

**A HOME AUTOMATION SYSTEM**

**18-088**

Design Document (DD)

|  |  |  |
| --- | --- | --- |
| Name | Student ID | Signature |
| O.Y.Navarathneraja | IT15050786 |  |
| A.R.J. Akwaththa | IT15093660 |  |
| H.M.P.H. Halpe | IT15052902 |  |
| W.S.H. Boteju | IT14034114 |  |

*Bsc. Special Honors Degree*

Department of Information

Sri Lanka Institute of Information Technology Sri Lanka

March 2018

# DECLARATION OF THE CANDIDATES AND SUPERVISOR

We declare that this is our own work and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

|  |  |  |
| --- | --- | --- |
| Name | Student ID | Signature |
| O.Y.Navarathneraja | IT15050786 |  |
| A.R.J. Akwaththa | IT15093660 |  |
| H.M.P.H. Halpe | IT15052902 |  |
| W.S.H. Boteju | IT14034114 |  |

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

**Name of the Supervisor:** S**ignature of the supervisor: Date: Mr. Yashas Mallawarachchi**

**Table of Contents**

[DECLARATION OF THE CANDIDATES AND SUPERVISOR 2](#_bookmark0)

[Figures 5](#_bookmark1)

[Introduction 6](#_bookmark2)

* 1. [Purpose 6](#_bookmark3)
  2. [Scope 6](#_bookmark4)
  3. [Definitions, Acronyms, and Abbreviations 6](#_bookmark5)
  4. [Overview 8](#_bookmark8)

1. [Overall Descriptions 9](#_bookmark9)
   1. [Product Perspective 9](#_bookmark10)
      1. [System interface 10](#_bookmark11)
      2. [User interfaces 11](#_bookmark13)
      3. [Hardware interfaces 13](#_bookmark14)
      4. [Software interfaces 13](#_bookmark15)
      5. [Communication interfaces 13](#_bookmark16)
      6. [Memory constraints 14](#_bookmark17)
      7. [Operations 14](#_bookmark18)
      8. [Site adaptation requirements 14](#_bookmark19)
   2. [Product functions 15](#_bookmark20)
      1. [Automating and Controlling the Functionality of the Garage Gate and Light. 15](#_bookmark21)
      2. [Automation and Controlling the Landscape Lighting, per Natural Light. 17](#_bookmark23)
      3. [Remotely Controlling High technical appliances 18](#_bookmark26)
      4. [Remotely Controlling Locking and Unlocking the Doors and Windows 19](#_bookmark29)
      5. [Ensuring the System is fully secured. 22](#_bookmark37)
   3. [User Characteristics 24](#_bookmark41)
   4. [Constraints 25](#_bookmark42)
   5. [Assumptions and dependencies 25](#_bookmark43)
   6. [Apportioning of requirements 25](#_bookmark44)
2. [Specific requirements 26](#_bookmark45)
   1. [External interface requirements 26](#_bookmark46)
      1. [User interfaces 26](#_bookmark47)
      2. [Hardware interfaces 31](#_bookmark54)
      3. [Software interfaces 32](#_bookmark55)
      4. [Communication interfaces 33](#_bookmark56)
   2. [Architectural Design 33](#_bookmark57)
      1. [High level Architectural Design 33](#_bookmark58)
      2. [Hardware and software requirements with justification 34](#_bookmark60)
      3. [Risk Mitigation Plan with alternative solution identification 35](#_bookmark61)
      4. [Cost Benefit Analysis for the proposed solution 37](#_bookmark63)
   3. [Performance Requirements 38](#_bookmark65)
   4. [Design Constraints 38](#_bookmark66)
   5. [Software system attributes 39](#_bookmark67)
      1. [Reliability 39](#_bookmark68)

[3.5.2. Availability 39](#_bookmark69)

* + 1. [Security 39](#_bookmark70)
    2. [Maintainability 39](#_bookmark71)

1. [Supporting information 40](#_bookmark72)
   1. [References 40](#_bookmark73)
   2. [Appendices 41](#_bookmark74)

Tables

[Table 1 : Acronyms and Abbreviations 7](#_bookmark6)

[Table 2: Definitions 7](#_bookmark7)

[Table 3 : Use case - To Automate the Functionality of the garage 15](#_bookmark22)

[Table 4 : Use case To Automate the Functionality of the Garage Light 16](#_bookmark24)

[Table 5 : Use case- To Automate the Functionality of the Landscape Lighting 17](#_bookmark25)

[Table 6 : Use case scenario for controlling appliances 18](#_bookmark27)

[Table 7 : Use Case scenario for view Status of appliances 18](#_bookmark28)

[Table 8 : Check Status 19](#_bookmark30)

[Table 9 : Lock Door/Windows 19](#_bookmark31)

[Table 10 :Unlock Doors/ Windows 20](#_bookmark32)

[Table 11 : Send Data Packets 20](#_bookmark33)

[Table 12 : Receive Data Packet 21](#_bookmark34)

[Table 13 : Calculate RSSI value 22](#_bookmark35)

[Table 14 : Detect variation of RSSI value 22](#_bookmark36)

[Table 15 : Send Notification 23](#_bookmark38)

[Table 16 : Use case scenario Interior lighting control system 23](#_bookmark39)

[Table 17 : Use case scenario for Temperature controlling system 24](#_bookmark40)

[Table 18 : Risk Mitigation Plan 35](#_bookmark62)

[Table 19 : Cost Benefit Analysis Table 37](#_bookmark64)

# Figures

[Figure 1: System Interaction with Hardware and Software 10](#_bookmark12)

[Figure 1 : Home Page 27](#_bookmark48)

[Figure 2 : Exterior Home Management Interface 27](#_bookmark49)

[Figure 3 : High technical Appliance interface 28](#_bookmark50)

Figure 4 :Main access Control interface 28

[Figure 5 : Window remote Control Interface 29](#_bookmark51)

Figure 6 :Door remote control interface 30

[Figure 7 : Interior lightning system interface 30](#_bookmark52)

[Figure 8 :Temperature controlling system interface 31](#_bookmark53)

[Figure 9 : High Level System Architecture 34](#_bookmark59)

# Introduction

## Purpose

The purpose of Software Design Document is to present a detailed description of the Designs of the Home Automation System, which is created to ensure users their Home is Safe. The Design Document provides the designs used by designers to upgrade or modify the current design of the system and by the developers as guidelines to implement the project.

## Scope

This document gives an elaborated description of the Home automation system. It specifies the structure and design of the entire system components, and how the software communicates with the micro-controllers.

Scope of this document is to give a basic idea about how the system is going to implement by using raspberry-pi, sensors, other hardware and a web based automatic controlling system. In this document, it covers the design criteria of interior light controlling and temperature controlling. Gathering environmental condition information from different sensor modules, thought that handling the system accordingly.

Expectation of implementing this component is to have a low-cost home automation system with a responsive web app. All the other components are connected through raspberry pi. This component will control all the hardware and check the current condition of all. Send notifications to the user via the responsive web app. Per the notifications received, appropriate action is taken.

