**Smart Fire Alarm System**

System / Software Requirements Specifications (SRS)

By: Jovian Sanjaya(P2233570), Muhammad Irsyhad (P2221779),

Elvin La Weng Hoe(P2221836),Nicholas Lee (P2204596)

**Table of Contents**

[Document Version 3](#_3znysh7)

[1. Purpose 4](#_2et92p0)

[1.1. Intended Audience 4](#_tyjcwt)

[1.2. Intended Use 4](#_3dy6vkm)

[1.3. Scope 4](#_1t3h5sf)

[1.4. Definitions and Acronyms 4](#_4d34og8)

[2. Overall System Description 5](#_2s8eyo1)

[2.1. Use Case Diagrams 5](#_17dp8vu)

[2.2. System Architecture 6](#_3rdcrjn)

[2.3. Functional Requirements 7](#_26in1rg)

[2.3.1.](#_lnxbz9) Initialisation and Fire Sensors 7

[2.3.2. Response](#_35nkun2) To The Fire Outbreak 8

[2.3.3.](#_1ksv4uv) Manual Functions by the Users 8

[2.4. Non-Functional Requirements 9](#_44sinio)

[2.4.1.](#_2jxsxqh) LCD and Keypad for ease of use 9

[3. Software Architecture 10](#_z337ya)

[3.1. Static Software Architecture 10](#_3j2qqm3)

# **Document Version**

| No | Update | Name | Date | Version |
| --- | --- | --- | --- | --- |
| 1. | Initial version | Smart Fire Alarm Systems | 27/12/2023 | 1.0 |
| 2. | Removed and changed some requirements. | Smart Fire Alarm Systems | 14/2/2024 | 2.0 |

# **Purpose**

## **Intended Audience**

This SRS document describes the System Requirements and Software Design for an IoT Coffee maker, and the target audience are System and Software Engineers working on the development of this project.

## **Intended Use**

The SRS defines the overall System Architecture and Requirements as well as the Software Architecture and Design. This document also contains the definition of the System Requirements which shall be used as the input for System Test cases and Software Unit Test cases.

## **Scope**

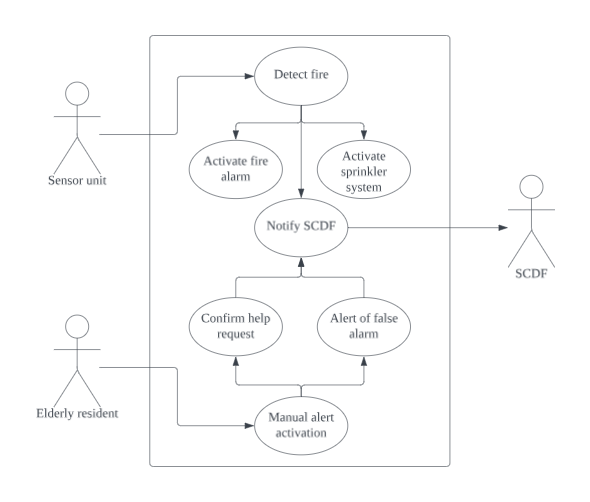
The Internet of Things (IoT) system aims to continually monitor senior citizens' houses for early fire signals in order to promptly alert the SCDF of any fire breakouts so they may take appropriate action and further inform the locals about the fire.

## **Definitions and Acronyms**

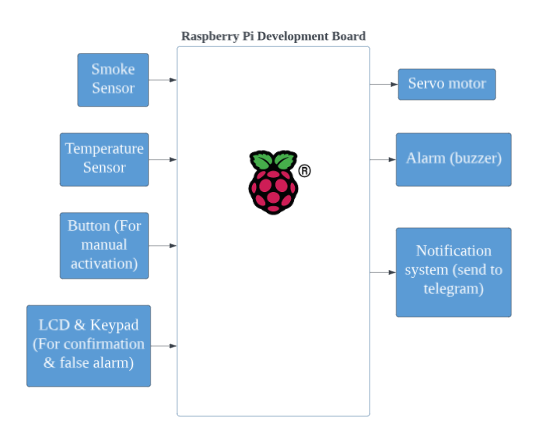
| **Acronym** | **Description** |
| --- | --- |
| IR | Infra-Red |
| LED | Light Emitting Diode |
| NFC | Near Field Communication |
| SW | Software |
| HW | Hardware |
| SCDF | Singapore Civil Defence Force |

