SMART HOME

Assignment 1-2 SRS Report

Abstract

This document drafts the software requirements and specifications of smart home project.

Abhisha Bhesaniya

Software Engineering Principles

Table of Contents

1 Introduction

1.1 Purpose

The purpose of this document is to ease the process of software design for Smart Home. This document will sketch all the requirements, specifications, and features to give the reader a complete understanding of Smart Home. The main reason behind creating this system is, We are living in the era where each one of us wants to control everything with his/her smartphone and that is why our customers want to automate his home with the latest technology, security is their priority but along with security they want to control their home appliances like TV, AC, washing machine, etc...

Intended Audience:

- Executives
- Customers
- Project Managers
- Software Developers
- Software Tester
- Hardware Tester

1.2 Scope

This project provides a real-time update of your home to identify if your windows or doors are open or closed when you are not at home. It also detects the presence of humans using the PIR sensor. You can also control your home appliances using the mobile application. All these sensors send data using the BLE protocol. You can check the status of your home using the mobile application.

By using this system, you can keep an eye on your home as well as control your home appliances from any corner of the world. As you have control of your appliances through your phone you can save electricity by turning off the appliance when you are not at your home.

1.3 Definitions, Acronyms, and Abbreviations

SRS:	Software Requirement Specifications
IEEE:	Institute of Electrical and Electronics Engineers
UI:	User Interface
PIR:	Passive Infrared Sensor
BLE:	Bluetooth for Low Energy
LE:	Low Energy
SHP:	Smart Home Project
Cloud:	The remote server hosted on the internet to store, manage, and process the data
Event:	When the state of PIN of microprocessor or microcontroller change

1.4 References

- IEEE (1998). IEEE Guide for Software Requirements Specifications. IEEE Std 830-1998
- https://conestoga.desire2learn.com/d2l/le/content/354354/viewContent/7302418/View
- https://senior.ceng.metu.edu.tr/2013/tranquillum/documents/srs.pdf

1.5 Overview

The rest of the SRS report is split up into two parts overall description and specific requirements of the smart home project.

The overall Description section of this SRS will give over the idea of this project. This section of the SRS will describe features and design, it will also contain what assumptions have been taken about the end-user and what constraints have been placed on the system. Upon completion of this section, the reader should have a thorough understanding of the smart home project.

The Specific Requirements section of this SRS is written by taking technical users in mind. The targeted audience of this section is the hardware and software developer. This section of the SRS report will describe design requirements, features in a technical manner, hardware requirements, expected hardware, and performance of hardware and software components.

2 Overall Description

2.1 Product Perspective

SMARTHOME is a generic solution for home automation enhanced with machine learning techniques. It uses a well-known and widely used wireless protocol BLE for communication between devices and allows you to collect data and control home appliances from a single web and mobile interface. The central processor of the smart home is raspberry pi. And the box which contains this processor is Coordinator Box. This central processor has all decision-making power. This processor has on board the Wi-Fi chip which is connected with home Wi-Fi router and using this Wi-Fi router's internet raspberry pi will update data on the online server and from there the user can access all the data. This is how raspberry pi will send data to the end-user.

2.1.1 System Interface

Users can send command of changing names of the sensor, sensor number, and no of receiving a message when events occur using user application. Then the coordinator box will send that command to a particular sensor selected by the user. Otherwise, the sensor will send data to the coordinator box and the coordinator box will send this data to the user.