## Definitions, Acronyms, and Abbreviations

**Table 1 : Acronyms and Abbreviations**

|  |  |
| --- | --- |
| SDD | Software Design Document |
| UI | User interfaces |
| API | Application programming interface |
| RAM | Random-access memory |
| LDR | Light dependent resister |
| IEEE | Institute of Electrical and Electronics Engineers |
| VCC | Voltage Common Collector (IC Power supply pin) |
| GND | Ground (Earth) |
| GPIO | General Purpose Input/output |
| AI | Artificial Intelligence |
| GSM | Global System for Mobile communications |
| IoT | Internet of Things |
| TTL | Time to Live |

**Table 2: Definitions**

|  |  |
| --- | --- |
| **Terms** | **Definitions** |
| Software Design Document | Software Design Document is a description of  a software product, that a software designer writes in order to give a software development team overall guidance to the architecture of the software project. |
| Bootstrap | Bootstrap is a free and open-source front-end library for designing responsive web applications. |
| Raspberry Pi | Raspberry Pi is a low-cost, basic computer that runs entirely on open-source software and gives students the ability to mix and match software according to the work they wish to do. |
| API | Application program interface (API) is a set of routines, protocols, and tools for building software applications.  An API specifies how software components should interact. |

|  |  |
| --- | --- |
| Internet of Things | The Internet of Things is the network of physical devices, vehicles, home appliances and other items embedded with electronics, software, sensors, actuators, and connectivity which enable these objects to connect and exchange data. |
| Image processing | Method to convert an image into digital form and perform some operation on it. |
| Machine learning | Machine learning is an application of artificial intelligence that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. |
| OpenCV | OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision. |
| Encryption | Encryption is the most effective way to achieve data security. To read an encrypted file, you must have  access to a secret key or password that enables you to decrypt it. |

## Overview

This document is written per the standards for Software Design Documentation explained in “*IEEE Recommended Practice for Software Design Documentatio*n”.

The first section in this document discusses the designs for the System, using Use case diagrams; the next section shows samples of UI designs.

The rest of the individualized document will discuss on the Product perspective, the User Characteristics, the Constraints that will limit the development implementation, Assumptions made prior to the development; furthermore, since our system interacts with hardware components, the Design Document will consist of the Specific requirements for the Embedded system oriented project which discuss on the descriptive interfaces such as system interface, user interface, Hardware and Software interfaces, etc., next we move on to the Architectural Design which describes the High level diagram, the hardware and software requirements, the Risk Mitigation Plan that would be taken if there is a system failure, and the Cost benefit Analysis; Finally the System software attributes are discussed in the document.

# Overall Descriptions

Raspberry pi is the main hardware of the system. Implementing the all functions by using this component. Raspberry Pi is a series of small single-board computers. It has evolved through several versions that feature variations in memory capacity and peripheral-device support.

Other most important hardware is Relay. It is the link between low power digital electronics and high-power devices. They allow digital circuits and digital microcontrollers to switch high power devices on and off. For the sensors, we are using motion detection sensors, temperature detection sensors etc.

To develop the responsive web application Technologies used to create the web application is; Bootstrap framework, HTML 5, JavaScript, PHP.

## Product Perspective

There are Bluetooth based home automation system using cell phone. Since our product is internet base we can have the ability to control the system far away from home. Which is not commercially available in the home automation systems.

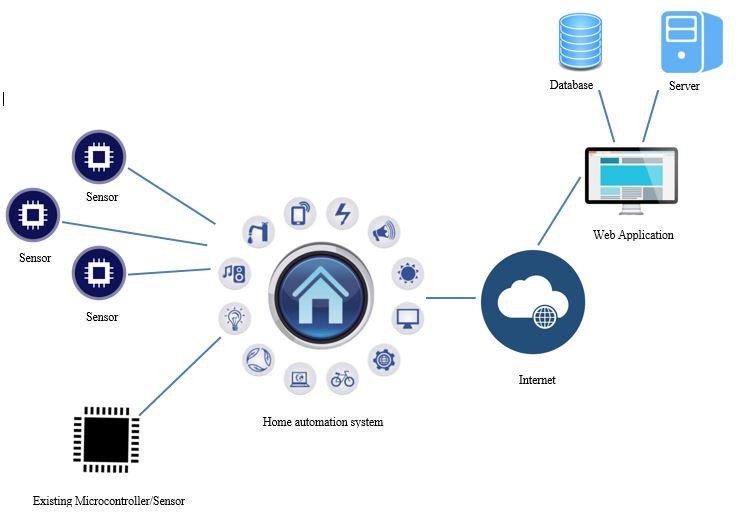
In voice recognition-based home automation system Hardware architecture consists of Arduino UNO and Smartphone. System comprises a DSP processor for the voice recognition function. Android OS has a built-in voice recognizing feature. This voice recognizing features is used to develop an app which control appliances from user voice command. This application converts the user voice command into text. Then it transmits that text message to Bluetooth module which is connected to Arduino. Advantage of voice- controlled home automation system is that user only need to pronounce the application name and the command. A voice recognition application provided a user-friendly interface to users and it has ability to add more appliances.

Drawback is this system failed to work correctly in noisy environments. That will be real disadvantage of this kind of system. Sometimes it’s only work with only one voice recognition. That will be another disadvantage. Its only work when we inside the home.

Can’t remotely controlled when we are far away from home. But in our home automation system we don’t have those kinds of drawbacks since it is a web base remotely control system.

### System interface

This diagram, allows the reader and user of the document to orient themselves to the design and flow of the functions.

