INDUSTRIAL SPECIFIC INTELLIGENT FIRE MANAGEMENT SYSTEM

PROJECT REPORT

Submitted by

SOWMIYA M G	**-**	**2031T306**
PURNIMA R G	**-**	**2031T304**
VAISHNAVI B	**-**	**2031T307**
MUKITHA YOGESWARI M	**-**	**1931036**

in partial fulfillment of the requirements for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

GOVERNMENT COLLEGE OF ENGINEERING

SALEM

(An Autonomous Institution)

ANNA UNIVERSITY, CHENNAI

INTRODUCTION

1.1 Project overview

Fire alarm systems are only effective if they can generate reliable and fast fire alerts with exact location of fire. There is a direct correlation between the amount of damage caused by fire and interventions time in various fire alarm systems. As the time of intervention decreases, the damage also decreases. Hence the most important factor in a fire alarm system is the reaction or response time of fire alarm system, that is, the time between fire detection and extinguishing.

Fire safety is the one among the various areas that can utilizes the extraordinary benefits of the Internet of Things (IOT) as it has led to much of the world becoming smarter and more connected. With IOT, safety alerts can be sent to hundreds of people fast and effectively. Several leading fire safety companies have already launched IOT-enabled fire detectors.

1. Industrial IOT Enabled Connected Detectors: There are variety of connected smoke and gas detectors for domestic and industrial applications. These connected detectors are able to communicate in real time with the other devices and can be programmed to take a limited judgemental call for a pre-decided action. The detectors can be accessed from anywhere using mobile apps and internet connectivity. In the event of an alarm, the detectors can sound a local alarm as well as send notifications on the mobile phones.
2. IOT Retrofitting: Technology is also available today to add connectivity to existing detectors. With a monitor, users don't have to change all the detectors. The monitor listens for the specific frequency of these detectors and sends an alert to its application. One single monitor can cover multiple detectors which covers large areas.

1.2 Purpose

1. It must be able to detect fire at all locations within a range
2. It gives out the alert at right time when the fire is detected

2.LITERATURE SURVEY

2.1 Existing problem

1. Current system uses hard wired interconnection which is having disadvantage of cost expensive, long time consuming and disruptive. A hard-wired system is also very difficult to maintain and too expensive to reconfigure when circumstances changes.
2. Fire alarm system are essential in alerting people before fire engulfs.However, fire alarm systems today require a lot of wiring to be installed.
3. The existing method produces only the alarm whenever gas is detected at any place. Due to this alarm, situation becomes haphazard. As a result, worker in the factory gets injured severely. Sometimes people do not realize the intensity of the fire.

2.2 References

[1] Liu Yunhong, Qi Meini, "The Design of Building Fire Monitoring System Based on ZigBeeWiFi 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] 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

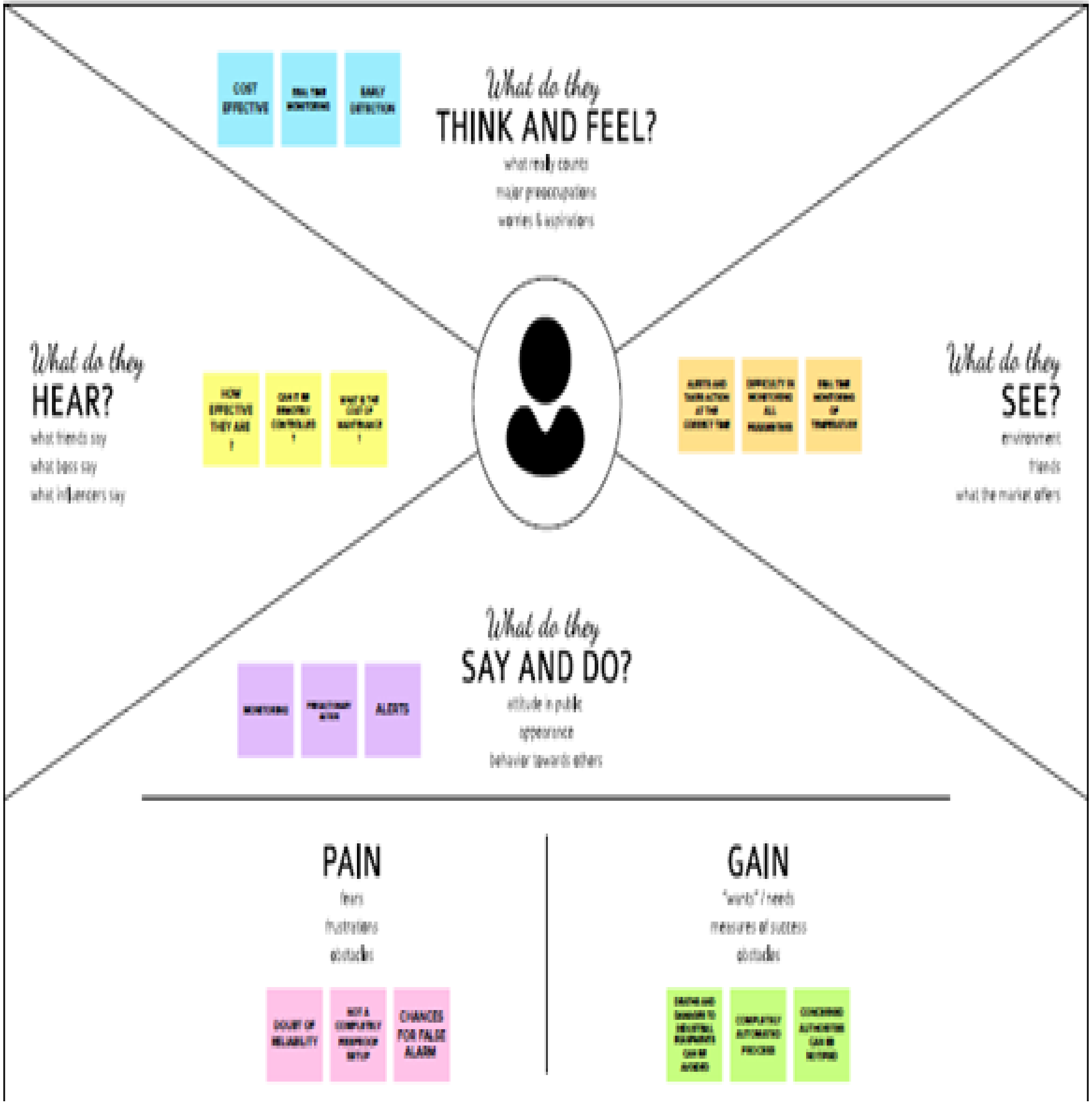
[4] Azka Ihsan Nurrahman, Kusprasapta Mutijarsa, "Intelligent fire management system prototype design and development", International Conference on Information Technology Systems and Innovation (ICITSI), IEEE, 2015

