Project Report

on

Office Automation & Security Using Raspberry Pi as Gateway

CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING

Summer Internship

At C-DAC (Pune)

Under the Guidance of

Mr. Suresh V

Joint Director

ESeG

C-DAC, Pune

Submitted By

Shubham Kumar

(B. Tech, Computer Science & Engineering, Tezpur University)

ACKNOWLEDGEMENT

This project "**Office Automation & Security Using Raspberry Pi as Gateway**" was a great learning experience for me and I am submitting this work to **Centre for Development of Advanced Computing (C-DAC, PUNE)** as a summer internship project.

The satiation and euphoria that accompany the successful completion of the project would be incomplete without mentioning name of the people who made it possible. It is my pleasure to express my sincere thanks to all of them.

First and foremost, I am very much thankful to ***Mr. Sanjay Wandhekar*** (Associate Director & HOD), ***Mr. Manjunatha*** (HPCS Dept.) and ***Mr. Abhishek Das*** (Internship Program Coordinator) for coordinating such a valuable two months summer internship.

I would like to take this opportunity to thank and express my deep sense of gratitude to my guide ***Mr. Suresh V*** (Joint Director, ESeG/IoT)**.** I am greatly indebted to him for providing his valuable guidance at all stages of the study, his advice, constructive suggestions, positive and supportive attitude and continuous encouragement without which it would never have been possible to complete the project.

I am very glad to mention the name of ***Mr. Apoorvaditya Varma*** (Project Engineer, ESeG) for his valuable guidance to work on this project. His guidance and support helped me to overcome various obstacles and intricacies during the course of project work. I am thankful to him for his technical support and to provide and guide me through all the necessities like hardware requirements and internet facilities up to last day of my internship.

Sincerely
Shubham Kumar

CERTIFICATE

TO WHOMSOEVER IT MAY CONCERN

This is to certify that

Mr. Shubham Kumar

Under-graduate student of Tezpur University has successfully completed the summer internship and his project on

Office Automation & Security Using Raspberry Pi as Gateway

from June 5th to August 4th 2017, in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering.

_____

Project Coordinator
Mr. Apoorvaditya Varma
Project Engineer, ESeG, C-DAC, Pune

_____

Project Guide
Mr. Suresh V
Joint Director, ESeG, C-DAC, Pune

Table of Contents

1 Abstract

1.1 Problem Statement

To implement an IoT based smart office model for power-efficient and secured workplace. The project targets to use an IoT model for office automation and security using Raspberry Pi as a gateway. Automation in office means automating the work of different sensors deployed in office to ease out the work of employee. Security in office means to secure certain regions in office and ensure authenticated entry either in office or in that secure area. One of the aim of this project is to do a brief research on power wastage in commercial buildings and by keeping that in reference, to implement an IoT model.

1.2 Abstract

The main purpose of this project is to establish and implement a small IoT model that can be used as for office automation and security. In our day to day life either in office, home or any other place such as school and hospital, we come across many situations where if we implement the effective IoT models for automation and security then it can ease out our life and save both power and time and hence money.

The project focuses on most commonly used modules of office, the door and the employee's cabin and certain secure areas. The highlight of this project is smart automation of the door lock, authentication using RFID, Theft capture system using camera and buzzers and the easy control of the indoor lights and fans switches.

Controlling of these operations can be done through smart device connected to Internet via the gateway and the operations will be performed by interfacing sensors like Rfid tags, buzzer, camera, PIR sensors etc. with Raspberry pi and ESP8266 Module.

This project concludes that these simple yet effective models can be made and can be implemented in a large project as their subpart.

1.3 Introduction

A Smart office can be visualized as a place where we can automate various things by constantly monitoring physical conditions such as temperature, humidity, light intensity etc. and switching appliances or other electronic devices on and off accordingly. The concept of Smart Office takes things a step further by introducing centralized control. In the most advanced form of smart office, there's a computer that does what you normally do yourself: it constantly monitors the state of the office and switches appliances on and off accordingly. So, for example, it monitors light levels coming through the windows and automatically raises and lowers blinds or switches the lights on at dusk. Or it detects movements across the floor and responds appropriately. If it knows you are in office, it switches ON light and fans in different rooms as you walk between them, if it knows you're out, it turns them OFF.

Similarly, in this project we have three parts.

- First Part is the door module where Raspberry Pi (Gateway) is established with sensors like Rfid Tags, Servo Motor and Buzzer. On Authenticated entry of employees tags, motor will automatically open the office door and allows person to enter otherwise buzzer will beep indicating presence of intruder.

- The second part is established in Employee's cabin where we have an ESP8266 Module with PIR sensor and Relays to control fans and light bulbs. The door module is connected with cabin module with the help of MQTT connection in between Raspberry Pi and ESP Module. The moment an employee enters through door, his/her cabin will start functioning. PIR sensor will start sensing the presence of employee and will direct relays to control fans and lights.

- The third part is of secure area where certain regions in office is considered to be highly confidential and only certain employee can enter. In that region, a Raspberry Pi is established with Camera, Buzzer and PIR sensor and whenever PIR sensor senses some movement it will capture image of that intruder and buzzer will start beeping. Both Raspberry Pi are connected with each other through HTTP connection.

2 Literature survey

2.1 Internet of Things

2.1.1 Definition

The **Internet of things** (**IoT**) is the inter-networking of physical devices, vehicles, buildings, and other items embedded with electronics, software, sensors, actuators and network connectivity which enable these objects to collect and exchange data.

In Simple terms, just like we have inter connection of computers forming a network called internet, the connection of various types of devices enabling them to communicate with each other and exchange data and information is defined as Internet of Things.

In Internet of Things every "thing" are provided with unique identifiers and have the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

2.1.2 IoT Architecture

IoT architecture consists of different layers of technologies supporting IoT. It serves to illustrate how various technologies relate to each other and to communicate the scalability, modularity and configuration of IoT deployments in different scenarios.

IoT architecture can be divided into following four layers viz,

1. Sensor Layer
2. Gateways or Network Layer
3. Management Service Layer
4. Application Layer

The lowest layer is made up of smart objects integrated with sensors. The sensors enable the interconnection of the physical and digital worlds allowing real-time information to be collected and processed. There are various types of sensors for different purposes. The sensors have the capacity to

take measurements such as temperature, air quality, speed, humidity, pressure, flow, movement and electricity etc. These sensors are connected to end devices which communicate to gateway.

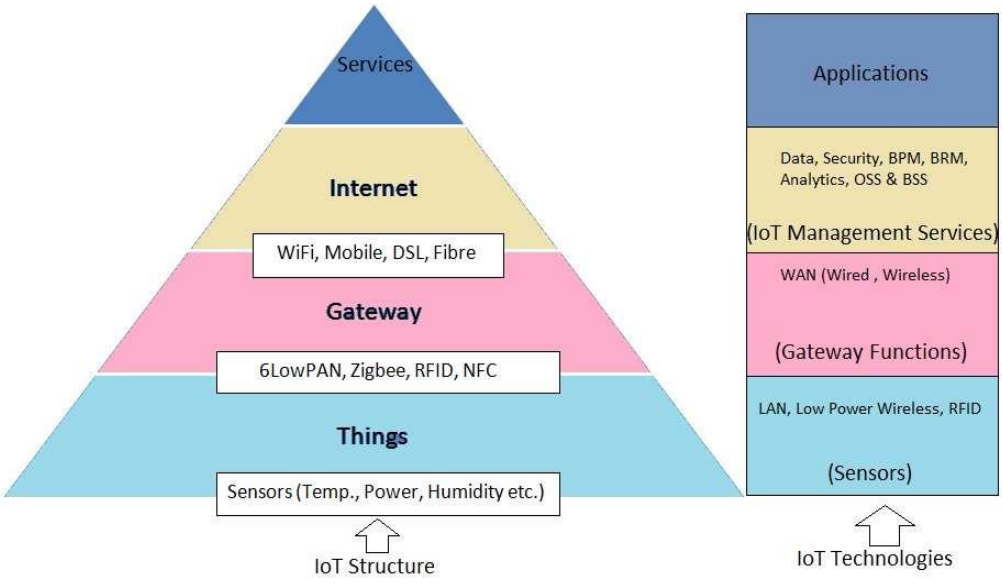


Figure 1 IoT Architecture

Source: http://www.rfwireless-world.com/IoT/IoT-architecture.html

Massive volume of data will be produced by these tiny sensors and this requires a robust and high performance wired or wireless network infrastructure as a transport medium. Current networks, often tied with very different protocols, have been used to support machine-to-machine (M2M) networks and their applications. With demand needed to serve a wider range of IoT services and applications such as high speed transactional services, context-aware applications, etc., multiple networks with various technologies and access protocols are needed to work with each other in a heterogeneous configuration. These networks can be in the form of a private, public or hybrid models and are built to support the communication requirements for latency, bandwidth or security. Gateways can handle multiple communication protocols. The typical wireless technologies used widely are 6LoWPAN, ZigBee, Zwave, RFID, NFC etc. Gateway interfaces with cloud using backbone wireless or wired technologies such as Wi-Fi, Mobile, DSL or Fiber.

The management service renders the processing of information possible through analytics, security controls, process modeling and management of devices.

The IoT application covers “smart” environments/spaces in domains such as: Transportation, Building, City, Lifestyle, Retail, Agriculture, Factory, Supply chain, Emergency, Healthcare, User interaction, Culture and tourism, Environment and Energy.

2.2 Office Automation

Office automation refers to the varied computer machinery and software used to digitally create, collect, store, manipulate, and relay office information needed for accomplishing basic tasks. Raw data storage, electronic transfer, and the management of electronic business information comprise the basic activities of an office automation system. Office automation helps in optimizing or automating existing office procedures. Office automation systems use both hardware as well as software solutions that will ease your workload.

