M.A.M. COLLEGE OF ENGINEERING AND TECHNOLOGY

TRICHY-621105

INDUSTRY-SPECIFIC INTELLIGENT FIRE MANAGEMENT SYSTEM

PROJECT REPORT

TEAM ID: PNT2022TMID45189

MENTOR: S.KAVITHA

TEAM MEMBERS:

1.SELVAGANAPATHI M(Leader)

2.ARAVIND T

3.LAIYANCEMARY A

4.SATHISH I

Project Report

1. **INTRODUCTION**
    1.1 Project Overview
    1.2 Purpose
2. **LITERATURE SURVEY**
    2.1 Existing problem
    2.2 References
    2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
    3.1 Empathy Map Canvas
    3.2 Ideation & Brainstorming
    3.3 Proposed Solution
    3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
    4.1 Functional requirement
    4.2 Non-Functional requirements
5. **PROJECT DESIGN**
    5.1 Data Flow Diagrams
    5.2 Solution & Technical Architecture
    5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
    6.1 Sprint Planning & Estimation
    6.2 Sprint Delivery Schedule
    6.3 Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
    7.1 Feature 1
    7.2 Feature 2
    7.3 Database Schema (if Applicable)
8. **TESTING**
    8.1 Test Cases
    8.2 User Acceptance Testing
9. **RESULTS**
    9.1 Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**
    Source Code
    GitHub & Project Demo Link

INDUSTRY-SPECIFIC INTELLIGENT FIRE MANAGENMENT SYSTEM

1.INTRODUCTION

1.1 Project Overview

FIRE DETECTION:

Most of us are familiar with smoke and heat detectors because these devices are installed in our homes. Detecting a fire is typically accomplished by installing smoke and/or heat detectors, manual pull stations and automatic sprinkler system water flow switches. Another important detection device is a water flow switch that detects movement of water in the sprinkler piping, typically done by a paddle-type device that is activated by water moving through a pipe when a sprinkler is activated.

OCCUPANT NOTIFICATION:

When it comes to warning building occupants of a fire, you need both audible and visible alarms. Audible alarms include horns (the appliance that produces that loud and very annoying sound) or speakers that play a pre-recorded message and/or allow emergency responders to provide spoken instructions to the occupants. Visual notification appliances, or strobes, can provide notification to occupants that are hearing impaired. Each building type will require a different type of occupant notification system.

CONTROL FUNCTIONS:

There are many different types of control functions that the typical fire alarm system performs. The activation of a duct mounted smoke detector will shut down the heating ventilation and air conditioning (HVAC) equipment to prevent the migration of smoke to non-affected areas of the building. A smoke detector in an elevator lobby will automatically recall the cars to a designated floor.

IOT TECHNOLOGY IN FIRE ALARM

An internet of things network is a system that gathers, transfers, and stores data using programmable software, sensors, electronics, and communication facilities. The system is designed to notify and alert a remote fire station and user/owner when a fire accident occurs.

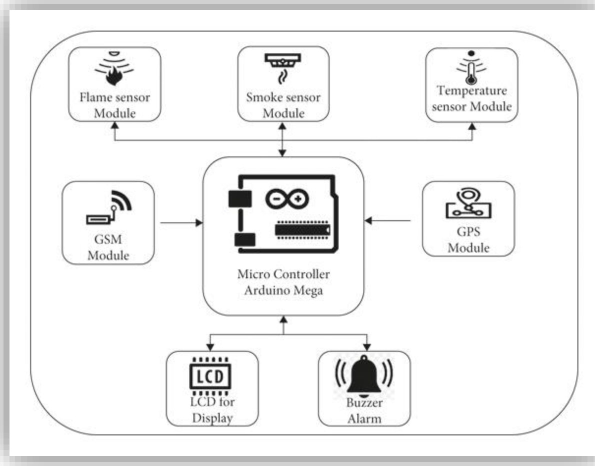
With the IoT, the system reads the data of heat and flame and analyzes it, and then immediately sends a call to the fire station through GSM and informs about its location. Thus, this research focuses on developing an affordable, responsive, and effective, low-cost fire detection system.

Due to its low cost and ease of development, it has recently been used in various applications. Hence, a fire detection system with the internet of things capabilities is essential to protecting homeowners' production units, stores, and other property, as it identifies early fires.

FIRE DEPARTMENT NOTIFICATION:

The codes require that fire alarm signals be automatically transmitted directly to the local fire department, or to a central station monitoring service, notifying the fire department of the emergency. This gives the fire department a signal immediately and allows them the opportunity to extinguish the fire before it gets too big.

Overall, the fire alarm industry is constantly evolving and it is critical that you stay ahead of the latest changes. We leverage our knowledge in fire alarm system technology and codes to deliver the latest solutions.



1.2 PURPOSE:

In the past few years, fire alarm systems have become increasingly sophisticated and more capable and reliable. The two main objectives are the **protection of life** and **property**. As a result of state and local codes, fire protection has become more concerned with life safety over the past two decades. Several safety measures have been implemented to address the problems caused by the fires and reduce the number of fatalities and property damage. Our project is to develop and review a fire alarm navigation system and application that uses the internet of things. Fire alarm systems are designed to warn people about fires in advance so that they can evacuate the fire-affected area and take immediate action to control the fire. There will be a GPS module, a flame sensor, a smoke sensor, buzzers, LEDs, and a GSM module to ensure early notification to authorities and fire stations. The aim is to reduce the loss of lives and property. A questionnaire was designed to conduct a brief survey in a multinational sports production company in Sialkot, Pakistan, regarding the IoT fire alarm navigation system. Besides installing the system in the factory, we compare the results with fire incident response time with and without this system at rescue 1122 fire head station.

Knowing that fire detection and alarm systems warn about fire outbreaks and allow action to be taken before conditions become out of control, they serve a very important purpose. All systems are designed to protect lives and property, so it is the designers' responsibility to understand all the possible aspects of fire risk and fire. Every fire alarm system must be designed for each building according to its specifications. **Our goal will be to develop a system that facilitates and enables firefighters to perform faster than ever.** This system is used to notify the fire station of where the fire is in the building. The system has never been used in Pakistan till now. This system can also be used in emergencies to reduce the time taken, such as fire incidents, and may reduce injury or mortality rates among fire victims. The system has been named **“IOT-based fire alarm system.”**

2.LITERATURE SURVEY:

Title: Urban Fire Risk Evaluation Based on 2-tuple AHP—Taking the8th Division with Shihezi City for Example

Author: CaihongYin; Kaixuan Qi; Kunze Li; Qiangling Duan; Lijing Gao; Jinhua Sun

Published in:2019

Abstract:

The evaluation of urban fire risk was an important gist of scientific and effective urban fire fighting management, planned and constructed. This study, took the 8th division with Shihezi city (Shi-City) as an example, an evaluation index system of urban fire risk was first built through analyzing the influential factors of fire risk in urban areas, which contained four first- class indexes and twenty-two second-class indexes. Then, to overcome the weaknesses of the analytic hierarchy process (AHP), 2-tuple fuzzy linguistic representation model was incorporated into AHP to calculate the weights of indexes. After that, an urban fire risk evaluation model was proposed. Finally, the developed model was applied into the fire risk evaluation of Shi-City and the fire risk rating of Shi-City was derived as slightly higher than medium, which offered significant guidance for fire control and safety management.

Title: Application of PHM Technology in the Design of Tank Fire Control System

Author: Jing Xu ; Yang Lei; Bin Liu; Chao Ji; Lijun Nan

Published in:2018

Abstract:

Combined with the process of Prognostics Health Management (PHM), the technology and application of armored vehicle fire control system PHM were discussed. The architecture of the health management system for tank fire control system was researched. According to the information characteristics of tank fire control system, the dual redundant bus transmission technology of FLEXRAY and CAN was applied, and the corresponding software and hardware systems were designed. Through the vehicle test, it was proved that the health management system will be effective for locating the fault, comparing the aim and assisting the soldier training. The data and video collected by this system were convenient for both maintenance and further study as the basic data.

Title: Fire Safety Management in Transportation of Municipal Wastes with the Use of Geographic Information System

Author: O.P.Savoshinsky ;A.A.Zakharova ; A.V.Pak

Published in: 2018

Abstract:

Fire safety management is one of the main tasks in the field of waste safety. The transportation of municipal waste was a complex management task that requires a highly skilled decision maker. The current management technique is based on the approach to the construction of systems based on the analysis, by assessing the set of initial factors, which does not allow to achieve the management goal. The proposed approach based on synthesis was devoid of this drawback. The application of the system was shown by the example of the use of geoinformation systems to the problem of fire safety in the transportation of municipal waste.

