GOVERNMENT COLLEGE OF TECHNOLOGY-COIMBATORE

INDUSTRY -SPECIFIC INTELLIGENT FIRE MANAGEMENT SYSTEM

DOMAIN-Internet of things

IBM-NALAIYATHIRAN

PROJECT REPORT

TEAM MEMBERS

1.ABIRAMI B

2.ARUNA S

3.VANITHA E S

4. SHOWMINI R

TEAM ID: PNT2022TMID06972

MENTOR:DEVI R

1.INTRODUCTION

Nowadays, fire incidents have become a critical issue, which must be dealt with on time without any unnecessary delay to avoid the loss in lives and belongings. According to the National Fire Protection Association (NFPA), two-third of U.S. household fires occur in premises with no working smoke alarms, alarms with no proper maintenance, or misplaced alarms. The appropriate allocation of fire alarms with a proactive warning could save lives and reduce property losses. The concept of Internet of things (IoT) nowadays is applied in many applications ranging from the smart industry, smart agriculture to smart healthcare , and smart home application .Nowadays, fires can get out of control because people intend to save money rather than installing proper fire alarm systems. Some problems are still on, such as affordability,effectiveness,and responsiveness . Many studies have been conducted to address these issues like ; however, fire detection issues are not addressed properly since these systems rely on machine vision, where the algorithms need more images to train, and the detection rate is not satisfactory Thus, this paper aims to minimize false alarms, provide faster response, and a new IoT approach that used Node-Red. The contribution is as follows:

(1) To determine which combinations and algorithms of sensors can accurately and quickly detect fires,

(2) We have designed and then developed a system that detects fire and activates the fire alarm,

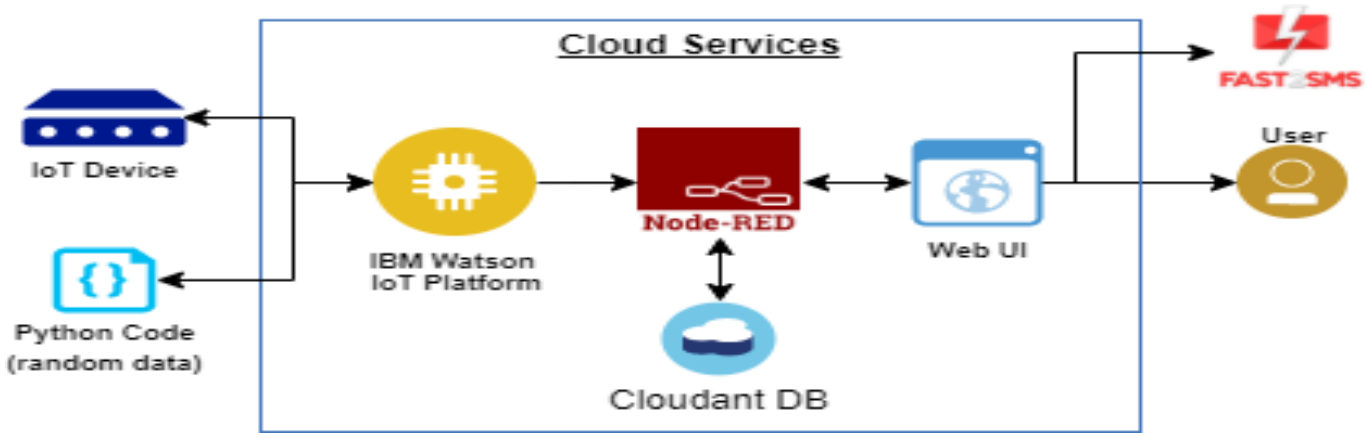
(3) the proposed system evaluates the situation and initiates an automatic water sprinkler where the water unit was designed separately, and

(4) the system analyses the collected data using IBM Watson platform which results in a faster response. Thus, the highlighted four points make the proposed system superior in terms of affordability, effectiveness, and responsiveness.

1.1 PROJECT OVERVIEW

Smart Fire Detection System with Automatic Water sprinkler has been developed to solve the slow response issue of fire accidents. The inputs provide readings for the system to analyze , such as sensors and Wi-Fi module that works as a

transmitter for the sensor readings. Temperature, gas, and flame sensors are inputs. The readings from the inputs are displayed on the web page. Updated readings are sent into a Wi-fi module that translates the data into a graphical and statistical manner. A web page created to analyze the data and a response extracted conditionally to launch a water sprinkler .If the sensor readings is greater than the threshold value , fast SMS notification will send to the user.



1.2 PURPOSE

- To implement a smart fire detection system that would not only detect the fire using integrated sensors but also alert property owners, emergency services and local police station to protect lives and their valuable asserts.
- It employs different integrated detectors such as heat, smoke and flame sensors.
- The IoT based sensors are smarter than conventional smoke detectors.
- Early detection will help to save lives and industry before the fire takes over the buildings.

2 LITERATURE SURVEY

In recent times, wireless sensor, wireless communications, wireless control and mobile digital technology became more and more prominent in our daily lives [1]. The wireless communication technology is mostly used in automatic centralized control of building. This paper is focused on a software build with the wireless technology to be handy and applicable for one safety mechanism. The safety mechanism that we are focussing on is fire detection. The designed model shall monitor the fire alarm to long distances thus ultimately helping in evacuation procedure as well. Modern fire safety mainly focuses on fire alarm. However, the control centre can't take proper evacuation and response in time [1]. The existing system has disadvantages that it cannot connect to all existing devices in that particular area where fire has taken place and also that the monitor is not connected to a central main server.

Therefore, this design is for transmitting the fire information to long distances within the building by using Zig-Bee Wi-Fi network and the detected signals are sent to monitoring centre by Wi-Fi network, which connects with personal terminal easily. A number of existing models were studied and their effectiveness was compared.

• Ahmed Imteaj et.al.

He Studied the problems faced by factory workers in times when fire breaks out. They proposed a system using Raspberry Pi 3 which is capable of detecting fire and providing information about area of fire. The Raspberry Pi controls multiple Arduino boards which are connected with several motors and cameras to capture the fire incident . In this, they discussed about the modern technology that can be used to reduce extremely unfortunate accidents caused by fire. We designed the whole system and calculated its effectiveness.

• Ondrej Krejcar

He proposed a model for location enhancement and personnel tracking using Wi-Fi networks. In this, he has represented the control system concept that is used in handling information of location and control unit operations. The location of the user present in the building, is obtained through WiFi access points . We have studied this to understand the usability of the Wi-Fi networks in live tracking and then have utilized this functionality to track fire and give information about location of fire to various devices intimating people about the mishap.

• Karwan Muheden

He have studied the safety features in home and industrial areas. They have designed new model using WSN. Not only have they incorporated temperature and humidity sensors but also included fire and smoke sensors while developing the model. They present a preceding study of WSN is able to detect fire alarm. It is for setting up a wireless sensor network with three sensors. An application was developed for getting home information.

• Azka Ihsan Nurrahman, Kusprasapta Mutijarsa

They have proposed a prototype for a centralized management system for homes or offices which helps better in managing the safety features. In this, home management system is required. This system controls the room lights by turning on and off automatically, it keeps the record of use of electronic device status, turning on and off the ac regulator automatically, it displays the room temperature in home. If fire is detected in the house, it turn on sprinkler at home, it supervises at home via surveillance cameras, take photos and store them including recordings of surveillance at home, it detects the movements of people at home, and provide notification when someone enters the house.