# **Overall System Description**

## **Use Case Diagrams**



## **System Architecture**

****

**Remarks**

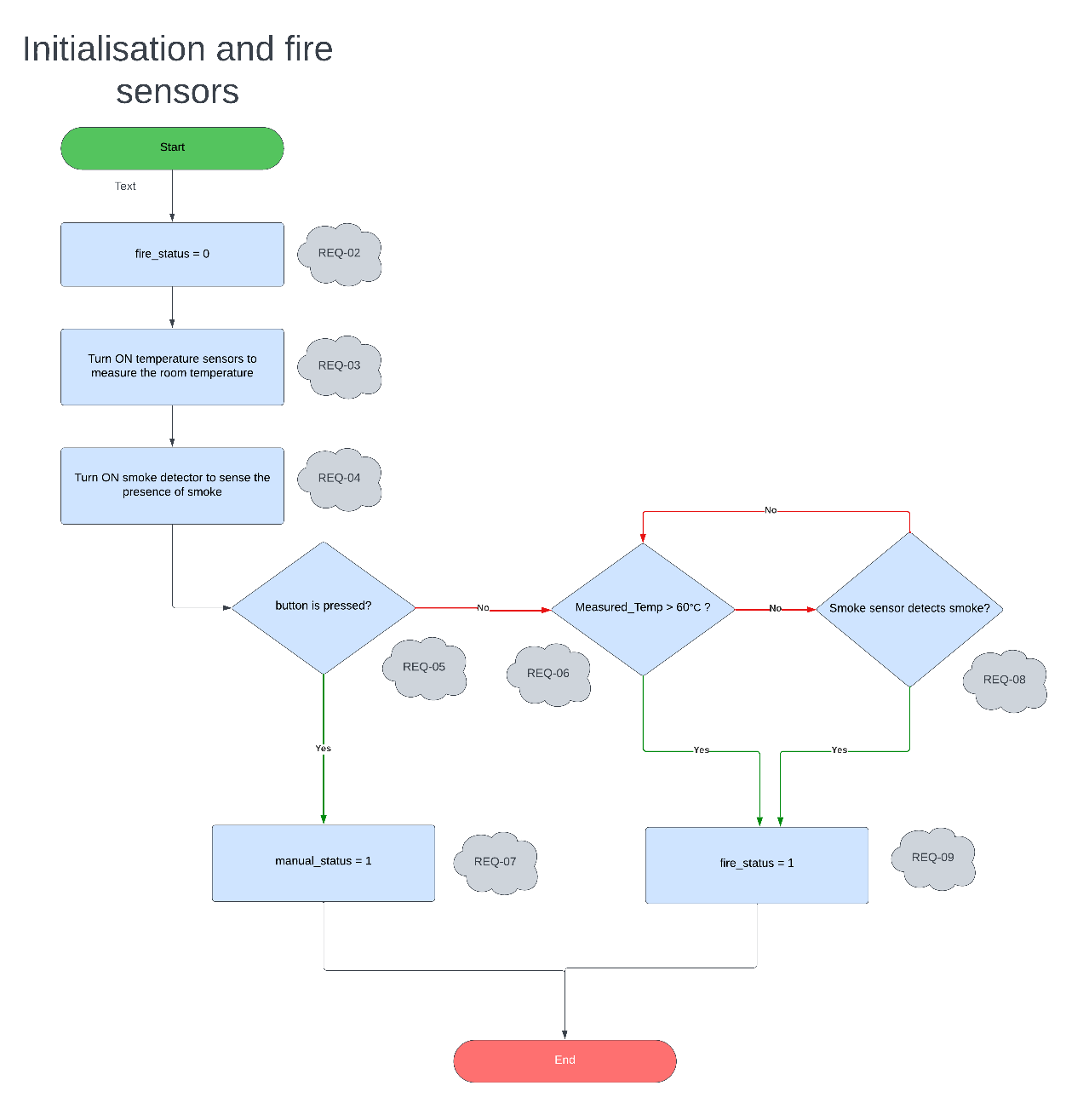
**The Smoke Sensor used for this project is InfraRed sensor.**

## **Functional Requirements**

### **Initialisation and Fire Sensors**

The first functional requirement is initialisation and fire stations. In this function, the system will start up and check if the user manually notifies SCDF by pressing the button or if fire breaks are detected by the sensors.