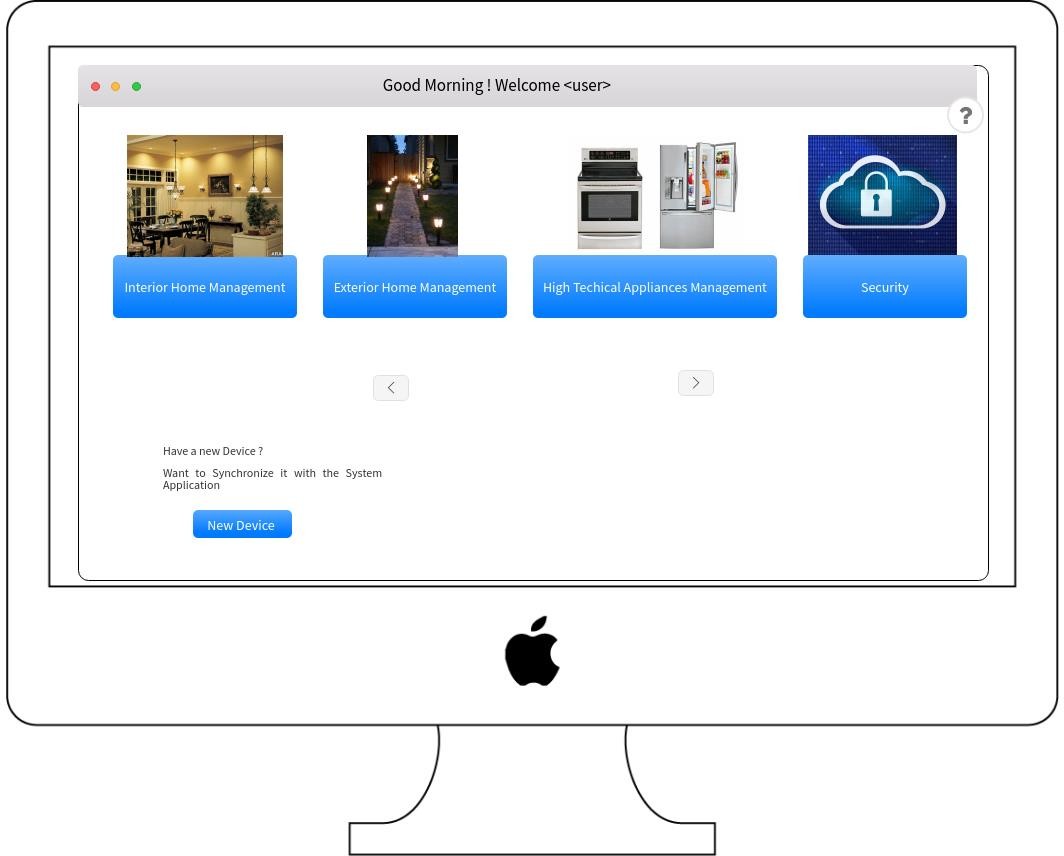


**Figure 1: System Interaction with Hardware and Software.**

### User interfaces

Below shows the sketch of a User Interface of the homepage of the home automation system web application. This interface comes after the registered user login. Through this interface, User can navigate to four main sections of home to control remotely.

* + - * Interior Home Management
      * Exterior Home Management
      * High Tech Appliances
      * Security



* **Interior Home Management**

This section includes the lightning system and cooling system of home. Controlling the devices assigned to the lightning and cooling system can be done through this button in the home page.

* **Exterior Home Management**

Garage area will be controlled by the button ‘Exterior Home Management’. Lightning of the garage and the control of the garage door can be handle in this section.

* **High Technical Appliances**

High technical appliances such as Fridge, TV and AC can be controlled remotely through the interface which going to be pop up after clicking the ‘High Tech Appliances’ button.

* **Security**

Access control system will be handle from the interface comes after clicking ‘Security’ button. Access control system includes remote control capability for doors and windows of home.

### Hardware interfaces

Hardware Required for “Smart Home Automation System”:

* Raspberry pi
* Light Sensor module
* Door/Window locking sensor
* Photo-resistor
* Capacitor
* Garage Sensor
* Breadboard
* Relay (To control 120-240V devices)
* Wires

### Software interfaces

Software’s required for “Smart Home Automation System”:

* Apache
* MY SQL
* Bootstrap 3
* VNC Viewer
* WebStorm

### Communication interfaces

* Internet speed over 100 Mbps going to be need for the site server since it retrieves and sends data from the sensors and to the sensors and send and receives notification from the user.
* End user should have at least 1Mbps speed to use the application without any delay.
* HTTP protocol use to communicate with the cloud since the system handles its data in a cloud.
* Wi-Fi module is used to maintain the interference connection to provide integration of the IOT devices.

### Memory constraints

Since all the data are sent to the cloud, mobile device need space only to store the apk , server need space only to store the web application and related database.

Memory requirement for implementing the functionalities:

* 25 GB cloud Storage
* 10 GB space on Hard Dish
* 20Gb IDE disc

### Operations

1. Proposed home automation system is consisting of two main applications.
   * Web application
   * Mobile application
2. User is logged into to the system, using the given credentials
3. User navigates to the Exterior Home Management section.
4. User can Control the Functionality of the Garage Gate through the application manually.
5. User can Control the Functionality of the Landscape Lighting as desired through the application manually.

### Site adaptation requirements

Site Adaptations Required for “Smart Home Automation System”:

* Application must always up and running in the cloud. User must have Internet connection to control home from anywhere at any time.
* Raspberry pi should have a power supply all the time.

## Product functions

### Automating and Controlling the Functionality of the Garage Gate and Light.

This is a subset component of the Exterior Home Management system for “Home Automation System”.

|  |  |
| --- | --- |
| ID | EM 01 |
| Title | Automating Garage Door |
| Description | The Garage Door will automatically Open when the Vehicle approaches |
| Actor | System, User |
| Pre-condition | Owner vehicle license plate is saved. Hardware components are connected. |
| Post-condition | When Owners vehicle is close by, the Garage door opens |
| Main Success Scenario | 1. Camera captures the approached vehicle 2. Compares with saved license plate number 3. Garage door opens for the vehicle 4. Vehicle enters the Garage 5. Camera monitors the distance 6. Compares with the set threshold 7. Garage Door Closes |
| Extension | 2a. The approaching vehicle is not the Home Owners Vehicle 2a.1 The Garage Door remains closed  6a. The threshold is not safe.  6a.1 A Beep sound notification is sent to the application to alert the owner.  6a.1.a The user can manual off the alert. |

**Table 3 : Use case - To Automate the Functionality of the garage**

|  |  |
| --- | --- |
| ID | EM 03 |
| Title | Automating Garage Light |
| Description | The Garage Lighting will automatically turn On, when the garage door opens and natural light is limited |
| Actor | System, User |
| Pre-condition | Garage door is opening.  Hardware components are connected. |
| Post-condition | Garage Light is on, when the vehicle is entering the garage. |
| Main Success Scenario | 1. The Owners vehicle is entering the Garage 2. The sensor detects the Limited Light in the Garage 3. The Garage light is turned On 4. The Garage Door closes 5. The Owner leaves the room 6. The Camera Detects there is no-one in the Garage 7. The Light is automatically turned off |
| Extension | 2. A. There is sufficient light in the Room  2. A.1 The Light remains switched off.  5. A. The owner manually turns the Light Off.  5. A.1 The Sensor gets updated on the status of the Light. |

**Table 4 : Use case To Automate the Functionality of the Garage Ligh****t**

### Automation and Controlling the Landscape Lighting, per Natural Light.

This is another subset component of the Exterior Home Management system for “Home Automation System”.

|  |  |
| --- | --- |
| ID | EM 02 |
| Title | Automating Landscape Light |
| Description | The landscape Lighting will automatically turn On, when natural light is  limited. |
| Actor | System, User |
| Pre-condition | Hardware components are connected. |
| Post-condition | Depending on the Natural Light, the Landscape light is switched on |
| Main Success Scenario | The sensor detects the UV Light  In the Evening, When the UV light is low The Landscape light is switched on  During Dawn, when the UV light gets high The Landscape Light is switched off |
| Extension | 1.a When the sensor is blocked or fails, an Alert is sent to the user |