2.3 Problem Statement Definition

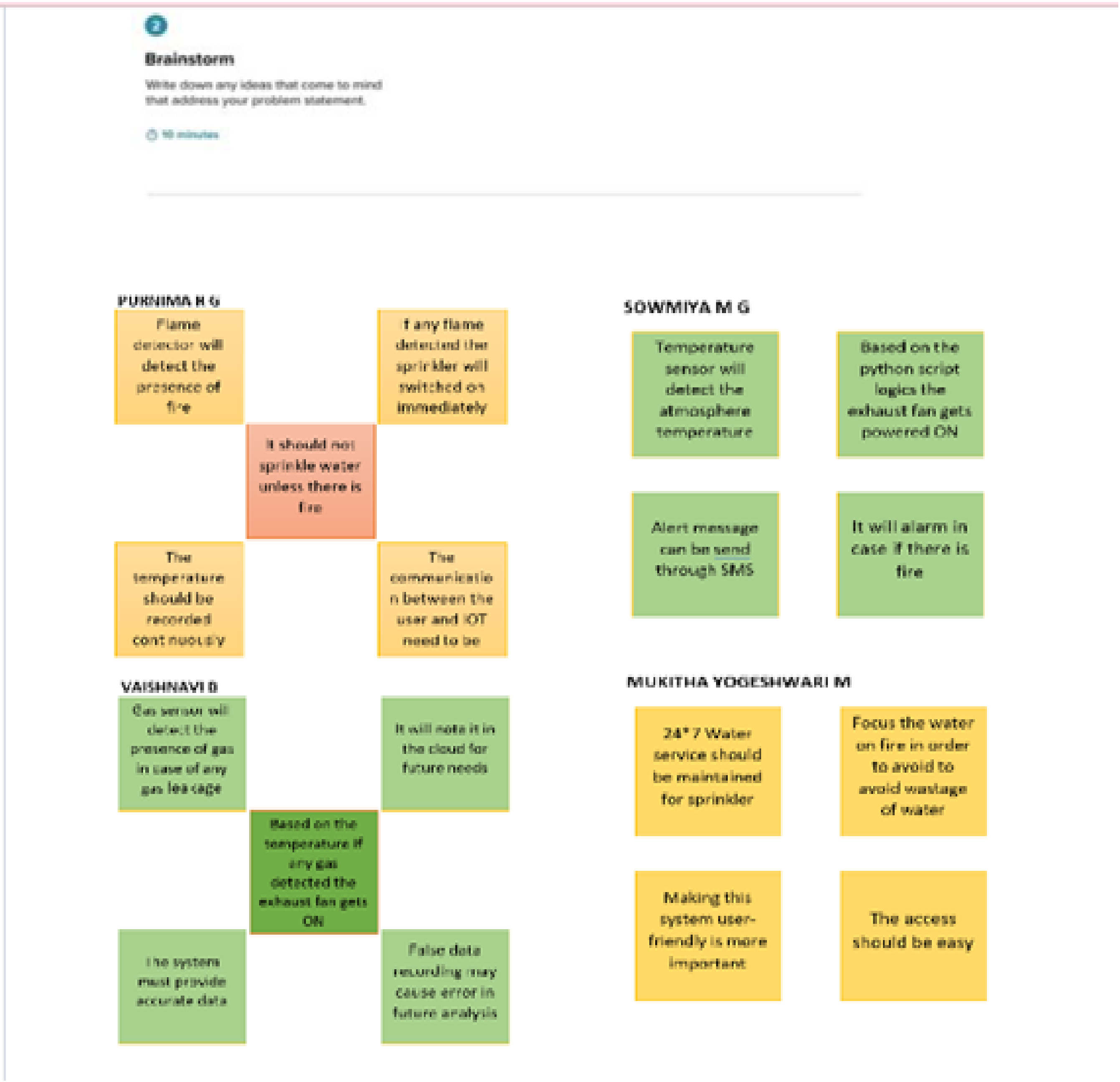
- Due to false warnings,productivity is affected.
- It fails to send the alert at right time.
- Industry Specific Intelligent fire management system are designed to prevent false alarming.

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map



3.2 Ideation &Brainstorming



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

PURNIMA R G

Flame detector will detect the presence of fire

f any flame detected the sprinkler will switched on immediately

It should not sprinkle water unless there is fire

The temperature should be recorded continuously

The communication between the user and IOT need to be

VAISHNAVI B

Gas sensor will detect the presence of gas in case of any gas leakage

It will note it in the cloud for future needs

Based on the temperature if any gas detected the exhaust fan gets ON

The system must provide accurate data

False data recording may cause error in future analysis

SOWMIYA M G

Temperature sensor will detect the atmosphere temperature

Based on the python script logics the exhaust fan gets powered ON

Alert message can be send through SMS

It will alarm in case if there is fire

MUKITHA YOGESHWARI M

24* 7 Water service should be maintained for sprinkler

Focus the water on fire in order to avoid to avoid wastage of water

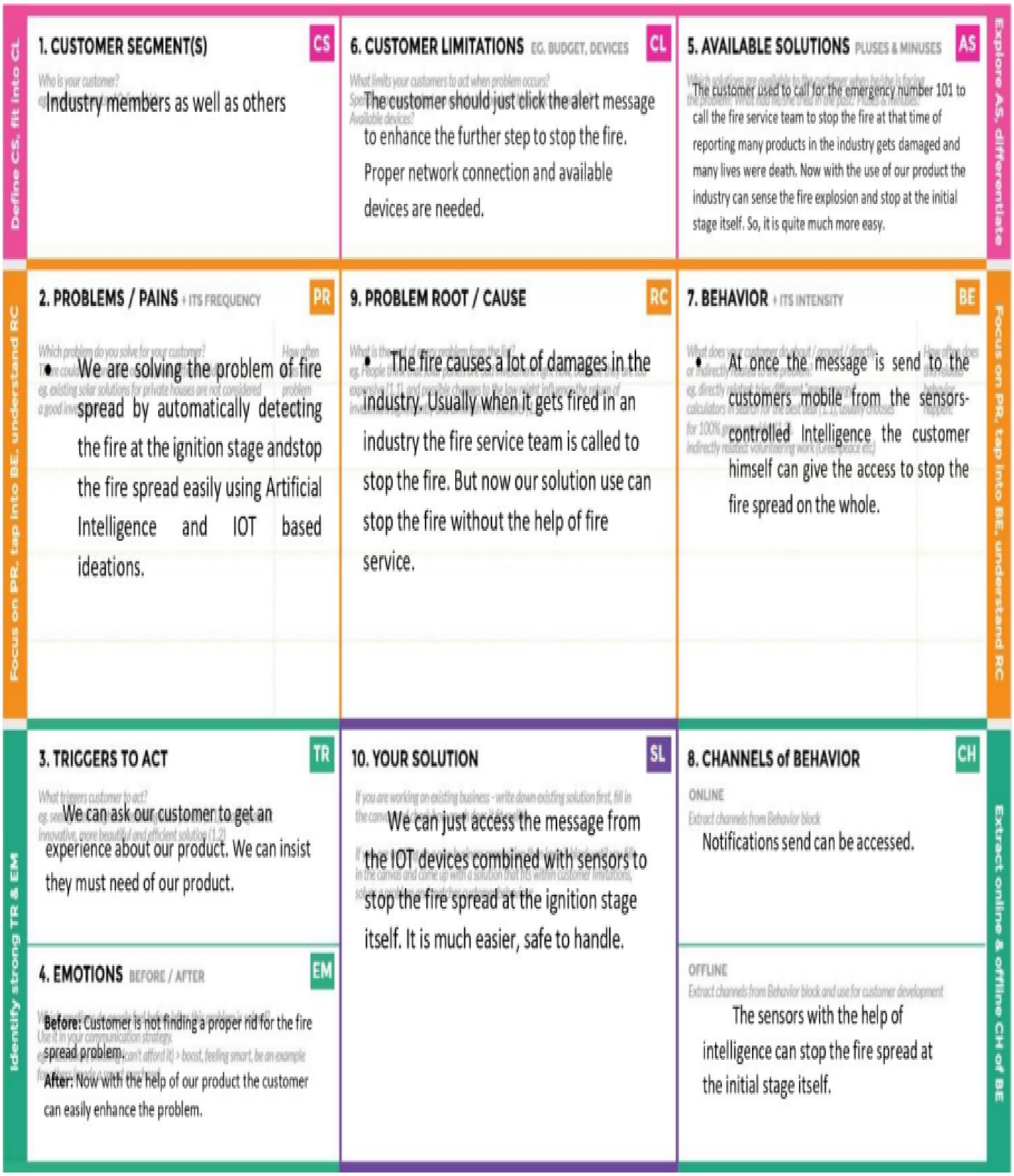
Making this system user-friendly is more important

