Industry Specific Intelligent Fire Management System

Team ID : PNT2022TMID02619

Project ID : IBM-Project-17285-1659633419

1. INTRODUCTION

1. Project Overview

The Industry specific intelligent fire management system is used to detect fires and protect workers in an industry in case. This system can also be used to detect gas leakage in the industry.

2. Purpose

The proposed system will detect fires and gas leakage, thereby protecting life and property from excessive damage.

2. LITERATURE SURVEY

1. Existing problem

According to gas leakage with auto ventilation and smart management system using IOT paper, there is either a gas detection system or a fire detection system alone. According to fire alarm system based on IOT paper, whenever the system detects smoke in houses, offices, banks etc. It alerts the people inside and makes them response quickly but this happens only during working hours. According IOT based fire and gas monitoring system paper, the existing has only fire or gas or temperature sensors.

2. References

1. IOT based fire and gas monitoring system published by Aayush Doshi, Yashraj Rai

2. An automated fire supression mechanism controlled using an arduino published by R.I. Rashid, S.M. Rafid, A. Azad

3. Survey of internet of things in fire detection and fire industries published by S.R.Vijayalakshmi, S.Muruganand

4. IOT safety and security system in smart cities published by El-Hadi Khoumeri, Rabea Cheggou, and Kamila Farhah

3. Problem Statement Definition

To detect fire and gas leakage in industries using IOT and to reduce the loss of life and properties by deploying IOT based fire detection system.

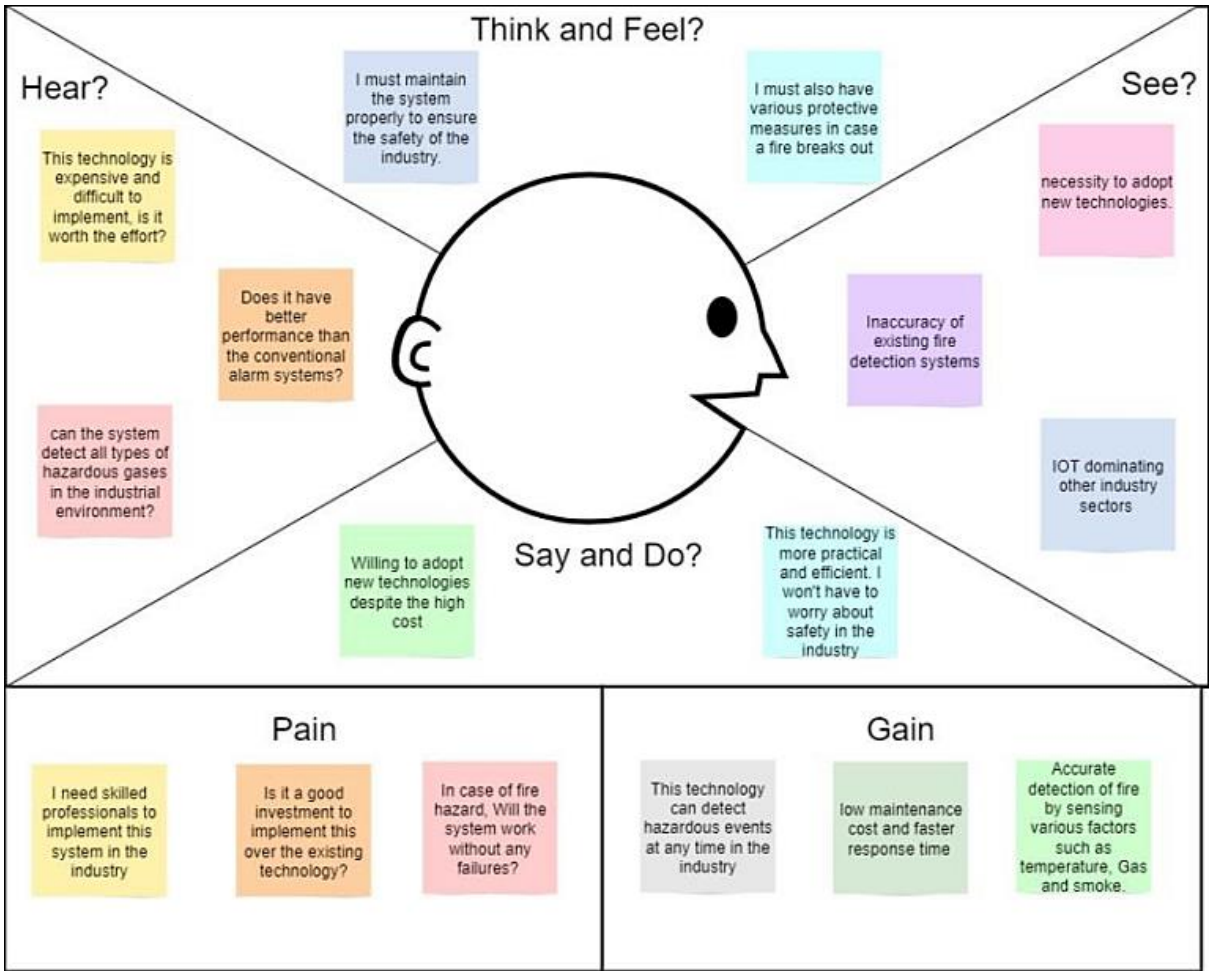
3. IDEATION & PROPOSED SOLUTION

1. Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

It is a useful tool to helps teams better understand their users.

Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

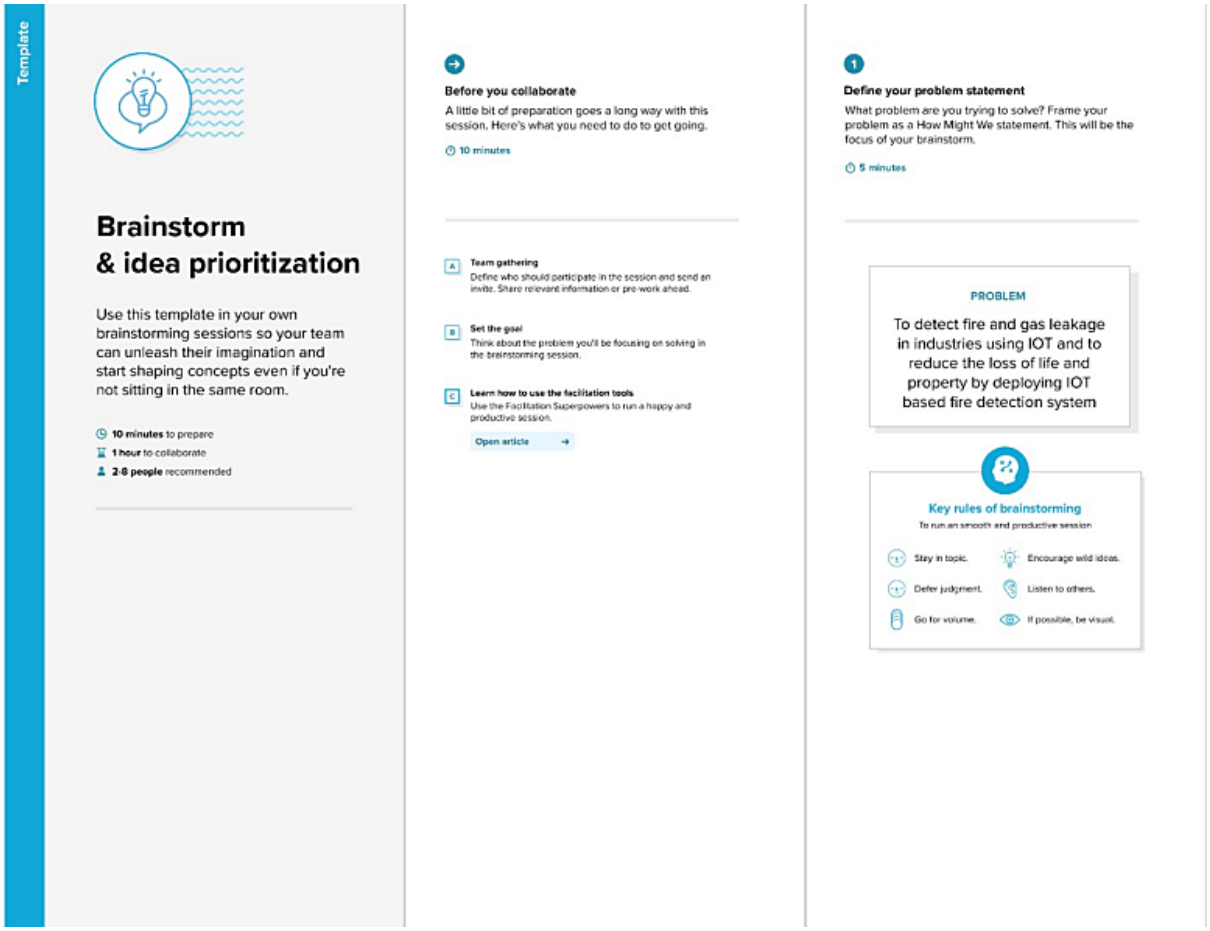


2. **Ideation & Brainstorming**

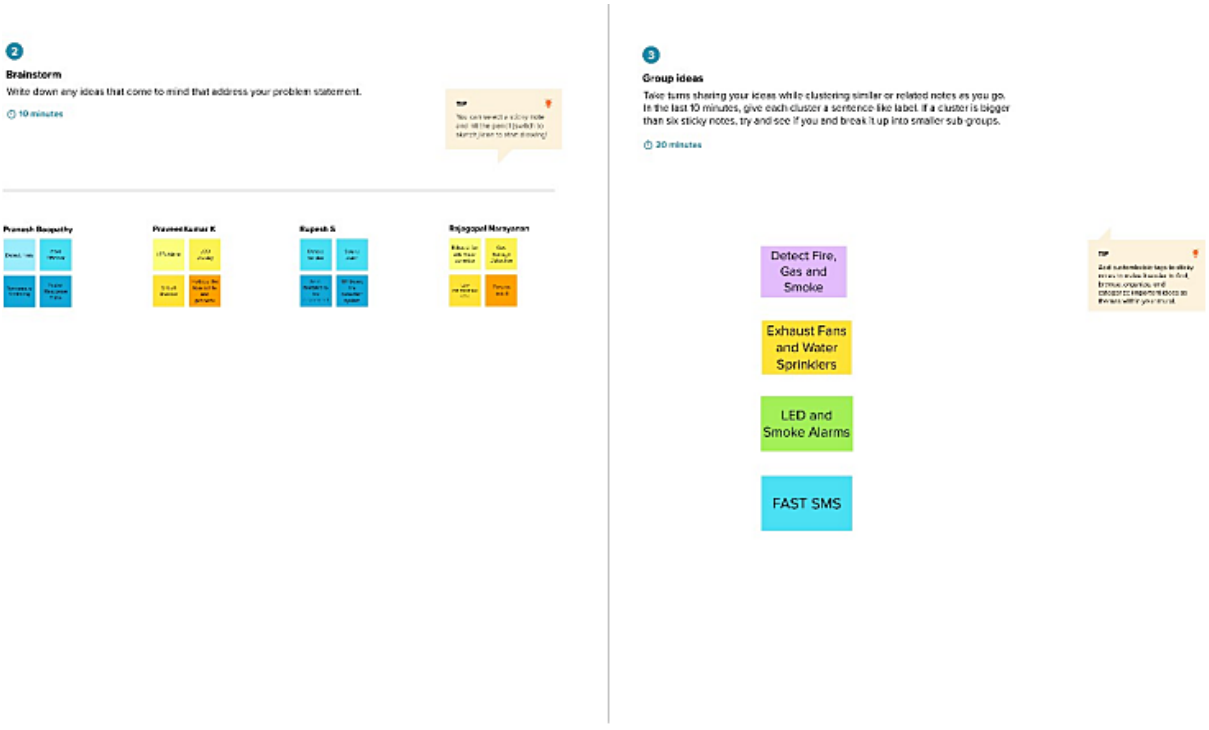
Brainstorming provides a free and open environment that encourages everyone wit hin a teamto participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants areencouraged to collaborate, helping each other develop a rich amount of creativ e solutions.

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

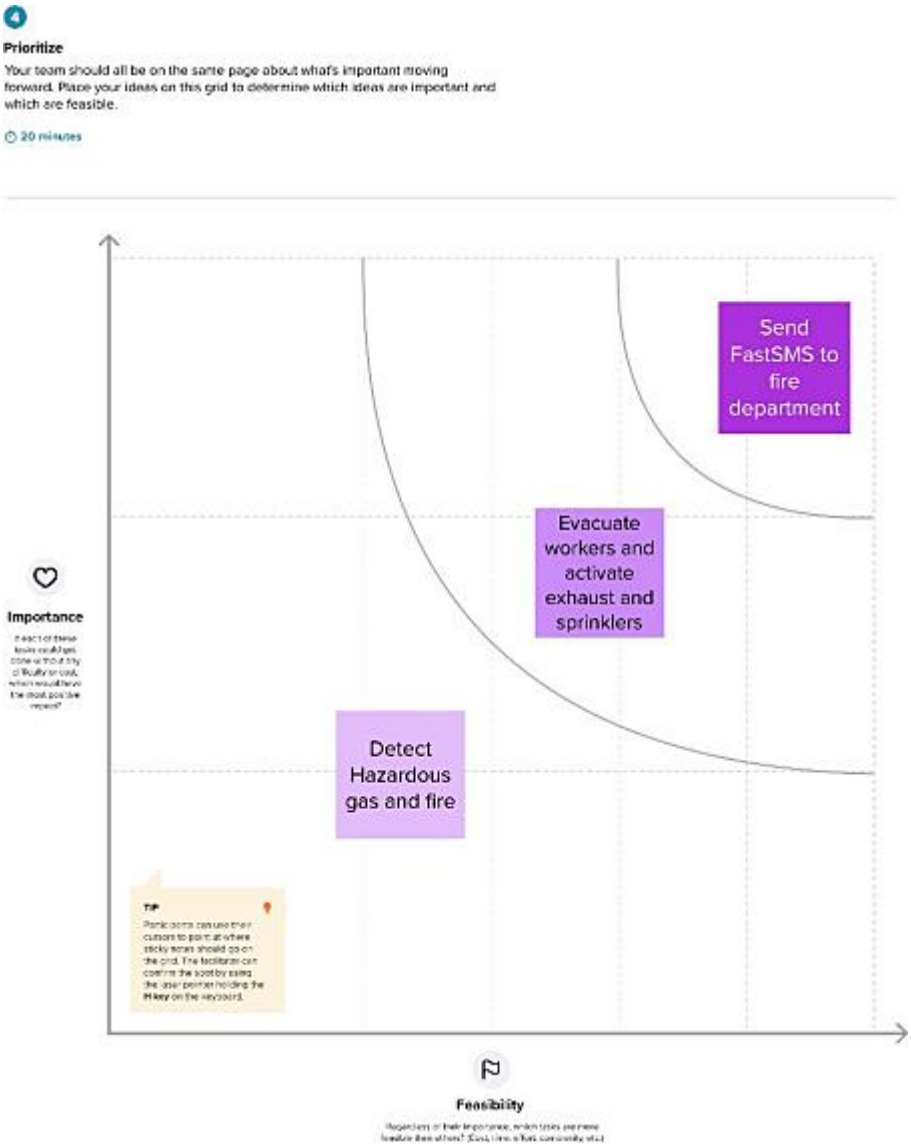
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization

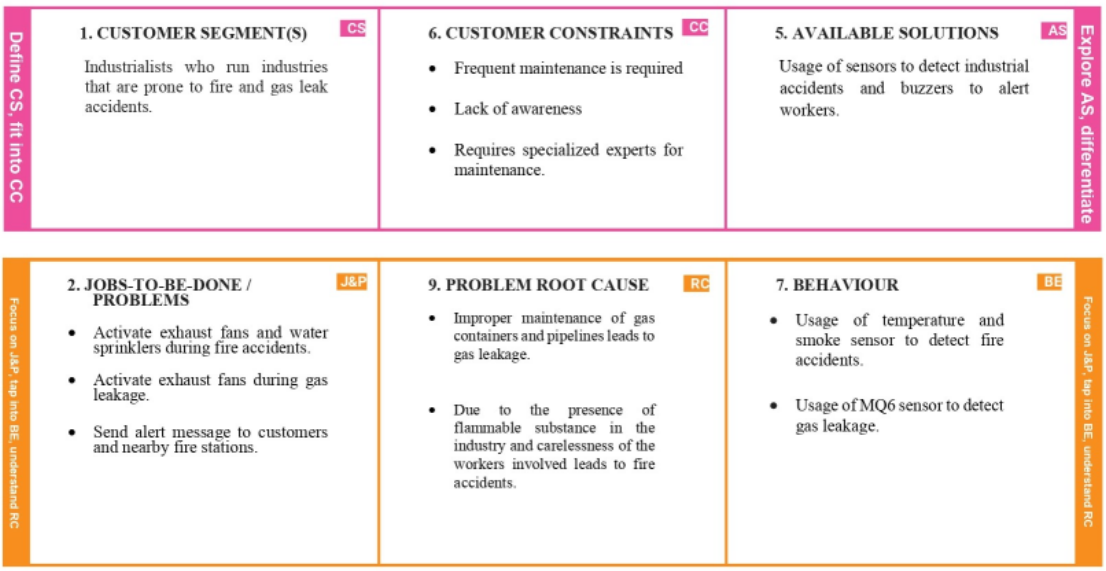


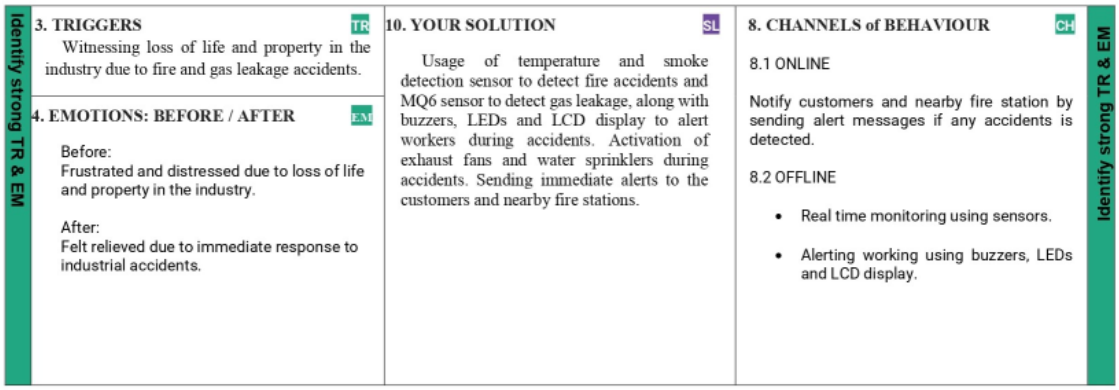
3. **Proposed Solution**

S. No.	Parameter	Description
1.	Problem Statement (Problem to besolved)	Many accidents occur due to fire and gas leakage in industries which leads to loss of life and property.
2.	Idea / Solution description	We can provide solution by detecting Fire, Gas,Temperature and smoke with the help of IOT technology and sensors.

