Otto von Guericke University Magdeburg

Faculty of Electrical Engineering and Information Technology

Chair of Integration Automation

Digital Engineering Project

Distributed Home Automation based on IoT Sensor Network

Submitted: 09.03.2018

By: Arun Teja Sriramula (209601)

Dinesh Kumar Kadari (209778)

Naveen Chand Madishetti (209790)

Examiner: Prof. Dr.-Ing. Christian Diedrich

Advisor: Dipl.-Ing. Sasanka Potluri

Distributed Home Automation based on IoT Sensor network

Abstract

The advent of Internet of Things, made Internet to get deep rooted into every technological innovation to be smart, intelligent, and easy accessibility. Likewise are the home automation devices. Conventional home automation is based on centralised control. Such systems suffer portability, flexibility and extensibility. To overcome the drawbacks, a distributed control in home automation with IoT is proposed. The proposal is visualized by means of client server architecture using LAMP server setup on a raspberry pi, an android app and arduino, NodeMCUs as clients. MFRC522 RFID reader, keypad, DHT-11 sensor, are at the physical level. The proposal is implemented for monitoring and controlling devices with a client-server architecture in the home and also maintain data logs for future needs.

Contents

List of Figures

List of Abbreviations

IoT	Internet of Things
LAMP	Linux, Apache, Mysql and PHP
RFID	Radio Frequency Identification
MIT	Massachusetts Institute of Technology
GSM	Global System for Mobile
WiFi	Wireless Fidelity
ISP	Internet Service Provider
OTA	Over The Air
LED	Light emitting diode
USB	Universal Serial Bus
IDE	Integrated development environment
ICSP	In Circuit Serial Programming
PWM	Pulse width modulation
FTDI	Future Technology Devices International
GPIO	General-purpose input/output
ADC	Analog to Digital Converter
UART	Universal asynchronous receiver-transmitter
TCP	Transmission Control Protocol/Internet Protocol
UDP	User Datagram Protocol
HTTP	Hyper Text Transfer Protocol Secure
FTP	File Transfer Protocol
WPA	Wireless Protected Access
WEP	Wired Equivalent Privacy
TKIP	Temporal Key Integrity Protocol
PIR	Passive infrared sensor
API	Application Program Interface
SSL	Secure Sockets Layer
OOP	Object-Oriented Programming
GUI	Graphical User Interface
JSON	JavaScript Object Notation
UML	Unified Modelling Language
UID	Unique Id

1. Introduction

Internet of things (IoT) means enabling network connectivity between various physical devices with embedded electronics such as sensors, actuators, microcontrollers and software environment for easy information exchange and control using Internet. The term internet of things is coined by Kevin Ashton of Proctor & Gamble, later MIT's Auto ID centre in 1999 [1].

With existing internet speed, the intervention of IoT with physical systems has opened new windows for research in cyber-physical systems. Data exchange with IoT architecture improves efficiency, accessibility, management in the technological fields such as smart grids, industrial automation systems, smart cities, intelligent transportation. Research and development which was, once possibly be made in static research institutions and universities, there IoT associated with augmented and virtual reality, is taking a new shape so as to decrease the wastage of resources there by increasing the availability, accessibility, and assessment.

IoT has extended its branches to smart city solutions, industrial automation, smart Home etc., IoT is enabling advanced data analytics easier. Big data is growing enormously with millions of embedded devices getting connected to internet. Artificial intelligence is quicker than ever before gained through IoT.

2. Background

The concept of home automation is not new to technology world [2]. Early home automation system were based on Radio frequency communication. Later with evolution of mobiles, embedded systems gaining boom in automating daily activities, home automation become smart to be later known as smart homes. XBee, GSM, Bluetooth, Zwave, Wi-Fi based communication simplified home automation technologies provides efficiency, security and control.

XBee, Z-Wave are based on mesh topologies. Both the above mentioned technologies are based on proprietary protocol. The data rates are low. Operating distance range from 10m-100m. GSM based home automation work based on AT commands sent from a mobile device to the controller via SMS. Home automation systems with GSM control lack graphical user interface (GUI) and speed of operation. Sometimes errors in GSM base station due to adverse weather conditions cause unexpected delays, resulting havocs in extreme situations [3]. Bluetooth technologies operates at 2.4 GHz range, but constrained due operating distance.

Bluetooth establishes adhoc, peer-to-peer network for communication making it less efficient in large sensor networks.

With IoT, Smart Home technologies improved the connectivity between devices and the users. Improvement in terms of security, privacy, faster communication despite being at a remote location is achieved. People can perform simple activities like switching, monitoring and controlling with single touch on their smart mobiles or their laptop screen. Despite having

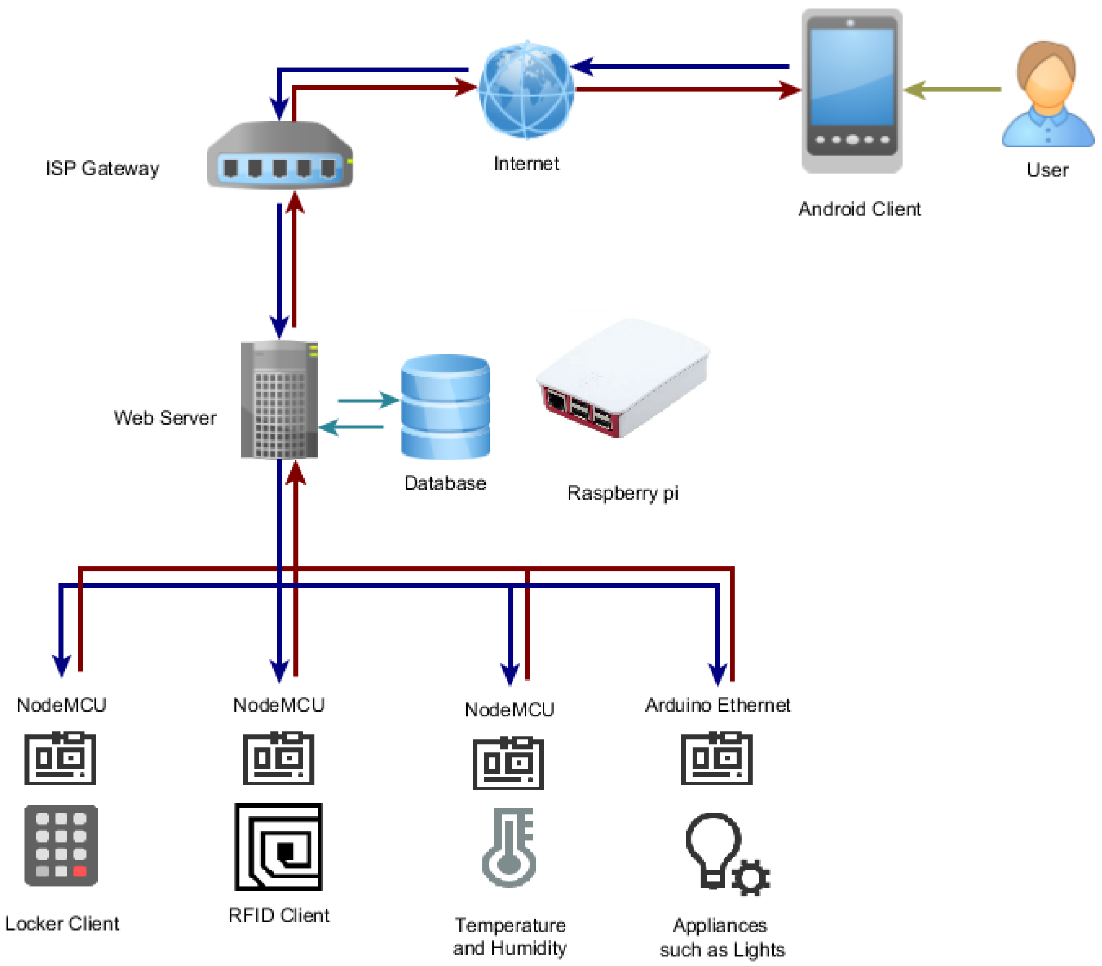

Fig. 1 Architectural Diagram of Distributed Home Automation system

many commercial smart home devices, their availability for public is being limited by portability, scalability, price and reach.