The access should be easy

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Enhancing the safety measures in industries that occur due to fire accidents and implementing the same.
2.	Idea / Solution description	Execution of fire management based on IOT consisting of Arduino UNO board that comprises fire detection and fire extinguisher system, with the help of sensors like (Temperature sensor, Smoke sensor, Flame Sensor)which has Fast SMS alert system.
3.	Novelty / Uniqueness	Making the best use of integrating certain tasks like temperature monitoring, gas monitoring, fire detection and automatic sprinklers so as to obtain accurate information about exact locations and to get response through SMS notifications and calls.
4.	Social Impact / Customer Satisfaction	Forecasting the mishap will notify the industry workers to migrate to better and safer buildings. Provides components with affordable prices and is highly feasible.
5.	Business Model (Revenue Model)	It is an industry-efficient product in all aspects. Provides a clear idea about the entire working mechanism of the system.
6.	Scalability of the Solution	Since, it deals with Arduino gadgets that must be capable of handling real-time signals from sensors. Helps in maintaining a large increase in workload without undue strain.

3.4 Problem Solution Fit



Define CS, fit into CL

1. CUSTOMER SEGMENT(S) CS

Who is your customer?

Industry members as well as others

6. CUSTOMER LIMITATIONS EG. BUDGET, DEVICES CL

What limits your customers to act when problem occurs?

Available devices?

The customer should just click the alert message to enhance the further step to stop the fire. Proper network connection and available devices are needed.

5. AVAILABLE SOLUTIONS PLUSES & MINUSES AS

The customer used to call for the emergency number 101 to call the fire service team to stop the fire at that time of reporting many products in the industry gets damaged and many lives were death. Now with the use of our product the industry can sense the fire explosion and stop at the initial stage itself. So, it is quite much more easy.

Explore AS, differentiate

Focus on PR, tap into BE, understand RC

2. PROBLEMS / PAINS + ITS FREQUENCY PR

Which problem do you solve for your customer?

- We are solving the problem of fire spread by automatically detecting the fire at the ignition stage andstop the fire spread easily using Artificial Intelligence and IOT based ideations.

9. PROBLEM ROOT / CAUSE RC

- The fire causes a lot of damages in the industry. Usually when it gets fired in an industry the fire service team is called to stop the fire. But now our solution use can stop the fire without the help of fire service.

7. BEHAVIOR + ITS INTENSITY BE

- At once the message is send to the customers mobile from the sensors-controlled Intelligence the customer himself can give the access to stop the fire spread on the whole.

Focus on PR, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TO ACT TR

What triggers customer to act?

We can ask our customer to get an experience about our product. We can insist they must need of our product.

4. EMOTIONS BEFORE / AFTER EM

Before: Customer is not finding a proper rid for the fire spread problem.

After: Now with the help of our product the customer can easily enhance the problem.

10. YOUR SOLUTION SL

We can just access the message from the IOT devices combined with sensors to stop the fire spread at the ignition stage itself. It is much easier, safe to handle.

8. CHANNELS of BEHAVIOR CH

ONLINE

Extract channels from Behavior block

Notifications send can be accessed.

OFFLINE

Extract channels from Behavior block and use for customer development

The sensors with the help of intelligence can stop the fire spread at the initial stage itself.

Extract online & offline CH of BE

4.REQUIREMENT ANALYSIS

4.1Functional requirement

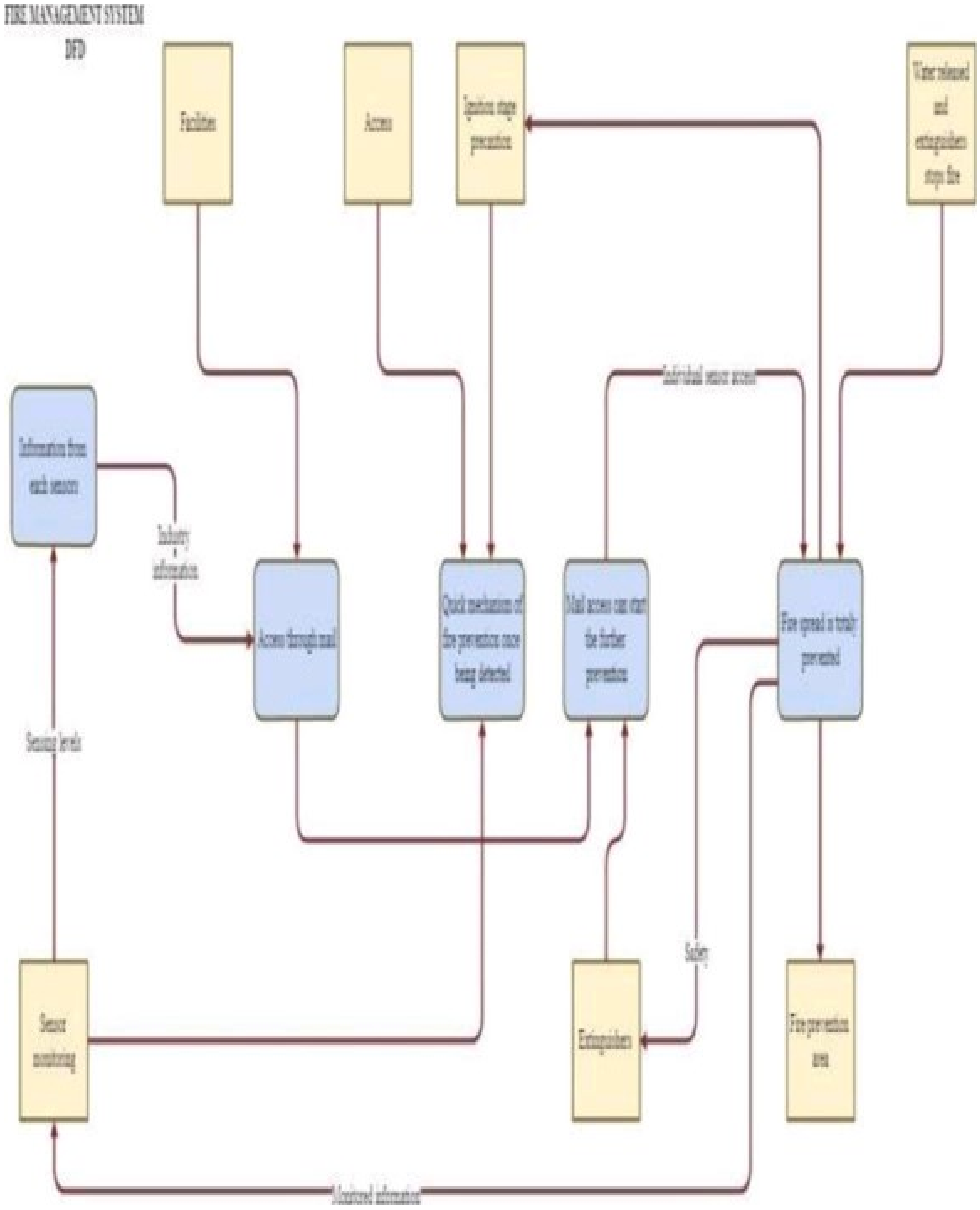
FR no	Functional Requirement(Epic)	Sub Requirement (Story / Sub-Task)
FR_1	User Registration	Registration through website or application Registration through Social medias Registration through Linkedin
FR_2	IJser Confirm action	Verification via Email or OTP
FR_3	User Login	Login through website or App using the respective username and password
FR_4	User Access	Access the app requirements
FR_5	User Upload	User should be able to upload the data
FR_6	User Solution	Data report should be generated and delivered to user for every 24 hours
FR_7	User Data Sync	API interface to increase to invoice system

