APPROVAL

This Project titled "**IoT Based Smart Home Automation**", submitted by **Md. Nazmul Islam** and **Sree Sushanto Chandro** to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 05 May 2018.

Board of Examiners

Chairman

Dr. Syed Akhter Hossain
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Dr. Sheak Rashed Haider Noori
Associate Professor and Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Md. Zahid Hasan
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

External Examiner

Dr. Mohammad Shorif Uddin
Professor
Department of Computer Science and Engineering
Jahangirnagar University

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Aniruddha Rakshit, Lecturer, Department of CSE, Daffodil International University**. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

Aniruddha Rakshit
Lecturer
Department of CSE
Daffodil International University

Submitted by:

Md. Nazmul Islam
ID: 152-15-6259
Department of CSE
Daffodil International University

Sree Sushanto Chandro
ID: 152-15-6273
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them.

We are highly indebted to Daffodil International University for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express our gratitude towards our supervisor **Aniruddha Rakshit, Lecturer, Daffodil International University** for his kind cooperation and encouragement which helped us in completion of this project.

Md. Nazmul Islam
&
Sree Sushanto Chandro

ABSTRACT

This report aims to describe the process and procedure to build a very effective home automation system titled as "Smart Home Control". The main objective of this system is to give a user the control of the power of his home appliances from the internet. The Smart Home project is a combination of Hardware and Software. Arduino has been used as the CPU of the Smart Home system which is configured to stay connected to the internet Arduino synchronizes the device statuses with the web server. There is a relay device for switching control. After the successful setup of both the hardware and software of the system, users will get access to their web interface. The software part consists a web server based on PHP. There is a responsive web interface that users have to use to setup and configure their devices which are connected to the Easy Home hardware interface. In the app GUI users will be able to manipulate their added devices. The app GUI is designed and developed to show the statuses of the connected devices in real time. There are Turn On and Turn Off buttons for each devices that users can use to control the power of their devices.

TABLE OF CONTENTS

PAGE

CONTENTS

CHAPTERS

CHAPTER 7: CONCLUSION AND FUTURE SCOPE

LIST OF FIGURE

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 General Introduction

In present time there is traditional home appliance control system and other remote control system people has been using. So the tradition system can't control home appliances remotely from long distance and there is other remote control system that has to be contain remote that is costly and destroyable. By using our project there is no need of remote just have to install an app in smartphone, in order to maintain its overall characteristic, there is an arrangement to provide reports of the home appliances and can be controlled remotely using android app.

Smart home control is a device that integrate with a web server and fetch the web data in real time and according this data the device will control home appliances. This project contains some sensors like motion sensor to detect the motion, sonar sensor to detect the distance, gas sensor to detect smoke and other gases and temperature sensor that is used to know home real time temperature.

1.2Motivation

- Nowadays everything is going to be depend on Internet.
- Internet of things (IoT) conceptualizes the idea of remotely connecting & monitoring real world objects through the Internet.
- This IoT project focuses on building a smart wireless home security system
- We can also utilized for home automation by making use of motion sensors.
- The status sent by the Wi-fi connected microcontroller managed system.
- Wi-Fi shield making use of which all the electrical appliances inside the home can be controlled & managed.

1.3 Objectives

Objectives of this projects are as follows:

a) Develop a home automation system using Arduino Uno, ESP-8266 NodeMcu and android application to control all home appliances.

b) Giving easy and flexible access to the devices from an Android application.

c) Develop smart feature Anti-theft mode that can get motion update using motion sensor.

d) Making the system secure for users.

e) Controlling fan speed and voltage of light from smart android app.

f) Power saving mode that can auto turn on/off appliances using motion sensor.

g) Used a gas sensor to detect gas leaking and fire in home.

h) Develop sleep mode that will auto off light when we get sleep and auto on when we wake up.

1.4 Overview

Home automation system [1] provides us an access to reach our desired devices to control it. There are few techniques and methods available that give people the opportunity to take this control. Our team has thought about implementing a new concept of home automation. I have an idea to gain control of home appliances that are based on a comparatively cheap embedded system “Arduino Uno and ESP”. This device gives us almost all the functionalities of a personal computer which runs on C based system. This embedded system connects all the hardware of our home automation system. It is connected to the internet. Our android application and web server which is based on secure Java, Android, C and PHP synchronizes device status with webserver in which there is a local server built with PHP and MYSQL

1.5 Expected Outcome

- We will implement a smart android apps and a device.
- This apps will provides security for a home and controlling by android phone.
- Smart security alert.
- The user cancontroll a house from anywhere connecting to the internet.
- We can save electric cost.
- User friendly and low cost and easy to access for any People.

CHAPTER 2

DESCRIPTION

2.1 Mobile Application

Smart home control operates via a mobile application designed for android OS. The owner of our Smart home control system can switch on and off electronic devices by a click of a button. The smart home control system knows its owner priority. Superior priority owner can decide who can be able to control which device in which room and can be monitored from the device application itself too. The android app will be password protected so that other intruders from outside can’t control devices. User id and passwords will match from webserver and gives access to specific users.

2.2 Hardware

Weare using Arduinouno and Esp-8266[2], 4 channel 5V Relay Module, voltage dimmer to control voltage of fan and light, Sensors(Motion, Sonar, Gas, Temperature) and an android powered phone to control our home. Arduinouno is an embedded chip consists of RAM, Video Controller, Power Supply Unit, CPU, GPU, and the ESP-8266 is the networking chip that can connect to internet through router etc. We are using a 5v12v~220v240v relay module to operate our home appliances. Relay module can protect devices from overvoltage and any other harm of electronic interferences.An android phone to operate as a remote controller.

2.3 Internet and Protocols

Smart home system operates via internet connectivity from a third party router as a hotspot or network traffic controller. Smart home system can provide a small range hotspot without internet to communicate with devices inside of its range. Devices can operate on local area network and world wide area network. Both devices can perform from local area network and world wide area network.

2.4 Security and Access

Smart home system can have two types of user. Admin user and Normal user. Admin user can access every device listed in the system and can configure device behavior if available. Admin user can distribute room access or device access to its normal user. A normal user can only access devices that are permitted by Administrator or Prime owner of the system.

CHAPTER 3

BACKGROUND STUDY

3.1 Related Work

The idea of a home automation has been an important issue in many publications and home appliances companies. Home automation is a house or living environment that contains the technology to allow devices and systems to be controlled automatically. Remote and local control is useful to keep home comfortable and to support the elderly and the disabled people.Bluetooth and remote controlled automation systems was popular. Many paper had been published in these topic. But In traditional system there has some limitation and problem. After revolution of IOT technology we have decided what if this automation can be implemented using IOT technology. In this paper, we discuss possible developments of IOT based home automation device instead of Bluetooth or remote control system.

