicedata = "3";

}

if (voicedata.matches("(.\*)switch(.\*)off(.\*)lamp(.\*)")

|| voicedata.matches("(.\*)turn(.\*)off(.\*)lamp(.\*)"))

{

dispdata = "Lamp off";

voicedata = "4";

}

if (voicedata.matches("(.\*)turn(.\*)on(.\*)bulb(.\*)"))

{

dispdata = "Lamp on";

voicedata = "3";

}

if (voicedata.matches("(.\*)turn(.\*)off(.\*)bulb(.\*)"))

{

dispdata = "Lamp off";

voicedata = "4";

}

if (voicedata.matches("(.\*)switch(.\*)on(.\*)fan(.\*)")

|| voicedata.matches("(.\*)turn(.\*)on(.\*)fan(.\*)"))

{

dispdata = "Fan on";

voicedata = "1";

}

if (voicedata.matches("(.\*)switch(.\*)off(.\*)fan(.\*)")

|| voicedata.matches("(.\*)turn(.\*)off(.\*)fan(.\*)"))

{

dispdata = "Fan off";

voicedata = "2";

}

if (voicedata.length() != 1) {

dispdata = "pardon";

voicedata = "4";

}

// execute thread

SendDatasdata = new SendData();

sdata.execute();

}

}

break;

}

}

@Override

public void onDestroy() {

if (tts != null) {

tts.stop();

tts.shutdown();

}

super.onDestroy();

}

@Override

public void onInit(int status) {

if (status == TextToSpeech.SUCCESS) {

tts.setLanguage(Locale.US);

tts.setLanguage(Locale.getDefault());

} else {

Log.e("TTS", "Initilization Failed!");

}

}

/\*\*

\*

\* class for sending data to remote location

\*

\*/

classSendData extends AsyncTask<String, Void, Void> {

int res = 0;

@Override

protected Void doInBackground(String... params) {

try {

Class.forName("com.mysql.jdbc.Driver");

con = DriverManager.getConnection("jdbc:mysql://"

+ ipaddress.getText().toString() + ":3306/gprseca",

"root", "password");

ps = con.createStatement();

res = ps.executeUpdate("UPDATE instructions SET ins='"

+ voicedata + "' WHERE passkey='MySecret'");

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

/\*\*

\*

\* called when the async task has been completed

\*

\*/

@Override

protected void onPostExecute(Void result) {

if (res > 0)

txtText.setText(dispdata);

else

txtText.setText("Server down");

tts.speak(txtText.getText().toString(), TextToSpeech.QUEUE\_FLUSH, null);

}/\*\*

\*

\* called before the async task has been completed

\*

\*/

@Override

protected void onPreExecute() {

}

@Override

protected void onProgressUpdate(Void... values) {

}

}

}