4.2 Non-Functional requirement

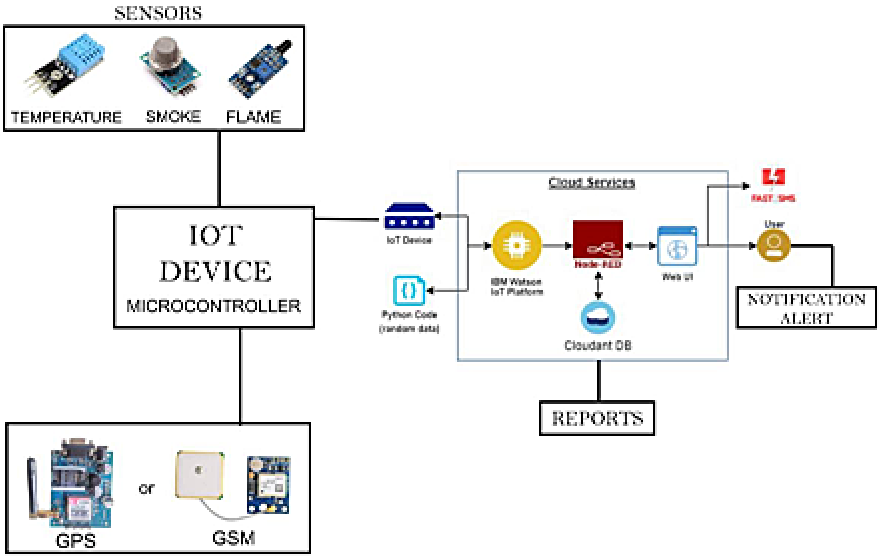
NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	Usability requirements includes language barriers and localization tasks. Usability can be assessed by Efficiency of use.
NFR-2	Security	Access permissions for the particular system information may only be changed by the system's data administrator.
NFR-3	Reliability	The database update process must roll back all related updates when any update fails.
NFR-4	Performance	The front-page load time must be no more than 2 seconds for users that access the website using a VoLTE mobile connection.
NFR-5	Availability	New module deployment must not impact front page, product pages, and check out pages availability and mustn't take longer than one hour.

5.PROJECT DESIGN

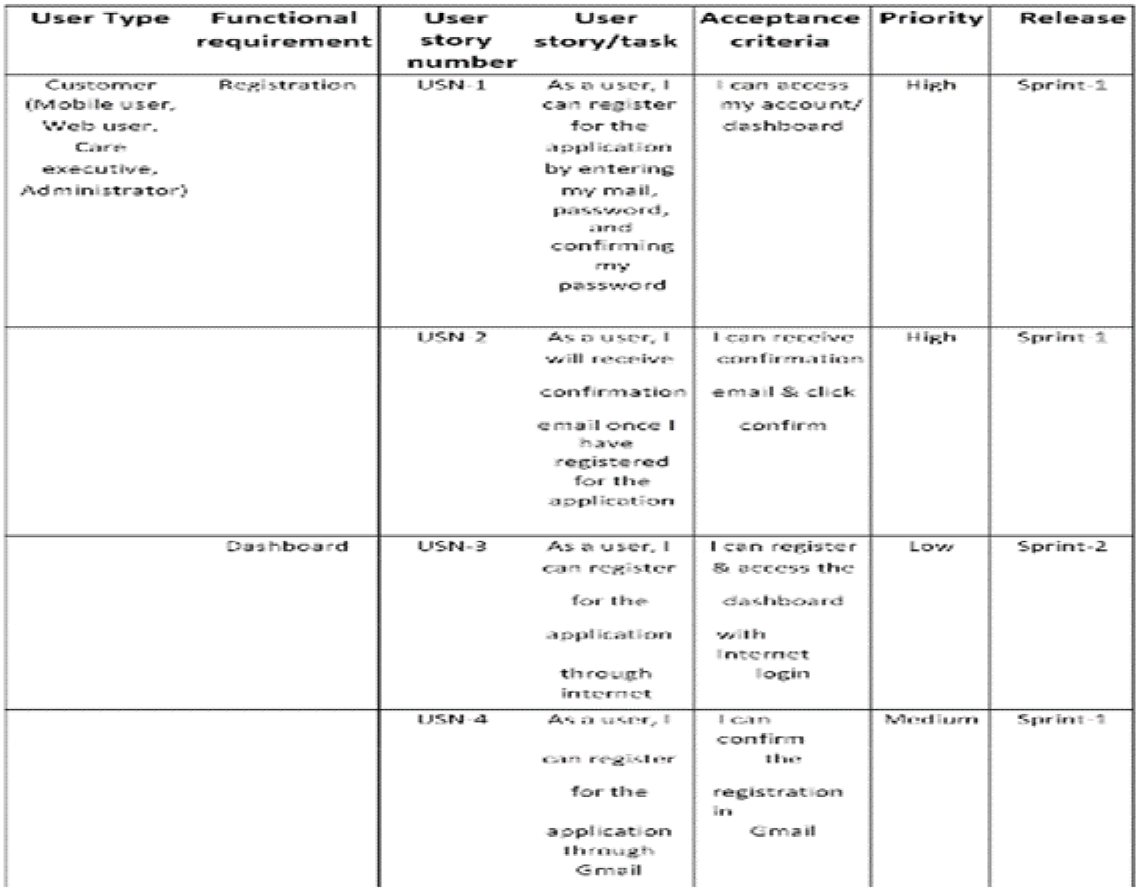
5.1 Data Flow Diagram


FIRE MANAGEMENT SYSTEM DFD
Facilities
Access
Ignition stage precaution
Water released and extinguishers stops fire
Individual sensor access
Information from each sensors
Industry information
Access through mail
Quick mechanism of fire prevention once being detected
Mail access can start the further prevention
Fire spread is totaly prevented
Sensing levels
Safety
Sensor monitoring
Extinguishers
Fire prevention area
Monitored information

