PROJECT REPORT

INDUSTRY SPECIFIC INTELLIGENT FIRE MANAGEMENT SYSTEM

TEAM ID:PNT2022TMID30584

ROLL NUMBER	NAME
613019104069	G.SARANYA
613019104066	S.SANDHIYA
613019104093	S.VASANTHA

CONTENT

CHAPTER-1

1. INTRODUCTION

1.1 Project Overview

The Smart fire management system includes a gas, flame, and temperature sensor to detect any environmental changes. The exhaust fans are turned on based on the temperature readings and the presence of any gases. If a flame is detected, the sprinklers will automatically activate. Emergency alerts are sent to the authorities and the Fire Station.

1.2 Purpose

- To provide a detect the status of the room using IoT devices
- To turn on sprinkler and exhaust fan when there is an accident
- To detect the flow of water
- To send and store the temperature status in a cloud storage
- To provide an easy management system on dashboard
- To provide an overview of what is happening to the user

CHAPTER-2

2. LITERATURE SURVEY

A Fire Detection System for Smart Home Based on IoT Data Analytics [1]: In this paper, an Internet of Things based Fire Detection System (FireDS-IOT) is designed to prevent people from fire by providing an alert message in the emergency. The system is designed using MQ135 (CO2), MQ-2(smog), MQ-7(CO) and DHT-11 (temperature) sensors embedded with Ardunio to get the fire event information in the surrounding more accurately.

A Wireless Sensor Network For Fire Detection and Alarm System [2]: This paper based on the wireless fire detection and control system is generally composed of a fire detection node, fire alarm node, and fire alarm control panel. The main module to make the entire system communicate wirelessly is the XBee module from Digi International, Inc. One feature of the XBee module that stand out most is the automation mesh network.

IOT- Based Fire Alarm System [3]: This paper based on IoT-based fire alarm system that is capable of detecting the presence of fire, communicating with the concerned parties by calling them when a fire is detected, and receiving and responding to SMS requests from the user. As an improvement, the sensing nodes could depend on a rechargeable battery source instead of a power supply.

The proposed wireless sensor network (WSN) consists of different sensors that share a single wireless network and used GSM. The proposed system results were tested in a smart home to reduce false warnings. Elias et al. also provided a solution using wireless sensor network that was embedded in a micro-controller board for fire hazard detection and fire monitoring purpose [4].

Yu et al. [5] collected the sensor readings for smoke intensity, humidity, temperature to use it in fire detection using Feed-forward neural network approach. The disadvantage of a Feed-forward approach is it demands high processing at the node level resulting in a large amount of power consumption which reduces the lifespan of the node. Also, cluster head destruction in the fire badly affects the robustness of the system.

Information gathered from different sensors such as heat, humidity and CO density light, will be sent on the cluster head using event detection mechanisms. Multiple sensors used to detect fire probability and direction are embedded in each node to reduce the false alarm rate and improve the efficiency [6].

2.1 Existing Problem

The situation is not ideal because fire management systems in homes and industries are not very reliable, efficient, or cost-effective, and lack advanced processing and features such as an automatic alert system for administrators and authorities. They are using older fire safety systems that cannot even activate the sprinkler system and do not communicate with one another properly to prevent false alarms. They also monitor the entire system using applications.

2.2Problem Statement Definiton

The fire management system in houses and industries is not very reliable, efficient, cost effective, and does not have any advanced processing and does not have any features like automatic alert system for admin and authorities and in many buildings there are using older fire safety system that cannot even activate the sprinkler system and all of them do not communicate with each other properly to prevent false alarms.