Title: Fire incidents Management System in the city of Manil athrough Geo-Mapping

Author: Maricor Y. Ingal; Ralph Louisse T. Tolentino; Mico J. Valencia; Francis F. Balahadia ;ArleneR. Caballero

Published in:2016

Abstract:

Fires had become a concern in recent years in the city of Manila, posing a threat to the entire community. Manila Fire District was facing problems in their internal transactions between different sub-stations. The study served as an automated fire incidents management system that can provide a chart and a summary based on the input data of each sub-station and can provide a map of all the fire incidents through geo-mapping in districts of Manila. This study, Manila Fire District implemented appropriate programs and lead awareness campaign to the community to help lessen fire incidents and mitigated its damages.

Title: Fire Safety Management Information System Design for KeySocial Organizations

Author: XuFang; ZhangDi; WangJun

Published in: 2014

Abstract:

The design and implementation of the fire safety management information systems of the networked key organizations and units, provide information sharing and services on fire- fighting facilities' operating conditions, fire alarm information, and fire management informationto the networked users, fire maintenance enterprises, and the fire supervision and administrative authorities so as to improve the fire safety management efficiency for these organizations and units, offered a scientific tool to the organizations to improve their fire safety management level, extended the functions of fire remote monitoring control system, and promoted fire prevention and controlled capability of the whole community.

Title: Discussion of Society Fire-Fighting Safety Management Internet of Things Technology System

Author: WangJun; ZhangDi; LiuMeng; XuFang; SuiHu-Lin; YangShu-Feng

Published in:2014

Abstract:

IOT is regarded as another information industry wave following computer, Internet and mobilecommunication network, and had become one of strategic dominant positions of new economic and technological development all over the world. The society fire-fighting safety management was an important application field of Internet of Things (IOT)technology. This paper combines application features of IOT technology according to fire-fighting business requirement to discuss the fire-fighting IOT systematic frame, plan society fire-fighting safety management IOT technology system, and proposed priority development points of society fire- fighting safety management IOT technology, here by provided reference for technology research and development of IOT technology in society fire-fighting safety management field.

Title: Automatic fire alarm and fire control linkage system in intelligent buildings

Author: WangSuli; LiuGanlai

Published in: 2010

Abstract:

This paper described a comprehensive program of an office building intelligent systems Fire Control Linkage System subsystem design, At the same time, it described the following: the idea of the system design, 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.

Title: **A System design of the Tahe's forest -Fire prevention Management System**

Author: XindanGao; NihongWang; JunLi

Published in:2010

Abstract:

This article paper aimed to introduces how a system was designed for Tahe's forest-fire- prevention management in Northeast China after a brief introduction to the overall functional characteristics, the overall function flow chart and the operating environment of the forest -fire - prevention management system. firstly, and then This system design consists of seven function modules, which were geographic information system module, fire-risk each function module of the system in detail, including geographic information system module, fire forecast module, forest -fire -alarm receiving module, blazes fire-put-out-aided decision-making module, forest- fire-put-out troops sending module, loss evaluation module, forest -fire -prevention office and information management module and as well as GPS real-time monitoring module. Among all modules, the geographic information system module was the core of those fire -prevention - management system, and other various modules were carried out various functions through links with the core module, based on its function, realized link. In conclusion, that this paper summarized the whole system design work done by this paper and as well as the advantages and disadvantages of this system.

Title: **Building fire rescue with evacuation management information system and its application**

Author: XuTao;MaoGuozhu;LiXin;ZhaoLin

Published in:2009

Abstract:

Building Fire Rescue with Evacuation Management Information System (BFREMIS) was established. And the evacuation model of BFREMIS was analyzed and presented in this paper. Based on the constructed network model, the evacuation of the teaching building inthe university was analyzed by using the software EVACNET4. The analysis items included:the total evacuation time, the floor clear time, evacuation bottleneck, and the visual path of the evacuation on MAPGIS platform. BFREMIS was valuable in building safety assessment and building fire rescue.

Title: **Forest Fire Management at Aggtelek National Park Integrated Vegetation Fire Management Program from Hungary**

Author: Agoston Restas

Published in:2006

Abstract:

Szendro Fire Department is located in the northeastern part of Hungary. The main task was to fight against wild fire and mitigate the impact of fire at the Aggtelek National Park-which belongs to the UNESCO World Heritage list. In 2004 the Fire Department started a project named Integrated Vegetation Fire Management (IVFM). The IVFM consist of two main parts: Peripheries and Modules. The Modules are: Tower based environment monitoring and fire detection system, Mobile command control unit and Static and dynamic decision support system. The Tower based environment monitoring and fire detection system addressed the Fire Department by hot information. The Static and dynamic decision supported system was based on robot reconnaissance aircraft (UAV-RRA)- dynamic parts and the GIS - static parts. The data supplied by the robot reconnaissance aircraft as combined with the GIS based fuel mode land other information to predict the fire activity.The environment monitoring and fire detection system and the Dynamic part (UAV-RRA) of Decision support system based on remote sensing.

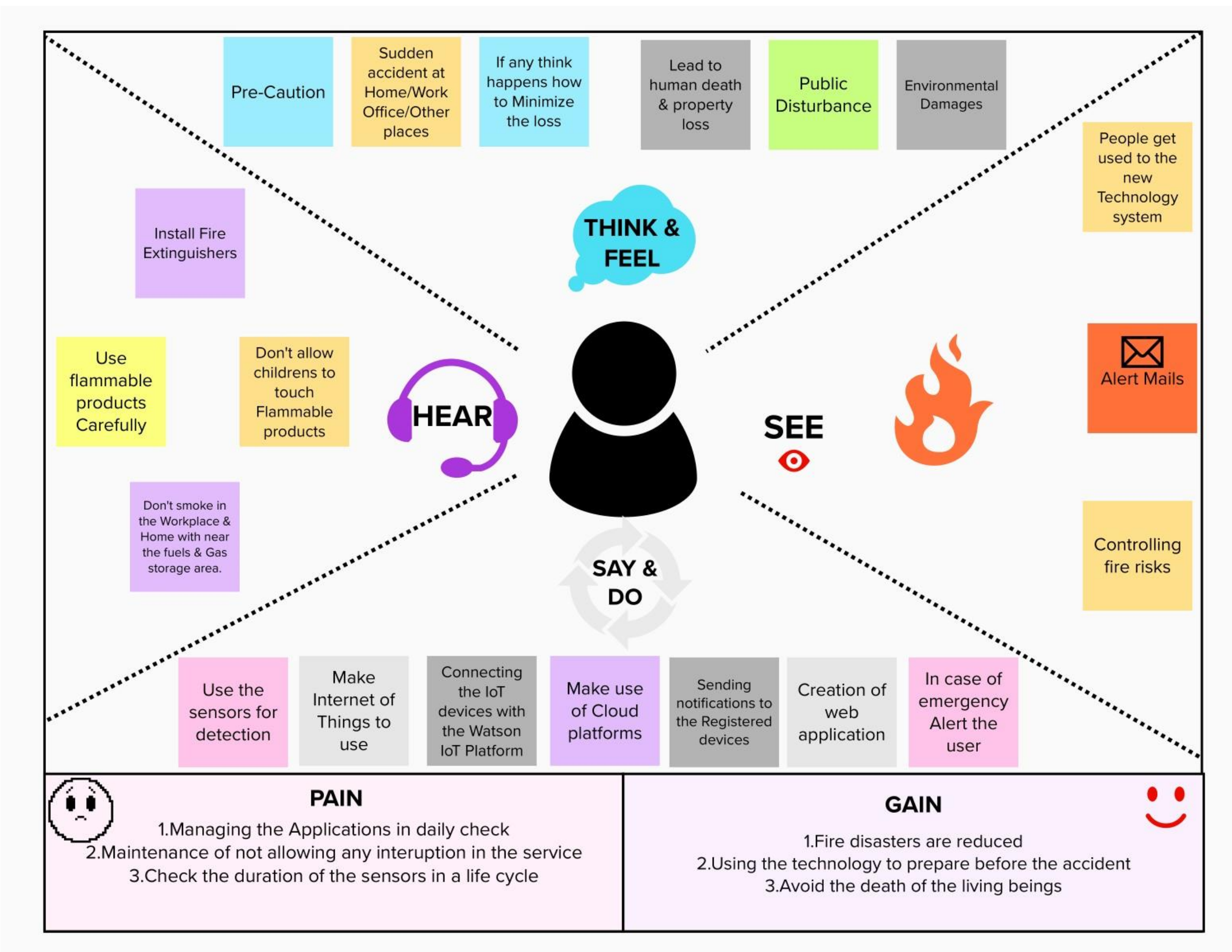
2.3 Problem Statement:

I am	A customer	I need a quality and innovative firemanagement system
I am trying to	Ensure the fire safety in myindustry/house	I can avoid the fire explosion in myIndustry and keep all my products safe from the fire spread
But	I want to handle it easily	By doing fire safety in the form ofautomation using the recent easy technologies
Because	The fire explosion should be automatically detected without anyhuman monitoring	The fire explosion at the ignition stage can be detected and controlled
Which makes me feel	That my Industry and my workers aresafe	It saves the Industry and lives of theworkers

3.Ideation and Proposed solution

3.1 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.