Disadvantage of traditional home automation system:

- Remote and Bluetooth have range limit, appliances can be controlled within this range.
- Remote controller which can be damaged is extra costly (remote and battery cost).
- Special features like Anti-theft, gas and smoke detection can’t be possible within this short range.
- Device last status can’t be retrieved.

Benefit of IOT base home automation system:

- Distance is now doesn’t matter anymore.
- No need of remote controller to operate appliances.
- Special features can be implemented.
- Devices last status can be saved in web server.

3.2 Java Basic

Java is a famous programming language worldwide and it has the biggest library that can help us to gain our goal. There are almost 3 billion devices operates in java. Java methods and OOP are worldwide using popular technology now.

3.3 Why Java?

Java program runs on JVM so we can write our program once and run everywhere. Java has an excellent security and faster efficient memory management technique. We have used java in android programming and a backup programming language for our project. Besides java We used Kotlin as a support library for our project too.

3.4 Java in our project

Java has a wide range of use that's why We decided to use java in android programming. Soon We will enable our machine learning scripts programmed with java and python. Java maintains most of network calls and json parsing in our project. In android side Java giving full support as primary language in android for network call, json parse, database handler, threads, services and running various android system services.

3.5 Intro to Modified C

C is mother of all programming languages and has excellent performance on every platform. C can be used for hardware/machine control, web application, mobile platform and so on. It has a tough and excellent security with a wide range library. But we have just used C in our arduinouno and Esp-8266 device.

3.6 Why C

One of the very strong reasons why C programming language is so popular and used so widely is the flexibility of its use for memory management. Programmers have opportunities to control how, when, and where to allocate and deallocate memory. Memory is allocated statically, automatically, or dynamically in C programming with the help of malloc and callocfunctions.[3].

3.7 PHP in our project

PHP is one of the most popular languages for web development. Using this language we can easily send data to outside world through JSON format. This language has a lot of work to do, which is very important for web development.

3.8 Intro to Android

Android is famous OS used in mobile devices. Android OS developed by Google based on Linux kernel. Nowadays android used in several gadget like phone, TV, tablet, Fridge, Motor Mobil etc. Android is very popular for its open SDK that can solve various problem and can communicate serial data and parallel data in local network or in internet. It has great features include hardware control and logical control with database.

3.9 Why Android?

Android is very popular and almost everyone has Easy phone powered by android. So we have chosen android as a controller that a user can interact with our home automation system.

3.10 Android in our project

Android has great portability and internet communication feature. So it enabled us to write an android application so that user can control electronic home appliances through network from anywhere in the world.

3.11 Volley technique (Network)

We have used manual volley technique for android to make network call and data processing within android application. Volley technique has great network management and threading library and services that can transmit data from device to network in application background without halting any android services or activity.

3.12 Sqlite3 database for android

Sqlite3 is very powerful and lightweight database management for android application and Python application. I will use sqlite3 in android application and web application too. Sqlite3 can do various data management query and relation in an easier way. It has great customization feature. So I preferred to use sqlite3 as database for lower memory and fast performance.

CHAPTER 4

SYSTEM DESIGN

4.1 Software Design

Language and Utilities used to design and develop web application.

ID	**LANGUAGE, TOOLS & UTILITIES**	**PLATFORM**	**USAGE**
1	PHP	Web Server	Communication with central database
2	Java & Android	Mobile Application	Used in mobile application programming to control hardware via phone and access central database
3	SqLite 3	Database Management	Mobile Application
4	Arduino	IDE	Used C Programming
5	Android Studio	IDE	Used for Mobile Application Development

