Sinhgad Institutes

A PROJECT REPORT

ON

Mobile based Home Automation System using IoT and Prediction Algorithm

Submitted to the Savitribai Phule Pune University in the

partial fulfillment of the requirements for the award of degree

of

BACHELOR OF COMPUTER ENGINEERING

BY

SIDDHARTH LODHA	**B120234391**
KEDARNATH SHINDE	**B120234388**
SAPNA TIWARI	**B120234410**
RAJASHRI PATIL	**B120234353**

DEPARTMENT OF COMPUTER ENGINEERING

Accredited by NBA

STES'

SINHGAD COLLEGE OF ENGINEERING

S.N. 44/1, Vadgaon Bk, Off Sinhgad Road Pune - 411041

Sinhgad Institutes

SINHGAD COLLEGE OF ENGINEERING

S.N. 44/1, Vadgaon Bk, Off Sinhgad Road

Pune - 411041

CERTIFICATE

This is to certify that the project report entitled

Mobile based Home Automation System using IOT and Prediction Algorithm

Submitted by

SIDDHARTH LODHA	**B120234391**
KEDARNATH SHINDE	**B120234388**
SAPNA TIWARI	**B120234410**
RAJASHRI PATIL	**B120234353**

is a bonafide work carried out and is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, Pune for the award of the Degree of Bachelor of Computer Engineering

This project work has not been earlier submitted to any other Institute or University for the award of any degree or diploma.

Place : Pune

Date :

Prof. E. Jayanthi
Internal Guide
Department of Computer Engineering

Prof. M. P. Wankhade
Head,
Department of Computer Engineering

Dr. S. D. Lokhande
Principal,SCOE,Pune

External Examiner

Acknowledgements

Every work is source which requires support from many people and areas. It gives us proud privilege to partially complete the project "Mobile based Home Automation System using IoT and Prediction Algorithm " under valuable guidance and encouragement of my guide Prof. E. Jayanthi. I am extremely grateful to our respected H.O.D.(Computer Dept.) Prof. M. P. Wankhade for providing all facilities and every help for smooth progress of seminar. I would like to thank all the Staff Member of Computer Engineering Department for timely help and inspiration for completion of the seminar. At last I would like to thank all the unseen authors of various articles on the Internet, helping me become aware of the research currently ongoing in this field and all my colleagues for providing help and support in my work.

Abstract

Availability of high speed mobile networks like 3G and LTE has lead to tremendous growth in mobile industry, providing various applications and services at the fingertips of the user. One such service is home automation. Home automation involves the control and automation of lighting, heating (such as smart thermostats), ventilation, air conditioning (HVAC), and security, as well as home appliances such as washer/dryers, ovens or refrigerators/freezers that use WiFi for remote monitoring.

The "Home Automation " concept has existed for many years. In 1975, X10 the first general purpose home automation network technology was developed. The home automation market is predicted to have a market value over US 10 billion by the year 2020. As home automation systems becomes more wide spread, they must be able to predict and then adapt to future events.

This proposed system presents the overall design of Home Automation System with low cost and wireless system. This system is designed to assist and provide support in order to fulfill the needs of elderly and disabled in home. Also, the smart home concept in the system improves the standard living at home. Main focus of this system is to control the household equipment's like light, fan, AC etc . This System uses a sequential prediction algorithm, which observes sequence of events to predict the next event in smart environment. This prediction is useful in many scenarios for e.g., predicting inhabitant activities provides a basis for automating interaction with environment and improves the inhabitants comfort. This is achieved using Arduino board and android mobile application.

List of Figures

List of Tables

Abbreviations

1.	IoT	Internet of things
2.	WiFi	Wireless Fidelity
3.	LTE	Long Term Evolution

Table 1: Abbreviations

Contents

CHAPTER 1

INTRODUCTION

1.1 Background And Basics

The term ”Internet of Things” was coined by Peter T. Lewis in a 1985 speech given at a U.S. Federal Communications Commission (FCC) supported wireless session at the Congressional Black Caucus 15th Legislative Weekend Conference. In his speech he states that ”The Internet of Things, or IoT, is the integration of people, processes and technology with connectable devices and sensors to enable remote monitoring, status, manipulation and evaluation of trends of such devices.”

Early home automation began with labor-saving machines. Self-contained electric or gas powered home appliances became viable in the 1900s with the introduction of electric power distribution and led to the introduction of washing machines (1904), water heaters (1889), refrigerators, sewing machines, dishwashers, and clothes dryers[1].

In 1975, the first general purpose home automation network technology, X10, was developed. It is a communication protocol for electronic devices. It primarily uses electric power transmission wiring for signalling and control, where the signals involve brief radio frequency bursts of digital data, and remains the most widely available.Though the concept of smart homes is new in India, considerable amount of work has been carried out in other countries, where smart hornes are already in place.

1.2 Literature Survey

In year 2007, an IEEE paper titled Online Sequential Prediction via Incremental Parsing: The Active LeZi Algorithm, authored by Karthik Gopalratnam and Diane J. Cook was published. In this paper, a sequential prediction algorithm called Active LeZi is constructed using data compression techniques. The basic data compression technique used is LZ78 algorithm. The problem of slow convergence rate in LZ78 algorithm is addressed in Active LeZi algorithm. Such a prediction algorithm can be used to develop a smart environment that acquires and applies knowledge about its inhabitants and their surroundings to adapt to them and meet their comfort and efficiency goals. Such a home could ideally control many aspects of the environment, such as climate,water, lighting, maintenance, and entertainment.

In 2012, an IEEE paper titled , An improved position prediction algorithm based on Active LeZi in Smart Home, authored by Hongqing Fang and Jinjin Ruan, was published. This paper proposes a Time Varying LeZi algorithm (TALZ), based on the Active LeZi (ALZ) algorithm which can be used in position prediction of inhabitants. TALZ results in better position prediction accuracy, also it has smaller time complexity and is suitable for for real-time, accurate position prediction in smart home environments.

In 2015, an IEEE paper titled Mobile based Horne Automation using Internet of Things(IoT), authored by Kumar Mandula, Ramu Parupalli, CH.A.S.Murty, E.Magesh and Rutul Lunagariya, was published. This paper discusses about loT and how it can be used for realizing smart horne automation using a micro-controller based Arduino board and Android mobile app. In this paper, two prototypes namely home automation using Bluetooth in an indoor environment and home automation using Ethernet in an outdoor environment are presented. The various electrical appliances can be controlled through the android app.

1.3 Project Undertaken

1.3.1 Problem Definition

"Mobile based Home Automation "is a Smart Medicine Assistance Application. In this section, Ethernet module is used for connecting Arduino board from any part of the world. Arduino's Ethernet module IP address and Port number can be used to locate remote device connected to the Internet in a smart home environment. Android mobile app can be used to control electrical appliances from a remote location. Ethernet shield is placed just above Arduino board wh ich is connected to RJ-45 for Internet connectivity.

smart home automation concept using low cost Arduino board for controlling various electrical appliances using an Android smart phone. Since loT is one of the upcoming technologies that can be used for horne automation, there are many challenges that are associated with it. One of the major challenges in the lack of standards for integrating various sensors, ap-

SCOE, Dept. of Computer Engineering 4 Year 2016-17 plications and other existing intelligent embedded devices. Providing unique IP addresses for connected devices and privacy security in a smart home environment is another big challenge.

Main objective of project is to manage and control physical objects around us in a more intelligent and meaningful manner and also improve quality of life by providing cost effective living including safety, security and entertainment. Smart objects gather useful contextual data autonomously and send to remote application servers for offering context aware or location based services. The word ”context” can refer to any location information, surrounding environment, people objects that are near by etc so that adaptive and personalized services can be provided to the user..

By using prediction algorithm the smart application will work.It will first store the daily activities in Dictionary of the Algorithm and then according to the users most reffed activities it will assigned a fixed frequency to that particular activity and will take the action according to frequency count.

If user wants to give explicit input to the Dictionary then application can run according the user Given input, all will have log-in and log-out facilities.

1.3.2 Scope Statement

Android controlled Smart Home Automation should be able to control the home appliances wirelessly with effectively and efficiently. Made the home appliances flexible in control, any device capable of Wi-Fi connectivity will able to control the home appliances from remote location.

Application that includes the features of switches and use of advancement technologies of android based smart phones. This can be used to control the switches of home appliances. Proposed system provides easy access of appliances by old aged and handicapped persons.

1.4 Organization of Report

The main body of the report is followed by executive summary giving briey the scope and objectives of the study through the abstract, list of gures in the document, list of tables and abbreviations.

Chapter 1 explains the importance of the topic by explaining the background and basics, literature survey and elaboration of the problem statement.

Chapter 2 presents the information on Project Planning and Management related activities by providing detailed Software Requirement Specication (SRS), Project Process Modelling, Cost and Effort Estimates and Project Scheduling.

Chapter 3 focuses on the analysis and design of the project undertaken.

Chapter4 discusses the testing strategies carried out on the different methodologies studied and applied.

Chapter 5 gives the references.

CHAPTER 2

PROJECT PLANNING AND MANAGEMENT

2.1 Detail System Requirement Specification (SRS)

2.1.1 Technologies to be Used

- AES 128 bit Encryption Decryption algorithm : The Advanced Encryption Standard or AES is a symmetric block cipher used by the U.S. government to protect classified information and is implemented in software and hardware throughout the world to encrypt sensitive data.Features of AES are as follows:

  1.Symmetric key symmetric block cipher

  2.128-bit data, 128/192/256-bit keys

  3.Stronger and faster than Triple-DES

  4.Provide full specification and design details

  5.Software implementable in C and Java.

- Wi-Fi: Wi-Fi is a local area wireless computer networking technology that allows electronic devices to connect to the network, mainly using the 2.4 gigahertz (12 cm) UHF and 5 gigahertz (6 cm) SHF ISM radio bands.

2.1.2 Requirement Specification

- Language Requirements:

  1. JDK 7 and above.

  2. Android studio.

- Hardware Requirments:

  1. Arduino

2.1.3 Product Function

- System provides Registration module to register user, User Register on Application with basic info store in SQLite Database.
- Provides Login module to login user on App.
- After successful login to the application User will able to send On/OFF command withWi-Fi send this message to Arduino device and itcontrol Fan/AC/Bulb Device.

2.2 Cost & Efforts Estimates (Mandatory)

In this project, only Hardware needs to be bought is Arduino microcontroller, GSM module for Arduino,relay switch.

Arduio Uno costs around Rs. 500, GSM module costs around 1000 and relay switch costs around 200.