Office automation system refers to using computer based methods of carrying out activities that primarily involve information related activities such as collecting information, storing and retrieving it, analyzing information, taking decisions based on information, and communicating it. As the name implies, these systems are applicable in office environment, which primarily deal with information rather than physical material and activities. However, the office automation systems need to interface with other physical systems, and may cover the information related aspects of physical activities also. For example, an attendance recording system is a type of automated system collects information of employees coming to and leaving the work place, and then uses this this information to prepare their attendance records.

Features are:

- Reduces work load.
- Good at multitasking.
- Boosts feasibility.

Advantages are:

- Office automation can get many tasks accomplished faster.
- It eliminates the need for a large staff.
- Less storage is required to store data.
- Multiple people can update data simultaneously in the event of changes in schedule.
- Reduce costs and overall use of resources.

3 Requirements

3.1 Hardware Requirements

3.1.1 Raspberry Pi 3

The Raspberry Pi is a credit-card-sized computer that plugs into your TV and a keyboard. It is a capable little computer which can be used in electronics projects, and for many of the things that your desktop PC does, like spread sheets, word processing, browsing the internet, and playing games. It also plays high-definition video.

Figure 2 Raspberry Pi 3B

All Models features a Broadcom **SoC** (**System on Chip**), which includes ARM compatible CPU and an on-chip GPU (a videocore IV).System on Chip refers to an integrated circuit that integrates all components of a computer or other electronic devices.

Features of Raspberry pi 3B:

- CPU: Broadcom BCM2837 SOC with 1.2 GHz 64-bit Quad-core ARM Cortex A53.
- GPU: 400MHz Video Core IV multimedia.
- Memory: 1GB LPDDR2-900 SDRAM (i.e. 900MHz)
- USB ports: 4.
- Video outputs: HDMI, composite video (PAL and NTSC) via 3.5 mm jack.
- Network: 10/100Mbps Ethernet and 802.11n Wireless LAN.

***GPIO Pins (General Purpose Input/output pins) of Raspberry pi 3B**:

- GPIO Pins are the Physical interface between Raspberry Pi and outside world.
- Newer Models of Raspberry Pi like 3B has 40 pins. Out of 40 pins, 26 pins are GPIO Pin and other are power or ground pins.
- Input can be from a sensor or a signal from another computer or device. Output can also do anything from turning on a led to sending a signal to another device.

Some Points on Serial Port communication on Raspberry pi 3B:

- In Raspberry Pi, COM1 is equivalent to GPIO14 (Tx) and GPIO15 (Rx) and is called /dev/ttyAMAO.
- In Raspberry Pi, we can use serial port as a terminal to log in, which is useful if you do not have a network connection in hand.
- We need to disable console login so that we alone get control of the port.
- /dev/ttyAMAO was a hardware serial port (UART) and it had very high performance. Hence it was nabbed for Bluetooth in newer versions of Raspberry Pi like Model 3B.
- Second serial port referred to as "mini uart" and lives at /dev/ttySO. It also calculates its bit timings from CPU cores frequency and if the CPU is under heavy load it can corrupt serial communication.
- Thus to Summarize:
    - /dev/ttyAMAO - Bluetooth
    - /dev/ttySO - GPIO Serial Port
- On Raspberry Pi 3, the second serial port is called /dev/ttySO and is by default mapped to GPIO Pins 14 and 15.
- Serial Port aliases – Serial 0 and Serial 1
- Raspberry Pi kernel sorts out where these points to, depending on which Raspberry Pi you are on.
- On Raspberry Pi 3 – Serial 0 will point to GPIO Pins 14 and 15 and use the "mini-uart" or /dev/ttySO.

* Refer to Annexure 7.1

3.1.2 Camera

Figure 3 Logitech Webcam

This webcam is directly connected via usb on Raspberry Pi. Using fswebcam module, we can directly run webcam through shell commands and capture images. Fswebcam is a small and simple webcam app for *nix. It can capture images from a number of different sources and perform simple manipulation on the captured image. The image can be saved as one or more PNG or JPEG files.

3.1.3 Grove Buzzer

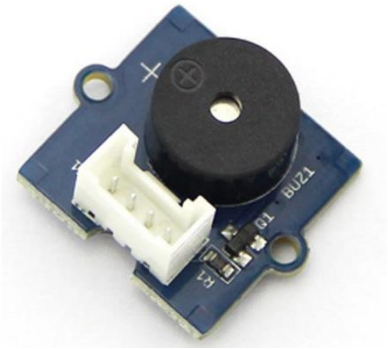


Figure 4 Grove Buzzer

Buzzer comes in two varieties.

Active buzzer: Just outputs a single tone when connected to Vcc and ground.

Passive buzzer: Similar to loudspeaker that needs a signal to make it work.

The Grove - Buzzer module has a piezo buzzer as the main component. The piezo can be connected to digital outputs, and will emit a tone when the output is HIGH. Alternatively, it can be connected to an analog pulse-width modulation output to generate various tones and effects.
It has four pins – Ground, VCC, NC and SIG.

Specifications:

- Operating Voltage: 4 to 8 V
- Sound Output ≥ 85 dB
- Resonant Frequency: 2300 ± 300 Hz.

3.1.4 RFID 125Khz Robokits Reader with tags

RFID – RADIO FREQUENCY IDENTIFICATION. It provides unique wireless identification for different objects. They do not have any power source, they take powers from nearby radio frequency emitters (called RFID readers). Thus this allow them to be of minimal size, but it also limit the distance they operate. In order to RFID to work, it needs antenna which make it larger. By combining RFID and its antenna, we create something called RFID tag.

Figure 5 Robokits RFID Reader

Specifications:

- Voltage: 5VDC
- Operating Frequency: 125Khz
- Read Distance: 10cm
- Data baud rate:9600BPS
- RFID tag data: 10 ASCII DATA （card no.）+ 2 ASCII DATA （XOR result）
- RFID Tag: Passive.

Pinouts of RFID reader [Refer to Annexure 7.1]

3.1.5 PIR Sensor

A **passive infrared sensor** (**PIR sensor**) is an electronic sensor that measures infrared (IR) light radiating from objects in its field of view. They are most often used in PIR-based motion detectors. All objects with a temperature above absolute zero emit heat energy in the form of radiation.

Usually this radiation isn't visible to the human eye because it radiates at infrared wavelengths, but it can be detected by electronic devices designed for such a purpose.

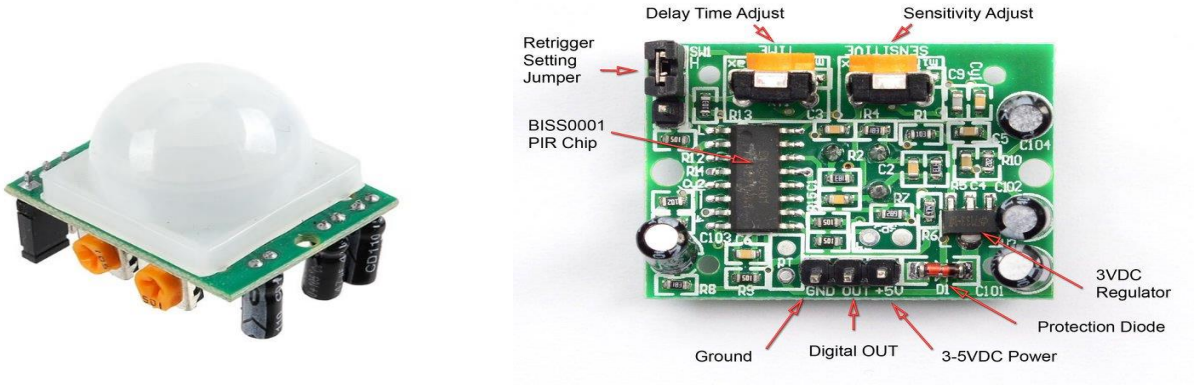


Figure 6 Bottom view PIR sensor

The term *passive* in this instance refers to the fact that PIR devices do not generate or radiate energy for detection purposes. They work entirely by detecting infrared radiation emitted by or reflected from objects. They do not detect or measure "heat".

A PIR-based motion detector is used to sense movement of people, animals, or other objects. They are commonly used in burglar alarms and automatically-activated lighting systems. They are commonly called simply "PIR", or sometimes "PID", for "passive infrared detector".

Newer Models of PIR sensors have two knobs which can be used to adjust the delay time and sensitivity of PIR sensors. These parameters can be adjusted by rotating the knobs in Anti-Clockwise (For Minimum value) and Clock-wise (For Maximum value).

3.1.6 Grove Servo

Grove - Servo is DC motor with gearing and feedback system. It is used in driving mechanism of robots. It has following features:

- Small Module
- Grove compatible interface
- Easy to use

Figure 7 Grove Servo

Specifications:

- Working Voltage : 4.8 to 6.0 V
- Torque : 1.5/1.8 Kg.cm
- Speed : 0.12/0.16 S/60º
- Size : 32 x 11.5 x 24 mm
- Weight : 8.5 gm

Usage: It has three wires: power, ground and signal.

- Power – Color: Red, connected to 5V
- Ground – Color: Black
- Signal – Color: Yellow/Orange/White.

3.1.7 NodeMCU

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the dev kits. The firmware uses the Lua scripting language. NodeMCU was created shortly after the ESP8266 came out.