5.2 Solution Architecture



5.3 User Stories



User Type	Functional requirement	User story number	User story/task	Acceptance criteria	Priority	Release
Customer (Mobile user, Web user, Care executive, Administrator)	Registration	USN-1	As a user, I can register for the application by entering my mail, password, and confirming my password	I can access my account/ dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
	Dashboard	USN-3	As a user, I can register for the application through internet	I can register & access the dashboard with Internet login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can confirm the registration in Gmail	Medium	Sprint-1

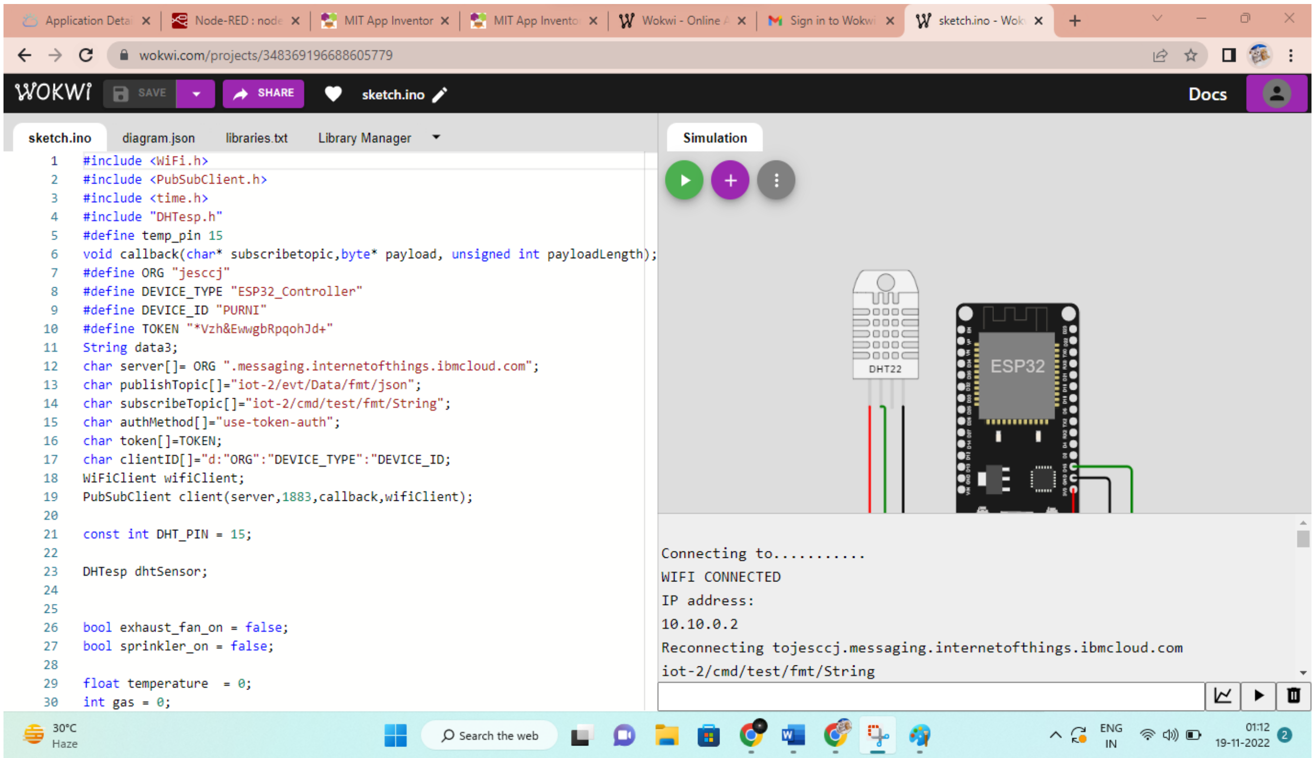
6.PROJECT PLANNING &SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement	User Story Number	User Story Task	Story Points	Priority	Team Members
Sprint-1	WOKWI	USN-1	using the wokwi we will connect the components and send to cloud	2	High	Sowmiya M G Vaishnavi B Purnima R G MukithaYogeshwariM
Sprint-2	Software	USN-2	IBM Watson lot NodeRed integration/ Test the browser device and workflow.	2	High	Sowmiya M G Vaishnavi B Purnima R G MukithaYogeshwariM
Sprint-3	Application Development/ Testing	USN-3	IJsing MIT App Inventor we have to create an App/ Testing the Application.	2	High	Sowmiya M G Vaishnavi B Purnima R G MukithaYogeshwariM
Sprint-4	WEB UI	USN-4	user interface with the Software	2	High	Sowmiya M G Vaishnavi B Purnima R G MukithaYogeshwariM