Link:

https://app.mural.co/t/ibmproject6843/m/ibmproject6843/1665900922592/ad9501be3a4851ad51700b 18ea3e3eb88ea91c02?sender=uabe2e895d86a60b502379287

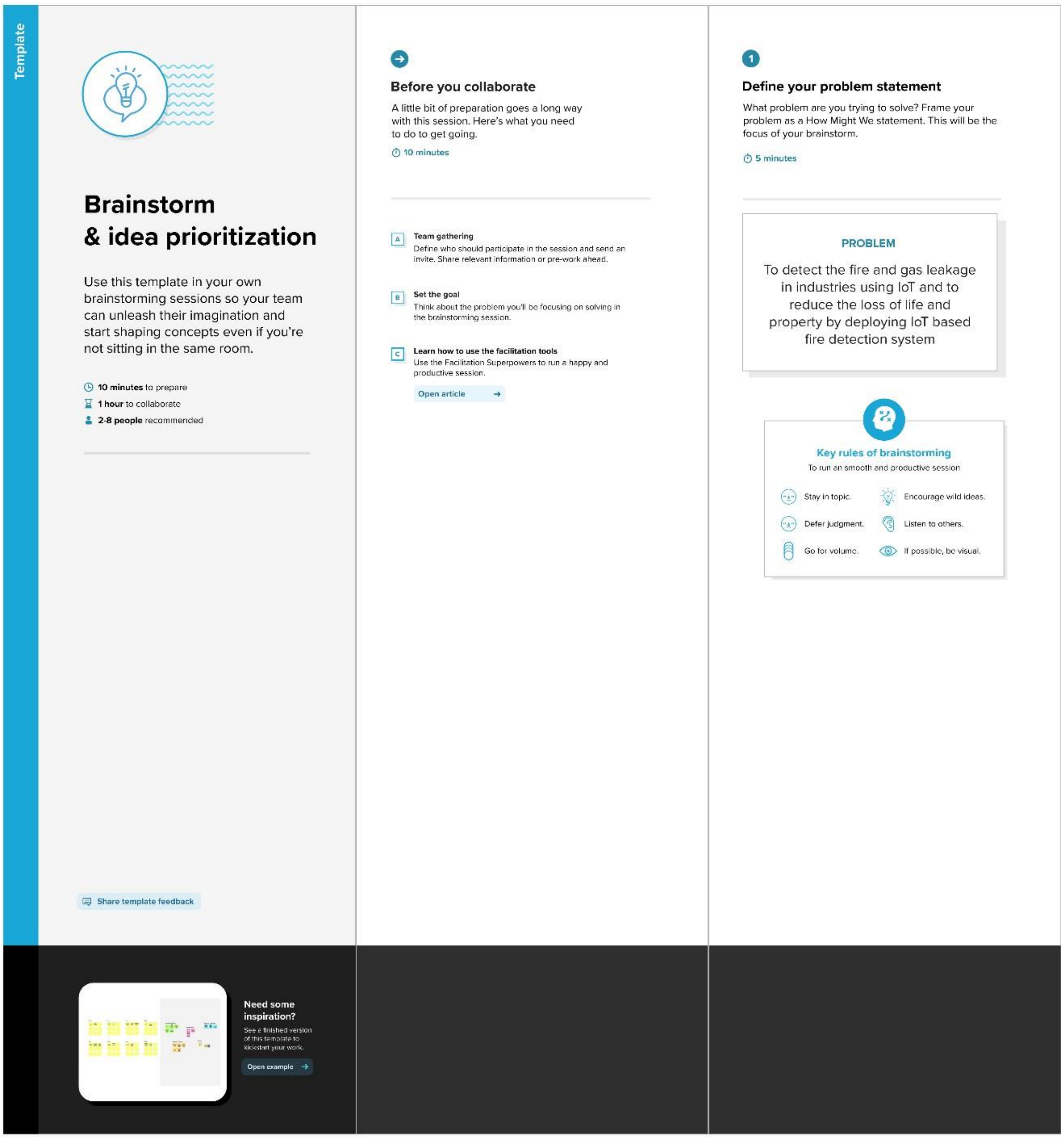
3.2 Ideation and Brainstorming

step 1: Team Gathering, Collaboration and Select the Problem Statement

Team was gathered in mural app for collaboration.

step 2: Then create Brainstorm, Idea Listing and Grouping

step 3: Idea Prioritization



2

Brainstorm

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

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

SELVAGANAPATHI M

- Voice Alert Systems
- Use multiple sensors for Detection

ARAVIND T

- Limit risk with remote monitoring
- Prior Indication

LAIYANCEMARY A

- Sprinkler water flow switch
- Smoke detection system

SATHISH I

- Compact in Size
- Using this techniques the notifications are sended to the registered device

3

Group ideas

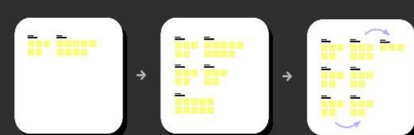
Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

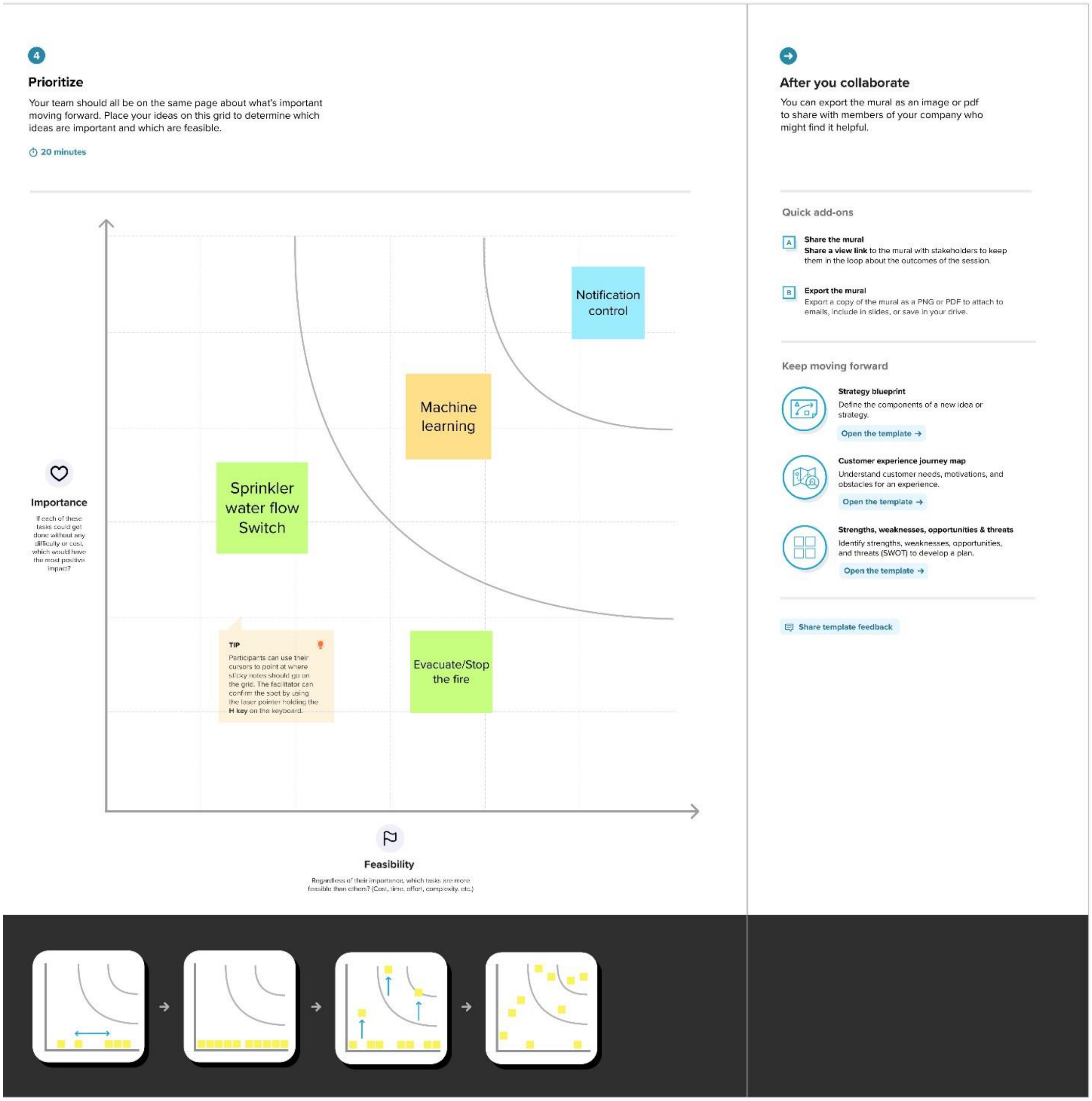
20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