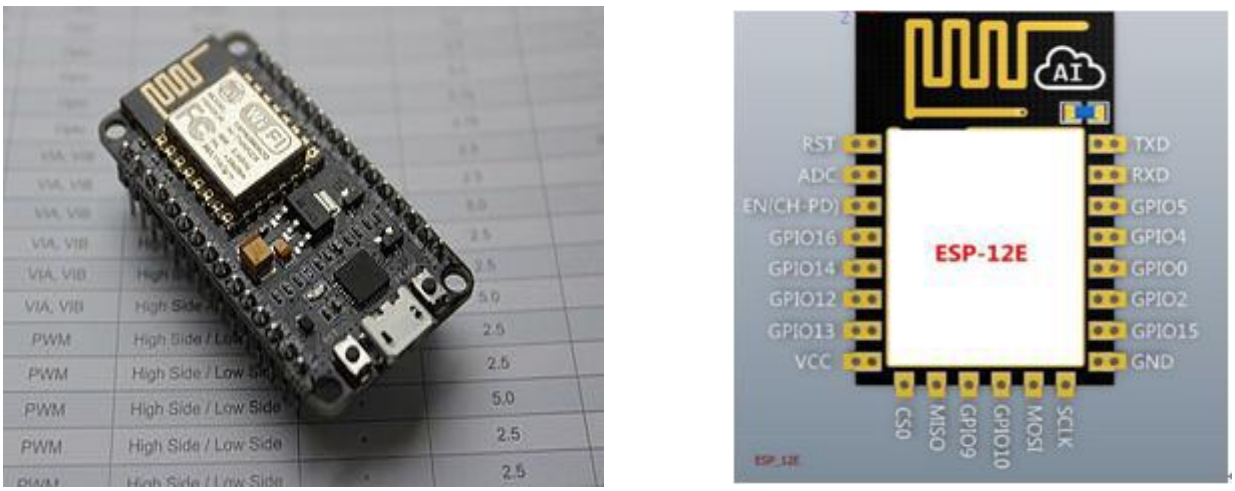


Figure 8 NodeMCU and ESP-12E Wi-Fi Module

ESP-12E Wi-Fi module is developed by Ai-thinker Team. core processor ESP8266 in smaller sizes of the module encapsulates Tensilica L106 integrates industry-leading ultra-low power 32-bit MCU micro, with the 16-bit short mode, Clock speed support 80 MHz, 160 MHz, supports the RTOS,

Integrated Wi-Fi MAC/BB/RF/PA/LNA, on-board antenna. ESP8266EX offers a complete and self-contained Wi-Fi networking solution; it can be used to host the application or to offload Wi-Fi networking functions from another application processor. When ESP8266EX hosts the application, it boots up directly from an external flash. In has integrated cache to improve the performance of the system in such applications.

NodeMCU and ESP8266 Pinouts [Refer to Annexure 7.1]

3.1.8 LED

A **light-emitting diode** (**LED**) is a two-lead semiconductor light source. It is a p–n junction diode that emits light when activated. When a suitable voltage is applied to the leads, electrons are able to recombine with electron holes within the device, releasing energy in the form of photons. This effect is called electroluminescence, and the color of the light (corresponding to the energy of the photon) is determined by the energy band gap of the semiconductor.

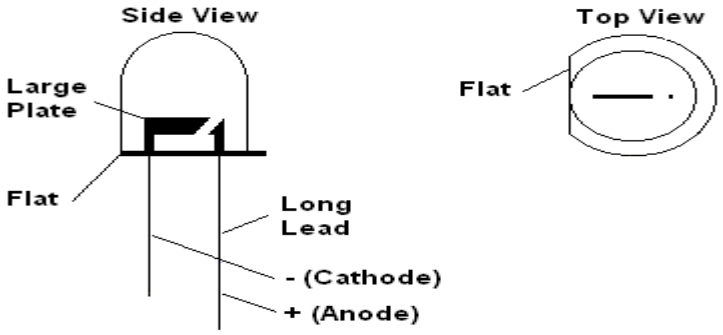


Figure 9 LED Pinouts

In this Project for small demo purpose, I have used led which will turn on or off based on presence detected by PIR sensor, but we can also attach relay which will control bulbs, fans etc. in the cabin.

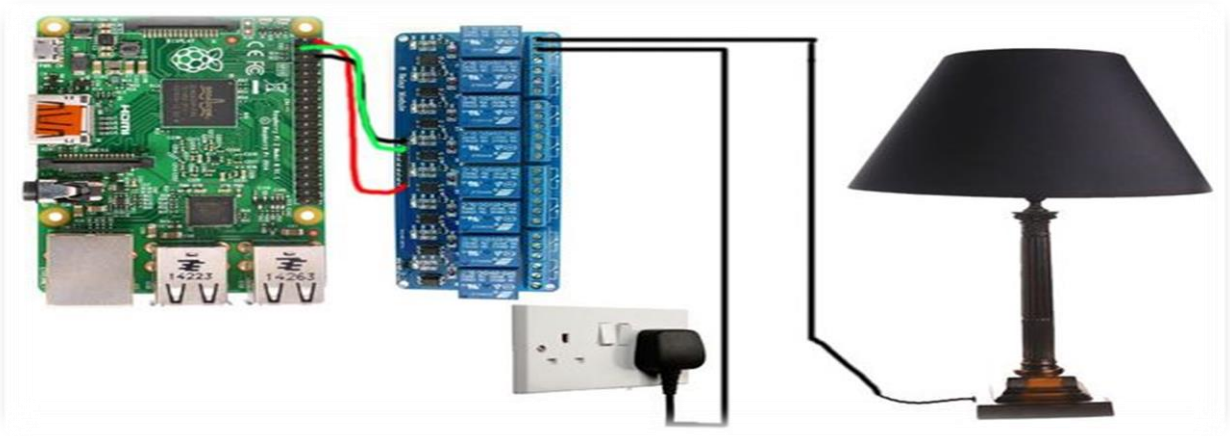


Figure 10 Lamp connected via Relay

3.2 Software Requirements

3.2.1 Operating System

Raspbian OS is installed on the Raspberry Pi on the micro SD card. **Raspbian** is the Raspberry Foundation's official supported Operating System. Apart from this, many third parties OS can also be installed on the Raspberry Pi like Ubuntu Mate, Snappy Ubuntu Core, WIN 10 IOT Core, RISC OS etc.

Foundation has also provided an easy OS installer for beginners called NOOBS. Raspbian was created by Mike Thompson and Peter Green as an independent project. Raspbian uses PIXEL, **P**i **I**mproved **X**windows **E**nvironment, **L**ightweight as its main desktop environment as of the latest update. It is composed of a modified LXDE desktop environment and the Openbox stacking window manager with a new theme and few other changes. The distribution is shipped with a copy of computer algebra program Mathematica and a version of Minecraft called Minecraft Pi as well as a lightweight version of Chromium as of the latest version.

3.2.2 Developing Language – Node.js

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux. Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

Features of Node JS:

- **Asynchronous and Event Driven** − All APIs of Node.js library are asynchronous, that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call.

- **Very Fast** − Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.
- **Single Threaded but Highly Scalable** − Node.js uses a single threaded model with event looping. Event mechanism helps the server to respond in a non-blocking way and makes the server highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.
- **No Buffering** − Node.js applications never buffer any data. These applications simply output the data in chunks.
- **Huge ecosystem of open source packages (npm) –** npm stands for node package manager. It allows to install different packages for our application.

Why Node.js?

- Because it is known for its speed and efficiency, thus making it great for real time applications.
- And we know IOT is reliant on data which comes from real time devices and apps.
- Node.js also provides a way for secure communication.
- It is fast because it is asynchronous and built on Google Chrome's V8 JavaScript Engine.

Express JS

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. With a myriad of HTTP utility methods and middleware at your disposal, creating a robust API is quick and easy. Express provides a thin layer of fundamental web application features, without obscuring Node.js features .Many popular frameworks are based on Express.

Unlike its competitors like Rails and Django, which have an opinionated way of building applications, express has no "best way" do something. It is very flexible and pluggable.

- Easy and flexible routing system.
- Integrates with many templating engines.
- Contains a middleware framework.

3.2.3 Database - MongoDB

MongoDB is an open-source document database and leading NoSQL database. MongoDB is written in C++. This tutorial will give you great understanding on MongoDB concepts needed to create and deploy a highly scalable and performance-oriented database.

Features

- Schema less − MongoDB is a document database in which one collection holds different documents. Number of fields, content and size of the document can differ from one document to another.
- Structure of a single object is clear.
- No complex joins.
- Deep query-ability. MongoDB supports dynamic queries on documents using a document-based query language that's nearly as powerful as SQL.
- Conversion/mapping of application objects to database objects not needed.
- Uses internal memory for storing the (windowed) working set, enabling faster access of data.

Why MongoDB?

- In IOT, every day we can come up with new type of devices or sensors, which will have new type and functions of data.
- Thus we cannot rely on the databases which were built for processing structured data.
- Because an IoT model may have a varieties of sensors processing different set of data and have different functions for these data.
- This is where mongoDB makes the work of an IoT application easy by providing high performance and by storing data in collections and documents.

3.2.4 Arduino IDE

The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java. It originated from the IDE for the languages Processing and Wiring.

In this Project, I have used Arduino IDE for writing codes for ESP8266 i.e. for cabin Module where a PIR sensor is deployed with led or relay. Arduino IDE comes very handy when we have to write code and flash into microcontrollers like NodeMCU. One of the advantage of IDE is that we can include libraries for various sensors and microcontrollers, it provide support for a large number of custom and third party libraries.

For NodeMCU (ESP – 12E Module), go to File → Preferences → Additional Boards Manager URLs and type – http://arduino.esp8266.com/stable/package_esp8266com_index.json

3.2.5 Fswebcam

Rather than using the Raspberry Pi camera module, we can use a standard USB webcam to take pictures and video on the Raspberry Pi. For using webcams on Raspberry Pi, we can install fswebcam and capture images.