Figure 4.1: Language and Utilities application

4.1.1Data flow Diagram of Smart Home Control

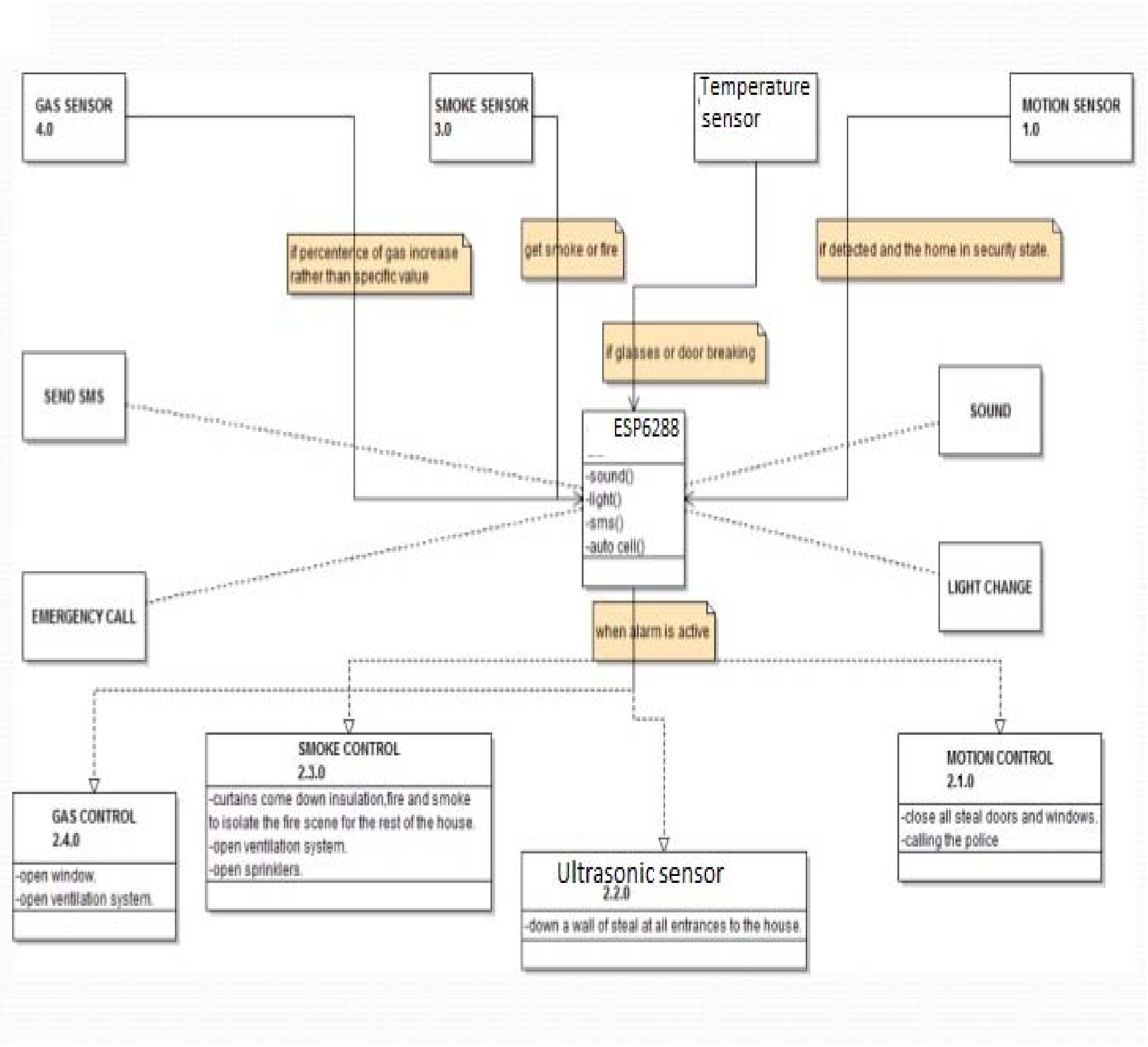

GAS SENSOR
4.0
SMOKE SENSOR
3.0
Temperature
sensor
MOTION SENSOR
1.0
if percentence of gas increase
rather than specific value
get smoke or fire
if detected and the home in security state.
if glasses or door breaking
SEND SMS
SOUND
ESP6288
-sound()
-light()
-sms()
-auto cell()
EMERGENCY CALL
LIGHT CHANGE
when alarm is active
SMOKE CONTROL
2.3.0
-curtains come down insulation,fire and smoke
to isolate the fire scene for the rest of the house.
-open ventilation system.
-open sprinklers.
MOTION CONTROL
2.1.0
-close all steal doors and windows.
-calling the police
GAS CONTROL
2.4.0
-open window.
-open ventilation system.
Ultrasonic sensor
2.2.0
-down a wall of steal at all entrances to the house.

Figure 4.1.1: Diagram of Smart Home Control

The home automation system is divided into 3 main section. They are listed below.

4.1.2Central Database & Control from Android Application.

All user authentication system and registration should be processed through a web application / Android application that will manage and control the central database of our system. Central database is designed in MYSQL so we have used PHP as our background web applications primary language. Central database relation are given in a flowchart.

We are passing an array of objects through JSON to ESP-8266 and arduino Uno and Android application. Arduino Uno, ESP-8266 and Android application gets the JSON data from central database. Central Database holds several tables like “IoT Device”, “User Control”, “Product registration and activation”, “Device control and status”.

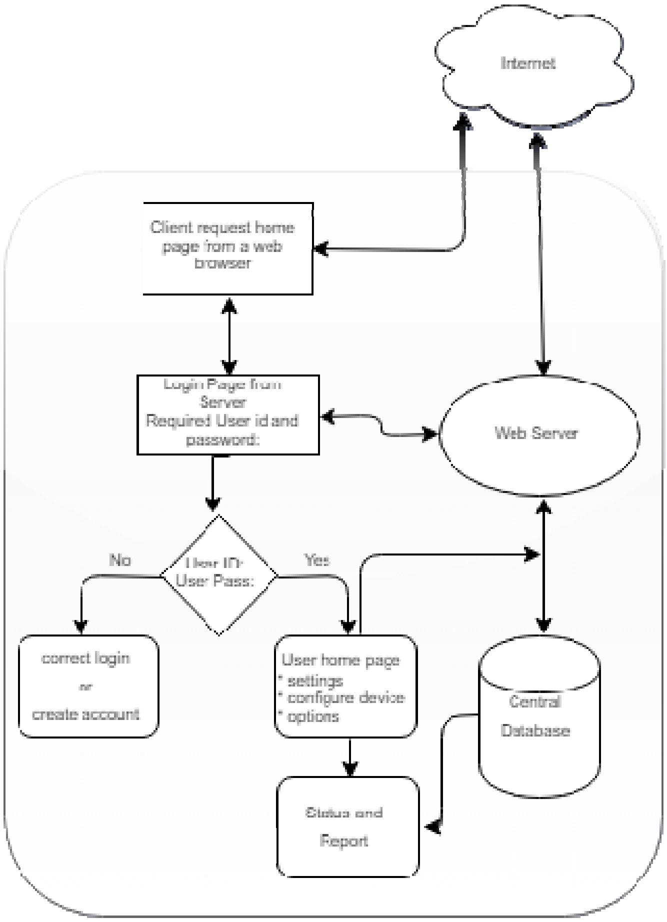

Internet
Client request home page from a web browser
Login Page from Server Required User id and password:
Web Server
No
User ID: User Pass:
Yes
correct login or create account
User home page
* settings
* configure device
* options
Central Database
Status and Report

Figure 4.1.2: Smart Home Communication in Web

4.1.3 Arduino Uno and Esp-8266

Arduinouno and Esp-8266 controls electronics home appliances by a 4 channel relay module connected to Arduino Uno GPIO pins. ESP-8266 requests web server for the JSON data as input from central database and sends to arduinouno to control the GPIO pins. A C program runs on a loop which can be stopped or restarted from a PHP web server. C program analyzes the JSON array and controls the defined pins to on and off a specific Relay from Relay Module. User can change pin status through websites or from android application. C program controls other properties of the system. Some feature is automatically controlled by C. C program is able to detect if it is connected to the internet or not. If a connection is successfully established to internet, C program switches to internet protocol call for data listening process.