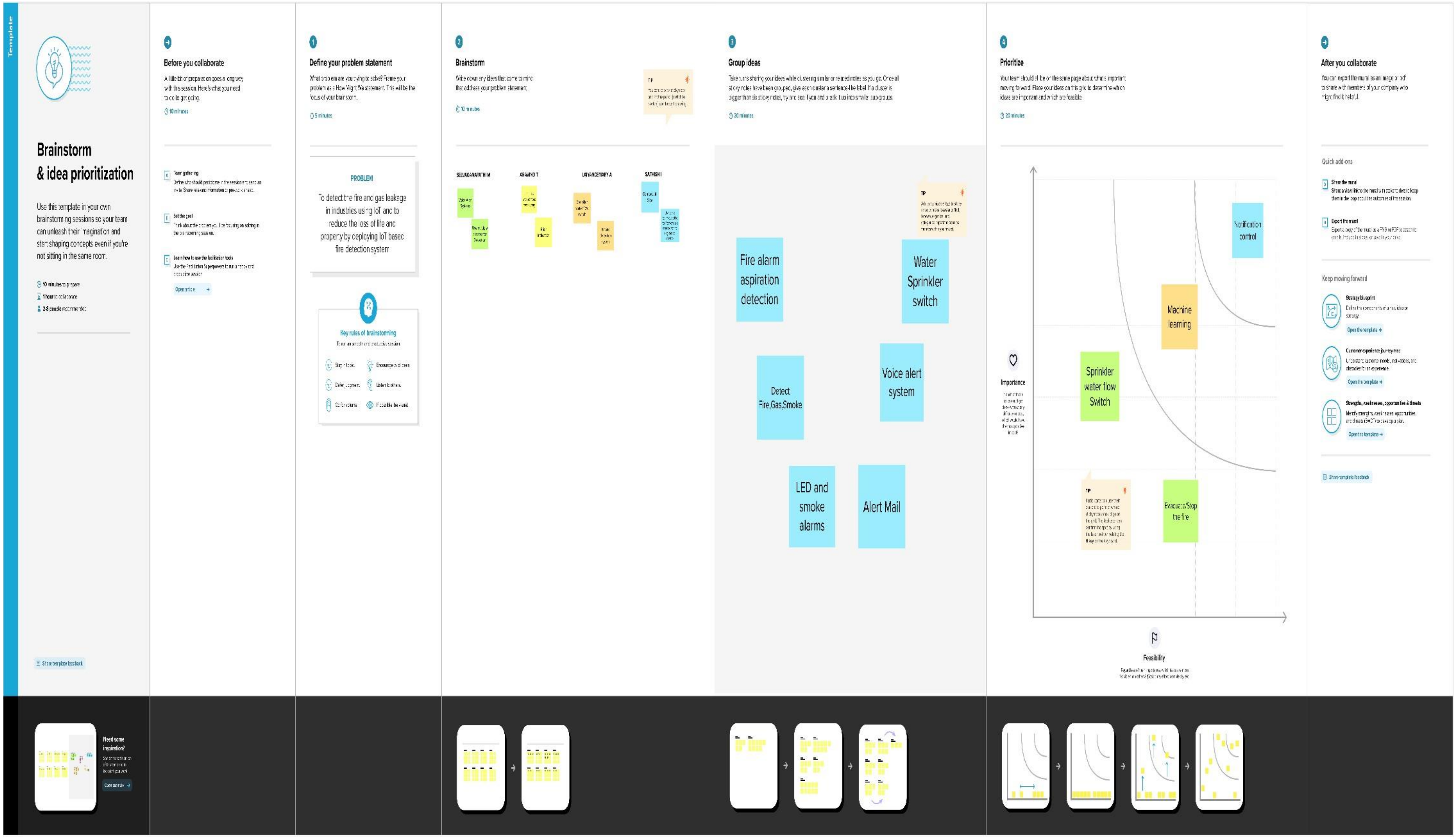
- Fire alarm aspiration detection
- Water Sprinkler switch
- Detect Fire,Gas,Smoke
- Voice alert system
- LED and smoke alarms
- Alert Mail





Link:

https://app.mural.co/t/ibmproject6843/m/ibmproject6843/1665929936366/30fb44145f0ed614aefe985 8c82d4e71a748399a?sender=uabe2e895d86a60b502379287



Template
Brainstorm
& idea prioritization
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.
Before you collaborate
Define your problem statement
PROBLEM
To detect the fire and gas leakage in industries using IoT and to reduce the loss of life and property by deploying IoT based fire detection system
Key rules of brainstorming
Brainstorm
Group ideas
Fire alarm aspiration detection
Water Sprinkler switch
Detect Fire,Gas,Smoke
Voice alert system
LED and smoke alarms
Alert Mail
Prioritize
Importance
Feasibility
Verification control
Machine learning
Sprinkler water flow Switch
Evacuate/Stop the fire
TIP
After you collaborate
Quick add-ons
Keep moving forward
Need some inspiration?

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	➢ The smart fire management system includes aGas sensor, Flame sensor and temperature sensors to detect any changes in the environment. ➢ Based on the temperature readings and if any Gases are present the exhaust fans are powered ON. ➢ If any flame is detected the sprinklers will be switched on automatically. ➢ Emergency alerts are notified to the authorities and Fire station ➢ Access the needs of fire breakout buildings.
2.	Idea/Solution description	➢ Intelligent fire management system is otherwise known as addressable fire alarms, intelligent control systems are more sophisticated than conventional firealarm systems and are able to provide an exact location of the event. ➢ IOT device ➢ Temperature sensor by IOT ➢ Gas sensor ➢ Flame sensor

3.	Novelty/Uniqueness	➢ Collabarotion work of IOT platform and sensors ➢ Prior information about fire breakout can be intimated ➢ By using IOT based sensor addressable systems provide a greater level of fire safety because they allow fire fighters to respond more quickly and effectively by pin pointing the exact location of a fire ina building.
4.	Social Impact/Customer Satisfaction	➢ It serves of value to users, Fire detection systems increase response times, as they are able to alert the correct people. In order to extinguish the fire.
5.	Business Model (Revenue Model)	➢ Reduces the amount of damage to the property. ➢ Fire detection systems can be connected to sprinklers that will automatically respond, When a fire is detected.
6.	Scalability of the Solution	➢ Python ➢ IOT Application Development ➢ IBM Cloud ➢ IBM Watson

3.4 Proposed Solution Fit

Project Title: Industry-specific intelligent fire management system **Project Design Phase-I - Solution Fit Template** **Team ID: PNT2022TMID45189**

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Industry members as well as others

6. CUSTOMER CONSTRAINTS CC

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 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 J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

We are solving the problem of fire spread by automatically detecting the fire at the ignition stage and stop 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 can use to stop the fire without the help of fire service.

7. BEHAVIOUR 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 J&P, tap into BE, understand RC

3. TRIGGERS TR

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 BEHAVIOUR CH

8.1 ONLINE

Notifications send can be accessed.

8.2 OFFLINE

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

4.REQUIREMENT ANALYSIS

4.1 Functional Requirements:

Following are the functional requirements of the proposed solution.

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	User Confirmation	Verification via Email or OTP
FR-3	User Login	Login through website or App using there spective user name 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 24hours
FR-7	User Data Sync	API interface to increase to in voice system

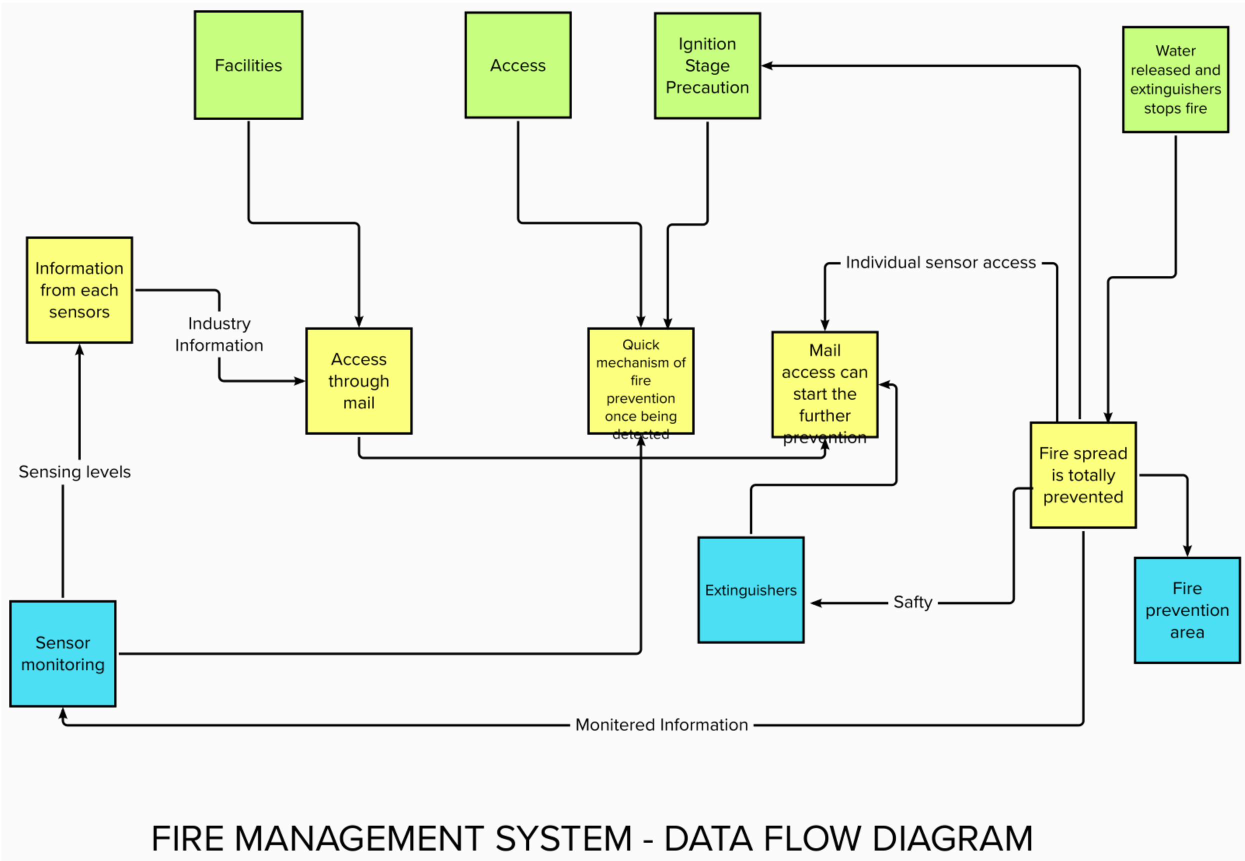
4.2 Non-functional Requirements:

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

FR 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 rollback 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 an VoLTE mobile connection.
NFR-5	**Availability**	New module deployment must not impact frontpage, product pages, and check out pages availability and mustn't take longer than one hour.
NFR-6	**Scalability**	We can increase scalability by adding memory, servers, or disk space. On the other hand, we can compress data, use optimizing algorithms.

5. PROJECT DESIGN

5.1 Dataflow Diagram

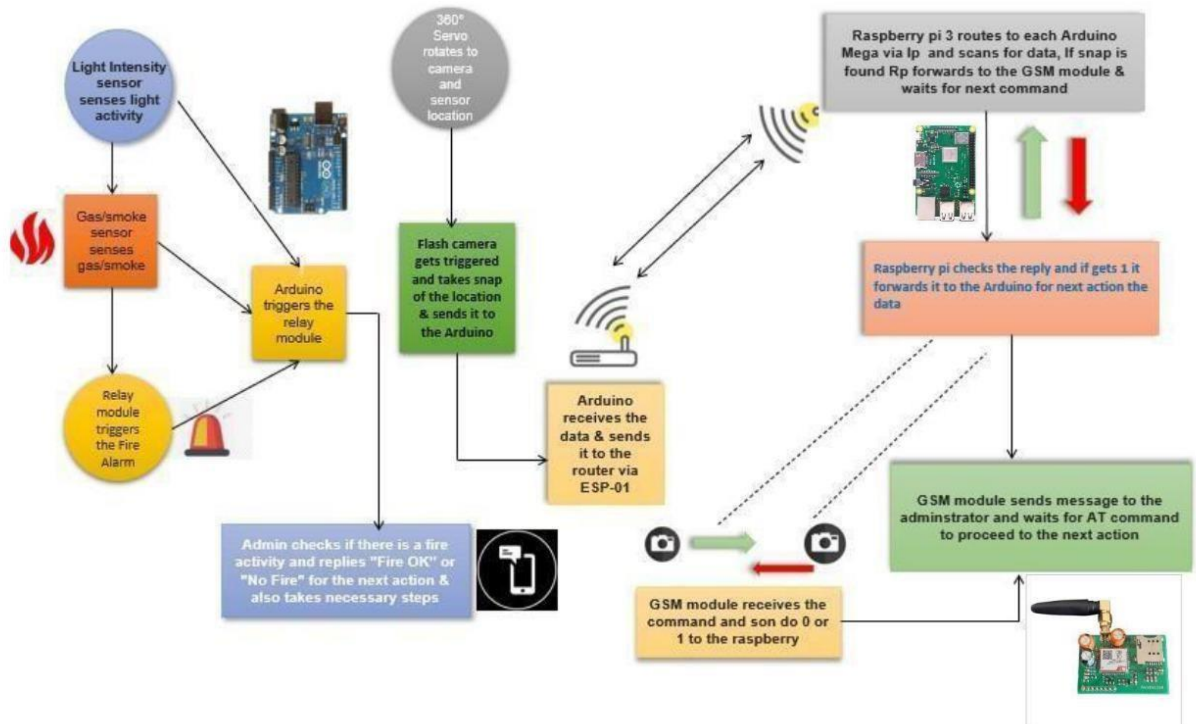


Link:

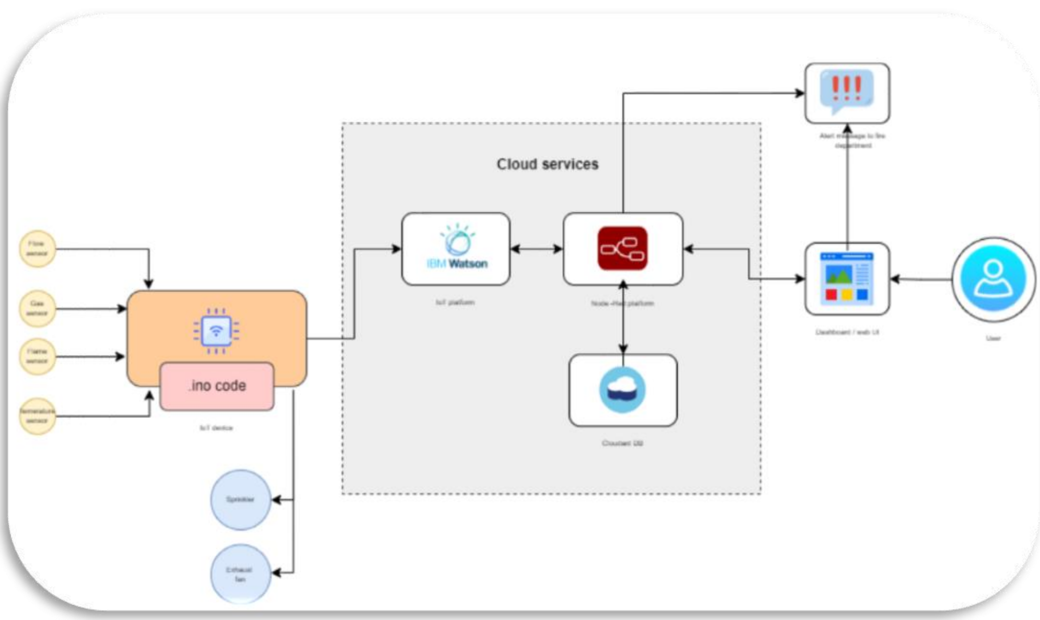
https://app.mural.co/t/ibmproject6843/m/ibmproject6843/1666002238550/9709c4c52155c75e06ec9c632c9e8dbe18f4a67d?sender=uabe2e895d86a60b502379287

5.2 Solution and Technical architecture

Solution Architecture



Technical Architecture



5.3 User Stories

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 PLANNING & SCHEDULING

6.1 Sprint planning and estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration & Login	USN-1	As a user, I can register for the application by entering my email, password, and confirmingmy password.	6	High	Selvaganapathi, Aravind
		USN-2	As a user, I will receive confirmation email once I have registered for the application	7	High	Laiyancemary, Sathish
		USN-3	As a user, I can log into the application by entering email & password	7	High	Selvaganapathi, Laiyancemary
Sprint-2	Sensor & Actuators	USN-4	In industry, sensor sense the fire and smoke.	10	High	Aravind, Sathish
		USN-5	If the sensor detected the fire, next step is extinguishing the fire with the help of Sprinkler.	10	High	Selvaganapathi, Sathish

