PROJECT REPORT

INDUSTRY-SPECIFIC INTELLIGENT FIRE MANAGEMENT SYSTEM

TEAM ID : PNT2022TMID54096

Team Members	Register Number
Gopinath.R.H	111919106016
Lokesh.B	111919106033
Jayasathyanarayanan.M	111919106024
Pragadeeshwaran.T	111919106047

1. INTRODUCTION

1.1 Project Overview

Safety is significant in these days and ages and it is vital that acceptable wellbeing framework be executed in spots of Structural Health Monitoring of structures. This system is used in building and home dwellings for the fire detection and prevention purpose. And it should be implemented in all the establishments where the risk of fire accidents is very high .The sensor nodes are placed in important areas of the building, which we create a network and the monitored data is transmitted to control unit through wireless sensor network and if the temperature or pressure reach above the threshold value and building damage is detected automatically, alerts the surroundings and take necessary precautions to prevent the disaster. This safety system that can be used in any Constructing and constructed environments. The sensor nodes detects the maximum level that it can withhold, in the meantime it calculates where the damage is occurring and remaining time that the building can offer further resistance to damage.

1.2 Purpose

This is an IOT-based fire management system. It can be monitored from anywhere with its status update like gas level, temperature, and also to check if there is any presence of fire. Customer Satisfaction is the main objective which makes them feel comfortable and safe to work or stay in that place. No need to panic, all emergency steps will be taken automatically like sprinkler turn On, ventilation fan turn On and alert message will be sent automatically.It replaces the normal fire alarm system because it has live track data and the record maintenance of the surrounding atmosphere of that place. This helps for future reference. Easy to convey the fire accident message to the fire station so that loss of life is reduced. In future, any update in the software and hardware can be implemented for better performance and accuracy in the fire detection level.

2. LITERATURE SURVEY

2.1 Existing problem

The existing problems of the system are:

Nowadays the fire management system is a fire alarm system which is pretty much used in several areas, but this fire alarm makes people panic at the time of emergency. So we need a smart fire management system to predict the chance of fire accidents occurring and also to inform the fire station and control room if any fire accident has occurred. Mainly to reduce the chance of fire accidents occurring.This fire management system has multiple sensors like temperature, gas, and flame sensors and also has a ventilation fan and water sprinkler. The data collected from the sensor is processed by using the microcontroller to enable the fan if flammable gas is present and the sprinkler will be enabled if any fire and large heat is detected by the flame and temperature sensors. This collected data from the sensors is sent to the cloud using the Wi-Fi connected to the microcontroller. From the cloud an alert message is sent to the fire station and the control room. Additionally, the live status of the

area will be updated to the control room.

2.2References

[1] Automatic Fire Alarm and Fire Control Linkage System in Intelligent Buildings Wang Suli Liu Ganlai School of Information Engineering Support Center JingDeZhen Ceramic Institute JingDeZhen Telecom JingDeZhen,JiangXi Province,China This paper describes a comprehensive program of an office building intelligent systems Fire Control Linkage System subsystem design, At the same time, it describes the following: the idea of the system designing, the system components, selecting equipment, the linkage of alarming and controlling gas extinguishing, and the technical features. Projects under this program have been completed, can realize the intelligent prediction of fire, automatic fire alarm and linkage functions.

[2] A Low-cost R-type Fire Alarm System for Old Houses Song-Shyong Chen1, Luke K. Wang2, Wei–Hsuan Li 2 and Wen Ping Chen2,3,* 1Hsiuping University of Science and Technology, Department of Information Networking Technology 2National Kaohsiung University of Applied Sciences, Department of Electrical Engineering 3Kaohsiung Medical University, Graduate Institute of Clinical Medicine Taiwan, R.O.C.The proposed system is using low-cost Arduino MEGA 2560 Development Board, synthesizing with PLC module, facilitated with QPSK modulating capability. A generic R-type fire alarm system is hereafter constructed, with a master-slave architecture connecting to fire sensors. No extra modification/rehabilitation is needed for those old buildings because the proposed fire alarm system is mounted on buildings' wiring system. Wiring and labor costs are substantially reduced, and the protection of both human and property are assured and guaranteed.

[3] Design of Distributed Factory Fire Alarm System,Mingyu Song, Wuxing Li, Xiaomin Zhang*, Li Liu, Yanke Ci, Xushan Peng,Yongping Li, Haosong Chen Department of Information Engineering, Dahongying University, Zhejiang Province, China, The dangers caused by fires are very great, causing property damage, casualties and environmental damage. Rapid detection of fire hazards and prompt response measures are the best means to reduce the damage caused by fire. The distributed plant fire alarm system can quickly detect the fire and issue an alarm to reduce the damage caused by the fire. The fire alarm system is a control system that integrates signal detection, transmission, processing and control. It mainly completes the basic functions of fire, smoke and temperature module monitoring fire, and studies the multi-point communication of nRF2401 wireless transceiver module.