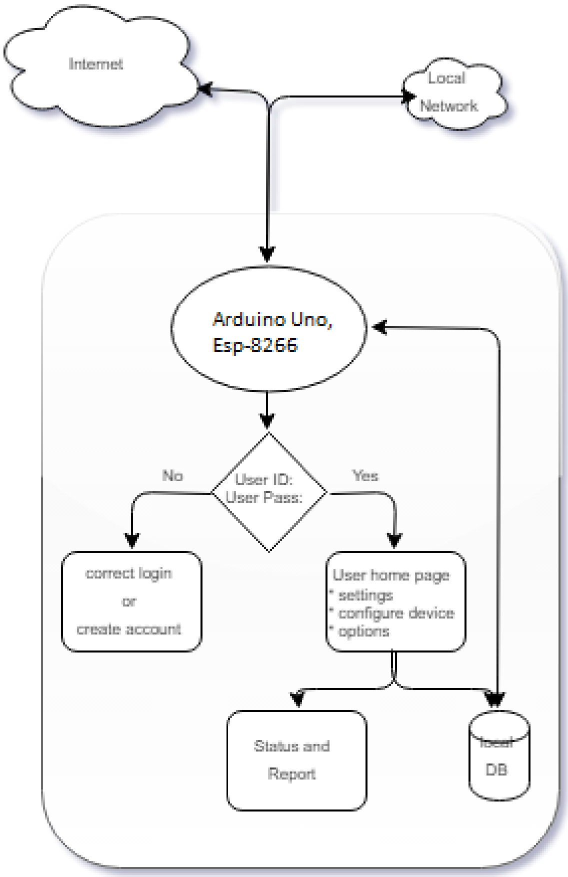


Figure 4.1.3: Smart Home Communication in Arduinouno

4.1.4 Android Application

Android application has limited functionality than web server as well as some unique functionality. User can access its home appliances directly through the application itself. User have to login by his/her credentials through the android application. User can register an account from the application too. User can list his/her Smart home Control system if they have multiple Smart home system implemented to their region. User have to select which room they want to control. After selecting specific room, android application will take the user to pin control window. User can change pin status from pin control window. Android application has its own activity and fragment lifecycle. So user can interact with the application much more easily and faster way. Android application records the changing pin status in a Sqlite3 database. After that a thread service queries the changed data and posts to central database. We have used volley technique for network call and data processing in Android. User can control pins without central database if he/she is in local network. Android application shows pop up if user not connected to the internet. In that case user have to find his/her ESP-8266 IP address and mark out if it is in same local network.

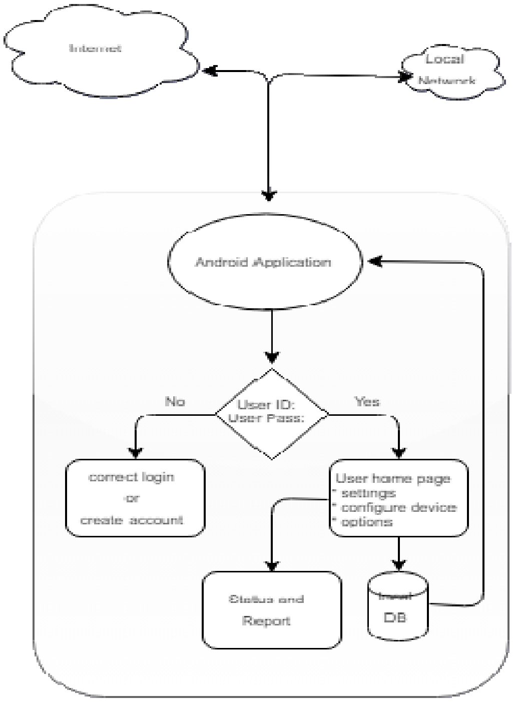
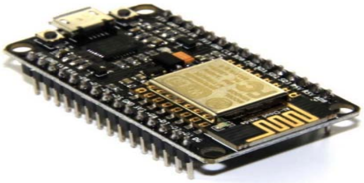


Figure 4.1.4: Data Flow Diagram of Android and Central Database

4.2 Hardware Design

Following electronic devices and components were used to build our hardware system.

	Name	Image	Use
1	Arduino Uno		Main control board
2	ESP-8266 NodeMcu		Networking board
3	Relay 5V Dc		Switching power to higher voltage devices
4	Resistor 150/270/1000/10000 Ohm		VA controller
6	Programmable Voltage dimmer		Voltage control
8	LED		Power light

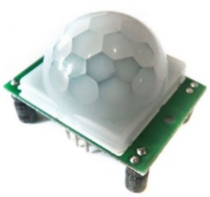
9	Temperature Sensor		Calculate temperature
10	Motion sensor		Detect Motion
11	Sonar sensor		Calculate distance
12	Gas sensor		Detect gas and smoke

Figure 4.2: Electronic devices and components were used to build our hardware system.

CHAPTER 5

IMPLEMENTATION

5.1 Making Json of requested data of specific user using PHP

```
importurllib, json, pprint
importrequests
fromcollectionsimportnamedtupl
e
fromarrayimport array
importRPi.GPIOasGPIO

defcontrolOperation(pin,
status):
print pin, status
# print type(pin) ,
type(status)
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
if status in ['HIGH']:
print ('light on')
GPIO.setup(pin, GPIO.OUT)
GPIO.output(pin, GPIO.HIGH)
print"light on "
else:
print ('light off')
GPIO.setup(pin, GPIO.OUT)
GPIO.output(pin, GPIO.LOW)
print"light off "

defscanAllData():
url =
"http://rpi.somee.com/Home/Res
ponce?userName=murad"
    r = requests.get(url)
```

```
jsonData = r.content
 x = json.loads(jsonData,
object_hook=lambda d:
namedtuple('X', d.keys())(
        *d.values()))
dataArray = x.pin
# print type (dataArray) #--what
type of variable it is?--
# global variable declaration
pinArray = []
statusArray = []
foriinrange(len(dataArray)):
        pin = dataArray[i].pin
        status =
dataArray[i].status

# ------ adding value in an array
------
pinArray.append(pin)
statusArray.append(status)

# ------ calling a method ------
controlOperation(pin, status)
# print pinArray
# print statusArray

for j inrange(2400):
scanAllData()
print j
```

5.2 Implement code of Java

```
private void postDataToServer(String swStatus, intid) {
  Retrofit retrofit = new Retrofit.Builder()
          .baseUrl(ConstantValue.SERVER1_POST_BASE_URL)
          .addConverterFactory(GsonConverterFactory.create())
          .build();
  Call<String>pinControlDataCall;
  PostControlDataAPIpostApi =
retrofit.create(PostControlDataAPI.class);
  String userName = "username";
  String password = "123456";
  String postURL = "";
  if (swStatus.toLowerCase().equals("true")) {
      postURL = "userpost?username=" + userName + "&password=" +
password + "&piPinId=" + id + "&status=" + swStatus;
  } else {
      postURL = "userpost?username=" + userName + "&password=" +
password + "&piPinId=" + id + "&status=" + swStatus;
  }
  pinControlDataCall = postApi.postViaURL(postURL);
  pinControlDataCall.enqueue(new Callback<String>() {
      @Override
      public void onFailure(Call<String> call, Throwable t) {
           Toast.makeText(this,”Transmission
failed”,Length.SHORT).Show;
      }
  });
}
```

5.3 Implementation of code and libraries

```
dependencies {
  compile fileTree(dir: 'libs', include: ['*.jar'])
  androidTestCompile('com.android.support.test.espresso:espresso-
core:2.2.2', {
      exclude group: 'com.android.support', module: 'support-
annotations'   })
  compile'com.android.support:appcompat-v7:26.+'
  compile'com.android.support:design:26.+'
  compile'com.android.support:recyclerview-v7:26.+'
  compile'com.squareup.retrofit2:retrofit:2.1.+'
  compile 'com.android.support.constraint:constraint-
layout:1.0.0-alpha7'
  compile'com.android.support:support-v4:26.+'
  compile'com.squareup.retrofit2:converter-gson:2.1.+'
}
```

5.4 Hardware Implementation

Hardware implementation requires an Arduino Uno and Esp-8266, Relay, Transistor 2N3904, Resistor, Sensor Module, Internet connectivity. We have used the electronic devices and components (figure :) to build the modules. Built hardware modules are shown below.