| **REQ\_ID** | **Requirement** |
| --- | --- |
| REQ-01 | Upon starting up, the flowchart in the figure 1 shall be executed. |
| REQ-02 | The fire status is set to 0, meaning there is no fire detected. |
| REQ-03 | The temperature sensor is turned on to measure the room temperature. |
| REQ-04 | The smoke detector is turned on to sense the presence of smoke. |
| REQ-05 | If the button is pressed, the manual\_status is set to 1.Otherwise, the temperature sensor will measure the surrounding temperature. |
| REQ-06 | If the measured temperature is greater than 60°C, then the fire status is set to 1, meaning a fire is detected. |
| REQ-07 | The manual\_status is set to 1 when button is pressed |
| REQ-08 | If the smoke sensor detects smoke, then the fire status is set to 1, meaning a fire is detected. |
| REQ-09 | If both the measured temperature is greater than 60°C and the smoke sensor detects smoke, then the fire status is set to 1, meaning a fire is detected. |

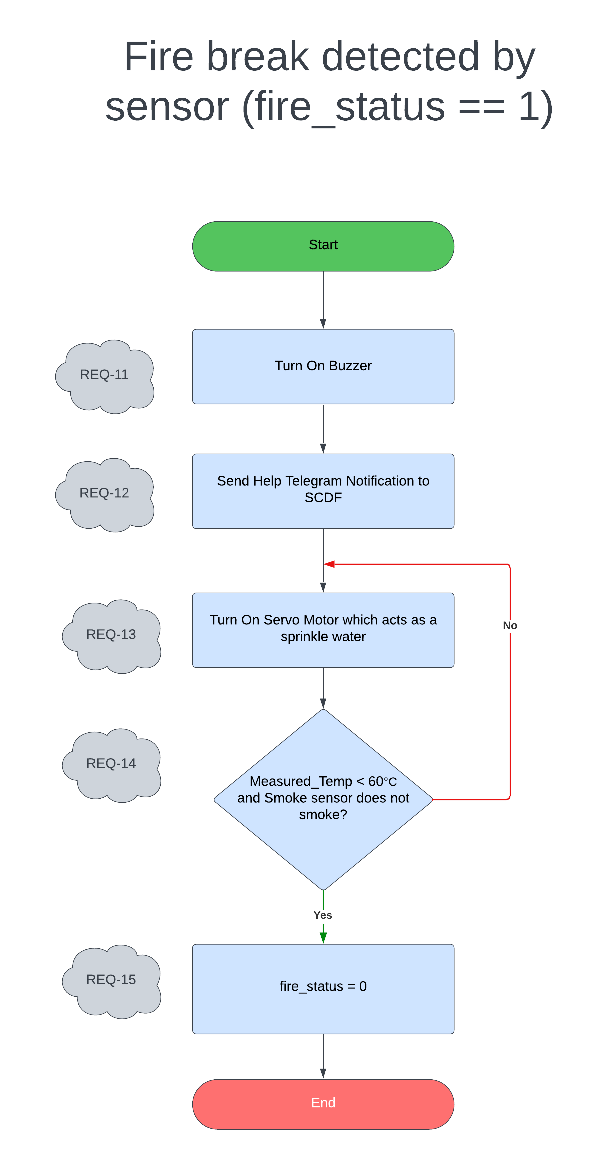


**Figure 1**

### **Response To the Fire Outbreak**

The second functional requirement is Response To the Fire Outbreak where if there is fire detected by the sensor (measured temperature is above 60 degree celsius and/or smoke is detected by smoke detector), servo motor which acts as a sprinkle water will turn on.

| **REQ\_ID** | **Requirement** |
| --- | --- |
| REQ-10 | When fire\_status == 1, the flowchart figure 2 shall be executed. |
| REQ-11 | The buzzer will turn on when fire\_status is set to 1. |
| REQ-12 | Telegram Notification will be sent to SCDF to alert them about the fire. |
| REQ-13 | Servo motor will be turned on which acts as a sprinkle of water. |
| REQ-14 | If the measured temperature is below 60 °C and the smoke sensor does not detect smoke, fire\_status is set to 0. Otherwise, the servo motor will be on. |
| REQ-15 | After fire is set off, fire\_status is set to 0. |



**Figure 2**

### **Manual Function by the Users**

The last functional requirement is Manual Function by the Users. If the users press the button, the system will ask to confirm the users to send notification to SCDF. If the user confirms, the system will send help notification to SCDF.

