**M S Ramaiah Institute of Technology**

(An Autonomous Institute, Affiliated to VTU)

MSR nagar, MSRIT post, Bangalore-54

A Dissertation Report on

**Human and Energy Resource Management**

Submitted by

Tejas D. Hasarali 1MS12CS120

Vikas H. 1MS12CS129

*In partial fulfillment for the award of the degree of*

# *Bachelor of Engineering in Computer Science & Engineering*



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**M.S.RAMAIAH INSTITUTE OF TECHNOLOGY**

**(Autonomous Institute, Affiliated to VTU)**

**BANGALORE-560054**

www.msrit.edu, **May 2015**

#### Contents

***Declaration i***

***Acknowledgements ii***

***Abstract iii***

1 **INTRODUCTION 7**

* 1. General Introduction
  2. Statement of the Problem.
  3. Objectives of the project
  4. Project deliverables
  5. Current Scope
  6. Future Scope.

1. **PROJECT ORGANIZATION 13**
   1. Software Process Models
   2. Roles and Responsibilities
2. **LITERATURE SURVEY 16**

3.1 Introduction

3.2 Main Body

3.3 Conclusion of Survey

1. **SOFTWARE REQUIREMENT SPECIFICATIONS 19**

4.1 Product Overview

4.2 External Interface Requirements

4.2.1 User Interfaces

4.2.2 Hardware Interfaces

4.2.3 Software Interfaces

4.2.4 Communication Interfaces

4.3 Functional Requirements

4.3.1 Functional Requirement 1.1

:

4.3.n Functional Requirement 1.n

4.4 Software System Attributes

4.4.1 Reliability

4.4.2 Availability

4.4.3 Security

4.4.4 Portability

4.4.5 Maintainability

4.4.6 Performance

4.5 Performance Requirements

4.6 Database Requirement

4.7 Design Constraints

4.8 Other Requirements

1. **DESIGN 26**
   1. Introduction
   2. Architecture Design
   3. Graphical User Interface
   4. Class Diagram and Classes (represent Inheritance, Aggregation and Association)
   5. Sequence Diagram
   6. Data flow diagram
   7. Metric calculation
2. **IMPLEMENTATION 39**
   1. Tools Introduction
   2. Technology Introduction
   3. Overall view of the project in terms of implementation
   4. Explanation of Algorithm and how it is been implemented
   5. Information about the implementation of Modules
3. **TESTING 47**

**7.1** Results and Snapshots

1. **CONCLUSION & SCOPE FOR FUTURE WORK 53**
2. **REFERENCES 54**

**DECLARATION**

We hereby declare that the project entitled “Human and Energy Resource Management” which is submitted for the subject "**Project Based Learning**" is our original work and the project has formed the basis in thepartial fulfillment for the award of the degree of Bachelor of Engineering in Computer Science & Engineering

Signature of the Students:

Place: Bengaluru

Date: 11-12-2015

**ACKNOWLEDGEMENT**

First and foremost we would like to thank Mrs. Sowmya B. J, Assistant Professor, Department of Computer Science and Engineering, M. S. Ramaiah Institute of technology, Bangalore, who was also our supervisor, for their consistent guidance on each and every step of our project and our HOD Dr. K.G.Srinivasa, Professor. Also we show our gratitude to all the teaching and non teaching staff of Computer science engineering, M.S Ramaiah Institute of technology. Without their push and motivation, this project would have not been completed. Their continuous support and help made this project feasible. Without their support we would not have been able to excel in our project. Lastly, we would like to thanks all our friends and teammates for believing in each other and our capabilities.

# ABSTRACT

Energy and Human Resources are not efficiently managed in most organizations thusly, our project aims at managing these resources by making energy consuming equipment automated and delineated human resource monitoring. Human and Energy resources management refers to the use of computer and [information technology](https://en.wikipedia.org/wiki/Information_technology) to control home appliances and features (such as windows or lighting). Systems can range from simple remote control of lighting through to complex computer/micro-controller based networks with varying degrees of intelligence and automation. Our system is adopted for reasons of ease, security and energy efficiency. In modern construction in industrialized nations, most homes have been wired for electrical power, telephones, TV outlets (cable or antenna), and a [doorbell](https://en.wikipedia.org/wiki/Doorbell). Many household tasks were automated by the development of specialized automated appliances. For instance, automatic [washing machines](https://en.wikipedia.org/wiki/Washing_machine) were developed to reduce the manual labor of cleaning clothes, and [water heaters](https://en.wikipedia.org/wiki/Water_heater) reduced the labor necessary for [bathing](https://en.wikipedia.org/wiki/Bathing). The use of gaseous or [liquid fuels](https://en.wikipedia.org/wiki/Liquid_fuels), and later the use of electricity enabled increased automation in heating, reducing the labor necessary to manually refuel [heaters](https://en.wikipedia.org/wiki/Stove) and [stoves](https://en.wikipedia.org/wiki/Kitchen_stove). Development of [thermostats](https://en.wikipedia.org/wiki/Thermostat) allowed more automated control of heating, and later cooling.

1. **INTRODUCTION**
   1. **GENERAL INTRODUCTION**

The Raspberry Pi is a series of credit card–sized single-board computers developed in the UK by the Raspberry Pi Foundation with the intention of promoting the teaching of basic computer science in schools.Raspberrypi2 uses the Broadcom BCM2836 as the Soc which contains a quad-core ARM Cortex-a7 processor with floating point & NEON, running at 900MHz and a Video core 4 GPU. The GPU is capable of Blu-Ray-quality playback, using H.264 at 40MBits/s. It has 1 GB of RAM and 4 USB 2.0 ports and one micro USB for 5v power supply. It has HDMI video output and persistent storage is provided by the micro SD card that can be inserted into its respective slot. It is a portable little computer which can be used in electronic projects, and for many of the things that your desktop PC does, like spreadsheets, word-processing and games. It has an IEEE 802.5 standard Ethernet card and also MAC – Media Access Control which is being controlled by a chip called as the ASIC – Application Specific Integrated Circuit. Internet communication is possible with the help of Ethernet card at approximately 100MB/s.

The Raspberry Pi hardware has evolved through several versions that feature variations in memory capacity, and peripheral device support.

This block diagram depicts models *A*, *B*, *A+*, and *B+*. Model *A* and *A+* and *Zero* lack the Ethernet and USB hub components. The Ethernet adapter is connected to an additional USB port. In model *A* and *A+* the USB port is connected directly to the SoC. On model *B+* the chip contains a five-point USB hub, of which four ports are available, while model *B* only provides two. On the model *Zero*, the USB port is also connected directly to the SoC, but it uses a micro USB (OTG) port.

It is essential to be well versed with functions of various components before plunging into other details regarding the system. The functions of the various components that are used in developing the system are given below.

1. IR Proximity Sensor:

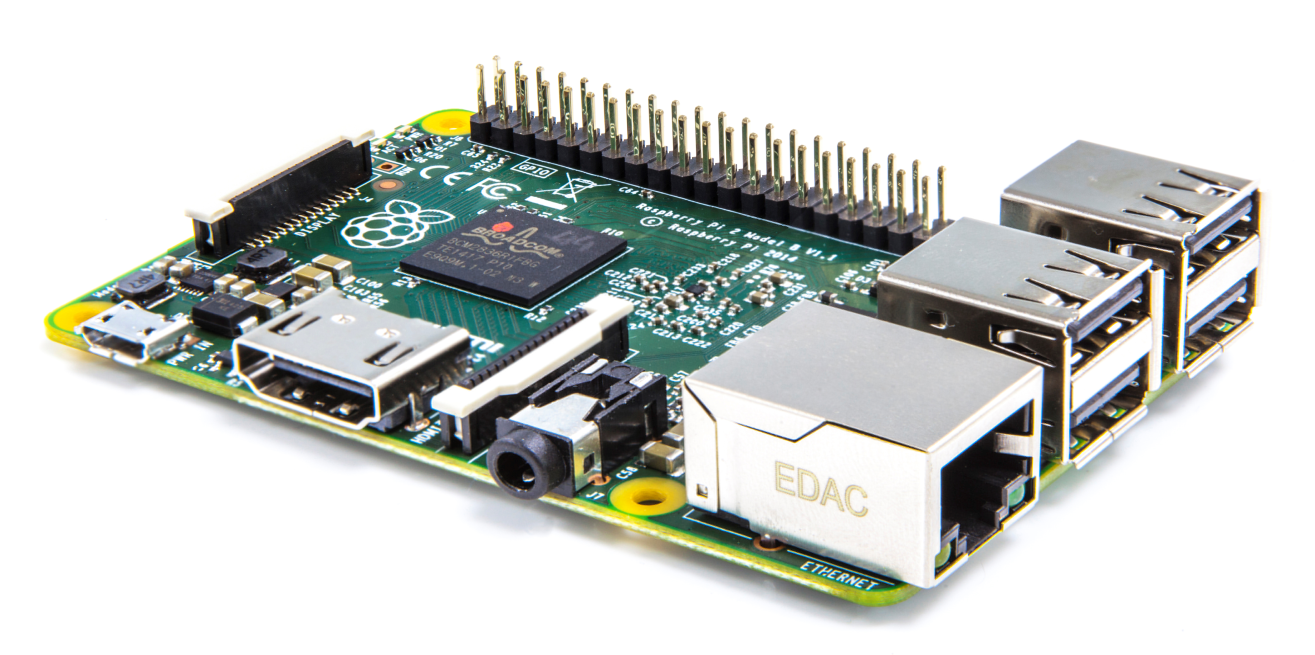
It works by applying a voltage to a pair of IR light emitting diodes (LED’s) which in turn, emit infrared light. This light propagates through the air and once it hits an object it is reflected back towards the sensor. The sensing unit, in the form of an integrated circuit (IC), detects the reflected infrared light and sends a corresponding signal to the output terminal.