2.3 Project Process Modeling

Flow of the project is planned as follows: Group formed on date 25 June and registered the group on 28 June. Searched for topics from 26 June to 28 June. Topic selected on 29 June. On July 1st, topic for the project finalized and submitted. From 2nd July to 8 July, suitable platform searched and finalized on 11th of July.

Block Diagram designed according to project idea on 1st August. Detailed study of the IEEE papers, Journals and project idea from 1st August to 20 August. Finalized the project problem definition and abstract on 1st September.

Searched for new ideas that can be implemented as modification. IDEA Matrix and Mathematical model designed from 5 Sept to 7 Sept. Review 1 given on 29 Sept. UML diagrams designed from 30 Sept to 1st October. SRS finalized on 3rd October. Review 2 given on 10th October.

2.4 Project Scheduling

2.4.1 Time Line Chart

Time Chart					
Phase 1	Month		Description	Start Date	Duration
	June		Topic Selection	29-June	10
	July		Topic Finalization	1-July	2
			Platform Selection	8-July	2
			Platform Finalization	11-July	2
	August		Block Diagram	1-August	3
			Synopsis	11-August	10
	September		Abstract	1-September	1
			IDEA Matrix	6-September	3
			Mathematical Model	7-September	5
			Review 1	29-September	1
	October		UMLs	1-October	10
			SRS	3-October	10
			PPT	4-October	3

Table 2.1: Time Line Chart

CHAPTER 3

ANALYSIS & DESIGN

3.1 IDEA matrix

I	D	E	A
Increase: Portability,Safety,Ability to control lights,Inhabitant's comfort	Dynamic: User schedule may have different data-sets and patterns,Can predict event as well as prompt user to take action	Eliminate: Need to inside home to control appliances	Accelerate: Turning off/on of appliances,Prediction of next event
Improve: Quality of life,Automation,Security	Decrease: Power consumption,Manual efforts	Easy to use: user friendly Android app	Adaptive: Can make decision in variety of situations
Intuituve: prediction algorithm makes it intuitive	discipline: username and password ensure security	Evaluate: probability of each of the actions	Aware: it is aware of the user's ations

Table 3.1: IDEA Matrix

3.2 Mathematical Model (Algorithm and Methodology)

Set Theory:

Let S be a main set,

$S \equiv \{S, s, X, Y, T, f_{main}, DD, NDD, f_{friend}, memory\ shared, CPU_{count}\}$

S(system) = Is our proposed system which includes following tuple.

s (initial state at time T) = GUI of Home Automation. The GUI

provides space to enter a query/input for user.

X (input to system)= Input Registration user has to first enter the

details. Also User input like ON/OFF request.

Y (output of system) = Prediction of the user behavior. Light will be

ON of on user request in Home Automation.

T (No. of steps to be performed) = These are the total number of steps required to process.

$f_{main}(main\ algorithm) =$

It contains Process P. Process P contains Input, Output and subordinates functions. It shows how th; prediction will be processed in

$DD_{(deterministicdata)} =$

It contains Database data. Here we have considered ON OFF Trigger values. which contains number on o f f trigger values. Light O

$f_{(friend)} =$

TD And IE. In our system, TD and IE are the friend functions of the main functions. Since we will be using both the functions, bo

Memory shared =

Database will store information like list of receivers, registration details and numbers of receivers. Since it is

$CPU_{count} = In\ our\ system,\ we\ require\ 1\ CPU\ for\ server\ and\ minimum\ 1\ CPU\ for\ client.\ Hence,\ CPU count\ is\ 2.$

Sub ordinate Functions:

Identify the processes as P.

S = { I, O, P ....}

P = {TD, AD}

Where,

1. TD is Trigger data for Arduino .

2. AD is Arduino Data.

3. P is processes.

TD= {U, MAX, LG}

Where,

U=ON OFF trigger.

MAX = {0, 1}

LG is output where Light ON or OFF base on U .

AD= {IG, Prediction Techniques, Info}

Where,

LG is input which is given to AD.

Arduino is use for Glow Light ON or OFF.

Algorithm:

- Step 1: register on App with Mobile no, UserName and Password.
- Step 2: Get user Trigger value.
- Step 3: call Arduino function.

    Step 3.1: Get U as Input to TD.

    Step 3.2: for i=0 to MAX.

  //MAX = maximum no of Trigger to be generated by user.

- Step 4: trigger goes to TD.

    Step 4.1: Get TD as Input.

    Step 4.2: Call Arduino function.

    Step 4.3: Process trigger data.

- Step 5: Display Result Light ON or OFF.
- Step 6: Stop.

3.3 Feasibility Analysis (NP Completeness Analysis)

The feasibility study is an evaluation and analysis of the potential of a proposed project which is based on extensive investigation and research to support the process of decision making. In complexity theory, a decision problem is P-complete (complete for the complexity class P) if it is in P and every problem in P can be reduced to it by an appropriate reduction. The notion of P-complete decision problems is useful in the analysis of:

1. Which problems are difficult to parallelize effectively.
2. Which problems are difficult to solve in limited space.

In computational complexity theory, a decision problem is NP-complete when it is both in NP and NP-hard. The set of NP-complete problems is often denoted by NP-C or NPC. The abbreviation NP refers to 'non-deterministic polynomial time. NP-complete problems are in NP, the set of all decision problems whose solutions can be verified in polynomial time; NP may be equivalently defined as the set of decision problems that can be solved in polynomial time on a non-deterministic Turing machine. A problem p in NP is NP-complete if every other problem in NP can be transformed (or reduced) into p in polynomial time.

According to the referred documents, P complete problems are solvable and work on real time data. Examples of P complete problems:

1. Circuit value problem
2. Linear programming
3. Horn satisfiablity

NP complete problems are also solvable but not necessarily working on real time data. Examples of NP complete problems are:

1. Apriory algorithm
2. Naive Byes Algorithm
3. ANN
4. Data Mining
5. Knapsack
6. Travelling Salesman problem

NP hard problems are unsolvable.

After studying and understanding the problem definition of the project, and after referring to the Feasibility Analysis documents and journals, this project title is concluded to come under NP complete problems.

