OOM Mini-Project Report

Indian Institute of Information Technology
Allahabad
Semester III, B.Tech(IT)

Fire Alarm System

Made By:-

Riya Goyal	IIT2019096
Maitry Jadiya	IIT2019100
Sonal	IIT2019122
Dev Bansal	IIT2019132

TABLE OF CONTENTS

INTRODUCTION

We have developed a GUI enabled file alarm system for CC3 building. The system involves inputs of various sensor devices like heat sensor, smoke detector and carbon monoxide detector.

The CC3 building is equipped with all these sensors at various places. The system receives the input from these sensors and draws the verdict to generate fire alarm or not. Furthermore, the system also localizes the area of the sensor that initiated the alarm. A log report is also generated where users could see the history of the sensed values by the sensors.

The UML Diagrams are also included in this manual with a brief introduction of what they are.

SYSTEM REQUIREMENTS

- ❖ Operating System: Any
- ❖ jdk latest version
- ❖ Java Netbeans
- ❖ Functioning sensors
    - ➢ Smoke Sensors
    - ➢ Heat Sensors
    - ➢ CO Sensors
- ❖ Proper internet connection
- ❖ Gmail account(For both Developer and User)

DEVELOPMENT TOOLS

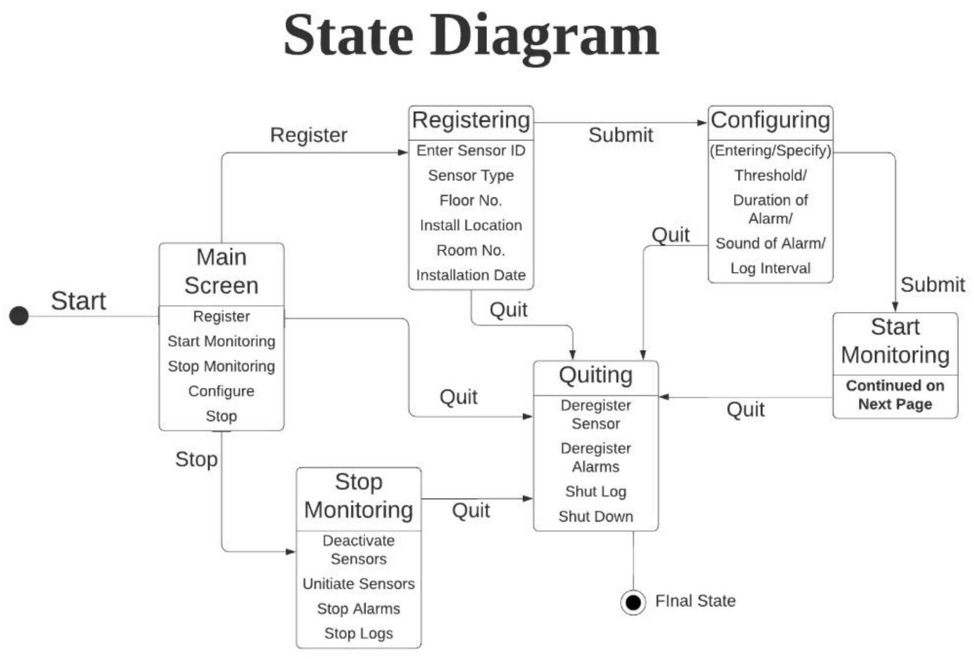
- ❖ Java GUI Frameworks such as Swing & AWT API.
- ❖ Abstract Window Toolkit
- ❖ Java Applet
- ❖ Mail API.
- ❖ Java Collections Framework
- ❖ Extra Libraries:(refer zip file).

FEATURES

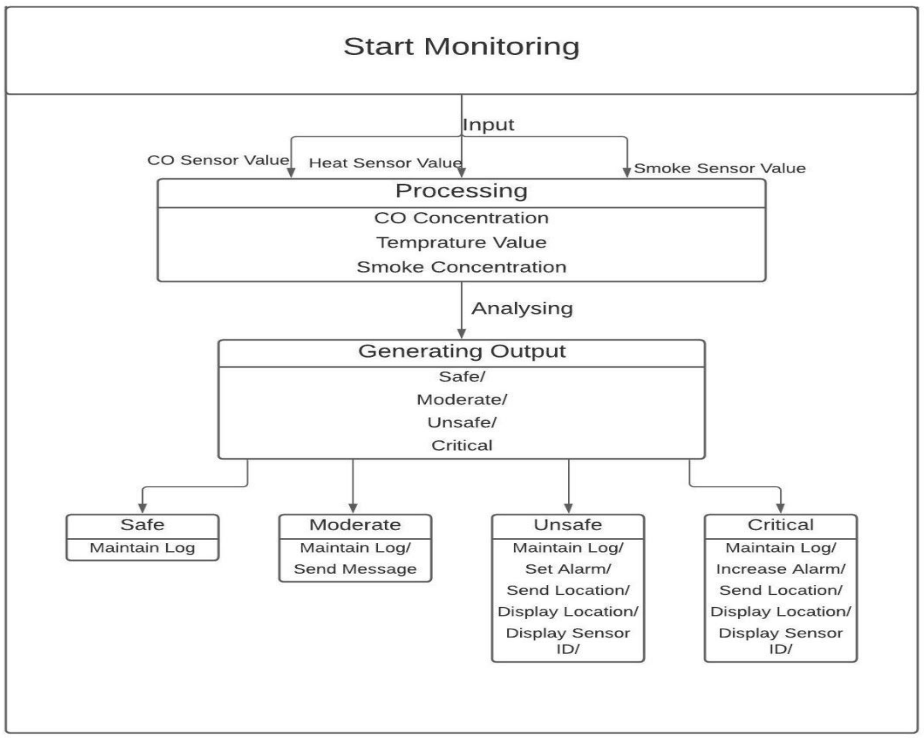
1. Mailing Functionality(Implemented through jar files added in Libraries Section of The Project)
2. Printing Functionality(Implemented through jar files added in Libraries of the Project)
3. jcalendar-1.4.jar file for Calendar Dates functionality implemented in Visitor Panel for Check-In Date and Check-Out Date(text fields) since Java GUI Frameworks such as Swing & AWT API does not support Datepicker implicitly.
4. The mail would be sent to the recipient to check the building at the specified location.

STATE DIAGRAM

State-diagrams show behaviour of classes in response to external stimuli (clicks/ keyboard inputs/ user inputs etc..)
It consists of starting points, intermediate states of the system, transitions which results in state change, and ending points.
The state Diagram for Fire Alarm system is as follows.

