# HOME AUTOMATION SYSTEM TO MANAGE HOME EQUIPMENTS AND APPLIANCES

**Project ID: 18-088**

**Design Document**

W.S.H Boteju. IT14034114

Bachelor of Science Special (Honors) in Information Technology Department of Software Engineering

Sri Lanka Institute of Information Technology Sri Lanka

May 2018

# Declaration

#### Author declare that this is author’s own work and this DD 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 author’s 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.

**Author:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Registration No | Signature | Date |
| W.S.H Boteju | IT14034114 |  | 2018.05.06 |

**Table of Contents**

[Declaration](#_bookmark0) 2

[Table of Contents](#_bookmark1) 3

1. [Introduction 5](#_TOC_250016)
   1. [Purpose 5](#_TOC_250015)
   2. [Scope 5](#_TOC_250014)
   3. [Definitions, Acronyms, and Abbreviations 6](#_TOC_250013)
   4. [Overview 7](#_TOC_250012)
2. [Overall Descriptions](#_bookmark2) 8
   1. [Product perspective 8](#_TOC_250011)
      1. [System interfaces 8](#_TOC_250010)

2.1.2 User interfaces .......................................................Error! Bookmark not defined.

* + 1. [Hardware interfaces 9](#_TOC_250009)
    2. [Software interfaces 9](#_TOC_250008)
    3. Communication interfaces ...................................Error! Bookmark not defined. 2.1.6 Memory constraints ..............................................Error! Bookmark not defined. 2.1.7 Operations .............................................................Error! Bookmark not defined. [2.2 Product functions](#_bookmark3) [10](#_bookmark3)

[2.2.1 Use case scenarios](#_bookmark4) 10

* 1. [User characteristics 11](#_TOC_250007)
  2. [Constraints 11](#_TOC_250006)
  3. [Assumptions and dependencies 11](#_TOC_250005)

1. [Specific Requirements 12](#_bookmark5)
   1. [External interface requirements](#_bookmark6) 12
      1. [User interfaces 12](#_TOC_250004)
      2. [Hardware interfaces](#_bookmark7) 12
      3. [Software interfaces](#_bookmark8) 13
      4. [Communication interfaces](#_bookmark9) 13
   2. [Architectural Design 13](#_TOC_250003)
      1. [High level architectural design](#_bookmark7) 13
      2. [Hardware and software requirements with justification](#_bookmark8) 14
   3. [Performance requirements 15](#_TOC_250002)
   4. [Design constraints 15](#_TOC_250001)
   5. [Software system attributes 16](#_TOC_250000)
      1. [Reliability](#_bookmark7) 16
      2. [Availability](#_bookmark7) 16
      3. [Security](#_bookmark7) 16
      4. [Maintainability](#_bookmark7) 16
2. References 17
3. **Introduction**

### Purpose

Purpose of the design document is in order to give a team overall guidance to the architecture of the software project. An SDD usually accompanies an architecture diagram with pointers to detailed feature specifications of smaller pieces of the design. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

### Scope

In home automation system we have a web application and hardware components. Web application is a responsive application implemented using bootstrap framework. User can control home appliances via web application.

Sensors are connected with the Raspberry Pi to detect environmental changes, events. System notifies the user with the changes. According to the changes user can control the home appliances.

Scope of this document is to give a clear idea of controlling high technical appliances.

To have a clear idea of the content this document divided into several sections. Overview section will describe beneficiaries of the project, main tasks and goals. In addition to that supporting software and hardware components, techniques and technical feasibilities, constraints and dependencies are clearly described in this document.

### Definitions, Acronyms, and Abbreviations

*Definitions*

|  |  |
| --- | --- |
| Term | Definition |
| 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. |

Table 1.3.1: Definitions.

*Acronyms and Abbreviations*

|  |  |
| --- | --- |
| SDD | Software Design Document |
| IoT | Internet of Things |
| API | Application Program Interface |
| GSM | Global System for Mobile communications |

Table 1.3.2: Acronyms and Abbreviations.

### Overview

Home automation provides high end solutions and uses advanced digital technology to automate our products and systems. Home automation is one of the most vital topics under research today. Many researchers have come up with different solutions such as Bluetooth based home automation, Voice based home automation, GSM based home automation etc. But it was not easy to reach the requirements of a user accurately. When comparing with other existing systems, we are planning to implement a system which satisfies the user, with having all the requirements in one application.

Project application is a responsive web app where users can access the application with any device such as mobile phone, desktop, tablet etc. User can control the home appliances remotely by connecting with the application. Hardware components such as Raspberry Pi, Relays, and sensors are connecting with the app via home Wi-Fi.