3.4 Architecture Diagram

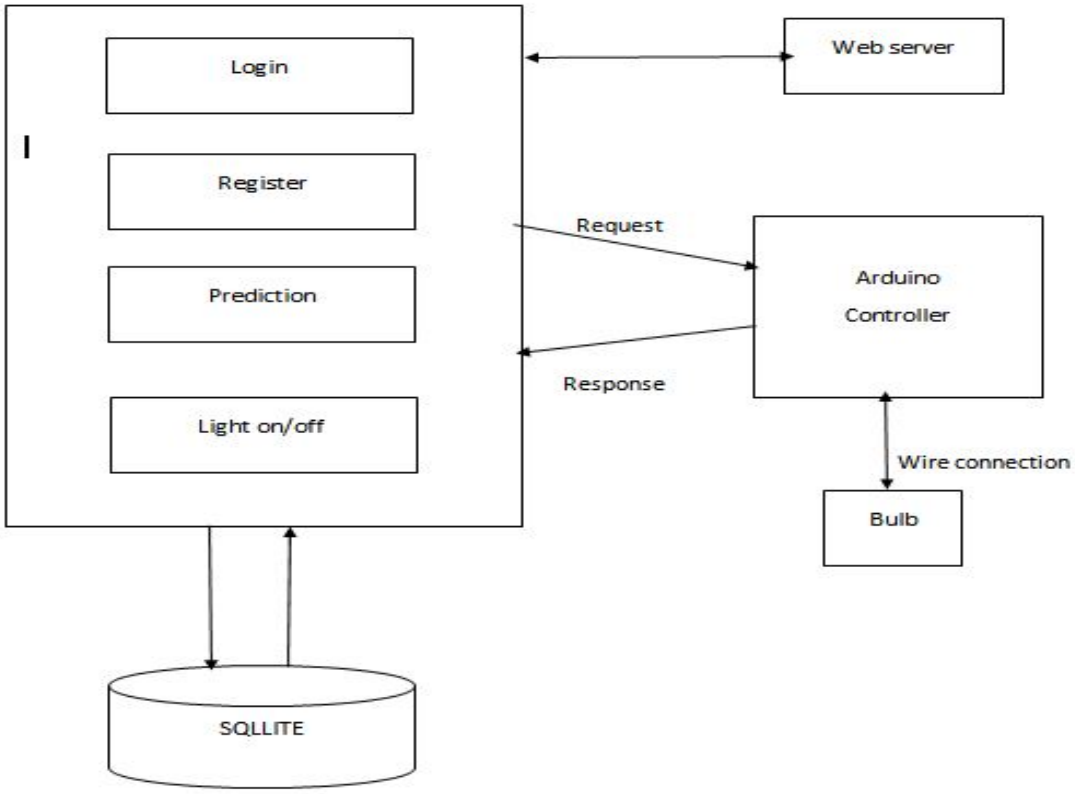

Figure 3.1: Architecture Diagram

3.5 UML Diagrams

3.5.1 Use-Case Diagrams

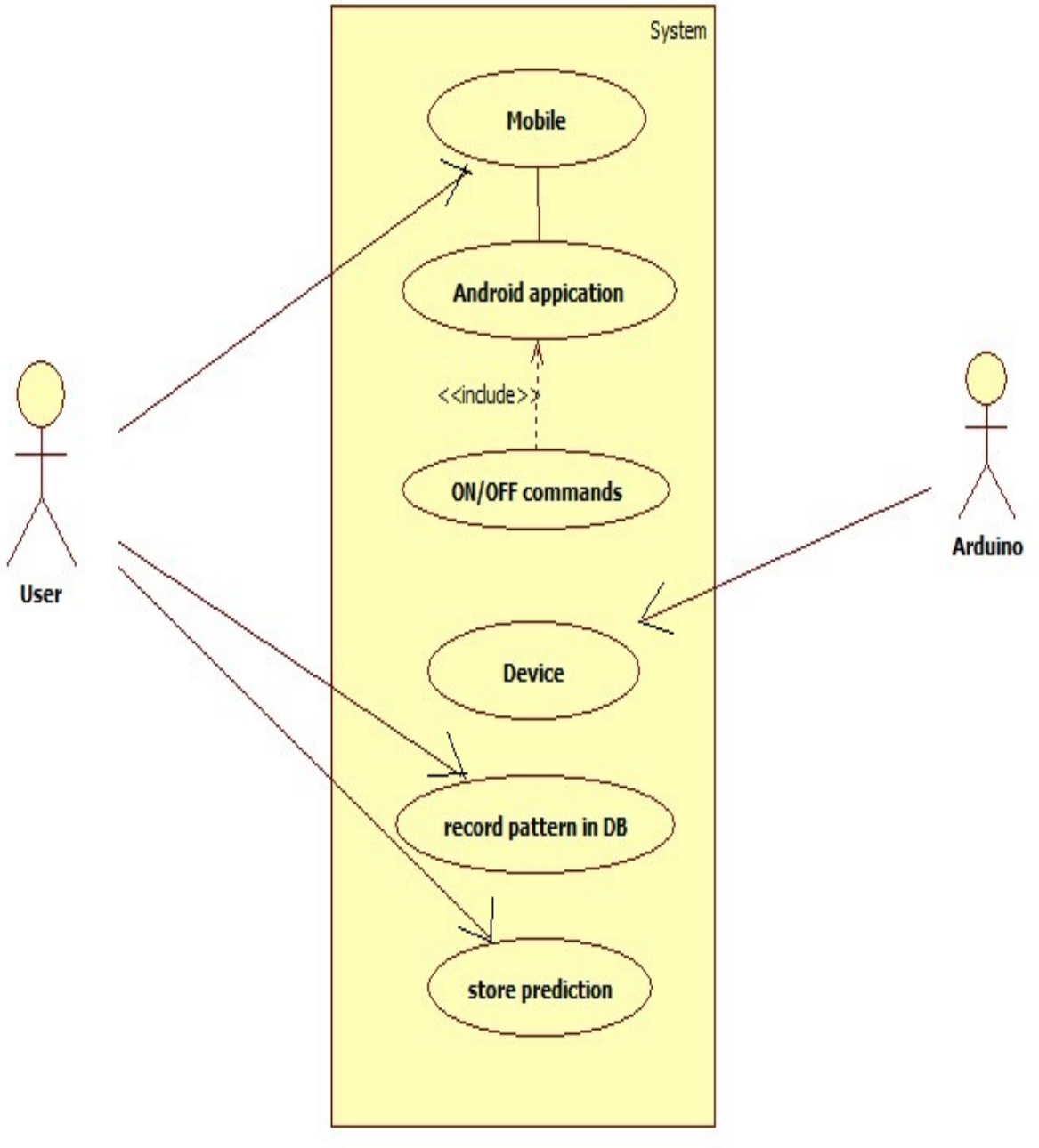

Figure 3.2: Use Case Diagram

3.5.2 Activity Diagram

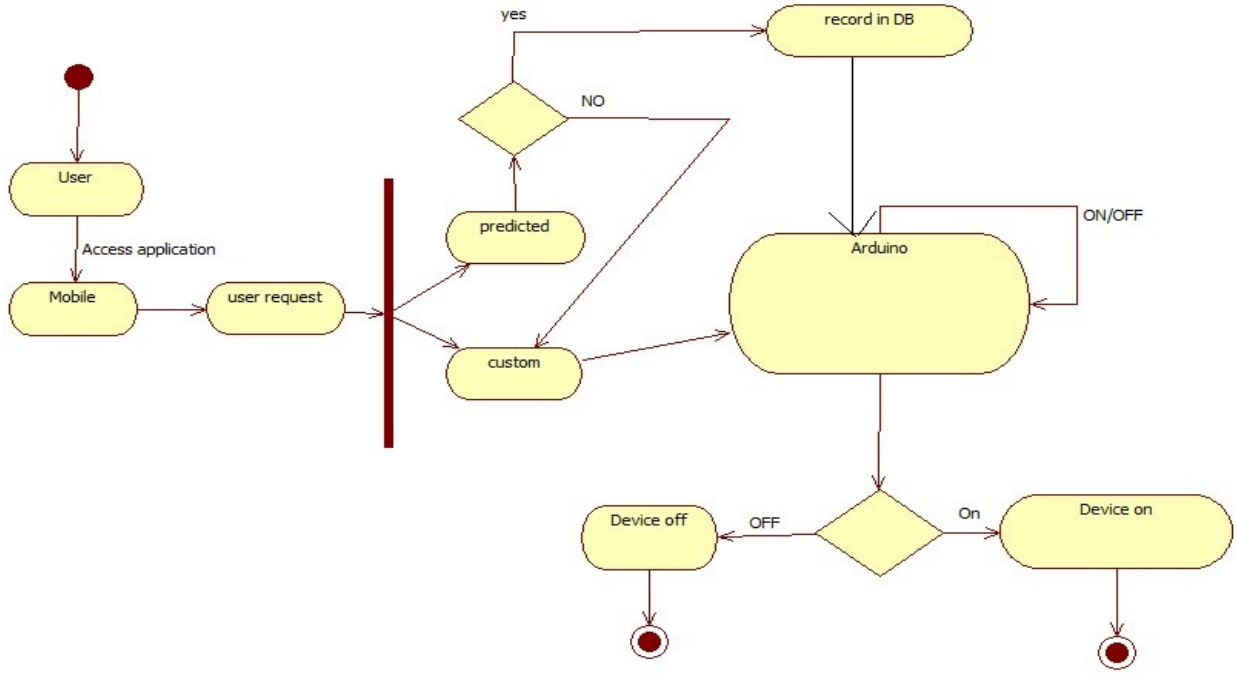

Figure 3.3: Activity Diagram

3.5.3 Class Diagrams

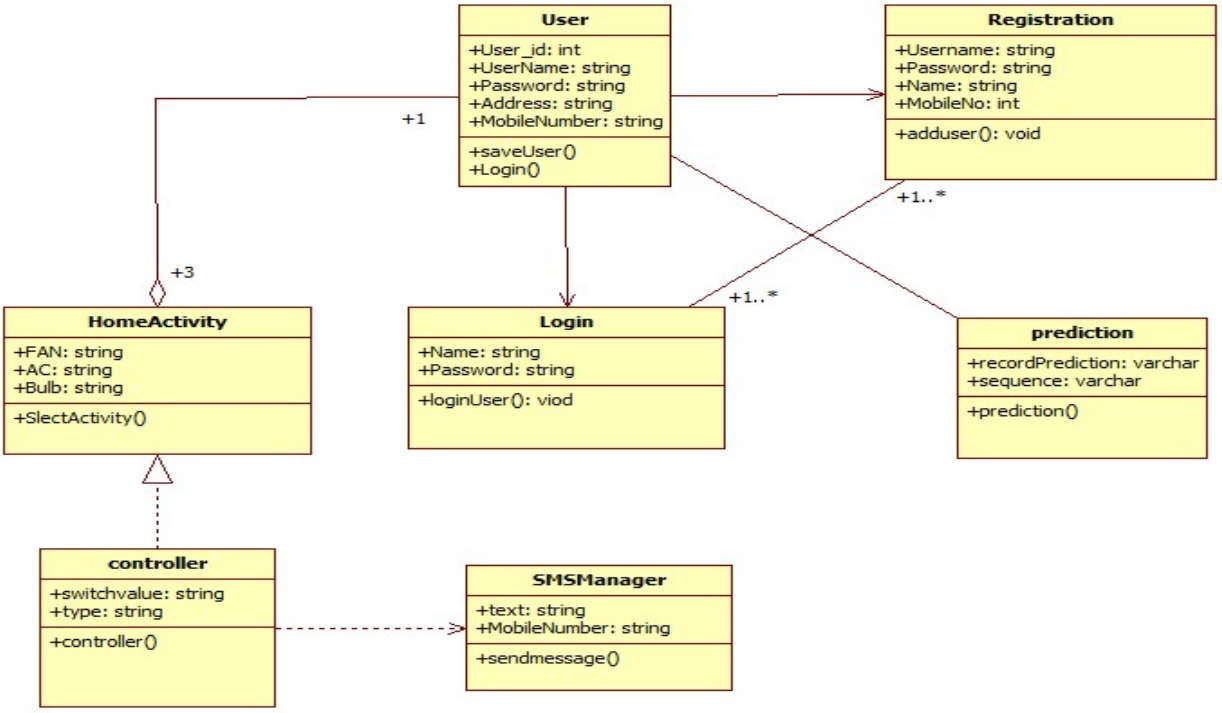

Figure 3.4: Class Diagram

3.5.4 ER Diagrams

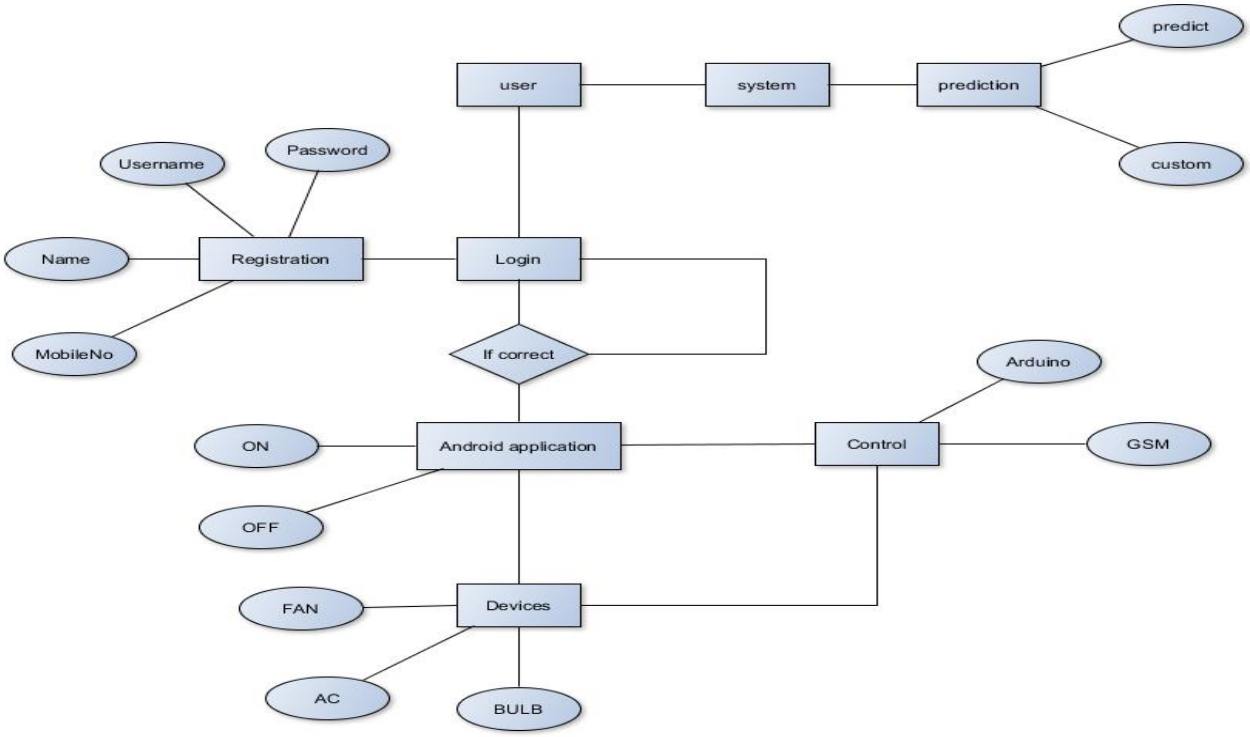

Figure 3.5: ER Diagram

3.5.5 Sequence Diagrams / DFDs

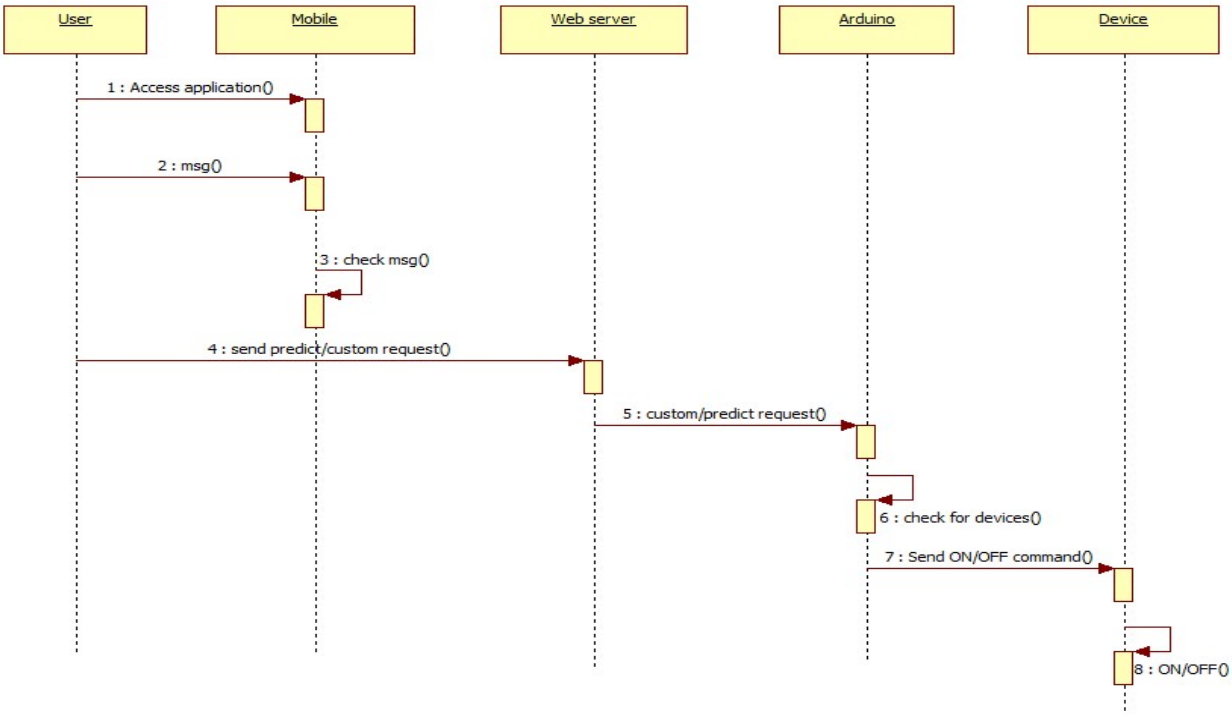

Figure 3.6: Sequence Diagram

3.5.6 State Transition Diagrams

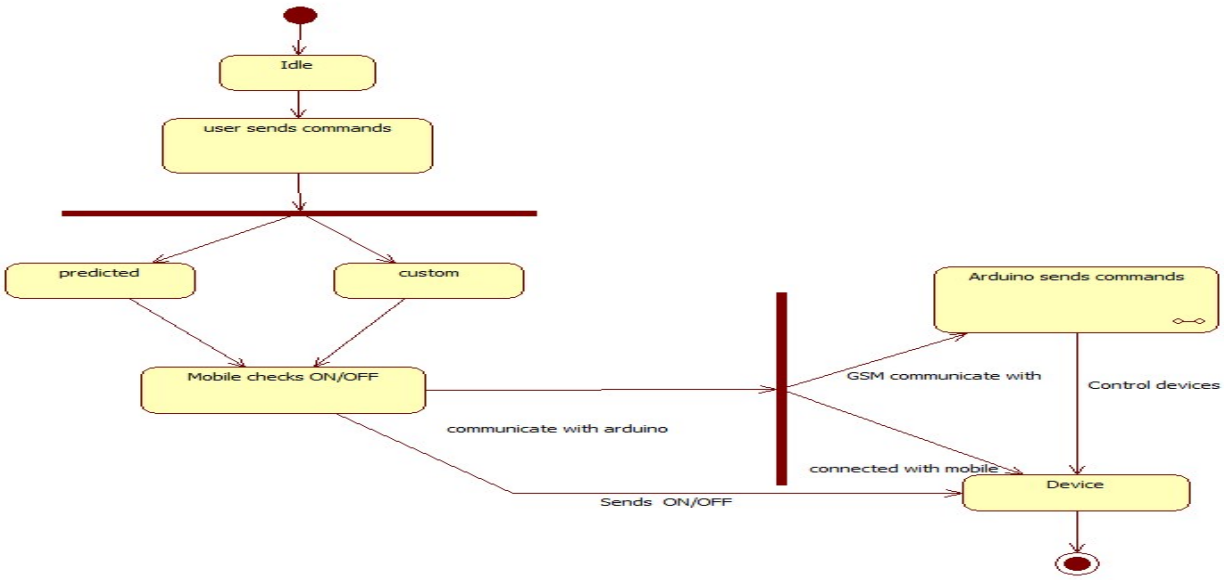

Figure 3.7: State Transition Diagram

3.5.7 Deployment Diagrams

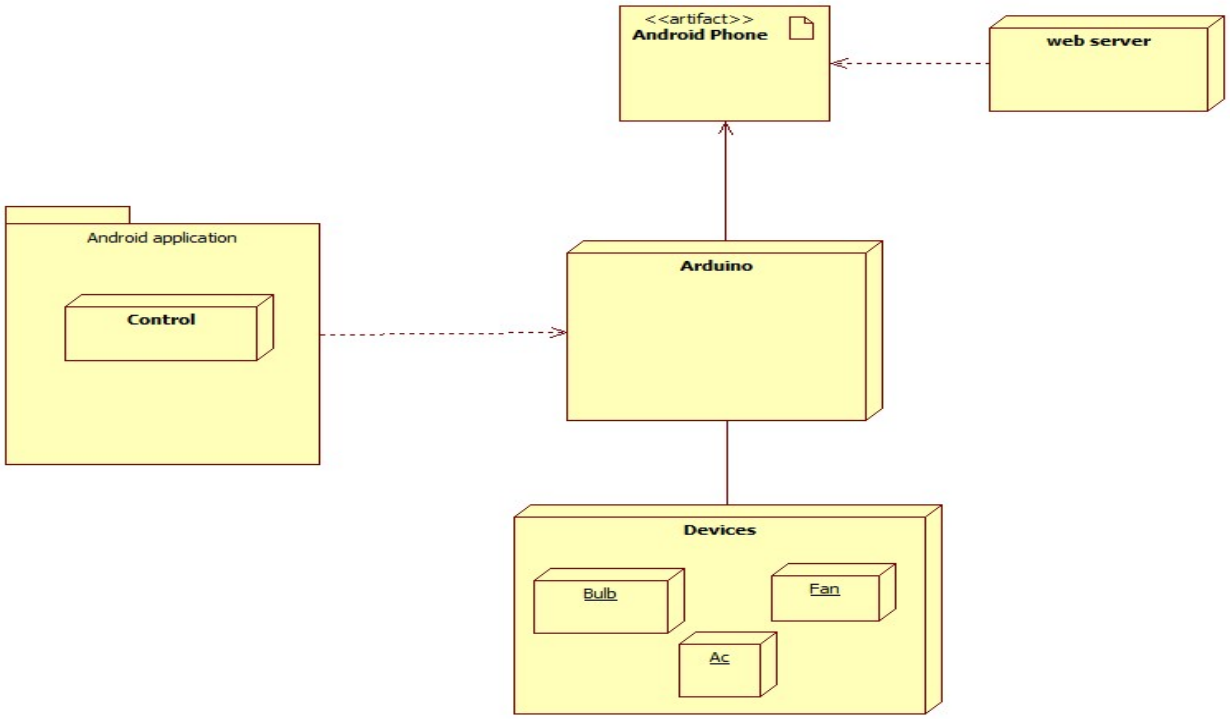

Figure 3.8: Deployment Diagram

CHAPTER 4

IMPLEMENTATION & CODING

4.1 Introduction

This chapter covers the listing of the code for major functionalities. Also covers the classes along with the responsibilities. For implementation purpose we need Java J2SE and JDK J2SE (Java 2 Standard Edition) Java is the language for development of the project. JDK is the development kit used to compile java programs. We have implemented Android application module using Android Studio and the server module is implemented in Eclipse IDE platform. Tomcat server is used. Arduino IDE is used to run code for interface between Android app and Anduino UNO microcontroller.

4.2 Database Schema

List of tables:

1. actions

2. users

4.2.1 actions Table

id - unique id which identifies the action

datetime - stores the date and time at which action is performed.

lightstatus - stores whether the light is ON/OFF

roomname - identifies the room

id	datetime	lightstatus	roomname

Table 4.1: actions

4.2.2 users Table

id - unique id for each user.

emailid - email id of the user.

fcmid - unique fcm id of the application

mobileno - mobile number of the user.

name - name of the user

password - password for log in.

id	emailid	fcmid	mobileno	name	password

Table 4.2: users

4.3 Operational Details & Classes

4.3.1 Operations

Home Automation App:

User is able to sign up and log into the app. He can control the operation of the lights in two rooms, drawing room and dining room.The app also displays the history of actions performed.

Server:

Manage the database through request sent or received. Authentication process is done through server. The server used in this system will be Tomcat server.

Database:

Store the users' information such as id, email id, name, mobile no., FCM id and password. It also stores user actions' information such as date time, status of light and room name.

4.3.2 Major Classes

Activity_Login.java :-

calls makeLoginTask() : This method creates the login task.

Activity_SignUp.java :-

call smakeRegisterTask() : allows user to sign up for the app.