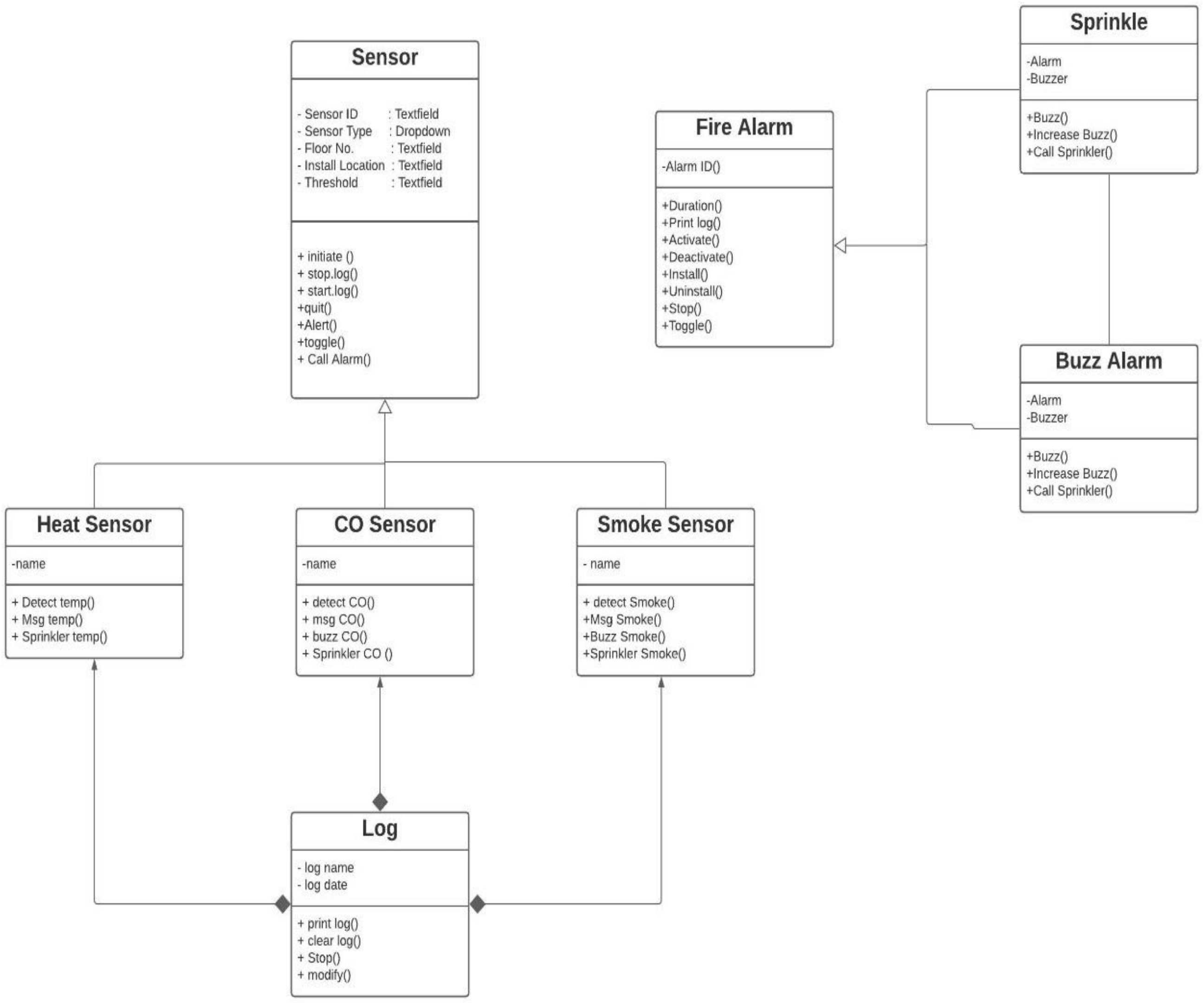


The state diagram embedded in the Start Monitoring state is continued below:



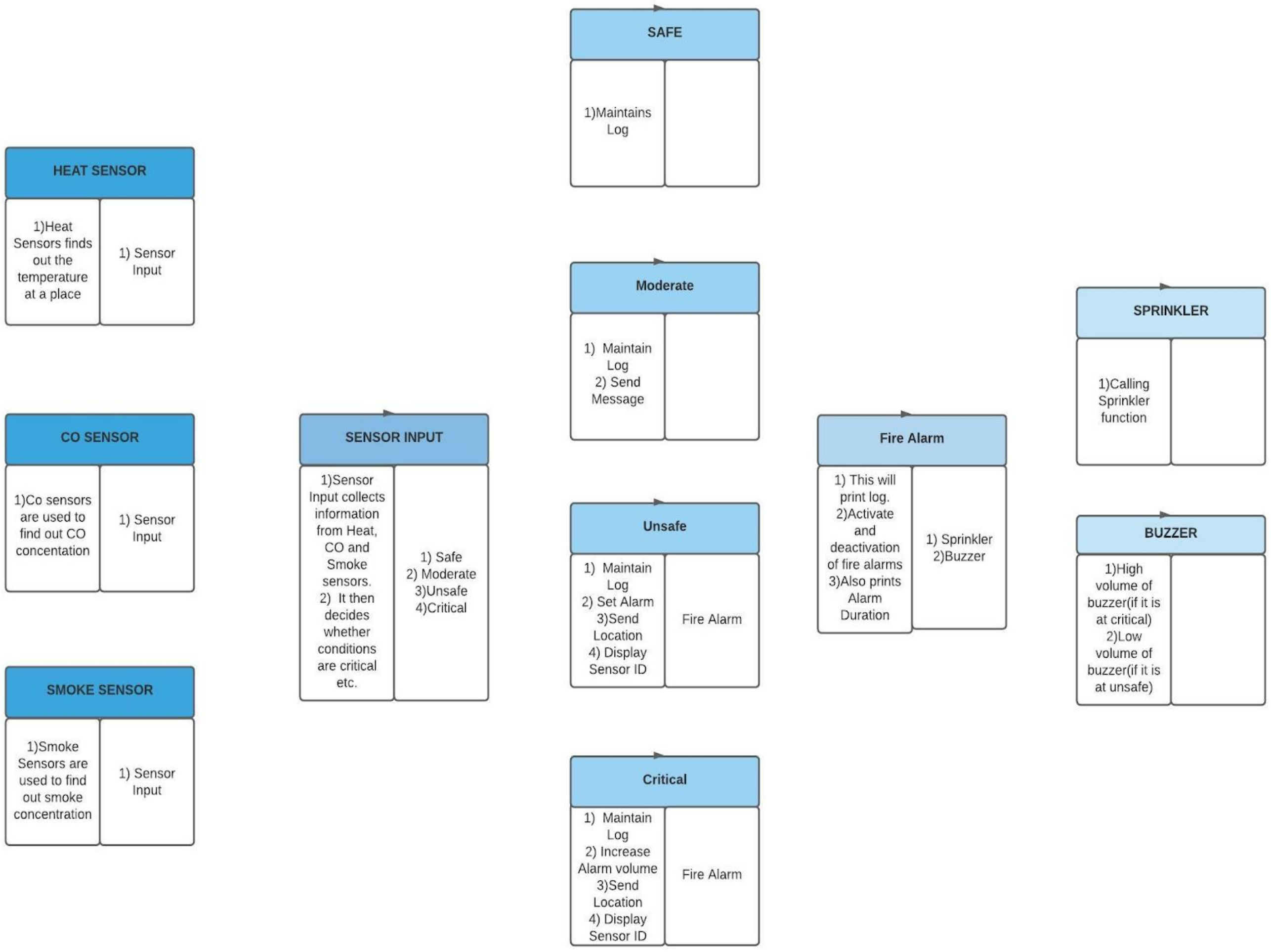
CLASS DIAGRAM

Class-diagrams show relationships of different classes with each other. Each class shows its attributes and methods and they are connected to other classes as per their dependency / relationship with each other (e.g. composition, association, aggregation, inheritance etc.)
It is attained with the help of standard symbols which is shown below.