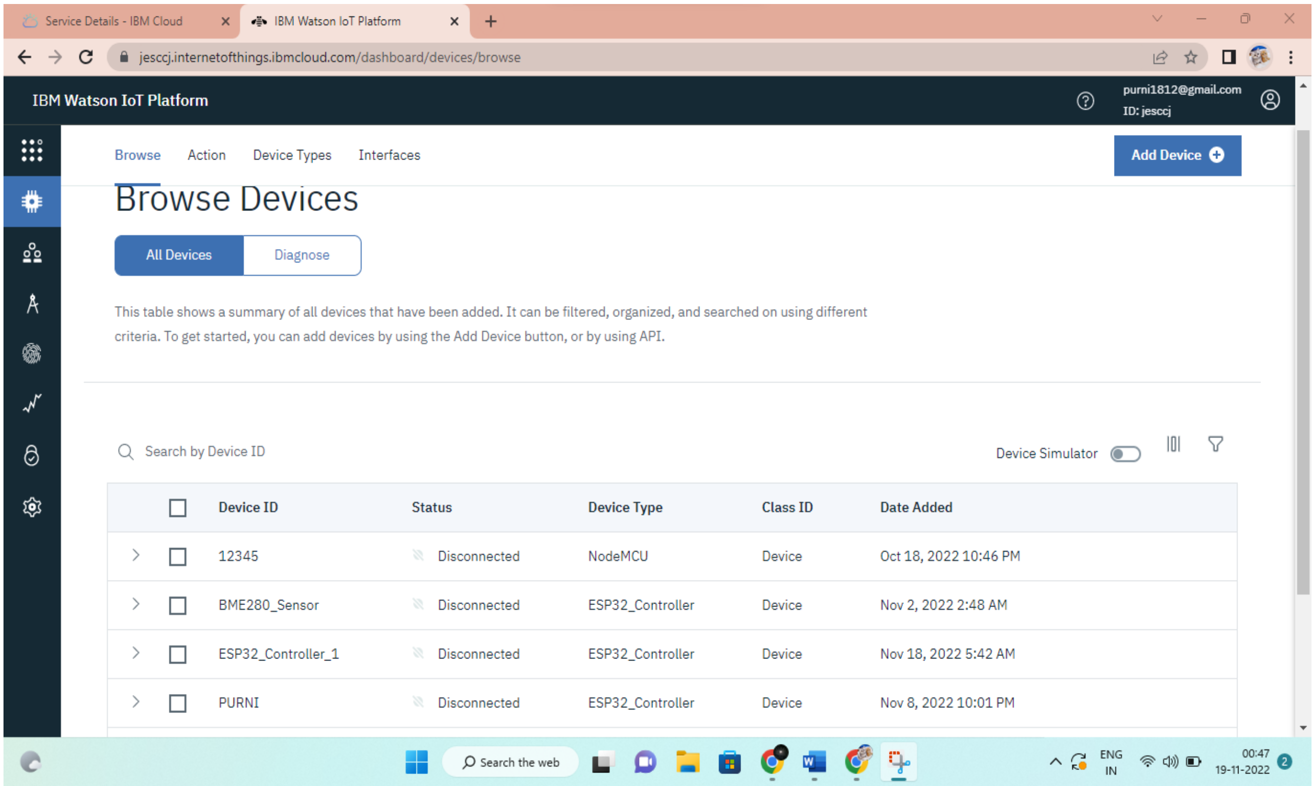
7.CODING & SOLUTIONING

7.1 Feature 1



```
#include <WiFi.h>
#include <PubSubClient.h>
#include <time.h>
#include "DHTesp.h"
#define temp_pin 15
void callback(char* subscribetopic,byte* payload, unsigned int payloadLength);
#define ORG "jesccj"
#define DEVICE_TYPE "ESP32_Controller"
#define DEVICE_ID "PURNI"
#define TOKEN "*Vzh&EwwgbRpqohJd+"
String data3;
char server[]= ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[]="iot-2/evt/Data/fmt/json";
char subscribeTopic[]="iot-2/cmd/test/fmt/String";
char authMethod[]="use-token-auth";
char token[]=TOKEN;
char clientID[]="d:"ORG":"DEVICE_TYPE":"DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server,1883,callback,wifiClient);

const int DHT_PIN = 15;

DHTesp dhtSensor;


bool exhaust_fan_on = false;
bool sprinkler_on = false;

float temperature  = 0;
int gas = 0;
```

```
Connecting to...........
WIFI CONNECTED
IP address:
10.10.0.2
Reconnecting tojesccj.messaging.internetofthings.ibmcloud.com
iot-2/cmd/test/fmt/String
```