2.1 EXISTING PROBLEM

After doing the literature survey we have listed some of the features that are existing in the now used fire alarm systems. The features of the existing system are as under.

- Identify status periodically - The system checks for a fire at particular intervals and not continuously or not in real time. This is a drawback as there will possibly be a time lag between the actual fire incident and when the fire will be reported due to periodic identification.
- Manual operation for transferring information Automatic operation is not facilitated in the pre sent systems.
- Not able to find the pressure point of the building which are likely to catch fire easily.
- Difficult to sense structural damage

2.2 REFERENCES

[1] Liu Yunhong, Qi Meini, "The Design of Building Fire Monitoring System Based on ZigBee-WiFi Networks" , Eighth International Conference on Measuring Technology and Mechatronics Automation, IEEE, 2016, pp-733-735

[2] Ahmed Imteaj, Tanveer Rahman, Muhammad Kamrul Hossain, Mohammed Shamsul Alam, Saad Ahmad Rahat, "An IoT based fire alarming and authentication system for workhouse using Raspberry Pi 3" , International Conference on Electrical, Computer and Communication Engineering (ECCE), IEEE, 2017

[3] Ondrej Krejcar, "Using of mobile device localization for several types of applications in intelligent crisis management",5th IEEE GCC Conference & Exhibition, IEEE, 2009

[4] Karwan Muheden, Ebubekir Erdem, Sercan Vançin, "Design and implementation of the mobile fire alarm system using wireless sensor networks", 17th International Symposium on Computational Intelligence and Informatics (CINTI), IEEE, 2016

[5] Azka Ihsan Nurrahman, Kusprasapta Mutijarsa, "Intelligent home management system prototype design and development", International Conference on Information Technology Systems and Innovation (ICITSI), IEEE, 2015.

2.3 PROBLEM STATEMENT DEFINITION

Industrial fires and explosions cost companies and government billions of rupees every year apart from the loss of lives, which cannot be described in monetary terms. These fires not only results in huge loss of lives and property but also disrupt production in the industry. Singular sensors were used for a long time in the event of detection of a fire, but these sensors can not measure the amount of fire to alert the emergency response units.

3.IDEATION AND PROPOSED SOLUTION

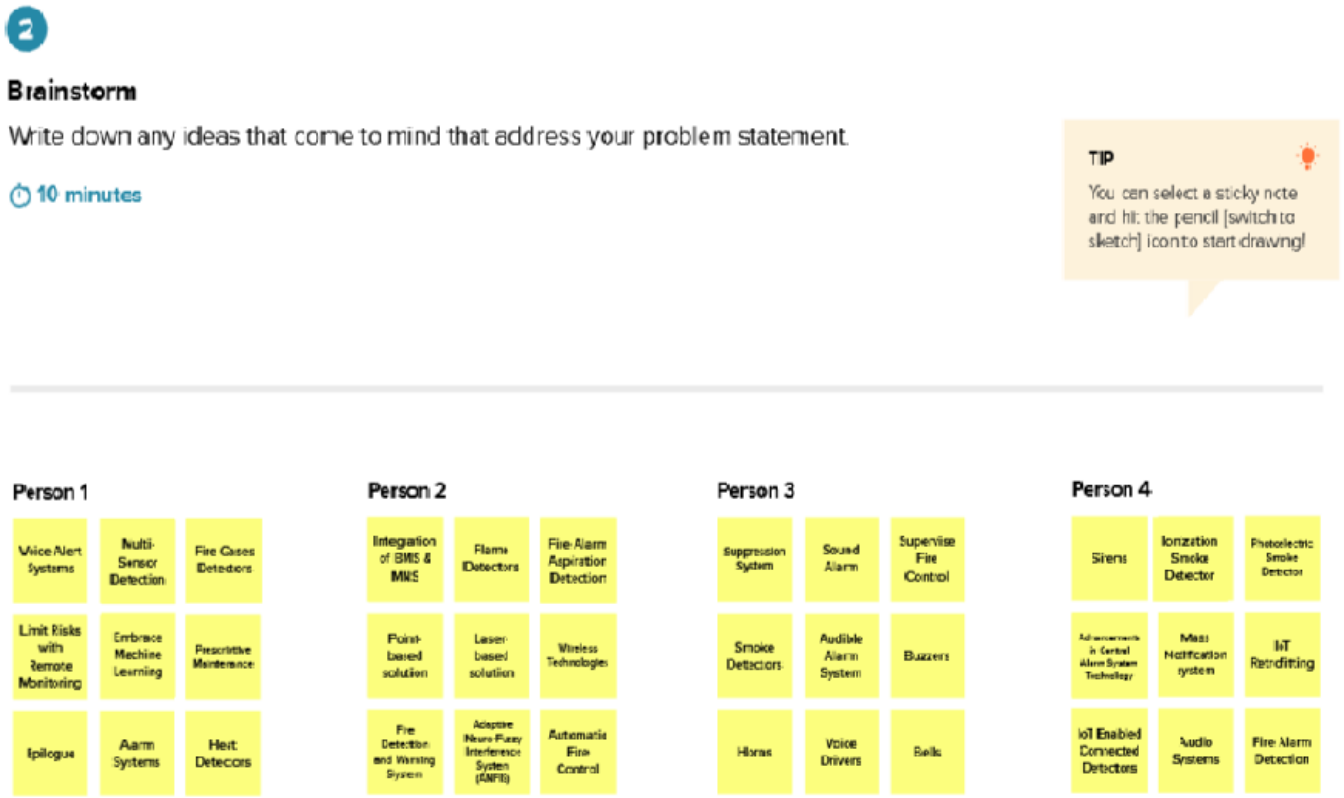
Ideation is expressed via graphical, written, or verbal methods , and arises from the past or present knowledge , influences , opinions, experiences and personal convictions.

3.1 EMPATHY MAP CANVAS

It serves as a foundation for understanding user experiences , which focuses on providing the experience customer want rather than forcing design teams to relay on guess work.

3.2 IDEATION AND BRAINSTORMING

Ideation is often closely related to the practice of brainstorming, a specific technique that is utilized to generate new ideas.