3.	Novelty / Uniqueness	An integrated system of temperature sensor, gas leakage sensor, flame sensor, buzzer, LCD display. The LCD display displays SAFE and ALERT messages. It provides fast and accurate messages.
4.	Social Impact / Customer Satisfaction	• It is used to prevent accidents caused by fire and gas leakage. • Highly accurate. • Easy installation and maintenance. • Cost efficient.
5.	Business Model (Revenue Model)	• Since it is easy to maintain and install so any industry can use our model. • Our model will help industries by preventing huge losses that occur due to fire accidents.
6.	Scalability of the Solution	Since our model is cost effective, any and every kind of industry can use our Industry Specific Intelligent Fire Management System

4. Problem Solution fit





4. REQUIREMENT ANALYSIS

1. Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration Manually
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP Confirmation via Phone call
FR-3	Payment Options	Cash on Delivery Net Banking Credit/Debit card
FR-4	Delivery and Installation	Doorstep Delivery Manual Installation and 1 year warranty Take Away
FR-5	Feedback	Via phone calls
FR-6	Fire and Gas leakage Monitoring	Temperature Sensor Flame Sensor MQ6 Sensor
FR-7	Alerting System	Buzzer LED lights LCD display

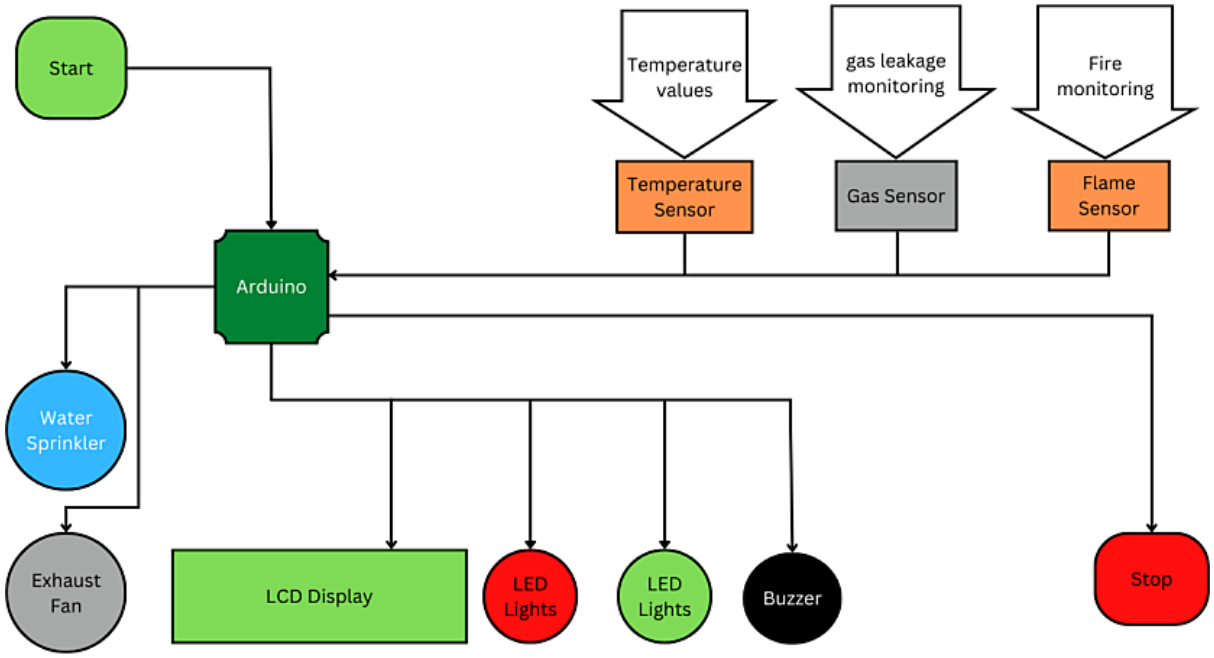
FR-8	After Detection	Water Sprinkler Exhaust Fan