**Table 5 : Use case- To Automate the Functionality of the Landscape Lighting**

### Remotely Controlling High technical appliances

|  |  |
| --- | --- |
| **Use Case ID** | 001 |
| **Use Case** | Control home appliances |
|  |  |
| **Pre-Conditions** | User should select control high technical  appliances section |
|  |  |
| **Actor** | User |
|  |  |
| **Main Flow of events** | 1. Use case starts with once after the user clicks the control high tech appliances button. 2. Application allow user to control home appliances 3. Use Case ends when the user closes the application. |
| **Alternative Flows** | 2. a) If Application does not allow to control the  home appliances, reload the page again. |

**Table 6 : Use case scenario for controlling appliances**

|  |  |
| --- | --- |
| **Use Case ID** | 002 |
| **Use Case** | View current status of the appliances |
|  |  |
| **Pre-Conditions** | User should login to the appliance |
|  |  |
| **Actor** | User |
|  |  |
| **Main Flow of events** | 1. Use case starts with once after the user clicks the control high tech appliances button. 2. Application allow user to view current status of appliances. 3. Use Case ends when the user closes the application. |
| **Alternative Flows** | 2. a) If application does not allow user to view status, check whether the appliances connected  with the system properly. |

**Table 7 : Use Case scenario for view Status of appliances**

### Remotely Controlling Locking and Unlocking the Doors and Windows

|  |  |
| --- | --- |
| Use Case ID | 001 |
| Use Case | Check Status |
|  |  |
| Pre-Conditions | User should log into the system |
| Primary User | Home Owner |
| Main Flow of the events | 1. Use case starts when user login into  the system |
|  | 2. Application will display the status  of the door or window |
|  | 3. Use case ends when the user closes  the application |
|  |  |
| * *Essential* | |

**Table 8 : Check Status**

|  |  |
| --- | --- |
| Use Case ID | 002 |
| Use Case | Lock Doors / Windows |
|  |  |
| Pre-Conditions | User should log into the system |
| Primary User | Home Owner |
| Main Flow of the events | 1. Use case starts when user login  into the system |
|  | 2. Application send a command to  the microcontroller to lock the door/window |
|  | 3. Use case ends when the user  closes the application |
|  |  |
| * *Essential* | |

**Table 9 : Lock Door/Windows**

|  |  |
| --- | --- |
| Use Case ID | 003 |
| Use Case | Unlock Doors / Windows |
|  |  |
| Pre-Conditions | User should log into the system |
| Primary User | Home Owner |
| Main Flow of the events | 1. Use case starts when user login into  the system |
|  | 2. Application send a command to the microcontroller to unlock the  door/window |
|  | 3. Use case ends when the user closes  the application |
|  |  |
| * *Essential* | |

**Table 10 :Unlock Doors/ Windows**

|  |  |
| --- | --- |
| Use Case ID | 004 |
| Use Case | Send Data Packets |
|  |  |
| Pre-Conditions | Sensor respond according to the request  come by the micro controller |
| Primary User | Sensor |
| Main Flow of the events | 1. Respond to the request comes from  the micro controller |
|  | 2. Change the status according to the  request |
|  | 3. Use case ends when sensor notify  the change to the micro controller |
|  |  |
| * *Essential* | |

**Table 11 : Send Data Packets**

|  |  |
| --- | --- |
| Use Case ID | 005 |
| Use Case | Receive Data Packets |
|  |  |
| Pre-Conditions | Sensor should send data packets to micro  controller |
| Primary User | Micro controller |
| Main Flow of the events | 1. Use case starts when micro controller receives data packets  from the sensor |
|  | 2. Micro controller then checks the command from the user is equal to  the status |
|  | 3. Use case ends when micro  controller accepts the data packets |
|  |  |
| * *Essential* | |

**Table 12 : Receive Data Packet**

|  |  |
| --- | --- |
| Use Case ID | 006 |
| Use Case | Calculate RSSI value |
|  |  |
| Pre-Conditions | Micro controller should receive data  packets from the sensor |
| Primary User | Micro controller |
| Main Flow of the events | 1. Use case starts when the  microcontroller receives data packets from the sensor. |
|  | 2. Check the RSSI value |
|  | 3. Use case ends when the RSSI value  get confirmed |
|  |  |
| * *Essential* | |

**Table 13 : Calculate RSSI value**

|  |  |
| --- | --- |
| Use Case ID | 007 |
| Use Case | Detect variation of RSSI value |
|  |  |
| Pre-Conditions | Micro controller should calculate the RSSI  value |
| Primary User | Micro controller |
| Main Flow of the events | 1. Use case starts when the  microcontroller detects RSSI value |
|  | 2. Examine the RSSI value variation  and notify the change |
|  | 3. Use case ends when RSSI variation  consider as a major impact. |
|  |  |
| * *Essential* | |

**Table 14 : Detect variation of RSSI value**

|  |  |
| --- | --- |
| Use Case ID | 008 |
| Use Case | Send Notification |
|  |  |
| Pre-Conditions |  |
| Primary User | Server |
| Main Flow of the events | 1. Use case starts when server gets  signals from the micro controller |
|  | 2. Server send notification to the  application |
|  | 3. Use case ends when server confirmed application receive the  notification |
|  |  |
| * *Essential* | |

**Table 15 : Send Notification**

* + 1. **Remotely Controlling Opening and Offing the Lights**

|  |  |
| --- | --- |
| **Use Case ID** | 001 |
| **Use Case** | Interior Lighting control |
|  |  |
| **Pre-Conditions** | Select whatever the place that you want to abject  light |
|  |  |
| **Actor** | User |
|  |  |
| **Main Flow of events** | 1. Use case starts with once after the user clicks the lighting control button. 2. Application allow user to control light of the appropriate section/room 3. Use Case ends when the user closes light control of the place. |
| **Alternative Flows** | 2. a) If Application does not allow to control the  Light control, reload the page again. |

**Table 16 : Use case scenario Interior lighting control system**

* + 1. **Remotely Controlling Temperature in a room**

|  |  |
| --- | --- |
| **Use Case ID** | 002 |
| **Use Case** | Temperature controlling |
|  |  |
| **Pre-Conditions** | User should select the room where you want to  control the temperature |
|  |  |
| **Actor** | User |
|  |  |
| **Main Flow of events** | 1. Use case starts with once after the user clicks the temperature control button. 2. Application allow user to view status the room. 3. Application suggest the appropriate temperature to the room. 4. Do the needful and control the room temperature. 5. Use Case ends when the user closes the temperature application. |
| **Alternative Flows** | 2. a) If application does not allow user to view status, check whether the appliances connected  with the system properly. |

**Table 17 : Use case scenario for Temperature controlling system**

## User Characteristics

A home automation system user is assumed to be average adult person with average technical skills to browse the web responsive app. There is no need for any programming skills to use the system. He/she uses the system daily, mostly to view home devices status and control devices such as modifying home lighting.

## Constraints

* System may crash due to any damage in the interconnection, so there shouldn’t be any damages.
* If there is any damage, due to rupturing of cables or the fibers; this would result in the entire system to crash. Therefore, it is advised for the System to be wired properly, and thoroughly tested, when deploying to the system.
* User should handle the Kit safely. If the user does not handle the kit safely or if he/she does not use the correct keys to perform operations, human errors may occur.
* Because of the limitations of sensors, system requires multiple sensors to detect a certain event.