1. Temperature Sensor:

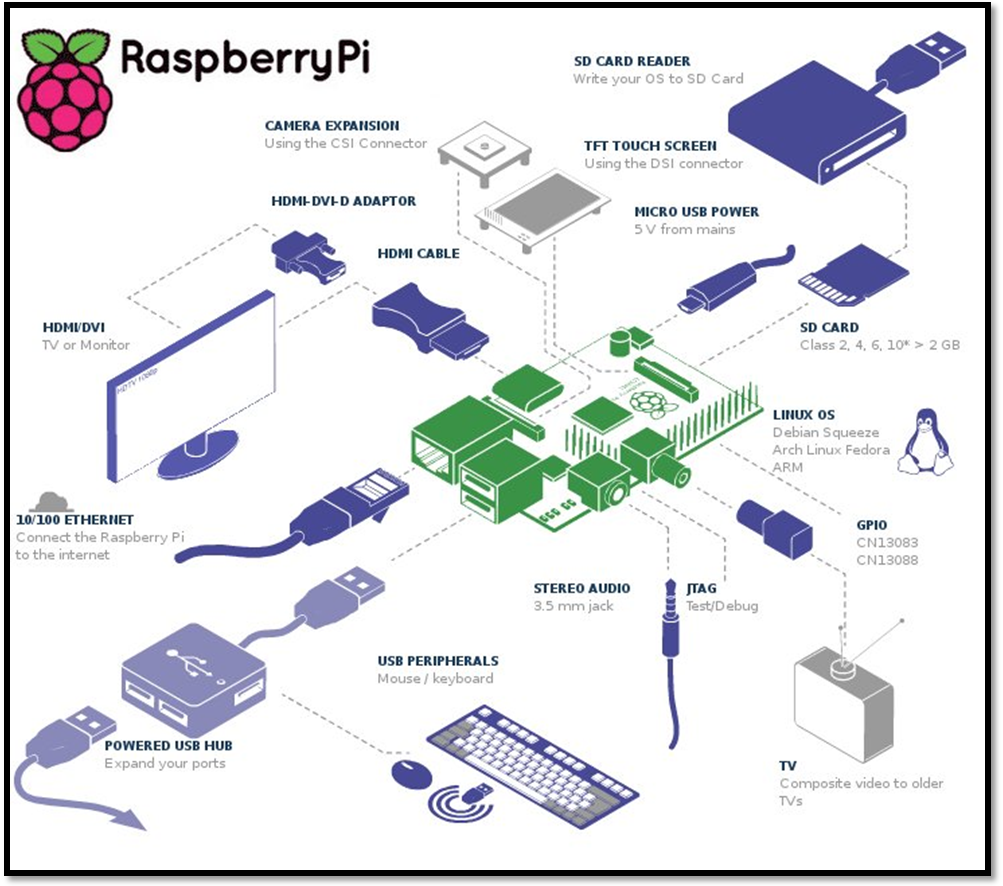
It can be used to detect the changes in the surrounding temperature with the help of temperature dependent variable resistors. The output current inversely depends on the temperature sensed by the sensor.

1. Pressure Sensor

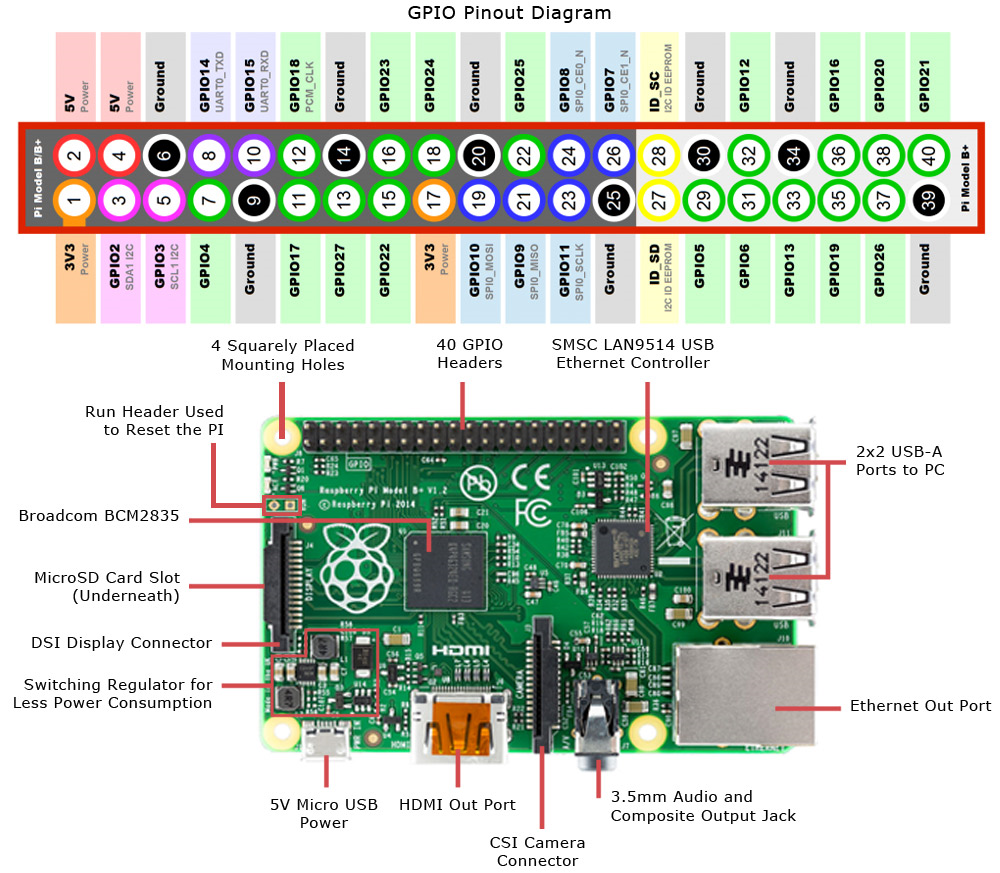
In an absolute Pressure Sensor, pressure is measured relative to perfect vacuum condition. Perfect vacuum is a condition where there is no matter present in the atmosphere and hence, nil air pressure exists in that region. Sensors based on absolute pressure measurement require strict specifications for precise outputs. Sensors based on this type of measurement are used in barometric or altitude related pressure measurements.