5.4.1Arduino uno and Esp-8266 Setup

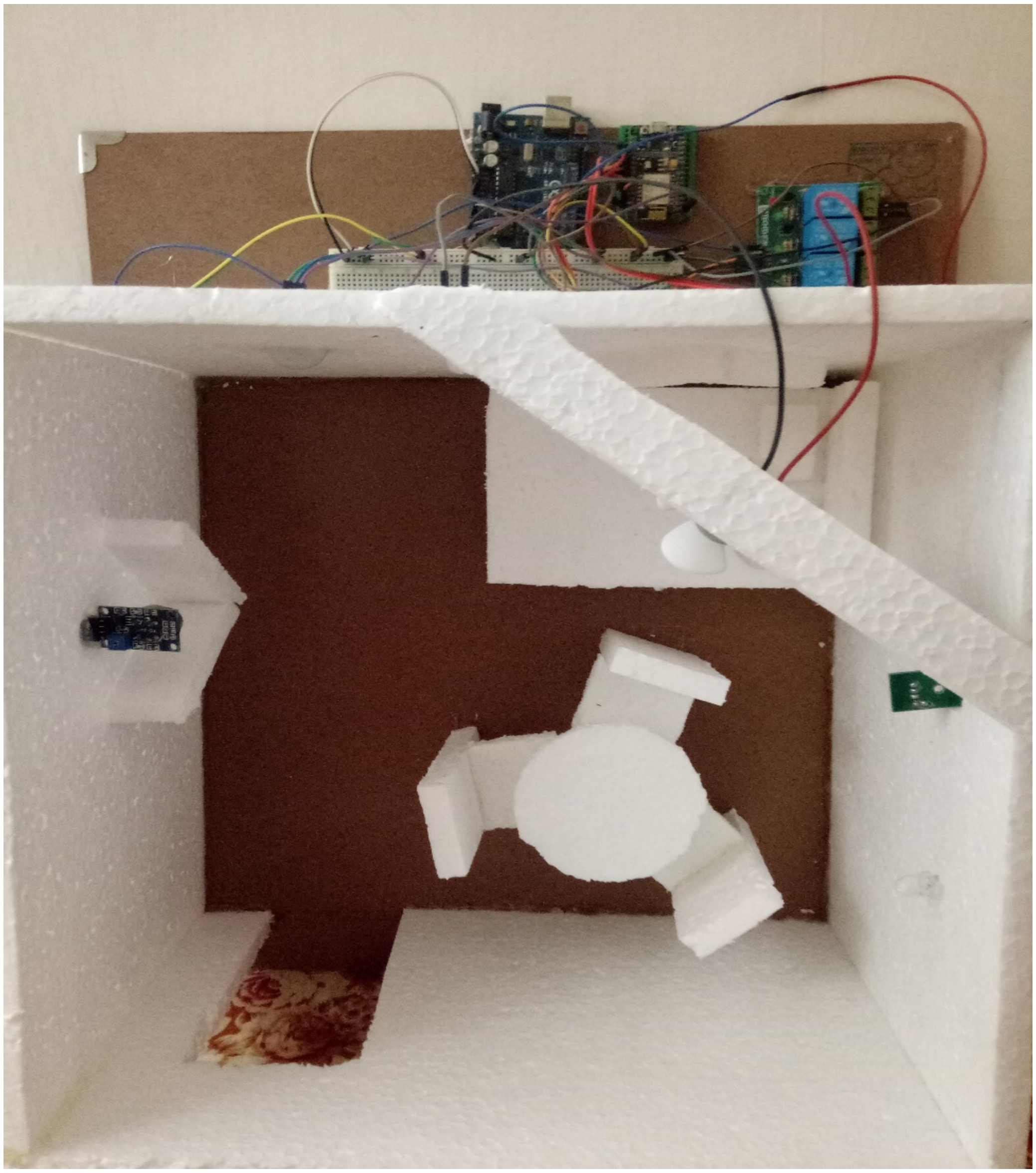


Figure 5.4.1: Smart home Setup

5.4.2 Relay Module

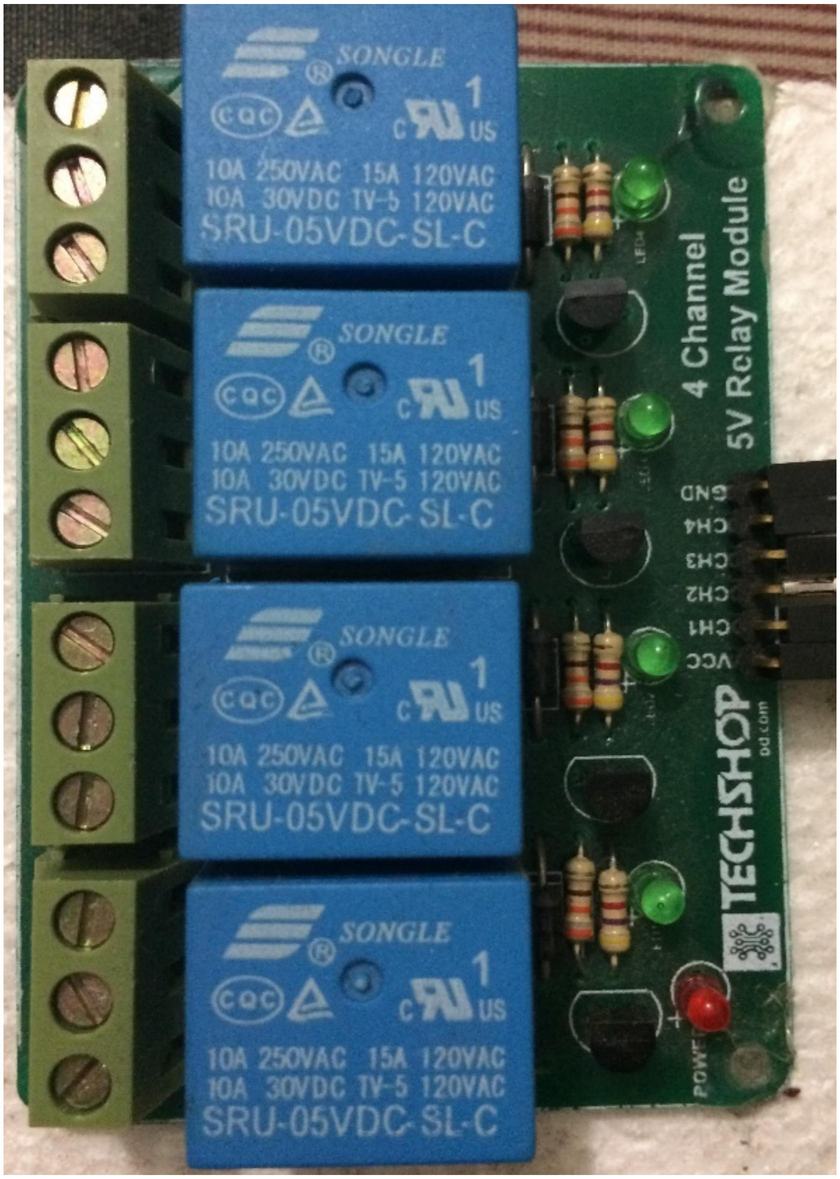


Figure 5.4.2: Relay Module

CHAPTER 6

EXPERIMENTAL RESULTS

6.1 Smart Home Control User Interface (Registration)

During the registration for a new user, the user must provide the MAC address of the ESP-8266 NodeMcu that has been used to build his/system along with the password given by the administrator. After the successful registration the users can login to their dashboard in our website.

6.2 Smart Home Control User Interface (Device Configuration)

After logging in to account user can configure connected devices according to the Pin numbers they are attached to. User has the options to name the devices with their locations to recognize in the control page easily.