<u>CRC DIAGRAM:</u>

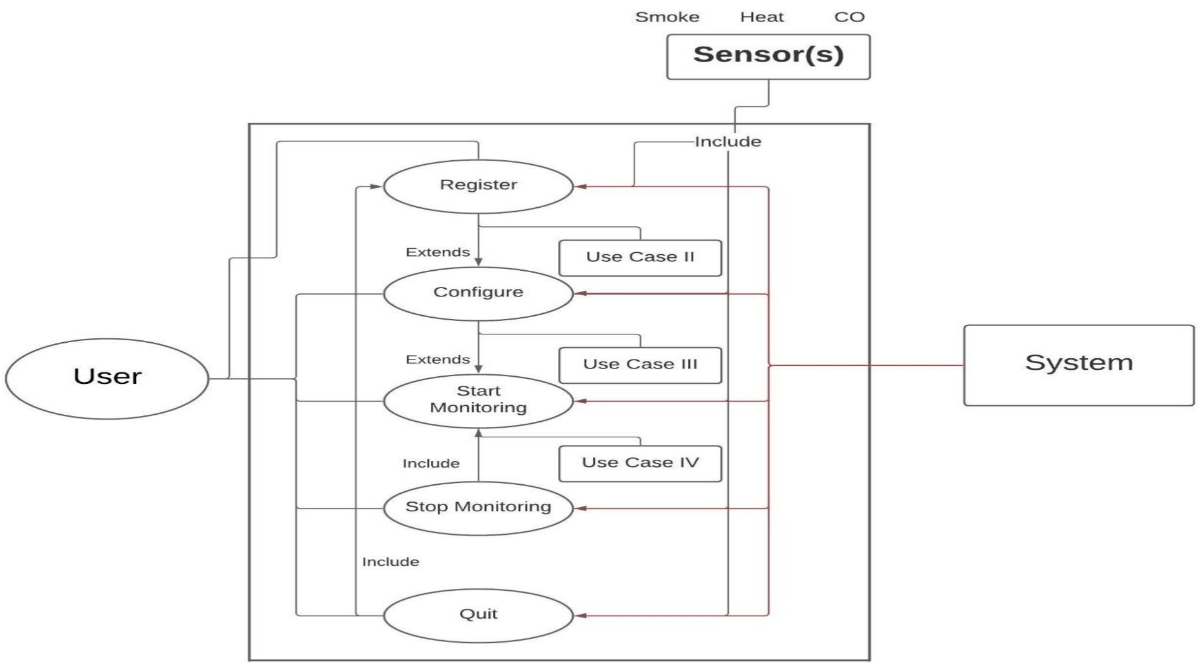
A class Responsibility Collaborator diagram is a collection of standard index cards that has been divided into three sections namely:
1)Class Name
2)Responsibilities
3)Collaborators
A class represents a collection of similar objects, a responsibility is something that a class knows or does ,and a collaborator is another class that the class interacts with to fulfil its responsibilities.


SAFE
1)Maintains Log
HEAT SENSOR
1)Heat Sensors finds out the temperature at a place
1) Sensor Input
Moderate
1) Maintain Log 2) Send Message
SPRINKLER
1)Calling Sprinkler function
CO SENSOR
1)Co sensors are used to find out CO concentation
1) Sensor Input
SENSOR INPUT
1)Sensor Input collects information from Heat, CO and Smoke sensors. 2) It then decides whether conditions are critical etc.
1) Safe 2) Moderate 3)Unsafe 4)Critical
Fire Alarm
1) This will print log. 2)Activate and deactivation of fire alarms 3)Also prints Alarm Duration
1) Sprinkler 2)Buzzer
Unsafe
1) Maintain Log 2) Set Alarm 3)Send Location 4) Display Sensor ID
Fire Alarm
BUZZER
1)High volume of buzzer(if it is at critical) 2)Low volume of buzzer(if it is at unsafe)
SMOKE SENSOR
1)Smoke Sensors are used to find out smoke concentration
1) Sensor Input
Critical
1) Maintain Log 2) Increase Alarm volume 3)Send Location 4) Display Sensor ID
Fire Alarm

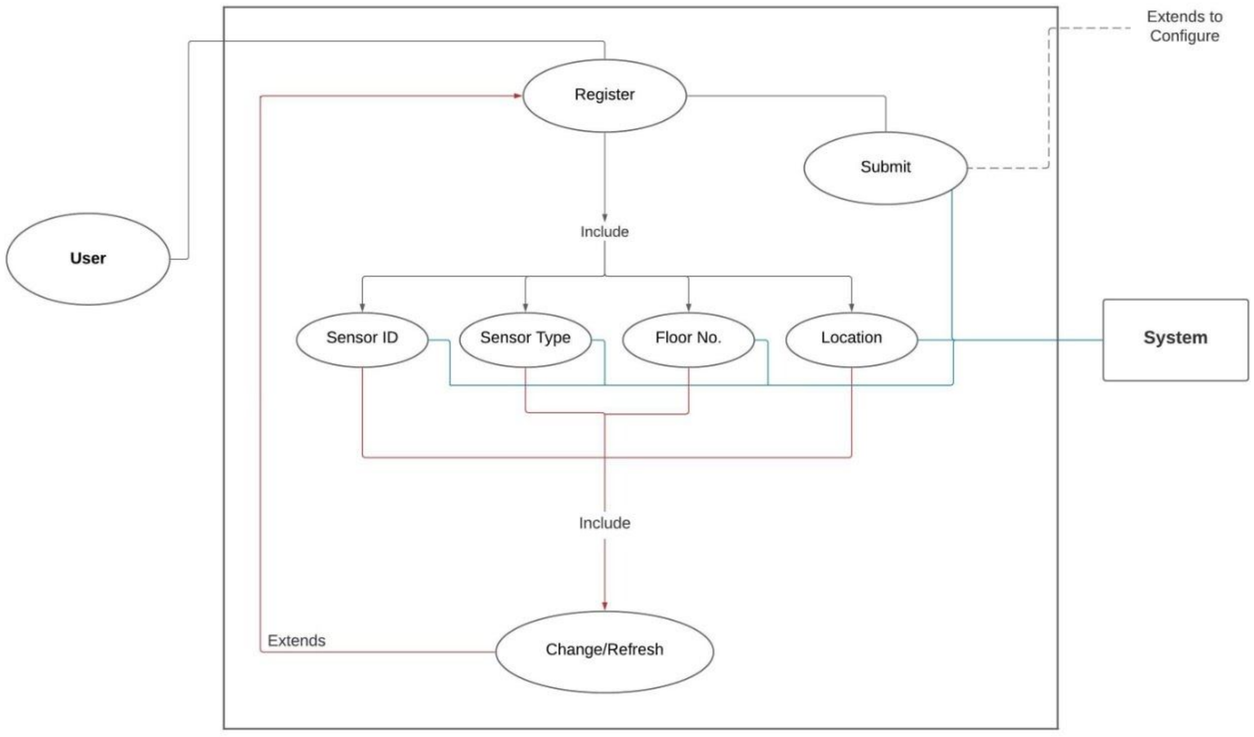
<u>USE CASE DIAGRAM</u>

Use case diagrams consist of actors, use cases and their relationships within specified system boundaries.
These are basically used to model the system/subsystem of an application or a program.
It basically shows the dynamic behaviour of our system. It shows who and how is he/she using the system and what do they want to achieve or what their target is .
It is a useful technique for identifying, clarifying and organizing system requirements.
It’s made up of a set of all possible sequences of interactions between system and users that define the feature to be implemented and resolution of any errors that may be encountered.