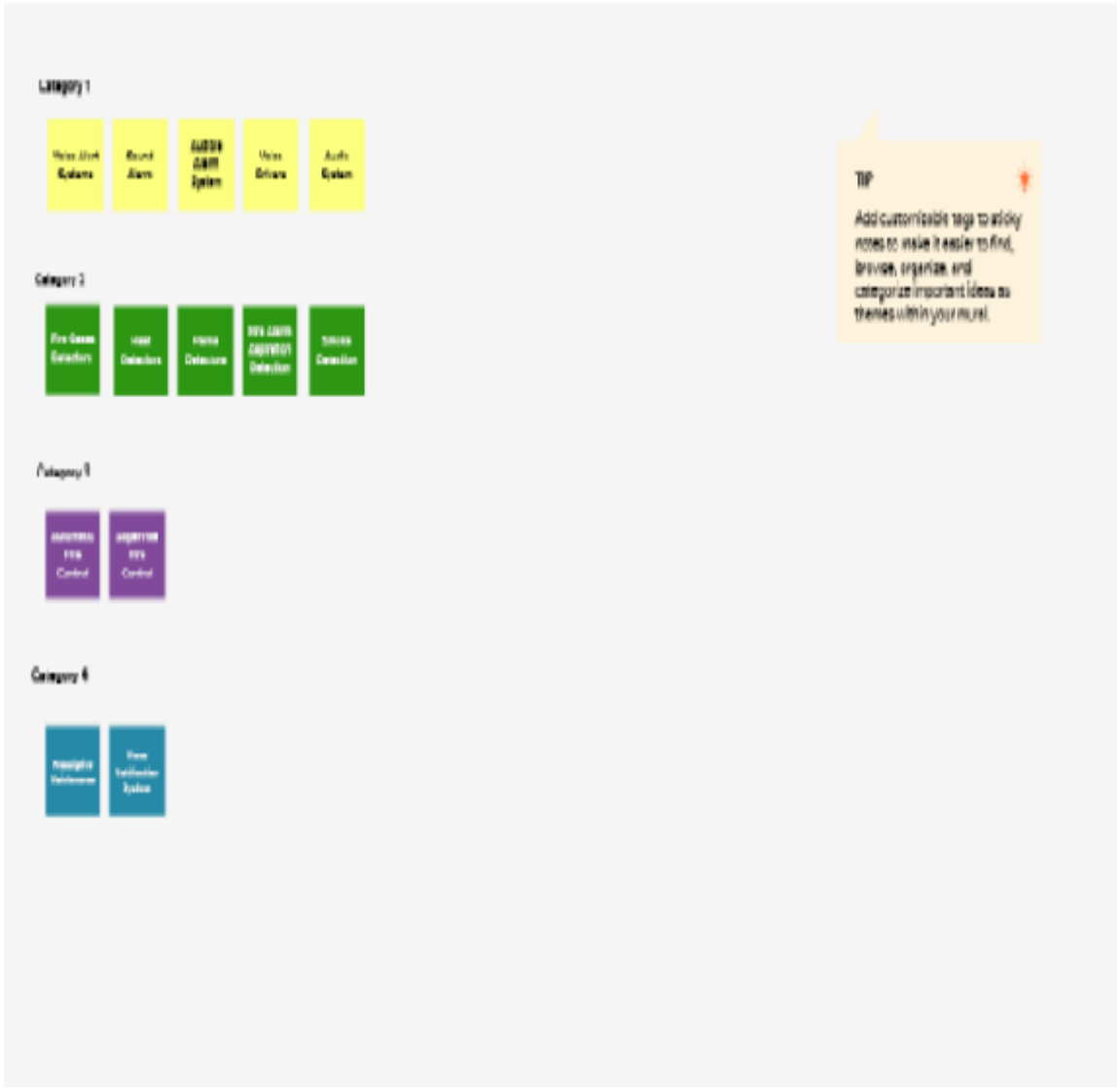


2

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

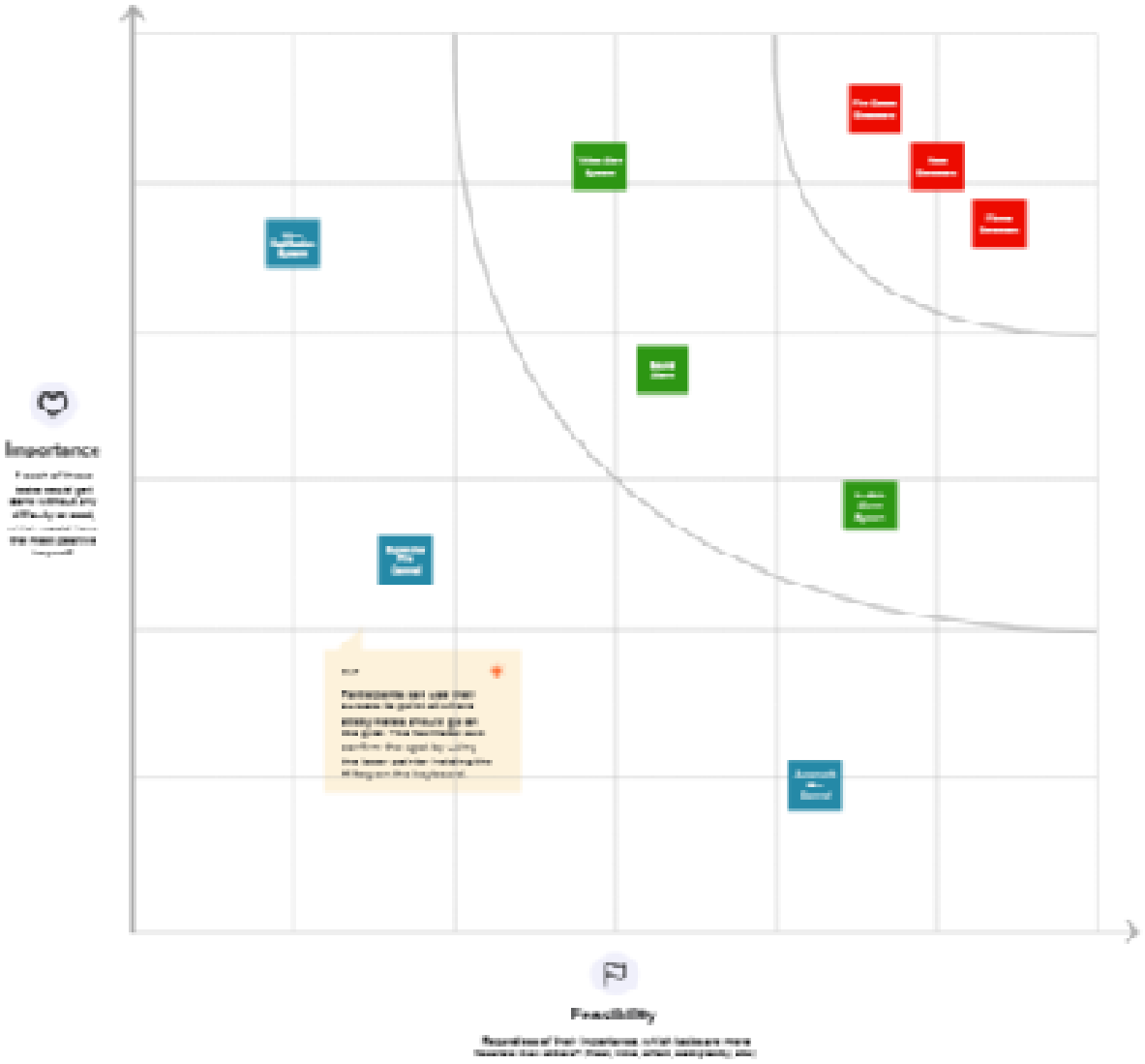


4

Prioritise

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and

20 minutes



3.3 PROPOSED SOLUTION

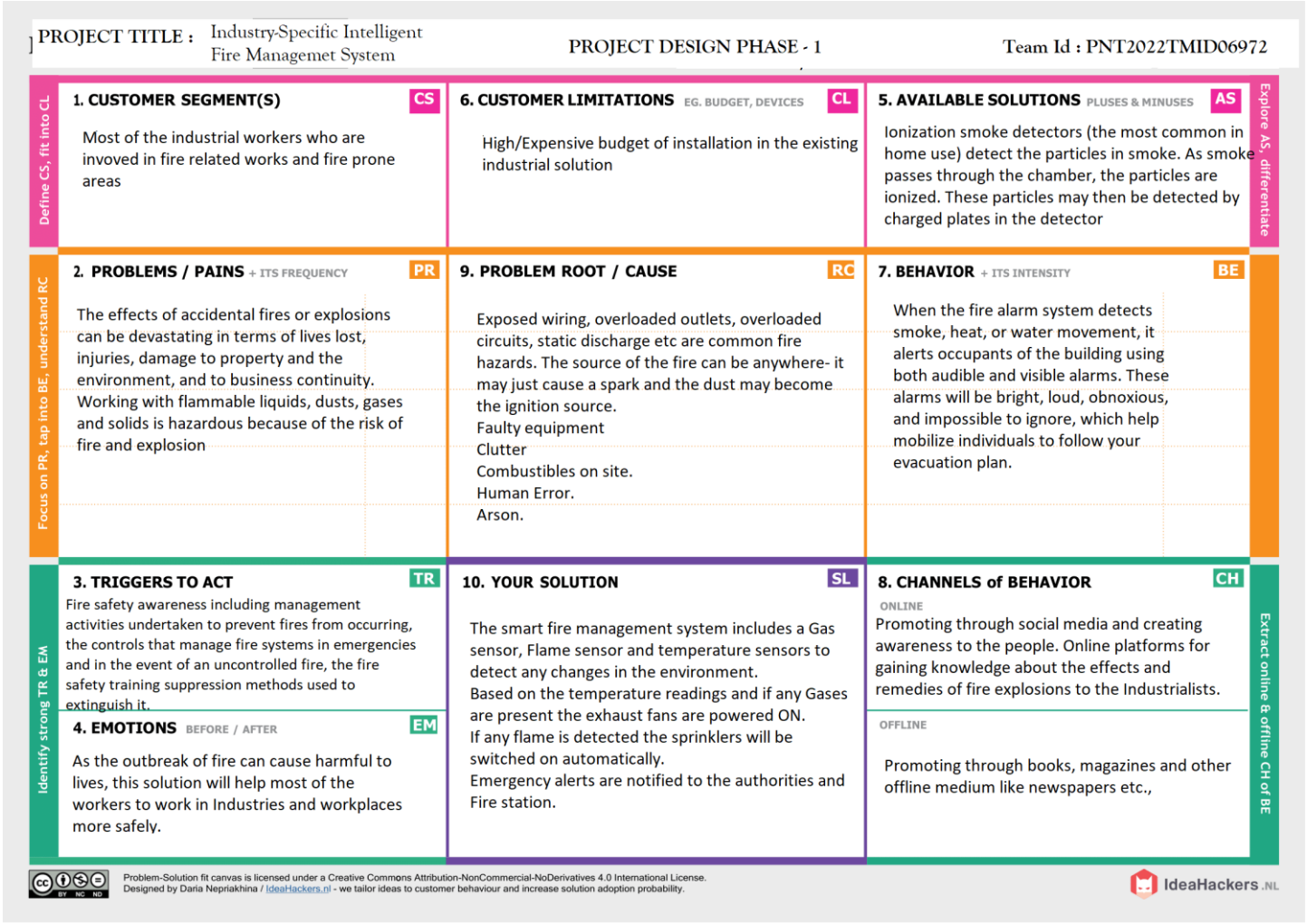
It should relate the current situation to a desired result and describe the benefits that will accrue when the desired result is achieved.

S.NO	PARAMETER	DESCRIPTION
1	Problem Statement (Problem to be solved)	John is an Industrialist who works in the cotton industry. He often fears that fire at any place in the industry might collapse the whole industry. He has a facility for extinguishing the fire as soon as he identifies the fire in the industry, but the problem is he cannot sense the fire instantly. Hence he needs an industry-specific fire Management system in his industry.
2	Idea / Solution description	-This fire detection designed by them will provide more improvement in early detection and quick response as well. -Our system includes two sides of subsystem the first one is the transmission subsystem and the second one is the reception (Rx) subsystem and there is a processing unit for each subsystem. -At the subsystem, the available sensors located their will gather the real-time information allocated for physical quantity to transmit them to the processing unit. - the fire is broke out there is a big prospect that the electricity will be catted off and here comes the benefit of the light that will going to turn on automatically.

3	Novelty / Uniqueness	Reduced alarm response time Flexibility Intelligence – devices communicate with the control panel and each other
4	Social Impact / Customer Satisfaction	Threat activities and warnings Fire, explosion or chemical hazards Suspicious/dangerous person alerts Severe weather notifications Power, operations or network disruptions