MainActivity.java :-

calls makestatusTask() : sets the status of the lights.

calls getdate() : returns date in yyyyMMdd HH mm ss format

Prediction.java :-

calls getDiningOnPrediction(): to predict when dining room light should be switched ON

calls getDiningOffPrediction(): to predict when dining room light should be switched OFF

calls getDrawingOnPrediction(): to predict when drawing room light should be switched ON

calls getDrawingOffPrediction(): to predict when drawing room light should be switched OFF

4.3.3 Code Listing

==========Activity_Login.java==========

```
package com.example.abc.homeautomation;

import android.Manifest;
import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Build;
import android.os.Bundle;
```

```
import android.support.v4.app.ActivityCompat;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

import com.android.volley.Request;

import com.android.volley.RequestQueue;

import com.android.volley.Response;

import com.android.volley.VolleyError;

import com.android.volley.toolbox.Volley;

import org.json.JSONException;

import org.json.JSONObject;

import java.util.HashMap;

import java.util.Map;

import database.SQLiteAdapter;


public class Activity_Login extends Activity {

    RSA rsa = new RSA();

    //edittext declaration

    EditText edtUserName,edtPassword;

    String strUserName,strPassword;

    //Button declaration

    Button btnLogin;

    TextView textSignup;

    private ProgressDialog pDialog;

    SQLiteAdapter dbhelper;

    String flag="0";
```

```
    public static String userLogin="";
    public static  int PERMISSION_REQUEST_CODE=101;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);//set layout
        if (Build.VERSION.SDK_INT>=23){

            checkPermission();
        }        dbhelper=new SQLiteAdapter(getApplicationContext());

        //call sqllite database constructor
      /*  dbhelper=new SQLiteAdapter(getApplicationContext());
        dbhelper.openToRead();
        int a=dbhelper.Get_Total_LoginStatus();
if(a>0){
             finish();
             Intent i=new Intent(getApplicationContext(),MainActivity.class);
             startActivity(i);

         }
 dbhelper.close();*/
 //initilisation of edittext edtUserName=(EditText)findViewById(R.id.editTextUserName);
edtPassword=(EditText)findViewById(R.id.editTextPassword);
    btnLogin=(Button)findViewById(R.id.buttonLogin);
    btnLogin.setOnClickListener(new View.OnClickListener() {
            @Override
public void onClick(View v) {
```

```
//get data from editttext
strUserName=edtUserName.getText().toString();
strPassword=edtPassword.getText().toString();


        if(!strUserName.equals(""))
                { if(!strPassword.equals(""))
                   {
                   //open database to read data from table
/*
dbhelper.openToRead();
ArrayList<User> userlist= dbhelper.retrieveAllUser();
for(int i=0;i<userlist.size();i++)        {
        String username=userlist.get(i).getUsername();
        String password=userlist.get(i).getPassword();
            String name=userlist.get(i).getName();
                        //check username and password
                            System.out.println("## db password:"+password);
                            System.out.println("## str password:"+strPassword);
                           String decrypt_password =rsa.Decrypt(password, rsa.getPrivateKey()).trim(
                            System.out.println("## decrypt_password:"+decrypt_password);
                if(strUserName.equals(username)&&strPassword.equals(decrypt_password))            {
                        }dbhelper.close();
if(flag.equals("1")){
*/makeLoginTask();


/*
}else {

    Toast.makeText(getApplicationContext(), "Login Failed", Toast.LENGTH_SHORT).show();
```

```
                    edtPassword.setText("");                edtPassword.setHint("Please Enter Pa
                    edtUserName.setText("");
                    edtUserName.setHint("Please Enter UserName");
                 }

*/
                }else{
                    edtPassword.setText("");
                    edtPassword.setHint("Please Enter Password");
                    edtPassword.requestFocus();
                 }

              }else{
                    edtUserName.setText("");
                    edtUserName.setHint("Please Enter UserName");
                    edtUserName.requestFocus();
              }


          }       });

textSignup=(TextView)findViewById(R.id.textViewSignup);
        textSignup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                //start signup Activity
              Intent i=new Intent(getApplicationContext(),Activity_SignUp.class);
                startActivity(i);
```

```
            }
        });
    }

public static boolean hasPermissions(Context context, String... permissions) {

        for (String permission : permissions) {
            if (ActivityCompat.checkSelfPermission(context, permission) != PackageManager.PERMISSION
                return false;
            }
        }

        return true;
    }

    private void checkPermission(){
        String[] PERMISSIONS = {android.Manifest.permission.SEND_SMS,android.Manifest.permission.ACC

        if(!hasPermissions(this, PERMISSIONS)){
            ActivityCompat.requestPermissions(this, PERMISSIONS, PERMISSION_REQUEST_CODE);
        }


    }

    private void makeLoginTask() {
        pDialog= ProgressDialog.show(Activity_Login.this, "", "Please Wait", true, false); //show pr
```

```
        Map<String, String> params=new HashMap<>();
        params.put("emailid",strUserName);
        params.put("password",strPassword);
        System.out.println("##  params.toString() " + params.toString());
        RequestQueue requestQueue = Volley.newRequestQueue(getApplicationContext());
        CustomRequest jsObjRequest = new CustomRequest(Request.Method.POST, ProjectConfig.Login, par

        requestQueue.add(jsObjRequest);
    }

    private Response.ErrorListener createRequestErrorListener() {
        return new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                Log.i("##", "##" + error.toString());
                pDialog.dismiss();
            }
        };
    }

    private Response.Listener<JSONObject> createRequestSuccessListener() {
        return new Response.Listener<JSONObject>() {
            @Override
            public void onResponse(JSONObject response) {

                pDialog.dismiss();
                String status = "";

                System.out.println("## in shake service response:" + response.toString());
```

```
                // ## in shake service response:{"message":"User Login Success","status":"Success"}
                try {
                    status=  response.getString("status");

                  String msg=response.getString("message");
                    if(status.equals("Success"))
                    {

                        pDialog.dismiss();
                        dbhelper.openToWrite();
                        dbhelper.deleteLoginStatus();
                        dbhelper.close();
                        dbhelper.openToWrite();
                        dbhelper.insertLoginStatus(1,"1");
                        dbhelper.close();
    //Show message
                        Toast.makeText(getApplicationContext(), msg, Toast.LENGTH_SHORT).show();
                        finish();
                        //start MAin activity
                        Intent i=new Intent(getApplicationContext(),MainActivity.class);
                        startActivity(i);


                    }
                    else
                    {            pDialog.dismiss();

            Toast.makeText(getApplicationContext(),status,Toast.LENGTH_SHORT).show();
```

```
                    }
                } catch (JSONException e) {
                    e.printStackTrace();
                }

            }
        };

    }

}

//============================================================

/*

=============MainActivity.java=============

*/

package com.example.abc.homeautomation;

import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
```

```
import android.view.View;

import android.widget.Button;

import android.widget.CompoundButton;

import android.widget.Switch;

import android.widget.Toast;

import com.android.volley.Request;

import com.android.volley.RequestQueue;

import com.android.volley.Response;

import com.android.volley.VolleyError;

import com.android.volley.toolbox.Volley;

import org.json.JSONException;

import org.json.JSONObject;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.HashMap;

import java.util.Map;

import database.SQLiteAdapter;


public class MainActivity extends Activity {


    private Switch light1,light2;

    SQLiteAdapter dbhandler;


    String result;

    String time,room,status;

    ProgressDialog pDialog;

  Button buttonHistory;

    String light1_Status,light2_Status;
```

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



        dbhandler=new SQLiteAdapter(getApplicationContext());


        dbhandler.openToRead();
        light1_Status=  dbhandler.getLight1Status();
        light2_Status=  dbhandler.getLight2Status();
        light1 = (Switch) findViewById(R.id.light1);
        light2 = (Switch) findViewById(R.id.light2);
        dbhandler.close();
if(light1_Status.equals("-1"))
        {


            dbhandler.openToWrite();
            dbhandler.insertStatus("1", "OFF");
            dbhandler.close();
            /*String time=getdate();
            dbhandler.openToWrite();
            dbhandler.insertHistory("Drawing room",time,"OFF");
            dbhandler.close();
*/
        }else{
            if(light1_Status.equals("ON"))
            {
```

```
                light1.setChecked(true);
            }else{
                light1.setChecked(false);
            }


        }

        if(light2_Status.equals("-1"))
        {
            dbhandler.openToWrite();
            dbhandler.insertStatus("2","OFF");
            dbhandler.close();

/*
            String time=getdate();
            dbhandler.openToWrite();
            dbhandler.insertHistory("Dining room",time,"OFF");
            dbhandler.close();
*/

        }else{

            if(light2_Status.equals("ON"))
            {
                light2.setChecked(true);
            }else{
                light2.setChecked(false);
```

```
    }

}

buttonHistory=(Button)findViewById(R.id.buttonHistory);
buttonHistory.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        Intent i=new Intent(getApplicationContext(),History_Activity.class);
        startActivity(i);

    }
});



light1.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {

    @Override
    public void onCheckedChanged(CompoundButton buttonView,
                                 boolean isChecked) {


        if (!isChecked) {
            //off
            new SendSMS(ProjectConfig.MobilenNumber,"@1F").execute();

            time=getdate();
```

```
            room="Drawing room";
            status="OFF";
            makestatusTask(time,"Drawing","OFF");
            dbhandler.openToWrite();
            dbhandler.update_light(1, "OFF");
            dbhandler.close();
            dbhandler.openToWrite();
            dbhandler.insertHistory(room,time,status);
            dbhandler.close();

        } else {
            //on
            new SendSMS(ProjectConfig.MobilenNumber,"@1O").execute();
            String time=getdate();

            room="Drawing room";
            status="ON";
            makestatusTask(time,"Drawing","ON");
            dbhandler.openToWrite();
            dbhandler.update_light(1,"ON");
            dbhandler.close();

            dbhandler.openToWrite();
            dbhandler.insertHistory(room,time,status);
            dbhandler.close();

        }

    }
```

```
        });


        light2.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
@Override
public void onCheckedChanged(CompoundButton buttonView,
                                        boolean isChecked) {
    System.out.println("## isChecked:"+isChecked);
                if (!isChecked) {
                    new SendSMS(ProjectConfig.MobilenNumber,"@2F").execute();


                     time=getdate();
                    room="Dining room";
                    status="OFF";
                    makestatusTask(time,"Dining","OFF");
                    dbhandler.openToWrite();
                    dbhandler.update_light(2,"OFF");
                    dbhandler.close();
                    dbhandler.openToWrite();
                    dbhandler.insertHistory(room,time,status);
                    dbhandler.close();
                } else {
                    new SendSMS(ProjectConfig.MobilenNumber,"@2O").execute();


                    time=getdate();
                    room="Dining room";
                    status="ON";
                    makestatusTask(time,"Dining","ON");
                    dbhandler.openToWrite();
```