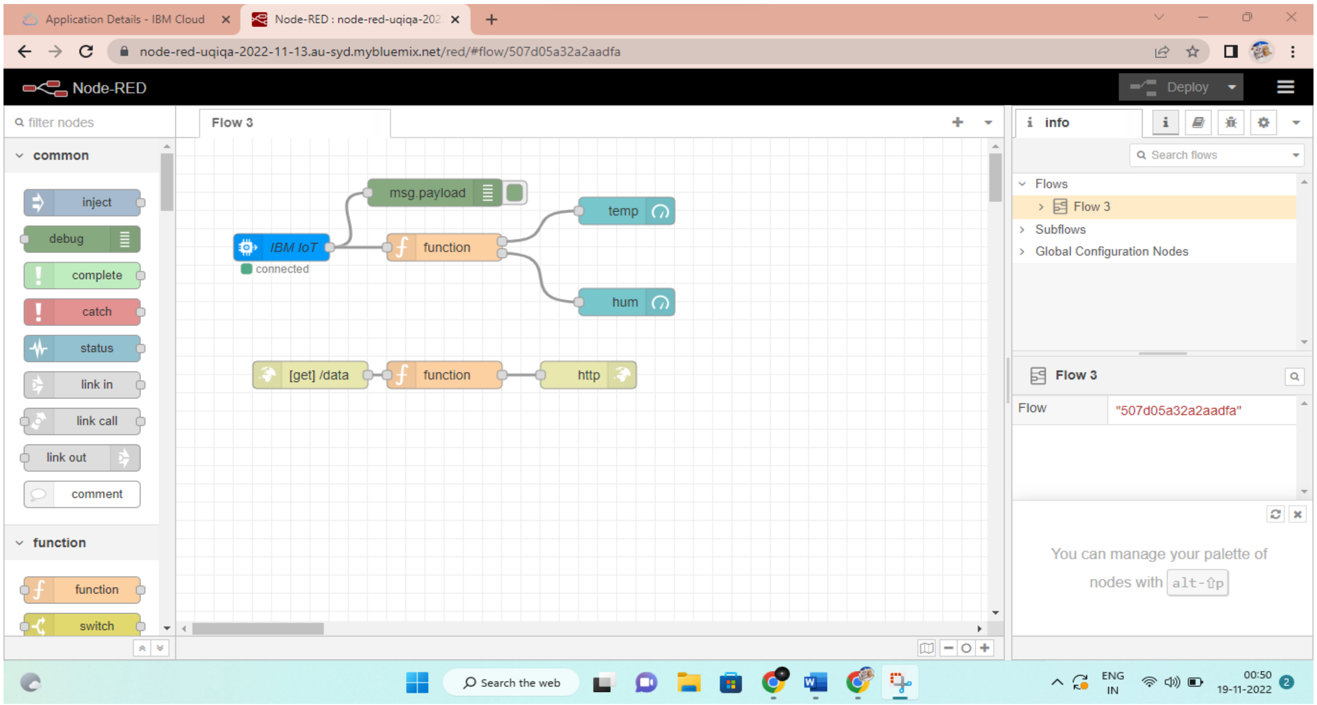


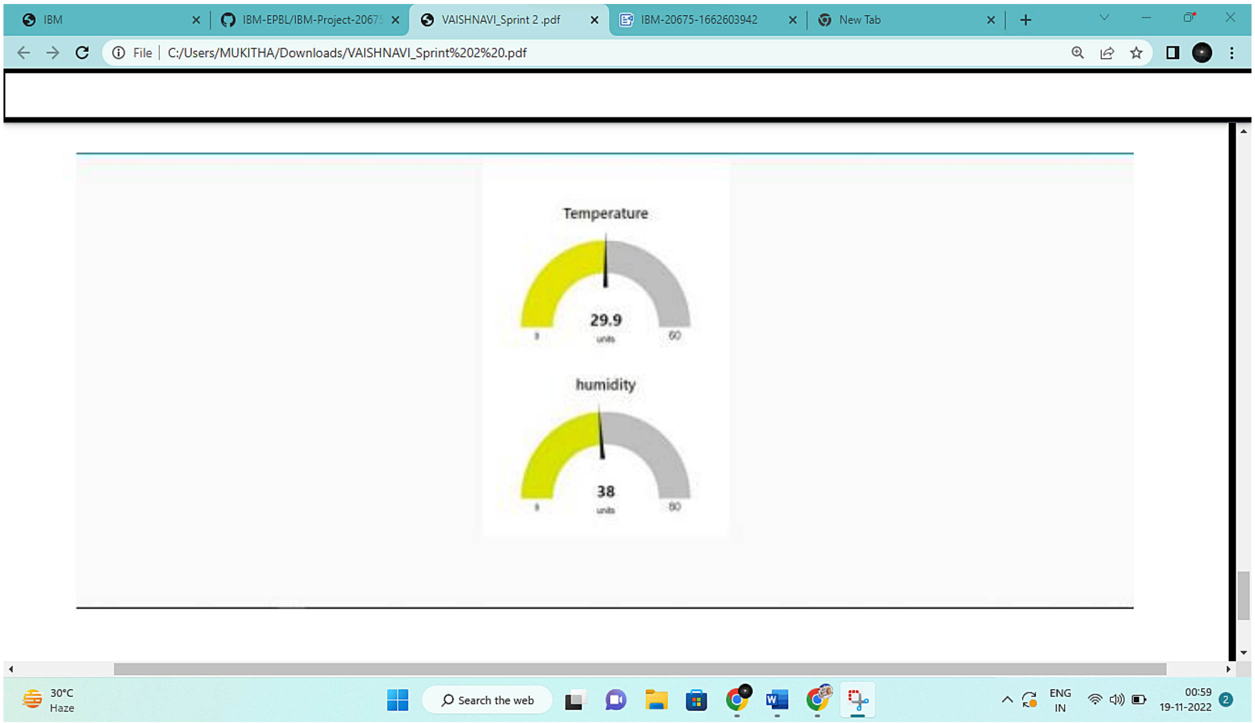
Browse Devices

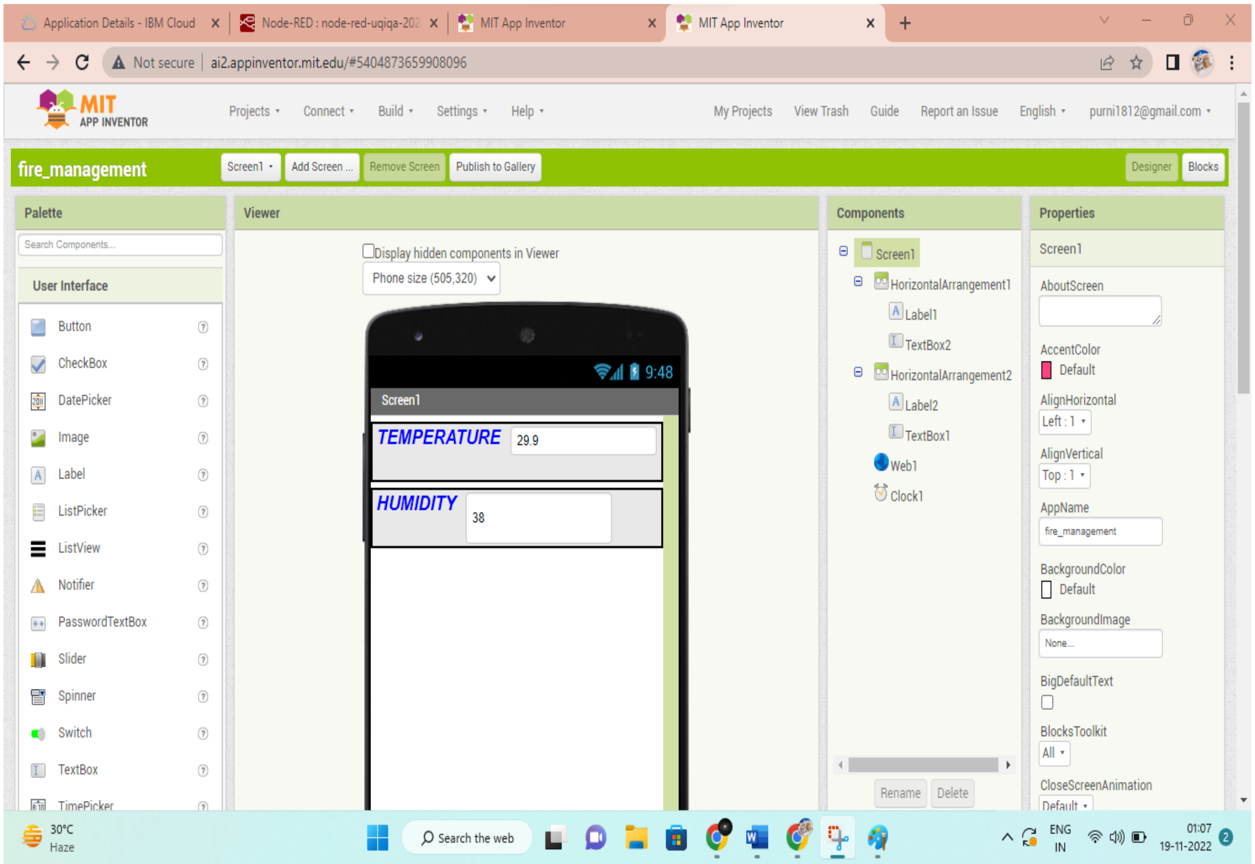
This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Device ID	Status	Device Type	Class ID	Date Added
12345	Disconnected	NodeMCU	Device	Oct 18, 2022 10:46 PM
BME280_Sensor	Disconnected	ESP32_Controller	Device	Nov 2, 2022 2:48 AM
ESP32_Controller_1	Disconnected	ESP32_Controller	Device	Nov 18, 2022 5:42 AM
PURNI	Disconnected	ESP32_Controller	Device	Nov 8, 2022 10:01 PM

7.2 Features 2







TEMPERATURE 29.9

HUMIDITY 38

- Screen1
  - HorizontalArrangement1
    - Label1
    - TextBox2
  - HorizontalArrangement2
    - Label2
    - TextBox1
  - Web1
  - Clock1

Screen1

AboutScreen

AccentColor: Default

AlignHorizontal: Left : 1

AlignVertical: Top : 1

AppName: fire_management

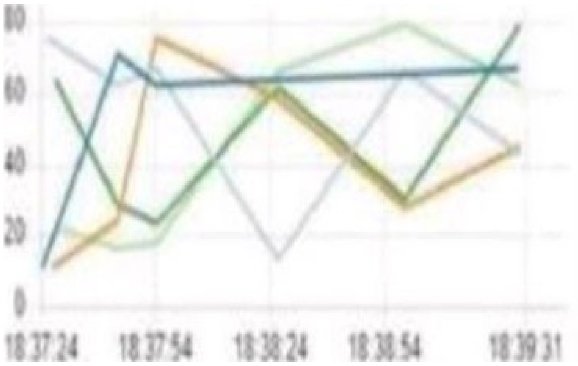
BackgroundColor: Default

BackgroundImage: None...

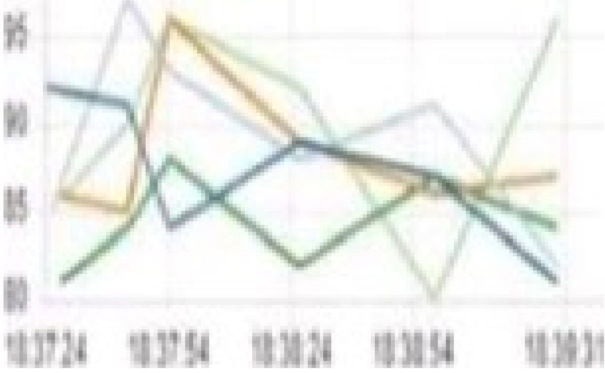
BigDefaultText: □

BlocksToolkit: All

CloseScreenAnimation: Default



Temperature



Humidity

9.RESULTS

9.1 Performance Metrics

1. Hours worked : 48 hours
2. Efficiency of the product:100%
3. Quality of the product:100%

10.ADVANTAGES

- It reduces the false warnings.
- The installation cost is low.
- This system monitors the surrounding 24/7.
- It improves security in industries and Offices.