Contemporary home automation systems are based on centralised controller. There are significant drawbacks in centralised controller based smart homes or home automation. Centralised controller systems are not safe considering the growing IoT infrastructure. They are vulnerable for security and hacking attacks, where whole system fails. Also they are not portable, once setup, and costs money, effort for changing. Additional devices cannot be installed to the existing system being proprietary. A novel solution is necessary for the future demands.

Distributed home automation system is a viable solution [4]. The architectural diagram depicts the concept to be implemented. Raspberry pi acts as server and also maintains copy of the data in the main webserver. This gives an advantage that even if the main server fails, data logs is retained and control instructions can be routed via local sever. Also the clients function independently. The property that NodeMCUs can be programmed over the air (OTA) is great advantage for in house maintenance for scalability and portability.

Through our project, we tried to build a distributed control home automation system is using arduinos, raspberry pi and few sensors for measuring temperature, humidity; RFID door access management; to control lights, to monitor appliances. Also we developed an Android app to get all the information directly to your mobile and to manage the control devices. Hardware components involved as follows.

1) NodeMCU's -3 2) Raspberry pi 3b 3) RFID MFRC-522 reader

4) Keypad 4x4 5) LED's-4 6) Buzzer

3. Hardware Description

3.1 Arduino

Arduino is an easy to use micro controller with an open source platform [5]. It consists of hardware unit which can be programmed with a piece of software. Arduino is very popular device with its basic programming to provide many different libraries for a complex micro controller experiments. Its open source platform allows students and researchers to perform several electronic experiments. This device can be preassembled or self-preparing kits based on requirement of user. There are different types of boards with different configurations. Every device consists of set of analog and digital pins used for input and output connections. For serial communications USB ports and different interfaces are given on board. C and C++ are typically used to program this microcontroller and interfaces on board help in loading different experimental programs into the device from computer. In addition, it provides an integrated development environment (IDE) depending on project processing making life easier for non-programmers.

Here we use an arduino UNO micro controller of ATmega328 type datasheet. It consists of 14 digital input/output pins (in which 6 pins are used for PWM outputs) and 6 analog input pins, ICSP header, USB port, 16 MHz ceramic resonator, power jack and reset button. It can

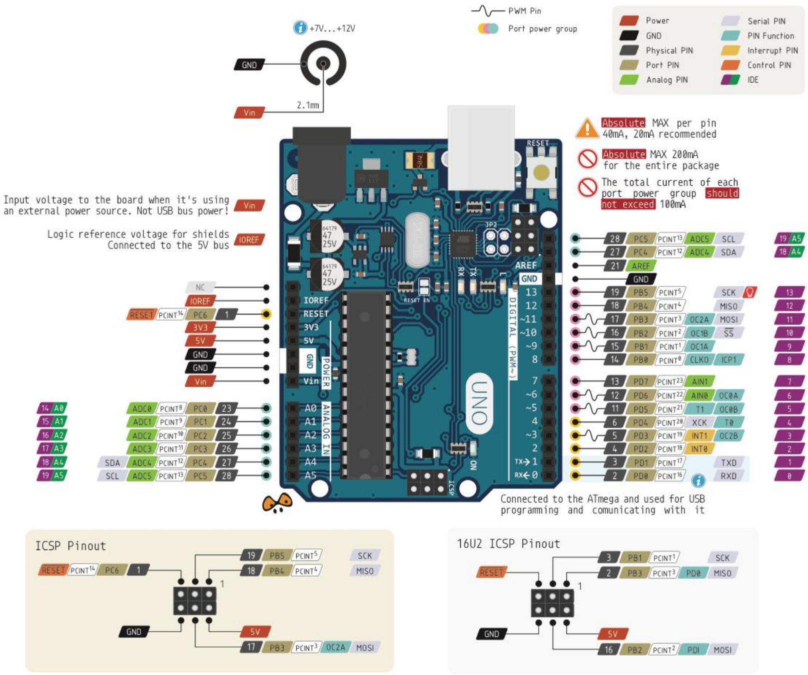

Fig. 3 Arduino Uno pin out diagram

be easily connected to computer with USB port. The major difference between UNO and earlier boards is, it does not use FTDI USB-to-Serial driver chip. Rather it uses Atmega16U2 programmed as USB-to-serial converter.

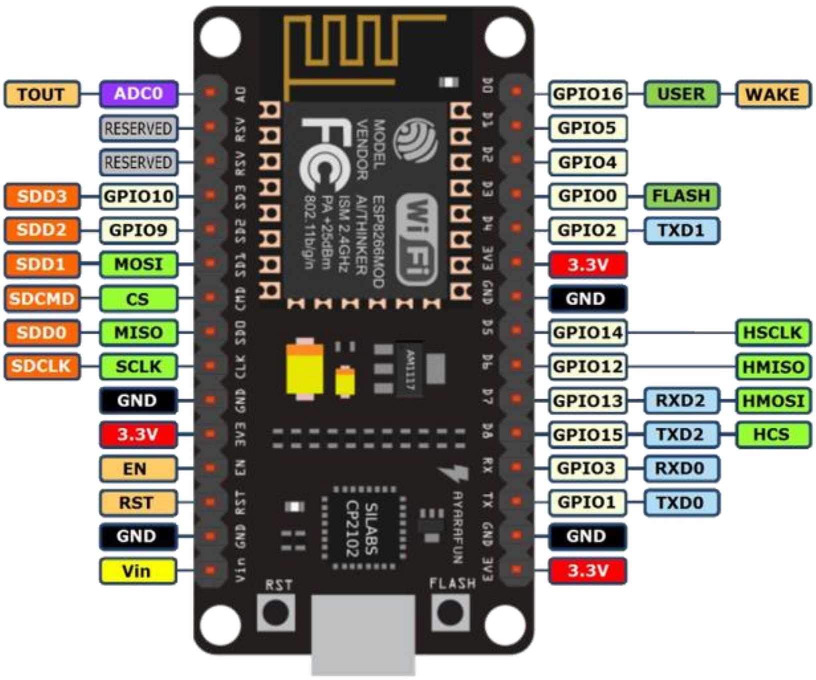

Fig. 2 NodeMCU pin out

3.2 NodeMCU (ESP8266-12E)

NodeMCU is low cost IoT development prototyping board. It is a WiFi SoC (System on Chip) which costs less than 2$, having provided with GPIO pins, ADC's and PWM. It is very compact, portable compared to arduino. ESP8266-12E is developed by Ai-thinker team. The manufacturers has given support for prototyping with NodeMCU in various languages such Lua which is similar to node.js and arduino. The firmware is written in Lua script and development kits are manufactured from Espressif systems [6]. The *Firmware* is a companion project to the popular NodeMCU dev kits, ready-made open source development boards with ESP8266-12E chips. NodeMCU supports SPIFF file system enabling file management and data operation. The technical details are as mentioned in the table

<table>
<tr><th>Categories</th><th>Items</th><th>Values</th></tr>
<tr><td rowspan="2">WiFi parameters</td><td>WiFi Protocols</td><td>802.11 b/g/n</td></tr>
<tr><td>Frequency Range</td><td>2.4 – 2.5Ghz</td></tr>
<tr><td rowspan="9">Hardware Parameters</td><td rowspan="2">Peripheral Bus</td><td>UART/HSPI/I2C/I2S/Ir Remote Control</td></tr>
<tr><td>GPIO /PWM</td></tr>
<tr><td>Operating Voltage</td><td>3.0 ~3.6V</td></tr>
<tr><td>Operating Current</td><td>Average value: 30mA</td></tr>
<tr><td>Operating Temperature Range</td><td>-40° ~ 125° C</td></tr>
<tr><td>Ambient Temperature Range</td><td>Normal temperature</td></tr>
<tr><td>Package Size</td><td>16 mm x 24 mm x 3mm</td></tr>
<tr><td>External Interface</td><td>NA</td></tr>
<tr><td>WiFi mode</td><td>Station/ softAP/ softAP+ station</td></tr>
<tr><td rowspan="4">Software Parameters</td><td>Security</td><td>WPA/WPA2</td></tr>
<tr><td>Encryption</td><td>WEP/TKIP/AES</td></tr>
<tr><td>Firmware upgrade</td><td>UART download / OTA(Over The Air)
Download & write firmware host</td></tr>
</table>