```
                    dbhandler.update_light(2,"ON");
                    dbhandler.close();

                    dbhandler.openToWrite();
                    dbhandler.insertHistory(room,time,status);
                    dbhandler.close();


                }

            }
        });


    }


String getdate()
{
    DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
    Date date = new Date();
    return  ""+dateFormat.format(date);
}

    public void makestatusTask(String time,String roomname,String status)
    {
        pDialog= ProgressDialog.show(MainActivity.this, "", "Please Wait", true, false); //show proc

        Map<String, String> params=new HashMap<>();
```

```
        params.put("datetime",time);
        params.put("roomname",roomname);
        params.put("lightstatus",status);
    System.out.println("##  param - " + params.toString());
        RequestQueue requestQueue = Volley.newRequestQueue(getApplicationContext());
        CustomRequest jsObjRequest = new CustomRequest(Request.Method.POST, ProjectConfig.savestaus,
        requestQueue.add(jsObjRequest);
    }

    private Response.ErrorListener createRequestErrorListener() {
        return new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                Log.i("##", "##" + error.toString());
                pDialog.dismiss();
            }
        };
    }

    private Response.Listener<JSONObject> createRequestSuccessListener() {
        return new Response.Listener<JSONObject>() {
            @Override
            public void onResponse(JSONObject response) {

                Log.i("#######", "####" + response.toString());
                //{"message":"Action Saved ?","status":"Saved"}
                pDialog.dismiss();
                try {
                    String result=response.getString("status");
```

```
                    if(result.equals("Saved"))
                    {
                        Log.d("#########"," room status :- "+ room+" "+time+" "+status);
                        Toast.makeText(getApplicationContext(),response.getString("message"),Toast.L
                        dbhandler.openToWrite();
                        dbhandler.insertHistory(room,time,status);
                        dbhandler.close();

                    }
                    else
                    {

                        Toast.makeText(getApplicationContext(),response.getString("message"),Toast.L

                    }
                } catch (JSONException e) {
                    e.printStackTrace();
                    System.out.println("##  e " + e);
                }

            }
        };

    };

public class SendSMS extends AsyncTask<Void, Void, Void> {

        String number, msg;
```

```
public SendSMS(String number, String msg) {
    this.number = number;
    this.msg = msg;
}

@Override
protected void onPreExecute() {
    super.onPreExecute();
}

@Override
protected Void doInBackground(Void... params) {
    result = SendSms.sendSMS(number, msg);
    Log.e("##########"," msg :- "+msg);
    return null;
}
@Override
protected void onPostExecute(Void aVoid) {
    super.onPostExecute(aVoid);
    Log.d("##############", "send Sms result :- " + result);
    try {
        JSONObject obj=new JSONObject(result);

        String errorresult=obj.getString("ErrorMessage");

        Toast.makeText(getApplicationContext()," msg "+errorresult,Toast.LENGTH_SHORT).show(
    } catch (JSONException e) {
        e.printStackTrace();
    }
```

```
}}

}

//=========================================================


//=========================================================


=============Prediction.java=============


package Test;


import java.text.DateFormat;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.List;


import Bean.Actions;

import DAO.ActionsDAO;


public class Prediction {


        public static String getDiningOnPrediction() throws ParseException

        {

                List<Actions> arrdiningon = ActionsDAO.getSingleRoomLightStatus("Dining","ON");
```

```
int sizeofarrdiningon=arrdiningon.size();

String datetimediningon[]=new String[sizeofarrdiningon];
System.out.println("Dining Room Light ON List   "+arrdiningon);
if(arrdiningon != null)
{

        int i=0;
        while(i < arrdiningon.size())
        {  String datetime=arrdiningon.get(i).getDatetime();
                datetimediningon[i]=datetime;
                String roomname=arrdiningon.get(i).getRoomname();
                String lightstatus=arrdiningon.get(i).getLightstatus();
                System.out.println("Date Time "+datetime+"\t Room Name  "+roomname+

                i++;
        }
}else
{
        System.out.print("Problem In getting SingleRoomLightStatus(Dining,ON)");
}

if(arrdiningon.size()>1)
{



System.out.println("\n\n");
for(int j=0;j<sizeofarrdiningon;j++)
{
```

```
        System.out.println("Date And Time of Dining Light ON  "+datetimediningon[j]
}

System.out.println("\n\n");
int differentinlighton=0;
for(int j=0;j<(sizeofarrdiningon-1);j++)
{
        String previousday=datetimediningon[j];
        String nextday=datetimediningon[j+1];

        System.out.println("Previous Day   "+previousday);
        System.out.println("Next Day  "+nextday);

                DateFormat formatter = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
                Date datePrevious = formatter.parse(previousday);

                Date dateCurrent = formatter.parse(nextday);
                long diff = dateCurrent.getTime() - datePrevious.getTime();
                diff=diff/1000;
                System.out.println("Date Difference in seconds  "+diff);
                diff=diff/60;
                System.out.println("Date Difference in minutes  "+diff);
                diff=diff/60;
                System.out.println("Date Difference in hrs  "+diff);
                differentinlighton=(int) (differentinlighton+diff);

}
System.out.println("\n\n");
System.out.println("Overall Difference in Hrs   "+differentinlighton);
```

```
int avgdiff=differentinlighton/(sizeofarrdiningon-1);
System.out.println("Average Difference in Hrs    "+avgdiff);



System.out.println("\n\n");
DateFormat formatter = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");

String lastdate=datetimediningon[sizeofarrdiningon-1];

System.out.println("Last Date And Time    "+lastdate);

Date currdate = new Date();

System.out.println("Current Date And Time   "+formatter.format(currdate));

String currdatenadtime=formatter.format(currdate).toString();

Date datelast = formatter.parse(lastdate);
Date currentdatetime=formatter.parse(currdatenadtime);



long diffnowandlast = currentdatetime.getTime() - datelast.getTime();
diffnowandlast=diffnowandlast/1000;
System.out.println("Date Difference in seconds   "+diffnowandlast);
diffnowandlast=diffnowandlast/60;
System.out.println("Date Difference in minutes   "+diffnowandlast);
diffnowandlast=diffnowandlast/60;
System.out.println("Date Difference in hrs   "+diffnowandlast);
```

```
            System.out.println("\n\n");
            if(diffnowandlast>avgdiff)
            {
                    System.out.println("\n\nPrediction : You should have ON Light Of Dinning Roo
                    return "Prediction : You should have ON Light Of Dinning Room...";
            }
            else if(diffnowandlast<avgdiff)
            {
                    System.out.println("\n\nPrediction : You have to wait ON Light Of Dinning Ro
                    return "Prediction : You have wait ON Light Of Dinning Room...";
            }
            else if(diffnowandlast==avgdiff)
            {
                    System.out.println("\n\nPrediction : ON The Light of Dining Room...\n\n");
                    return "Prediction : ON The Light of Dining Room...";
            }
            else
            {
                    System.out.println("\n\nPrediction : Can not predict to ON The Light of Dinn
                    return "Prediction : Can not predict to ON The Light of Dinning Room...";
            }

             }

             return "Prediction : Can not predict to ON The Light of Dinning Room...";

    }
```

```
public static String getDiningOFFPrediction() throws ParseException
{
        List<Actions> arrdiningoff = ActionsDAO.getSingleRoomLightStatus("Dining","OFF");


        int sizeofarrdiningoff=arrdiningoff.size();

        String datetimediningoff[]=new String[sizeofarrdiningoff];
        System.out.println("Dining Room Light OFF List   "+arrdiningoff);
        if(arrdiningoff != null)
        {

                int i=0;
                while(i < arrdiningoff.size())
                {  String datetime=arrdiningoff.get(i).getDatetime();
                        datetimediningoff[i]=datetime;
                        String roomname=arrdiningoff.get(i).getRoomname();
                        String lightstatus=arrdiningoff.get(i).getLightstatus();
                        System.out.println("Date Time "+datetime+"\t Room Name  "+roomname+


                        i++;
                }
        }
        else
        {
                System.out.print("Problem In getting SingleRoomLightStatus(Dining,OFF)");
        }
```

```
if(arrdiningoff.size()>1)
{
System.out.println("\n\n");
for(int j=0;j<sizeofarrdiningoff;j++)
{
        System.out.println("Date And Time of Dining Light OFF  "+datetimediningoff[
}



System.out.println("\n\n");
int differentinlightoff=0;
for(int j=0;j<(sizeofarrdiningoff-1);j++)
{   String previousday=datetimediningoff[j];
        String nextday=datetimediningoff[j+1];

        System.out.println("Previous Day   "+previousday);
        System.out.println("Next Day  "+nextday);

                DateFormat formatter = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
                Date datePrevious = formatter.parse(previousday);

                Date dateCurrent = formatter.parse(nextday);
                        long diff = dateCurrent.getTime() - datePrevious.getTime();
                diff=diff/1000;
                System.out.println("Date Difference in seconds  "+diff);
                diff=diff/60;
                System.out.println("Date Difference in minutes  "+diff);
                diff=diff/60;
                System.out.println("Date Difference in hrs  "+diff);
```

```
                differentinlightoff=(int) (differentinlightoff+diff);

        }
        System.out.println("\n\n");
        System.out.println("Overall Difference in Hrs   "+differentinlightoff);

        int avgdiff=differentinlightoff/(sizeofarrdiningoff-1);
        System.out.println("Average Difference in Hrs   "+avgdiff);



        System.out.println("\n\n");
        DateFormat formatter = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");

        String lastdate=datetimediningoff[sizeofarrdiningoff-1];

        System.out.println("Last Date And Time   "+lastdate);

        Date currdate = new Date();

        System.out.println("Current Date And Time  "+formatter.format(currdate));

        String currdatenadtime=formatter.format(currdate).toString();

        Date datelast = formatter.parse(lastdate);
       Date currentdatetime=formatter.parse(currdatenadtime);



       long diffnowandlast = currentdatetime.getTime() - datelast.getTime();
       diffnowandlast=diffnowandlast/1000;
```

```
System.out.println("Date Difference in seconds  "+diffnowandlast);
diffnowandlast=diffnowandlast/60;
System.out.println("Date Difference in minutes  "+diffnowandlast);
diffnowandlast=diffnowandlast/60;
System.out.println("Date Difference in hrs  "+diffnowandlast);