Installing fswebcam package: `sudo apt-get install fswebcam`

Basic Usage: Open terminal, enter command fswebcam followed by desired name of image.
E.g. - `fswebcam image.jpg`

A more advance command with various parameters: `fswebcam -r 1280x720 --no-banner /home/pi/webcam/image_name.jpg`
`-r` sets resolution
`--no-banner` represents there should not be any label or banner on image.
`/home/pi/webcam/` represents path or location where image is to be stored.
`image_name.jpg` represents name of image.

A more detailed information about fswebcam and various options and parameters can be found here - https://manpages.debian.org/jessie/fswebcam/fswebcam.1.en.html

In this Project, I have written this fswebcam command in a shell file and to capture images I have simply run this file through node.js codes.

3.3 Communication Protocols

3.3.1 MQTT

MQTT stands for Message Queue Telemetry Transport. It is an ISO standard publish-subscribe lightweight messaging protocol for use on top of TCP/IP Protocol. It is a Publish/subscribe, extremely simple and lightweight messaging protocol, designed for constrained devices and low-bandwidth, high-latency or unreliable networks. The design principles are to minimize network bandwidth and device resource requirements whilst also attempting to ensure reliability and some degree of assurance of delivery. These principles also turn out to make the protocol ideal of the emerging "machine-to-machine" (M2M) or "Internet of Things" world of connected devices, and for mobile applications where bandwidth and battery power are at a premium.

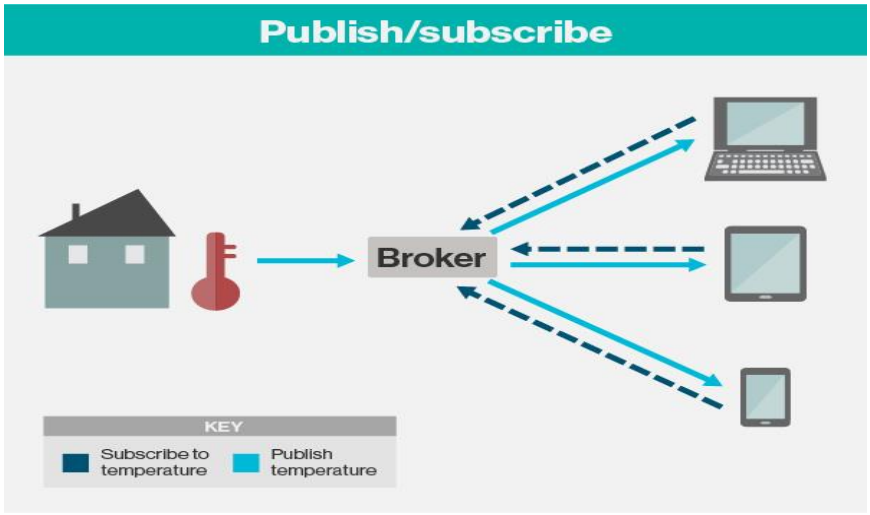


Figure 11 MQTT Publish/subscribe

Source: http://internetofthingsagenda.techtarget.com/definition/MQTT-MQ-Telemetry-Transport

These characteristics make it ideal for use in constrained environments or low-bandwidth networks with limited processing capabilities, small memory capacities and high latency. The MQTT design minimizes network bandwidth requirements while attempting to ensure reliability of data.

Publish-Subscribe messaging Pattern:

- Publisher and subscriber does not have knowledge of each other.
- Publisher do not program the message to be sent directly to specific receivers or subscribers.
- Instead they characterize published message into classes without knowledge of which subscribers is present.

- Subscribers express interest in one or more classes and only receive messages that are of interest, without knowledge of publishers.

Publish Subscribe messaging Pattern requires a message broker. Publishers post messages to an intermediary broker and subscriber register subscriptions with that broker, letting broker perform filtering. Message filtering is of topic-based, content-based and hybrid. The broker normally performs a store and forward function to route message from Publisher to subscriber.

In this project I have used Mosquitto as broker which was installed on Raspberry Pi. The MQTT Protocol was used between Door module (Raspberry Pi) and indoor module (ESP8266). The topics used were AUTH_EMP and EMP_AVAILABILITY.
AUTH_EMP represents which authorized employee entered through door, so that his cabin can start functioning. EMP_AVAILABILITY represents whether employee is present in his cabin or not.

3.3.2 HTTP

The Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, and hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web. HTTP functions as a request–response protocol in the client–server computing model. A web browser, for example, may be the client and an application running on a computer hosting a website may be the server. The client submits an HTTP request message to the server. The server, which provides resources such as HTML files and other content, or performs other functions on behalf of the client, returns a response message to the client. The response contains completion status information about the request and may also contain requested content in its message body.

In this Project this protocol is used twice.

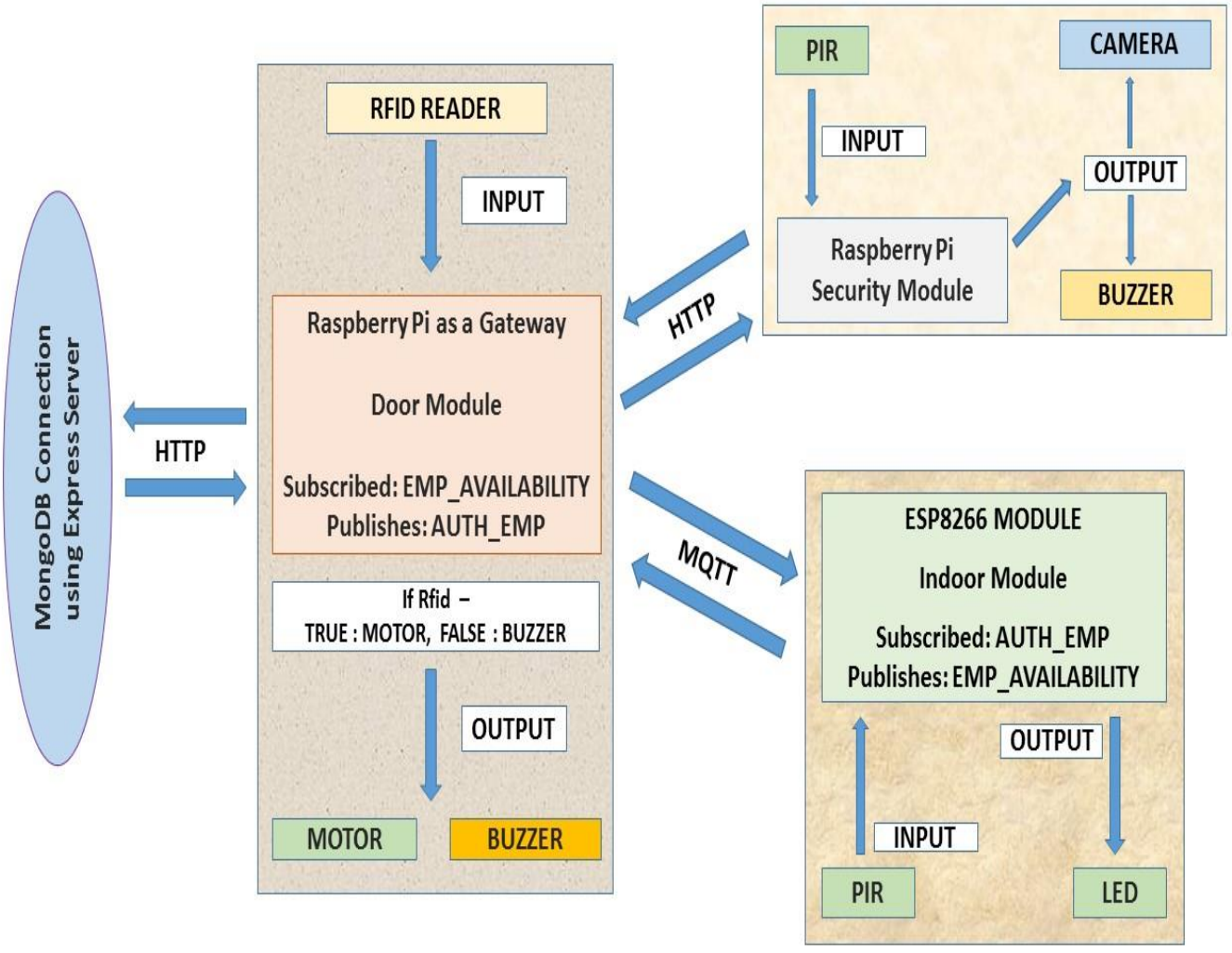
- First HTTP connection is between MongoDB and express application running on Raspberry Pi (Door Module). It is used to create, update, retrieve data from database and display it on web application built.
- Second HTTP connection is between Door module and Security Module where using this Protocol we can access images of intruder captured on security module from web application running on Door module.