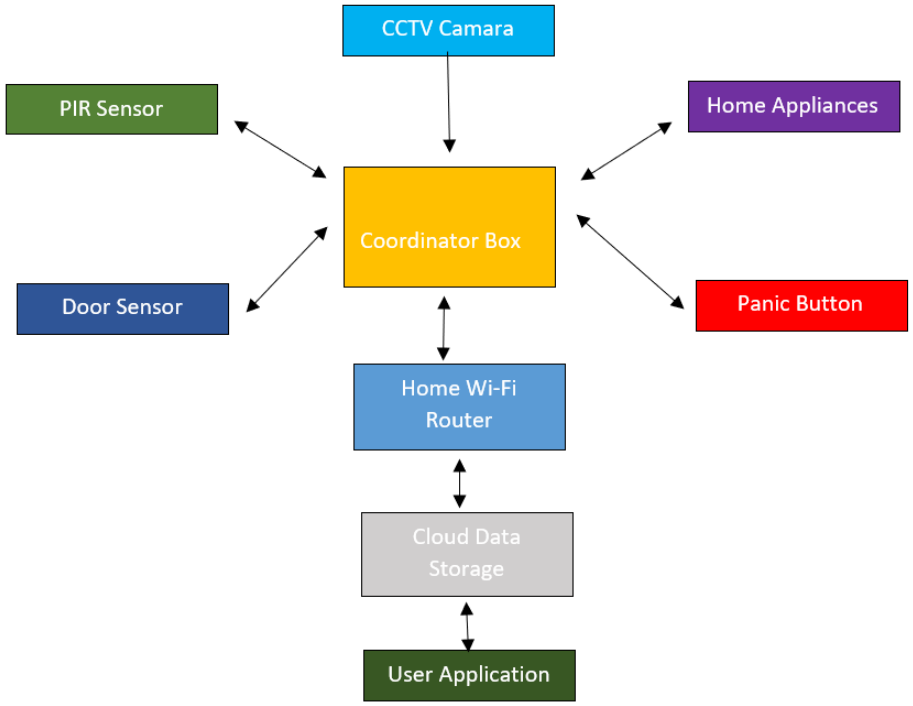

Figure 1 Simple overview of System Interface for Smart Home

2.1.2 User Interface

There are 3 user interfaces through which the user can interact with the smart home system. The first one is a mobile application in the user's phone, second is computer application in computer or laptop and the third is interface display at home.

On the user interface display, there will be 6 main menus. The 6 main menus for the smart home project are home, security, Appliances, Panic, PIR, and CCTV. These main menus will allow access to several customized options.