System.out.println("\n\n");
if(diffnowandlast>avgdiff)
{
        System.out.println("\n\nPrediction : You should have OFF the Light Dinning R
        return "Prediction : You should have OFF the Light Dinning Room...";
}
else if(diffnowandlast<avgdiff)
{
        System.out.println("\n\nPrediction : You have wait to OFF the Light Of Dinni
        return "Prediction : You have wait to OFF the Light Of Dinning Room...";
}
else if(diffnowandlast==avgdiff)
{
        System.out.println("\n\nPrediction : OFF The Light of Dinning Room...\n\n");
        return "Prediction : OFF The Light of Dinning Room...";
}
else
{
        System.out.println("\n\nPrediction : Can not predict to OFF The Light of Din
        return "Prediction : Can not predict to OFF The Light of Dinning Room...";
}
```

```
            }
            return "Prediction : Can not predict to OFF The Light of Dinning Room...";
    }
    public static String getDrawingOnPrediction() throws ParseException
    {
            List<Actions> arrdrawingon = ActionsDAO.getSingleRoomLightStatus("Drawing","ON");


            int sizeofarrdrawingon=arrdrawingon.size();

            String datetimedrawingon[]=new String[sizeofarrdrawingon];
            System.out.println("Drawing Room Light ON List   "+arrdrawingon);
            if(arrdrawingon != null)
            {

                    int i=0;
                    while(i < arrdrawingon.size())
                    {


                            String datetime=arrdrawingon.get(i).getDatetime();
                            datetimedrawingon[i]=datetime;
                            String roomname=arrdrawingon.get(i).getRoomname();
                            String lightstatus=arrdrawingon.get(i).getLightstatus();
                            System.out.println("Date Time "+datetime+"\t Room Name  "+roomname+


                            i++;
                    }
            }
```

```
else
{
        System.out.print("Problem In getting SingleRoomLightStatus(Drawing,ON)");
}

if(arrdrawingon.size()>1)
{
System.out.println("\n\n");
for(int j=0;j<sizeofarrdrawingon;j++)
{
        System.out.println("Date And Time of Drawing Light ON  "+datetimedrawingon[
}


System.out.println("\n\n");
int differentinlighton=0;
for(int j=0;j<(sizeofarrdrawingon-1);j++)
{


        String previousday=datetimedrawingon[j];
        String nextday=datetimedrawingon[j+1];

        System.out.println("Previous Day   "+previousday);
        System.out.println("Next Day  "+nextday);

                DateFormat formatter = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
                Date datePrevious = formatter.parse(previousday);
```

```
                    Date dateCurrent = formatter.parse(nextday);



                    long diff = dateCurrent.getTime() - datePrevious.getTime();
                    diff=diff/1000;
                    System.out.println("Date Difference in seconds  "+diff);
                    diff=diff/60;
                    System.out.println("Date Difference in minutes  "+diff);
                    diff=diff/60;
                    System.out.println("Date Difference in hrs  "+diff);
                    differentinlighton=(int) (differentinlighton+diff);

}
System.out.println("\n\n");
System.out.println("Overall Difference in Hrs   "+differentinlighton);

int avgdiff=differentinlighton/(sizeofarrdrawingon-1);
System.out.println("Average Difference in Hrs   "+avgdiff);



System.out.println("\n\n");
DateFormat formatter = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");

String lastdate=datetimedrawingon[sizeofarrdrawingon-1];

System.out.println("Last Date And Time   "+lastdate);

Date currdate = new Date();
```

```
 System.out.println("Current Date And Time  "+formatter.format(currdate));

 String currdatenadtime=formatter.format(currdate).toString();

 Date datelast = formatter.parse(lastdate);
Date currentdatetime=formatter.parse(currdatenadtime);



long diffnowandlast = currentdatetime.getTime() - datelast.getTime();
diffnowandlast=diffnowandlast/1000;
System.out.println("Date Difference in seconds  "+diffnowandlast);
diffnowandlast=diffnowandlast/60;
System.out.println("Date Difference in minutes  "+diffnowandlast);
diffnowandlast=diffnowandlast/60;
System.out.println("Date Difference in hrs  "+diffnowandlast);



System.out.println("\n\n");
if(diffnowandlast>avgdiff)
{
        System.out.println("\n\nPrediction : You should have ON the Light Of Drawing
        return "Prediction : You should have ON the Light Of Drawing Room...";
}
else if(diffnowandlast<avgdiff)
{
        System.out.println("\n\nPrediction : You have wait to ON the Light Of Drawin
        return "Prediction : You have wait to ON the Light Of Drawing Room...";
}
```

```
            else if(diffnowandlast==avgdiff)
            {
                    System.out.println("\n\nPrediction : On The Light of Drawing Room...\n\n");
                    return "Prediction : On The Light of Drawing Room...";
            }
            else
            {
                    System.out.println("\n\nPrediction : Can not predict to On The Light of Draw
                    return "Prediction : Can not predict to On The Light of Drawing Room...";
            }
             }

             return "Prediction : Can not predict to On The Light of Drawing Room...";
    }
    public static String getDrawingOFFPrediction() throws ParseException
    {
             List<Actions> arrdrawingoff = ActionsDAO.getSingleRoomLightStatus("Drawing","OFF");


             int sizeofarrdrawingoff=arrdrawingoff.size();

             String datetimedrawingoff[]=new String[sizeofarrdrawingoff];
             System.out.println("Drawing Room Light OFF List   "+arrdrawingoff);
             if(arrdrawingoff != null)
             {

                     int i=0;
                     while(i < arrdrawingoff.size())
                     {
```

```
                String datetime=arrdrawingoff.get(i).getDatetime();
                datetimedrawingoff[i]=datetime;
                String roomname=arrdrawingoff.get(i).getRoomname();
                String lightstatus=arrdrawingoff.get(i).getLightstatus();
                System.out.println("Date Time "+datetime+"\t Room Name  "+roomname+

                i++;
        }
}
else
{
        System.out.print("Problem In getting SingleRoomLightStatus(Dining,OFF)");
}
if(1< arrdrawingoff.size())
{
System.out.println("\n\n");
for(int j=0;j<sizeofarrdrawingoff;j++)
{
        System.out.println("Date And Time of Drawing Light OFF  "+datetimedrawingof
}


System.out.println("\n\n");
int differentinlightoff=0;
for(int j=0;j<(sizeofarrdrawingoff-1);j++)
{
```

```
            String previousday=datetimedrawingoff[j];
            String nextday=datetimedrawingoff[j+1];

            System.out.println("Previous Day   "+previousday);
            System.out.println("Next Day  "+nextday);

                DateFormat formatter = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
                Date datePrevious = formatter.parse(previousday);

                Date dateCurrent = formatter.parse(nextday);



                long diff = dateCurrent.getTime() - datePrevious.getTime();
                diff=diff/1000;
                System.out.println("Date Difference in seconds  "+diff);
                diff=diff/60;
                System.out.println("Date Difference in minutes  "+diff);
                diff=diff/60;
                System.out.println("Date Difference in hrs  "+diff);
                differentinlightoff=(int) (differentinlightoff+diff);

        }
        System.out.println("\n\n");
        System.out.println("Overall Difference in Hrs   "+differentinlightoff);

        int avgdiff=differentinlightoff/(sizeofarrdrawingoff-1);
        System.out.println("Average Difference in Hrs   "+avgdiff);
```

```
System.out.println("\n\n");
DateFormat formatter = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");

String lastdate=datetimedrawingoff[sizeofarrdrawingoff-1];

System.out.println("Last Date And Time   "+lastdate);

Date currdate = new Date();

System.out.println("Current Date And Time  "+formatter.format(currdate));

String currdatenadtime=formatter.format(currdate).toString();

Date datelast = formatter.parse(lastdate);
Date currentdatetime=formatter.parse(currdatenadtime);


long diffnowandlast = currentdatetime.getTime() - datelast.getTime();
diffnowandlast=diffnowandlast/1000;
System.out.println("Date Difference in seconds  "+diffnowandlast);
diffnowandlast=diffnowandlast/60;
System.out.println("Date Difference in minutes  "+diffnowandlast);
diffnowandlast=diffnowandlast/60;
System.out.println("Date Difference in hrs  "+diffnowandlast);


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

```
            if(diffnowandlast>avgdiff)
            {
                    System.out.println("\n\nPrediction : You should have OFF the Light Of Drawin
                    return "Prediction : You should have OFF the Light Of Drawing Room...";
            }
            else if(diffnowandlast<avgdiff)
            {
                    System.out.println("\n\nPrediction : You have wait to OFF the Light Of Drawi
                    return "Prediction : You have wait to OFF the Light Of Drawing Room...";
            }
            else if(diffnowandlast==avgdiff)
            {
                    System.out.println("\n\nPrediction : OFF The Light of Drawing Room...\n\n");
                    return "Prediction : OFF The Light of Drawing Room...";
            }
            else
            {
                    System.out.println("\n\nPrediction : Can not predict to OFF The Light of Dra
                    return "Prediction : Can not predict to OFF The Light of Drawing Room...";
            }

             }
             return "Prediction : Can not predict to OFF The Light of Drawing Room...";

    }
    public static void main(String[] args)
    {
            // TODO Auto-generated method stub
            try {
```

```
                Prediction.getDiningOnPrediction();
            } catch (ParseException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            try {
                Prediction.getDiningOFFPrediction();
            } catch (ParseException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            try {
                Prediction.getDrawingOnPrediction();
            } catch (ParseException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            try {
                Prediction.getDrawingOFFPrediction();
            } catch (ParseException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }

        }

}

//=========================================================
```