## Assumptions and dependencies

* There should be a strong internet connection for the application to receive alerts and notifications.
* The Mobile device has sufficient storage to handle data
* The modules and components are connected correctly.

## Apportioning of requirements

To begin implementation, the hardware components must be connected and mounted correctly and tested.

Once hardware connection is complete, the order of implementation for a successful “Home Automation System” is having a fully functional and thoroughly tested set of the four main features:

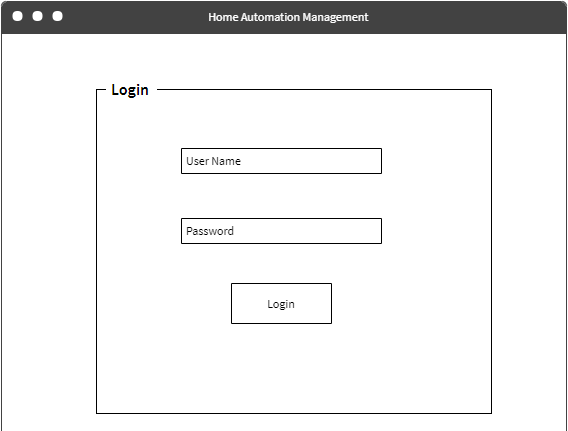
* Interior Lighting Management
* Exterior Lighting Management
* Controlling High technical appliances
* Security

The final Functionality for our Home Automation System is the ability to have an easy and simple flow to integrate and synchronize other devices into our System.