CHAPTER-3

3.IDEATHON AND PROPOSED SOLUTION

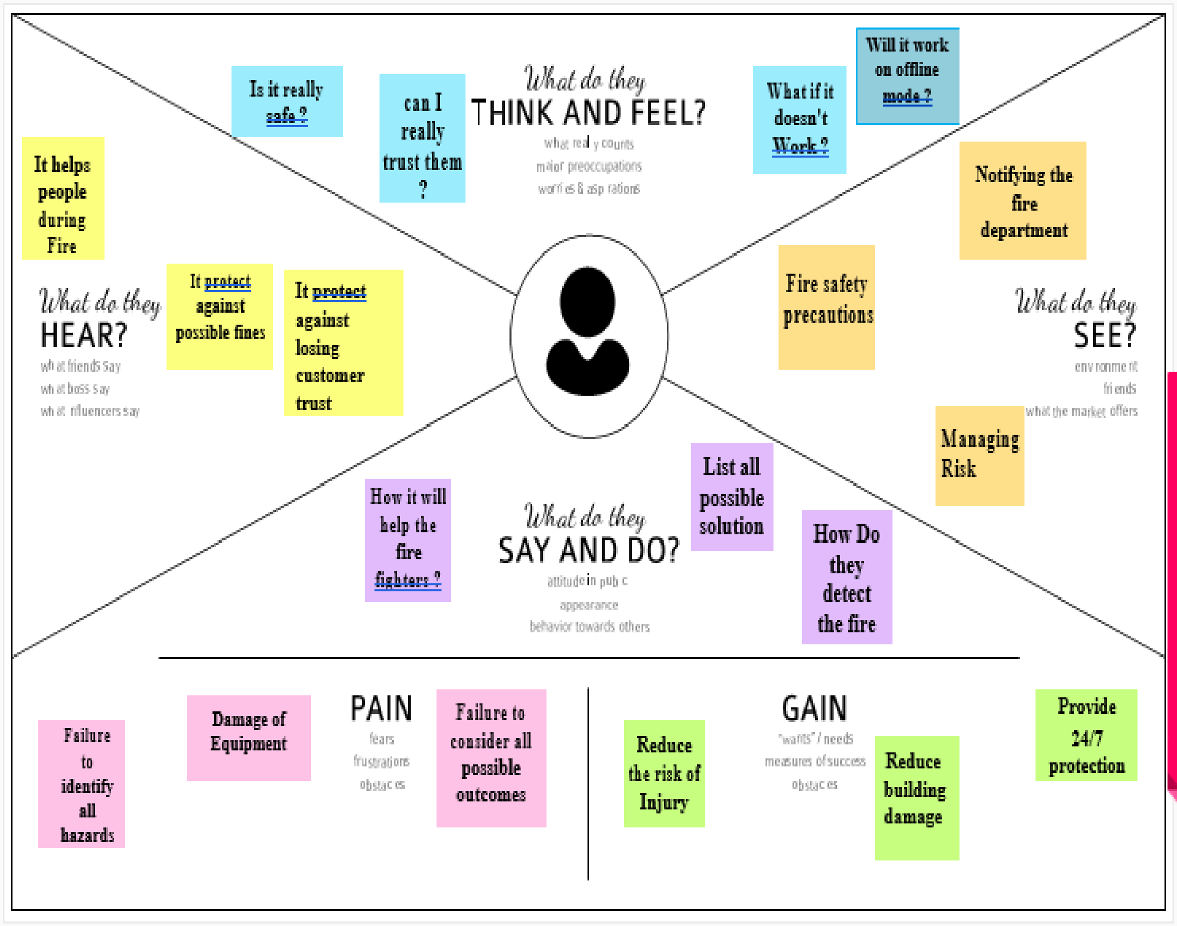
3.1Empathy Map Canvas

- An empathy map is a simple, easy-to-understand visual that captures knowledge about a user's behaviors and attitudes.
- It is a useful tool for assisting teams in better understanding their users.
- Creating an effective solution necessitates understanding the true problem and the person experiencing it.
- The map-making exercise helps participants consider things from the user's perspective, including his or her goals and challenges.

Empathy Map Canvas

Gain insight and understanding on solving customer problems.

Build empathy and keep your focus on the user by putting yourself in their shoes.

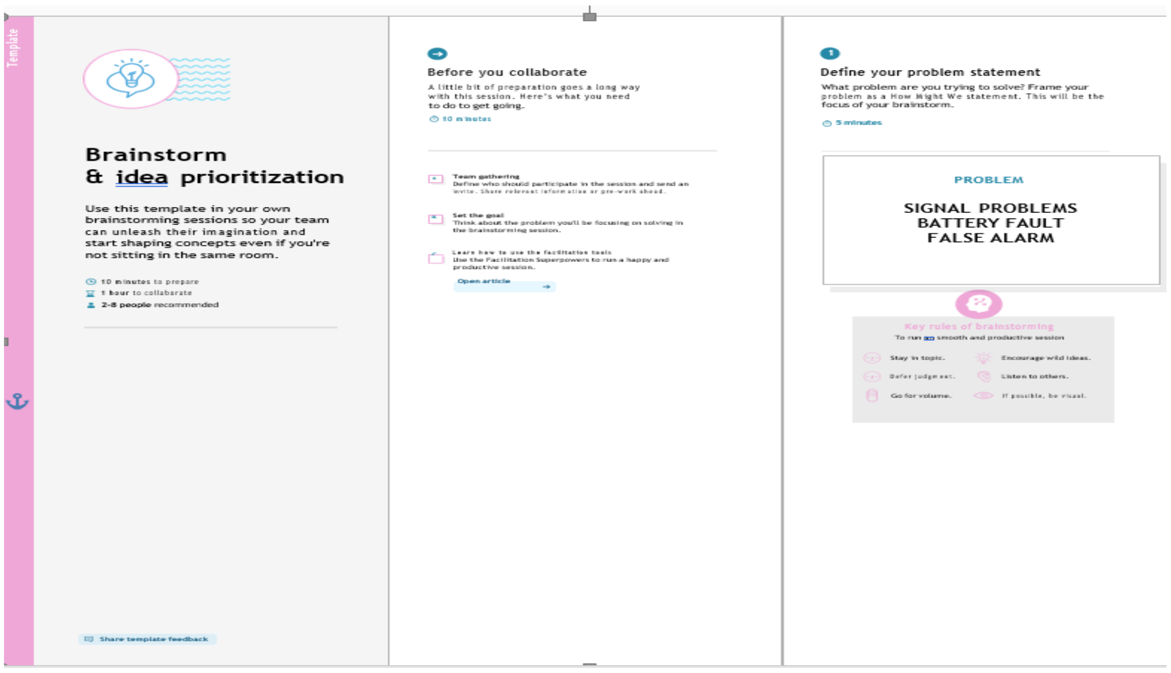


Share your feedback

3.2Ideation and Brainstorming

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

Team was gathered in mural app for collaboration.