==========ThreadStart.java==========

```
package Test;

import java.text.ParseException;
import java.util.concurrent.TimeUnit;

public class ThreadStart extends Thread
{
        public void run()
        {
//                if(Thread.currentThread().getName().equals("DiningON"))
//                {
//                        System.out.println("Thread Dining ON is working");
//                        try {
//                                Prediction.getDiningOnPrediction();
//                        } catch (ParseException e) {
//                                // TODO Auto-generated catch block
//                                e.printStackTrace();
//                        }
//                }
//                if(Thread.currentThread().getName().equals("DiningOFF"))
//                {
//                        System.out.println("Thread Dining OFF is working");
//                        try {
//                                Prediction.getDiningOFFPrediction();
//                        } catch (ParseException e) {
```

```
//                              // TODO Auto-generated catch block
//                              e.printStackTrace();
//                      }
//              }
//              if(Thread.currentThread().getName().equals("DrawingON"))
//              {
//                       System.out.println("Thread Drawing ON is working");
//                       try {
//                              Prediction.getDrawingOnPrediction();
//                      } catch (ParseException e) {
//                              // TODO Auto-generated catch block
//                              e.printStackTrace();
//                      }
//              }
//              if(Thread.currentThread().getName().equals("DrawingOFF"))
//              {
//                       System.out.println("Thread Drawing OFF is working");
//                       try {
//                              Prediction.getDrawingOFFPrediction();
//                      } catch (ParseException e) {
//                              // TODO Auto-generated catch block
//                              e.printStackTrace();
//                      }
//              }


            try {
                    Prediction.getDiningOnPrediction();
            } catch (ParseException e) {
```

```
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }
                try {
                        Prediction.getDiningOFFPrediction();
                } catch (ParseException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }
                try {
                        Prediction.getDrawingOnPrediction();
                } catch (ParseException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }
                try {
                        Prediction.getDrawingOFFPrediction();
                } catch (ParseException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }


        }
        public static void startthreadoperations() throws InterruptedException
        {
                ThreadStart t1=new ThreadStart();
//                ThreadStart t2=new ThreadStart();
//                ThreadStart t3=new ThreadStart();
```

```
//                ThreadStart t4=new ThreadStart();
//                t1.setName("DiningON");
//                t2.setName("DiningOFF");
//                t3.setName("DrawingON");
//                t4.setName("DrawingOFF");
                while(true)
                {
                        t1.start();
                        //TimeUnit.SECONDS.sleep(10000);
                }
//                t2.start();
//                t3.start();
//                t4.start();

        }
        public static void main(String args[]) throws InterruptedException
        {
                ThreadStart.startthreadoperations();
        }

}

//=========================================================
```

4.4 Screenshots of Major Functionalities

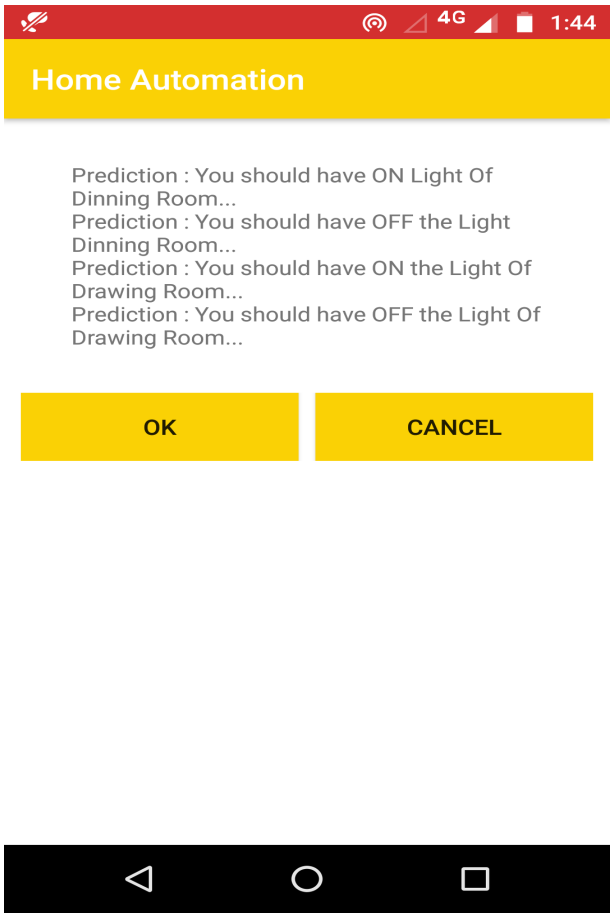

Figure 4.1: Prediction

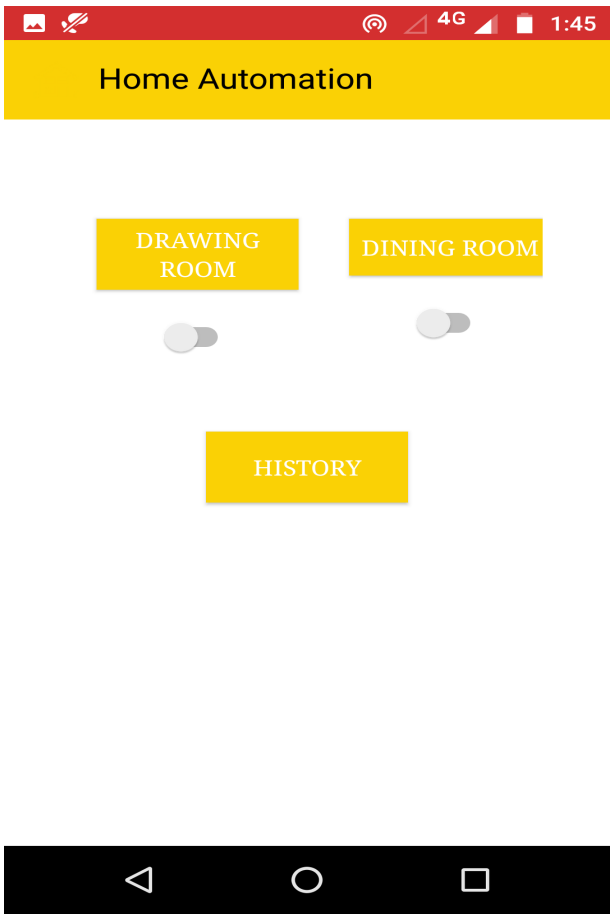

Figure 4.2: Room Selection

Figure 4.3: History

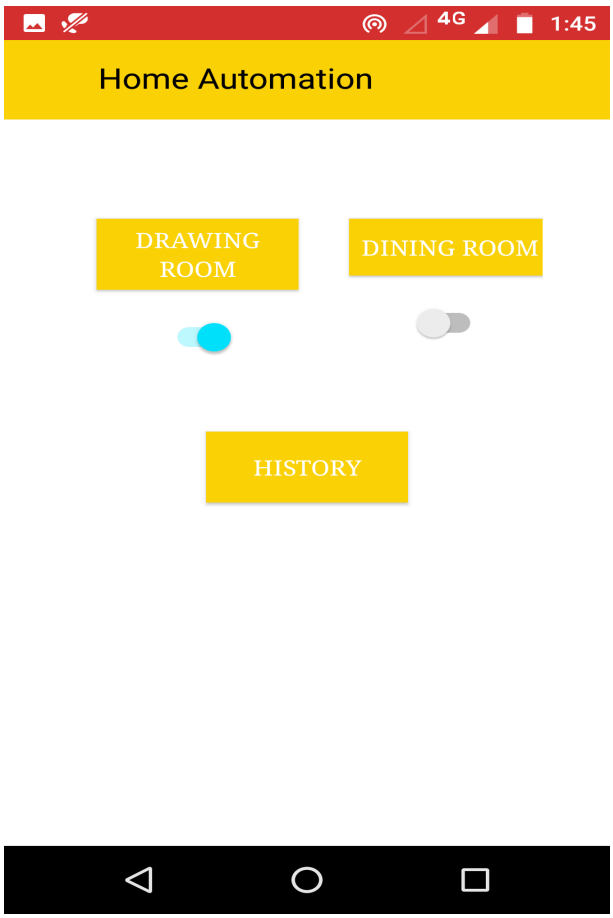

Figure 4.4: Drawing room prediction started

CHAPTER 5

TESTING

5.1 Unit Testing

This type of testing is performed by the developers before the setup is handed over to the testing team to formally execute the test cases. Unit testing is performed by the respective developers on the individual units of source code assigned areas. The developers use test data that is separate from the test data of the quality assurance team. The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.

Limitations of Unit Testing: Testing cannot catch each and every bug in an application. It is impossible to evaluate every execution path in every software application. The same is the case with unit testing. There is a limit to the number of scenarios and test data that the developer can use to verify the source code. So after he has exhausted all options there is no choice but to stop unit testing and merge the code segment with other units.

5.2 Integration Testing

The testing of combined parts of an application to determine if they function correctly together is Integration testing. There are two methods of doing Integration Testing Bottom-up Integration testing and Top Down Integration testing.

Bottom-up Integration Testing: This testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds.

Top-down Integration Testing: This testing, the highest-level modules are tested first and progressively lower-level modules are tested after that.

In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing. The process concludes with multiple tests of the complete application, preferably in scenarios designed to mimic those it will encounter in customers' computers, systems and network.

5.3 Acceptance Testing

This is arguably the most importance type of testing as it is conducted by the Quality Assurance Team who will gauge whether the application meets the intended specifications and satisfies the client's requirements. The QA team will have a set of pre written scenarios and Test Cases that will be used to test the application.

More ideas will be shared about the application and more tests can be performed on it to gauge its accuracy and the reasons why the project was initiated. Acceptance tests are not only intended to point out simple spelling mistakes, cosmetic errors or Interface gaps, but also to point out any bugs in the application that will result in system crashers or major errors in the application.

By performing acceptance tests on an application the testing team will deduce how the application will perform in production. There are also legal and contractual requirements for acceptance of the system.

CHAPTER 6

CONCLUSION

This proposed system presents the overall design of Home Automation System with low cost and wireless system. This system is designed to assist and provide support in order to fulfil the needs of elderly and disabled in home. In addition, the smart home concept in the system improves the standard of living at home. Focus of this system has been to control the household equipment's like light, fan, AC, etc. This System uses a sequential prediction algorithm, which observes sequence of events to predict the next event in smart environment. This prediction is useful in many scenarios for e.g., predicting inhabitant activities provides a basis for automating interaction with environment and improves the inhabitants comfort. This is achieved using Arduino board and android mobile application.

CHAPTER 7

REFERENCES

[1] Kumar Mandula, Ramu Parupalli, CH.A.S.Murty, E.Magesh, Rutul Lunagariya "Mobile based Horne Automation using Internet of Things(IoT), 2015,International Conference on Control,lnstrumentation, Communication and Computational Technologies (lCCICCT)

[2] Karthik Gopalratnam and Diane J. Cook, "Active LeZi: An Incremental Parsing Algorithm for Sequential Prediction ", 2012, FLAIRS

[3] Karthik Gopalratnam, Diane J. Coo, "Online Sequential Prediction via Incremental Parsing: The Active LeZi Algorithm ", 2007, IEEE

[4] https://en.wikipedia.orgiwikilInternecoCThings

[5] Mr. Pranay P. Gaikwad, Mrs. Jyotsna P. Gabhane, Mrs. Snehal S. Golait, "A Survey based on Smart Homes System Using Internet-of-Things ", 2015,International Conference on Computation of Power, Information and Communication,

[6] https://en.wikipedia.org/wiki/Home_automation