2. **Non-Functional requirements**

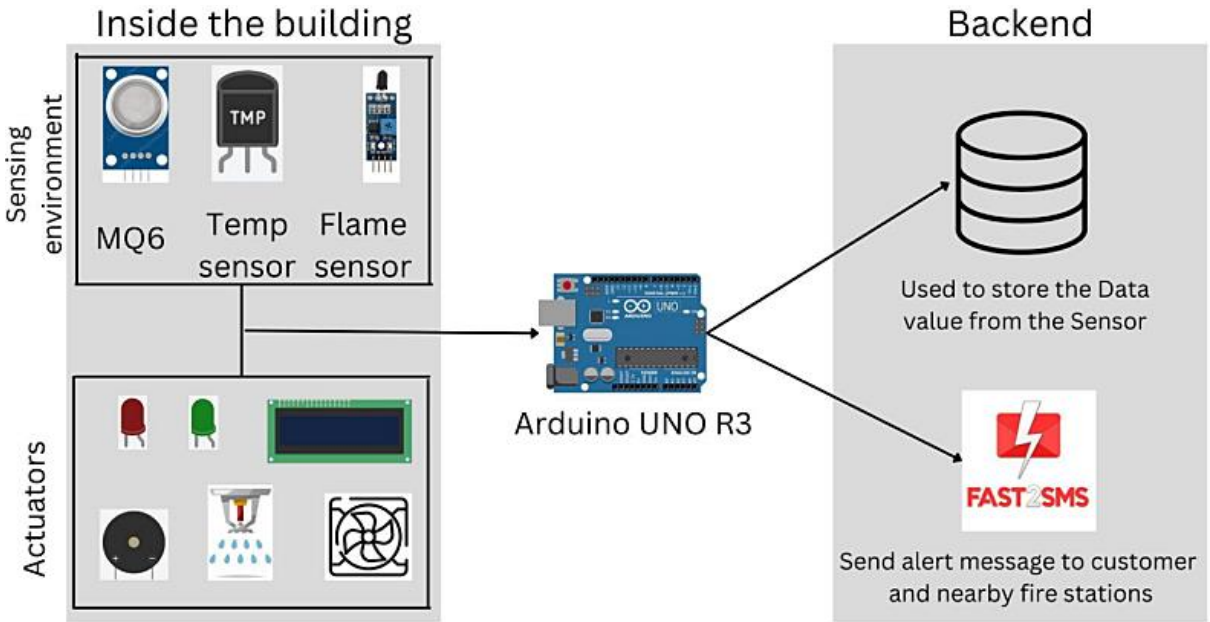
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Have accessible, readily comprehensible and multilingual manual which helps to use the producteasily and effectively.
NFR-2	**Security**	Only the System Administrator should be able to access, modify or delete information related, and grant access to or revoke from authorisedpersons.
NFR-3	**Reliability**	Maintenance should be done once every two weeks. Repair and update the product if required.
NFR-4	**Performance**	The product should be able to constantly monitor, detect fire and gas leakage accidents and alert concerned individuals to reduce/minimize loss of life and property.
NFR-5	**Availability**	Available anytime. Free installation. Will repair anytime in case of malfunctioning.
NFR-6	**Scalability**	Increasing memory, equipping additional servers and Customizing the product if necessary.

