Table of Contents

[Document Version 1](#_bookmark0)

1. [Purpose 2](#_bookmark1)
   1. [Intended Audience 2](#_bookmark2)
   2. [Intended Use 2](#_bookmark3)
   3. [Scope 2](#_bookmark4)
   4. [Definitions and Acronyms 2](#_bookmark5)
2. [Overall System Description 3](#_bookmark6)
   1. [Use Case Diagrams 3](#_bookmark7)
   2. [System Architecture 3](#_bookmark8)
   3. [Functional Requirements 3](#_bookmark9)
      1. [Status Report 3](#_bookmark10)
      2. [Detect fire and smoke 4](#_bookmark11)
      3. [Alert and putting out fire 5](#_bookmark12)
      4. [Manual switch and Communicating with SCDF (Remote Access) 6](#_bookmark13)
   4. [Non-Functional Requirements 7](#_bookmark14)
      1. [High alert protocol 7](#_bookmark15)
3. [Software Architecture 8](#_bookmark16)
   1. [Static Software Architecture 8](#_bookmark17)

# Document Version

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Update | Name | Date | Version |
| 1. | Initial version | SRS\_project\_SmartFireAlertSystem | 5/6/2025 | 1.0 |

# Purpose

## Intended Audience

This SRS document describes the System Requirements and Software Design for a smart fire alert system and the target audience are the SDCF working on the development of an IoT driven system which continuously monitors the homes of elderly residents for early signs of fire .

## 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

## Definitions and Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Description** |
| LED | Light Emitting Diode |
| BUZZ | Buzzer |
| SW | Software |
| HW | Hardware |
| THS | Temperature and Humidity Sensor |
| VS | Visual Sensor |
| LS | Light Sensor |
| LCD | LiquidCrystal display |

# Overall System Description

## Use Case Diagrams

A screenshot of a device

AI-generated content may be incorrect.

## System Architecture

Rasberry Pi Development board



**SPI\_ADC\_CH01**

**GPIO24**

**GPIO17**

**GPIO18**

**I2C**

**SPI\_ADC\_CH02**

**GPIO22**

**GPI026**

ServoMotor

Switch

LCD

LDR

Buzzer

Infared sensor

LED

Temperature

sensor

## Functional Requirements

### Status Report

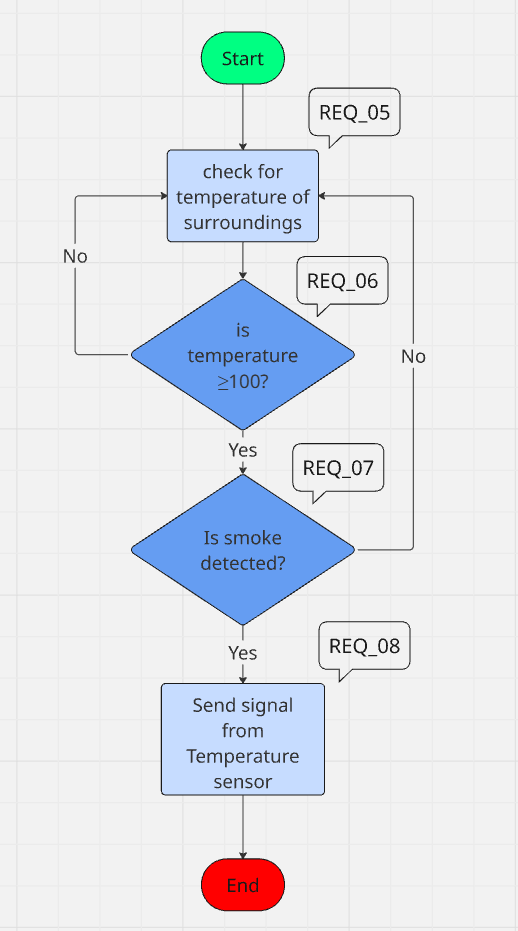
-Allows both SCDF and elderly to see instructions on LCD and hear buzzer

|  |  |
| --- | --- |
| **REQ\_ID** | **Requirement** |
| REQ-01 | When the Smart Fire Alert System is first powered ON, the text below shall be displayed on the LCD screen  print (“System Activated”) |
| REQ-02 | When Smart Fire Alert System is ON, buzzer, temperature sensor and are activated  Print (“buzzer activated”) |

### Detect fire and smoke

* Check if there is a fire or smoke being produced within the area. If it is, the temperature sensor sends a signal

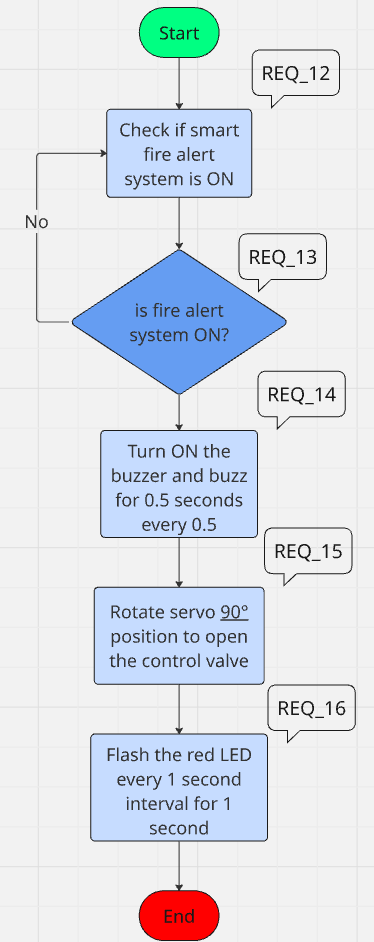
|  |  |
| --- | --- |
| **REQ\_ID** | **Requirement** |
| REQ\_03 | Within the area, the temperature sensor checks for the temperature. If the temperature exceeds 100, check for smoke using the smoke detector. |
| REQ\_04 | From REQ\_03, check for smoke within the area. If smoke is detected, turn ON the  smart fire alert system. |



### Alert and putting out fire

-After fire alert is activated when the temperature exceeding 100 the sever motor rotates 90 degrees as a switch which allows water to reach the sprinklers to extinguish the fire. Furthermore, the buzzer and LED activate every 0.5 second and 1 second.