4 System Design and Flowchart

This Project can be divided into three modules and by understanding working and flow of these individual modules, we can know the flow of whole system. Here are the three modules:

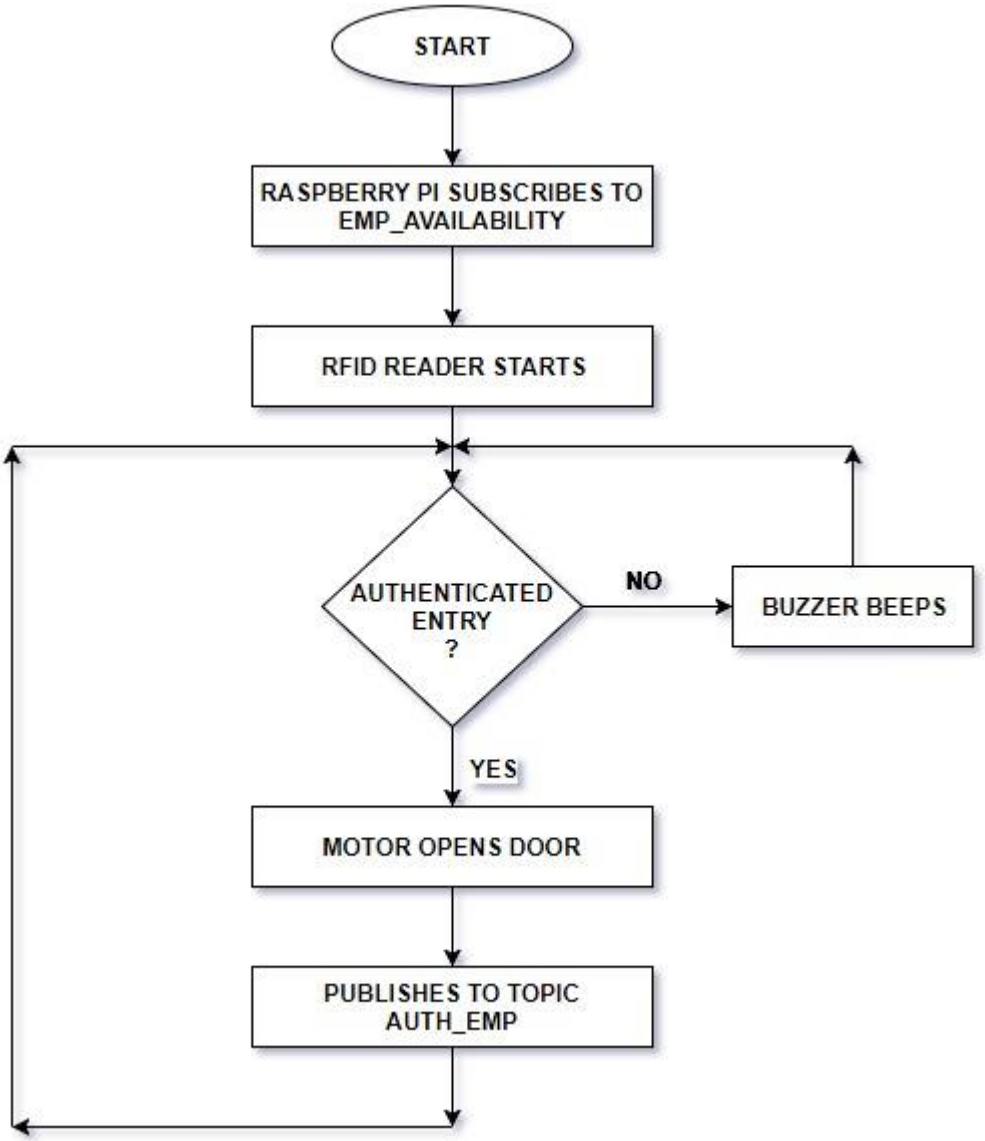
- Door Module
- Indoor Module
- Security Module

Indoor module and security module is connected with the door module which in turn connected to the database. The overview of whole system can be seen in this diagram.



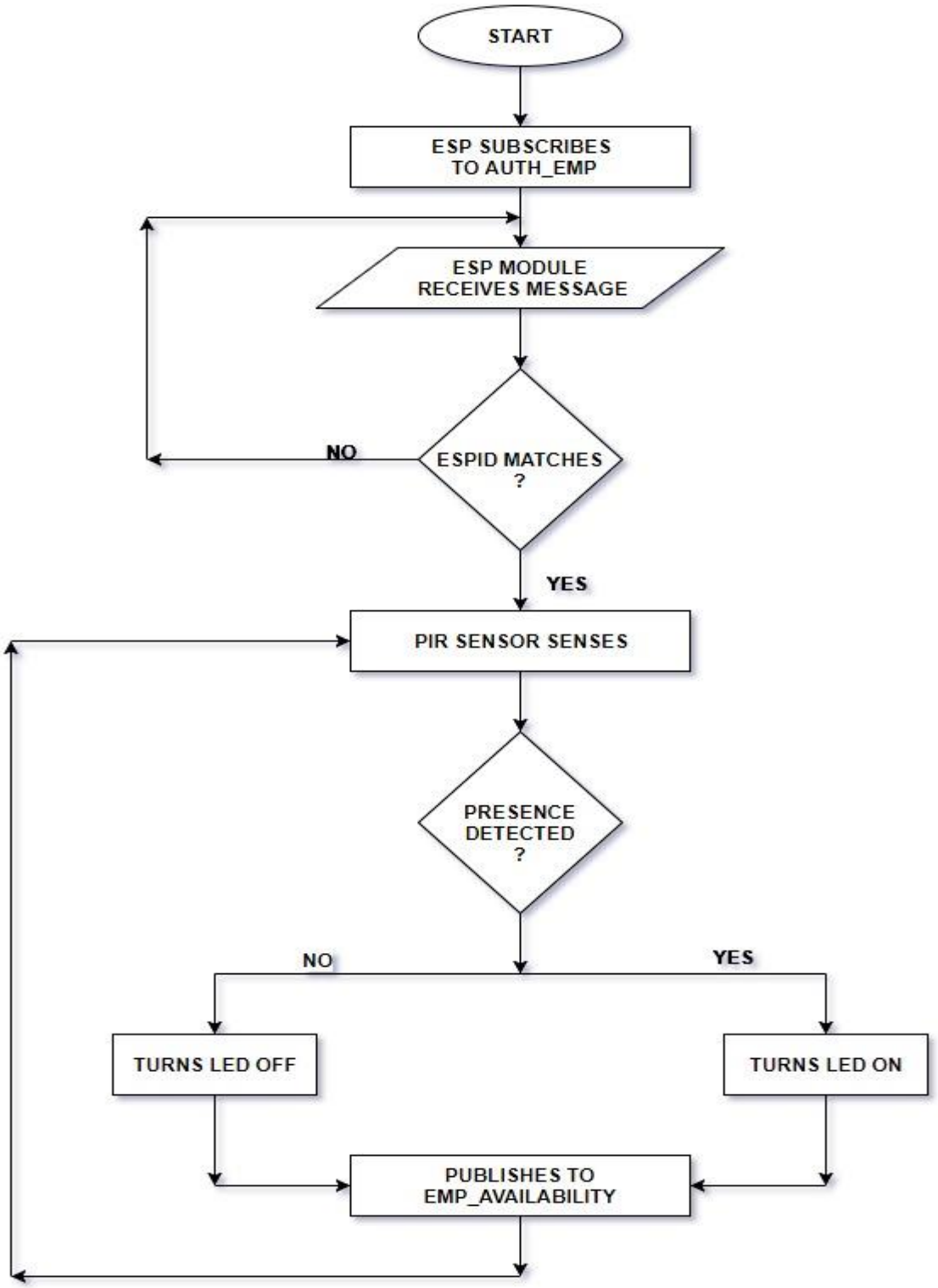
4.1 Door Module

- It comprises of Raspberry Pi 3B, RFID Reader, Motor and Buzzer.
- Raspberry deployed is the gateway and is connected to MongoDB (Database).
- The Door Module is connected to Indoor Module with MQTT connection and to security module with HTTP connection.
- Raspberry Pi subscribes to the topic EMP_AVAILABILITY and publishes to the topic EMP_AUTH to the broker.
- Initially Raspberry Pi will be running. Whenever an employee with Rfid Tags will come to door and will tap their tags on reader, the tags will be matched with those stored on database. If user is genuine i.e. his tag information is stored on database then motor will rotate to open the door and allow employee to enter and go to the respective cabin. If tag is not identified, buzzer will start beeping indicating intruder's presence.
- Following process is automated when an employee enters through the door.
  - First of all the database is updated with current date and time as last swipe time of that employee.
  - The MQTT connection established between Door Module and Indoor Module helps in sending signal as a message through broker so that only his/her cabin's ESP Module will start functioning and message exchange will now take place on the respective topics.
- An express based web application is made which has following functionalities.
  - It allows admin to login.
  - Admin then can register new employee.
  - He can see the last time an employee entered through door and whether an employee is present in his/her cabin.
  - As there is connection established between Door module and security module, he can also see the images of employee who passed through security module.



4.2 Indoor Module

- It comprises of ESP8266 module, PIR sensor and LED.
- It is connected to Door module through MQTT connection.
- It publishes to the topic EMP_AVAILABILITY and subscribes to the topic EMP_AUTH to the broker.
- Whenever an employee enters through door, Raspberry Pi sends message to the ESP Module deployed in his/her respective cabin and the indoor module starts functioning.