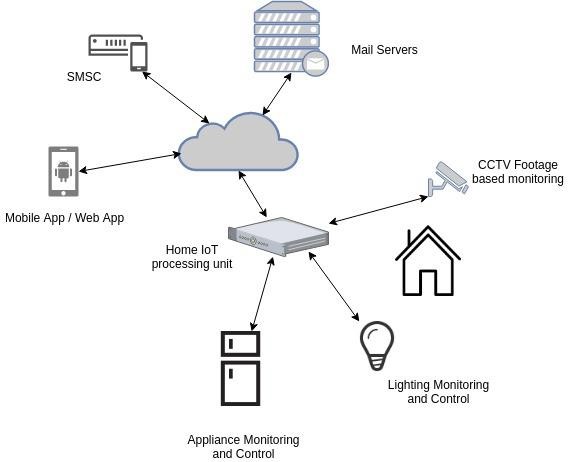


Figure 1.0: Overview of the system

1. **Overall Descriptions**

### 2.1 Product perspective

Home automation system includes controlling high technical appliances. Examples for high tech appliances are refrigerator, rice cooker, washing machine, Air conditioners etc.

According to this function it protects appliances from damage due to Power failures / Lighting, Monitoring the status of the high technical devices, controlling the basic actions of home devices. Monitoring surveillance feeds and hence, acts accordingly.

When concerning about the high technical appliances, lot of high tech appliances are expensive and consuming more power. So there should be a way to protect them from thundering, instant power failures, avoid circuit burns and also to control remotely

##### System interfaces

Since this is a web based responsive application system interface basically align with users operating system communication with the web server. Since the application is set to work on any Windows/Android/IOS platform at the moment these protocols will apply to the system interface communication channel.

##### User interfaces

Proposed system is a web responsive application. All user interfaces are described in detailed in section 3.1.1.

Key user interfaces,

* + - Main user interface
      * Login of the responsive web application.
    - Secondary user interface
      * Select the automation section.
    - Final user interface

Device monitoring and controlling interface.

##### Hardware interfaces

Hardware requirements will need to run the developed application without having any problem. For the designing, implementation and testing purposes we have identified few hardware requirements such as,

-Mobile phones – Android OS 5.0 or later

-Desktop /Laptop

-Tablet

##### Software interfaces

These software components, mainly target to use in creating applications,

* + - WebStorm
    - IDE for Raspberry pi

##### Communication interfaces

Mobile device or personal computer should connect with the Wi-Fi or mobile data to connect the application. Home appliances should connect with the Raspberry Pi to communicate with the application.

* + 1. **Memory constraints**
    - 1 GB RAM or higher mobile phone/PC

##### Operations

* + - Normal operations
      * User can login to the application by using username and password.
      * Will be able to select the controlling section of the home.
      * Able to monitor and control home devices/appliances via the application.
    - Special operations
      * Notify user when instant power failures.
      * Determine which appliance needs to switch on back.
      * Detect circuit burnings.

### 2.2 Product functions

##### 2.2.1 Use case scenarios

|  |  |
| --- | --- |
| **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 2.2.1 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 2.2.2 Use case scenario for view current status of appliances

### 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 crashes 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 the entire system will crashed. So System should wire accordingly when deploying 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 needs lot of sensors to detect a certain event.

### Assumptions and dependencies

Assuming the mobile devices has enough memory and power available for sending and obtaining data. Another assumption is that a continuous internet connection is available for the user. Assuming home appliances and system components are always connected with home Wi-Fi.

## Specific Requirements

### 3.1 External interface requirements

##### User interfaces

User interfaces are required to analyze the user friendliness of the application. User interacts with the system via interfaces. All functional requirements are implemented with limited number of interfaces to

reduce complexity and to make the application much more user friendly.

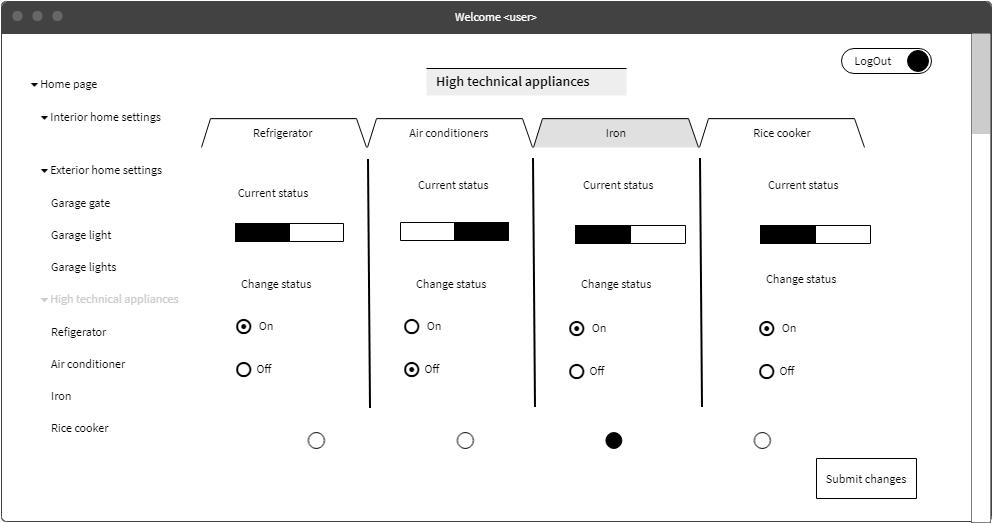


Figure 3.1.1.1 Control high technical appliances.

