IoT Smart Fire Alert – System/ Software Requirements Specification (SRS)

Table of Contents

[1. Purpose 2](#_Toc174724249)

[1.1 Intended Audience 2](#_Toc174724250)

[1.2 Intended Use 2](#_Toc174724251)

[1.3 Scope 2](#_Toc174724252)

[1.4 Definitions and Acronyms 3](#_Toc174724253)

[2. Overall System Description 4](#_Toc174724254)

[2.1 Use Case Diagram 4](#_Toc174724255)

[2.2 System Architecture 5](#_Toc174724256)

[2.3 Functional Requirements 6](#_Toc174724257)

[2.3.1 Main Script 6](#_Toc174724258)

[2.3.2 HMI and Configuration 6](#_Toc174724259)

[2.3.2 Automatic Fire Detection 8](#_Toc174724260)

[2.3.3 Manual SOS Switch 11](#_Toc174724261)

[2.3.4 Fire Alarm Deactivation Key (False Alarm) 12](#_Toc174724262)

[2.3.5 Alarm 13](#_Toc174724263)

[2.3.6 SCDF Notification System 14](#_Toc174724264)

[2.3.7 Sprinkler System 14](#_Toc174724265)

[3. Software Architecture 15](#_Toc174724266)

[4. System Test Cases 17](#_Toc174724267)

# Purpose

## 1.1 Intended Audience

This SRS document outlines the System Requirements and Software Design for a Smart Fire Alert System. The primary users are the individuals living independently in homes or rental apartments in Singapore. Designed for individuals who may have limited mobility or other health considerations that require clear fire alerts.

## 1.2 Intended Use

This SRS document describes the overall System Architecture, Functional Requirements and Software Architecture and Design. The intended use of the system is to continuously monitor and inform the elderly of a fire in their homes with an effective alarm system and alert the local fire department about the fire.

## 1.3 Scope

Input

1. Automatic Fire Detection System
   * Continuous monitoring of temperature and light intensity using sensors to identify a fire breakout.
2. Manual SOS Switch
   * In the event of an emergency where residents need to alert SCDF for urgent help required, this switch can be used.
3. Fire Alarm Deactivation System
   * This will only allow authorized personnel to turn off the fire alarm should there be a false activation of the fire alarm.

Output

1. HMI LCD Interface

* For configuration, threshold adjustment and debugging.

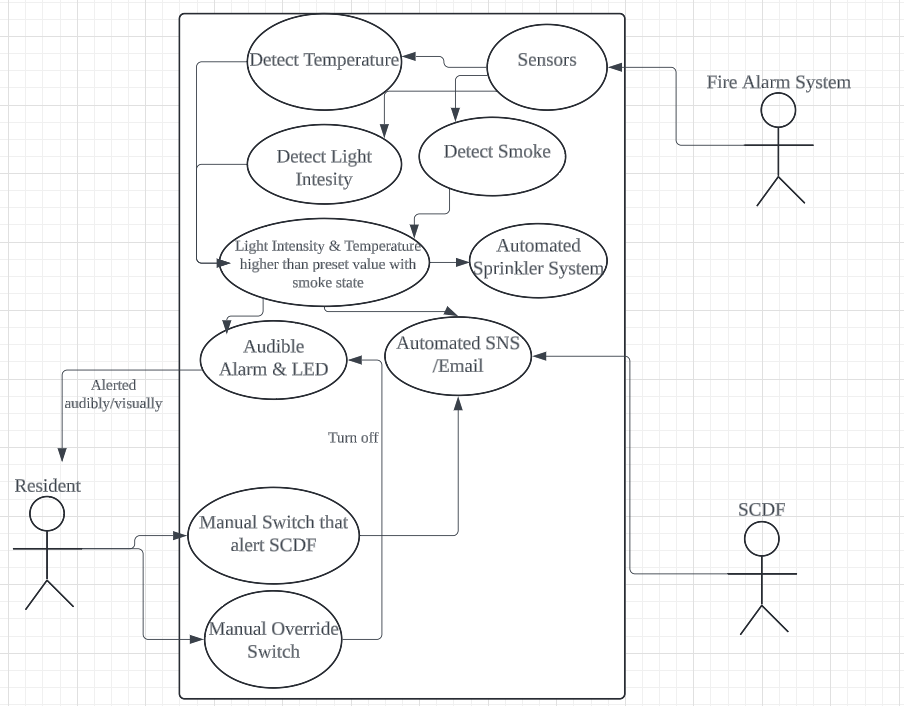
1. Alarm System
   * Alarm to alert the residents living in the area to evacuate to safety.
2. Notification System
   * Automated notification system to request help from SCDF with clear instructions.
3. Sprinkler System
   * To pump water to the sprinklers to suppress the fire and minimize any further fire spread.