Raspberry Pi2

****

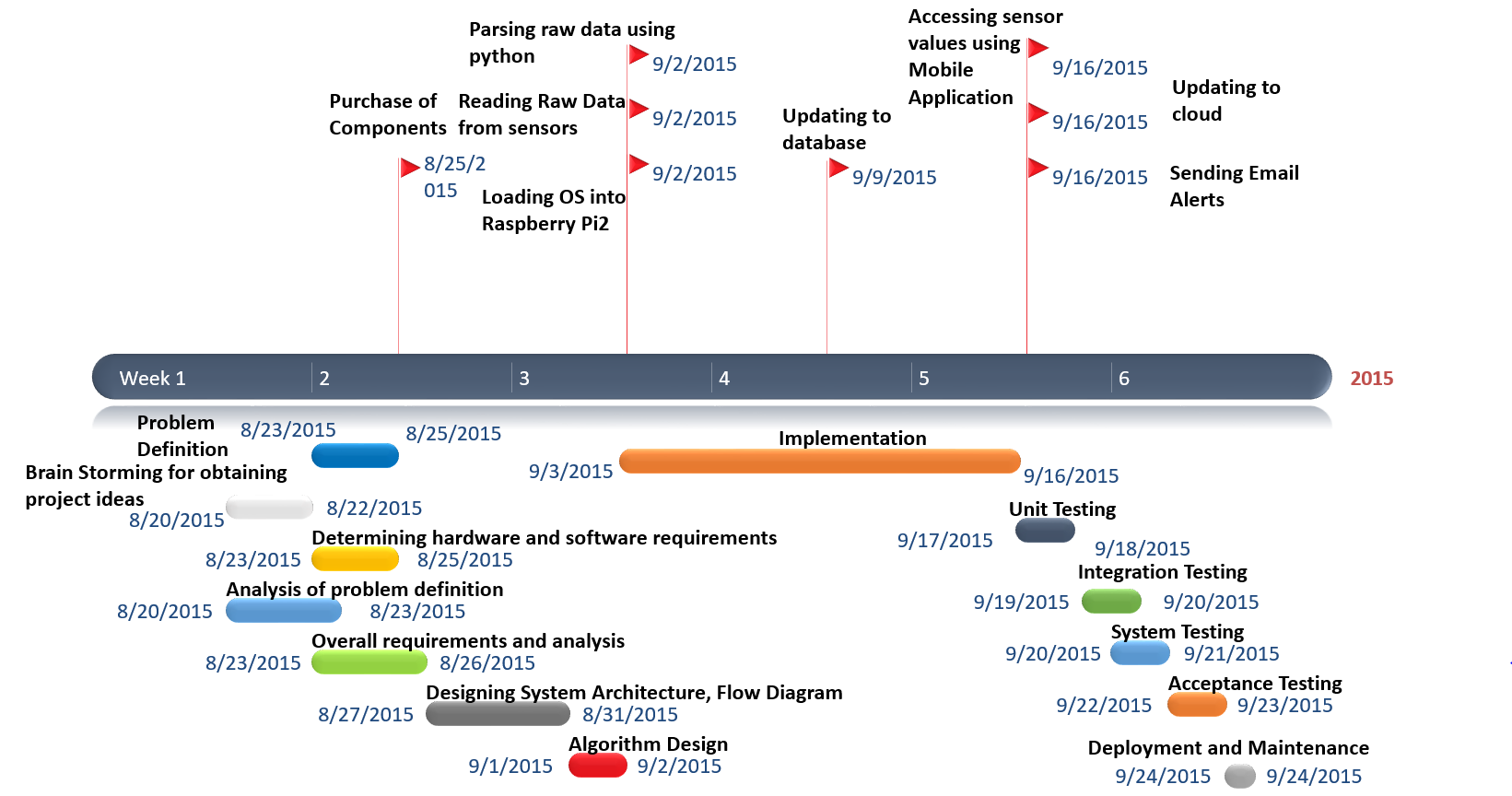
General Interfaces

****

GPIO Pin out diagram

* 1. **PROBLEM STATEMENT**
* Energy and Human Resources are not efficiently managed in most organizations thusly, our project aims at managing these resources by making energy consuming equipment automated and delineated human resource monitoring.
  1. **OBJECTIVES OF THE PROJECT**

1. The objective of the project is conservation of energy which is taken for granted in urban areas since the paucity is not yet felt by them.
2. Another important goal is to easily have a check on the number of hours an employee works in his day of work. The immediate effect would be to identify and manage the workaholics and remiss workers in the institution.
3. To complete the project in time and considering all the economic criteria for an engineering project.
4. To meet all the requirements of the problem statement.
5. For the team members to work in synergy in the team as a single output oriented entity.

****

Project Timeline

* 1. **CURRENT SCOPE**

It is important to conserve energy because the natural resources that are essential for production of these energy are being depleted faster than they can be regenerated. This problem can be minimised by automating resource usage using our proposed model.The man force in a particular company though eminent in their fields are still humans. Workaholics need to be managed by HR department and our project does exactly that by informing authorities when an employee exceeds his maximum working hours per week. Contrary to that, there will be employees who spend time in the campus without working. Our model gives statistics on the number of hours an employee has worked

* 1. **FUTURE SCOPE**

The objective of the project is conservation of energy which is taken for granted in urban areas since the paucity is not yet felt by them. Another important goal is to easily have a check on the number of hours an employee works in his day of work. The immediate effect would be to identify and manage the workaholics and remiss workers in the institution. To complete the project in time and considering all the economic criteria for an engineering project. To meet all the requirements of the problem statement. For the team members to work in synergy in the team as a single output oriented entity.

**2. PROJECT ORANIZATION**

**2.1 SOFTWARE PROCESS MODELS**

Extreme programming is the best suited software development process model for wireless temperature monitoring system.

Extreme programming (XP) is a software development methodology which is intended to improve software quality and responsiveness to changing customer requirements. As a type of agile software development it advocates frequent "releases" in short development cycles, which is intended to improve productivity and introduce checkpoints at which new customer requirements can be adopted.

Other elements of extreme programming include: programming in pairs or doing extensive code review, unit testing of all code, avoiding programming of features until they are actually needed, a flat management structure, simplicity and clarity in code, expecting changes in the customer's requirements as time passes and the problem is better understood, and frequent communication with the customer and among programmers. The methodology takes its name from the idea that the beneficial elements of traditional software engineering practices are taken to "extreme" levels. As an example, code reviews are considered a beneficial practice; taken to the extreme, code can be reviewed *continuously*, i.e. the practice of pair programming.

XP describes four basic activities that are performed within the software development process: coding, testing, listening, and designing. Each of those activities is described below.

#### Coding

The advocates of XP argue that the only truly important product of the system development process is code – software instructions that a computer can interpret. Without code, there is no working product.

Coding can also be used to figure out the most suitable solution. Coding can also help to communicate thoughts about programming problems. A programmer dealing with a complex programming problem, or finding it hard to explain the solution to fellow programmers, might code it in a simplified manner and use the code to demonstrate what he or she means. Code, say the proponents of this position, is always clear and concise and cannot be interpreted in more than one way. Other programmers can give feedback on this code by also coding their thoughts.

#### Testing

Extreme programming's approach is that if a little testing can eliminate a few flaws, a lot of testing can eliminate many more flaws.

* Unit tests determine whether a given feature works as intended. A programmer writes as many automated tests as they can think of that might "break" the code; if all tests run successfully, then the coding is complete. Every piece of code that is written is tested before moving on to the next feature.
* Acceptance tests verify that the requirements as understood by the programmers satisfy the customer's actual requirements.

System-wide integration testing was encouraged, initially, as a daily end-of-day activity, for early detection of incompatible interfaces, to reconnect before the separate sections diverged widely from coherent functionality. However, system-wide integration testing has been reduced, to weekly, or less often, depending on the stability of the overall interfaces in the system.

#### Listening

Programmers must listen to what the customers need the system to do, what "business logic" is needed. They must understand these needs well enough to give the customer feedback about the technical aspects of how the problem might be solved, or cannot be solved. Communication between the customer and programmer is further addressed in the *planning game*.

#### Designing

From the point of view of simplicity, of course one could say that system development doesn't need more than coding, testing and listening. If those activities are performed well, the result should always be a system that works. In practice, this will not work. One can come a long way without designing but at a given time one will get stuck. The system becomes too complex and the dependencies within the system cease to be clear. One can avoid this by creating a design structure that organizes the logic in the system. Good design will avoid lots of dependencies within a system; this means that changing one part of the system will not affect other parts of the system.

**2.2 ROLES AND RESPONSIBILITIES**

Pair programming is the main theme of the efforts put in this project.

The main responsibilities are:

1. Hardware connections, sensor raw data collection and processing.
2. Creating the local and cloud databases. (Tejas inclined)
3. Mobile app development. (Vikas inclined)
4. Creation of Broker instance in cloud.
5. Intimation system, particularly the use of google mailbox to send e-mails.

**3. LITERATURE SURVEY**

**3.1 INTRODUCTION**

People Counter Module – The module constitutes of detecting direction of movement using two IR sensors to increment or decrement the count of number of people in the room. Time Calculation – The module qualifies the work output of employees using a pressure and heat detector. The occupancy of a cubicle chair implies some weight and heat is present on it. Thus this module calculates time by taking input from the sensor. MQTT communication - Enable the Raspberry Pi and mobile devices to send and receive small messages and data to and from each other. Cloud storage Module – Real time Data storage and access through the cloud. Local DB Module – Provide the local database storage for the RTO database and stores the image in binary format.

**3.2 MAIN BODY**

RaspberryPi is a mini computer which is of the size of a credit card. The operating system is called Raspbian OS which is simple and is optimized for Raspberry pi. It’s an open source operating system based on Debian. Once the operating system has been loaded in the Raspberry Pi using the SD card which is of class 10 or higher the necessary hardware connections are made. It is essential to be well versed with functions of various components before plunging into other details regarding the system. The functions of the various components that are used in developing the system are given below.