	Software development	Supports cloud server development / SDK for custom development
	Network Protocols	IPv4, TCP/ UDP/ HTTP/ FTP
	User Configuration	AT Instruction Set, Cloud Server, Android/ iOS App[1]

Table 1: Technical specifications of ESP8266-12E

3.3 Raspberry pi

Raspberry pi is a single board extremely small and cheap computer used in different fields like Robotics, Smart home technologies and various educational projects. There are several versions of raspberry pi with different architectures. Present version of raspberry pi (raspberry pi 3b) consists of a quad-core ARM Cortex –A53 processor with a speed range of 700 MHz to 1.4GHz. This allows system to perform faster, when compared to older versions. It is user friendly, flexible device and allows to install different operating systems like windows, Raspbian, Ubuntu, Linux or Windows based IOT core. It provides a platform to connect hardware and software to create several experiments. It collects data from various sensor inputs like Temperature, Humidity, Smoke, fire and PIR sensors networks and sends the data to raspberry pi and analyses the appropriate indication with help of pre-programmed data given by user, which is later sent to software devices like iPad, phones and computers.

The basic thing for every machine is its controller controlling its operations, in addition to that internally connected components transmit data packets to network using network protocols. Controller can be static or user defined. When compared to other controller boards, raspberry pi is quite simple and easy to handle. Its operating system Raspbian gives the opportunity to structure and program the interface according to user requirements easily.

3.3.1 Getting started with Raspberry pi as Server

A raspberry pi is minicomputer requires a monitor, keyboard and mouse as similar to general desktop computer. It lacks the secondary memory, which is provided by means of an external micro SD memory card, preferably 16gigabyte capacity.

3.3.2 Installing OS

1. Download image file of raspbian OS to your working laptop/desktop from raspberry pi official site https://www.raspberrypi.org/downloads/

[1] Esp-12E ai-thinker module

2. Extract the contents of the OS image and copy to SD card.

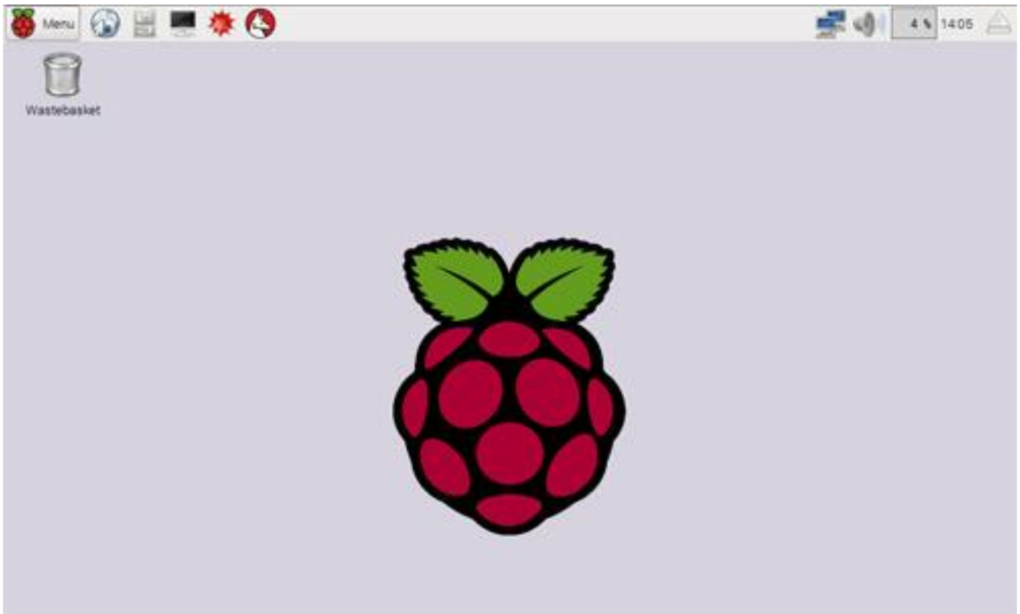

Fig. 4 Raspbian OS Desktop

3. Insert the SD card onto raspberry pi motherboard slot and give all the necessary connections for power, display and inputs.
4. You can see the screen as shown in the figure

3.5 Android

Android is one of the open sources Linux based operating system for the smart phones and tablets, recognized with high number of users. Android provides a common platform for application development, which states the applications developed by the developers can be used on different smart phones, tablets and wearables supported by android

. The operating system of android includes great features. Some of them are as follows

- User interface
- High connectivity
- Memory storage
- Media support
- Multi-touch and Multi-tasking

The mobile applications of android are mostly developed in Java and Kotlin using the Android Software Development Kit.

The platform version of the android is identified with the API level and Version code. API level is the value that identifies the android framework API version given by a version. The present latest platform version with API level and version code is listed below

Version Code	APL Level	Android Platform Version
Oreo	27	8.1.0
Oreo	26	8.1.0
Nougat	25	7.1
Nougat	24	7.0
Marshmallow	23	6.0

Table 2: Android versions

The android application can be developed in different operating systems environment, the following are among them

- Microsoft Windows XP or Higher versions
- Mac OS
- Linux including GNU C Library

Android applications are developed with many advanced technologies, but most of developers rely on the tools such as

- Android Studio (Mostly Used)
- Eclipse IDE (Deprecated)