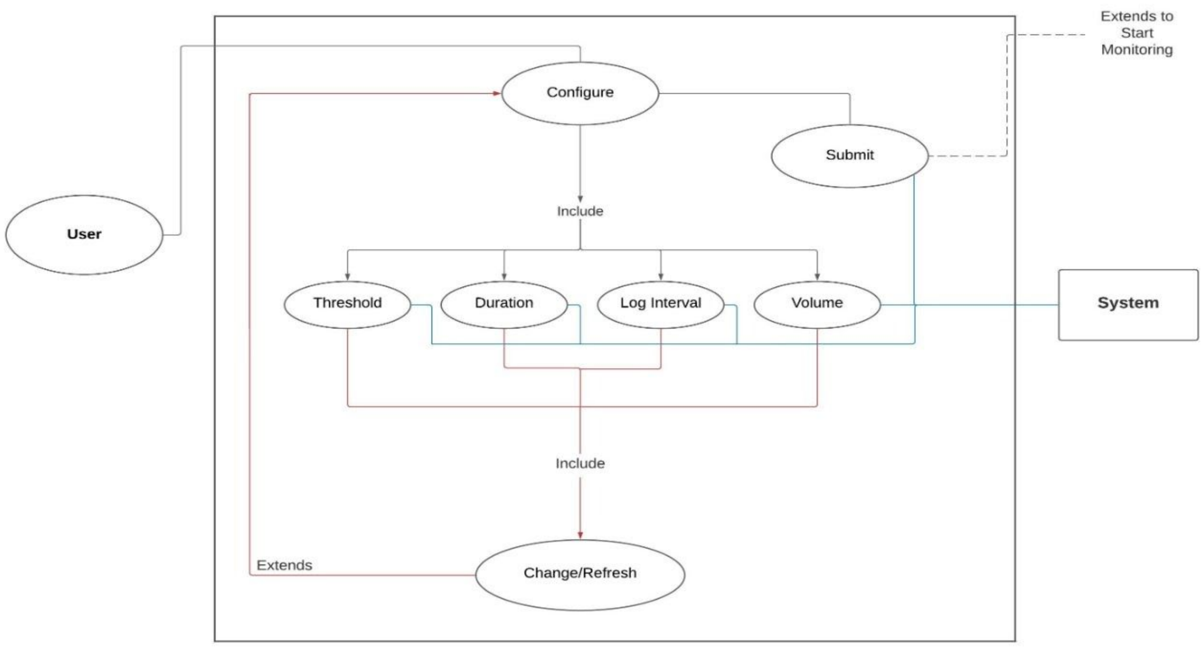
Use Case Diagram 1



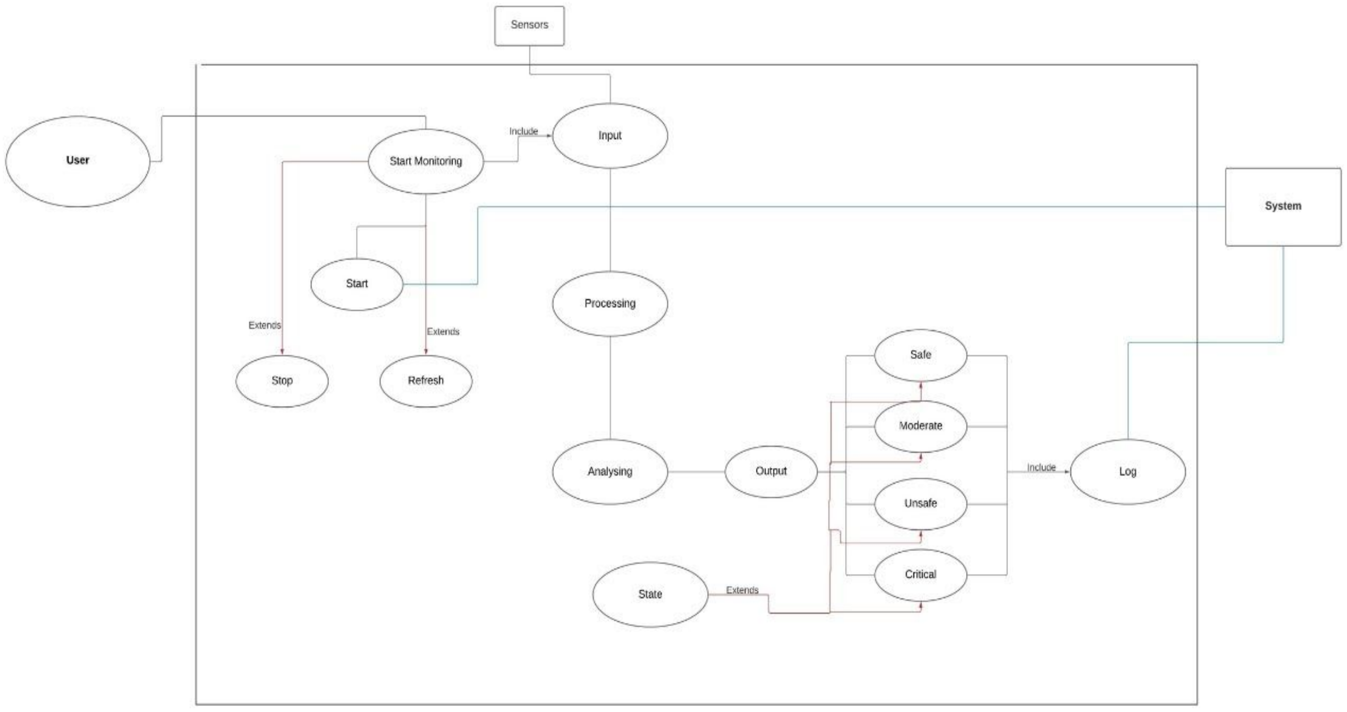
Use Case Diagram II



Use Case Diagram III



Use Case Diagram IV

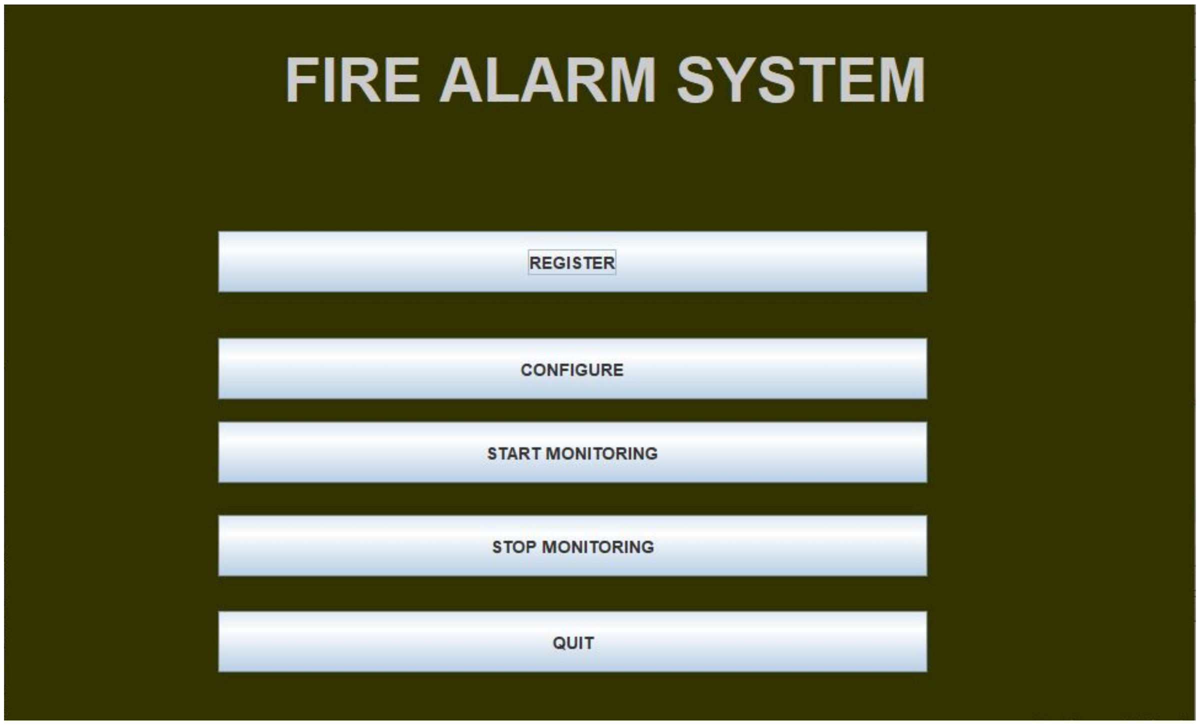


USER INTERFACE

Main Page:

So, in the main page you have 5 buttons which will be functional when the user will click on it.

1. **Register:** User will be directed to the Register a sensor page whose functioning is described below.
2. **Configure:** User will be directed to the Configure a sensor page whose functioning is described below.
3. **Start Monitoring:** Users will be directed to start monitoring a sensor page where the user will get dynamic values of sensors which will be updated automatically after chosen time period by the user.
4. **Stop Monitoring:** Here all the sensors which were registered and configured will stop functioning and the log table will be generated.
5. **Quit:** From this button the user can stop the program.