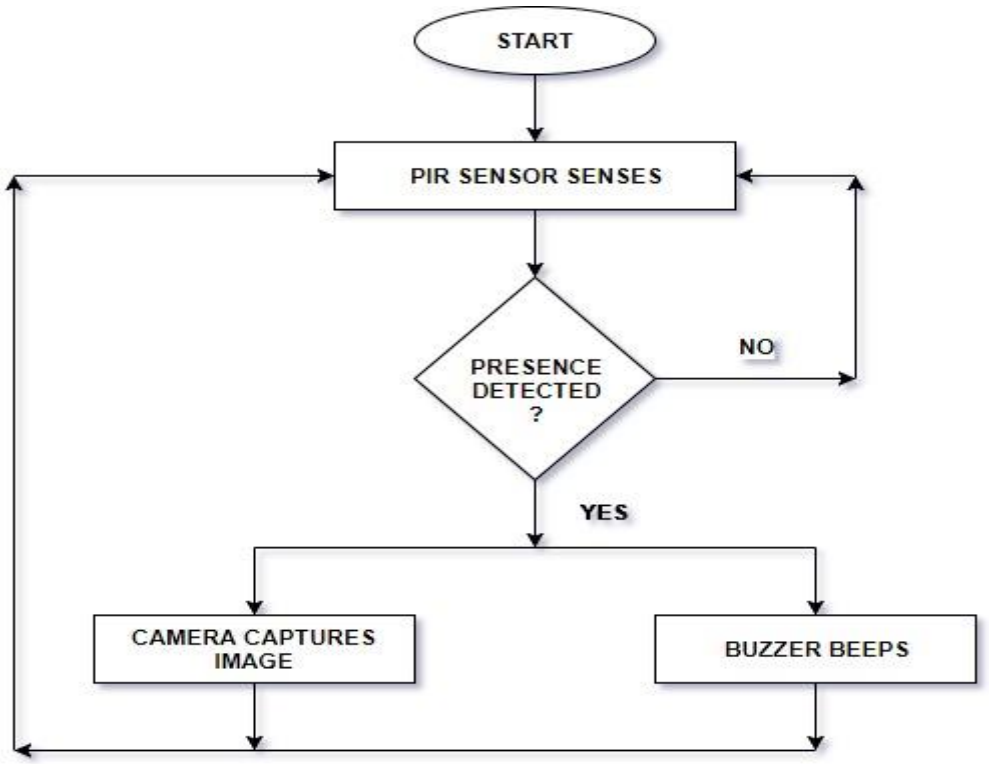


- Indoor module start functioning means the PIR sensor start sensing the presence of employee and the moment it detects the presence, it turns on the LED.
- The PIR sensor continuously detects the presence, if it detects that employee is not present in the cabin for 20 seconds, it automatically turn OFF the LED.

- For turning ON the LED, it senses for continuous 10 seconds. This is because if someone enters and leaves the cabin immediately, then LED will not glow and thus will save power.
- In both the cases whether employee is present or absent, ESP8266 module publishes message to Raspberry Pi which in turn update the database and the information on application.
- Here for demo purpose, I have used LED. But we can connect Relay, which can be used to control the bulbs and fan in the cabin.

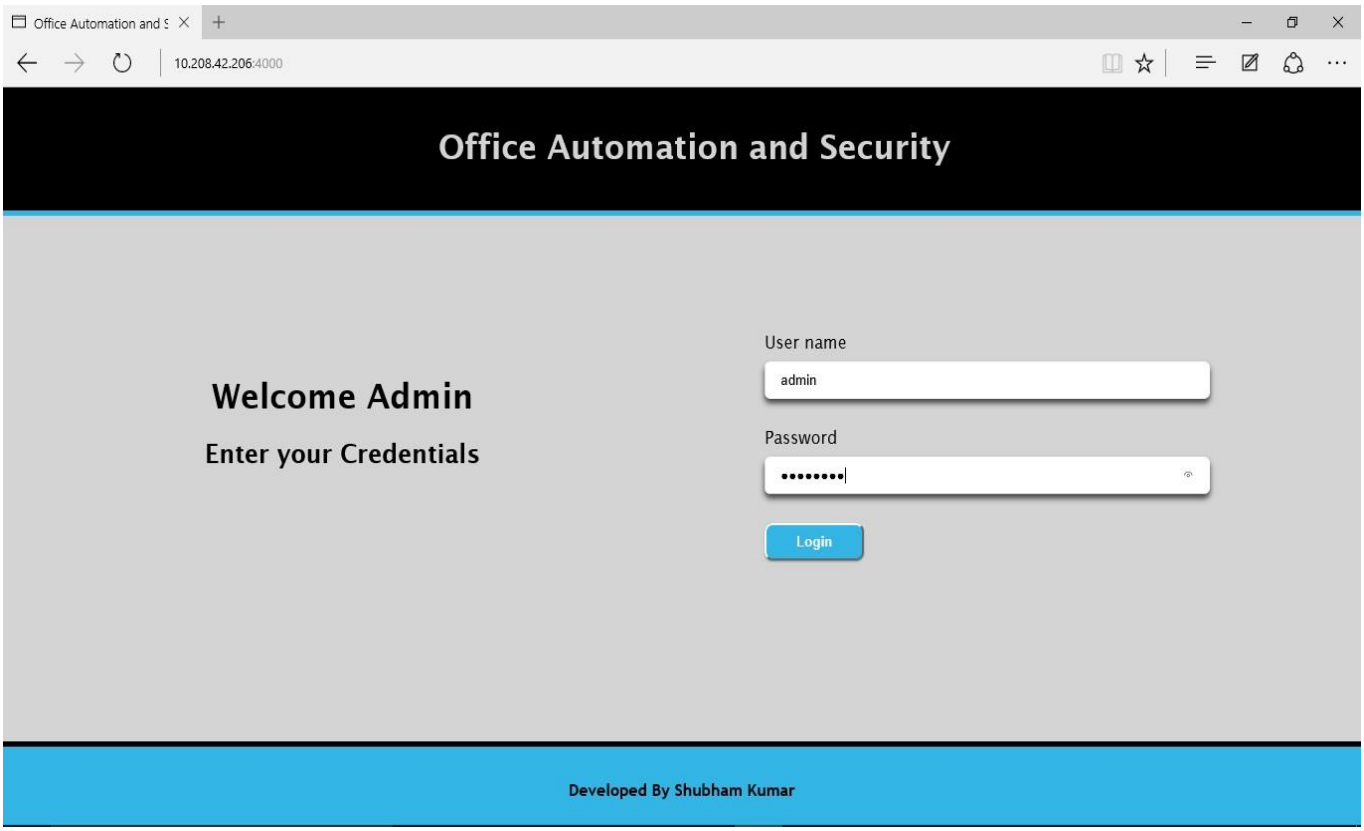
4.3 Security Module

- It comprises of Raspberry Pi, PIR sensor, Camera and buzzer.
- It is connected to Door module through HTTP connection.
- This module generally represents an area in office, where accessibility is given to certain people and is highly confidential or secure.
- Here PIR sensor continuously detects any movement and if it detects any presence, it captures image of that person and will beep buzzer indicating presence of intruder.
- An express based application is built to show all images captured and can be accessed through the gateway by HTTP connection.