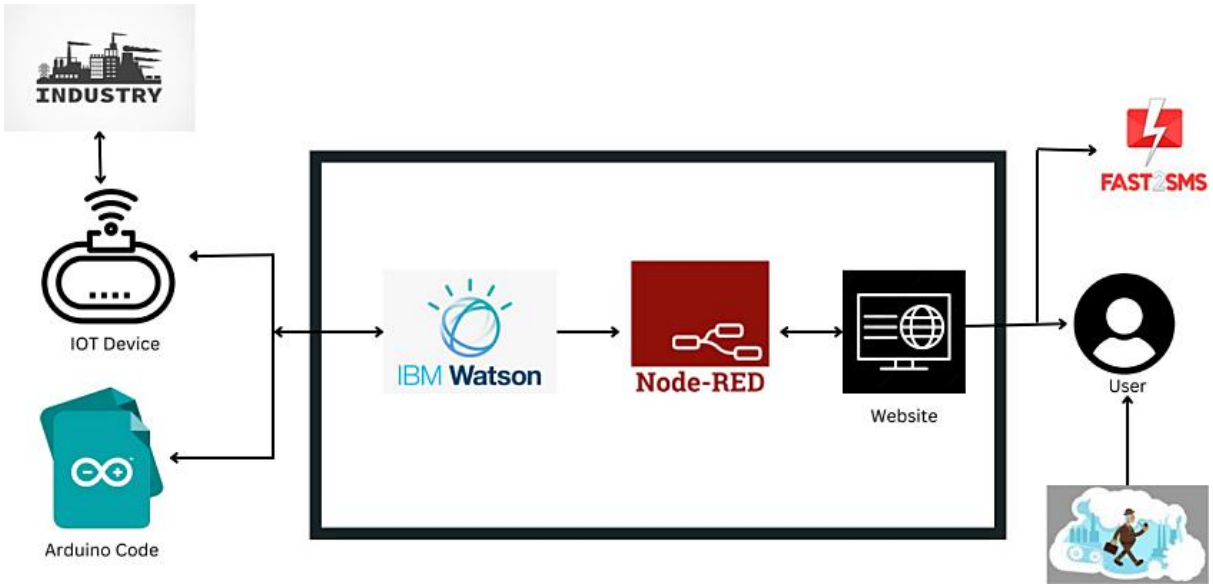
5. PROJECT DESIGN

1. **Data Flow Diagrams**



2. **Solution & Technical Architecture**





3. **User Stories**

User Type	Functional Requirement (Epic)	User StoryNumber	User Story / Task	Acceptance criteria	Priority	Release
Technician	Installation	USN-1	As a technician, I can install the product effectively.	I can assess the working of the product.	High	Sprint-1
Customer (Data Types)	Data Viewing	USN-1	As a user, I can view the temperature readings.	Data from the hardware.	High	Sprint-1
		USN-2	As a user, I can detect the gas leakage and alert the workers.	Data from the hardware.	High	Sprint-1
		USN-3	As a user, I can detect fire and alert the workers.	Data from the hardware.	High	Sprint-1
Administrator	Data Access and modification	USN-1	As an administrator, I can access any information related to the product and modify the same.	Authorised persons only.	High	Sprint-1
	Storage	USN-2	As an administrator, I can store the sensor values in the database and view the same.	Authorised persons only.	High	Sprint-2

6. PROJECT PLANNING & SCHEDULING

1. Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Hardware or Software Simulation	USN-1	Making Hardware device or using Wokwi to connect temperature, flame,gas sensor to Nano with Arduino Uno.	2	High	Pranesh Boopathy, Rupesh S, Praveen kumar K, Rajagopal Narayanan
Sprint-1	Cloud Software	USN-2	Create Device in the IBM Watson IOTPlatform and link it to Node-Red.	2	High	Pranesh Boopathy, RupeshS, Praveenkumar K, Rajagopal Narayanan
Sprint-2	Web page	USN-3	Develop a webpage or Web UI.	2	High	Pranesh Boopathy, RupeshS, Praveenkumar K, Rajagopal Narayanan
Sprint-1	Linking	USN-4	Link Device, IBM cloud and thedeveloped webpage.	2	High	Pranesh Boopathy, Rupesh S, Praveen kumar K, Rajagopal Narayanan
Sprint-1	Dashboard	USN-5	Design the modules and test the device.	2	High	Pranesh Boopathy, RupeshS, Praveenkumar K, Rajagopal Narayanan

2. **Sprint Delivery Schedule**

Sprint	Total StoryPoints	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

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

1. **Feature 1**

Flame Sensor - This sensor is used to monitor or detect industrial fires.

```
// lowest and highest sensor readings:
const int sensorMin = 0;     // sensor minimum
const int sensorMax = 1024;  // sensor maximum

void setup() {
  // initialize serial communication @ 9600 baud:
  Serial.begin(9600);
}
void loop() {
  // read the sensor on analog A0:
      int sensorReading = analogRead(A0);
  // map the sensor range (four options):
  // ex: 'long int map(long int, long int, long int, long int, long
int)'
      int range = map(sensorReading, sensorMin, sensorMax, 0, 3);

  // range value:
  switch (range) {
  case 0:    // A fire closer than 1.5 feet away.
    Serial.println("** Close Fire **");
    break;
  case 1:    // A fire between 1-3 feet away.
    Serial.println("** Distant Fire **");
    break;
  case 2:    // No fire detected.
    Serial.println("No Fire");
    break;
  }
  delay(1);  // delay between reads
}
```

2. **Feature 2**

Temperature Sensor - This sensor is also used to monitor fires but with the help of room temperature readings.

```
float temp;
int tempPin = 0;

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

void loop() {
  temp = analogRead(tempPin);
  // read analog volt from sensor and save to variable temp
  temp = temp * 0.48828125;
  // convert the analog volt to its temperature equivalent
  Serial.print("TEMPERATURE = ");
  Serial.print(temp); // display temperature value
  Serial.print("*C");
  Serial.println();
  delay(1000); // update sensor reading each one second
}
```

3. **Feature 3**

MQ6 sensor (Gas sensor) - This sensor is used to detect gas leakages in the industry. MQ6 sensor can detect various gases like LPG, Methane, Propane, etc.

```
int LED = 12;
        int BUZZER = 13;
        int LPG_sensor = 3;// MQ-6 SENSOR
        int LPG_detected;
        void setup()
        {
          Serial.begin(9600);
          pinMode(LED, OUTPUT);
          pinMode(BUZZER, OUTPUT);
          pinMode(LPG_sensor, INPUT);
        }
        void loop()
        {
           LPG_detected = digitalRead(LPG_sensor);
           Serial.println(LPG_detected);
          if (LPG_detected == 1)
          {
            Serial.println("LPG detected...");
            digitalWrite(LED, HIGH);
            digitalWrite(BUZZER, HIGH);
          }
          else
          {
            Serial.println("No LPG detected.");
            digitalWrite(LED, LOW);
            digitalWrite(BUZZER, LOW);
```

```
    }
}
```