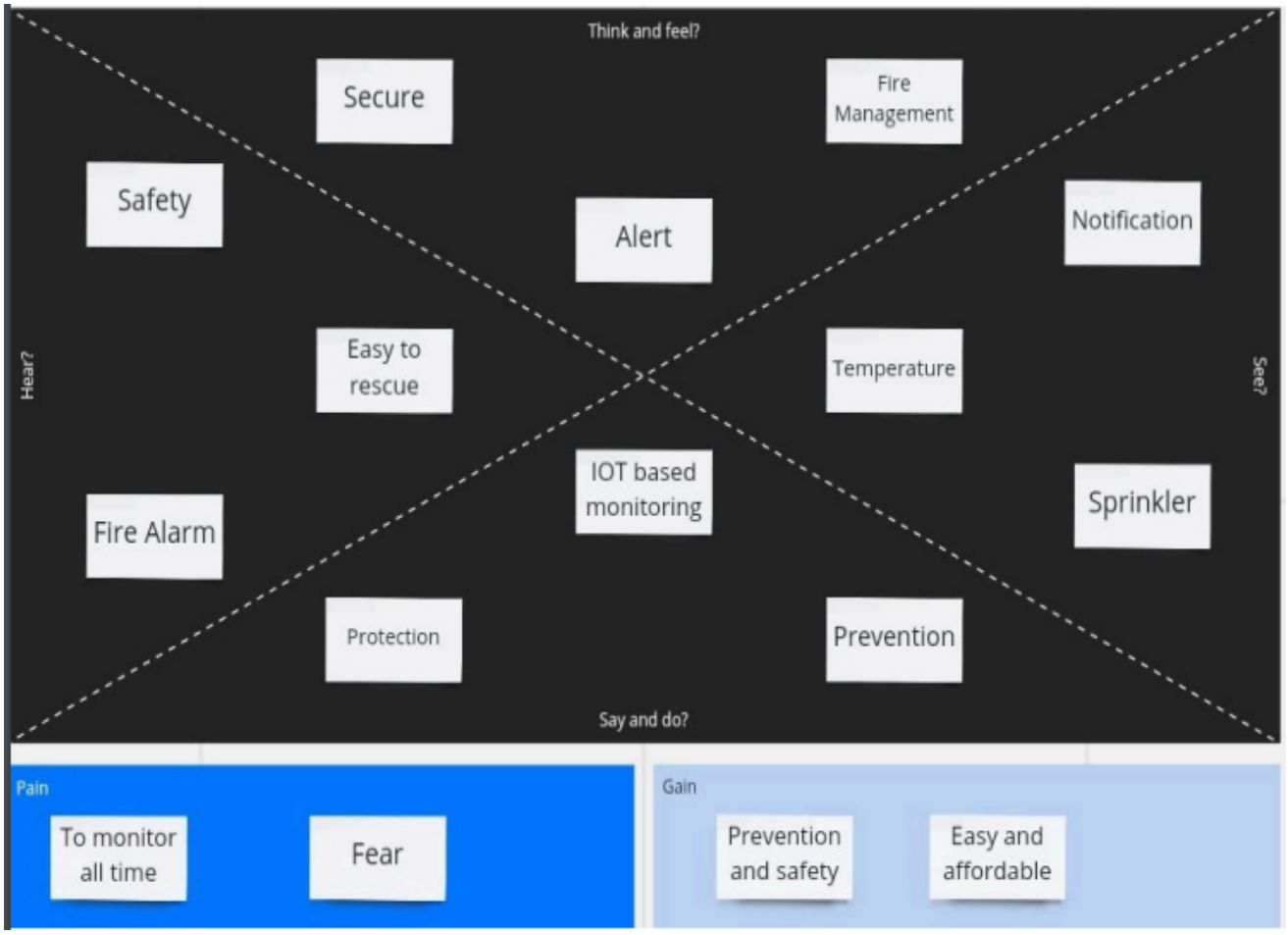
[4] A Wireless Sensor Network for Fire Detection and Alarm System, Patrick Jason Y. Piera, Joseph Karl G. Salva Department of Electrical and Electronics Engineering University of San Carlos Talamban, Cebu City, Philippines, Fire can really be devastating to properties if improperly managed, it is due to this problem that the fire detection and alarm systems were sought for. However, traditional fire alarm system is based on a wiring network which have drawbacks and limitations such as inflexibility of the FDAS layout plan during building construction, and difficulties in renovation where the removal and relocation of traditional FDAS requires additional amount of work. To address these problems, a fire detection and alarm system that is based on wireless sensor network was developed. The FDAS is mainly composed of a fire detection node, a fire alarm node, and a fire alarm control panel. The wireless communication of the nodes was achieved using XBee as the wireless transceiver.

2.3 Problem Statement Definition

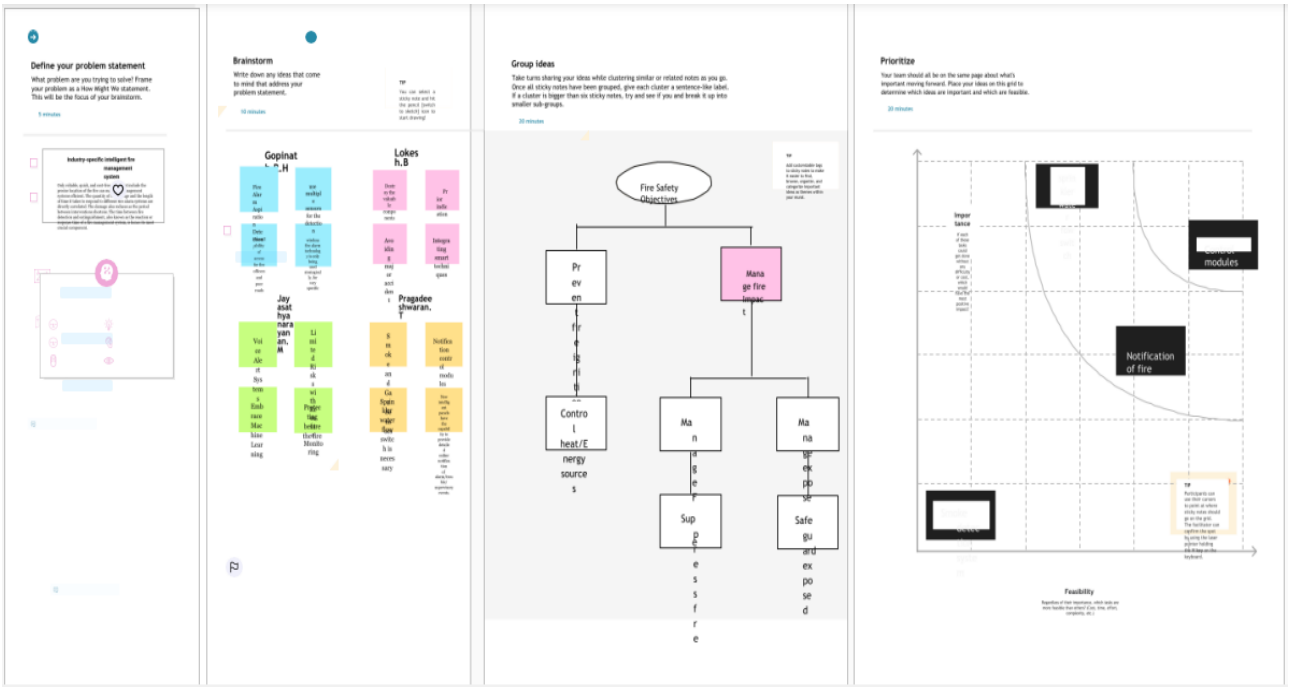
Now a days fire management system is fire alarming system which is pretty much used in several areas, but this fire alarm makes people to panic at the time of emergency. So we need a smart fire management system to predict the chance of fire accident occurring and also to inform the fire station and control room if any fire accident has occurred. Mainly to reduce the chance of fire accident occurring.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation and Brainstorming