Figure 2 Sample picture of User Interface Home panel

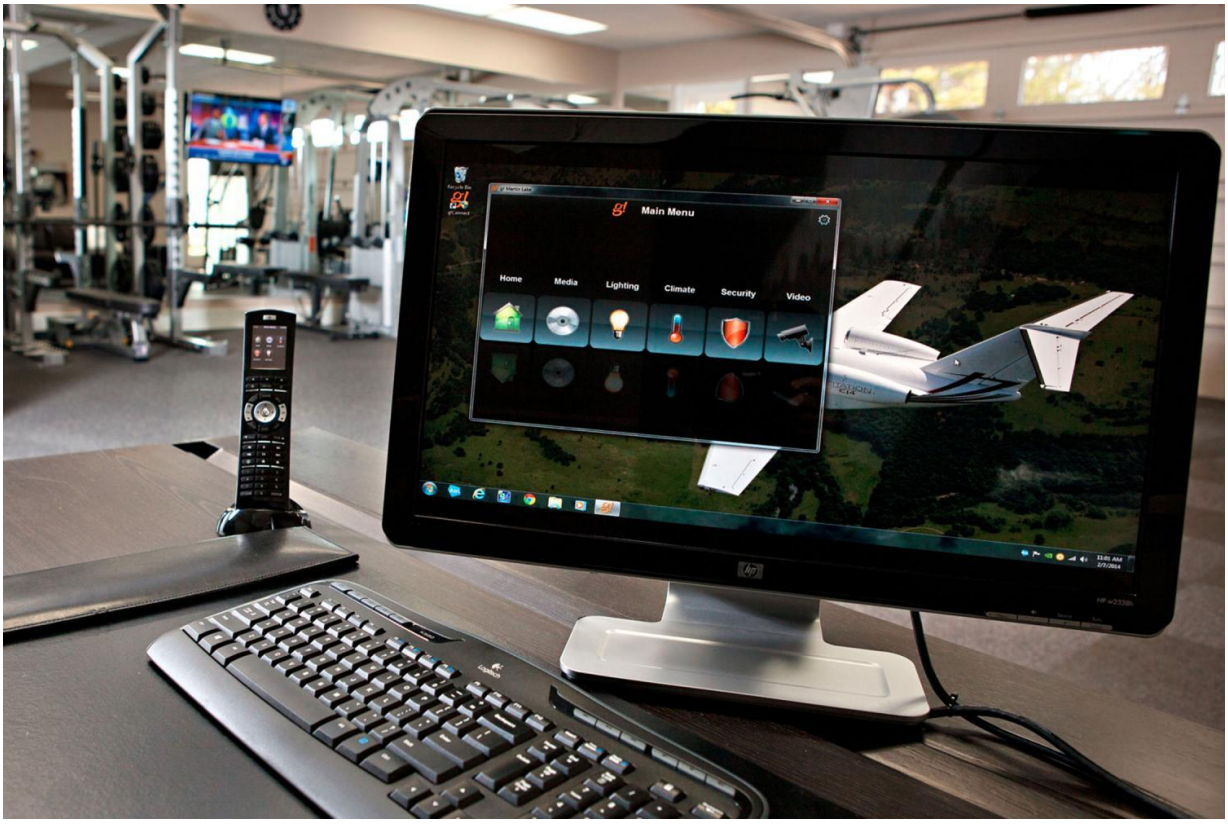

Figure 3 Sample picture of User Interface Computer Application

Figure 4 Sample picture of User Interface Mobile Application

2.1.3 Hardware Interface

Raspberry pi is the microprocessor through which the software will interface with hardware. This microprocessor will control all the other hardware including the display panel. The software will also interface with BLE and Wi-Fi device drivers to send and receive data.

2.1.4 Software Interface

The smart home project software will control all the peripherals connected with raspberry pi. Along with this, it will also control MY SQL database to store and retrieve use and different sensor's data. There will be mobile and computer applications to receive data and to set the sensor's name and other parameters.

2.1.5 Communication Interface

- 802.11 IEEE Wi-Fi Communication Protocol
- Bluetooth 5.0 LE

2.1.6 Memory Constraints

Raspberry pi is a central processor for this project, and it has its OS called Raspbian which can be load into the memory card. So, to avoid memory constraints the memory card which will be used will be 64GB in size. Moreover, we will use the cloud to store sensor and user data so, there will be no memory constraints.

2.1.7 Operations

The operation modes for the Smart home are listed below.

- **General operation mode:** After seating everything in our system will work in the general mode in which you can receive the status of the door sensor, PIR, Panic, live footage of CCTV, and Appliances status. There will be an option to enter another mode in this mode.
- **User set setup mode:** In this mode, the user can set name and every other parameter of devices by sitting in his home through display penal.
- **User Remote set up mode:** In this mode, users can set parameters of sensors through mobile or computer applications from outside as well as from inside of home too.
- **Administration mode:** This mode can be accessed by the system developer only. Whenever there is any update in software and user wants upgraded system at that time developer will use this mode.

2.1.8 Site Adaptation Requirements

To install this system on-site and to work properly there should be a Wi-Fi router on site. Other than that, there is no need for a separate power plug for every sensor as all the sensors are battery operated.

2.2 Product Functions

In our Smart Home project, the user indicates the person purchasing the product to take advantage of home security and energy efficiency for his/her home. The user has the right to choose a device, turn it on & off, getting & updating the sensor info of the device, and adjust it according to the configuration's types of the device.

When the PIR sensor will detect the presence of human it will send a command to raspberry pi to turn on the light of that area. Whenever there is an event on the sensor it will send data of event to raspberry pi and raspberry pi will update the data on the cloud and if there is any output control with this event it will perform this too. The below diagram shows the main functionalities of a smart home.