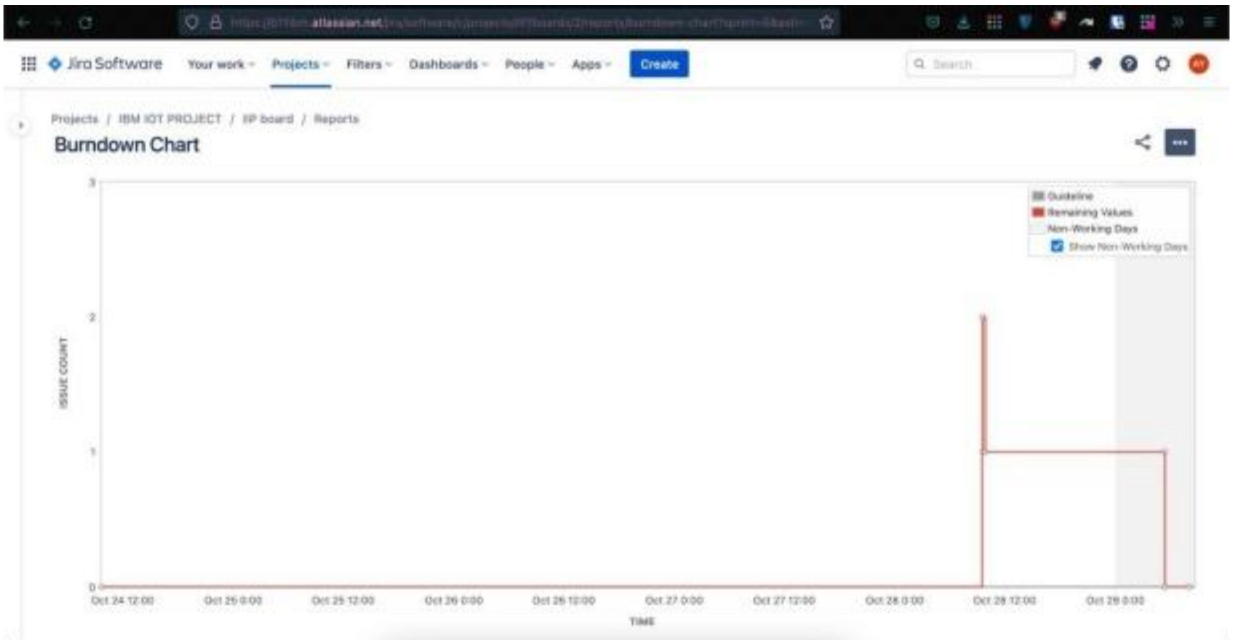
Sprint-3	Cloud	All the values are stored in the cloud database.	20	High	Aravind, Laiyancemary
Sprint-4	Siren & Event management	If the fire is detected, employee should Evacuate by the intimation by Siren/Buzzer.	10	High	Selvaganapathi, Aravind
		Notification message will be sent to the fire Department, proprietor.	10	High	Sathish, Laiyancemary

6.2 Sprint Delivery schedule

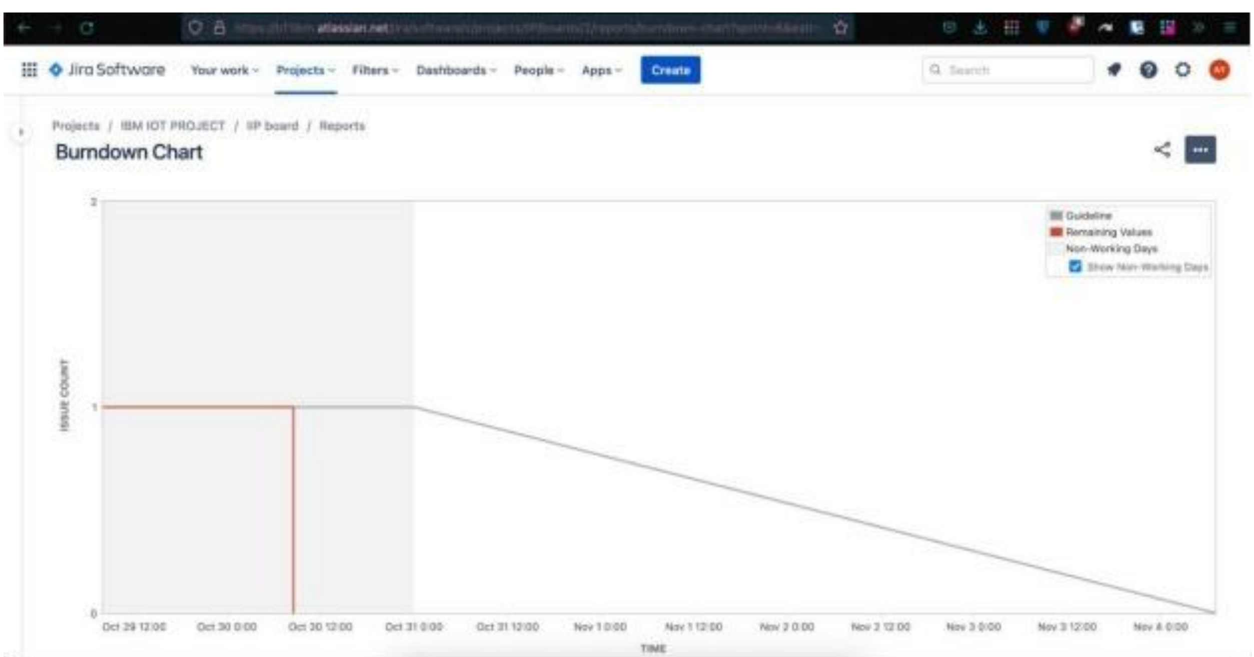
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022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