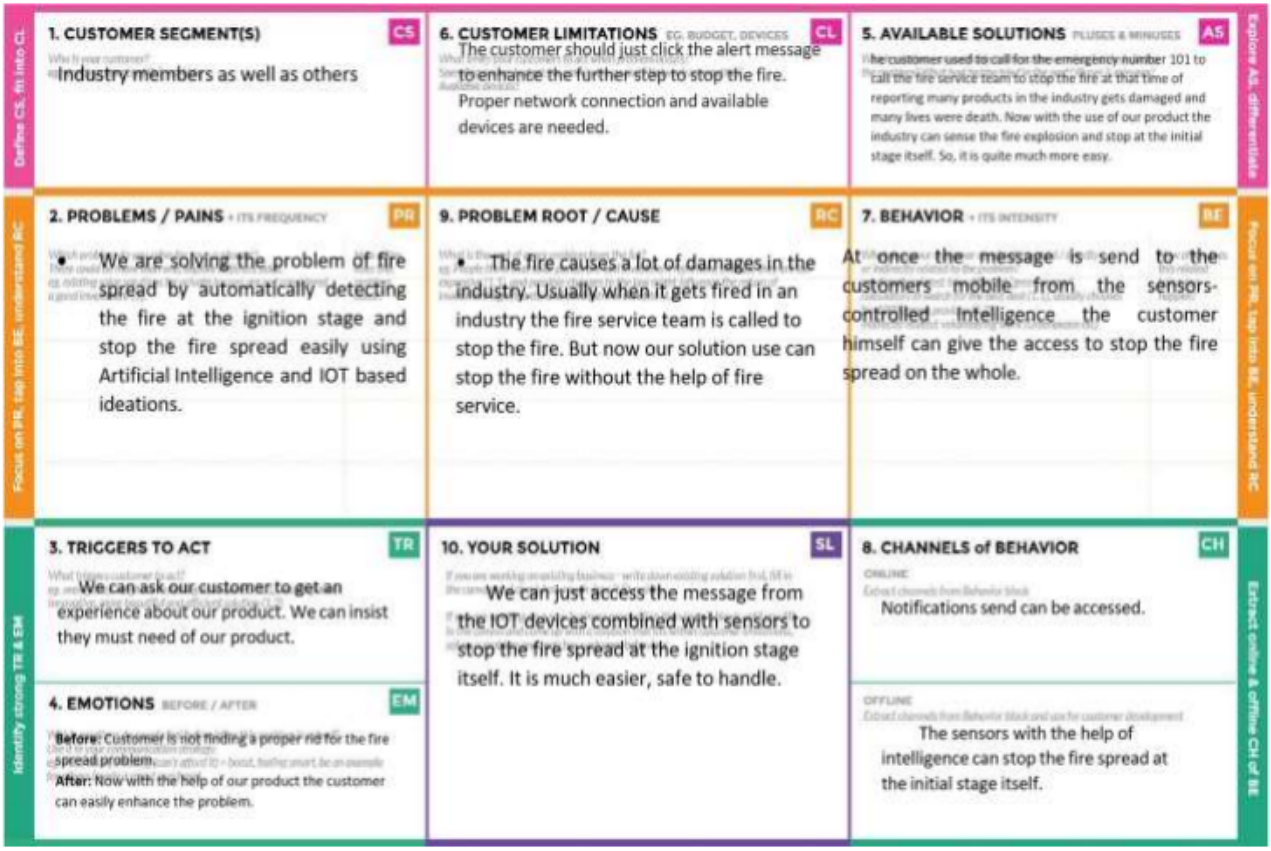


3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Now a day's fire management system is fire alarming system which is pretty much used in several areas, but this fire alarm makes people to panic at the time of emergency. So we need a smart fire management system to predict the chance of fire accident occurring and also to inform the fire station and control room if any fire accident has occurred. Mainly to reduce the chance of fire accident occurring
2.	Idea / Solution description	This fire management system has multiple sensors like temperature, gas, and flame sensors and also has a ventilation fan and water sprinkler. The data collected from the sensor is processed by using the microcontroller to enable the fan if flammable gas is present and the sprinkler will be enabled if any fire and large heat is detected by the flame and temperature sensors. This collected data from the sensors is sent to the cloud using the Wi-Fi connected to the microcontroller. From the cloud an alert message is sent to the fire station and the control room. Additionally, the live status of the area will be updated to the control room.
3.	Novelty / Uniqueness	This is an IOT-based fire management system. It can be monitored from any Where with its like status update like gas level, temperature, and also to check is there any presence of fire.
4.	Social Impact / Customer Satisfaction	It feels comfortable and safe to work or stay in that place. No need to panic, all emergency steps will be taken automatically like sprinkler turn On, ventilation fan turn On and alter message will be sent automatically

5.	Business Model (Revenue Model)	This IOT-based fire management system replaces the normal fire alarm system because it has live track data and the record maintenance of the surrounding atmosphere of that place. This help for future reference. Easy to convey the fire accident message to the fire station so that loss of life is reduced
6.	Scalability of the Solution	In future, any update in the software and hardware can be implemented for better performance and accuracy in the fire detection level. More features can be added in the hardware and software to be user friendly and steps to be taken immediately.

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User cloud account	User should create a IBM cloud account to access the dashboard
FR-2	User notification	User should give the email ID and phone number to send the message alter through email and text message

4.2 Non-functional Requirements:

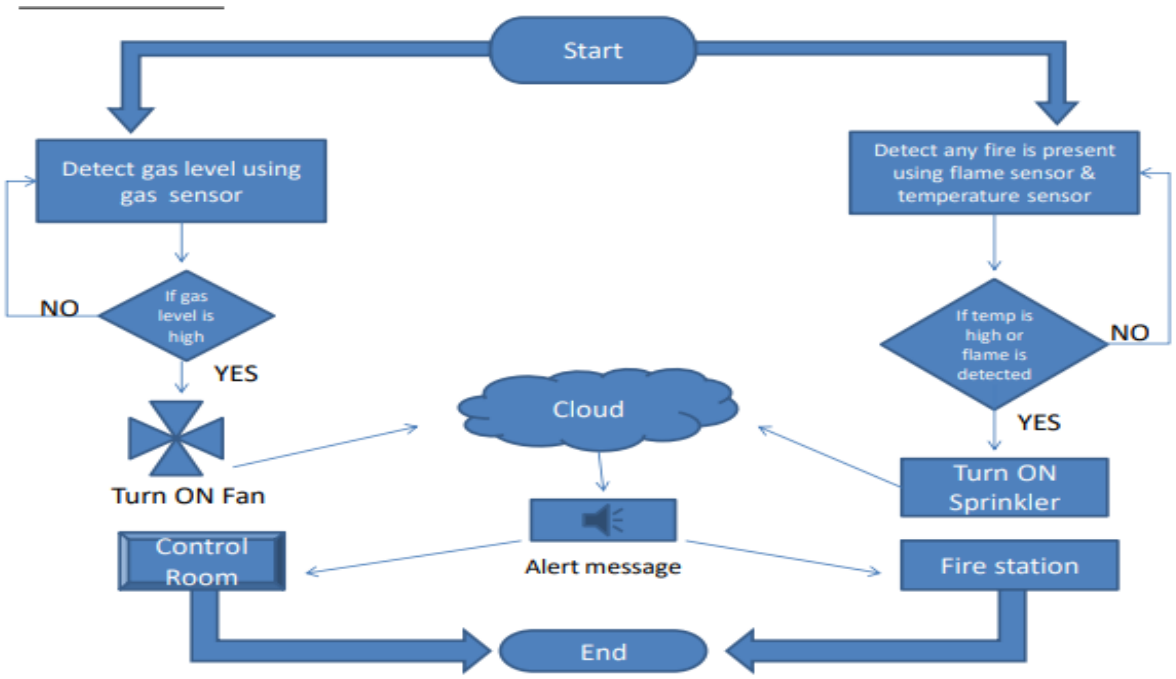
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Should be dynamically update the data in the dashboard.
NFR-2	**Security**	Should be secure enough that the right data is displayed in the dashboard
NFR-3	**Reliability**	Should send alert message immediately when fire is detected
NFR-4	**Performance**	Should send accurate data and should be less latency in message sending.
NFR-5	**Availability**	Should be active for 24/7.