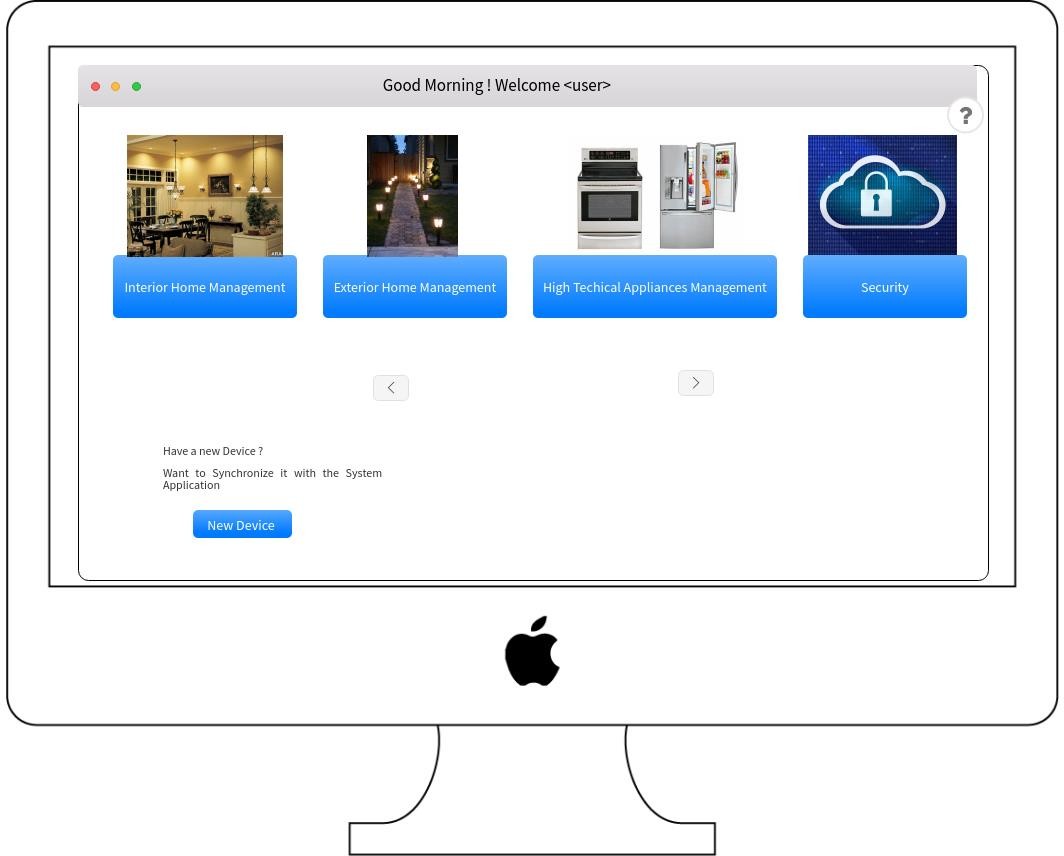
# Specific requirements

## External interface requirements

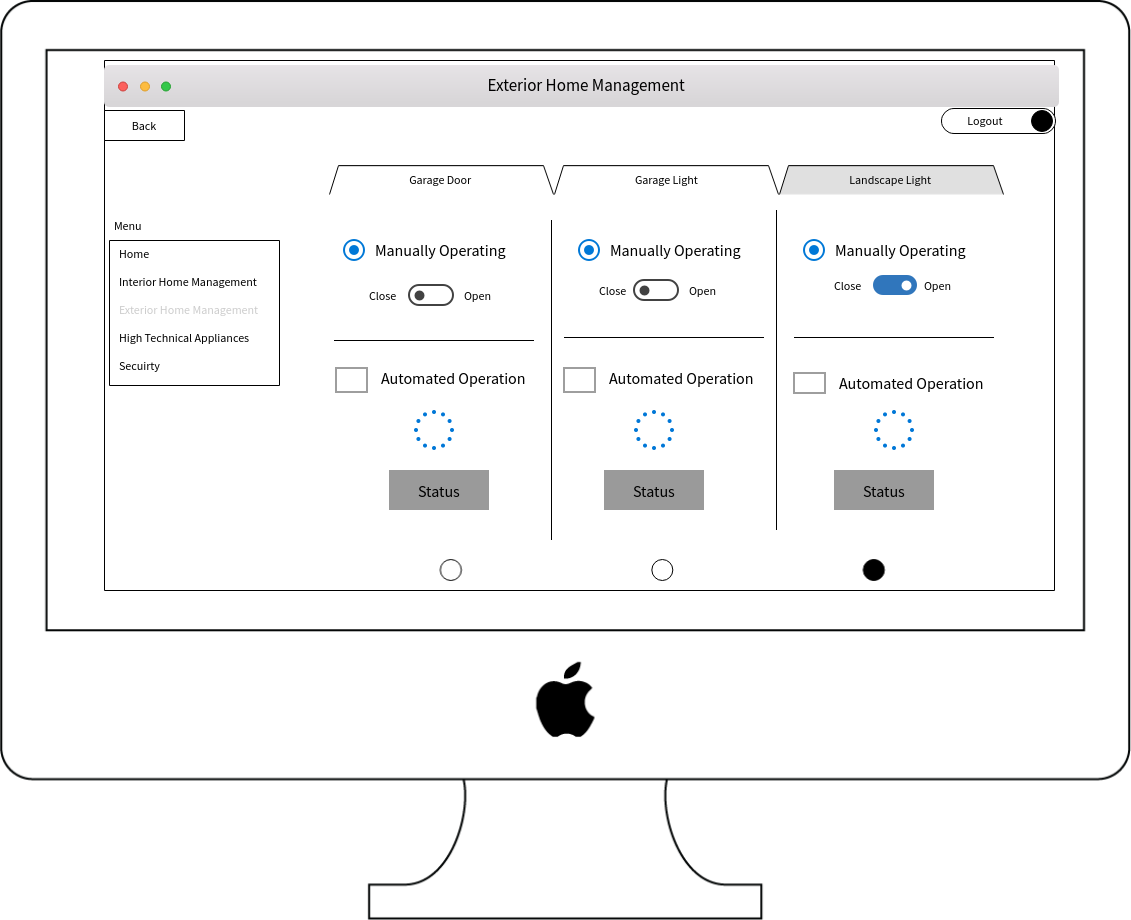
### User interfaces



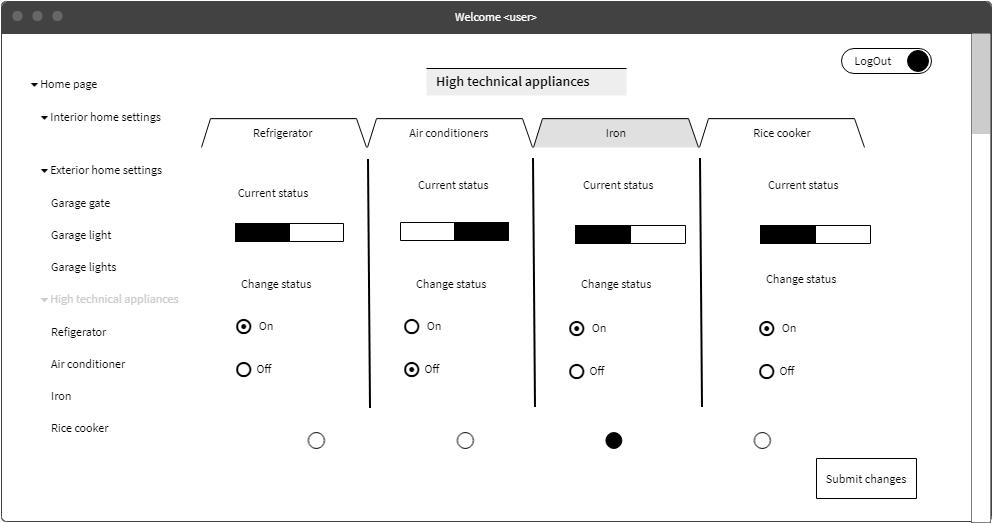
**Figure 2 : Login Page**



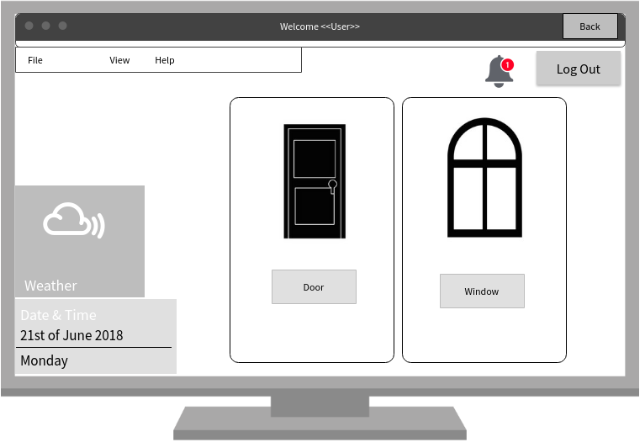
**Figure 3 : Home Page**



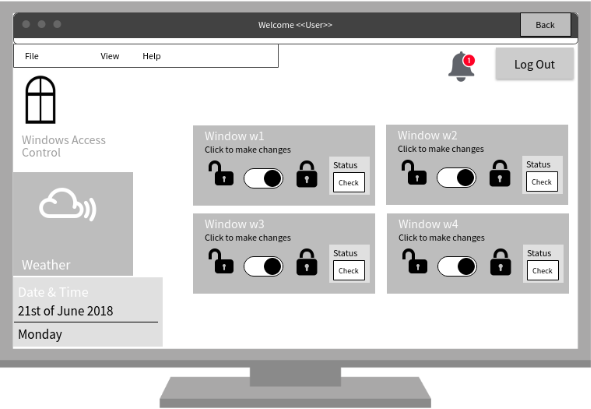
**Figure 4 : Exterior Home Management Interface.**



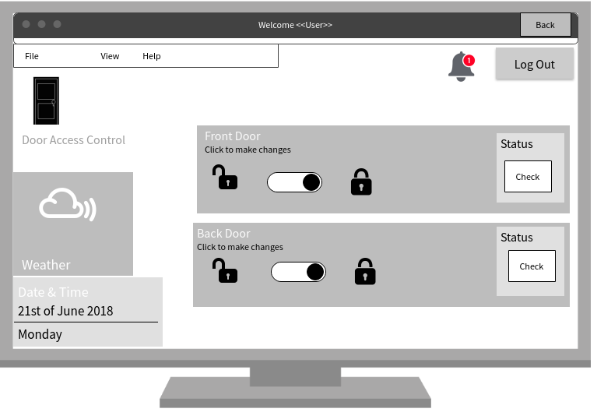
**Figure 5 : High technical Appliance interface**



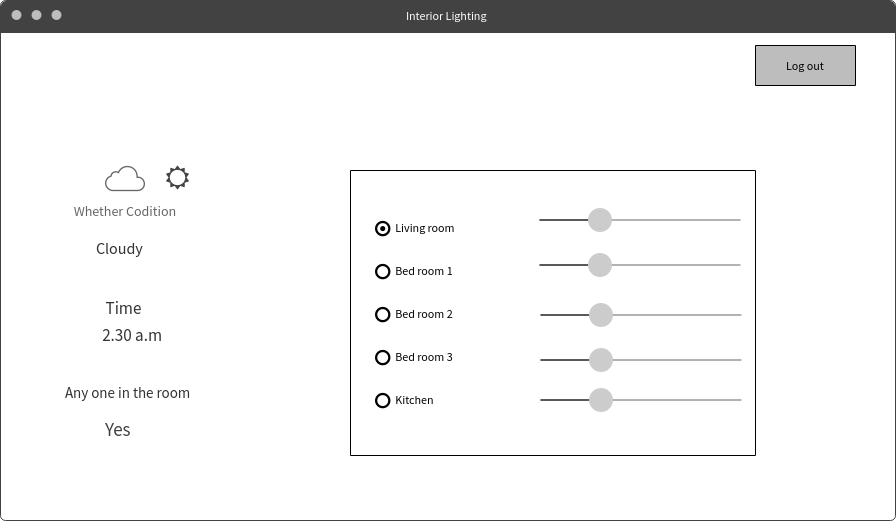
**Figure 6 :Main access Control interface**