6.3 Reports from JIRA

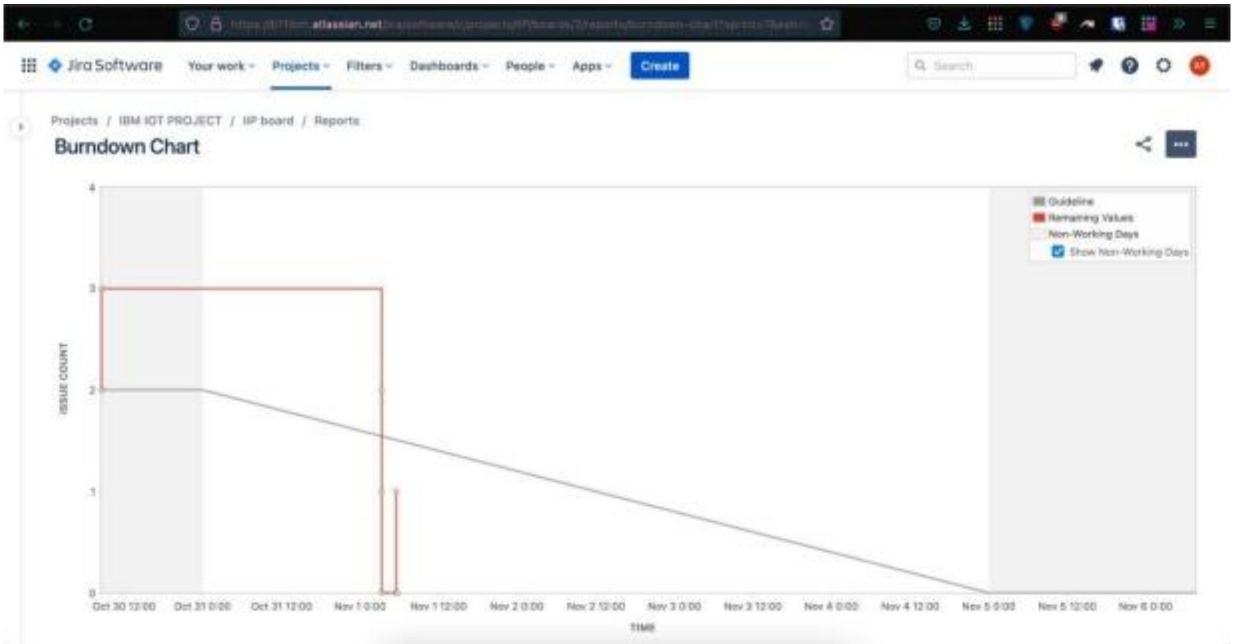
Sprint 1



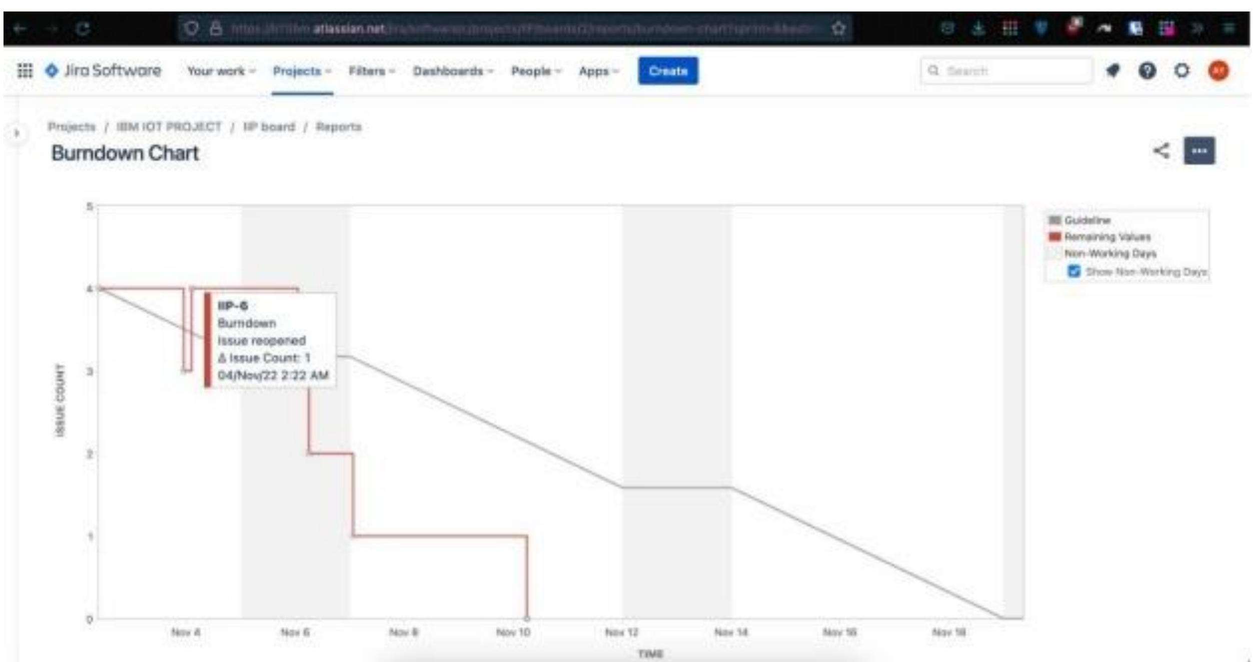
Sprint 2



Sprint 3



Sprint 4



7.Coding and Solutioning

7.1 Feature 1(Tinkercad – Fire Alarm System)

```
long randNumber;

float temp;

float vout;

float vout1;

int LED = 13;

int gasSensor;

int piezo = 7;

void setup()

{

pinMode(A0,INPUT);

pinMode(A1, INPUT);

pinMode(LED,OUTPUT);

pinMode(piezo,OUTPUT);

Serial.begin(9600);

randomSeed(analogRead(0));

}

void loop()

{
```

```
vout=analogRead(A1);

vout1=(vout/1023)*5000;

temp=(vout1-500)/10;

gasSensor=analogRead(A0);

if (temp>=70)

{

digitalWrite(LED,HIGH);

Serial.print("Fan is ON Automatically When HIGH Temperature in Degree(Temp>=70C) ");

Serial.print("\n");

}

else

{

digitalWrite(LED,LOW);

}

if (gasSensor>=100)

{

digitalWrite(piezo,HIGH);

Serial.print("Fan is ON Automatically When HIGH Level of Gas is detected(Gas=>100ppm) ");

Serial.print("\n");

}

else
```

```
{
digitalWrite(piezo,LOW);
}
{
if (randNumber>=40)
{
Serial.print("Sprinkler is ON Automatically When detected the flame limit reached the value>=40");
Serial.print("\n");
}
else
{//nothing
}
}
Serial.print("\n");
Serial.print("Temperature in Degree(C)= ");
Serial.print(" ");
Serial.print(temp);
Serial.print("\n");
Serial.print("Level of Gas(ppm)= ");
Serial.print(" ");
Serial.print(gasSensor);
```

Serial.print("\n");

randNumber = random(80);

Serial.print("Flamesensor(cd)= ");

Serial.print(" ");

Serial.println(randNumber);

Serial.print("\n");

Serial.println();

delay(1000);

}

Explanation

- **Fan is ON Automatically When HIGH Temperature in Degree(Temp>=70C)**
- **Fan is ON Automatically When HIGH Level of Gas is detected(Gas=>100ppm)**
- **Sprinkler is ON Automatically When detected the flame limit reached the value>=40**

(Link:

https://www.tinkercad.com/things/bbEQTXiq982)

7.2 Feature 2(WOKWi – Fire Alarm System)

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "DHT.h"// Library for dht11
#define DHTPIN 15     // what pin we're connected to
```

```
#define DHTTYPE DHT22   // define type of sensor DHT 11
#define LED 2

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 "i3869j"//IBM ORGANITION ID
#define DEVICE_TYPE "abcd"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"     //Token
String data3;
float t, g, f;


//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event
perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT command
type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client 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();
  pinMode(LED,OUTPUT);
  delay(10);
  Serial.println();
  randomSeed(analogRead(0));
  wificonnect();
  mqttconnect();
```

```
}

void loop()// Recursive Function
{

  t = dht.readTemperature();
  g = random(80);
  f = random(80);
  Serial.print("Temperature:");
  Serial.println(t);
  Serial.print("Gas:");
  Serial.println(g);
  Serial.print("Flame:");
  Serial.println(f);

  PublishData(t, g, f);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
  {
    if(t>=60){
      Serial.print("Fan is ON Automatically When HIGH Temperature in
Degree(Temp>=60C) ");//C-Celsius
      Serial.print("\n");
    }
    else{
      Serial.println("Temperature level is normal");
      }
    }
    {
    if(g>=40){
      Serial.print("Fan is ON Automatically When HIGH Gas in (Gas>=40ppm)
");//ppm-Patrs Per Million
      Serial.print("\n");
    }
    else{
      Serial.println("Gas level is normal");
      }
    }  {
    if(f>=30){
      Serial.print("Sprinkler is ON Automatically When detected the flame limit
reached the value>=30cd");//cd-candela
      Serial.print("\n");
    }
```

```
    else{
      Serial.println("No flame is Detected");
      }
    }
}



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

void PublishData(float temp, float gas, float flame) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String payload = "{\"Temperature\":";
  payload += temp;
  payload += "C," "\"Gas\":";
  payload += gas;
  payload += "ppm," "\"Flame\":";
  payload += flame;
  payload += "cd}";


  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");
  }

}


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);//passing the wifi credentials to establish
the connection
  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, unsigned int payloadLength)
{

  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);
  for (int i = 0; i < payloadLength; i++) {
    //Serial.print((char)payload[i]);
    data3 += (char)payload[i];
  }
  Serial.println("data: "+ data3);
  if(data3=="lighton")
```

```
  {
Serial.println(data3);
digitalWrite(LED,HIGH);
  }
  else
  {
Serial.println(data3);
digitalWrite(LED,LOW);
  }
data3="";
}
```

Explanation:

- **Fan is ON Automatically When HIGH Temperature in Degree(Temp>=60C)**
- **Fan is ON Automatically When HIGH Gas in (Gas>=40ppm)**
- **Sprinkler is ON Automatically When detected the flame limit reached the value>=30cd**

Link:

https://wokwi.com/projects/348343557862457939

Feature 3(Python – Device(Laptop))

```
import time

import sys

import ibmiotf.application

import ibmiotf.device

import random




#Provide your IBM Watson Device Credentials

organization = "g7vqi6"

deviceType = "abcd"
```

```
deviceId = "12"

authMethod = "token"

authToken = "12345678"


# Initialize GPIO



def myCommandCallback(cmd):

   print("Command received: %s" % cmd.data['command'])

   status=cmd.data['command']

   if status=="fanon":

      print ("Fan is on")

   else :

      print ("Fan is off")

   if status=="sprinkleron":

      print ("sprinkler is on")

   else :

      print ("sprinkler is off")

   #print(cmd)



try:
```

```
        deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": 
authMethod, "auth-token": authToken}

        deviceCli = ibmiotf.device.Client(deviceOptions)

        #.............................................


except Exception as e:

        print("Caught exception connecting device: %s" % str(e))

        sys.exit()


# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type 
"greeting" 10 times

deviceCli.connect()


while True:

    #Get Sensor Data from DHT11


    temperature=random.randint(0,100)

    gas=str(random.randint(0,100))

    flame=str(random.randint(0,100))


    data = { 'temperature' : temperature, 'gas': gas, 'flame' : flame }

    #print data

    def myOnPublishCallback():

      print ("Published Temperature = %s C" % temperature, "gas = %s ppm" % gas,"flame = %s 
cd" % flame, "to IBM Watson")
```

```
    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)

    if not success:

      print("Not connected to IoTF")

    time.sleep(10)