## 1.4 Definitions and Acronyms

|  |  |
| --- | --- |
| **Acronym/Abbreviation** | **Definition** |
| IR | Infra-Red |
| LED | Light Emitting Diode |
| LCD | Liquid Crystal Display |
| SCDF | Singapore Civil Defense Force |
| SW | Software |
| HW | Hardware |
| SMS | Short Message Service |
| Config | Configure/Configuration |
| 2 SF | 2 Significant Figures |
| SOS | “Urgent Help Needed” |
| GPIO | General Purpose Input/Output |
| HMI | Human-Machine Interface |

# 2. Overall System Description

## 2.1 Use Case Diagram



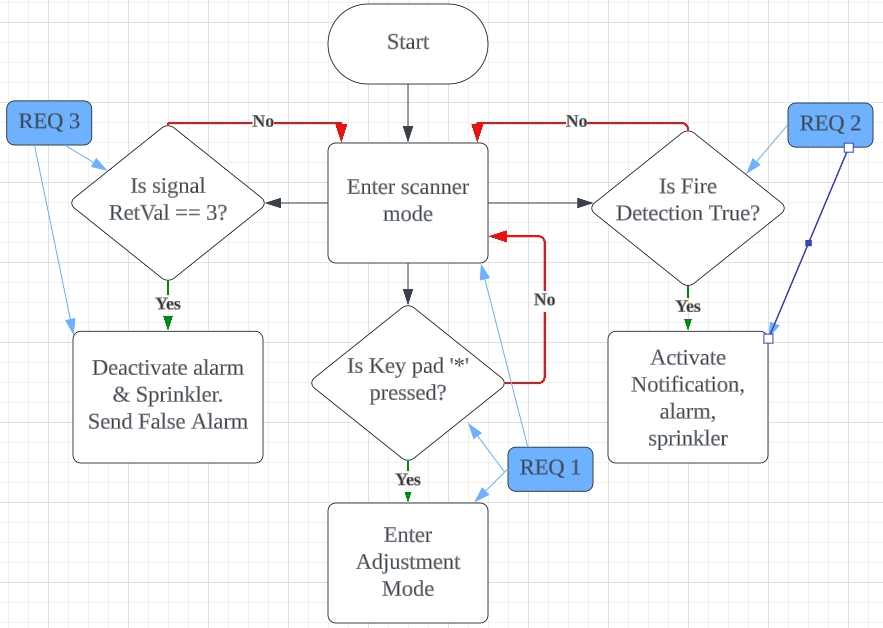
## 2.2 System Architecture

|  |
| --- |
| Raspberry Pi Development Board  **SPI\_ADC GPIO26    GPIO21 GPIO24     GPIO22 GPIO18**  **SPI&GPIO11 I2C**  **GPIO 5,6,12,13,16,19,20**  LCD  Buzzer  LED  Servo Motor  Keypad  RFID Reader  Slide Switch  Temperature & Humidity Sensor (DHT11)  Light-Dependent Resistor (LDR) |