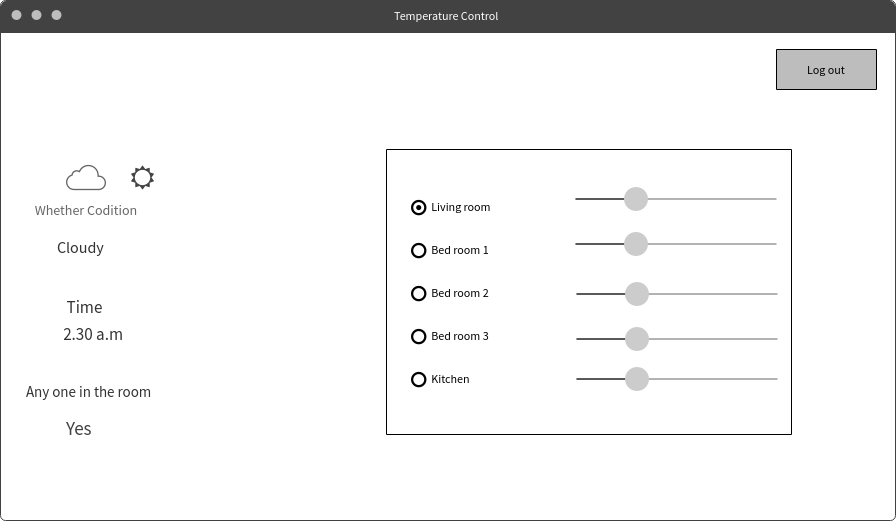
**Figure 7** **: Window remote Control Interface**



**Figure 8 :Door remote control interface**



**Figure 9 : Interior lightning system interface**



**Figure 10 :Temperature controlling system interface**

* + 1. **Hardware interfaces**
* **Raspberry Pi**

A cheap and small computer device when compared to normal computers use in day to day life. This device should connect to a monitor to get the display to control the functions. Raspberry pi[8] is the micro controller use in the proposed home automation system in order to control all the IOT devices.

* **Door/Window locking sensor**

This sensor is going to relate to the Raspberry pi using a relay circuit to control the doors and windows remotely. Sensor will attach to the door lock and it will relate to the Raspberry pi over the Internet using Wi-Fi. Sensor will react to the requests that are coming from the Raspberry pi and make changes in the door lock as per user wish.

* **Photo-Resistor (Light Sensor)**

Also, known as Light-dependent Resistor (LDR) or Photo-cell are considered as light sensitive devices which is used to indicate the presence or absence of light.

In the dark, the Resistance is high, measures up to 1MΩ; but when the Resistor is exposed to Light, the Resistance Drops to about a few ohms.

* **Light Sensor Module**

Light Sensor Modules are known as photoelectric devices because it converts light energy (Photons) into electrical (electrons) signals.

Light Sensors is a passive device and is commonly known as “Photoelectric Device”.

* + The Working Voltage is 3.3 – 5 V
  + The Output current >= 15 mA, which can directly light up an LED light.
  + The Output from the Light Sensor is Digital Switching (High or Low voltage on pin D) and Analog Signals (Voltage Output on pin A).

Photoconductive Light Sensor are Semi-conductor devices that use light energy to control the flow of electrons and henceforth the current flowing through them. They are commonly called as Light Dependent Resistors (LDR)

* + 1. **Software interfaces**
* Apache

To host the web application of the proposed home automation system apache server is going to be use. Apache server is well recommended by most of the web developers and considered as secure, reliable and fast.

* My SQL

My SQL will be use as the database of the web application since My SQL is considered as the most popular open source database in the world for web-based application.

* Bootstrap

Since the web application is making at first in proposed home automation system, bootstrap use to make it responsive to the mobile application.

* VNC Viewer [9]

VNC viewer software acts as a graphical interface tool to control the raspberry pi micro controller.

* + 1. **Communication interfaces**

Mobile and web application should have an unbroken communication with the cloud to send and receive data to control access of the home automation system. TCP/IP protocol is using for the internal communication while HTTP protocol is using for the communication via internet.

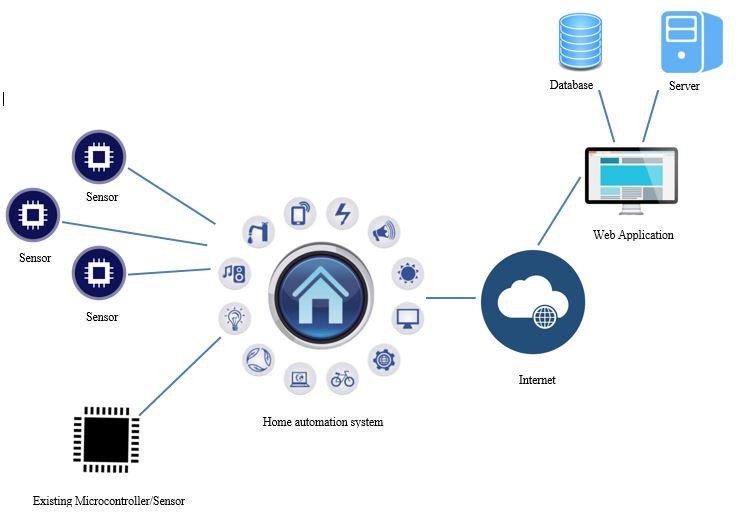
To maintain network connection WIFI module will be used and mobile data may use as per user’s wish to connect mobile application to the network.

## Architectural Design

### High level Architectural Design

Software Architecture is also known as High Level Software Design, this is the first design step after all the requirements for the System is analyzed. The goal is to define a software structure that fulfills the functional and non-functional requirements.

This diagram, allows the reader and user of the document to orient themselves to the design and flow of the functions.



**Figure 11 : High Level System Architecture.**

### Hardware and software requirements with justification

* **Hardware Requirements Raspberry Pi**
  + Low Cost, basic computer that runs entirely on open source software.
  + The reason we selected Raspberry pi rather than Arduino as a development board, is because Raspberry Pi is compatible, uses minimum power and voltage supply.
  + Another reason is, our functionalities involve Image Processing, and it requires OpenCV, which is not supported by Arduino.

**Relays**

* + An Electrical Operated Switch.
  + Necessary to control the circuit with low-power signal.

**Breadboard**

* + A solderless device for temporary prototype with electronics and test circuit designs.
* **Software Requirements IDE for Raspberry Pi**

Raspberry has a wide range of IDE that provide programmers with excellent interfaces to develop the source code, applications and systems

Languages used for the web application development: PHP, JavaScript, HTML 5

* WAMP Server – Hosting the web application.
* Cloud Server – To access it through public network.
* MYSQL – Data Storage

### Risk Mitigation Plan with alternative solution identification

Risk Mitigation planning is the process of developing and introducing options and actions to enhance opportunities and reduce threats to the project objectives. To handle Risk Mitigation, the options available are; Accept, Avoid, Control, Transfer, and Monitor.

As our system control and handles the safety of Homes, it is vital that the System has an effective Risk Mitigation Procedure, to provide comfort, security, and peace of Home for the home owners.