3.5.1 Android Architecture

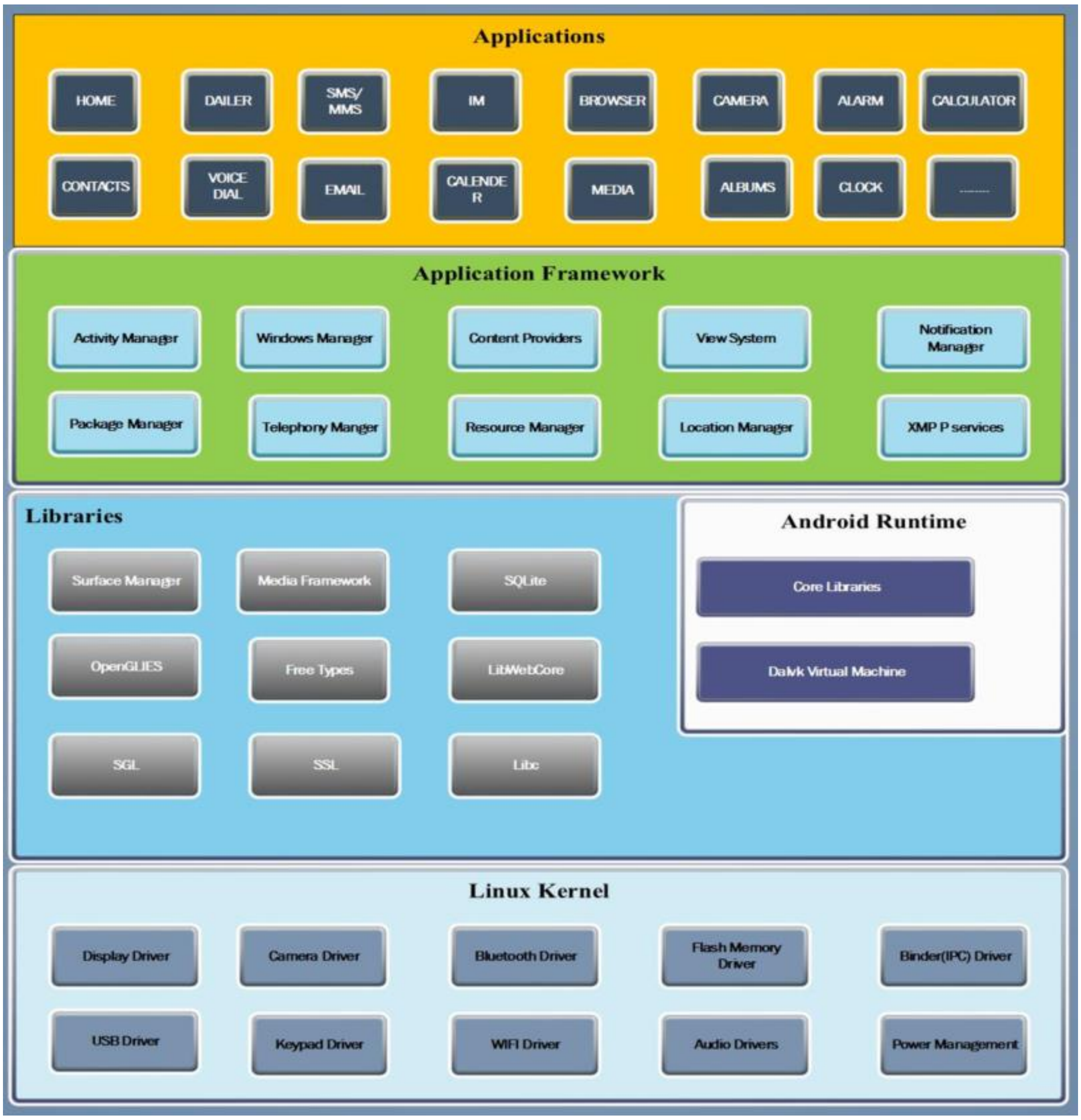

Fig. 5 Android architecture

The android architecture involves of software components on different layers. The architecture of android deals with four main layers out of five as shown in the above figure.

3.5.2 Linux Kernel:

Linux kernel is the bottom layer of the architecture involves the interfacing to peripheral hardware of the device like display drivers, USB drivers, etc.

3.5.3 Libraries:

The libraries of architecture contain set of libraries with open source web kit, SQLite database for storage and sharing data. It also involves SSL libraries meant for internet security.

The following are list of android libraries

- Android. app
- Android. content
- Android. OpenGL
- Android. OS
- Android. Text
- Android. Widget
- Android. Web kit

3.5.4 Android Runtime

The android runtime has java virtual machine optimized and designed for android called Dalvik Virtual Machine. It serves purpose of memory management and multi-threading.

3.5.5 Application Framework

The top second layers deal most of the higher end services for applications by java classes

The framework of android involves with key features cited below

- Activity Manager
- Content Providers
- Resource Manager
- Notifications Manager
- View System

3.5.6. Applications

The top layer of the android architecture is application layer in which the application is installed. Which defines like the mobile having the applications like calendar, calculator, Media player, etc. are viewed and used.

4. Methodology

We believed to set up client server based architecture for distributed home automation. Raspberry pi 3 is reliable and compact mini sized computer suited to set up as main server. ESP8266-12E WiFi enabled microcontroller is used as client with sensors for smart home. An Android app for the user could be good to control the devices remotely. Android device also acts like client enable to manage the devices with GUI. The Major steps involved in the project are as follows.

1. Make raspberry pi suitable to work as a Server

- Setup Raspbian OS and enable static IP.
- Install LAMP Server stack and create Database.
- Write PHP scripts to process the client requests.

2. Program NodeMCU's to work as clients to communicate with the Server.
    - Program 3 NodeMCU's each for RFID sensor, Keypad, Appliances control.
    - Program arduino for sending data from Temperature and Humidity sensor.
3. Build an android App to control and Monitor the devices remotely

5. Implementation

5.1 Setting up the Server by installing LAMP Stack

A LAMP stack is open source software to enable sever for hosting dynamic websites and web apps. The LAMP acronym stands for Linux operating system, Apache HTTP Server, MySQL relational database management system and the PHP programming language [7]. Apache web server is one of the most popular web servers used the internet world. PHP is object oriented programming (OOP) language in server side scripting. It is easy to understand and learn for those who has experience in any other OOPs paradigm.

The server side scripting is implemented in PHP. Although its alternatives Node.js, Node-RED can also be used for this project, PHP is simple, portable compared to the previously mentioned tools which are complex and require in depth knowledge for implementation [8].

LAMP server is setup on raspberry as follows.

1. Install Apache using terminal typing commands

    Sudo apt-get install update

    Sudo apt-get install apache2

2. On displaying text about memory required, Press Y to continue
3. We need to give access permissions for Apache2 to allow .htaccess override */var /www* directory.

    Sudo nano /etc/apache2/apache2.conf

4. Change content as follows

```
<Directory /var/www/>
        Options Indexes FollowSymLinks
        Allow Override All
        Require all granted
</Directory>
```

5. Start apache service by command

    Sudo service apache2 restart

6. Now we can install PHP

    Sudo apt-get install php libapache2- mod-php -y

7. Installing MySQL with following command

    Sudo apt-get install mysql-server php-mysql –y

    Sudo service apache2 restart

8. We can install phpMyAdmin a GUI for managing SQL database and tables to avoid using terminal

    Sudo apt-get install phpmyadmin –y

9. During the installation process, config MySQL password on prompting screen to choose password
10. Configuring static IP on Pi

    Sudo nano /etc/dhcpcd.conf

11. Change file contents as required for Ethernet as well as WLan

    interface eth0 # Wlan0 for WiFi#

    static ip_address=192.168.1.XX/24

    static routers=192.168.1.1 #router IP#

    static domain_name_servers=192.168.1.1

12. If you enter static IP in your browser it displays apache debian home, indicating LAMP server is successfully setup
13. By placing all server side PHP file scripts in the path */var/www/html/* our smart home server starts running [http://www.meccanismocomplesso.org/en/lamp-installing-web-server-on-raspberry]

5.2 Client side scripting

5.2.1 RFID Access system

The concept is developed in such a way that new users can be given access as well disabling access to unauthorised and lost card system. Whenever the person tries to enter into the house with an ID, the Tag value, Name of the user and the Time of access is recorded into the database.

Only key authorised user can alter the Access system with an App on his Android mobile. The Records of data is maintained in the raspberry pi, the main server.

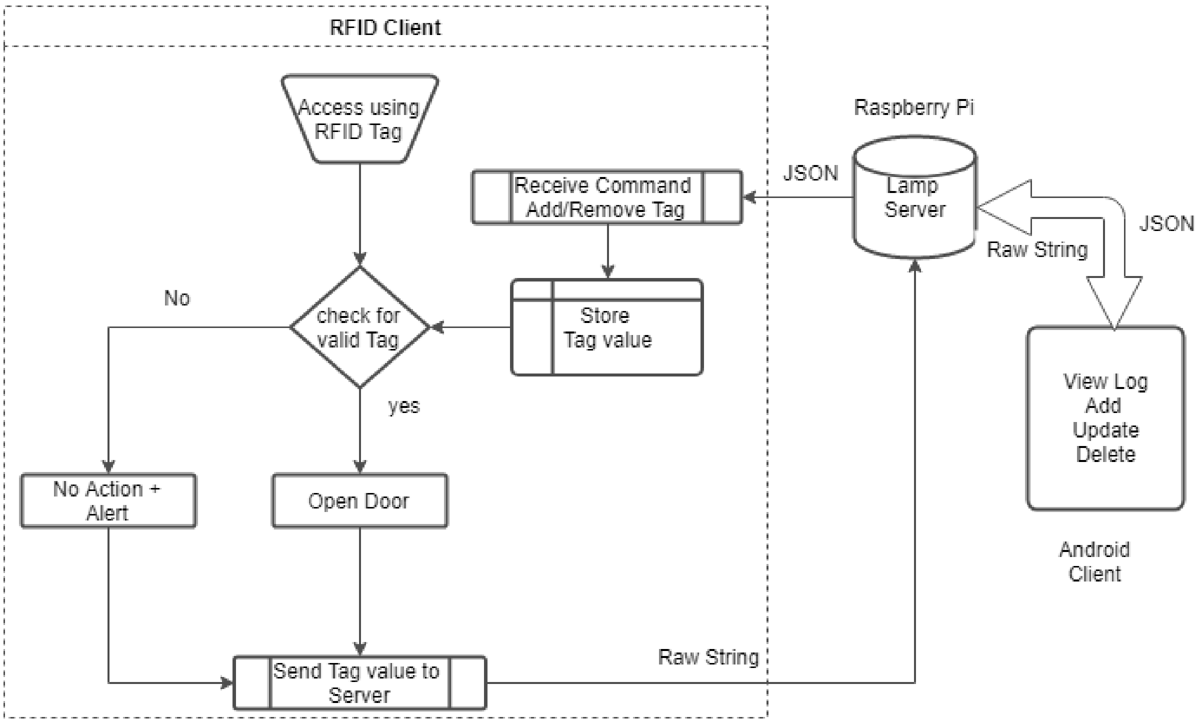

Fig. 6 RFID management flow diagram

Above diagram shows the working environment with RFID client. This process consists of RFID sensor with a tag id, lamp server, android client and database.