## 2.3 Functional Requirements

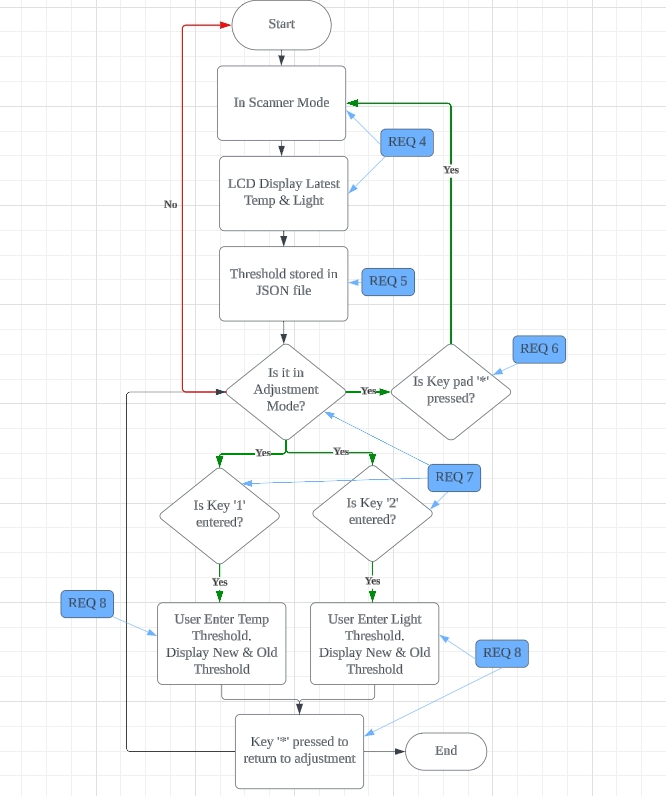
### 2.3.1 Main

|  |  |
| --- | --- |
| **REQ\_ID** | **REQUIREMENTS** |
| REQ-01 | When the system turns on, it should enter “scanner” menu. When ‘\*’ is pressed on keypad, it should switch to “adjustment” menu. |
| REQ-02 | If fireDetection becomes True, it should send notification, set alarm and sprinkler active. |
| REQ-03 | If signal RetVal = 3 is received from deactivation system, turn off alarm, sprinkler and send false alarm notification. |



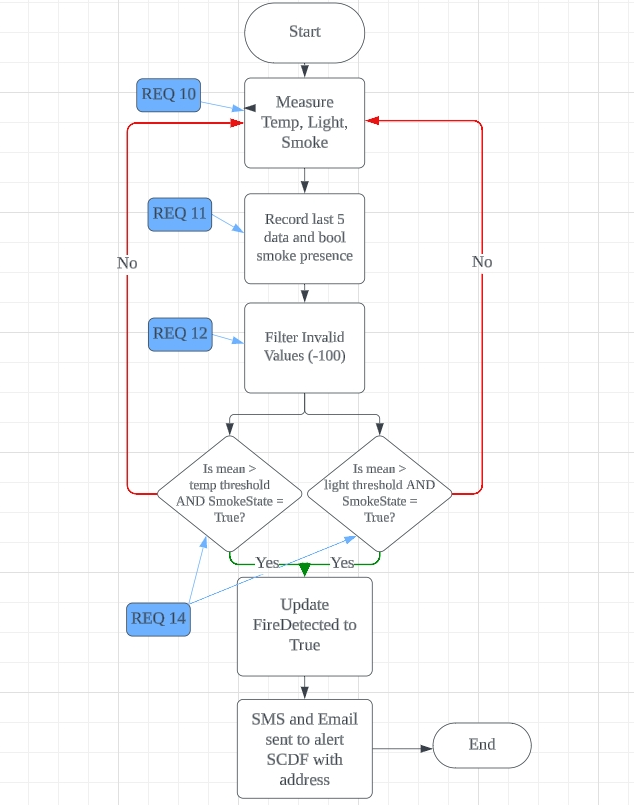
### 2.3.2 HMI and Configuration

|  |  |
| --- | --- |
| **REQ\_ID** | **REQUIREMENTS** |
| REQ-04 | When the System turns on, it will be in “scanner” menu. LCD should constantly update the LCD to display the latest temperature and light intensity values. When scanning, LCD will display:  Line 1 = “Scanning Now.”  Line 2 = “Temp:xx Light:xx” |
| REQ-05 | The threshold values should be stored and loaded from previous runs of the program using a .JSON file. |
| REQ-06 | While in the “adjustment” menu. If key \* is entered in the keypad in “adjustment” mode, return to “scanner” menu. |
| REQ-07 | After entering “adjustment” mode, there should be an option for them to select to adjust either the threshold of 1-Temperature or 2-Light. It should display the current threshold values too. LCD displays:  Line 1 = “1-TempThres: xx”  Line 2 = “2-LghtThres: xx”  While in the “adjustment” menu,  if key ‘1’ is entered in the keypad, enter the temperature adjustment menu, if key ‘2’ is entered in the keypad, enter the light adjustment menu, |
| REQ-08 | In temperature/light “adjustment” menu, xx represents the current threshold value whilst yy shows the new value being inputted by keypad.. Display on LCD (change accordingly for temp/light):  Line 1 = "Temp Thresholds"  Line 2 = “Old:xx, New:yy”  If key \* is entered in the keypad, submit the changes and update threshold value, then return to “adjustment” menu in REQ-07. |
| REQ-09 | While in the selective “adjustment” menu in REQ-08. When # is pressed, a backspace is simulated by shortening the value of yy by removing the last digit. |