NFR-6	**Scalability**	Should be accessed through any where

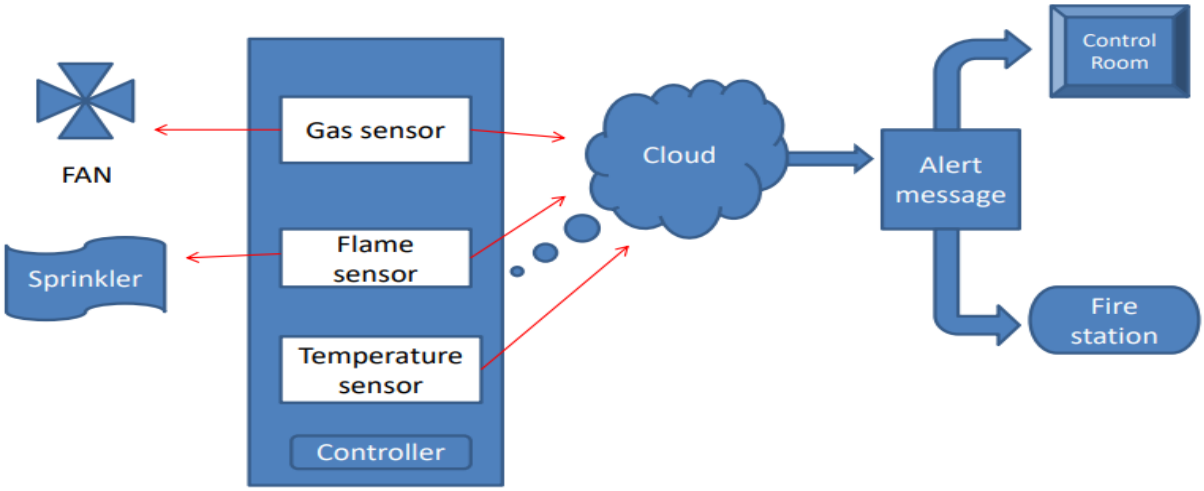
5 PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture

5.2.1 Solution Architecture



5.2.2.Technology Architecture

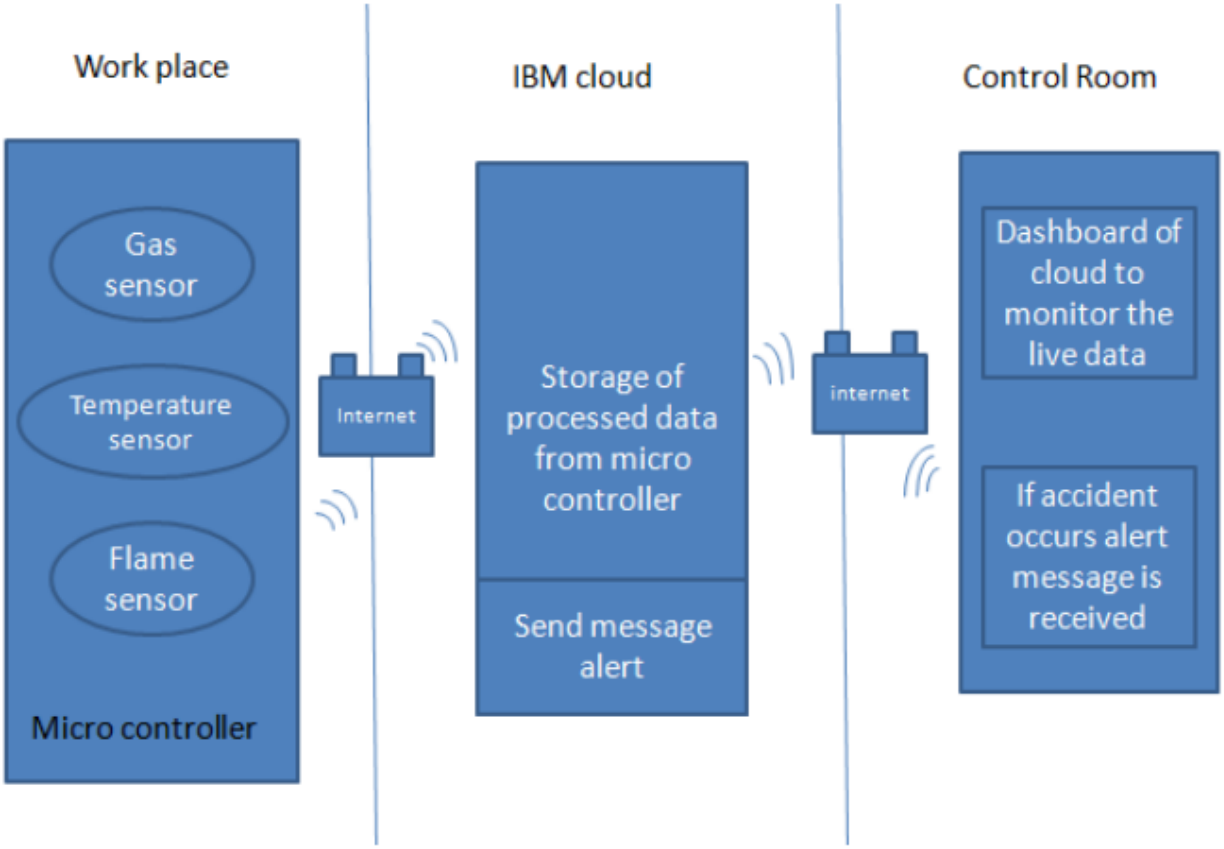


Table-1 : Components & Technologies:

S. No	Component	Description	Technology
1.	User Interface	User can interact with node red applicationdashboard	node red
2.	Gas Sensor	gas sensor used detect presence of flamable gas	Embedded C
3.	Temperature Sensor	Used to detect the level of temperature	Embedded C
4.	Flame sensor	Used to detect the presence of fire	Embedded C
5.	Database	Data Type, Configurations etc.	IBM Cloud
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	Component - File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	Raspberry Pi microcontroller	To process the data from the sensors	Python
9.	Infrastructure Cloud	Application Deployment on Cloud Cloud Server Configuration :	Technology - IBM Cloud

Table-2: Application Characteristics:

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Node Red IBM Watson	IOT, Internet
2.	Security Implementations	Data should be secure and accuracy	Firewall, Firebase
3.	Scalable Architecture	Used to access data from anywhere	IBM Cloud
4.	Availability	Available 24/7 data is transferred all-time	IBM Cloud
5.	Performance	Data are accurate and accessed by internet	IBM Cloud