Steps carried in this experiment are as follows,

1. When a RFID is accessed, in the first step it compares whether the given input tag is valid or not. If it is a valid tag then the access is granted, invalid tag tends to no action.
2. Both valid and invalid tags are sent to server to store the tag values. Server stores all assessed tag values in database.
3. Lamp server is responsible to receive commands, Add/Remove tag operations are done on this server.
4. Using Android client, we can view, add, delete and update tag values and names.

5.2.2 Temperature and humidity sensor

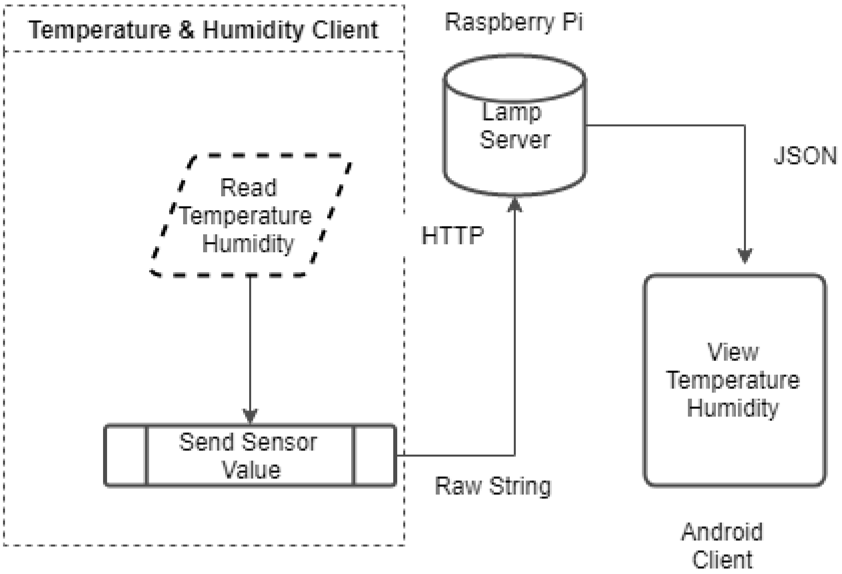

Fig. 7 DHT-11 flow diagram

Above diagram shows the working environment with temperature and humidity client. This process consists of Temperature and Humidity sensors, lamp server, android client and database.

Steps carried in this experiment are as follows,

1. A fully connected Temperature and Humidity sensor network generates the values and sent to the server
2. Sensor reading from controller logic can be understood from the flow diagram.
3. Through the app all readings from the server are visualized

5.2.3 Keypad locker

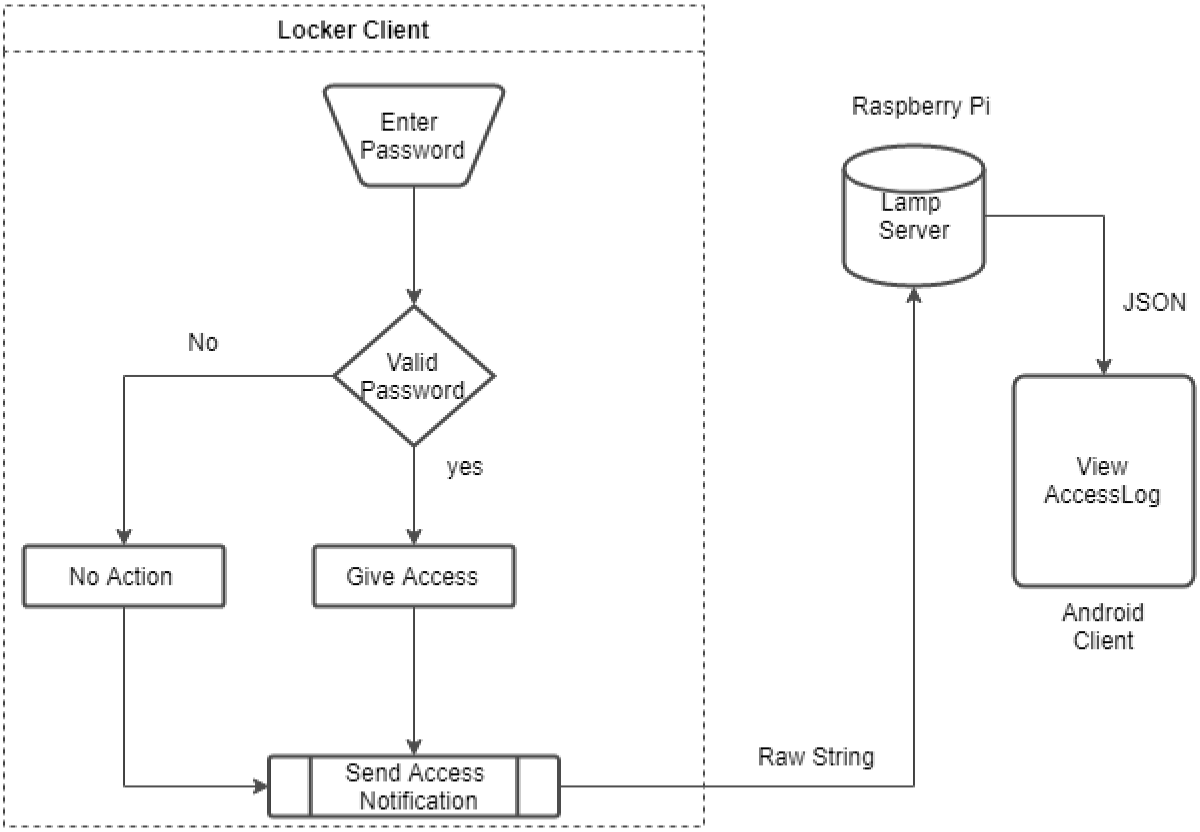

Fig. 8 Keypad (locker) flow diagram

Above diagram shows the working logic with Locker client. This process consists of keypad, lamp server, android client and database.