All functional requirements for managing high tech appliances are implement in this interface.

##### Hardware interfaces

* Raspberry pi
* Relays to control 120-240V devices
* Breadboard
* Wires
* Sensors

##### Software interfaces

* Windows 10 Operating System
* WebStorm
* IDEs for Raspberry Pi

### Communication interfaces

Internet,

If user wants to perform any activity that requires internet, there should be proper internet connection.

* 1. **Architectural Design**

##### High level Architectural Design

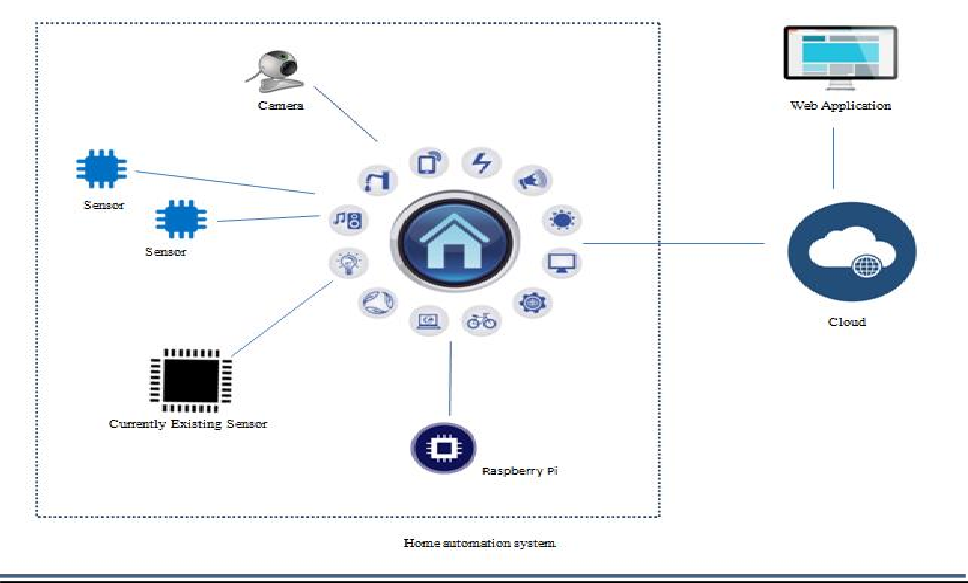


Figure 3.2.1.1 High level architectural diagram

##### Hardware and software requirements with justification

* Hardware requirements
* 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. We are planning to use Raspberry pi as the development board except Arduino as the development board because we have image processing section. For image processing it is important to have OpenCv, Arduino does not support OpenCv.

* Relays

A relay is an electrically operated switch. Relays are used where it is necessary to control a circuit by a separate low-power signal, or where several circuits must be controlled by one signal.

* Breadboard

A breadboard is a solderless device for temporary prototype with electronics and test circuit designs. Most electronic components in electronic circuits can be interconnected by inserting their leads or terminals into the holes and then making connections through wires where appropriate.

* Software requirements
* IDE for Raspberry

Raspberry Pi has a wide range of IDEs that provide programmers with good interfaces to develop source code, applications and system.

* 1. **Performance requirements**

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.

### Design constraints

In system design, a design constraint refers to some limitation on the conditions, when the system is developed, or on the requirements of the system. During system design, it is as important to identify each design constraint as it is to elicit requirements since the design constraints place an overall boundary around the system design process.

* Implementation constraints

Web responsive application is target to implement using HTML 5, JavaScript, and Python.

* Time Factor

This will be done as a partial requirement to the degree under the guidance of SLIIT. Therefore, in dealing with projects artifacts it is strictly advised to follow the given schedules and deadlines.

Therefore, dealing with the time limitations has become one of the major concerns among the development team in the process of designing.

* Human resource factor

This module will be done as a group project. Even though the sub modules have been divided among group members it’s very difficult to come to a conclusion when carrying out some work since there are many more ideas generated by the group members. Finding an appointment to have group meetings and to meet supervisor have become another problem with the busy schedules of everyone.

### Software system attributes

##### Reliability

System should function well according to the specifications under normal operating environment without any crashes. System should be tested with multiple users accessing the application to ensure reliability.

##### Availability

Home 365 automation applications should have been available at any time. User will be able to browse the application and control appliances any time.

##### Security

Users have done the authentication when using the web application. 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. To control high technical appliances, user first needs to login to the system, So unauthorized people cannot access the application.

##### 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 according to the object-oriented principals and modularization. Also, the source code will be well commented and documented for any changes or modifications done in future.

## 3 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].
2. 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_automati)a[rchgate.net/publication/311923032\_An\_overview\_of\_home\_automati](http://www.researchgate.net/publication/311923032_An_overview_of_home_automati) on\_systems [Accessed :11-May-2018]
3. “Best Smart Home Gadgets of 2018” [Online]

Available: https://[www.tomsguide.com/us/best-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 ]