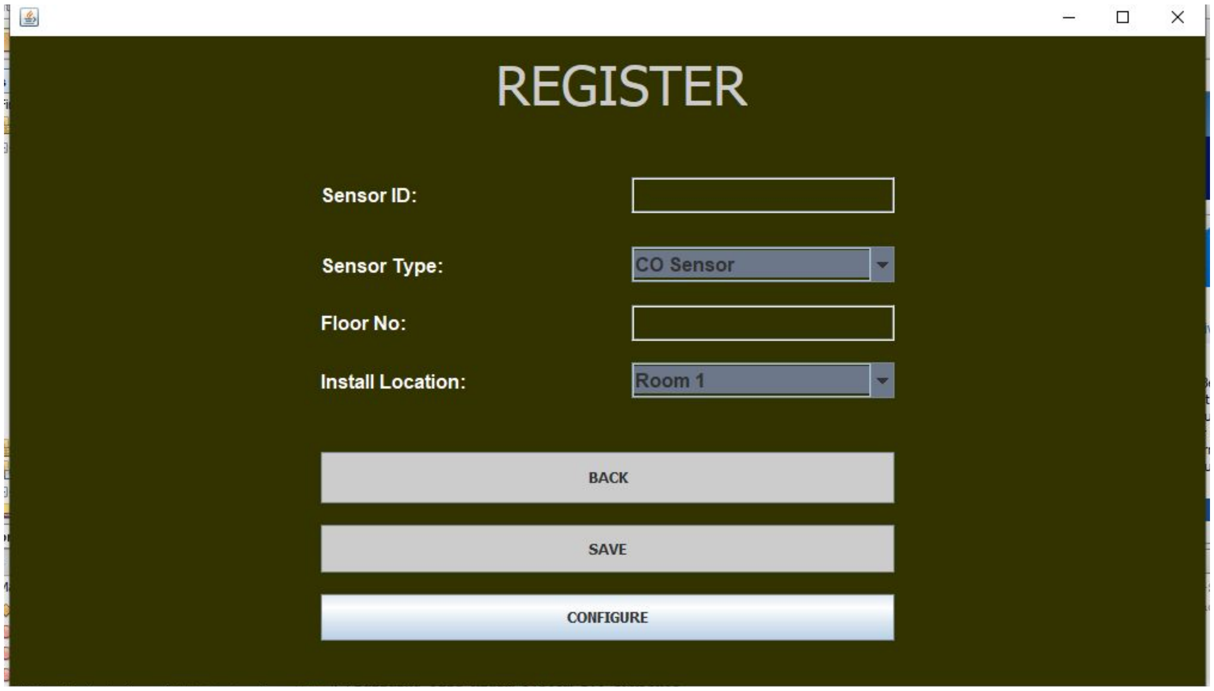
Register Page:

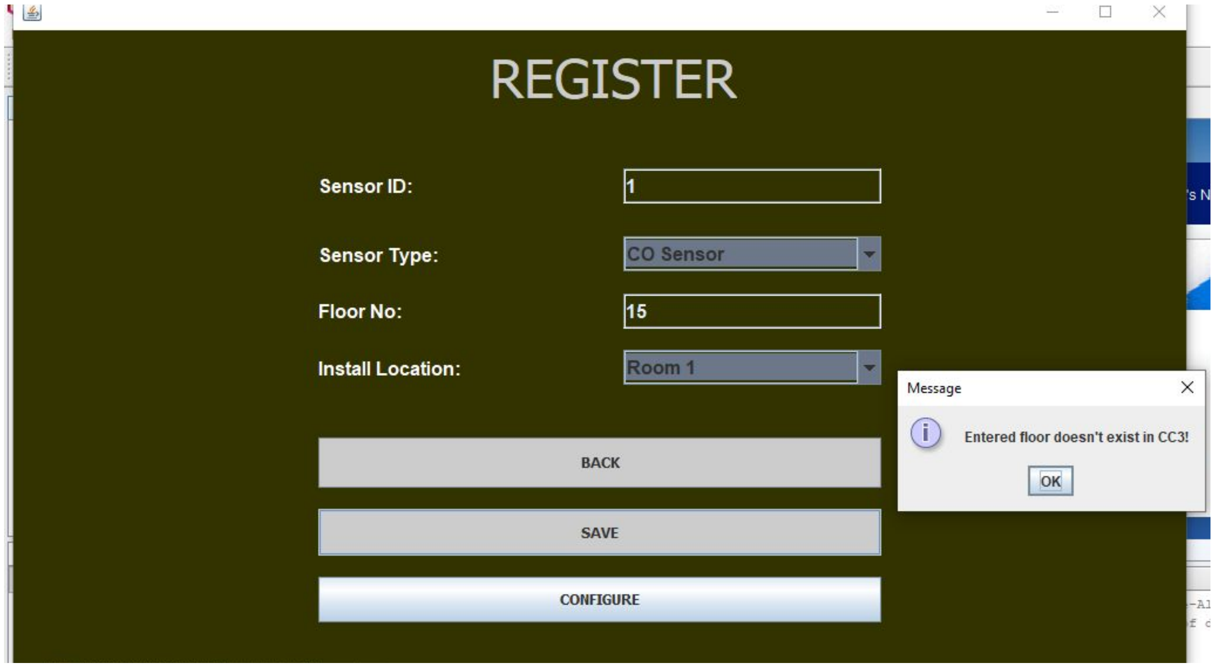
In this page you will enter all the details of the sensor which includes:

- ❖ Sensor ID: <Textfield> (ID type validation)
- ❖ Sensor Type: <Dropdown>
- ❖ Installation Location :
  - Floor No: <Textfield> (Number type validation)
  - Room No: <Textfield>

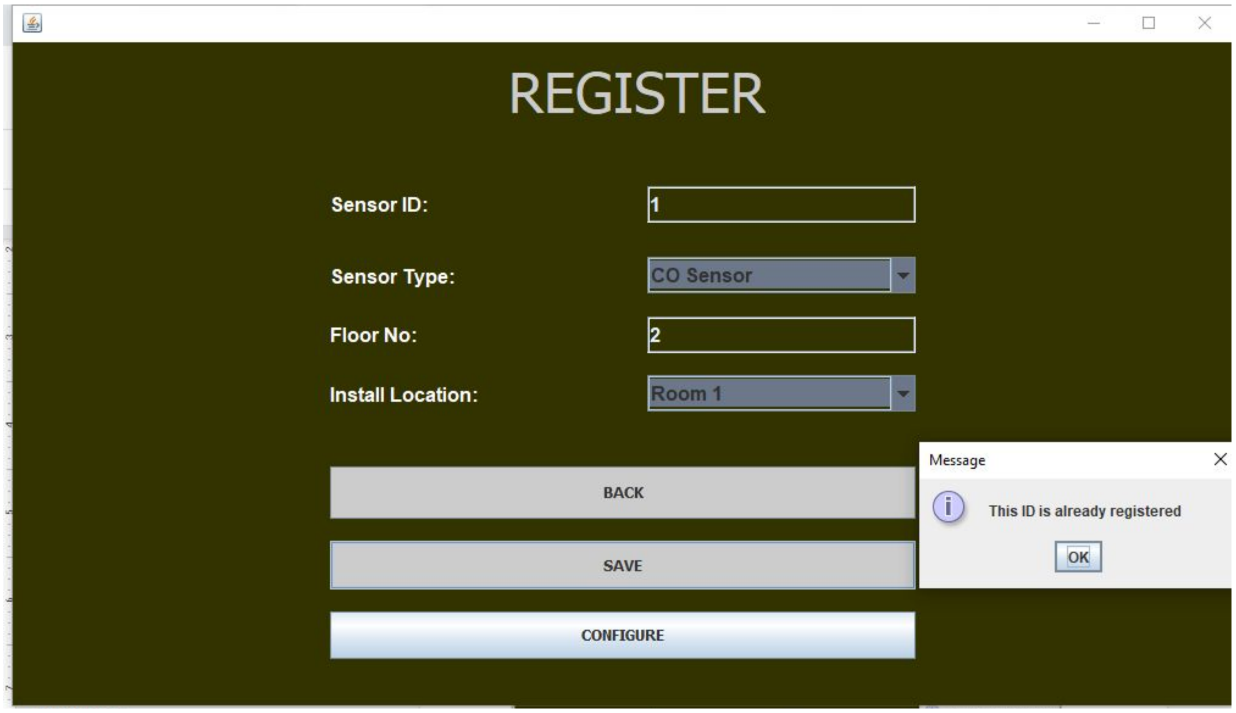
Also, apart from the details user will find three buttons which are

- ❖ Back: user will be directed to the main page
- ❖ Save: the sensor is registered successfully which is confirmed by a popup screen.
- ❖ Configure: User will be directed to Configure page
- ❖ If there's any irrelevant data entered then, these pop ups let the user know about the correction.