Steps carried in this experiment are as follows,

1. A password is given to the system in the first step of experiment.
2. In next step system checks whether the given password is valid or invalid.
3. A valid password gets the access and invalid password results in no action.
4. All these accessed notifications are sent to server and viewed on android client

5.2.4 Appliance Control

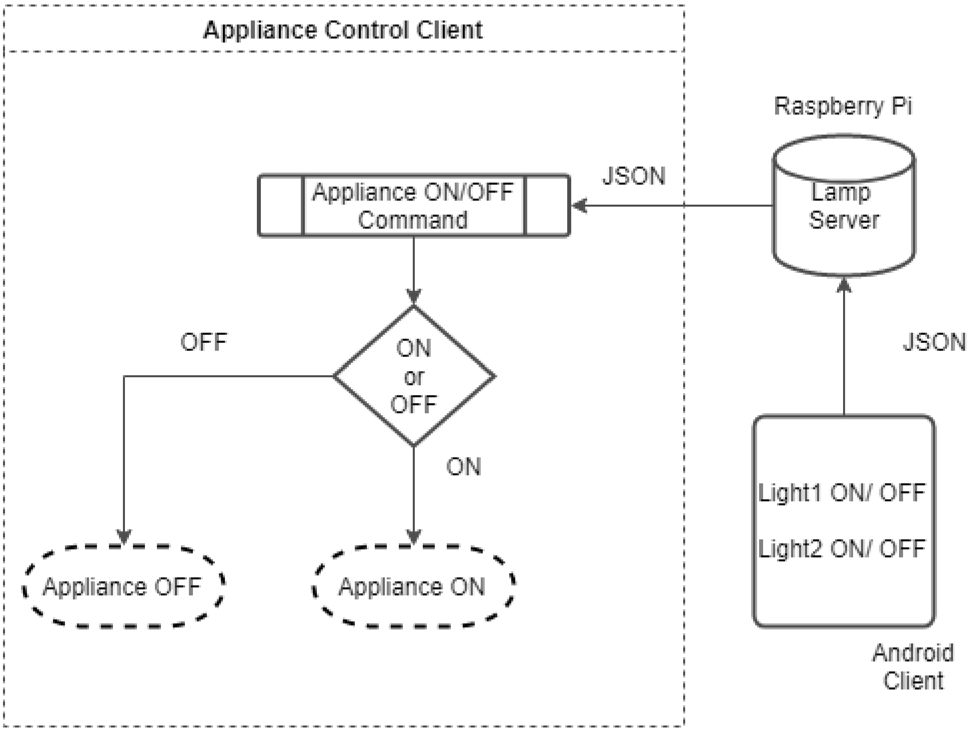

Fig. 9 Appliance control flow digram

Above diagram shows the working environment for Appliances client. This process consists of keypad, lamp server, android client and database.

Steps carried in this experiment are as follows

1. The arduino is always in reception mode to receive commands from the server
2. The server sends commands in a JSON format whether which appliance is ON/OFF like {Appliance: light1, Status: ON}, {Appliance: light1, Status: OFF} etc.
3. JSON library is added to the code so that the client decode JSON then either light1 or light2 or any appliance initialized to the respective pin and the appliance gets switched ON and OFF

5.3 Android App of Home Automation System

5.3.1 System Design and Action Flow

The home automation system android application provides the ease of monitoring and control of the appliances in the smart home. The system design and action flow of the android is represented by the UML diagram (Use Case Diagram).

The Use Case diagram represents the user interaction with an actor or actors to reach the requirements of the system. The whole system functionality is visualized shown in the following Use Case diagram.

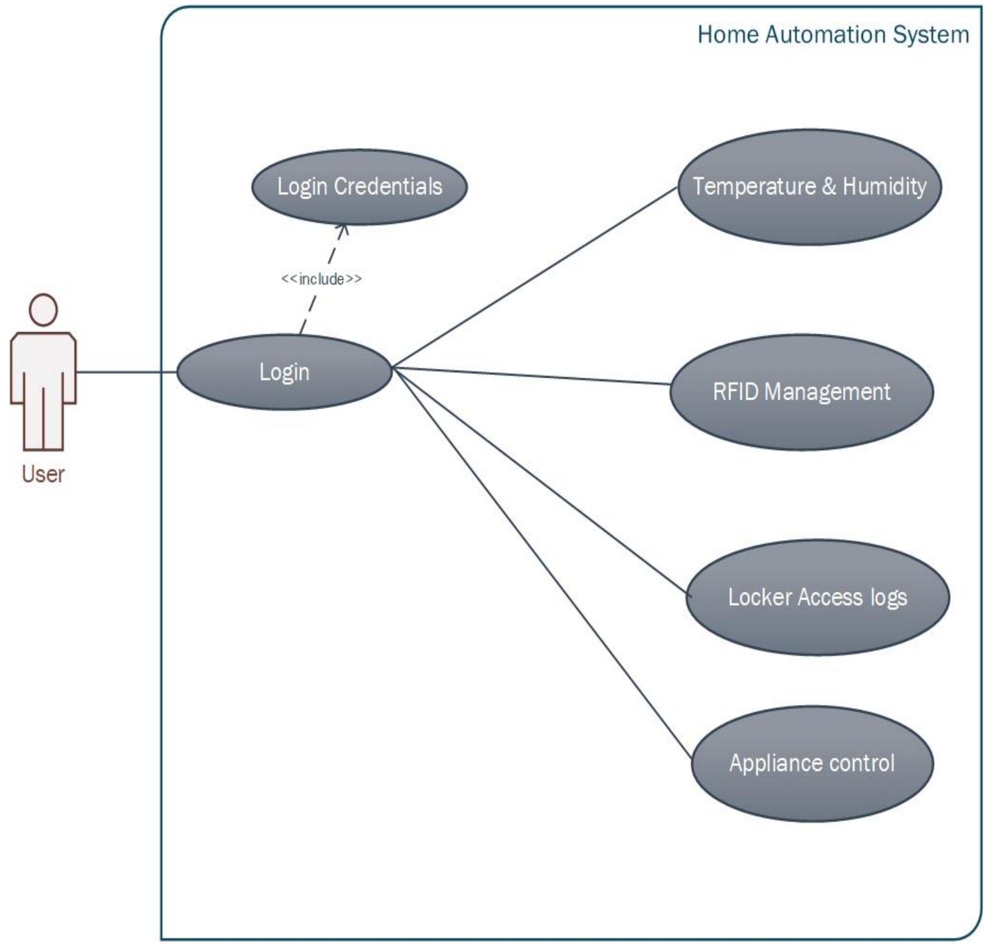

Fig. 10 Use case diagram of the home automation with android

The home automation systems of android application contain four Use Cases defines the systems. The user actions are represented in the form of use case.

- Temperature & Humidity Monitoring
- RFID Management
- Locker Access Logs Monitoring

- Appliance Control

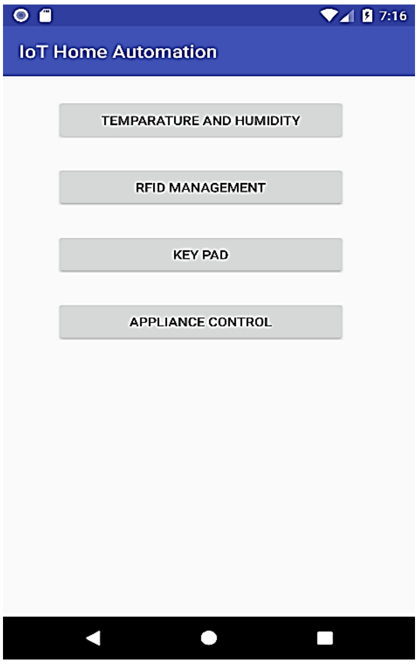

Fig. 11 Smart Home App Home Screen

The home automation system application, monitoring and control are accessed by Login Credentials as shown in the use case diagram with include dependency for login. They include dependency represent the compulsory action to be performed for further action in the system.

5.3.2 Temperature & Humidity Monitoring

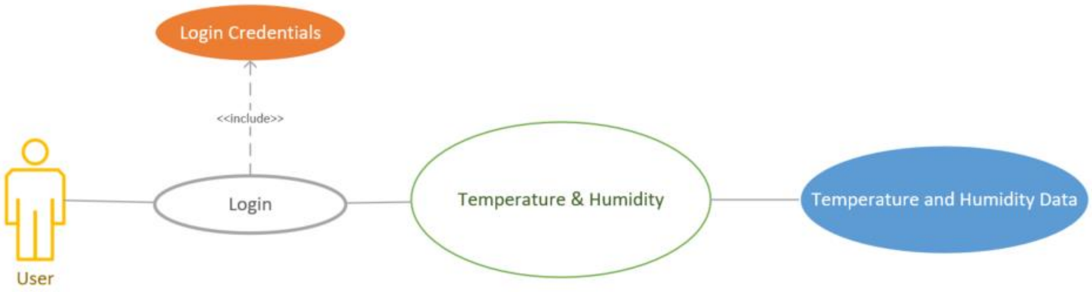

Fig. 12 Use Case: Temperature and Humidity Monitoring

Temperature and Humidity monitoring activity of the android app provides the information of the home environment including the time stamp from LAMP server.