Two DS18B20 sensors are used in this project. This semiconductor temperature sensor offers high accuracy and high linearity over an operating range of about 55 degree centigrade to 150 degree centigrade. It is an integrated circuit, and can therefore include extensive signal processing circuitry within the same package as the sensor thus eliminating additional circuitries. This sensor can be directly connected to the Raspberry pi’s GPIO pin and thus capable of direct and reliable communication with it. Other features are that it’s lightweight and has a wide power supply range (2.7 V to 5.5 V) and converts the temperature to a digital value in less than 1second.

The Raspberry Pi camera module can be used to take high-definition video, as well as stills photographs. So the Raspberry pi camera module takes pictures of the surroundings when the temperature exceeds the user defined value.

The various components of Raspberry- Pi are:

• SD Card slot is used for inserting an SD card which of class 10 or higher to load the raspbian OS and to provide persistent storage.

• Micro USB Power Port enables the Raspberry pi to obtain current that’s needed for booting OS and running various applications. It usually requires a 5volt ,1amp charger.

• Ethernet Port is used to connect to the Internet using RJ45 lan cable.

• HDMI OUT(High Definition Multimedia Interface) is used with HDTVs and monitors with HDMI input.If not and HDMI-VGA adapter is used to make it compatible with VGA monitors. This is needed only during the initial stage of loading the OS.

• BROADCOM BCM 2835: Raspberrypi2 uses the Broadcom BCM2836 as the Soc which contains a quad-core ARM Cortex-a7 processor with floating point & NEON, running at 900MHz and a Video core 4 GPU. The GPU is capable of Blu-Ray-quality playback, using H.264 at 40MBits/s. It has 1 GB of RAM and 4 USB 2.0 ports and one micro USB for 5v power supply [3].

• GPIO pins allows us to control and interact with real world .

* Various software applications are used namely
* SD card Formatter to initially erase all the data on the SD card.
* Angry IP scanner to detect the IP addresses of all devices connected to a Wi-Fi network.
* Putty is a SSH telnet software which is used to remotely access the Raspbian OS’s terminal.
* VNC viewer is used to access the virtualized Raspbian OS.

Python is used for developing and writing the code to remotely access the raw temperature and parse it to obtain the actual temperature and also to send Gmail Alerts along with pictures of the surroundings if the temperature exceeds the threshold value.

* SQLite database for offline persistent storage.
* Apache Apollo broker service
* Android Studio Android application..

**3.3 CONCLUSION OF SURVEY**

Thus all the software applications and technology needed to start implementing the project have been thoroughly studied. Further the various requirements of the project have been assessed. The various functionalities of Raspberrypi2 are also studied. The literature survey has been successfully completed and now the software requirements specification needs to be done and later design has to be started and finally finish with implementation, deployment and testing.

**External Interface Requirements**

1. **User Interfaces**: Dropbox is used as a cloud service interface, which is customized for Raspberry Pi 2. At any given time a Manager should be able to check the number of employees inside the office and the number of employees who are sitting at their desk and working. At the end of the day the manger should be sent a mail containing information about total number of hours worked by every employee. If any employee’s number of working hours exceeds a certain limit per week then the Human Resource Department as well as the Respective Manger should be informed. The interface should be able to accept required inputs, it should be simple and easy to use. The user should be able to access the above information through any device.
2. **Hardware Interfaces**: Raspberry Pi 2 Model B should be the hardware device which is responsible for co-coordinating various activities of different components of the project. Infrared proximity sensors has to be used to keep track the number of employees inside the office. Pressure and Temperature sensor has to be used to keep track the number of hours worked by every employee. RaspberryPi2 along with three different sensors should acts as the hardware interface in the project.
3. **Software Interfaces**: Python has to be used as programming language to keep track the number of people inside the office with the help of IR proximity sensor. Using this data the energy consuming equipment should be controlled. Also to keep track the number of hours worked by every employee with the help of Temperature and Pressure sensors python has to be used. At the end of the day number of hours worked by every employee should be sent to the Manger using SMTP protocol. If number of hours worked by any employee exceeds a predefined hours per week then the Human Resource Department as well as Manger should be informed. Thus the python together with the library routines should act as the software interface.
4. **Communication Interfaces**: A Manager should be able to access the required information through internet using the cloud service or by using the android mobile application. SMTP should be the protocol used for sending the required information through e-mail and MQTT as a communication protocol between the hand held devices and the could services as well as between the Raspberry Pi and could services. Hence MQTT and SMTP are used as communication protocol between different components.

**Functional Requirements**

1. **Functional Requirement 1.1**: Manger Allocation: A manager should be allocated to a set of employees and all the information related to these employees should be sent to this manager.

1. **Functional Requirement 1.2**: Accessing Raspberry Pi Remotely: The user should be able to access the Raspberry Pi interface remotely to change the functionalities remotely by using Putty and VNC viewer.
2. **Functional Requirement 1.3:** Cabin Resource monitor: With the help of the data collected from Temperature and Pressure sensor the resources inside the cabin of every employee should be controlled.

1. **Functional Requirement 1.4:** Office resource monitor: Using the data collected from the IR proximity sensor the number employees inside the office is kept track and depending on the number of employees inside the building the resources hast to be controlled.
2. **Functional Requirement 1.5:** Human Resource monitor: Using the data obtained from Temperature and Pressure sensors the number hours worked by each employee is kept track and if any employees working hours exceed a pre-defined limit then alert message should be sent to the respective Manger and the Human Resource Department.
3. **Functional Requirement 1.6:** Could service Interface: Manger and Human Resource Department should be able to access the employee information through the cloud interface and should be able to make necessary changes.
4. **Functional Requirement 1.7:** Android Application: An android application has to be developed to obtain the information through cloud services and to get alert messages.
5. **Functional Requirement 1.8:**  Storage: The number of hours worked by every employee has to be logged in the storage provided. Interface to access this information has to be created in a simple manner.
6. **Functional Requirement 1.9:** MQTT: Manager and Human Resource Department has to be informed and alerted through the android application using MQTT protocol
7. **Functional Requirement 1.10:** SMTP: Manager and Human Resource Department has to be informed and alerted through Email using SMTP protocol.

**Software System Attributes**

1. **Reliability**: The system should be free of errors and it should be able to handle the worst case scenarios. The permissible rate of system crash per month should be less than one.

1. **Availability**: The system should be available all the time and the maintenance has to be done only at night when the load is minimum. The system should be able to respond to changes in the state of sensors and log those data.
2. **Security**: The system should be safe and secure, should be able to resist the hacking attacks. The employee data log should be very secure and only authorized persons can be able to access it. Also these data should be Read Only.
3. **Portability**: The system should be able to work in different operating systems without or minimal changes. Changing the platform on which the system works should be a very easy task.
4. **Maintainability:** The system needs to be continuously monitored and maintained. If the system goes down by any failures then it should be repaired immediately. If any changes are required in the system it should be provided by updates.
5. **Performance**: The system’s performance is measured in terms of responsiveness, efficiency and user experience. A system with high performance is desired. Performance is high if the throughput is high, latency is low, response time is less and system dependability exists. The system should respond as soon as there is any changes in the sensor state and it take necessary actions.

**Performance requirements**

The only way in which systems will meet their performance targets is for them to be specified clearly and unambiguously. It is a simple fact that if performance is not a stated criterion of the system requirements then the system designers will generally not consider performance issues. While loose or incorrectly defined performance specifications can lead to disputes between clients and suppliers. In many cases performance requirements are never ridged as system that does not fully meet its defined performance requirements may still be released as other consideration such as time to market.

In order to assess the performance of a system the following must be clearly specified:  
• **Response Time**- The storage should be able to log the data as soon as the state of the sensor changes. It should not take more than 30 seconds for this operation. Also the Manager and the HR department should be informed within 1 minute of event.

• **Workload**- At a time the system should be able to process the data of 500 users and log them. Also 100 users should be able to access could service interface.

**• Scalability**- If the number of users increases then the by just adding new hardware the software should be able to handle the extra users. The system should be able to accommodate changes as and when required.

• **Platform-** The system should work in Windows, Linux and iOS platforms with minimum or no changes. The mobile application for Android, Windows and iOS should be developed. The system should work in all the platform mentioned above without any performance degradation.

**Database requirements**

The database is used to log the working duration of every employee, based on this data the number of hours worked by the employee is kept track and it can be used to track the performance of the employees. Also the number of employees inside the office is logged in the database. The database is highly dynamic as the data in it constantly gets updated based on the state of the sensors. This data has to be processed as and when it arrives and if it meets any constraints then the desired operation has to be performed. This logged data has to be cleared every year to save storage and just the total number of hours worked should be saved. MySQL is used as query language, as it is simple and efficient for Relational Database Management Systems.