Entered floor doesn't exist in CC3!


REGISTER

Sensor ID: 1

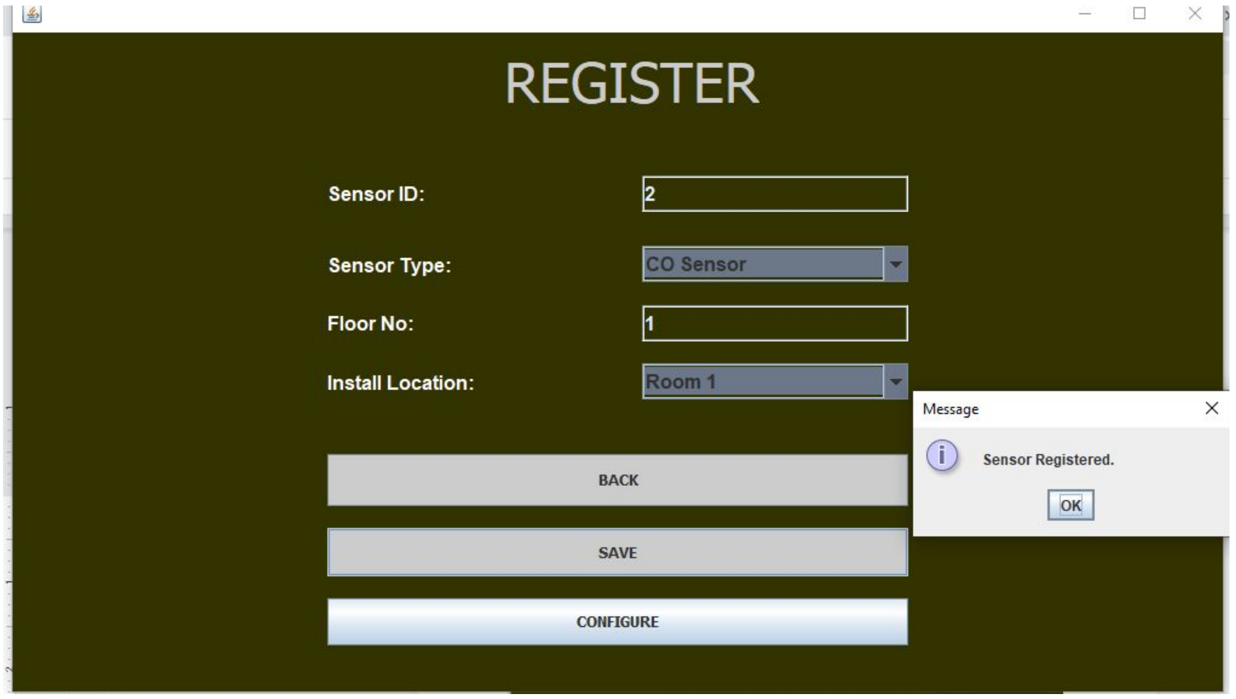
Sensor Type: CO Sensor

Floor No: 2

Install Location: Room 1

Message

This ID is already registered


REGISTER

Sensor ID: 2

Sensor Type: CO Sensor

Floor No: 1

Install Location: Room 1

Message

Sensor Registered.

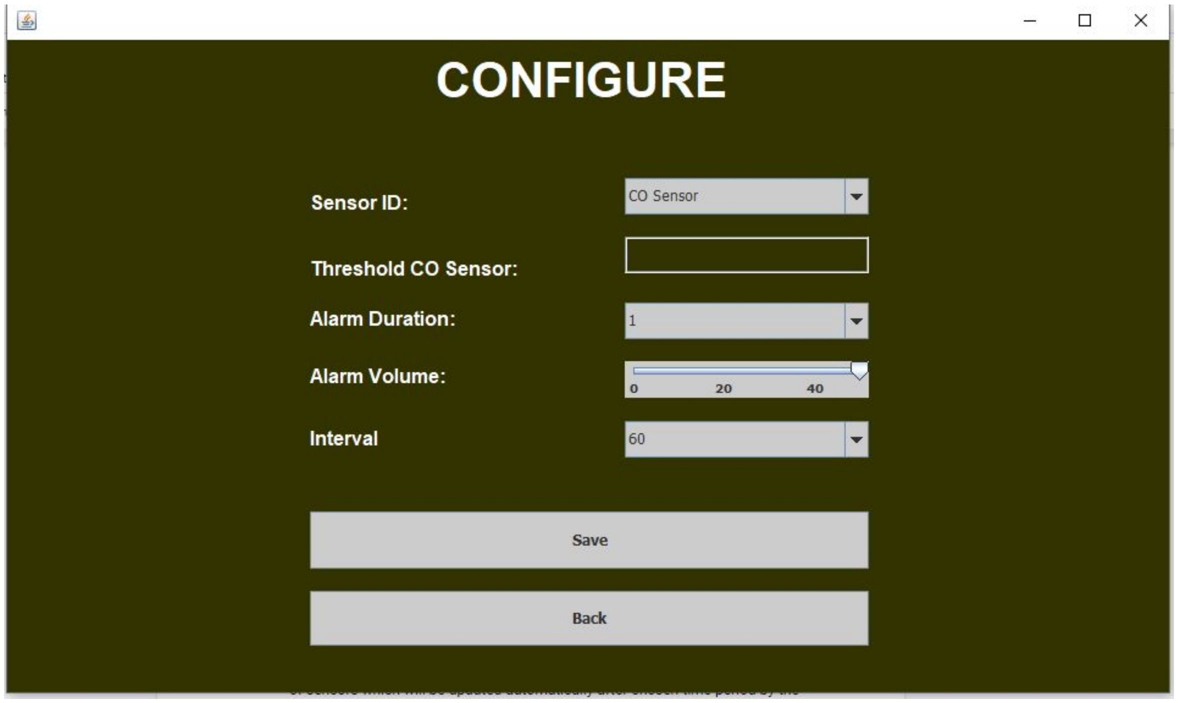
Configure Page:

In this page the sensor is configured. The details required are:

- Sensor ID: <dropdown> (CO/ Heat/ Smoke Sensor)
- Threshold: <Textfield> (Number type validation)
- Duration: <dropdown> (Number type validation)

- Volume: <slider>
- Log Interval: <dropdown> (Number in sec)
- If there’s any irrelevant data entered then, these pop ups let the user know about the correction.

After entering all the details the user will have two options whether to return to the main screen using the back button and save (i.e. configure) the details using the Save Button.