5	Business Model (Revenue Model)	By using this system, the user can detect the fire instantly and able to control it.
6	Scalability of the Solution	By implementing this system, the people can efficiently and effectively gain knowledge about the fire accident. This system can also be integrated with the android application where the users can be notified real-time.

3.4 PROBLEM SOLUTION FIT

It means we have to found a problem with our customer and that the solution we have realized for it actually solves our customers problem.



4.REQUIREMENT ANALYSIS

Solution requirements describe specific characteristics that a product must have to meet the needs of the stakeholders and the business itself. They fall into two large groups.

- Functional requirements define what a product must do, what its features and functions are.
- Non-functional requirements describe the general properties of a system. They are also known as quality attributes.

4.1 FUNCTIONAL REQUIREMENT

Following are the functional requirements of the proposed solution.

S.NO	FUNCTIONAL REQUIREMENT(EPIC)	SUB REQUIREMENT(STORY/SUB TASK)
1	User Requirements	Workers and Product Protection Automatic Sprinkler System Monitors Smoke ,Gas and Temperature
2	User registration	Manual Registration Registration through webpage Registration through Form Registration through Gmail
3	User Confirmation	Confirmation via Phone Confirmation via Email Confirmation via OTP
4	Payment Options	Options Cash on Delivery Net Banking/UPI Credit/Debit/ATM Card
5	Product Delivery and Installation	Door Step delivery Take away Free Installation and 1 year Warranty
6	Product Feedback	Through Webpage Through Phone calls Through Google forms