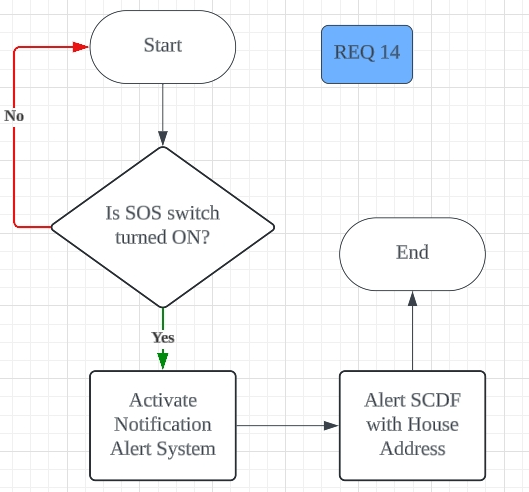
### 2.3.2 Automatic Fire Detection

|  |  |
| --- | --- |
| **REQ\_ID** | **REQUIREMENTS** |
| REQ-10 | Constantly ping sensors to collect data on the temperature, light intensity and presence of smoke in the surroundings. Store the last 5 recorded data (of temperature and light intensity) in an array. Save the bool for smoke presence too. |
| REQ-11 | Take and store the mean value of the last 5 recorded data of temperature and light intensity in REQ-10. This mean value will be the final value used to compare with threshold values to detect abnormally high temperature/light intensity. |
| REQ-12 | Take any invalid values (-100) of temperature reading in REQ-10 and filter it out. |
| REQ-13 | IF mean detected temperature AND smoke presence = True > fire detection threshold temperature OR  IF mean detected light intensity AND smoke presence = True > fire detection threshold light intensity  Update the bool fireDetected to True and return it to main.py. |



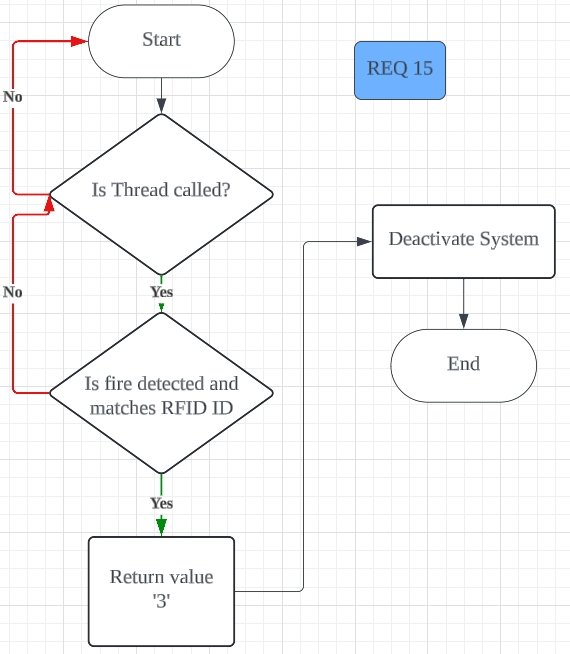
### 2.3.3 Manual SOS Switch

|  |  |
| --- | --- |
| **REQ\_ID** | **REQUIREMENTS** |
| REQ-14 | When the manual SOS switch (slide switch) is switched to ON, then send a notification to alert SCDF for help, with House Address, only ONCE. |