CONFIGURE

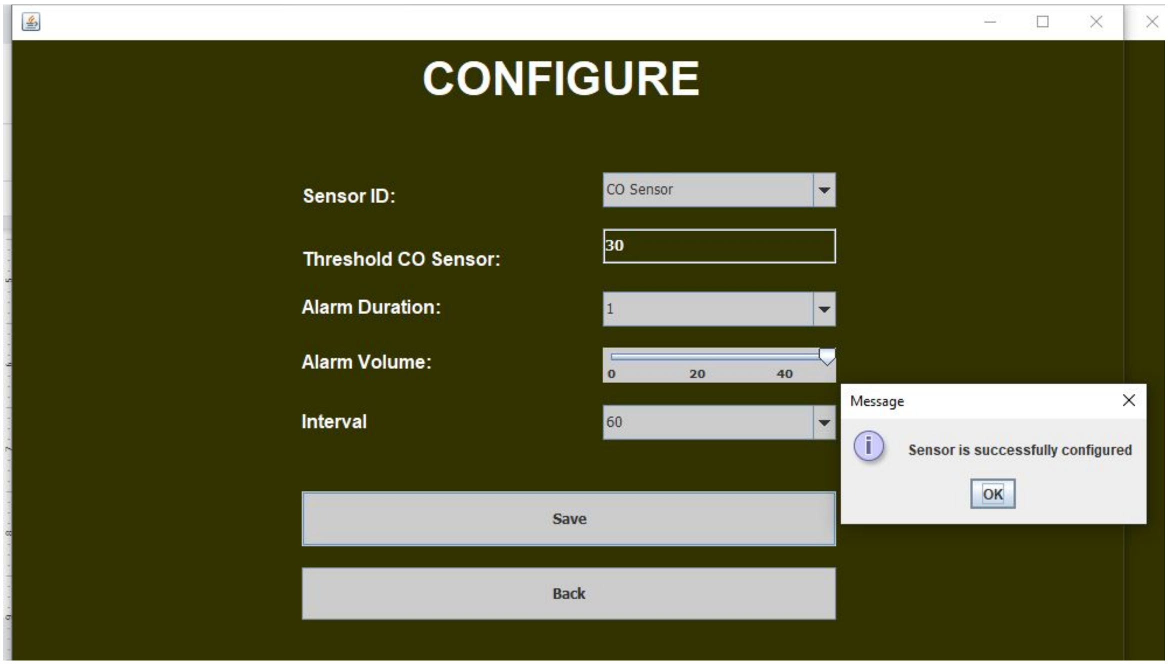
Sensor ID: CO Sensor

Threshold CO Sensor:

Alarm Duration: 1

Alarm Volume:

Interval 60



CONFIGURE

Sensor ID: CO Sensor

Threshold CO Sensor: 30

Alarm Duration: 1

Alarm Volume:

Interval 60

Message

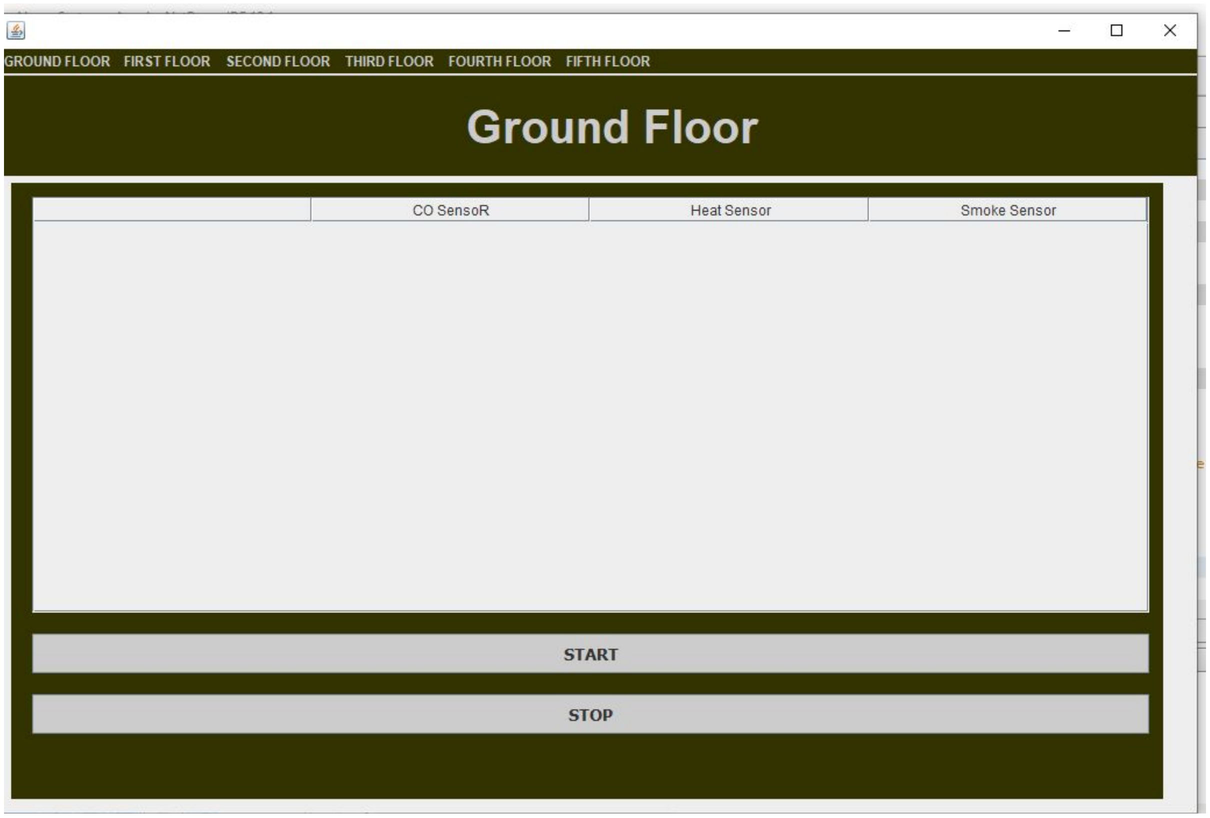
Sensor is successfully configured

Start Monitoring:
In this page user will be able to find two buttons:

- ❖ Start: Reading being sensed at each sensor and the user will get dynamic values of sensors which will be updated automatically after chosen time period by the user.

The values below threshold value are shown in green colour while above the defined threshold are in red colour alerting the user via an email.
Also, Log report is generated and saved for the events at regular intervals.

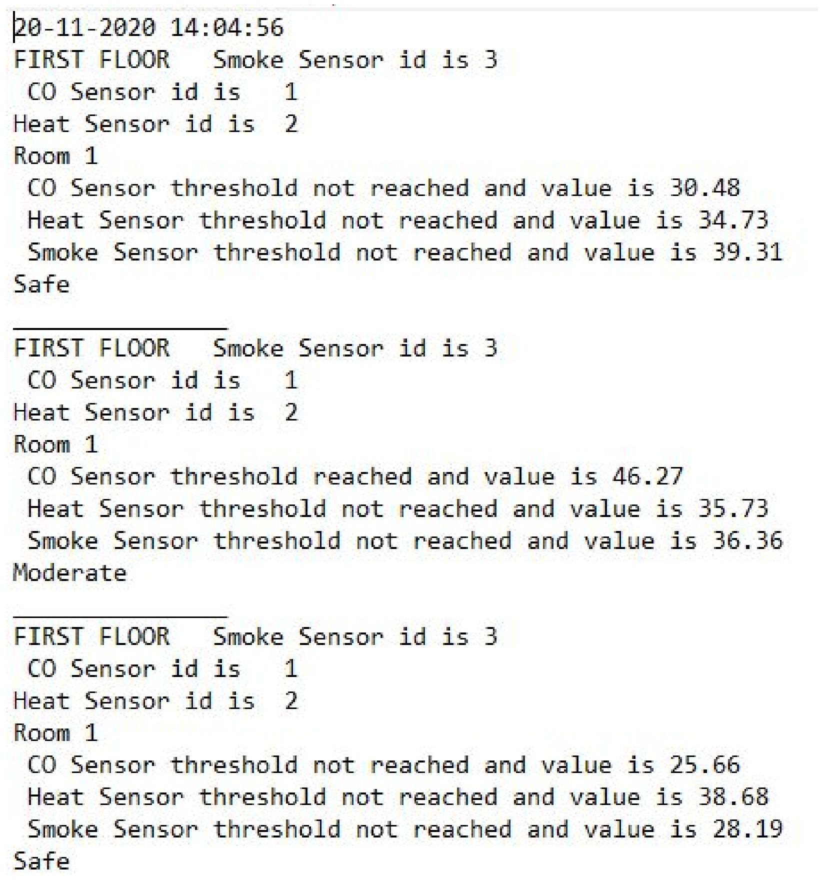
- ❖ Stop: All the sensors which were registered and configured will stop functioning and the log table will be generated which could be seen in the file.