**Table 18 : Risk Mitigation Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Impact** | **Probability** | **Plan** |
| System is breeched | * Loss of Data * Intrudes have a high chance of breaking and entering. * User has no way of monitoring the house. | 45% | * System will switch to Lockdown in 1 minute. * Alert notifications are sent accordingly. * Identify the root cause of the breech. * Ensure the cause does not get repeated in the future. |

|  |  |  |  |
| --- | --- | --- | --- |
| Camera Failure | Intruders have opportunity to enter the home  Garage door automatic opening and closing will not function | 55% | * Alert the User * Identify the root cause of the breech. * Ensure the cause does not get repeated in the future. |
| Sensors | Burnt Sensors; Expenses to buy or repair the damaged sensor.  Functionality will not work, automatically. | 40% | * Alert the User * Identify the root cause of the breech. * Ensure the cause does not get repeated in the future. |
| High Voltage appliances | Circuit burns, leading to Fire and or Injuries  Functionality will not work, automatically. | 78% | * Alert the User, as Critical. * Identify the root cause of the breech. * Ensure the cause does not get repeated in the future. |
| Server Downtime | Data loss  System will be dark, giving opportunity for intruders. | 45% | * Have Quality website monitoring. * Notify the user * Have backup hosting provider with instant TTL times. * Identify the root cause of the breech. * Ensure the cause does not get repeated in the future. |

|  |  |  |  |
| --- | --- | --- | --- |
| Internet Connection Lost | Data loss  Opportunity for intruders to hack the system | 85% | * Back up the data; i.e. Auto save the data. * Notify the user; on the connection situation * Update the user with the notifications & alerts |

### Cost Benefit Analysis for the proposed solution

**Table 19 : Cost Benefit Analysis Table**

|  |  |
| --- | --- |
| **Cost** | **Benefit** |
| Raspberry Pi | * Low Cost Computer * Convenient * Portable * Solid State Storage * Consumes Low power * Expansion Capabilities * Built in HDMI capable graphics |
| Sensors | * Requires small voltage and power for its operations * Fast and provides immediate output. * Available in various sizes and packages |
| Cameras | * Compatible to any platforms * Low Cost * High-Definition images * Adjustable lens, hence ability to fine tune the focus control |

## Performance Requirements

A system will meet its performance targets only when its specified clearly and unambiguously.

**Response Time**

The response time for a system should be between 0.1 second to 1.0 second; this is the range limit for the user’s flow of thoughts to stay uninterrupted and avoid receiving any special feedback due to the delay.

**Workload**

This is described as the scenarios the User is likely to encounter and execute. The performance of the system is dependent on how the load is delivered to the system. **Scalability**

The increase in the systems workload that the system should be able to process.

This application will be developed as a responsive application and therefore will be running on a mobile device or desktop computer. When compared with a desktop computer, the processing power of a mobile device is low. Due to this reason, the application should be developed in a way which it will consume minimum number of resources work efficiently.

The main target is to make the application to run smoothly and efficiently and provide the user an accurate and quick result making it a very reliable application to use in day to day activities.

“Home Automation System” also has the added and Unique ability to integrate and synchronize other sensors or devices currently being used in the house of the user**.**

## Design Constraints

There are no Design Constraints to this system. The Development team will create a suitable design and work on it.

## Software system attributes

The requirements in this section specify the required reliability, availability, security and maintainability of the software system.

### Reliability

System should function well per the specifications under normal operating environment without any crashes. System should be tested with multiple users accessing the application to ensure reliability, and the probability of failure is minimal.

### 3.5.2. Availability

Smart Home Automation System should be available to the user at any time. User will be able to browse the application and control appliances any time and from anywhere.

The User should receive alerts and notifications to the mobile phone and can give out commands from anywhere and at any time, if a strong internet connection is available.

The Sensors and Micro-controller should be connected and active throughout the time

### Security

The Home owner should be authenticated with username and password. When a guest is staying in the House, the guest will be provided with another set of credentials, these will have limited access to the House.

Mobile phone security plus application passwords can be used externally. All communications are open to internet so future modification should be done after integrating and launching of this version of the product.

### Maintainability

Maintainability is defined as the probability of performing a successful repair action within a given time. The proposed application will be easily maintained because application is developed per the object-oriented principals and modularization. Also, the source code will be well commented and documented for any changes or modifications done in future.

Proper coding standards and name conventions, along with best Practices will be used.

# Supporting information

## References

1. “Software design document” En.wikipedia.org [Online]

Available: https://en.wikipedia.org/wiki/Software\_design\_description. [Accessed: 08- May- 2018].

1. "Unified Modeling Language", En.wikipedia.org, 2018. [Online].

Available: https://en.wikipedia.org/wiki/Unified\_Modeling\_Language. [Accessed: 11- May- 2018].

1. Muhammad Asadullah, Ahsan Raza “An overview of home automation systems” [Online]

Available:htt[ps://www.rese](http://www.researchgate.net/publication/311923032_An_overview_of_home_)a[rchgate.net/publication/311923032\_An\_overview\_of\_home\_](http://www.researchgate.net/publication/311923032_An_overview_of_home_) automation\_systems [Accessed :11-May-2018]

1. “Best Smart Home Gadgets of 2018” [Online]

Available: htt[ps://www.tomsguide.com/us/best](http://www.tomsguide.com/us/best-smart-home-gadgets%2Creview-2008.html)-[smart-home-gadgets,review-2008.html](http://www.tomsguide.com/us/best-smart-home-gadgets%2Creview-2008.html) [Accessed: 12-May-2018]

1. “Raspberry Pi” En.wikipedia.org, [Online]

Available: https://en.wikipedia.org/wiki/Raspberry\_Pi [Accessed: 13-May-2018]

1. “OpenCV” En.wikipedia.org, [Online]

Available: https://en.wikipedia.org/wiki/OpenCV [ Accessed:12-May-2018]

1. Light dependent resistors, [Online]

Available: <http://www.resistorguide.com/photoresistor/>[Accessed: 12-May-2018]

1. Anon, (2018). [eBook] Available at: https:/[/www.omega.com/tempe](http://www.omega.com/temperature/z/pdf/z110-)r[ature/z/pdf/z110-](http://www.omega.com/temperature/z/pdf/z110-) 114.pdf [Accessed 14 May 2018].
2. Anon, (2018). [eBook] Available at: <http://www.ijsdr.org/papers/IJSDR1604038.pdf> [Accessed 15 May 2018].
3. Mozilla Hacks – the Web developer blog. (2018). How to build your own private smart home with a Raspberry Pi and Mozilla’s Things Gateway – Mozilla Hacks - the Web developer blog. [online]

Available at: <https://hacks.mozilla.org/2018/02/how-to-build-your-own-private->smart- home-with-a-raspberry-pi-and-mozillas-things-gateway/ [Accessed 15 May 2018].

1. The MagPi Magazine. (2018). VNC: Beginner's guide to Raspberry Pi remote access. [online]

Available at: <https://www.raspberrypi.org/magpi/vnc-raspberry-pi/> [Accessed 15 May 2018]

## Appendices

-