5.3 User Stories

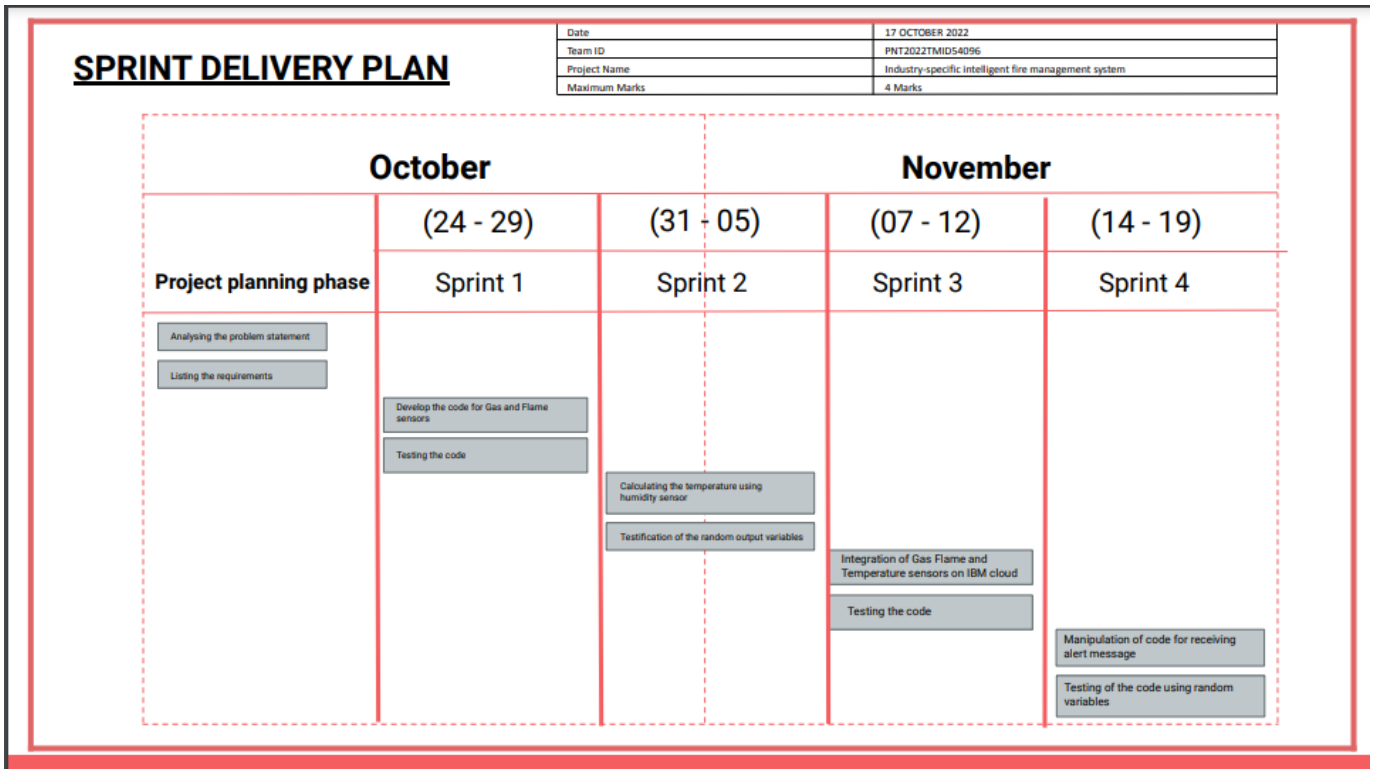
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Gopinath.R.H Lokesh.B Jayasathyanarayanan.M Pragadeeshwaran.T
Sprint-1		USN-2	As a user, I will receive a confirmation email once I have registered for the application	1	High	Gopinath.R.H Lokesh.B Jayasathyanarayanan.M Pragadeeshwaran.T
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	Gopinath.R.H Lokesh.B Jayasathyanarayanan.M Pragadeeshwaran.T
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	Gopinath.R.H Lokesh.B Jayasathyanarayanan.M Pragadeeshwaran.T
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Gopinath.R.H Lokesh.B Jayasathyanarayanan.M Pragadeeshwaran.T
	Dashboard					

6. PROJECT PLANNING & SCHEDULING

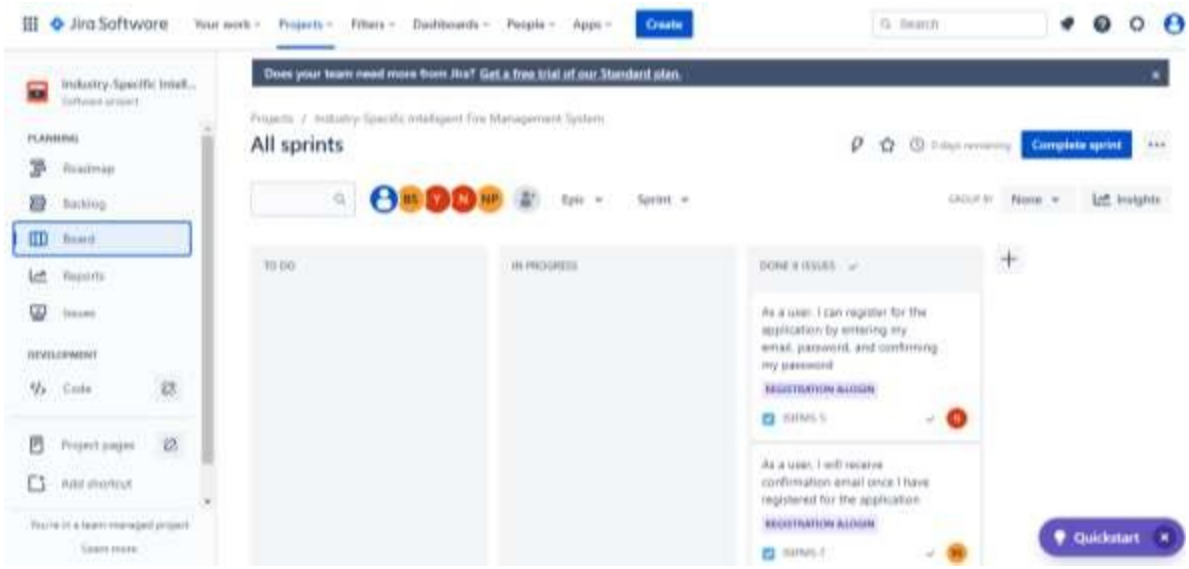
6.1 Sprint Planning & Estimation

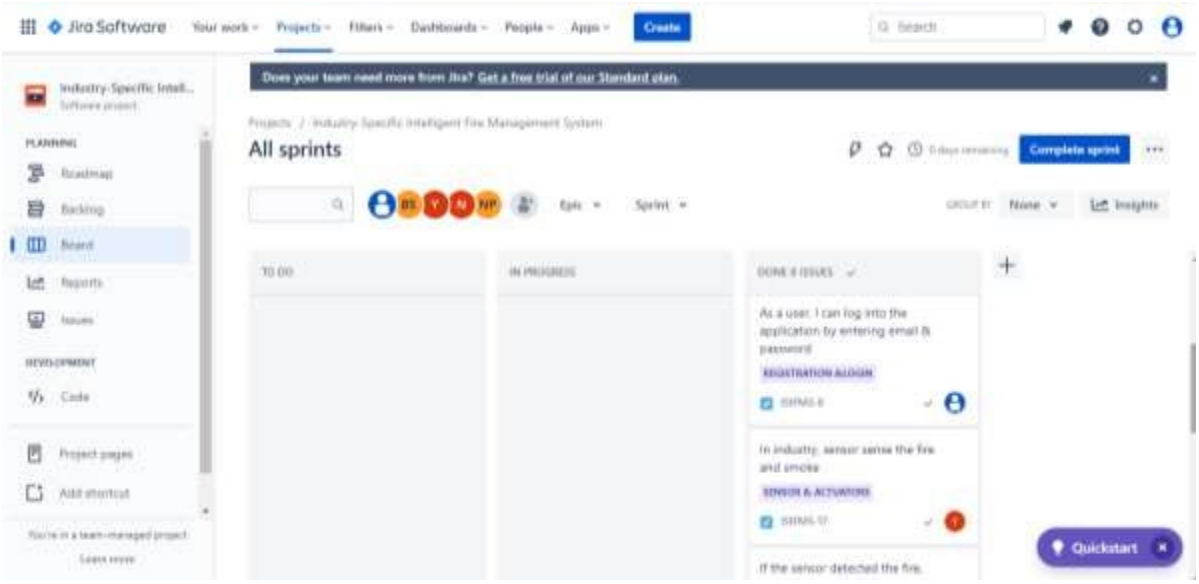
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