4.2 NON-FUNCTIONAL REQUIREMENTS

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

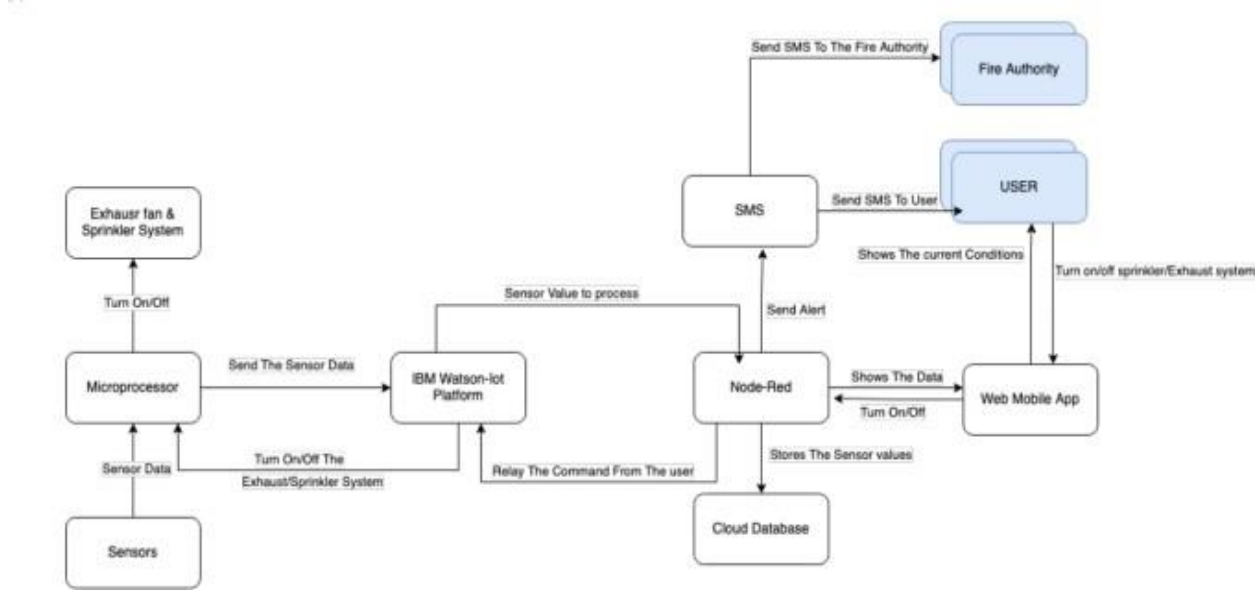
S.NO	NON-FUNCTIONAL REQUIREMENT	DESCRIPTION
1	Usability	Have a clear and self-explanatory manual. Easier to use. Easily accessible by everyone.
2	Security	Security Are inspected monthly by the Fire Alarm Technician. Inspected and tagged by a contractor annually.
3	Reliability	Hardware requires a regular checking and service. Software may be updated periodically. Immediate alert is provided in case of any system failure.
4	Performance	The equipment must have a good user interface It should have a minimal energy requirement It has to save lives of people and things
5	Availability	Availability All the features will be available when the user requires. It depends on the need of the user and the customization of the user has done.
6	Scalability	The product has to cover all the space of industry irrespective of the size or area.

5. PROJECT DESIGN

Project design is an early phase of the project life cycle where ideas, processes,resources and deliverables are planned out.

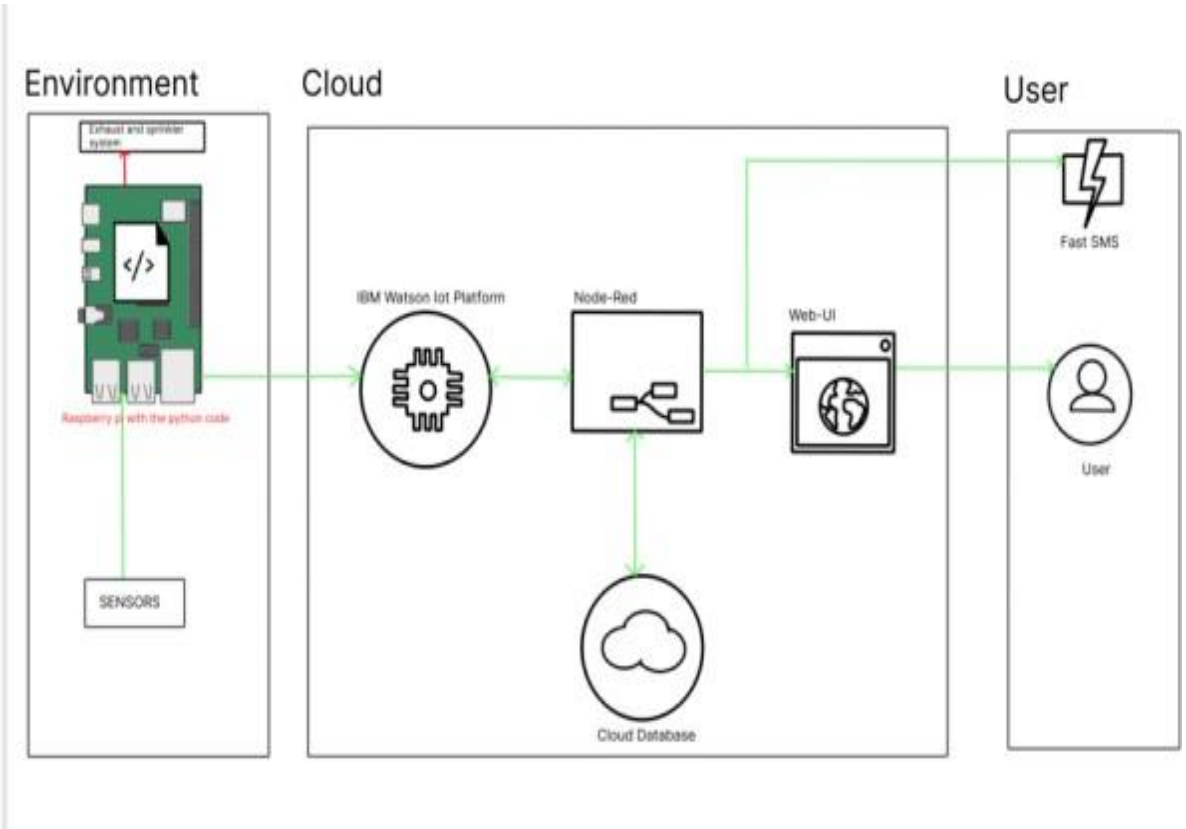
5.1 DATA FLOW DIAGRAMS

It is a graphical or visual representation using a standardized set of symbols and notations to describe a business's operation through data movement.



5.2 SOLUTION AND TECHNICAL ARCHITECTURE

Technical architecture is a form of IT architecture that is used to design computer systems



5.3 USER STORIES

User story is an informal , general explanation of a sftware feature return from the perspective of the end user/customer.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Monitor The Environment	USN-1	User can monitor the sensor data receiving from the microprocessor	User Can See the dashboard with sensor information	Medium	Sprint 4
	Turn on or off the sprinkler and exhaust fan.	USN-2	User can turn on / off exhaust fan and sprinkler if need in that circumstance	Can turn on / off the sprinkler and exhaust fan	Medium	Sprint 4
	Authentication	USN-2	User needed to be authenticated while turning on/off the exhaust and sprinkler system	Authenticate the user for USN-2 Fuctionality	Medium	Sprint 4
Sensing	Sensing The Environment	-USN 3	Need to Sense the environment using the sensors attached to the microprocessor	Getting Data from the sensors	High	Sprint 1
Extinguish	Actuators	USN 4	If the sensors sense the fire then the immediate next step is to turn on the exhaust fan and the sprinkler system	Extinguishing the fire	High	Sprint 1
Data	Sending data to ibm Watson Hot platform	USN 5	All the sensor Data received from the microprocessor are send to the IBM Watson Lot platform	Showing in the Watson Dashboard	Medium	Sprint 2
	Node-red	USN 6	Sending the data to further process in the cloud for storing and alert purpose		High	Sprint 3
	Data Storing	USN 7	All the sensor values are stored in an cloud database	Storing the data	Low	Sprint 3
Notification	Event notification	USN 8	FIre alertMessage will send to fire department	Notifying the authorities	High	Sprint 4