| **REQ\_ID** | **Requirement** |
| --- | --- |
| REQ-16 | When manual\_status == 1, the flowchart figure 3 shall be executed. |
| REQ-17 | User confirms fire break: press "1" to alert, "2" to cancel. |
| REQ-18 | User cancels fire break: press "2" to stop, else stay here. |
| REQ-19 | Alert SCDF for fire: sends emergency notification if confirmed |
| REQ-20 | User confirms to send false alarm: press “1” to send false alarm. |
| REQ-21 | User cancels false alarm: press “2” to cancel, the alert proceeds and returns to start. |
| REQ-22 | Alert SCDF for false alarm: sends notification if user confirms mistake. |
| REQ-23 | After the SCDF is notified (emergency or false alarm), the manual\_status is set back to 0. |

A diagram of a diagram

Description automatically generated with medium confidence **Figure 3**

A black text on a white background

Description automatically generated

**Figure 4**

## **Non-Functional Requirements**

### **LCD and Keypad for ease of use**

LCD and keypad are used for convenience of the users to show status of the system and to avoid misunderstanding of both parties from false alarm. From figure 4, it can be seen the keypad usage in this smart fire alarm system.

| **REQ\_ID** | **Requirement** |
| --- | --- |
| REQ-24 | LCD Screen to show the following upon starting up the system  Line 1 = “1.Activate alert”  Line 2 = “2.Cancel alert” |
| REQ-25 | If the user presses on a button that is not 1 or 2, display the following  Line 1 = “Invalid Choice.”  Line 2= “Choose 1 or 2” |
| REQ-26 | If the user decides to cancel alert, display the following  Line 1 = “Alert Cancelled” |
| REQ-27 | If a fire is not detected BUT the user has pressed the button to contact SCDF, and activate alert, display the following  Line 1 = “Send false alarm?”  Line 2 = “1. Yes 2. No” |
| REQ-28 | If the user decides to press the button to proceed with sending false alarm, display the following  Line 1 = “Sending false”  Line 2 = “alarm” |
| REQ-29 | If the user decides to press the button to continue with the alert, contacting SCDF,  Display the following  Line 1 = “Alert remains.” |

# **Software Architecture**

## **Static Software Architecture**

The Software Architecture defines the various Software Components that are developed to realize the implementation of the system requirements.

**Hardware**

**hal\_temp\_humidity\_sensor.py**

**hal\_servo.py**

**dht11.py**

**hal\_buzzer.py**

**hal\_input\_switch.py**

**hal\_ir\_sensor.py**

**hal\_LCD.py**

**hal\_keypad.py**

**Software**

**resident\_alerts.py**

**scdf\_tele\_notification.py**

**manual\_alert.py**

**test\_scdf\_tele\_notification.py**

**test\_main.py**

**test\_resident\_alerts.py**

**main.py**

**display\_graph.py**

**false\_alarm.py**

**Group Contributions**

Jovian

File

1. display\_graph.py (thingspeak)
2. false\_alarm.py
3. scdf\_tele\_notification.py (telegram)
4. test\_scdf\_tele\_notification.py
5. SRS document - Requirements
6. Creating Docker image and container and push it to docker hub
7. extra features
8. fire\_demo.py (for test case)
9. fire\_alarm\_system.py (for test case)
10. Github READme

Branch

1. Jovian
2. Master
3. Implementation

Irsyhad

File

1. SRS document - use case diagram
2. resident\_alerts.py on autoalert branch
3. manual\_alert.py on implementation branch - made changes to fit main.py
4. main.py on implementation branch - combined code from manual\_alert.py and resident\_alerts.py
5. sprint planning template

Branch

1. autoalert
2. implementation
3. master

Nicholas

File

1. manual\_alert.py
2. scdf\_tele\_notification (Modified the conditions for false alarm)
3. devop.mp4- video
4. SRS document
5. System Test Report (Did Tests for REQ-01-REQ-16)

Branch

1. master
2. Nicholas
3. Implementation

Elvin

File

a)main.py (Modify the code for test cases)

b)test\_main.py

c)SRS document (Updating the SRS document by removing and changing requirements and flowcharts)

d)System Test Report (Did System Tests for REQ-17 to REQ-29 and filled up the report.)

Branch

a)implementation

b)master