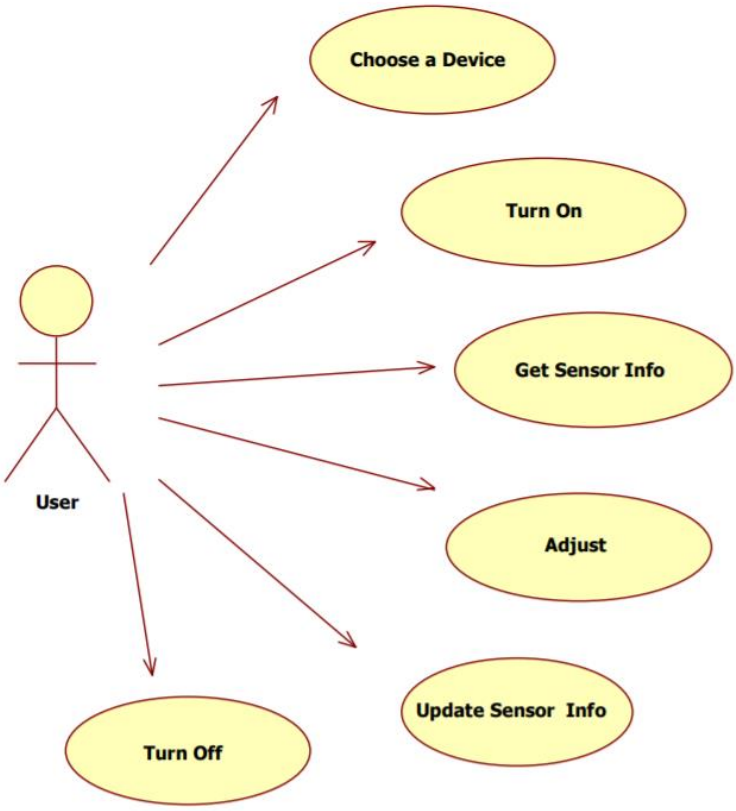

Figure 5 Functional Diagram

2.3 User Characteristics

Users should know how to use android application. Nowadays this is not a problem as everyone is using a smartphone. But, the need training for how to set name and other parameters of sensors as well as how to control home appliances from remotely.

2.4 Constraints

- Must use Bluetooth 5.0 LE version of Bluetooth to last battery for a long time
- Must use IEEE 802.11 Wi-Fi communication protocol
- Must use Raspberry PI 3 or Raspberry PI 4 as these two series contain onboard Bluetooth and Wi-Fi chip.
- The mobile Application size must be within 100MB
- The Computer Application size must be within 1GB
- The dimensions of the Display must not exceed then normal Tablet size.

2.5 Assumptions and Dependencies

- User has Compatible smartphone
- User has a high-speed Wi-Fi connection at home

2.6 Apportioning of Requirements

In future releases, our system will allow connecting wireless sensors other then BLE protocol. We can also add voice command to give access to users to access menus of UI.

3 Specific Requirements

3.1 External interfaces

This section will describe the inputs and outputs of the smart home project.

3.1.1 Input

Input name: Door Sensor
Description of purpose: Provides status of the door
Source of input: Door sensor senses the status of door and sends it to the raspberry pi microprocessor through BLE
Relationship to other inputs/outputs: Display status of the door
Format of input: Formatted data (sensor number, sensor type and status of door 1 for open and 0 for close)

Input name: Window Sensor
Description of purpose: Provides status of the window
Source of input: Door sensor senses the status of the window and sends it to the raspberry pi microprocessor through BLE
Relationship to other inputs/outputs: Display status of Window
Format of input: Formatted data (sensor number, sensor type and status of window 1 for open and 0 for close)

Input name: PIR Sensor
Description of purpose: Detect the presence of human
Source of input: PIR sensor senses the presence of human and sends it to the raspberry pi microprocessor through BLE
Relationship to other inputs/outputs: Turn on and off the light of that particular premises
Format of input: Formatted data (sensor number, sensor type and status of PIR sensor 1 for the presence of human and 0 for the absence of human)