8. # TESTING
    1. **Test Cases**
        1. Verify the working of the flame sensor
        2. Verify the working of the temperature sensor
        3. Verify the working of the gas sensor(MQ6)
        4. Verify whether the users receive notifications in case of accidents.
        5. Verify whether the fire stations receive notifications in case of accidents.
        6. Verify the working of the buzzer installed.
    2. **User Acceptance Testing**

Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were 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	5	2	1	8
Totals	24	14	13	26	77

Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	6	0	0	6
Client Application	51	0	0	51
Security	2	0	0	2

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

1. **Performance Metrics**

   Hours worked - 36 hours(Approx)
   Quality of the product - 90%
   Efficiency of the product - 95%

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

1. Usage of multiple sensors to detect gas leakage and fire accidents simuntaneously.
2. Sprinklers and exhaust fans are used to extinguish fires.

DISADVANTAGES:

1. The main drawback with conventional panels is that one cannot tell which device has been activated within a circuit. The fire may be in one small room, but as far as emergency responders can tell, a fire could exist anywhere within a zone.

11. CONCLUSION

This system can detect both fire and gas spillage with savvy solutions and smart notification. This system provides both security and safety compared to other existing system. This system sends notification to nearby fire stations in case of any fire accidents in the industries.

12. FUTURE SCOPE

The future will be with multicriteria detection in which the detector will be more of a sensor, with the detection more for the products of combustion, such as carbon monoxide, carbon dioxide, sulfur dioxide, nitrogen oxides in addition to heat and particulate matter.

13. APPENDIX

Source Code

```
// Change These Credentials with your Blynk Template credentials
// Change These Credentials with your Blynk Template credentials
#define BLYNK_TEMPLATE_ID "TMPLqCSC89Q2"
#define BLYNK_DEVICE_NAME "Fire Detection"
#define BLYNK_AUTH_TOKEN "PxJ7MvV-hMXaEwKe39Lip9vLqZRNSCOX"

#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include<OneWire.h>
#include<DallasTemperature.h>
#include <BlynkSimpleEsp8266.h>

char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "praveen";
// Change your Wifi/ Hots
pot Name char pass[] = "2
4092001"; // Change your
Wifi/ Hotspot Password
```

```
BlynkTimer timer;

#define fire D2 #define smoke A0
#define ONE_WIRE_BUS D4
#define GREEN D5 #define RED D6

#define buzzer D7int fire_Val = 0;
int data = 0;
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature DS18B20(&oneWire);
float temp = 0;
WidgetLED led(V1);

void setup() //Setup function - only function that is run in deep sleep mode
{
  Serial.begin(9600);
  //Start the serial
  output at 9600
  baudpinMode(GREEN, O
  UTPUT);
  pinMode(smoke,INPUT);
  pinMode(buzzer,OUTPUT);
  pinMode(fire, INPUT);
   pinMode(RED, OUTPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(ONE_WIRE_BUS, INPUT);

  Blynk.begin(auth, ssid, pass);//Splash screen delay
  delay(2000);
  timer.setInterval(500L, mySensor);
}

void loop() //Loop function
{
  Blynk.run();
  timer.run();
}

void mySensor()
{
fire_Val = digitalRead(fire);
data = analogRead(smoke);
  Blynk.virtualWrite(V2,data);
  DS18B20.requestTemperatures();

  temp = DS18B20.getTempCByIndex(0);
  Blynk.virtualWrite(V3,temp);
  if ((fire_Val == HIGH)||(data > 500)||(temp > 35))
  {
    Blynk.logEvent("fire_alert");
```

```
    digitalWrite(GREEN, LOW);
    digitalWrite(RED, HIGH);
    tone(buzzer, 1000);
    Blynk.virtualWrite(V0, 1);
    Serial.print("fIRE Level: ");
    Serial.println(fire_Val);
    Serial.write("fire detected");led.on();
  }

  else
  {
    digitalWrite(GREEN, HIGH);
    digitalWrite(RED, LOW);
    noTone(buzzer);
    Blynk.virtualWrite(V0, 0);
    Serial.print("fIRE Level: ");
    Serial.println(fire_Val);
    led.off();
    Serial.write("no fire detected");

    Serial.println(data); Serial.println(temp);
  }
}
```

GitHub Link
https://github.com/IBM-EPBL/IBM-Project-17285-1659633419
Project Demo Link
https://drive.google.com/file/d/1OMDsJQLua2bI1sp9cdJGBqJQ-LScNs2e/view?usp=drivesdk