6. PROJECT DESIGN AND PLANNING

6.1SPRINT PLANNING AND ESTIMATION

Sprint	Functional requirement	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	ABIRAMI B ARUNA S
Sprint- 1	Simulation	USN-2	Connect sensors and Arduino with python.	1	HIGH	VANITHA E S SHOWMINI R
Sprint -2	Software	USN-3	Creating device in the IBM Watson IoT platform and workflow using Node-red.	2	LOW	ABIRAMI B ARUNA S
Sprint -1	MIT App	USN-4	Develop a mobile application for the Fire Management System using MIT app inventor.	2	MEDIUM	VANITHA E S SHOWMINI R
Sprint- 1	Login	USN-5	As a user, I can log into the application by entering email & password	1	HIGH	ABIRAMI B ARUNA S
Sprint- 1	Dashboard	USN-6	As a user ,I can get notification alert	1	MEDIUM	ABIRAMI B ARUNA S
Sprint- 3	Testing and development phase 1	USN-7	Testing the system performance , For an emergency case and its deployed	2	HIGH	VANITHA E S SHOWMINI R
Sprint- 3	Testing and development phase 1	USN-8	Sending an alert SMS to the fire authority in case of fire.	2	HIGH	ABIRAMI B ARUNA S
Sprint- 4	Implementation	USN-9	Deployment of IOT based industrial specific fire management system . I can see and use the system for 24/7.	2	HIGH	VANITHA E S SHOWMINI R

6.2 SPRINT DELIVERY SCHEDULE

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

7. CODING AND SOLUTIONING

7.1 FEATURE CODE 1: This set of code checks for false alarm

```
  //find the accident status 'cause fake alert may be caused by some mischief activities

if(temp < 45 ){
  if(flame > 650){
    accidentstatus = "Need Auditing";
    isfanon = true;
    issprinkon = false;
  }
  else if(flame <= 10){
    accidentstatus = "nothing happened";
    isfanon = false;
    issprinkon = false;
  }

}else if(temp >= 45 && temp <= 55 ){
  if(flame <=650 && flame >100  ){
    issprinkon = true;
    accidentstatus = "moderate";
    if(gas > 150){
      isfanon = true;
    }
    else{
      isfanon = false;
    }
  }
  else if(flame <= 100 && flame > 10){
    issprinkon = true;
    isfanon = false;
    accidentstatus = "moderate";
  }

}else if(temp > 55){
```

```
    if(flame > 650){
      gas = 500 + rand()%500;


      accidentstatus = "severe";
      issprinkon  =  true;
      isfanon = true;
    }
    else if(flame < 650 && flame > 400 ){
      gas = 300 + rand()%500;
      accidentstatus = "severe";
      issprinkon = true;
      isfanon = true;
    }
  }
  else {
    accidentstatus = "Need Auditing";
    isfanon = false;
    issprinkon = false;
  }


  if(issprinkon){
    if(flow){
      sprinkstatus = "working";
    }
    else{
      sprinkstatus = "not working";
    }
  }
  else if(!issprinkon){
    sprinkstatus = "ready";
  }
  else {
    sprinkstatus = "something's wrong";
  }
```

7.2 FEATURE CODE 2: It sends the data to IBM IoT Watson platform

```
     void PublishData(float temp, int gas ,int flame ,int flow,bool isfanon,bool
issprinkon) {

  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String payload = "{\"temp\":";
  payload += temp;
  payload += "," "\"gas\":";
  payload += gas;
  payload += "," "\"flame\":";
```

```
  payload += flame;
  payload += "," "\"flow\":";
  payload += ((flow)?"true":"false");
  payload += "," "\"isfanon\":";
  payload += ((isfanon)?"true":"false");
  payload += "," "\"issprinkon\":";
  payload += ((issprinkon)?"true":"false");
  payload += "," "\"accidentstatus\":";
  payload += "\""+accidentstatus+"\"";
  payload += "," "\"sprinkstatus\":";
  payload += "\""+sprinkstatus+"\"";
  payload += "}";


  Serial.print("Sending payload: ");
  Serial.println(payload);


  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it sucessfully upload data on the cloud then it
will print publish ok in Serial monitor or else it will print publish failed
  } else {
    Serial.println("Publish failed");
  }

}
```

8.TESTING

8.1 TEST CASES

SL.NO	INPUT	OUTPUT	RESULT
01.	Gas:9 Temperature:24 Flame:109	Exhaust fan on:TRUE Sprinklers:ON	Passed
02.	Gas:124 Temperature:24 Flame:693	Exhaust fan on:TRUE Sprinklers:OFF	Passed
03.	Gas:238 Temperature:24 Flame:155	Exhaust fan on:TRUE Sprinklers:ON	Passed
04.	Gas:305 Temperature:24 Flame:479	Exhaust fan on:FALSE Sprinklers:OFF	Passed
05.	Gas:20 Temperature:24 Flame:531	Exhaust fan on:TRUE Sprinklers:OFF	Passed

SL.NO	INPUT	OUTPUT	RESULT
01.	Gas:46 Temperature:21.2 Flame:109	Exhaust fan on:TRUE Sprinklers:ON	Passed
02.	Gas:113 Temperature:21.2 Flame:693	Exhaust fan on:TRUE Sprinklers:OFF	Passed
03.	Gas:228 Temperature:21.2 Flame:155	Exhaust fan on:FALSE Sprinklers:ON	Passed
04.	Gas:9 Temperature:21.2 Flame:479	Exhaust fan on:FALSE Sprinklers:OFF	Passed
05.	Gas:124 Temperature:21.2 Flame:531	Exhaust fan on:TRUE Sprinklers:OFF	Passed

SL.NO	INPUT	OUTPUT	RESULT
01.	Gas:46 Temperature:21.2 Flame:109	Exhaust fan on:TRUE Sprinklers:ON	Passed
02.	Gas:113 Temperature:21.2 Flame:693	Exhaust fan on:TRUE Sprinklers:OFF	Passed
03.	Gas:228 Temperature:21.2 Flame:155	Exhaust fan on:FALSE Sprinklers:ON	Passed
04.	Gas:9 Temperature:21.2 Flame:479	Exhaust fan on:FALSE Sprinklers:OFF	Passed
05.	Gas:124 Temperature:21.2 Flame:531	Exhaust fan on:TRUE Sprinklers:OFF	Passed

8.2 USER ACCEPTANCE TESTING (UAT)

DEFECT ANALYSIS

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	9	0	2	1	12
External	0	0	1	0	1
Fixed	19	24	25	14	82
Not Reproduced	0	0	2	0	2
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	28	24	30	15	97

TESTCASE ANALYSIS

Section	Total Cases	Not Tested	Fail	Pass
Client Application	4	0	0	4
Security	2	0	0	2
Exception Reporting	11	0	0	11
Final Report Output	5	0	0	5