    deviceCli.commandCallback = myCommandCallback



# Disconnect the device and application from the cloud

deviceCli.disconnect()
```

Explanation

- The action taken by the user is received as a command and stored in a buffer.
- The event in the device is done according to the command
- It checks for a secret encrypted pin for performing that event

8 TESTING

8.1 Testcases

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)
Sensor_OO1	Functional	Microcontroller	Sensor data is properly taken	The connections to the circuit	1.Open the simulator in wokwi.	Random values generated ,	Get the values and print it in the	Working as	Pass		N
Sensor_OO2	Functional	Microcontroller	Sensor data is parsed as json	The microcontroller should	1.Open the simulator in wokwi.	Random values generated ,	Get the values and print it in the	Working as	Pass		N
Work_001	Functional	Microcontroller	To check for fake alarm	The sensor values are taken	1.Simulate the device(do a practical	Random values generated ,	Accident status is properly updated	Working as	Pass		N
Work_002	Functional	Microcontroller and	The data should be sent to IBM	The device setup is completed	1.Start the simulation in wokwi.	Random values generated ,	The values are shown in recent	Working as	Pass		N
Work_OO3	Functional	Node-red	The data should be sent to	The necessary packages	1.Login to node red editor	values got from the iot	The debug area should show the	Working as	Pass		N
Work_OO4	Functional	Node-red	Verify that the json data is parsed	A configured node-red with	1.Login to node red editor	values got from the iot	the debug menu shows the output	Working as	Pass		N
Database_001	Storage	Cloudant	The received data is stored in database in a key value pair	The node red is connected with cloudant node	1.login to cloudant dashboard. 2.create new database. 3. connect the database with node red and then give the database name in required field	values got from the iot device	After sending the data the data is stored in cloudant	Working as expected	Pass		N
SMS_001	API	sms API	The sms is sent when there is fire alert	The node red should be configured to send a post request	1.Simualte the fire in the simulator(if real hardware is used real fire is used). 2.or click the sent alert button in	"Fire alert at xyz industries Hurry" And the trigger inputs	sms receiving to the given phonenum	Working as expected	Pass		N
Work_005	Functional	UI	Even at times of emergency sometimes manual control is required	the dashboard interaction elements is connected to the node-red	1. in the dashboard enter the correct pin 2.click the action to be done	The action by user	manual command system works only	Working as expected	Pass		N
Auth_001	Functional	UI	Verify that the correct pin is entered	text filed is given in dashboard to enter pin	1.The correct pin is entered 2.then necessary action is required	1234	command is sent successfull	working as expected	Pass		N
Auth_002	Functional	UI	Verify that it handles when wrong pin is entered	text filed is given in dashboard to enter pin	1.The correct pin is entered 2.then necessary action is required	141324 63363 1 001 fds	Show a message that the entered pin is wrong	Working as expected	Pass		N
SMS_002	Functional	Microcontroller	Verify that the message is not sent continuously when there is fire it sends a message then waits for 10 minutes even after that if the fire exists it sends again	the sms funtionality should be implemented	1.Simulate a fire accident scenario 2.or click the send alert button on the dashboard 3.wait for the message to be sent	the event is simulated or triggered	The service should not spam continuous messages to authorities as fire won't be down within fraction of seconds	Working as expected	Pass		N

8.2 UAT

Defect analysis

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

9 Result

9.1 Performance Metrics

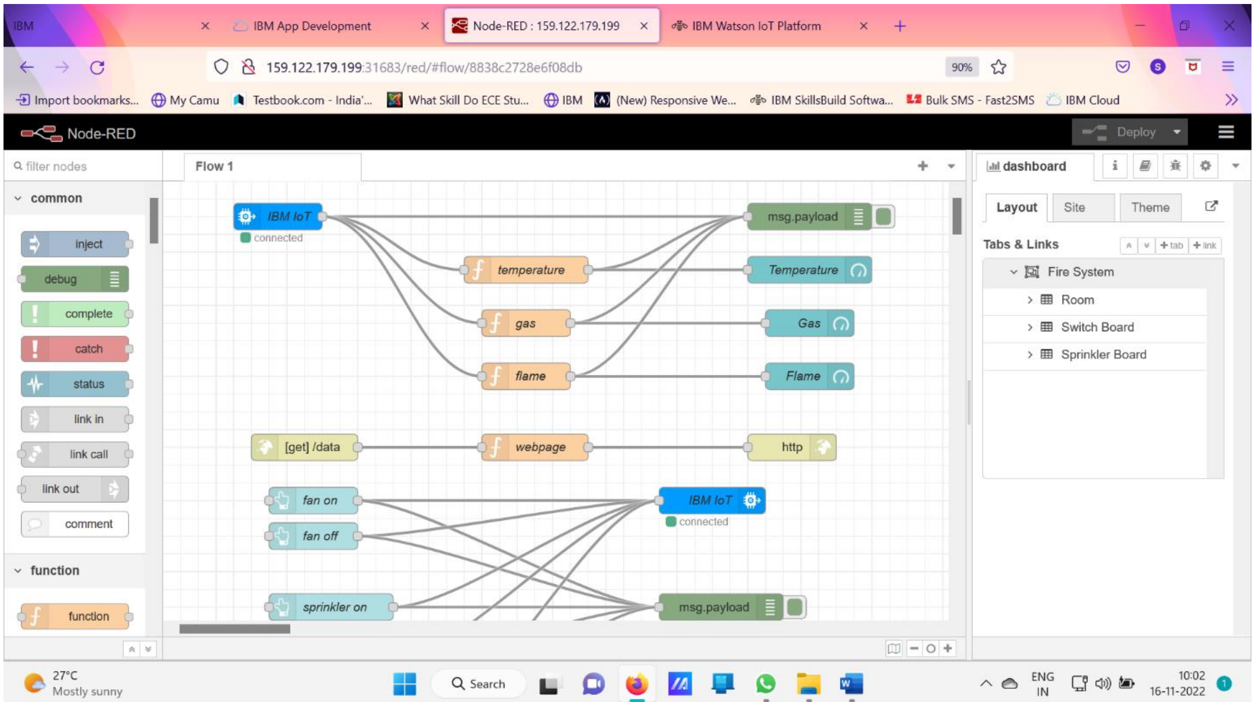
CPU Usage

The micro version of c++/python is make the best use of the CPU. For every loop the program runs in O(1) 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:

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.

Garbage collection:

In the server-side garbage collection is done by the Node framework. In the IoT device ,c++/python does not have any garbage collection features . But it is not necessary in this scenario as the memory is used again for storing the data. Any dangling pointer or poorly handled address space is not allocated.

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.

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:

//WOKWi Platform

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "DHT.h"// Library for dht11
#define DHTPIN 15     // what pin we're connected to
#define DHTTYPE DHT22   // define type of sensor DHT 11
#define LED 2

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 "i3869j"//IBM ORGANITION ID
#define DEVICE_TYPE "abcd"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"     //Token
String data3;
float t, g, f;


//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client 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();
  pinMode(LED,OUTPUT);
  delay(10);
  Serial.println();
  randomSeed(analogRead(0));
  wificonnect();
  mqttconnect();
}

void loop()// Recursive Function
{

  t = dht.readTemperature();
  g = random(80);
  f = random(80);
  Serial.print("Temperature:");
  Serial.println(t);
  Serial.print("Gas:");
  Serial.println(g);
  Serial.print("Flame:");
  Serial.println(f);

  PublishData(t, g, f);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
  {
    if(t>=60){
      Serial.print("Fan is ON Automatically When HIGH Temperature in
Degree(Temp>=60C) ");//C-Celsius
      Serial.print("\n");
    }
    else{
      Serial.println("Temperature level is normal");
      }
    }
    {
    if(g>=40){
      Serial.print("Fan is ON Automatically When HIGH Gas in (Gas>=40ppm)
");//ppm-Patrs Per Million
      Serial.print("\n");
    }
```

```
    else{
      Serial.println("Gas level is normal");
      }
    }  {
    if(f>=30){
      Serial.print("Sprinkler is ON Automatically When detected the flame limit
reached the value>=30cd");//cd-candela
      Serial.print("\n");
    }
    else{
      Serial.println("No flame is Detected");
      }
    }
}



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

void PublishData(float temp, float gas, float flame) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String payload = "{\"Temperature\":";
  payload += temp;
  payload += "C," "\"Gas\":";
  payload += gas;
  payload += "ppm," "\"Flame\":";
  payload += flame;
  payload += "cd}";


  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");
  }
```

```
}


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);//passing the wifi credentials to establish
the connection
  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, unsigned int payloadLength)
{
```

```
  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);
  for (int i = 0; i < payloadLength; i++) {
    //Serial.print((char)payload[i]);
    data3 += (char)payload[i];
  }
  Serial.println("data: "+ data3);
  if(data3=="lighton")
  {
Serial.println(data3);
digitalWrite(LED,HIGH);
  }
  else
  {
Serial.println(data3);
digitalWrite(LED,LOW);
  }
data3="";
}
```

//Python Code(Laptop as a Device)

```
import time

import sys

import ibmiotf.application

import ibmiotf.device

import random



#Provide your IBM Watson Device Credentials

organization = "g7vqi6"

deviceType = "abcd"

deviceId = "12"

authMethod = "token"

authToken = "12345678"


# Initialize GPIO
```

```
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="fanon":
        print ("Fan is on")
    else :
        print ("Fan is off")
    if status=="sprinkleron":
        print ("sprinkler is on")
    else :
        print ("sprinkler is off")
    #print(cmd)




try:
	deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": authToken}
	deviceCli = ibmiotf.device.Client(deviceOptions)
	#..............................................

except Exception as e:
	print("Caught exception connecting device: %s" % str(e))
	sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
deviceCli.connect()
```

```
while True:
        #Get Sensor Data from DHT11

        temperature=random.randint(0,100)
        gas=str(random.randint(0,100))
        flame=str(random.randint(0,100))

        data = { 'temperature' : temperature, 'gas': gas, 'flame' : flame }
        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temperature, "gas = %s ppm" % gas,"flame =
%s cd" % flame, "to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(10)

        deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

Github Link: https://github.com/IBM-EPBL/IBM-Project-44952-1660727584

Demo Video: https://drive.google.com/drive/folders/1qZOtUPHrPNIn87Qngvjtc-WTp93rfV1E?usp=sharing (Video Folder)