The above use case represents the user interaction in the system to perform the action for monitoring the Temperature and humidity. The user actions are as stated below

- Start the application

- Enter the login credentials to login and to view home screen
- Click on the temperature and humidity activity
- View the live information of the home environment (temperature and humidity)
- Click on home button for further actions if required
- Close the application.

The temperature and humidity activity in android UI is represented in the recycler view including the time stamp, temperature and humidity data as shown below.

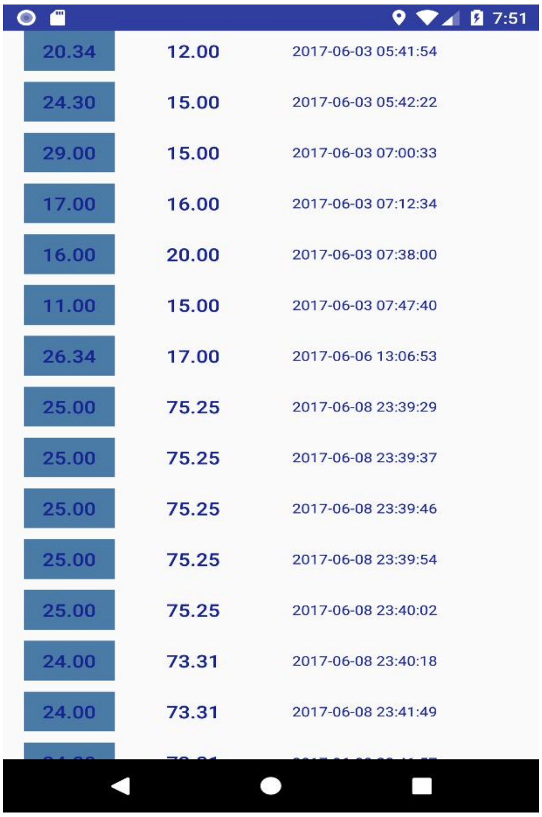

Fig. 13 App displaying Temperature and Humidity

5.3.3 RFID Management

The RFID Management of the system android app includes

- The View logs of the users in the home
- Adding the new user
- Deleting the user
- Updating the user information

The action flow of the RFID management activity is described in the use case diagram below.

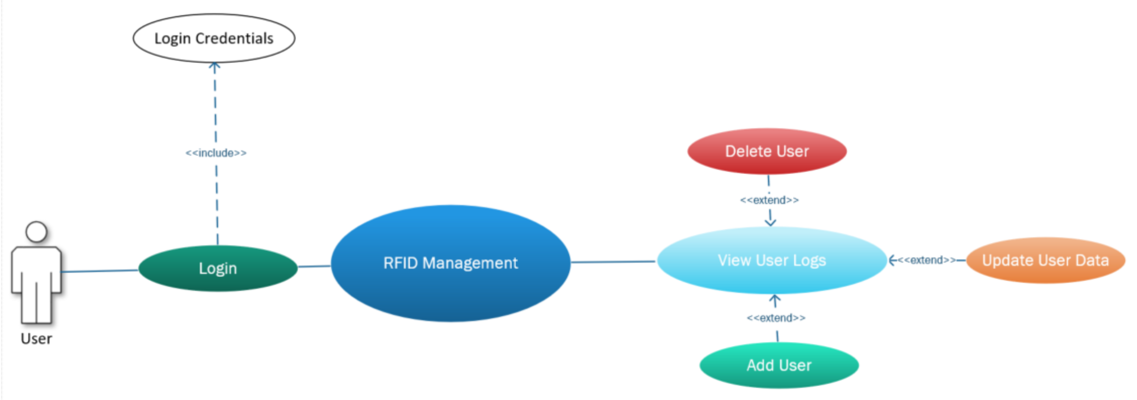

Fig. 14 Use Case RFID Management

The user interactions for the usage of the RFID activity are listed in the following steps

- Start Application
- User login the app with login credentials- include dependency
- Click on RFID Management and perform required action (View logs\ Add User\ Delete User\ Update data)- extend dependency
- Back to home screen by clicking on home button
- Signing off the application

The User interface of view logs looks like android app in the figure and consists of the user logs indicating the accessed users in the recycler view and consisting of two buttons for the addition or deletion of the user.

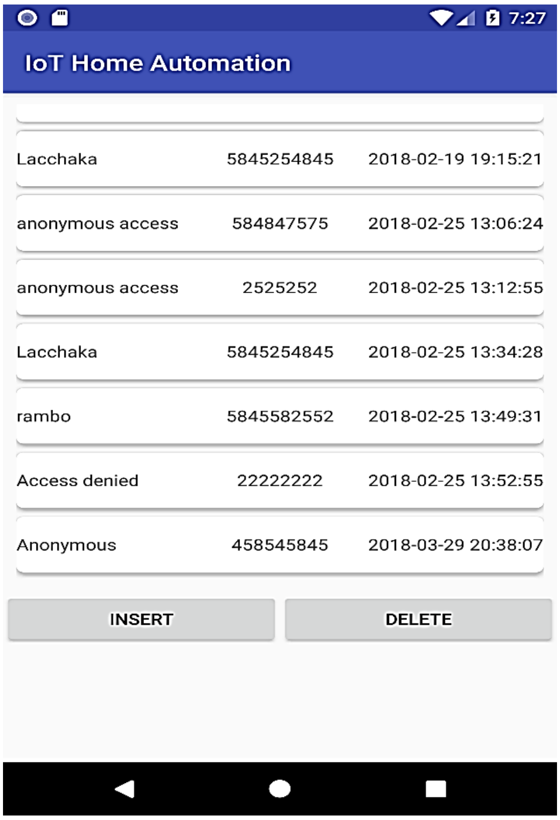

Fig. 15 App Displaying RFID access logs

Add User:

Use Case	**RFID Management** (Addition of new user)
Requirement	App with login credentials.
Actors Goal	To add new user
Pre-Condition	Enter the following details • Uid • Name • Tag Value
Post Condition	Click on insert button
Result	User Added to the server

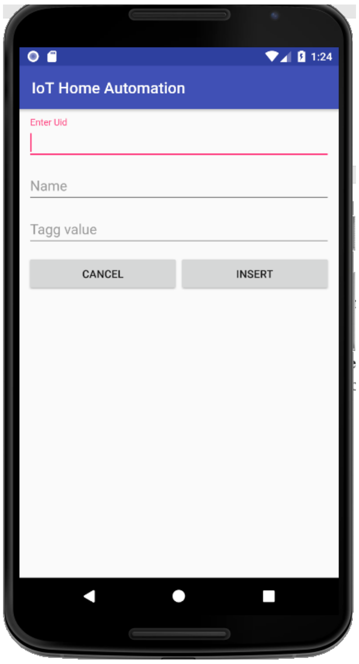

Fig. 17 Add User

Fig. 16 Delete User

Use Case	**RFID Management** (Delete user)
Requirement	App with login credentials.
Actors Goal	To delete user
Pre-Condition	Enter the UID
Post Condition	Click on delete button
Result	User deleted from the server

Update User:

Use Case	**RFID Management** (Update user data)
Requirement	App with login credentials.
Actors Goal	To update user data
Pre-Condition	Enter the UID and name
Post Condition	Click on update button
Result	User data updated to the server

5.3.4 Locker Access Logs Monitoring

The Locker access logs monitoring visualizes the accessed user of the locker or a room in the home. The action flow of the actors or user in the activity are follows

- Start the application
- Enter the login credentials to login and to view home screen
- Click on the locker access logs activity
- View the access log of the users
- Click on home button for further actions if required
- Close the application

The below Use Case of the home automation represent the user interaction for the locker access logs monitoring activity.