9. RESULTS

9.1PERFORMANCE METRICES

CPU USAGE

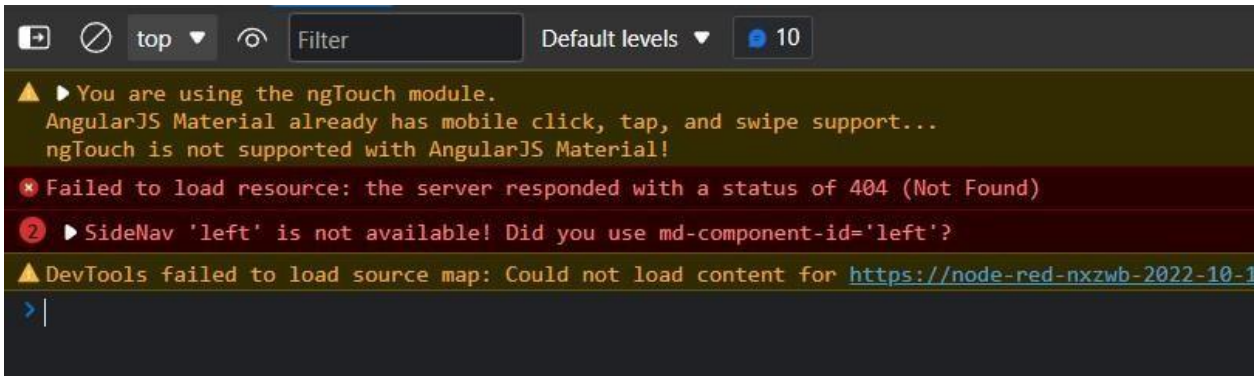
The micro version of c++ is make the best use of the CPU. For every loop the program runs in one time, neglecting the network and communication. The program sleeps for every 1 second for better communication with MQTT. As the program takes O(1) time and the compiler optimizes the program during compilation there is less CPU load for each cycle. The upcoming instructions are on the stack memory, so they can be popped after execution.

MEMORY USAGE

The sensor values , networking data are stored in sram of the ESP32 . It’s a lot of data because ESP32 has only limited amount of memory (520 KB) .For each memory cycle the exact addresses are overwritten with new values to save memory and optimal execution of the program.

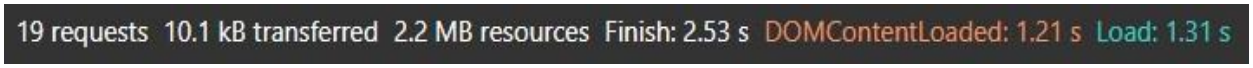
ERROR RATES

The errors rates are very low as the backend and dashboard is handled with node-red.The exceptions are handled in a proper way as it does not affect the usability of the system

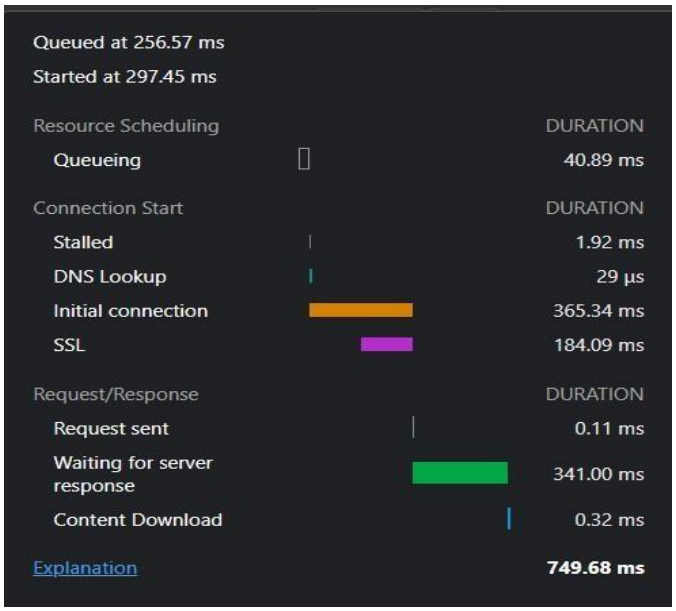


LATENCY AND RESPONSE TIME:

The DOM handling of the received data is optimal and latency is low .After the DOM is loaded the entire site is loaded to the browser



The server also responses quickly . The average time of response is respectable

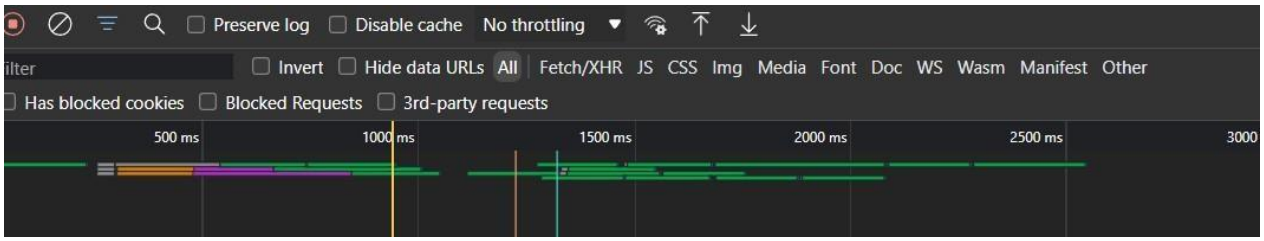


For the data sent from the IoT device (considering the sleep of one second from the IoT), the response is much quicker .We can easily see the delay caused by the sleep function

The average time is well over optimal value

Average time = $(5ms + 2600ms)/2$

$= 1302.5$



10. ADVANTAGES AND DISADVANTAGES:

Advantages

- Active monitoring for gas leakage and fire breakout
- Automatic alerting of admin as well as fire authorities using SMS
- Automatically turning on/off sprinkler as well as exhaust fan
- Authentication is required to turn on/off of sprinkler and exhaust fan as well as sending SMS alert manually
- It automatically detect false fire breakout reducing unnecessary panic
- by using flow sensors we can confirm that the sprinkler system is working as it intended
- All device status can be shown in a dashboard
- Users can see the dashboard using a web application

Disadvantages

- Always need to connect with the internet [Only to Send the SMS alert]
- If the physical device is damaged the entire operation is collapsed
- Need large database since many data is stored in cloud database every second

11. CONCLUSION

So in conclusion our problem premise is solved using IoT devices by creating a smart management system that solves many inherent problems in the traditional fire management system like actively monitoring for fire breakouts as well as gas leakage and sending SMS alerts to the admin as well as to the fire authorities .

12. FUTURE SCOPE

The existing devices can be modified to work in different specialized environment as well as scale to house use to big labs[Since fire accidents can cause major loss in human lives in homes to big industries] as well as it can be used in public places , vehicles.

13. APPENDIX

Esp32 - Microcontroller :

ESP32 is a series of low-cost, low-power system on a chip microcontrollers with integrated Wi-Fi and dual-mode Bluetooth

Memory: 320 KiB SRAM

CPU: Tensilica Xtensa LX6 microprocessor @ 160 or 240 MHz

Power: 3.3 V DC

Manufacturer: Espressif Systems

Predecessor: ESP8266

Sensors :

DHT22 - Temperature and Humidity sensor

The DHT22 is a basic, low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin (no analog input pins needed).

Flow Sensors

A flow sensor (more commonly referred to as a “flow meter”) is an electronic device that measures or regulates the flow rate of liquids and gasses within pipes and tubes.

MQ5 - Gas sensor

Gas sensors (also known as gas detectors) are electronic devices that detect and identify different types of gasses. They are commonly used to detect toxic or explosive gasses and measure gas concentration.