**Design Constraints**

1. **Hard drive space**: The amount of space an application needs for storage and execution purpose is major design limitation as Raspberry Pi uses a SD card for persistent storage.
2. **Application memory Usage**: The amount of memory space needed for an application to run is an important limitation in the design as Raspberry Pi has only 1GB RAM and it needs to effectively utilize it.
3. **Budget**: The system should be efficient with high performance at the same time it should be within the budget.
4. **Application Quality:** The system should be free of errors and it has to be tested extensively to achieve this. The crash rate should be very minimum.

**Other requirements**

1. **Implementation Requirement:** Python should be used as programming language for extracting the data, Java for Mobile application and MySQL as querying language for database.
2. **Loading OS Requirement:** To load the OS into the SD card another PC is used and then it is installed using Raspberry Pi board.
3. **Network Requirement:** To enable raspberry pi to access internet through Wi-Fi, adapter is used. The internet connection should be of standard bandwidth for uninterrupted functioning of system.
4. **SD card Requirement:** The SD card that is used for loading OS and which provides persistent storage for Raspberry Pi must be of class 10 or higher to increase the performance
5. **Jumper Wire Requirement:** Female-male, Male-female, Male-male and Female-female jumper wires are used.
6. **DESIGN**

**Introduction**

In order in to optimize the design process and also to ensure that the design is efficient and the final product designed meets the desired requirements and achieves the goals and objectives of the project, it is essential to divide the project into various components or modules whose description has to be defined clearly.**Modular design**, or "modularity in design", is a design approach that subdivides a system into smaller parts called modules or skids [8], which can be independently created and then used in different systems. A modular system can be characterized by functional partitioning into discrete scalable, reusable modules, rigorous use of well-defined modular interfaces, and making use of industry standards for interfaces.The different modules needed and their description for the Human and Energy Resource Management are given below

1. *People Counter Module – The module constitutes of detecting direction of movement using two IR sensors to increment or decrement the count of number of people in the room.*
2. *Time Calculation – The module qualifies the work output of employees using a pressure and heat detector. The occupancy of a cubicle chair implies some weight and heat is present on it. Thus this module calculates time by taking input from the sensor.*
3. *MQTT communication - Enable the Raspberry Pi and mobile devices to send and receive small messages and data to and from each other.*
4. *Cloud storage Module – Real time Data storage and access through the cloud.*
5. *Local DB Module – Provide the local database storage for the RTO database and stores the image in binary format.*

**Algorithm**

**Algorithm 1**

Input pins:

6 - Input from IR detector I

40 - Input from IR detector II

Output pins:

39 - Output to LED/Relay Circuit-230V light

Count is managed by the following algorithm

Count():

while(true):

if(Outside\_detector==First\_obstructed&&Inside\_detector==Second\_obstructed)

{

Increment count

If count == 1:

Switch on resources

}

elif(Outside\_detector==Second\_obstructed&&Inside\_detector==First\_obstructed)

{

Decrement count

If count == 0:

Switch off resources

}

**Algorithm 2**

Input pins:

6 - Input from IR detector I

40 - Input from IR detector II

Time worked or Spin-up time is calculated by the following algorithm

Spin\_up():

While true:

if(temp\_sensed~human\_temp&&pressure\_sensed>=critical\_pressure)

{

Continue\_time\_calculation

}

if(temp\_sensed!~human\_temp&&pressure\_sensed<critical\_pressure)

{

Pause\_time\_calculation

Update database

}

if(time\_calculated>=max\_work\_hours\_per\_week)

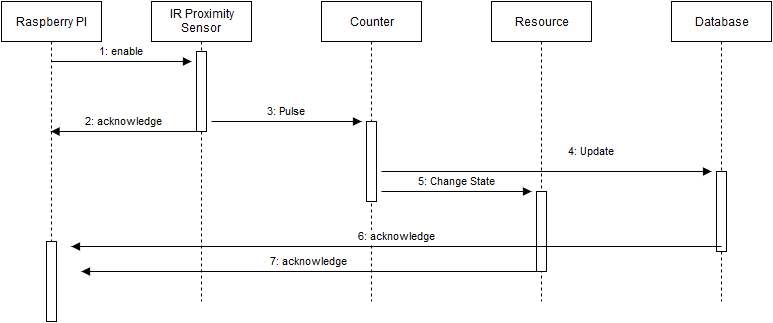
{

Send email to HR and authorities concerned

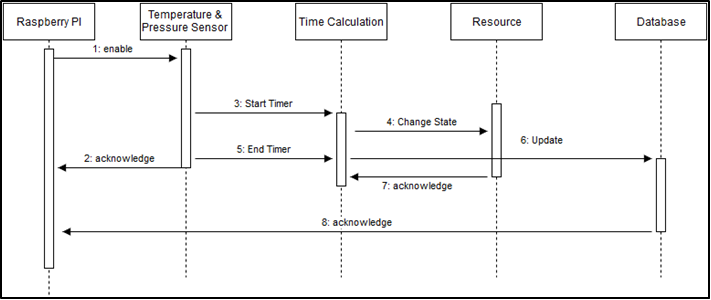
}

**Sequence diagram**

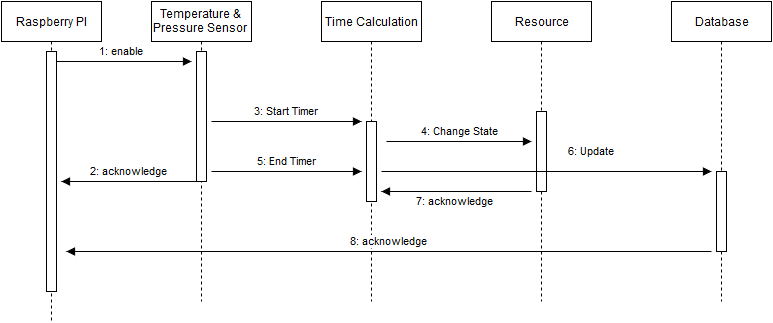
**Sequence diagram** is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams** or **event scenarios** [8]. This is shown in the Figure.