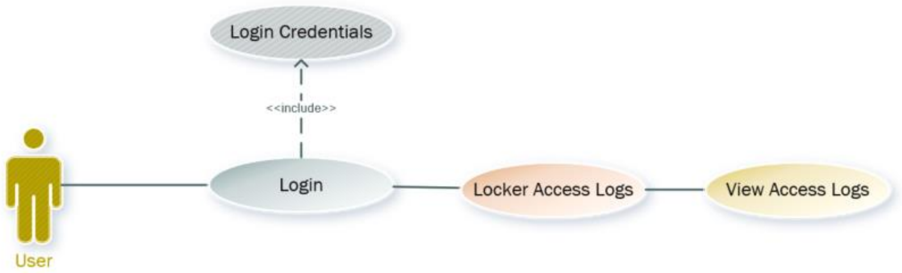

Fig. 18 Use Keypad Locker Access

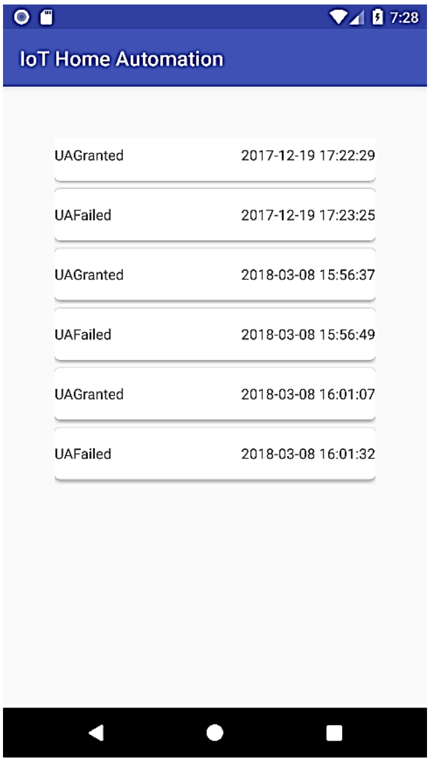

Fig. 19 App displaying access logs

5.3.5 Appliance Control

The appliance control activity defines switching of appliances in the home. The detailed user interactions of the user are represented in the following use case of Appliance control.

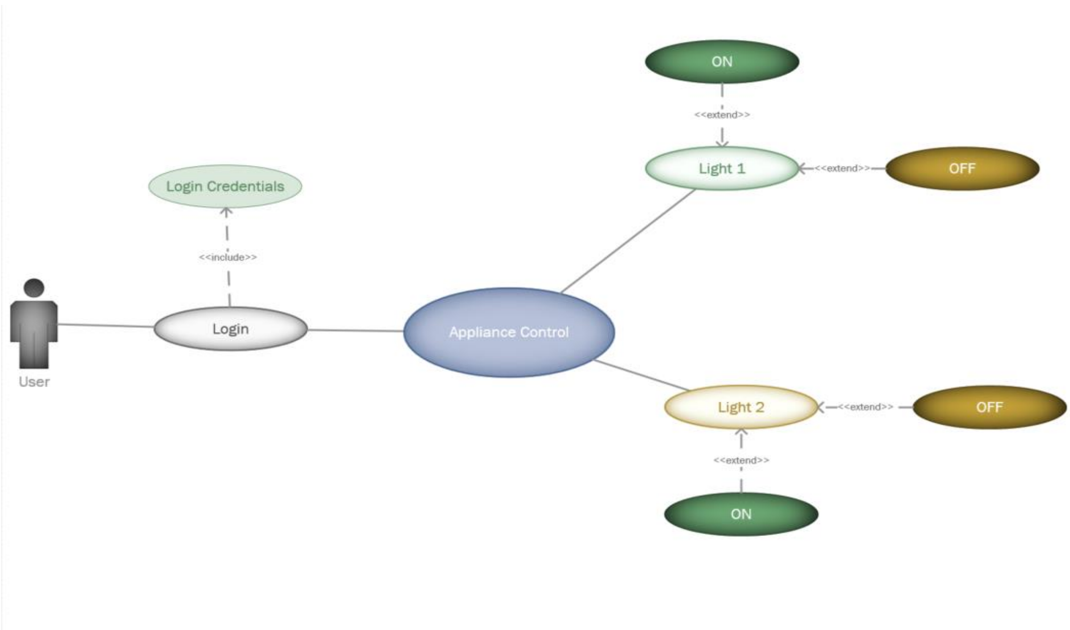

Fig. 20 Use Case: Appliance control

Switching light 1:

Use Case	Appliance control (light 1 ON/OFF)
Requirement	App with login credentials.
Actors Goal	To ON/OFF the light 1
Condition	Click on Toggle button
Result	Achieved the switching action

Switching light 2:

Use Case	Appliance control (light 2 ON/OFF)
Requirement	App with login credentials.
Actors Goal	To ON/OFF the light 2
Condition	Click on Toggle button
Result	Achieved the switching action

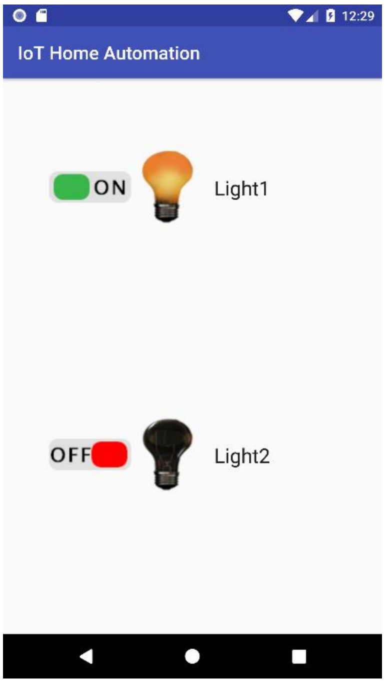

Fig. 21 App user interface for switching

6. Result

Home Automation based on IoT and using Android app is fascinating. With the help of App we could control and monitor devices in the house. By just single touch we could switch on or switch off the lamps. Each lamp can depict any other electric or electronic appliance in the home. Also users can be tracked when each user has entered the home through main entrance using RFID based access door. If someone has opened the safe locker by entering any PIN, the status information gets displayed in the app.

7. Conclusion

Through this project we were able to build a distributed home automation system which portable, flexible, scalable and within price reach. The system worked as the methodology imagined and modelled. We established a client-server architecture, raspberry pi being the central server and app, devices as clients.

Using the hardware capabilities of raspberry pi, additional functionalities can be extended improving the system for more secure and safe home environments using computer vision. The data logs in the database is a resource for further analysis using machine learning. Machine learning on such data sets also can give measure of analytics about device management, user’s personal habits on various home electronics, power consumption and life time of the products.

References

[1] K. Ashton, "rfidjournal," [Online]. Available: http://www.rfidjournal.com/articles/view?4986.

[2] D. Hendricks, "iotevolutionworl," 22 April 2014. [Online]. Available: http://www.iotevolutionworld.com/m2m/articles/376816-history-smart-homes.htm. [Accessed 30 March 2018].

[3] O. H. I. N. T. S.-B. A. O. O. T. A. Abdulrahmana, "Design, Specification and Implementation of a Distributed Home," *Procedia Computer Science,* vol. 94, pp. 473-478, 2016.

[4] M. B. V. V. Reza Abrishambaf, "Distributed home automation system based on IEC61499 function blocks and wireless sensor networks," in *2017 IEEE International Conference on Industrial Technology (ICIT)*, Toronto, ON, Canada, 2017.

[5] "Arduino," Arduino, [Online]. Available: https://www.arduino.cc/en/Guide/Introduction. [Accessed 30 March 2018].

[6] "NodeMCU," [Online]. Available: https://nodemcu.readthedocs.io/en/master/. [Accessed 29 March 2018].

[7] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Arduino.

[8] R. Foundation, Raspberry pi, [Online]. Available: https://github.com/raspberrypi/documentation.

[9] P. Wayner, "infoworld," [Online]. Available: https://www.infoworld.com/article/3166109/application-development/php-vs-nodejs-an-epic-battle-for-developer-mind-share.html. [Accessed 29 March 2018].

[10] [Online]. Available: https://en.wikipedia.org/wiki/LAMP_(software_bundle). [Accessed 29 March 2018].