6.3 Smart Home User Interface (Control Device)

In the Control Devices tab, configured rooms are listed for each system. User can see the list of devices by clicking the room names and control them.

6.4 Intro to Smart Home Android Application

User has to login first. After successful login operation the Android application will sync its current status with web database and hardware interface through internet.Intro of different activities of android app given below.

6.4.1 Android App Splash Screen

Figure 6.4.1: Splash Screen

6.4.2 Login Fragment

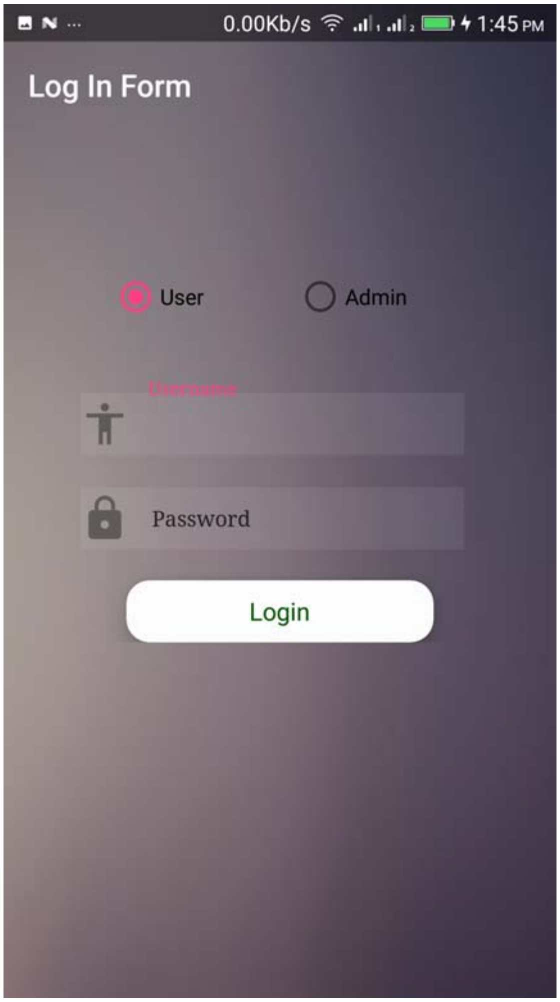
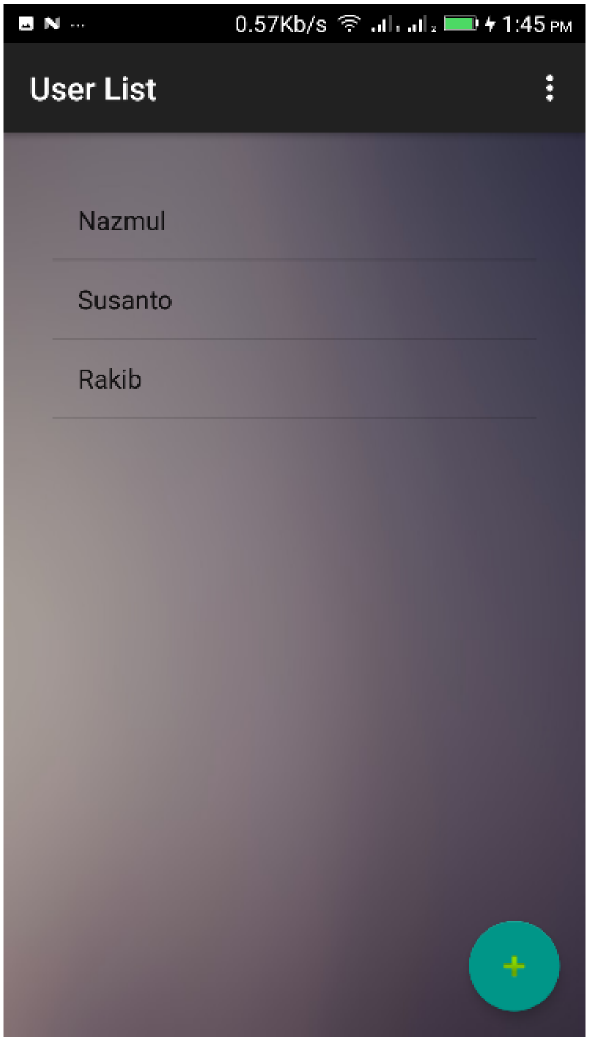