Calculation of number of people



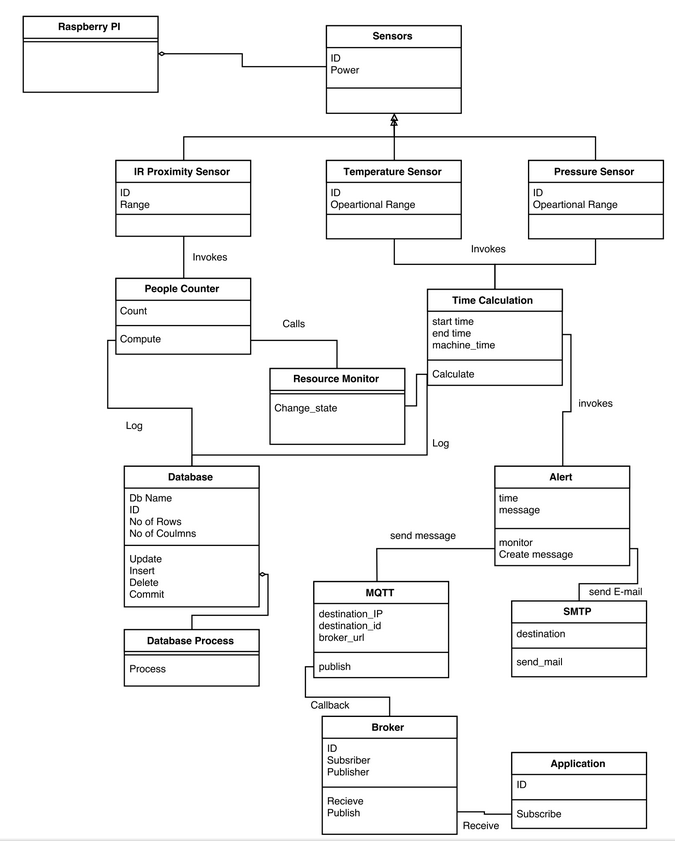
Calculation of time worked



Intimation to authorities

**Class Diagram**

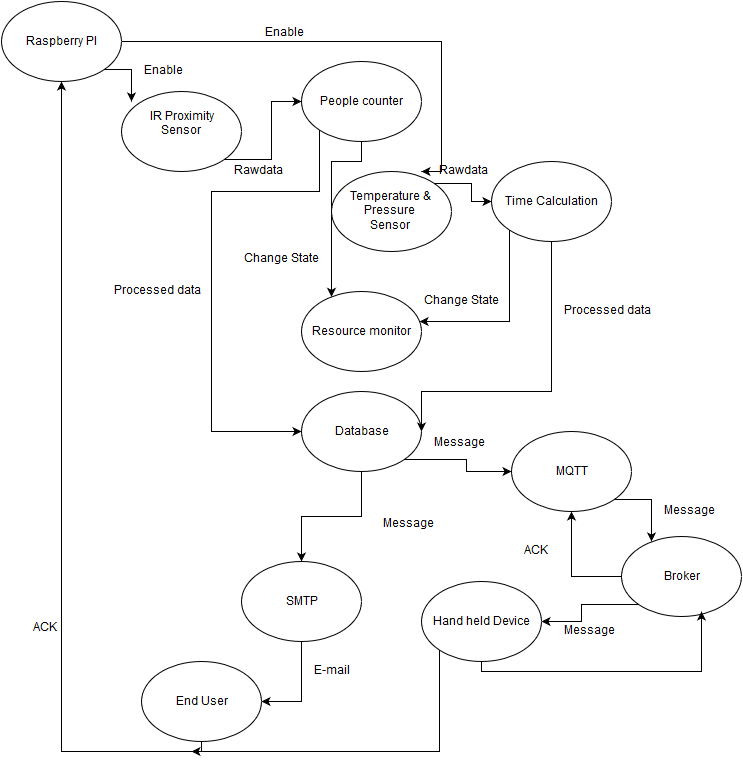
In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects [8]. It’s shown in the figure.



Class Diagram

**Data Flow Diagram**

A **data flow diagram** (**DFD**) is a graphical representation of the "flow" of data through an information system, modeling its processaspects [8]. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). It’s shown in the figure.



Data Flow Diagram

1. **Implementation**

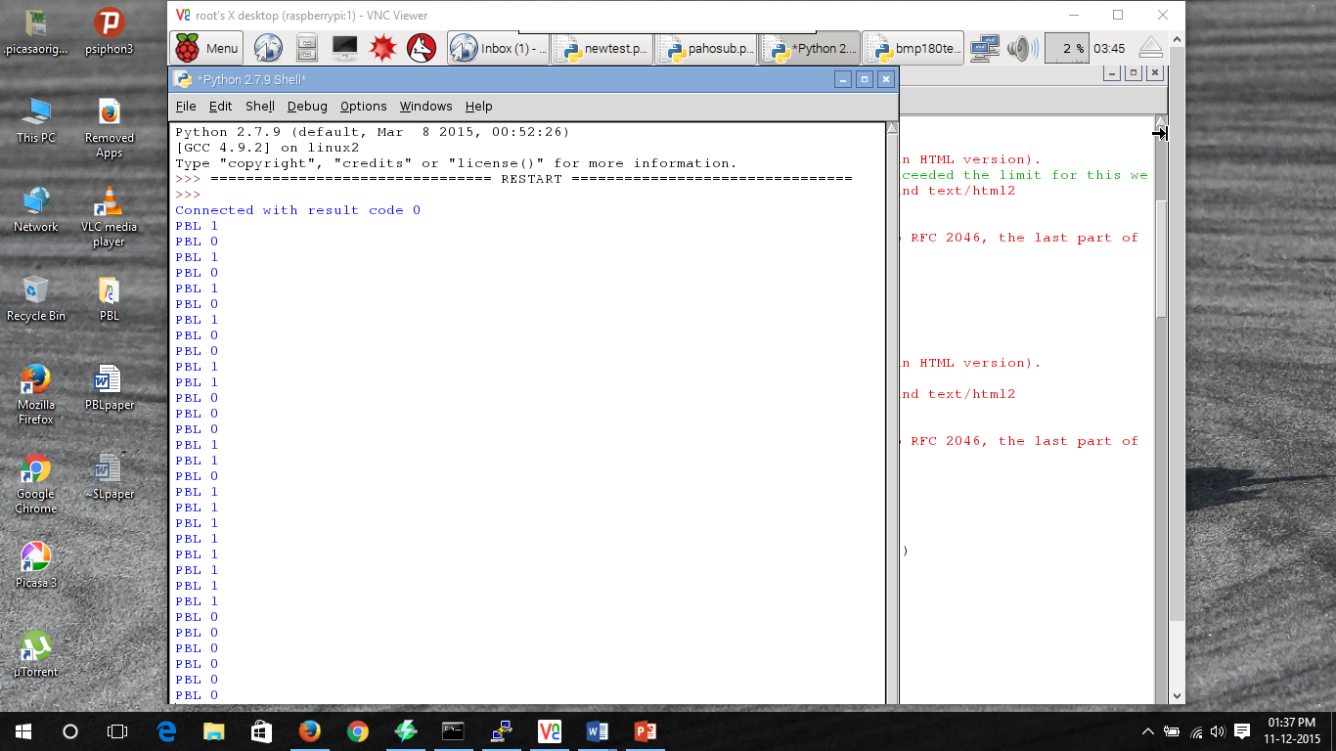
**Reading Distance values from sensors**

Two proximity sensors are used to detect direction of motion of the employee.

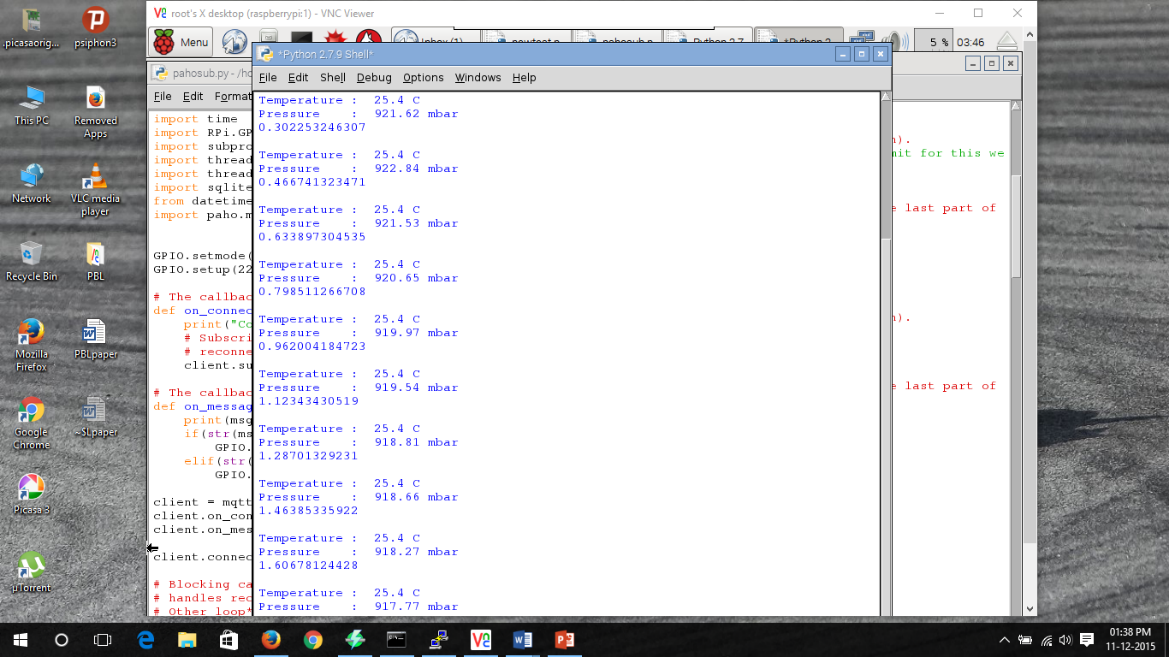
The energy consuming equipment in the area are managed according to the direction of employee sensed.

Pressure and temperature sensors are used to detect employee presence in the cubicle.