11.DISADVANTAGES

1. This system cannot be implement in large scale industries.
2. The Control pannel need to be replaced, if it gets damaged.

12.CONCLUSION

- ➤ This system helps in reducing false warning.
- ➤ This system intimates the authorities at right time about the suitiation.
- ➤ As the system is cost effective it can be easily implemented in small scale industeries.

13.FUTURE SCOPE

- ✓ With the addition artificial Intelligent technology,Fire management system can be made automated.
- ✓ With the use of PIR(Passive Infrared Sensor)the count of human can be detected in that area and prioritize it,which helps in human life saving.

14.APPENDIX

14.1 Source Code

```
#include<WiFi.h>

#include<PubSubClient.h>

#include<time.h>

#include"DHTesp.h"#def
ine temp_pin 15

void callback(char* subscribetopic,byte* payload, unsignedint payloadLength);
#define ORG "jesccj"

#define DEVICE_TYPE "ESP32_Controller"

#define DEVICE_ID "PURNI"

#define TOKEN "*Vzh&EwwgbRpqohJd+"

String data3;

char server[]= ORG ".messaging.internetofthings.ibmcloud.com";
charpublishTopic[]="iot-2/evt/Data/fmt/json";

charsubscribeTopic[]="iot-2/cmd/test/fmt/String";

charauthMethod[]="use-token-auth";

char token[]=TOKEN;
char clientID[]="d:"ORG":"DEVICE_TYPE":"DEVICE_ID;

WiFiClient wifiClient;

PubSubClient client(server,1883,callback,wifiClient);


constint DHT_PIN = 15;


DHTesp dhtSensor;

bool exhaust_fan_on = false;

bool sprinkler_on = false;


float  temperature   =
0; int gas = 0;

int flame = 0;


String flame_status = "";

String accident_status = "";

String sprinkler_status = "";


voidsetup()                              {
Serial.begin(99900);


wificonnect();

mqttconnect();


  dhtSensor.setup(DHT_PIN, DHTesp::DHT22);

}
```

```
voidloop() {
 srand(time(0));

   //initial variable

temperature = random(-20,125);
gas = random(0,1000);
int flamereading =
random(200,1024);
 flame = map(flamereading,0,1024,0,2);

  TempAndHumidity  data = dhtSensor.getTempAndHumidity();
Serial.println("Temperature:  "+  String(data.temperature,  2)  +
"°C");
Serial.println("Humidity: " + String(data.humidity, 1) + "%");
Serial.println("---");
delay(1000);
if(data.temperature<38){
PublishData1(data.temperature);
      flame_status = "No Fire";
      Serial.println("Flame Status : "+flame_status);
   }
   else{
     PublishData2(data.temperature);
     flame_status = "Fire is Detected";
```

```
      Serial.println("Flame Status : "+flame_status);

    }
    if(data.humidity<30){
      PublishData3(data.humidity);
      Serial.println("Gas Status : Gas leakage Detected");
    }
    else{
       PublishData4(data.humidity);
       exhaust_fan_on = false;
       Serial.println("Gas Status : No Gas leakage Detected");
    }

    //send the sprinkler status
   if(data.temperature<38){
       sprinkler_status = " not working";
       Serial.println("Sprinkler Status : "+sprinkler_status);
 }else{
       sprinkler_status = " working";
       Serial.println("Sprinkler Status : "+sprinkler_status);
    }

    //toggle the fan according to gas

    if(data.humidity<30){
        exhaust_fan_on = true;
        Serial.println("Exhaust fan Status : Working"); }
else{
        exhaust_fan_on = false;
        Serial.println("Exhaust fan Status : Not Working");
    }

    Serial.println("");
    Serial.println("");
              Serial.println("   -----------------*************---------------------");
Serial.println("");
              Serial.println("");
delay(1000);
if(!client.loop()){
mqttconnect();
}
}void           PublishData1(float           temp){
mqttconnect();
      String   payload   =   "{\"temp\":";
payload += temperature;
payload += ",\"nrml!\":""\"temperature less than 38\"";
payload += "}";
Serial.print("Sending payload: ");
Serial.println(payload);

if(client.publish(publishTopic,(char*)payload.c_str())){
    Serial.println("publish ok");
  } else{
    Serial.println("publish failed");
  }
}
voidPublishData2(float
temperature){
  mqttconnect();
  String payload = "{\"temp\":";
  payload += temperature;
   payload += ",\"ALERT!!\":""\"temperature greater than 38\"";
  payload += "}";
Serial.print("Sending          payload:          ");
Serial.println(payload);
  if(client.publish(publishTopic,(char*)payload.c_str())){
    Serial.println("publish ok");
  } else{
    Serial.println("publish failed");
   }
}
void PublishData3(float humidity){
mqttconnect();
String payload = "{\"hum\":";
payload += humidity;
payload += ",\"ALERT!!\":""\"humidity less than 30\"";
payload += "}";
Serial.print("Sending payload: ");
Serial.println(payload);
  if(client.publish(publishTopic,(char*)payload.c_str())){
    Serial.println("publish ok");
  } else{
    Serial.println("publish failed");
   }
}
voidPublishData4(float  humidity){
mqttconnect();
  String payload = "{\"hum\":";
  payload += humidity;
  payload += ",\"nrml!!\":""\"humidity greater than 30\"";
  payload += "}";
  Serial.print("Sending payload: ");
  Serial.println(payload);
if(client.publish(publishTopic,(char*)payload.c_str())){
  Serial.println("publish ok");
  } else{
  Serial.println("publish failed");
   }
}
void  mqttconnect(){
if(!client.connected()){
Serial.print("Reconnecting  to");
Serial.println(server);
```

```
    while(!!!client.connect(clientID, authMethod, token)){
     Serial.print(".");
     delay(500);
    }
    initManagedDevice();
    Serial.println();
  }
}

void wificonnect(){
  Serial.println();
  Serial.print("Connecting to");

  WiFi.begin("Wokwi-GUEST","",6);
  while(WiFi.status()!=WL_CONNECTED){
                        delay(500);
  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");
  }
}
 void callback(char* subscribeTopic, byte* payload, unsignedint payloadLength){
Serial.print("callback invoked for
topic:");   Serial.println(subscribeTopic);
 for(int i=0; i<payloadLength; i++){
data3 += (char)payload[i];
  }
      }
```

13.2 GitHub

GitHub Link:

https://github.com/1BM-EPBL/IBM-Project-20675-1659760254

Wokwi Link:

https://wokwi.com/projects/348369196688605779

Demo Link:

https://drive.google.com/file/d/19rtD4K5V98m19sVTni11HjXFJjtLWfm-/view?usp=drivesdk