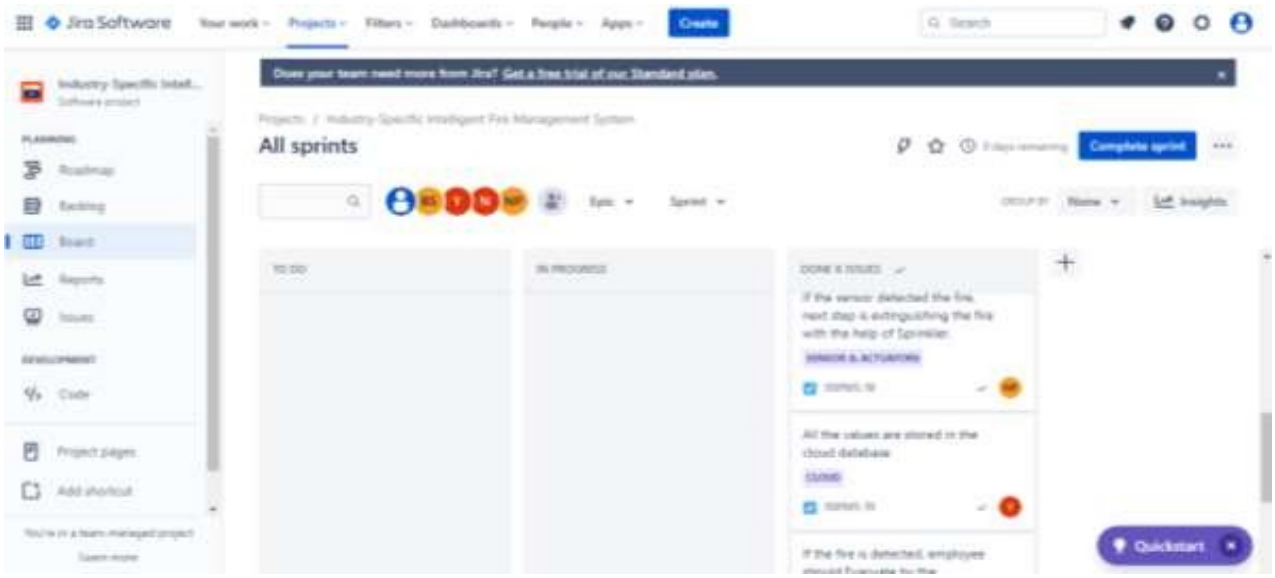
6.2 Sprint Delivery Schedule

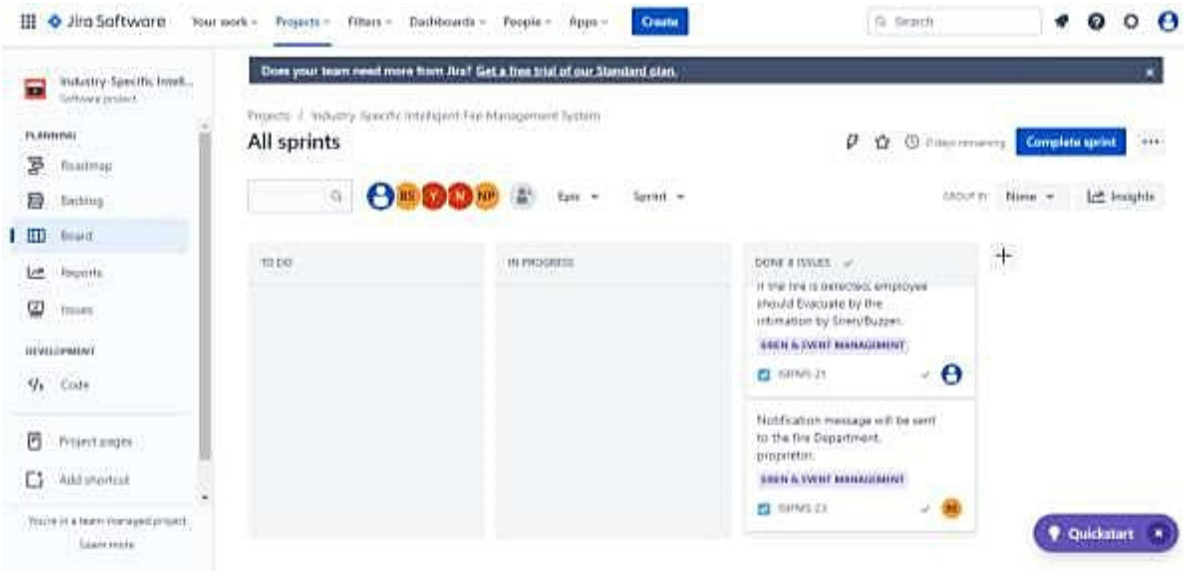


6.3 Reports from JIRA






Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Does your team need more from Jira? Get a free trial of our Standard plan.
All sprints
Complete sprint
Epic
Sprint
Insights
TO DO
IN PROGRESS
DONE 6 ISSUES
Roadmap
Backlog
Board
Reports
Issues
Code
Project pages
Add shortcut
Quickstart


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Does your team need more from Jira? Get a free trial of our Standard plan.
All sprints
Complete sprint
Epic
Sprint
Insights
TO DO
IN PROGRESS
DONE 6 ISSUES
Roadmap
Backlog
Board
Reports
Issues
Code
Project pages
Add shortcut
Quickstart

7. CODING & SOLUTIONING

7.1 Feature 1

Integration of all the sensors

```
#define BLYNK_TEMPLATE_ID "TMPL-uhc59_T"
#define BLYNK_DEVICE_NAME "Fire alert"
#define BLYNK_AUTH_TOKEN "jkfkhu5fzDC9_PBdtssloT9OmXq3THwb"

#define BLYNK_FIRMWARE_VERSION        "0.1.0"

#define BLYNK_PRINT Serial
//#define BLYNK_DEBUG

#define APP_DEBUG

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

#include "DHT.h"
#define DHTPIN 5
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);
char auth[]=BLYNK_AUTH_TOKEN;
char ssid[]="OPPO A52";
char pass[]="6380604277";

int Gas=A0;
int Flame=4;
int buzz=2;
int redLight=3;
int greenLight=4;
float sensorvalue;
int flamevalue;
void setup() {
    pinMode(Gas, INPUT);
    pinMode(Flame, INPUT);
    pinMode(buzz,OUTPUT);
    pinMode(redLight,OUTPUT);
    pinMode(greenLight,OUTPUT);
  Serial.begin(115200);
  Blynk.begin(auth,ssid,pass);
    dht.begin();
    }

void loop() {
     sensorvalue = analogRead(Gas);
```

```
     flamevalue= digitalRead(Flame);
    Blynk.run();
Blynk.virtualWrite(V0,sensorvalue);
    Blynk.virtualWrite(V1,!(flamevalue));
    Serial.print("Gas value:");
    Serial.println(sensorvalue);
    Serial.print("flame state:");
    Serial.println(!(flamevalue));
    float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(t);
    if(flamevalue==0){
        tone(buzz,1000,200);
        digitalWrite(redLight,HIGH);
        digitalWrite(greenLight,LOW);
    }
    else{
        noTone(buzz);
        digitalWrite(redLight,LOW);
       digitalWrite(greenLight,HIGH);
    }
    if(sensorvalue>500){
        tone(buzz,1000,200);
        digitalWrite(redLight,HIGH);
        digitalWrite(greenLight,LOW);
    }
    else{
         noTone(buzz);
        digitalWrite(redLight,LOW);
       digitalWrite(greenLight,HIGH);
}
}
```