Flame sensors

A flame-sensor is one kind of detector which is mainly designed for detecting as well as responding to the occurrence of a fire or flame. The flame detection response can depend on its fitting

SOURCE CODE

```
 #include <WiFi.h>//library for wifi

#include <PubSubClient.h>//library for MQtt
#include "DHT.h"// Library for dht11
#include <cstdlib>
#include <time.h>
#include <mjson.h>

//#include <HTTPClient.h>
#define DHTPIN 15     // what pin we're connected to
#define DHTTYPE DHT22   // define type of sensor DHT 11


DHT dht (DHTPIN, DHTTYPE);// creating the instance by passing pin and typr of dht
connected

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

//-------credentials of IBM Accounts------

#define ORG "8li4cg"
#define DEVICE_TYPE
"Fire_4"
#define DEVICE_ID "fire"
#define TOKEN "HC_riXSv8?Wrh+B4S-"

String data3 = "";
String accidentstatus ="";
String sprinkstatus = "";
float temp =0;
bool isfanon = false;
bool issprinkon = false;
bool cansprinkoperate = true;
```

```
bool canalertsent = true;
bool cansentalert = false;
int gas = 0;
int flame = 0;
int flow = 0;
long int cooldown= 600;




char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/data/fmt/json";
char subscribetopic[] = "iot-
2/cmd/command/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;


//-------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client
id by passing parameter like server id,portand wificredential


void setup()// configureing the ESP32
{
  Serial.begin(115200);
  dht.begin();
  //if real gas sensor is used make sure the senor is heated up for acurate readings
  /*
    - Here random values for readings and stdout were used to show the
      working  of the devices as physical or simulated devices are not
      available.
  */


  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}

void loop()
{



  temp = dht.readTemperature();
  //setting a random seed (only for random values not in real life scenarios)
  srand(time(0));
```

```
//initial variable activities like declaring , assigning
gas = rand()%400;
int flamereading = rand()%1024;
flame = map(flamereading,0,1024,0,1024);
int flow = ((rand()%100)>50?1:0);



//find the accident status 'cause fake alert may be caused by some mischief activities
if(temp < 45 ){
  if(flame > 650 ){
    accidentstatus = "Need Auditing";
    if(canfanoperate)
      isfanon = true;
    else
      isfanon = false;
    issprinkon = false;
  }
  else if(flame <= 10){
    accidentstatus = "nothing happened";
    isfanon = false;
    issprinkon = false;
  }

}else if(temp >= 45 && temp <= 55 ){
  if(flame <=650 && flame >100  ){

    if(cansprinkoperate)
      issprinkon = true;
    else
      issprinkon = false;
    accidentstatus = "moderate";
    if(gas > 160 && canfanoperate ){
      isfanon = true;
    }
    else{
      isfanon = false;
    }
  }
  else if(flame <= 100 && flame > 10){
   if(cansprinkoperate)
      issprinkon = true;
    else
      issprinkon = false;
    isfanon = false;
    accidentstatus = "moderate";
  }

}else if(temp > 55){
  if(flame > 650){
    gas = 500 + rand()%500;
    accidentstatus = "severe";
```

```
      if(cansprinkoperate)
        issprinkon = true;
      else
        issprinkon = false;
      if(canfanoperate)
        isfanon = true;
      else
        isfanon = false;
    }
    else if(flame < 650 && flame > 400 ){
      gas = 300 + rand()%500;
      accidentstatus = "severe";
      if(cansprinkoperate)
        issprinkon = true;
      else
        issprinkon = false;

      if(canfanoperate)
        isfanon = true;
      else
        isfanon = false;

    }
  }
  else {
    accidentstatus = "Need moderate Auditing";
    isfanon = false;
    issprinkon = false;
  }


  if(issprinkon){
    if(flow){
      sprinkstatus = "working";
    }
    else{
      sprinkstatus = "not working";
    }
  }
  else if(!issprinkon){
    sprinkstatus = "ready";
  }
  else {
    sprinkstatus = "something's wrong";
  }


  PublishData(temp,gas,flame,flow,isfanon,issprinkon);

  //a cooldown period is set as the values and situations are random in real life
scenarios the time can be reduced or neglected
  if(accidentstatus=="severe" && cooldown >= 600){
```

```
    cooldown = 0;
    sendalert();
    PublishData(temp,gas,flame,flow,isfanon,issprinkon);
    cansentalert = false;

  }

  if(cooldown > 999999){
    cooldown = 601;
  }




  delay(1000);
  ++cooldown;
  if (!client.loop()) {
    mqttconnect();
  }
}



/*.....................................retrieving to
Cloud. ............................*/

void PublishData(float temp, int gas ,int flame ,int flow,bool isfanon,bool issprinkon) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String payload = "{\"temp\":";
  payload += temp;
  payload += "," "\"gas\":";
  payload += gas;
  payload += "," "\"flame\":";
  payload += flame;
  payload += "," "\"flow\":";
  payload += ((flow)?"true":"false");
  payload += "," "\"isfanon\":";
  payload += ((isfanon)?"true":"false");
  payload += "," "\"issprinkon\":";
  payload += ((issprinkon)?"true":"false");
  payload += "," "\"cansentalert\":";
  payload += ((cansentalert)?"true":"false");
  payload += "," "\"accidentstatus\":";
  payload += "\""+accidentstatus+"\"";
  payload += "," "\"sprinkstatus\":";
  payload += "\""+sprinkstatus+"\"";
  payload += "}";
```

```
  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it sucessfully upload data on the cloud then it
will print publish ok in Serial monitor or else it will print publish failed
  } else {
    Serial.println("Publish failed");
  }

}


void mqttconnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting client to ");
    Serial.println(server);
    while (!!!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }

     initManagedDevice();
     Serial.println();
  }
}
void wificonnect() //function defination for wificonnect
{
  Serial.println();
  Serial.print("Connecting to ");

  WiFi.begin("Wokwi-GUEST",  "",  6);
  while (WiFi.status() != WL_CONNECTED) {
    delay(100);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void initManagedDevice() {
  if (client.subscribe(subscribetopic)) {
    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}

//handles commands from user side
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
```

```
{

  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);
  for (int i = 0; i < payloadLength; i++) {

    data3 += (char)payload[i];
  }
  Serial.println("data: "+ data3);

  const char *s =(char*) data3.c_str();
  double pincode = 0;

  if(mjson_get_number(s, strlen(s), "$.pin", &pincode)){
    if(((int)pincode)==67993){
        const char *buf;
        int len;

        if (mjson_find(s, strlen(s), "$.command", &buf, &len))  // And print it
        {

          String command(buf,len);
          if(command=="\"cantfan\""){

            canfanoperate = !canfanoperate;
          }
          else if(command=="\"cantsprink\""){
            cansprinkoperate = !cansprinkoperate;
          }else if(command=="\"sentalert\""){
            resetcooldown();
          }
        }



    }
  }

  data3="";
}

void resetcooldown(){
  cooldown = 0;
}

//sent alert request to node-red
void sendalert(){
cansentalert = true;
cooldown = 0;}
```

GITHUB LINK : https://github.com/IBM-EPBL/IBM-Project-2106-1658449925