Input name: Panic Button
Description of purpose: Alert of emergency
Source of input: Panic button sends a signal to the raspberry pi microprocessor through BLE. When someone presses the button
Relationship to other inputs/outputs: send an alert message on UI

Format of input: Formatted data (sensor number, sensor type, and status of PIR sensor 1 for alert and 0 otherwise)
Input name: CCTV Camara
Description of purpose: To keep eye on home for security purpose
Source of input: record video and send it to the raspberry pi microprocessor through BLE
Relationship to other inputs/outputs: Upload live footage on cloud and from there users can access videos through UI.
Format of input: Formatted data

Input name: turn on AC
Description of purpose: To turn on AC on a particular setting
Source of input: Any of the UI
Relationship to other inputs/outputs: Turn on the AC and Display status of the machine on UI
Format of input: Formatted data

Input name: turn on washing machine
Description of purpose: To turn on washing machine on a particular setting
Source of input: Any of the UI
Relationship to other inputs/outputs: Turn on the washing machine and Display status of the machine on UI
Format of input: Formatted data

Input name: turn on the dishwasher
Description of purpose: To turn on the dishwasher on a particular setting
Source of input: Any of the UI
Relationship to other inputs/outputs: Turn on the dishwasher and Display status of the machine on UI
Format of input: Formatted data

Input name: Set Parameters
Description of purpose: To configure sensor data like sensor Name, sensor Type, no of time message should be sent after any event
Source of input: Any of the UI
Relationship to other inputs/outputs: Update in data of string being send through BLE Like sensor type, sensor name
Format of input: Formatted data

Input name: System Mode
Description of purpose: To set the mode of a system like general operation mode, user setup mode, user remote setup mode and administration mode
Source of input: Any of the UI
Relationship to other inputs/outputs: Update in data of string being send through BLE Like sensor type, sensor name
Format of input: Formatted data

Input name: Receive Logs
Description of purpose: send data of a particular sensor from the database to UI.
Source of input: Any of the UI
Relationship to other inputs/outputs: Display the requested data on UI.
Format of input: Formatted data

Input name: Setting
Description of purpose: this menu contains a variety of basic settings to control the screen of Smart home penal like text size, text color, brightness, layout, automatic display turnoff time, etc.
Source of input: Any of the UI
Relationship to other inputs/outputs: Display the change as per input on UI.
Format of input: Formatted data

Input name: Turn on or off Screen
Description of purpose: to turn on or off the UI penal screen
Source of input: Any of the UI
Relationship to other inputs/outputs: Display on or off as per input.
Format of input: Formatted data

Input name: Remote Request
Description of purpose: to access the system from mobile or computer application
Source of input: Mobile or computer application
Relationship to other inputs/outputs: Display the requested data on UI
Format of input: Formatted data

3.1.2 Output

Output name: Door open
Description of purpose: send an alert of the door open on UI
Source of output: All the UI
Relationship to other inputs/outputs: Display the alert message on UI
Format of output: Formatted data

Output name: Window open
Description of purpose: send an alert of the window open on UI
Source of output: All the UI
Relationship to other inputs/outputs: Display the alert message on UI
Format of output: Formatted data

Output name: PIR on Detect
Description of purpose: control the lights of that area
Source of output: the light of that area
Relationship to other inputs/outputs: turn on the light of that area
Format of output: Formatted data

Output name: Panic Button
Description of purpose: send an alert of emergency on All the UI
Source of output: All the UI.
Relationship to other inputs/outputs: Display the emergency alert message on UI
Format of output: Formatted data

Output name: Turn ON AC
Description of purpose: to turn on AC at a particular temperature
Source of output: AC
Relationship to other inputs/outputs: turn on AC of that area and display the status of the machine on UI.
Format of output: Formatted data

Output name: Turn OFF AC
Description of purpose: to turn OFF AC at a particular temperature
Source of output: AC
Relationship to other inputs/outputs: turn OFF AC of that area and display the status of the machine on UI.
Format of output: Formatted data