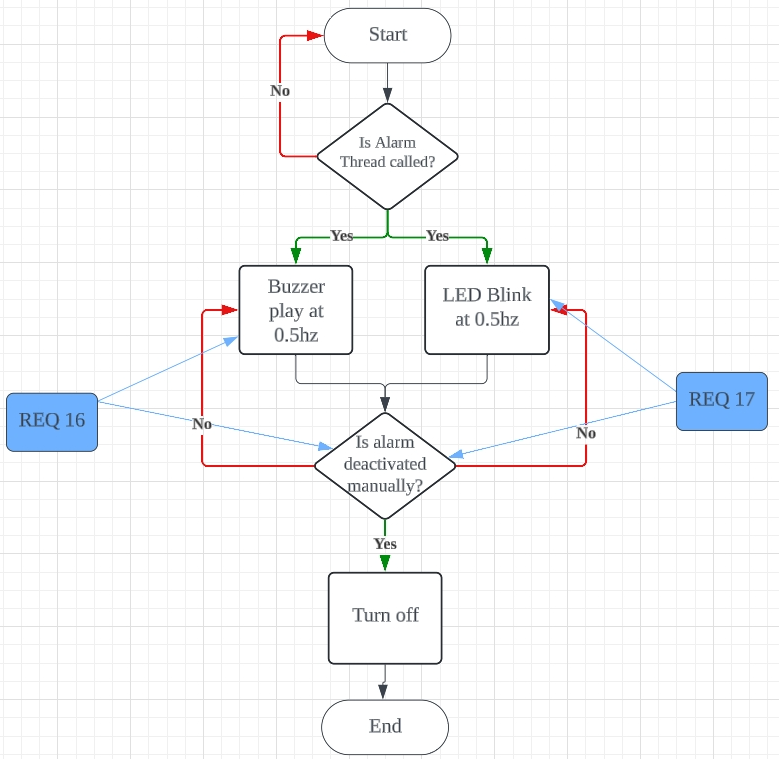
### 2.3.4 Fire Alarm Deactivation Key (False Alarm)

|  |  |
| --- | --- |
| **REQ\_ID** | **REQUIREMENTS** |
| REQ-15 | When called by main.py, it should start as a thread.  When fireDetection is True and RFID ID matches the correct ID, return a value of 3 to main.py & deactivate system. |



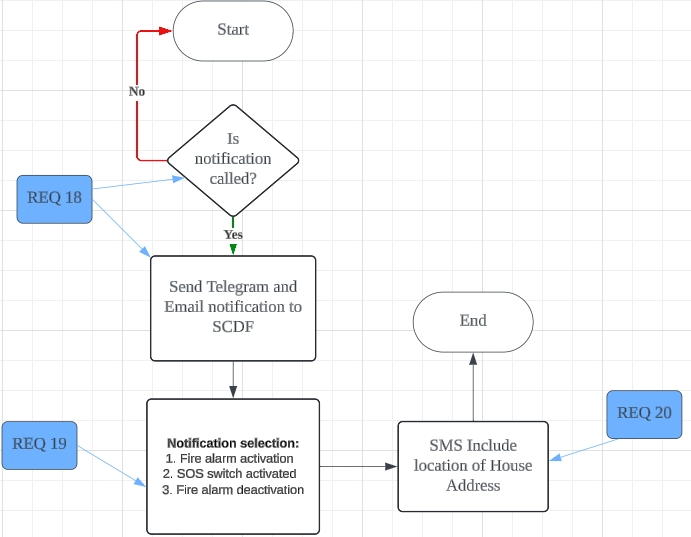
### 2.3.5 Alarm

|  |  |
| --- | --- |
| **REQ\_ID** | **REQUIREMENTS** |
| REQ-16 | Thread should start when called by main.py. It should stop when called to.  When the fire is detected (fireDetected = True), buzzer will constantly play a loud sound ON and OFF at 0.5Hz until fire alarm is deactivated manually. |
| REQ-17 | Thread should start when called by main.py. It should stop when called to.  When the fire is detected (fireDetected = True), LED must blink OFF and ON at 0.5Hz constantly until fire alarm is deactivated manually. |