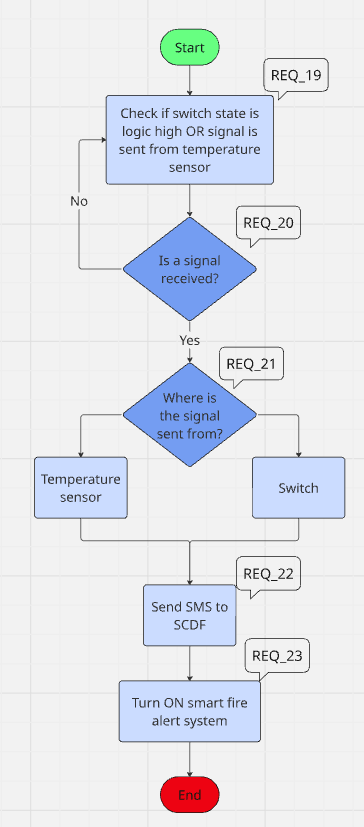
|  |  |
| --- | --- |
| **REQ\_ID** | **Requirement** |
| REQ\_09 | When fire alert system is activated, server motor rotates 90 degrees and opens the  control valve |
| REQ\_10 | When fire alert system is activated, activate the buzzer every 0.5 second interval |
| REQ\_11 | When fire alert system is activated, blink the red LED every 1 second interval |



### Manual switch and Communicating with SCDF (Remote Access)

* There are 2 parts here: the Manual Switch & Communication with SCDF. The manual switch is used to manually activate the smart fire alert system instead of waiting for the temperature sensor to send a signal OR if the system does not activate when a fire breaks out. Once the temperature sensor sends a signal or the switch is flipped to logic high, a SMS alert is sent to SCDF.

|  |  |
| --- | --- |
| **REQ\_ID** | **Requirement** |
| REQ\_17 | The Raspberry Pi checks if a signal is sent from the temperature sensor. If so, a  message is sent to the SCDF, and the smart fire alert system turns ON |
| REQ\_18 | If there is a fire and the automatic system is not working or the user wants to  manually activate the fire alert system, the user can flip a switch to logic high which sends a message to the SCDF, and the smart fire alert system turns ON |

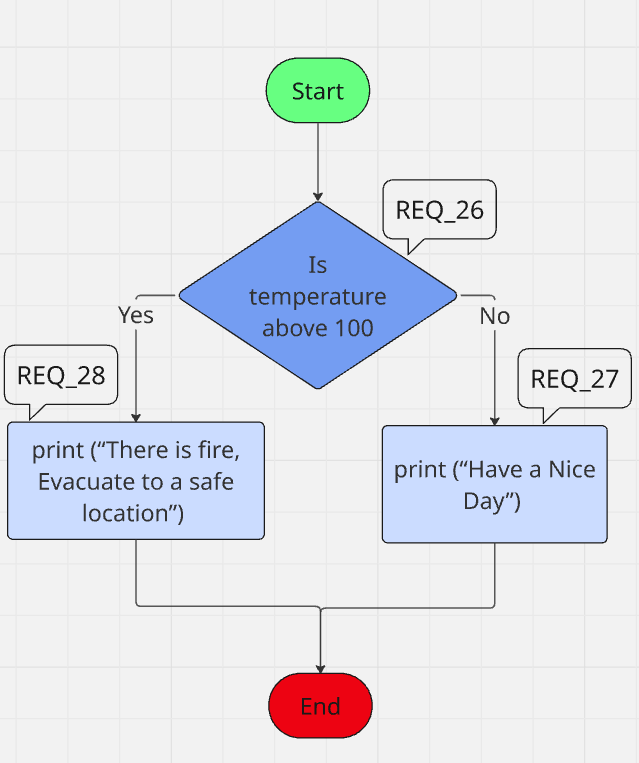


## Non-Functional Requirements

### High alert protocol

-For the rest of the time when there is no fire (when temperature is less than 100) the system is on high alert where it will have activated the temperature sensor as in requirement 2.3.1 and the LCD will have printed “Have a nice day.”, but in a case of a fire LCD will print “There is a fire, evacuate to a safe location”.

|  |  |
| --- | --- |
| REQ\_24 | If temperature <100, the text below shall be displayed on the LCD screen  print (“Have a Nice Day”) |
| REQ\_25 | If temperature >= 100, the text below shall be displayed on the LCD screen  print (“There is fire, Evacuate to a safe location”) |



# 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 Application Layer(HAL)

HAL\_LCD.py

HAL\_TempSensor.py

HAL\_IRSensor.py

HAL\_Switch.py

HAL\_Buzzer.py

HAL\_Servo.py

#### Application Layer

Main.py

TempSmoke.py

AlertSystem.py

RemoteAcess.py