Figure 6.4.2: Login Fragment

6.4.3 Admin Panel



User List

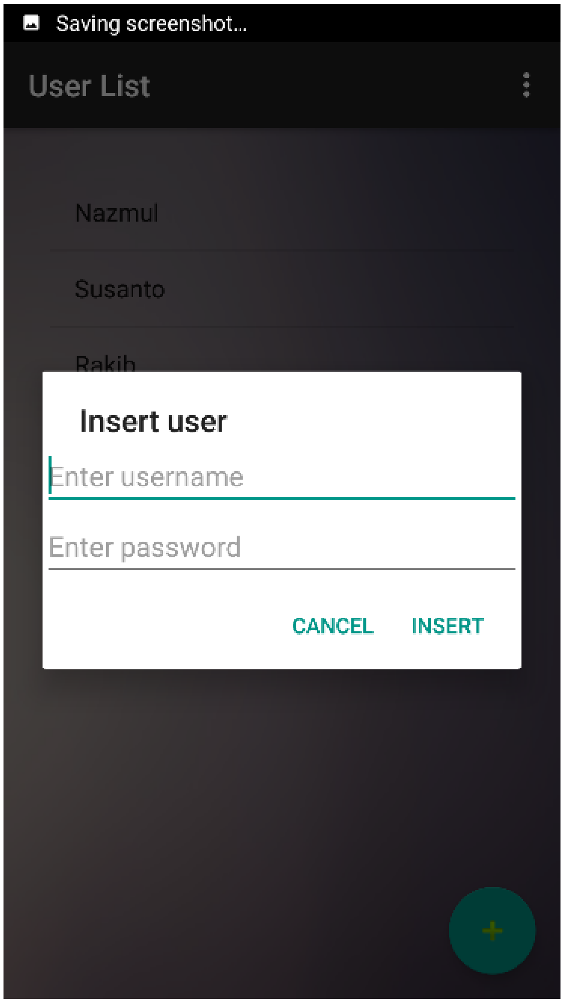
Nazmul

Susanto

Rakib

Figure 6.4.3: Admin Panel

6.4.4 User Insertion Panel



Saving screenshot...

Figure 6.4.4: User Insertion Panel

6.4.5 User Control Panel

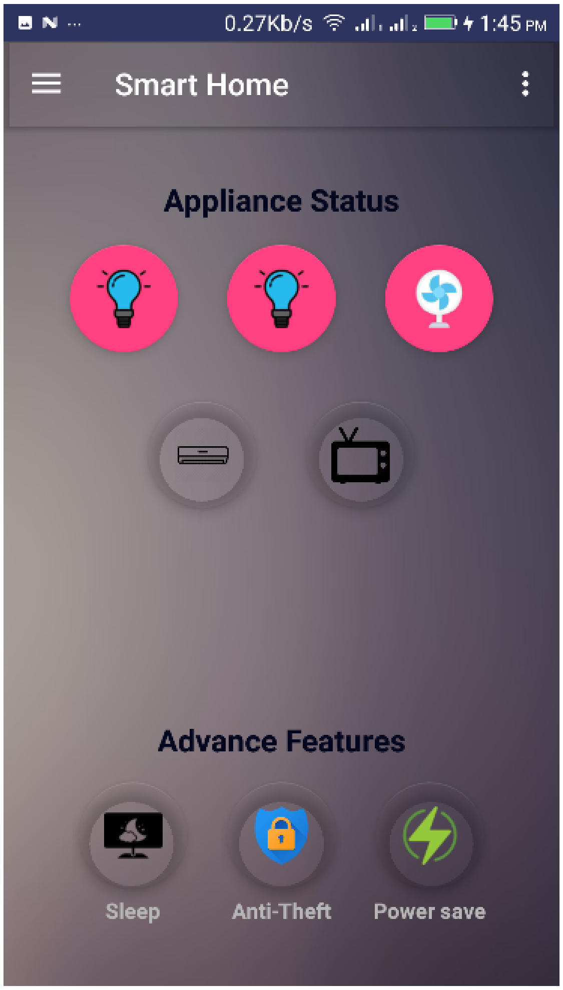


Figure 6.4.5: User Control Panel

6.4.6 Fan Speed Control

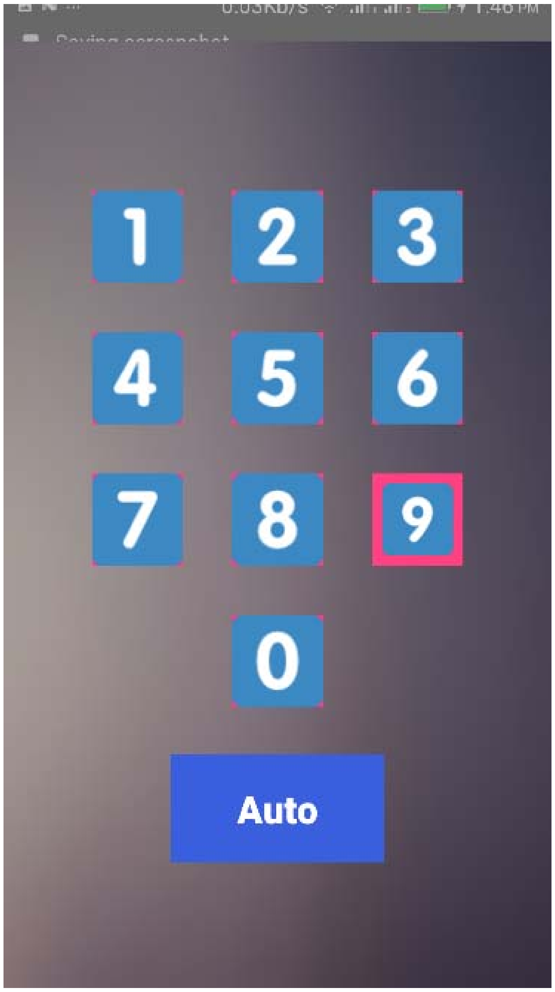


Figure 6.4.6: Fan Speed control

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Discussion and Conclusion

The project has proposed the idea of smart homes that can support a lot of home automation systems. A smart home contains a connection between wireless communication, sensors, monitoring and tracking. Smart homes are a huge system that includes multiple technologies and applications that can be used to provide security and control of the home easily.[5]

There is various home automation system with audio, video, gesture control, sensor controlled and lot more facilities can be made from locally available parts. There is also UART based procedure and some are powered by sophisticated microcontroller and single on board computers like Arduino, Esp-8266, raspberry pi, orange pi etc. Using home automation system the applications are so many like turn on lights, fans, TV, AC, and some are automated without any human interaction like garden lights, windows shades, vacuum cleaner, Air Conditioner, water tank & water pump. In future almost every home appliances will be smarter and automated. Smart home system can get sensor data and able to take action based on various calculation of sensor data. Easy home system hardware can be extended with several electronic components like shift register for more appliances, microcontroller for more sensor connection, wireless module so that devices can be easily moveable. The application of smart home idea is limitless.