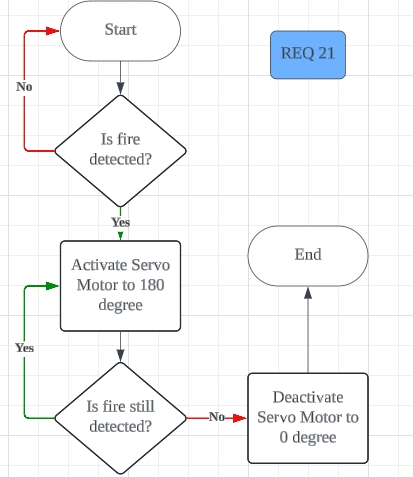
### 2.3.6 SCDF Notification System

|  |  |
| --- | --- |
| **REQ\_ID** | **REQUIREMENTS** |
| REQ-18 | When function is called, send a notification to Telegram and an email to set email. |
| REQ-19 | An input selection should be included in the function in REQ-17.  Types of selection:   * + - 1. Fire alarm activation       2. SOS switch activated and help needed       3. Fire alarm deactivation & false alarm   A dictionary of key = selection and value = message header AND message content should be created. This dictionary should accurately return the corresponding message header or message content to customise the notification text accordingly. |
| REQ-20 | The notification system need to include the location of the fire or help needed (House Address). |



### 2.3.7 Sprinkler System

|  |  |
| --- | --- |
| **REQ\_ID** | **REQUIREMENTS** |
| REQ-21 | When called on by main.py and a fire is detected (fireDetected = True), activate the servo motor and perform:   * Turn to 180° if fireDetected is True * Turn to 0° if fireDetected is False |



# 3. Software Architecture

Application Layer (AL)

main.py

notification.py

sprinkler.py

hmi.py

detection.py

sos\_switch.py

deactivation.py

alarm.py

Hardware Abstraction Layer (HAL)

input\_switch.py

hal\_temp\_humidity\_  
sensor.py

hal\_buzzer.py

hal\_adc.py

hal\_servo.py

dht11.py

hal\_keypad.py

hal\_led.py

hal\_lcd.py

hal\_rfid\_reader.py

|  |  |
| --- | --- |
| **No.** | **Peripherals Used** |
| 1 | LED |
| 2 | Buzzer |
| 3 | LCD |
| 4 | Keypad |
| 5 | RFID |
| 6 | LDR |
| 7 | Servo Motor |
| 8 | Temperature and Humidity Sensor |
| 9 | Slide Switch |