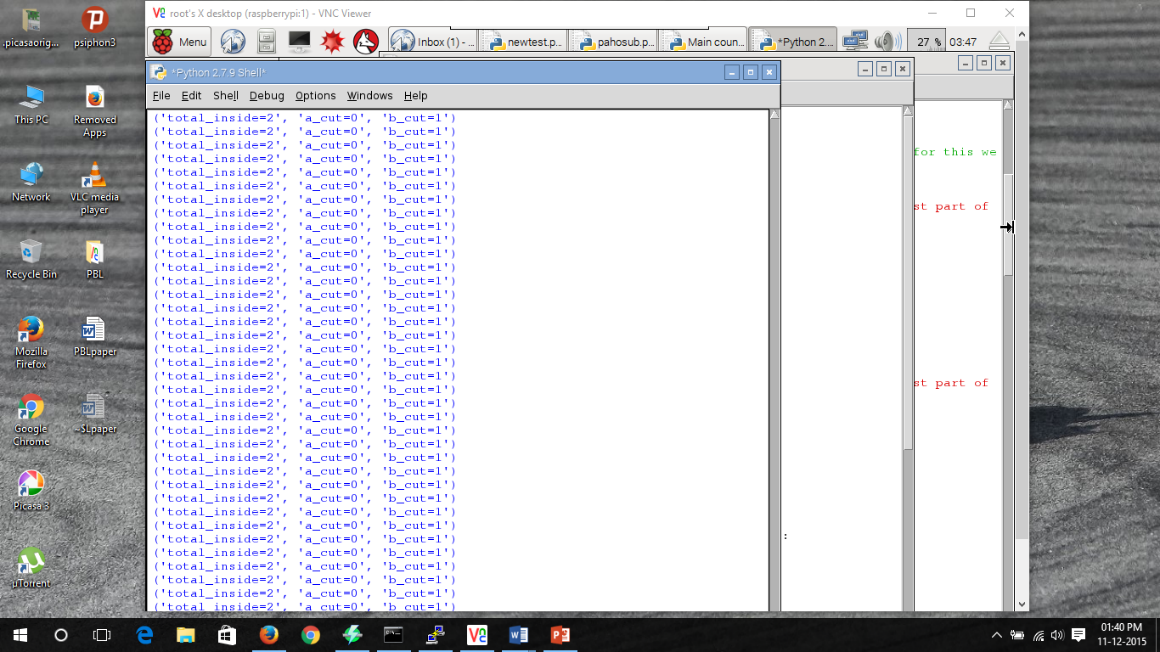
This data can be used to manage cubicle resources and calculate the number of hours worked per week.



Mobile app output



Time calculation output



No. of people count

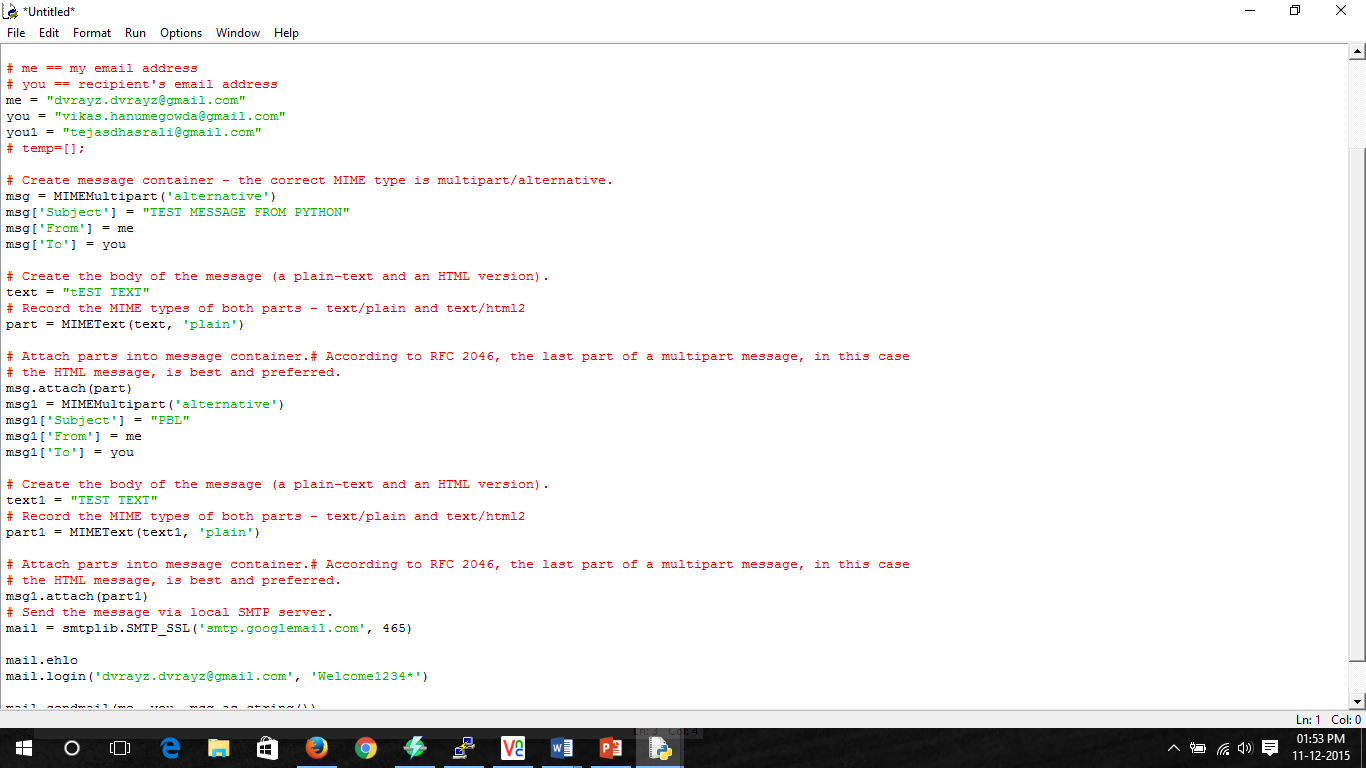


Mobile interface

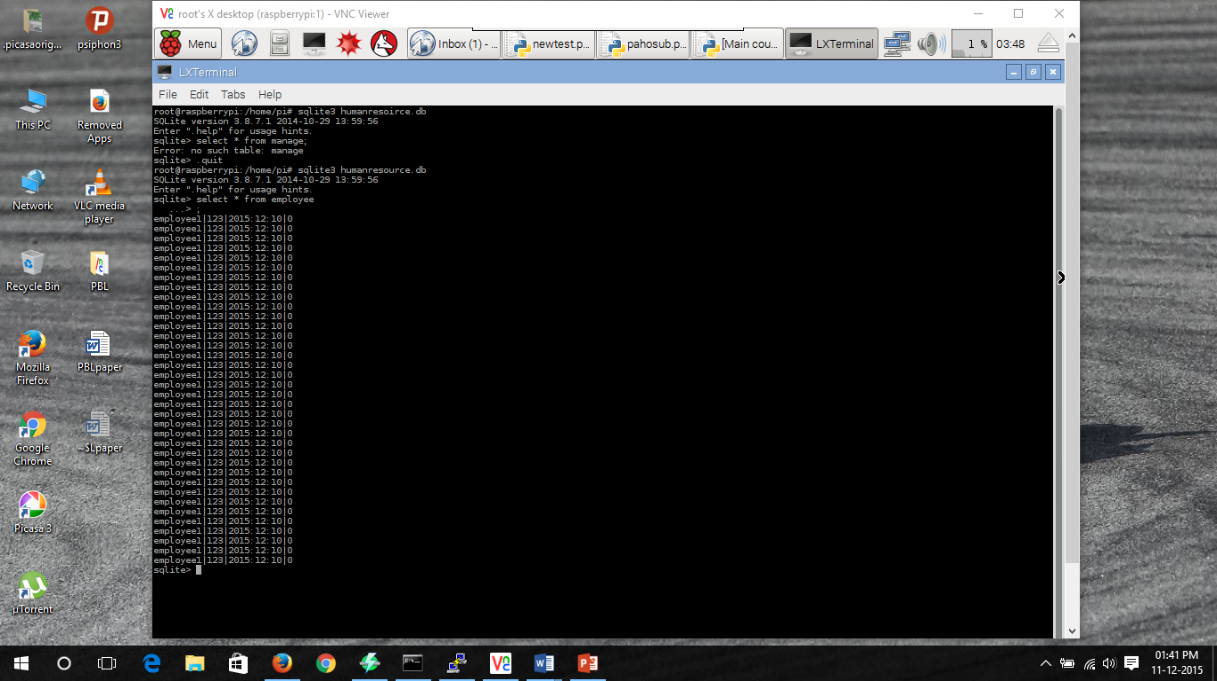
**Gmail alerts and Cloud Updates**

As per the users command, the messages are sent to the broker cloud service and the status of the energy consuming resource is controlled. When the working hours of a particular employee exceeds the pre-determined time limit, a mail is sent to the manager. Also, daily reports are sent to the manager concerned about his subordinates’ working statistics

.