7.2 Limitations & Future Scope

Till date home automation has been looked upon as something to only on/off your devices from mobile phone, but the scenario is changing rapidly, regular automation combined with smart wireless network of sensors started to understand the human environment. Sensors like motions sensor, temperature sensors, light intensity sensors, smoke detectors, humidity sensors etc. are being accommodated with home automation and the data from these sensors would further be used to analyzed to understand use patterns and in-turn with deep learning used for the intelligence of the device itself.

It will start understanding its user and keep making sure the user is at his most ease and security. [6]Predicting the future of just about anything is very risky business. Home automation is an industry that largely started with X10 devices in 1980. Today, we believe the future of home automation will very much ride the digital age and develop along with the computer and networking systems in the years to come. Initially it appeared, companies such as Microsoft and Exceptional Innovation with their Life/ware software were positioning the Windows Media Center PC as the heart and soul of a complete solution for home automation relying on web services to seamlessly interface with lighting controls, climate controls, security panels, and IP surveillance cameras to compliment the digital media management capabilities of Windows Media Center. However, Exceptional Innovation stopped selling their systems for residential installations, and Microsoft Media Center capabilities have disappeared in the Windows 8 OS. [7]

Today the "Internet of Things" has emerged the way.

Every smart home based system has to pass on some security thread that costs loss of property and digital privacy. As like every security has vulnerability, this smart home system has low security when thread is physical. Every Linux machine can be braked when attack happens in local network inside of firewall blindness. Sensor data has low precision and that can be solved by using expensive sensors. The system is central controller based so every device has to be connected to a centralized point or they can be wireless by using wireless modules and pairing with smart home local network.

REFERENCES

[1] WIKIPEDIA [Online], Home Automation System. Available: <<https://en.wikipedia.org/wiki/Web_application/>>, Last access on12-03-2018 at 10.00pm.

[2] Hernando Barragan [Online]: The Arduino history Available:<<http://www.circuitstoday.com/story-and-history-of-development-of-arduino/>>Last access on 12-03-2018 at 10.05pm.

[3]Learn about C programming available at: <<http://cs-fundamentals.com/tech-interview/c/why-we-use-c-programming-language.php/>>Last access on 12-03-2018 at 10.20pm.

[4]Learn about arduinoIDE available at:<<https://programmingelectronics.com/tutorial-3-arduino-ide-and-sketch-overview/>>Last asses on 13-03-2018 at 1.00pm.

[5]Learn about Conclusion and overview available at:<<http://smarthomes-technology.blogspot.com/p/conclusion-and-additional-work.html />>Last asses on 16-03-2018 at 9.00pm.

[6] Learn about future scope available at: <<https://www.quora.com/What-is-the-future-scope-of-home-automation/>>Last asses on 01-04-2018 at 12.00pm.

[7]Learn about of dataflow diagram available at:http://www.homeautomationinfo.com/Drupal/homeautomation_futurehttps://www.google.com.bd/>>Last asses on 02-04-2018 at 1.00pm.

[8]Motion sensor device study guide available at :<<https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor?view=all Last asses on 03-04-2018 at 1.00am.

[9]Arduino microcontroller study guide available at:<http://www.me.umn.edu/courses/me2011/arduino/arduinoGuide.pdf/>>Last asses on 03-04-2018 at 2.00am

[10]Home automation related work available at :<< https://en.wikipedia.org/>>Last asses on 25-04-2018 at 9.30pm.

[11] Esp8266 arduinowifi study guide available at :<https://www.geekstips.com/esp8266-arduino-tutorial-iot-code-example/>>Last asses on 26-04-2018 at 10.30pm.

[12] Guide for Relay Module with Arduinostudy guide available at:<https://randomnerdtutorials.com/guide-for-relay-module-with-arduino/>>Last asses on 27-04-2018 at 11.30pm.

APPENDIX

```
  "packages": [
    {
      "maintainer": "ESP8266 Community",
      "help": {
        "online": "http://esp8266.com/arduino"
      },
      "websiteURL": "https://github.com/esp8266/Arduino",
      "platforms": [
        {
          "category": "ESP8266",
          "help": {
            "online":
"http://esp8266.github.io/Arduino/versions/2.0.0/"
          },
          "url":
"https://github.com/esp8266/Arduino/releases/download/2.0.0/esp8266-
2.0.0.zip",
          "checksum": "SHA-
256:00911ee9f5d4802b279fd81a5a8177fd6fef4888d04ebe93fc3952bac85d909c",
          "version": "2.0.0",
          "architecture": "esp8266",
          "archiveFileName": "esp8266-2.0.0.zip",
          "boards": [
            {
              "name": "Generic ESP8266 Module"
            },
            {
              "name": "Olimex MOD-WIFI-ESP8266(-DEV)"
            },
            {
              "name": "NodeMCU 0.9 (ESP-12 Module)"
            },
            {
              "name": "NodeMCU 1.0 (ESP-12E Module)"
            },
            {
              "name": "Adafruit HUZZAH ESP8266 (ESP-12)"
            },
            {
              "name": "SparkFun Thing"
            },
            {
              "name": "SweetPea ESP-210"
            },
            {
              "name": "WeMos D1"
            },
            {
              "name": "WeMos D1 mini"
            }
          ],
          "size": "5606497",
          "toolsDependencies": [
            {
              "packager": "esp8266",
              "version": "0.4.6",
              "name": "esptool"
            },
            {
              "packager": "esp8266",
              "version": "1.20.0-26-gb404fb9-2",
              "name": "xtensa-lx106-elf-gcc"
            },
            {
              "packager": "esp8266",
              "version": "0.1.2",
              "name": "mkspiffs"
            }
          ],
          "name": "esp8266"
        },
        {
          "category": "ESP8266",
          "help": {
            "online":
"http://esp8266.github.com/Arduino/versions/2.3.0/"
          },
          "url":
"https://github.com/esp8266/Arduino/releases/download/2.3.0/esp8266-
2.3.0.zip",
          "checksum": "SHA-
256:884188a54bc75f1f2e7f827619f8cb1980dc3f430f92f9b07186ed7f4db8c738",
          "name": "esp8266",
          "version": "2.3.0",
          "architecture": "esp8266",
          "archiveFileName": "esp8266-2.3.0.zip",
          "size": "8595050",
          "toolsDependencies": [
            {
              "packager": "esp8266",
              "version": "0.4.9",
              "name": "esptool"
            },
            {
              "packager": "esp8266",
              "version": "1.20.0-26-gb404fb9-2",
              "name": "xtensa-lx106-elf-gcc"
            },
            {
              "packager": "esp8266",
              "version": "0.1.2",
              "name": "mkspiffs"
            }
          ],
```