	CO SensoR	Heat Sensor	Smoke Sensor

LOG REPORT

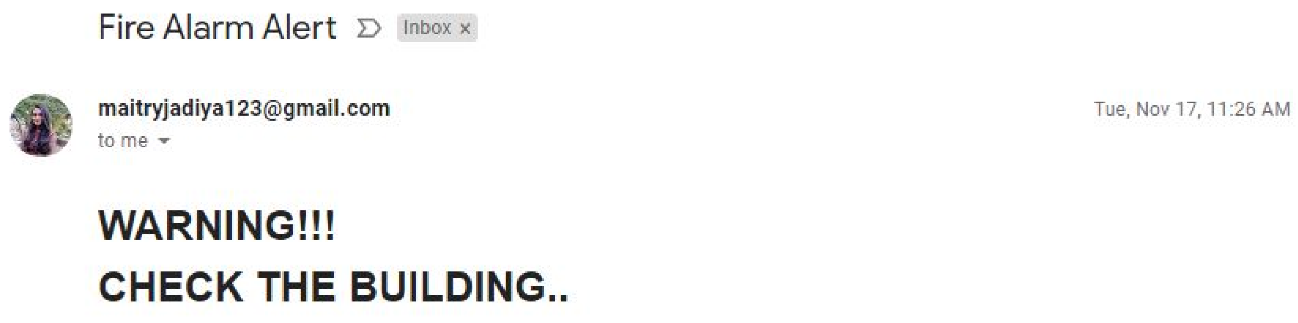
Here, users could see the history of the sensed values by the sensors and this report also consists of details of the device which triggered the alarm as it has the location (i.e. floor no. and room no.) of the sensor in the log generated.



```
20-11-2020 14:04:56
FIRST FLOOR   Smoke Sensor id is 3
 CO Sensor id is   1
Heat Sensor id is  2
Room 1
 CO Sensor threshold not reached and value is 30.48
 Heat Sensor threshold not reached and value is 34.73
 Smoke Sensor threshold not reached and value is 39.31
Safe
_______________
FIRST FLOOR   Smoke Sensor id is 3
 CO Sensor id is   1
Heat Sensor id is  2
Room 1
 CO Sensor threshold reached and value is 46.27
 Heat Sensor threshold not reached and value is 35.73
 Smoke Sensor threshold not reached and value is 36.36
Moderate
_______________
FIRST FLOOR   Smoke Sensor id is 3
 CO Sensor id is   1
Heat Sensor id is  2
Room 1
 CO Sensor threshold not reached and value is 25.66
 Heat Sensor threshold not reached and value is 38.68
 Smoke Sensor threshold not reached and value is 28.19
Safe
```

FEATURES ADDED

- An email is sent to the user when any alarm is triggered i.e. any sensor surpasses the threshold value.



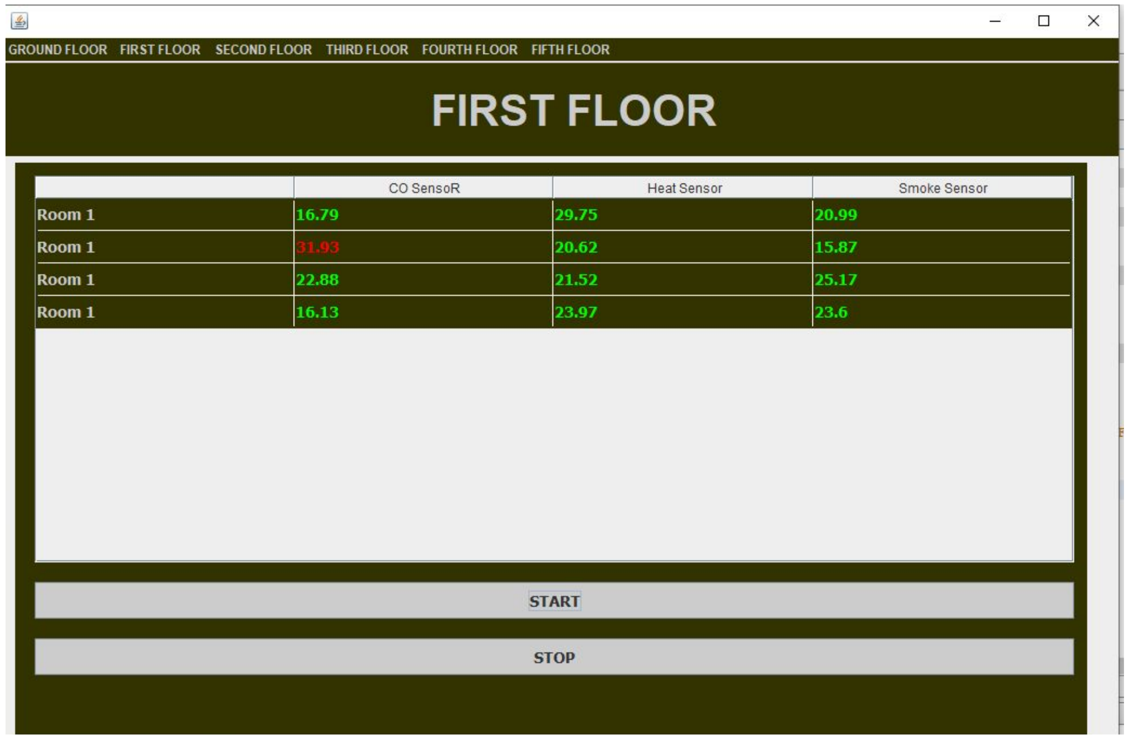
Fire Alarm Alert Inbox ×

maitryjadiya123@gmail.com
to me

Tue, Nov 17, 11:26 AM

WARNING!!!
CHECK THE BUILDING..

- The color of the font of the values is also changed to green and red just to highlight the effect and proper functional buttons are also added to make the system more user friendly.



FIRST FLOOR

	CO SensoR	Heat Sensor	Smoke Sensor
Room 1	16.79	29.75	20.99
Room 1	31.93	20.62	15.87
Room 1	22.88	21.52	25.17
Room 1	16.13	23.97	23.6

HOW IT WORKS ?

The main screens are made using Swing API which provides a better user experience and linking and functioning of pages are then hard coded using required methods and object instances as per needed. Similarly, smaller details like where to use the slider, textbox or dropdown is added.

How do sensors get their values?
A random generator class is defined into which ranvalue method is defined which gives random values varying from threshold/(1.5) to threshold *(1.5).
Threshold value is taken from the user in the configure sensor screen.

How is the log report generated ?
To generate a log report the values are obtained from the table which is made in the start monitoring page i.e. the output obtained by each registered sensor and are appended in the file named as log.txt and the log table can be viewed there with the time and date of the log table.

Once, any one of the sensors attains a value greater than the threshold value the mail is sent to the recipient in which mail package is used.

REFERENCES

- GeeksforGeeks
- Stack Overflow
- Tutorialspoint
- Java: The Complete Reference by Herbert Schildt