Output name: Turn ON washing machine
Description of purpose: to turn on washing machine at a particular setting
Source of output: washing machine
Relationship to other inputs/outputs: turn on the washing machine and display the status of the machine on UI.
Format of output: Formatted data

Output name: Turn off washing machine
Description of purpose: to turn off washing machine
Source of output: washing machine
Relationship to other inputs/outputs: turn on the washing machine and display the status of the machine on UI.
Format of output: Formatted data

Output name: Turn ON Dishwasher
Description of purpose: to turn on the dishwasher at a particular setting
Source of output: dishwasher
Relationship to other inputs/outputs: turn on the dishwasher and display the status of the machine on UI.
Format of output: Formatted data

Output name: Turn OFF Dishwasher
Description of purpose: to turn off the dishwasher
Source of output: dishwasher
Relationship to other inputs/outputs: turn off the dishwasher and display the status of the machine on UI.
Format of output: Formatted data

Output name: Database Receive
Description of purpose: to display data of sensors on UI.
Source of output: UI
Relationship to other inputs/outputs: take data from the cloud database and display on UI.
Format of output: Formatted data

3.2 Functions

This section contains the use cases for the smart home project. These use cases are written by keeping in mind the user is a household user of the Smart home system.

Use case: Mode selection
Precondition: the system should be on and in default mode
Trigger: user will change from current mode to another mode selected by the user using smart home UI
Successful scenario: Raspberry pi will change from the current model to the requested mode.
Possible exceptions: N/A

Use case: Door/ Window sensor state change
Precondition: Door should be connected with raspberry pi's on bord Bluetooth
Trigger: when someone open or close door
Successful scenario: Raspberry pi will send data to the onboard Wi-Fi chip to update data in the cloud through a home Wi-Fi router.
Possible exceptions: onboard Wi-Fi fails to connect with Home Wi-Fi -> try for reconnecting and once it is done update the data on the cloud so that the user can access data and get alerts if the door is open.

Use case: PIR sensor state change
Precondition: PIR should be connected with raspberry pi's on bord Bluetooth
Trigger: when a human enters in that area
Successful scenario: PIR sensor sends data to Raspberry pi and it will send data to onboard Wi-Fi chip to update data in the cloud through a home Wi-Fi router.
Possible exceptions: onboard Wi-Fi fails to connect with Home Wi-Fi -> try for reconnecting and once it is done update the data on the cloud so that users can access data and raspberry pi will turn on the light of that area.

Use case: Panic Button state change
Precondition: Panic Button should be connected with raspberry pi's on bord Bluetooth
Trigger: when human Press the panic button
Successful scenario: Panic button sends data to Raspberry pi and it will send data to onboard Wi-Fi chip to update data in the cloud through a home Wi-Fi router.
Possible exceptions: onboard Wi-Fi fails to connect with Home Wi-Fi -> try for reconnecting and once it is done update the data on the cloud so that users can access data and get alert.

Use case: Washing Machine state change
Precondition: Washing machine's control circuit's BLE should be connected with raspberry pi's on bord Bluetooth
Trigger: when user change status from any of the UI
Successful scenario: Raspberry pi will send the command to the Washing machine's BLE to drive the relay and turn on or off the machine then will send data to onboard Wi-Fi chip to update data in the cloud through a home Wi-Fi router.
Possible exceptions: onboard Wi-Fi fails to connect with Home Wi-Fi -> try for reconnecting and once it is done update the data on the cloud so that users can access data.

Use case: Dishwasher state change
Precondition: Dishwasher's control circuit's BLE should be connected with raspberry pi's on bord Bluetooth
Trigger: when user change status from any of the UI
Successful scenario: Raspberry pi will send the command to the dishwasher's BLE to drive the relay and turn on or off the machine then will send data to onboard Wi-Fi chip to update data in the cloud through a home Wi-Fi router.
Possible exceptions: onboard Wi-Fi fails to connect with Home Wi-Fi -> try for reconnecting and once it is done update the data on the cloud so that users can access data.