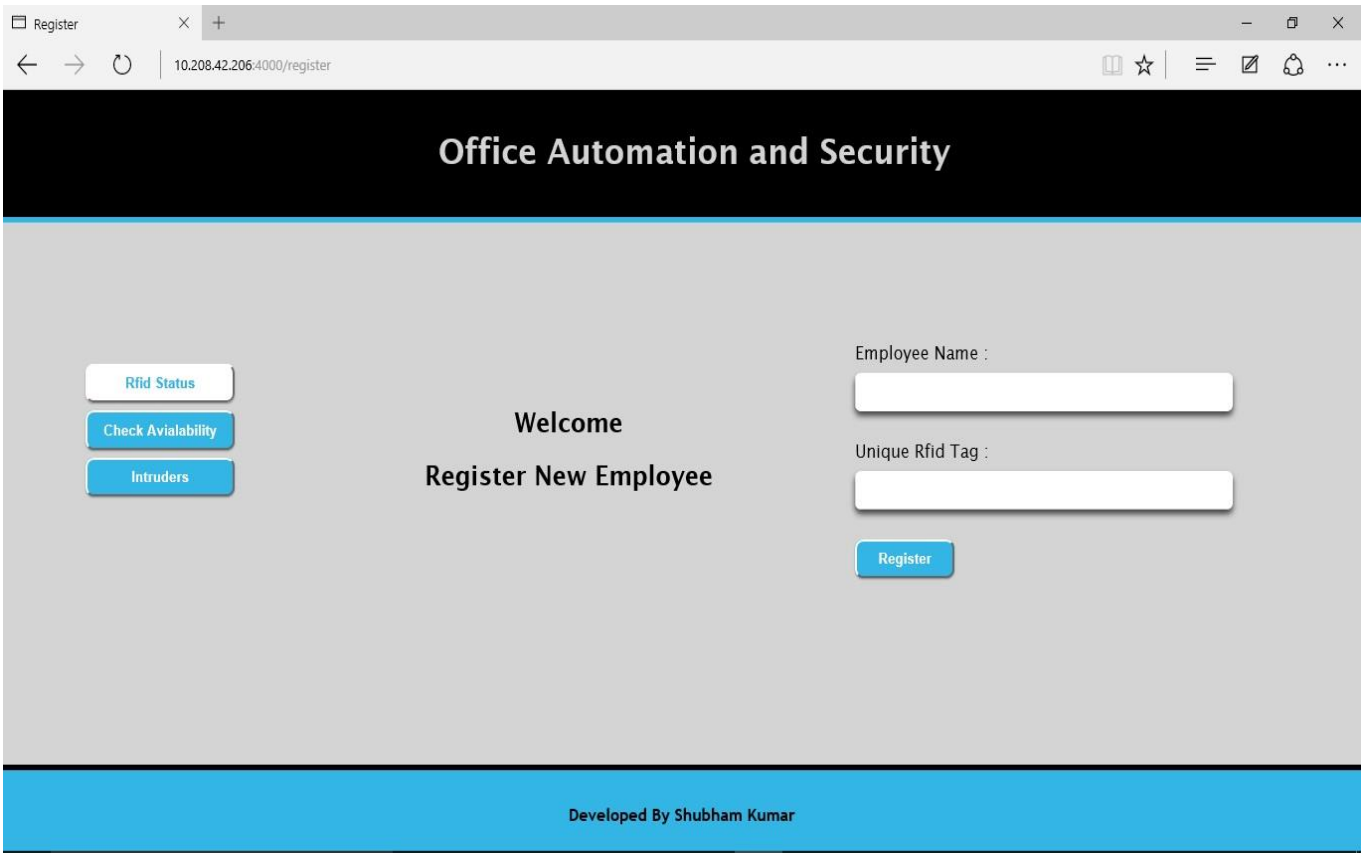


5 Web Application

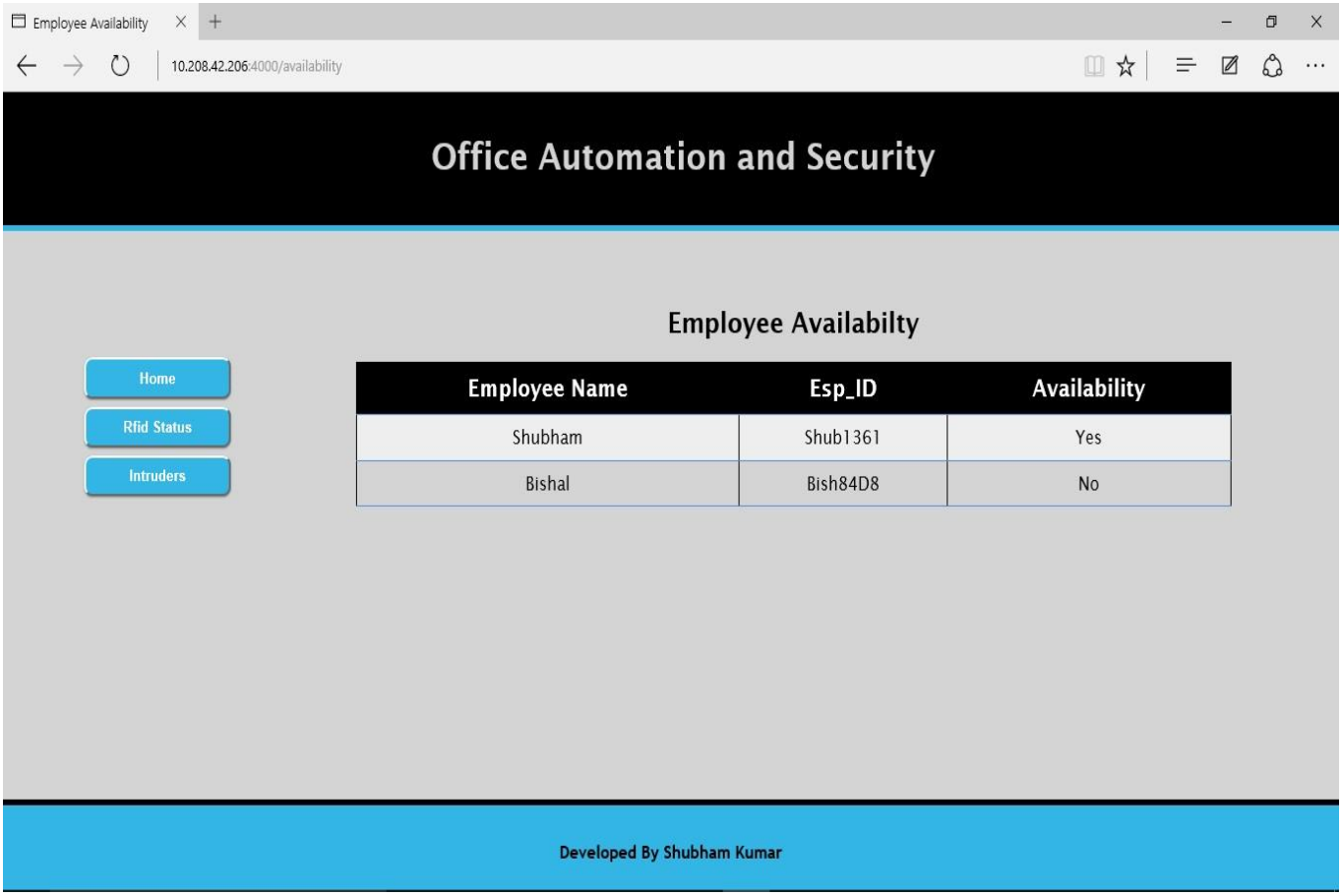
Index Page – Admin Login



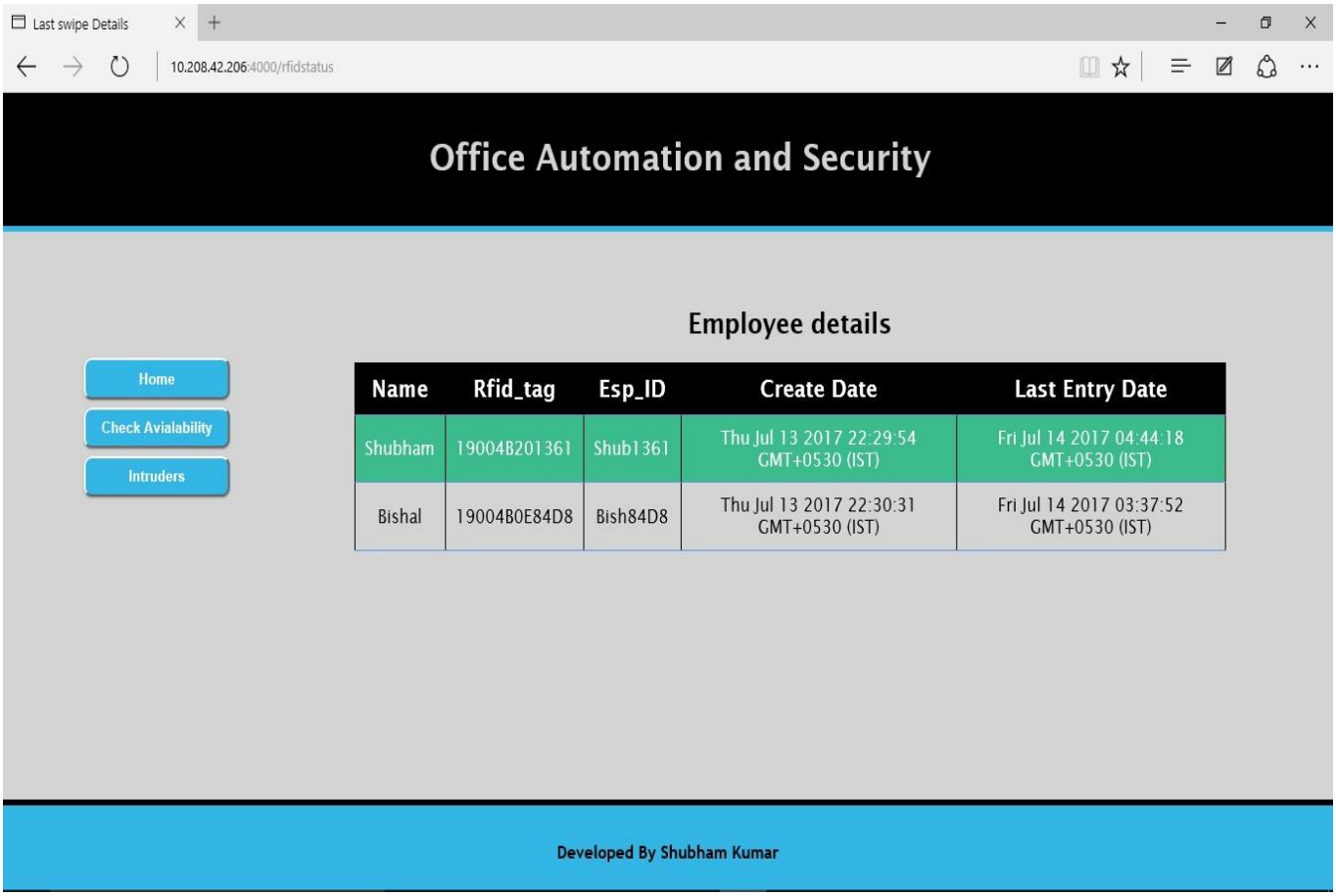
Web Page for registering new Employee



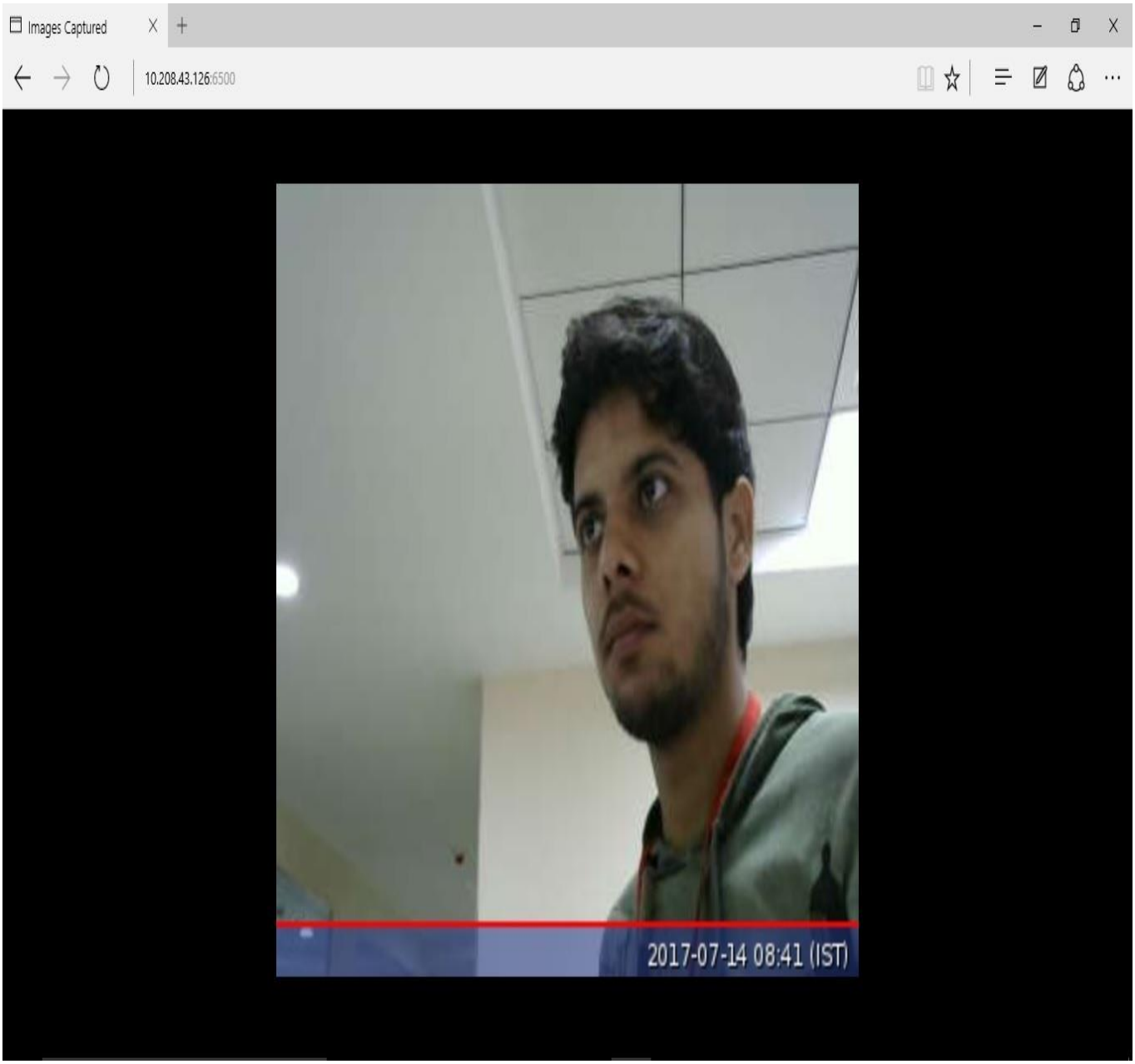
Web Page showing Employee Availability



Web Page showing Employee Details and their Last entry



Web Page showing Intruder's Image



6 Conclusion and Future Recommendations

It was a great learning experience. I have been introduced with concepts of Internet of Things and automation. I started from scratch and now I have a sound knowledge of everything I did in this internship. I gained knowledge of various sensors and how to program them. After completing this project, I have realized how important is to understand the hardware on which you are going to run your code and how this amazing concept of automation can ease out our daily life.

After completion of this project, I can conclude that these type of small yet effective models can be implemented on large scale in commercial buildings and can be enhanced according to need.

- The door module can be implemented almost everywhere like home, schools, hospitals.
- The security module can also be implemented in places like banks and home where jewellery and money are being kept.
- The indoor module can be used on large scale in commercial buildings for conserving power.

We know energy resources are very limited to us and we are continuously exploiting it. There are many good automation products like NEST's *Thermostat which is doing a great job of saving power. In this project, indoor module is automatically turning off all the electrical components attached to it, if no one is present in the cabin at that time. Thus its intention is to conserve power and I am sure that if this is implemented on a large scale it will give good results.

Due to time constraint of two months I could not enhance this project further but for future recommendations and scope, I would suggest these points:

- This project can be extended on large scale i.e. in real life scenario.
- There are certain features which should be enhanced like security in different communication between modules.