# 4. System Test Cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **REQ\_ID** | **TEST CASE DESCRIPTION** | **TEST STEPS** | **PRE-CONDITION(S)** | **EXPECTED RESULT** |
| TEST-01 | REQ-01 | When the system turns on, it should enter “scanner” menu. When ‘\*’ is pressed on keypad, it should switch to “adjustment” menu. | 1. Check what the LCD displays first.  2. Press \* on keypad.  3. Check what the LCD changes the display to. | Start the app | It should display  “Scanning Now  Temp:xx Light:xx”  and after you press \*, it should change to “Welcome to  Adjustment Mode”. |
| TEST-02 | REQ-02 | If fireDetection becomes True, it should send notification, set alarm and sprinkler active. | 1. Check Telegram and email for notification.  2. Listen for buzzer output.  3. Look for LED and servo motor output. | Trigger fireDetection as stated in TEST-13 | “Fire Alert” notification gets sent, buzzer makes sounds and LED flashes at 0.5Hz. |
| TEST-03 | REQ-03 | If signal RetVal = 3 is received from deactivation system, turn off alarm, sprinkler and send false alarm notification. | 1. Check Telegram and email for notification.  2. Listen for buzzer output.  3. Look for LED and servo motor output. | Trigger TEST-02, then put RFID tag on reader | “False Alarm” notification gets sent, buzzer no longer makes sound and LED turns off. |
| TEST-04 | REQ-04 | When the System turns on, it will be in “scanner” menu. LCD should constantly update the LCD to display the latest temperature and light intensity values. When scanning, LCD will display:  Line 1 = “Scanning Now.”  Line 2 = “Temp:xx Light:xx” | 1. Check what the LCD displays.  2. Blow into the DHT11 temperature sensor.  3. Flash phone light into the LDR. | Start the app | It should display “Scanning Now  Temp:xx Light:xx”.  Temp value should increase after blowing. Light value should increase after flashing. |
| TEST-05 | REQ-05 | The threshold values should be stored and loaded from previous runs of the program using a .JSON file. | 1. After modifying threshold value, restart the app.  2. Enter adjustment mode again.  3. Check threshold values. | Modification of threshold value in adjustment menu | It should display the same threshold value before and after the app restart. |
| TEST-06 | REQ-06 | While in the “adjustment” menu. If key \* is entered in the keypad in “adjustment” mode, return to “scanner” menu. | 1. Press \* on keypad.  2. Observe any change in LCD display. | Enter adjustment menu by pressing \* on keypad | It should display  “1-TempThres: xx  2-LghtThres: xx” then change to  “Scanning now.  Temp:xx Light:xx” |
| TEST-07 | REQ-07 | After entering “adjustment” mode, there should be an option for them to select to adjust either the threshold of 1-Temperature or 2-Light. It should display the current threshold values too. LCD displays:  Line 1 = “1-TempThres: xx”  Line 2 = “2-LghtThres: xx”  While in the “adjustment” menu,  if key ‘1’ is entered in the keypad, enter the temperature adjustment menu, if key ‘2’ is entered in the keypad, enter the light adjustment menu, | 1. Press either ‘1’ or ‘2’ on keypad  2. Observe the change in LCD display  3. Observe which threshold adjustment menu was entered, either temperature or light. | Enter adjustment menu by pressing \* on keypad | It should enter the threshold adjustment menu specific for either temp or light as reflected in TEST-08.  Pressing 1 in keypad must enter temperature thres adjustment.  Pressing 2 in keypad must enter light thres adjustment. |
| TEST-08 | REQ-08 | In temperature/light “adjustment” menu, xx represents the current threshold value whilst yy shows the new value being inputted by keypad.. Display on LCD (change accordingly for temp/light):  Line 1 = "Temp Thresholds"  Line 2 = “Old:xx, New:yy”  If key \* is entered in the keypad, submit the changes and update threshold value, then return to “adjustment” menu in REQ-07. | 1. Type some numbers on the keypad.  2. Observe any changes to the LCD, particularly to the “New:xx” part.  3. Press \* on keypad and observe the changes. | Having entered adjustment mode, press ‘1’ or ‘2’ on the keypad. | Numbers typed on keypad should appear in the “New:xx” part of the LCD accurately.  Pressing \* should return you back to the adjustment menu showing “1-TempThres: xx  2-LghtThres: xx”  where the xx is now reflected to show the NEW threshold value. |
| TEST-09 | REQ-09 | While in the selective “adjustment” menu in REQ-08. When # is pressed, a backspace is simulated by shortening the value of yy by removing the last digit. | 1. Type some numbers on the keypad.  2. Then, press # on the keypad.  3. Observe any changes to the LCD display | Having entered adjustment mode, press ‘1’ or ‘2’ on the keypad. | The number typed should be backspaced, removing the last digit of number typed. |
| TEST-10 | REQ-10 | Constantly ping sensors to collect data on the temperature, light intensity and presence of smoke in the surroundings. Store the last 5 recorded data (of temperature and light intensity) in an array. Save the bool for smoke presence too. | 1. Print both the list of temperature and of light in the code.  2. Print the smoke presence bool value too.  3. Put your hand over IR a few times  4. Check the terminal.  5. Check the displayed values on the LCD. | Start the app. (So it will be in scanner mode) | The display light and temperature values on the LCD should be added to the list of 5, constantly updating.  SmokeDetected bool should reflect the detection state of IR sensor |