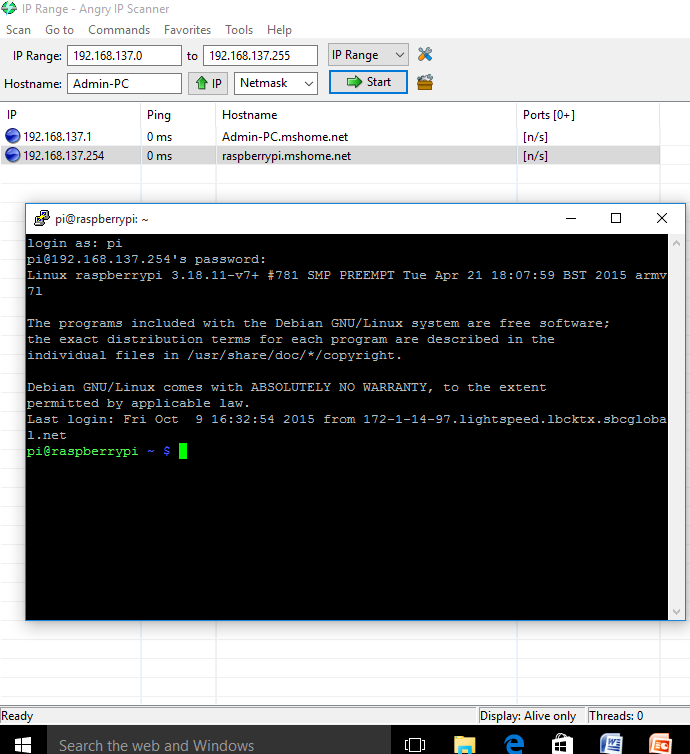
Gmail Alert code



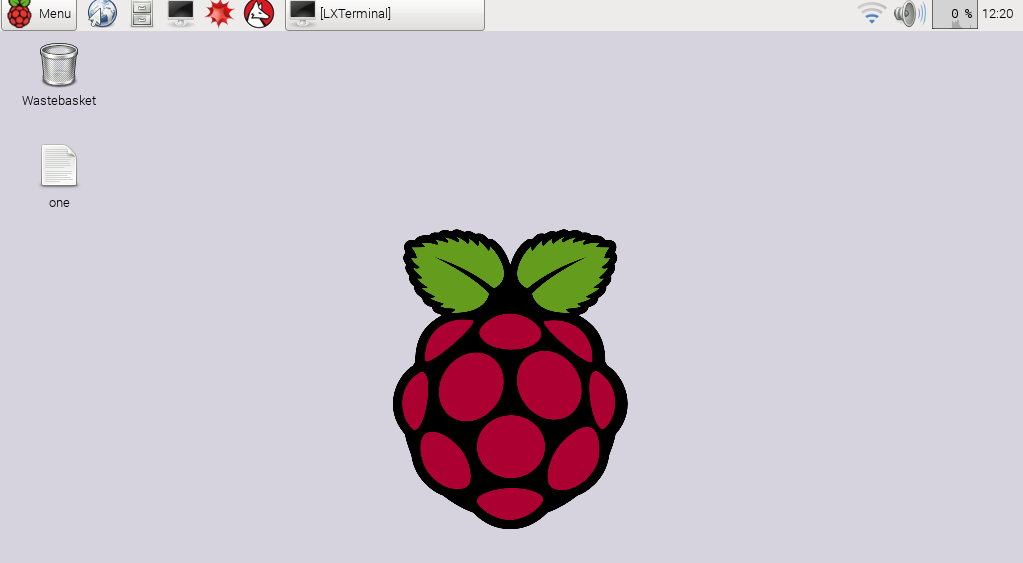
SQLite Database output

**7. TESTING**

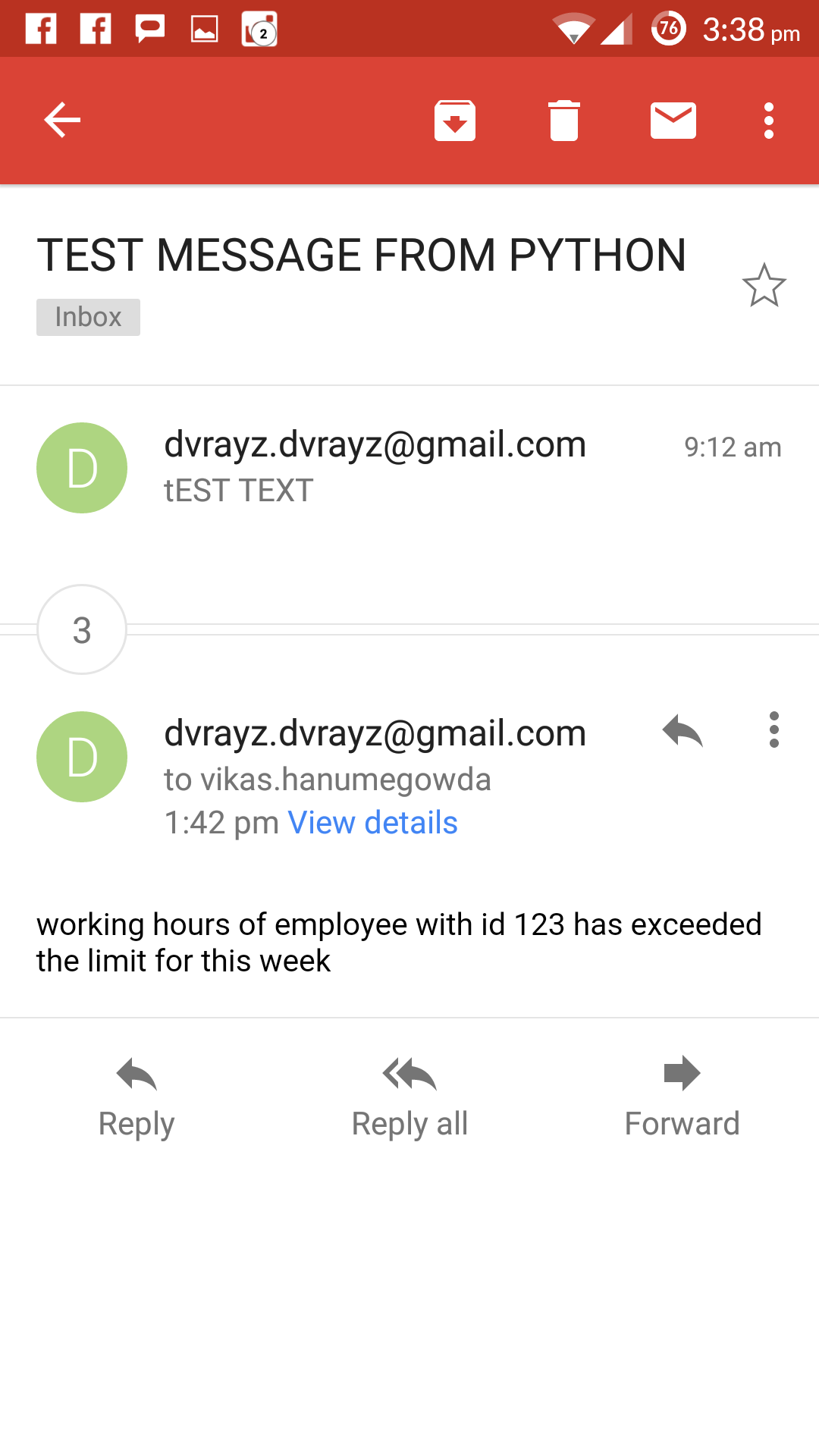
**7.1 Results and Snapshots**



Angry IP Scanner and Putty



VNCviewer



Gmail Alert

**8. CONCLUSION AND SCOPE FOR FUTURE WORK**

Thus we have designed a Human and Energy management system. It is important to conserve energy because the natural resources that are essential for production of these energy are being depleted faster than they can be regenerated. This problem can be minimised by automating resource usage using our proposed model.

The man force in a particular company though eminent in their fields are still humans. Workaholics need to be managed by HR department and our project does exactly that by informing authorities when an employee exceeds his maximum working hours per week. Contrary to that, there will be employees who spend time in the campus without working. Our model gives statistics on the number of hours an employee has worked.

**9. REFERENCES**

The following online links were used

* <http://www.element14.com/community/thread/34969/l/wireless-monitoring-system-using-raspberry-pi?displayFullThread=true>
* <http://www.slideshare.net/KrishnaKumar272/smart-wireless-surveillance-monitoring-using-raspberry-pi>
* <https://en.wikipedia.org/wiki/Wiki>
* <https://www.raspberrypi.org/>
* <http://www.projects.privateeyepi.com/home/temperature-gauge>
* <http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>

The following books were very helpful

* Raspberry Pi User Guide by Eben Upton and [Gareth Halfacree](http://www.amazon.com/Gareth-Halfacree/e/B0088CA5ZM/ref=sr_ntt_srch_lnk_1?qid=1449728684&sr=1-1).
* Shan and M. Richardson Getting *Started with Raspberry pi*, 2012 :O'Reilly Media, Inc
* Raspberry Pi Projects for the Evil Genius by [Donald Norris](http://www.amazon.com/Donald-Norris/e/B00FRYI2GI/ref=sr_ntt_srch_lnk_2?qid=1449728684&sr=1-2).