- Privacy is major concern of people now-a-days, certain ways should be developed to meet the requirements of security and privacy of data that is being transferred from sensor to gateway and between different modules.
- Also we can include features like voice controlled automation and we can build android application for mobiles.
- Certain standards can be designed for all type of working and for future enhancements of this project.

*Refer to https://nest.com/thermostat/meet-nest-thermostat/

7 References

1. https://www.wikipedia.org/
2. https://www.youtube.com/
3. https://www.raspberrypi.org/
4. http://www.rfwireless-world.com/IoT/IoT-architecture.html
5. http://robokits.co.in/wireless-solutions/rfid/rfid-reader-module-with-antenna-usb-uart
6. https://manpages.debian.org/jessie/fswebcam/fswebcam.1.en.html
7. http://wiki.seeed.cc/Grove-Servo/
8. https://www.baldengineer.com/mqtt-tutorial.html
9. http://internetofthingsagenda.techtarget.com/definition/MQTT-MQ-Telemetry-Transport
10. http://www.instructables.com/id/How-to-Use-MQTT-With-the-Raspberry-Pi-and-ESP8266/
11. https://blog.risingstack.com/getting-started-with-nodejs-and-mqtt/
12. https://www.npmjs.com/
13. https://docs.mongodb.com/v3.0/core/crud-introduction/
14. https://zellwk.com/blog/crud-express-mongodb/

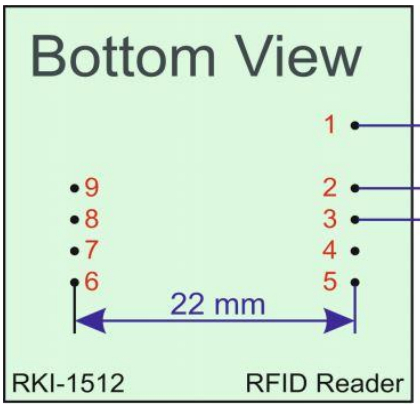
8 Annexures

8.1 Pin configurations of Hardware Used

Raspberry Pi 3B GPIO Pins

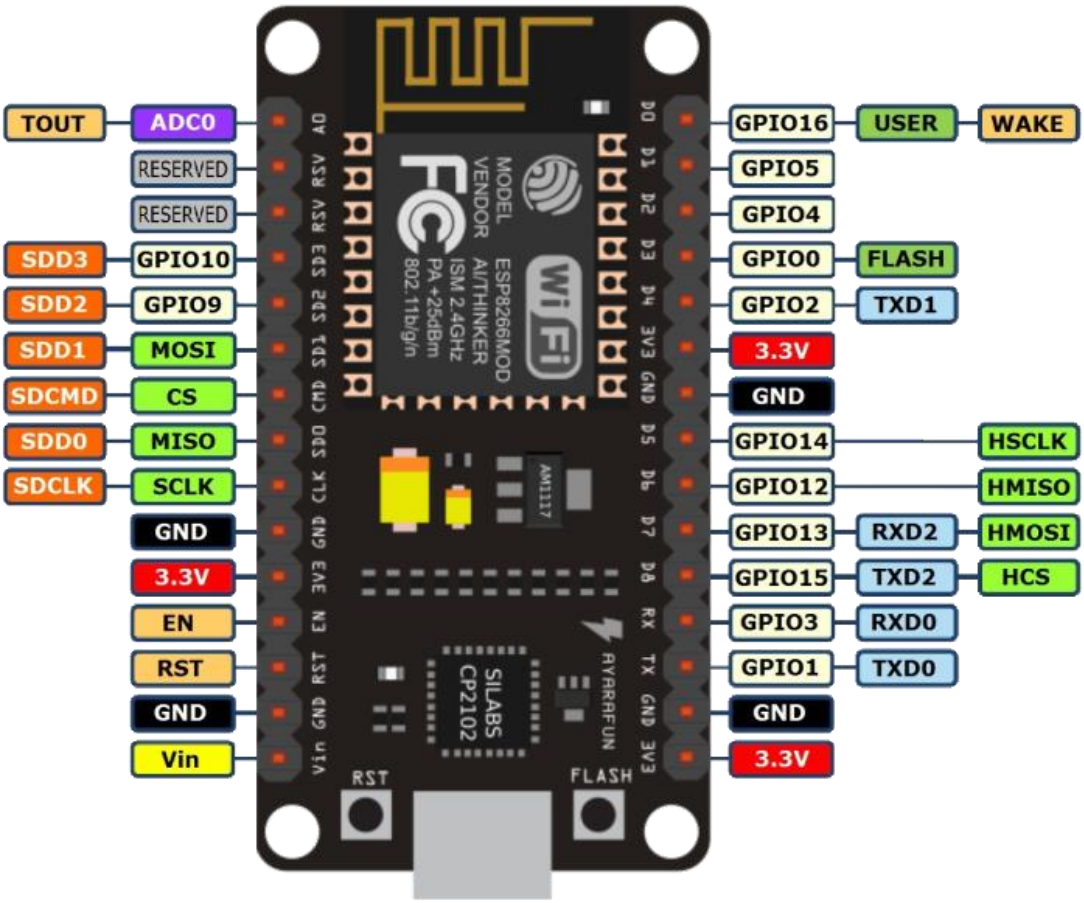
Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I2C)		DC Power 5v	04
05	GPIO03 (SCL1 , I2C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I2C ID EEPROM)		(I2C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

RFID Reader Pin out Bottom view



1	VCC	5V
2	GND	GND
3	BEEP	BEEP AND LED
4	NC	NOT CONNECTED
5	NC	NOT CONNECTED
6	SEL	HIGH IS UART,LOW IS WEIGAND
7	TX	UART TX
8	D1	WEIGAND DATA 1 (Optional)
9	D0	WEIGAND DATA 0 (Optional)

NodeMCU and ESP8266 Pin Out



NO.	Pin Name	Function
1	RST	Reset the module
2	ADC	A/D Conversion result.Input voltage range 0-1v,scope:0-1024
3	EN	Chip enable pin.Active high
4	IO16	GPIO16; can be used to wake up the chipset from deep sleep mode.
5	IO14	GPIO14; HSPI_CLK
6	IO12	GPIO12; HSPI_MISO
7	IO13	GPIO13; HSPI_MOSI; UART0_CTS
8	VCC	3.3V power supply (VDD)
9	CS0	Chip selection
10	MISO	Salve output Main input