| TEST-11 | REQ-11 | Take and store the mean value of the last 5 recorded data of temperature and light intensity in REQ-10. This mean value will be the final value used to compare with threshold values to detect abnormally high temperature/light intensity. | 1. Print both the list of temperature and of light in the code.  2. Print both the avg/mean value for temperature and for light.  3. Check the terminal | Start the app. (So it will be in scanner mode) | The avg/mean value should display the mean value of the list of 5 recorded values of light intensity and temperature on the terminal. |
| TEST-12 | REQ-12 | Take any invalid values (-100) of temperature reading in REQ-10 and filter it out. | 1. Print the list of temperature readings in the code.  2. Check the terminal | Start the app. (So it will be in scanner mode) | There should not be any value of -100 in the list of temperature readings. |
| TEST-13 | REQ-13 | IF mean detected temperature > fire detection threshold temperature OR mean detected light intensity > fire detection threshold light intensity  OR smoke presence = True,  Update the bool fireDetected to True and return it to main.py.  fireDetected = fire OR light AND smoke | A1. Put hand over the smoke detector and blow into the DHT11 temperature sensor until it hits threshold value.  A2. Observe for any alarms  B1. Put hand over smoke detector and flash light on LDR.  B2. Observe for any alarms  C1. Do not put hand over smoke detector and flash light on LDR as well as blow into the DHT11 temperature sensor.  C2. Observe for any alarms | Start the app. (So it will be in scanner mode). Adjust threshold values if needed. | Test A should have the alarm sounded.  Test B should have the alarm sounded.  Test C should NOT have the alarm sounded. |
| TEST-14 | REQ-14 | When the manual SOS switch (slide switch) is switched to ON, then send a notification to alert SCDF for help, with House Address, only ONCE. | 1. Turn the slide switch to ON state.  2. Look out for any Telegram notifications or email | Start the app | There should be a notification received. |
| TEST-15 | REQ-15 | When called by main.py, it should start as a thread.  When fireDetection is True and RFID ID matches the correct ID, return a value of 3 to main.py. | 1. Tap RFID ID on the reader  2. Observe if the alarm stays ON or turns OFF. | Trigger fire alarm like in TEST-13 | Alarm should turn off after RFID is tapped. |
| TEST-16 | REQ-16 | Thread should start when called by main.py. It should stop when called to.  When the fire is detected (fireDetected = True), buzzer will constantly play a loud sound ON and OFF at 0.5Hz until fire alarm is deactivated manually. | 1. Check Telegram and email for notification.  2. Listen for buzzer output.  3. Look for LED and servo motor output. | Trigger fireDetection as stated in TEST-13 | “Fire Alert” notification gets sent, buzzer makes sounds and LED flashes at 0.5Hz. |
| TEST-17 | REQ-17 | Thread should start when called by main.py. It should stop when called to.  When the fire is detected (fireDetected = True), LED must blink OFF and ON at 0.5Hz constantly until fire alarm is deactivated manually. | 1. Check Telegram and email for notification.  2. Listen for buzzer output.  3. Look for LED and servo motor output. | Trigger TEST-02, then put RFID tag on reader | “False Alarm” notification gets sent, buzzer no longer makes sound and LED turns off. |
| TEST-18 | REQ-18 | When function is called, send a notification to Telegram and an email to set email.  An input selection should be included in the function in REQ-17.  Types of selection:  1. Fire alarm activation  2. SOS switch activated and help needed  3. Fire alarm deactivation & false alarm | 1. Select ‘1’ in terminal for fire alarm and enter.  2. Observe for any Telegram messages and emails.  3. Repeat from step 1 for ‘2’ for help needed and ‘3’ for false alarm. | Run notification.py (this will enter the debug() function made specifically for testing) | A notification should be sent via Telegram AND email should be received. |
| TEST-19 | REQ-19 | A dictionary of key = selection and value = message header AND message content should be created. This dictionary should accurately return the corresponding message header or message content to customise the notification text accordingly. | 1. Test sending notifications for all 3 cases.  2. Observe the content and header of the notification and email. | Send notification.py like in TEST-18. | The message content should reflect the value selected:  1. Fire Alarm  2. Help needed  3. False Alarm |
| TEST-20 | REQ-20 | The notification system needs to include the location of the fire or help needed (House Address). Allow for changing of location via a .JSON file as well. | 1. Observe the contents of the notifications. | Send notification.py like in TEST-18. | The location in the notification should reflect what was written in the .JSON file. |
| TEST-21 | REQ-21 | When called on by main.py and a fire is detected (fireDetected = True), activate the servo motor and perform:   * Turn to 180° if fireDetected is True * Turn to 0° if fireDetected is False | 1. Trigger the alarm like in REQ-13.  2. Watch the servo motor for any change in position.  3. Turn off the alarm with RFID like in TEST-15.  4. Watch the servo motor for any change in position. | Start the app (so it’s in scanner mode) | When alarm activated, it should turn servo from 0° to 180°.  When alarm deactivated, it should turn servo from 180° to 0°. |