7.2 Feature 2

Calculation of Temperature using Humidity Sensor

```
#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);

  dht.begin();
}

void loop() {
  delay(1000);
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(t);
}
```

8. TESTING

8.1 Test Cases

Date	15-Nov-22
Team ID	PNT2022TMID54096
Project Name	Project - Industry-Specific Intellig
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
TC_001	Functional	IBM cloud	Create the IBM Cloud services which are being used in this project.	IBM Cloud Login ID & Password	1.Go to IBM Cloud signup page 2.Enter e-mail id and other credentials 3.Enter a password	https://cloud.ibm.com/login	Should be able to create the IBM Cloud account.	Working as expected	Pass	Results verified	No		Lokesh.B Gopinath.R.H Jayasathyanarayanan.M
TC_002	Functional	IBM Cloud	Configure the IBM Cloud services which are being used in completing this project.	IBM Cloud Login ID & Password	1.Go to Cloud login 2.Enter user ID & Password 3.Verify login by the popup display	https://cloud.ibm.com/	Should able login to IBM Cloud and navigated to IBM Cloud dashboard page	Working as expected	Pass	Results verified	No		Lokesh.B Gopinath.R.H Jayasathyanarayanan.M
TC_003	Functional	IBM Watson IoT Platform	IBM Watson IoT platform acts as the mediator to connect the web application to IoT devices, so create the IBM Watson IoT platform.	IBM Watson IoT Platform Login ID & Password	1.Login to IBM Cloud 2.Click Catalog 3.Search IoT and click create 4.Go to resource list and search Internet of Things platform 5.Press Launch and click Sign in IBM Watson Platform	https://ilph7t.internetofthings.ibmcloud.com/dashboard/devices/browse	Should be able to navigate to IBM IoT Watson Platform	Working as expected	Pass	Results verified	No		Lokesh.B Gopinath.R.H Jayasathyanarayanan.M Pragadeeshwaran.T
TC_004	Functional	IBM Watson	To create a device in the IBM Watson IoT platform and get the device credentials.	IBM Watson IoT Platform Login ID & Password	1.Login to IBM Watson Platform 2. Click Add Device 3.Enter the details and click Finish. Note down the Device ID, Device Name, Authentication key.	Device Credentials	Should be able to get Device details	Working as expected	Pass	Results verified	No		Lokesh.B Gopinath.R.H Jayasathyanarayanan.M Pragadeeshwaran.T
TC_005	Functional	IBM Cloud(Node Red)	Configure the connection security and create API keys that are used in the Node-RED service for accessing the IBM IoT Platform.	Node Red Installation	1. Search "Node-red" in catalog 2. Wait for some time to completely configure the Node-Red.	https://cloud.ibm.com/apps/88fda362-1eda-4920-85eb-11e5b6c8f4fb?ace_config=%7B%22region%22%3A%22eu-gb%22%2C%22crn%22%3A%22crn%3Av1%3Abluemix%3Apublic%3Acf%3Aeu-gb%3As%2Ffdb114aa-278c-4dbd-970f-83535d92b25d%3A%3Acf-application%3A88fda362-1eda-4920-85eb-11e5b6c8f4fb%22%2C%22resource_id%22%3A%2288fda362-1eda-4920-85eb-11e5b6c8f4fb%22%2C%22orgGuid%22%3A%22ca907961-2e66-4411-b1ea-0c39c68cc6eb%22%2C%22spaceGuid%22%3A%22fdb114aa-278c-4dbd-970(	Should be able to open Node-Red service	Working as expected	Pass	Results verified	No		Lokesh.B Gopinath.R.H Jayasathyanarayanan.M Pragadeeshwaran.T

Date	15-Nov-22
Team ID	PNT2022TMID54096
Project Name	Project - Industry-Specific Intelli
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
TC_008	Functional	Python 3.7.0	After developing python code, commands are received just print the statements which represent the control of the	ython 3.7.0(64 bit) installatio	1.Download and install Python 3.7.0 2. Open Node-Red or FAST SMS	et the output from the cod	Should be able to display the commands like Exhaust Fan ON, Exhaust Fan OFF	Working as expected	Pass	Results verified	No		Lokesh.B Gopinath.R.H
TC_009	Functional	IBM Cloudant DB	Store the sensor values - Temperature, Flame Level and Gas Level in the Cloud	IBM Cloud Account	1.Run the python code 2.Verify the displayed output	Output from the python cod	Should be able to store the sensor values generated by the python script in the cloud	Working as expected	Pass	Results verified	No		
TC_010	Web UI	Node Red & Fast SMS	Create Web UI in Node- Red	Fast SMS Login ID & password	1.Go to Node Red. Select http in & http response. Add functions and select another http in and http response. Connect them to IBM IoT output and function.Print the command statements such as Alarm ON/OFF and sensor 2.Go to the Fast sms and copy the url and paste it on Node	Sensors values and message will be sent	Message will be generated and send to the expected numbers.	Working as expected	Pass	Results verified	No		Gopinath.R.H Jayasathyanarayanan.M Lokesh.B Pragadeeshwaran.T
TC_011	Functional	IBM Cloudant DB	Configure the Node-RED flow to receive data from the IBM IoT platform and also use Cloudant DB nodes to store the received sensor data in the cloudant DB	IBM Cloud Login ID & Password	1.Go to IBM cloud, search Cloudant in Catalog, Add new dashboard, go to Node Red 2.Connect to cloudant and verify the results	Cloudant is connected in the NODE RED	User should be able to connect the Cloudant and Node Red and be able to see the created cloud database with the sensor values	Working as expected	Pass	Results verified	No		Gopinath.R.H

8.2 User Acceptance Testing

Purpose of Document : The purpose of this document is to briefly explain the test coverage and open issues of the Industry-specific intelligent fire management system project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis :

Section	Total Cases	Not Tested	Fail	Pass
Print the Sensor values	7	0	0	7
Client Mobile Application	51	0	0	51
Security	2	0	0	2

his report shows the number of resolved or closed bugs at each severity level, and how theywere resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	0	0	1	8
Totals	24	14	13	26	70

Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9. RESULTS

9.1 Performance Metrics

NFT - Risk Assessment								
S.No	Project Name	Scope/Feature	Functional Changes	Hardware Changes	Software Changes	Load/Volume Changes	Risk Score	Justification
1	Light ON/OFF	Existing	Low	No Changes	Low	>5 to 10%	GREEN	Changes occurs less
2	Fast SMS	New	No changes	No Changes	Low	>5 to 10%	GREEN	Changes occurs hardly
3	Sensor values	Existing	Moderate	No Changes	Moderate	>10 to 30%	ORANGE	Some changes occurs

NFT - Detailed Test Plan				
S.No	Project Overview	NFT Test approach	Approvals/SignOff	Assumptions/Dependencies/Risk
1	Python script	Python coding	https://www.python.org/url?q=query#heroku	Depend on the delivered code
2	Node Red	Sensor & command values	https://nodered.org/	Sensor values
3	Fast sms	Alarm/Sensor notification	https://www.fast2sms.com/	Notifications

End Of Test Report								
S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Identified Defects (Detected/Closed/Open)	Recommendations	Approvals/SignOff
1	Python Code	Python coding	Met	Pass	GO	Closed	Efficient code	https://www.python.org/url?q=query#heroku
2	Node Red	Sensors&command values	Met	Pass	GO	Closed	Sensing the values perfectly	https://nodered.org/
3	Fast sms	Light/Sensor notification	Met	Pass	GO	Closed	Notifies the users at correct time	https://www.fast2sms.com/

10.ADVANTAGES AND DISADVANTAGES

The Advantages of this Industry-Specific Intelligent Fire Management system are as follows

- The user need not require expertise knowledge to control this system. This system is simple. The user can easily view the sensor values and take control actions.
- The control actions are taken automatically.
- If it is implemented in hardware, then the cost of implementation will be affordable.
- As we are sensing the sensor values continuously, any slight change in the environmentis detected
- This system is in User-Friendly format.

The Disadvantage of this Industry-Specific Intelligent Fire Management system are as follows

- This system will not be able to reason out the origin of fire.
- This system will not provide the escape route if there is fire outbreak.
- If the industry has specific changes in the environment, then this system will gives false alarm.

11.CONCLUSION

On completion of this project, We can conclude that the gas leak detection in the project system is amazing. Successfully applicable in industrial scenarios. In

dangerous situations we can save valuable human lives with this system. The sensor module externally out throws an alert. A gas sensor node detects gasses such as CO2, oxygen, propane.Whereas a fire sensor node detects for a heated environment and a temperature sensor is used to show the particular temperature at the surroundings.Finally all the random variables are linked through cloud computing technologies and the alert is sent to the required person who has the access to the clouding platform through necessary login credentials where the obtained output is organized , manipulated and displayed.

12.FUTURE SCOPE

The future scope of this project is to add additional features like triggering the extinguisher automatically, predict the escape route if the fire outbreaks ,to show the live status through smart watches and to implement this system in real time embedded systems

13.APPENDIX

Source Code

```
#include<SoftwareSerial.h>
#include<TinyGPS.h>

SoftwareSerial gsm(7,8);

#define BLYNK_TEMPLATE_ID "TMPL-uhc59_T"
#define BLYNK_DEVICE_NAME "Fire alert"
#define BLYNK_AUTH_TOKEN "jkfkhu5fzDC9_PBdtssloT9OmXq3THwb"

#define BLYNK_FIRMWARE_VERSION        "0.1.0"

#define BLYNK_PRINT Serial
//#define BLYNK_DEBUG

#define APP_DEBUG

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

#include "DHT.h"
#define DHTPIN 5
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);
char auth[]=BLYNK_AUTH_TOKEN;
char ssid[]="OPPO A52";
char pass[]="6380604277";

int Gas=A0;
int Flame=4;
int buzz=2;
int redLight=3;
int greenLight=4;
float sensorvalue;
int flamevalue;
void setup() {
    pinMode(Gas, INPUT);
    pinMode(Flame, INPUT);
    pinMode(buzz,OUTPUT);
    pinMode(redLight,OUTPUT);
    pinMode(greenLight,OUTPUT);
  Serial.begin(115200);
  Blynk.begin(auth,ssid,pass);
    dht.begin();
    gsm.begin(9600);
    }
```

```
47
48 void loop() {
49      sensorvalue = analogRead(Gas);
50      flamevalue= digitalRead(Flame);
51     Blynk.run();
52  Blynk.virtualWrite(V0,sensorvalue);
53     Blynk.virtualWrite(V1,!(flamevalue));
54     Serial.print("Gas value:");
55     Serial.println(sensorvalue);
56     Serial.print("flame state:");
57     Serial.println(!(flamevalue));
58     float h = dht.readHumidity();
59   float t = dht.readTemperature();
60
61   if (isnan(h) || isnan(t)) {
62     Serial.println("Failed to read from DHT sensor!");
63     return;
64   }
65
66   Serial.print("Humidity: ");
67   Serial.print(h);
68   Serial.print(" %\t");
69   Serial.print("Temperature: ");
70   Serial.print(t);
71     if(flamevalue==0){
72         tone(buzz,1000,200);
73         digitalWrite(redLight,HIGH);
74         digitalWrite(greenLight,LOW);
75         gsm.println("AT+CMGF=1\r");
76         delay(1000);
77         gsm.print("AT+CSMP=17,167,0,0\r");
78         delay(1000);
79         gsm.println("AT+CMGS=\"+916380604277\"\r");//replace x by
   your number
80         delay(1000);
81         gsm.write("Fire alert\n");
82         delay(100);
83         gsm.write("location:Latitude:13.0663,Longitude:80.1112
   currentlocation:http://maps.google.com/maps?&z=15&mrt=yp&t=k&q=13
   .0663,80.1112");
84         delay(100);
85          gsm.println((char)26);
86          delay(1000);
87     }
88     else{
89         noTone(buzz);
90         digitalWrite(redLight,LOW);
91        digitalWrite(greenLight,HIGH);
```

```
    }
    if(sensorvalue>500){
        tone(buzz,1000,200);
        digitalWrite(redLight,HIGH);
        digitalWrite(greenLight,LOW);
        gsm.println("AT+CMGF=1\r");
        delay(1000);
        gsm.print("AT+CSMP=17,167,0,0\r");
             delay(1000);
             gsm.println("AT+CMGS=\"+916380604277\"\r");//replace
   x by your number
             delay(1000);
              gsm.write("Gas has Leakead\n");
              delay(100);

   gsm.write("location:Latitude:13.0663,Longitude:80.1112
   currentlocation:http://maps.google.com/maps?&z=15&mrt=yp&t=k&q=13
   .0663,80.1112");
              delay(100);
              gsm.println((char)26);
              delay(1000);
         }
         else{
              noTone(buzz);
             digitalWrite(redLight,LOW);
            digitalWrite(greenLight,HIGH);
     }
     }
```

Github & Project Demo Link

Github Link:

https://github.com/IBM-EPBL/IBM-Project-49803-1660840347

Project Demo Link:

https://github.com/IBM-EPBL/IBM-Project-49803-1660840347/blob/0c31ae3a2149a0ccdc5ceae791a69c2179c9fe84/Final%20Deliverables/Project%20Demo%20Video.mp4