Use case: Air Conditioner state change
Precondition: Air conditioner's control circuit's BLE should be connected with raspberry pi's on bord Bluetooth
Trigger: when user change status from any of the UI
Successful scenario: Raspberry pi will send the command to the Air conditioner's BLE to drive the relay and turn on or off the machine then will send data to the onboard Wi-Fi chip to update data in the cloud through a home Wi-Fi router.
Possible exceptions: onboard Wi-Fi fails to connect with Home Wi-Fi -> try for reconnecting and once it is done update the data on the cloud so that users can access data. The user enters out of range temperature range -> system will enter the closest value which is in the limit.

Use case: Remote Access
Precondition: Smart home application is installed in mobile
Trigger: when a user requests a change from the mobile or computer application
Successful scenario: Raspberry pi will receive the command from Wi-Fi and set the data as required.
Possible exceptions: onboard Wi-Fi fails to connect with Home Wi-Fi -> try for reconnecting and once it is done update the data on the cloud so that users can access data. the smartphone is not connected with the local network ->we can't proceed further if there is no internet available.

Use case: Enter Password
Precondition: the system should be in default mode
Trigger: when the user enters the password from any of the UI
Successful scenario: Raspberry pi will compare the password stored before and if it matches user can access every functionality.
Possible exceptions: Password mismatch->display message on UI entered password is wrong to enter the password again.

Use case: Password Change
Precondition: the system should be in use setup mode
Trigger: when user select change password option from any of the UI
Successful scenario: Raspberry pi will add a new password in the database and ask for a login again.
Possible exceptions: Password mismatch->display message on UI entered password is not matching with the password with the reentered password tab.

Use case: Touch Menu Icon on UI
Precondition: Smart home UI screen should be on
Trigger: when user touch menu icon from any of the UI
Successful scenario: Raspberry pi will perform the function
Possible exceptions: N/A

Use case: Maintenance Task Alert
Precondition: Smart home system should be turned on
Trigger: when the battery level is low
Successful scenario: Raspberry pi will display the message on UI screen
Possible exceptions: N/A

Happy Path Scenario
When everything is going with the flow:

- A smart home UI panel will be responsive and easy to use.
- A smart home system will update the status of every sensor at regular intervals.
- A smart home system will allow us to change the sensor's parameter.
-

3.3 Performance Requirements
- Minimum 10 BLE sensors should be connected at a time with raspberry pi' s Bluetooth
- Button response of the smart home UI penal must be less than 1 second
- Command send to any sensor should have less than 0.5 second response time
- 13 should be the size of text display on smart home UI penal
- Power consumption must be as low as possible
- Android or iOS app should fit in all size of phone and tablet with 1290x920 resolution

3.4 Logical Data Base Requirements

MYSQL database must be used in a smart home project for database-related tasks.

This database must consist of data into this format date, time, sensor name, sensor no, and value of the sensor when there is an event, or it should update data every 5 minutes.

When users make any change in the sensor in the user set up mode or user remote set up mode it must be stored in the database in this format date-time sensor name, sensor no, changed data.

Every change which is made in any part of the system should be in the record for at least 3 years and the sensor log should be there as per user requirements.

3.5 Design Constraints

- Raspberry pi 3 or 4 must be your central processor
- The software of Smart home must be written in python
- Smart home UI panel must have a touch screen with backlight
- Every sensor must go into sleep mode after sending data until the next event to save battery
- Every sensor must use BLE protocol
- MYSQL must be used for database

3.6 Software System Attributes

- **Reliability:** The software developed for the smart home project must perform all the function under any critical situation without failure
- **Availability:** Any update in smart home project software should be accessible for every previous system if the user wants to upgrade which is already installed.
- **Security:** software should not allow any other software which has the same protocol but from other company to enter into the system.
- **Maintainability:** software for the smart home project should follow layered software structure, industry programming standard. And it must use an industry-standard version control system which will make the program more maintainable.
- **Portability:** Smart Home project program should be portable through the flash memory of the existing smart home device.

4 Appendices

N/A