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.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Share template feedback

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

Open article

1

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

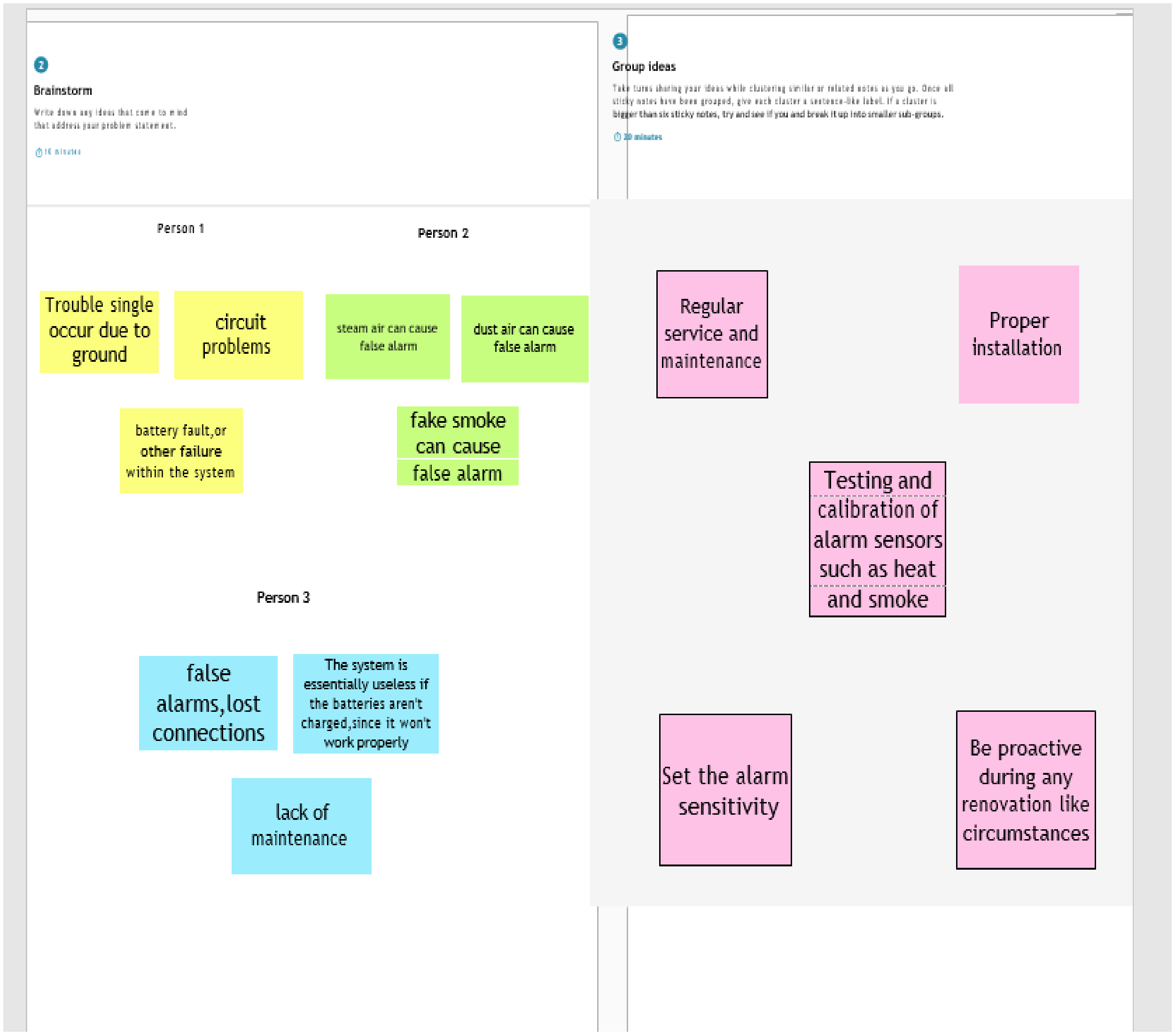
PROBLEM

SIGNAL PROBLEMS
BATTERY FAULT
FALSE ALARM

Key rules of brainstorming
To run an smooth and productive session

Stay in topic.	Encourage wild ideas.
Defer judgment.	Listen to others.
Go for volume.	If possible, be visual.

Step 2: Brainstorm, Idea Listing and Grouping



2

Brainstorm

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

10 minutes

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

Person 1

Trouble single occur due to ground

circuit problems

battery fault,or other failure within the system

Person 2

steam air can cause false alarm

dust air can cause false alarm

fake smoke can cause false alarm

Person 3

false alarms,lost connections

The system is essentially useless if the batteries aren't charged,since it won't work properly

lack of maintenance

Regular service and maintenance

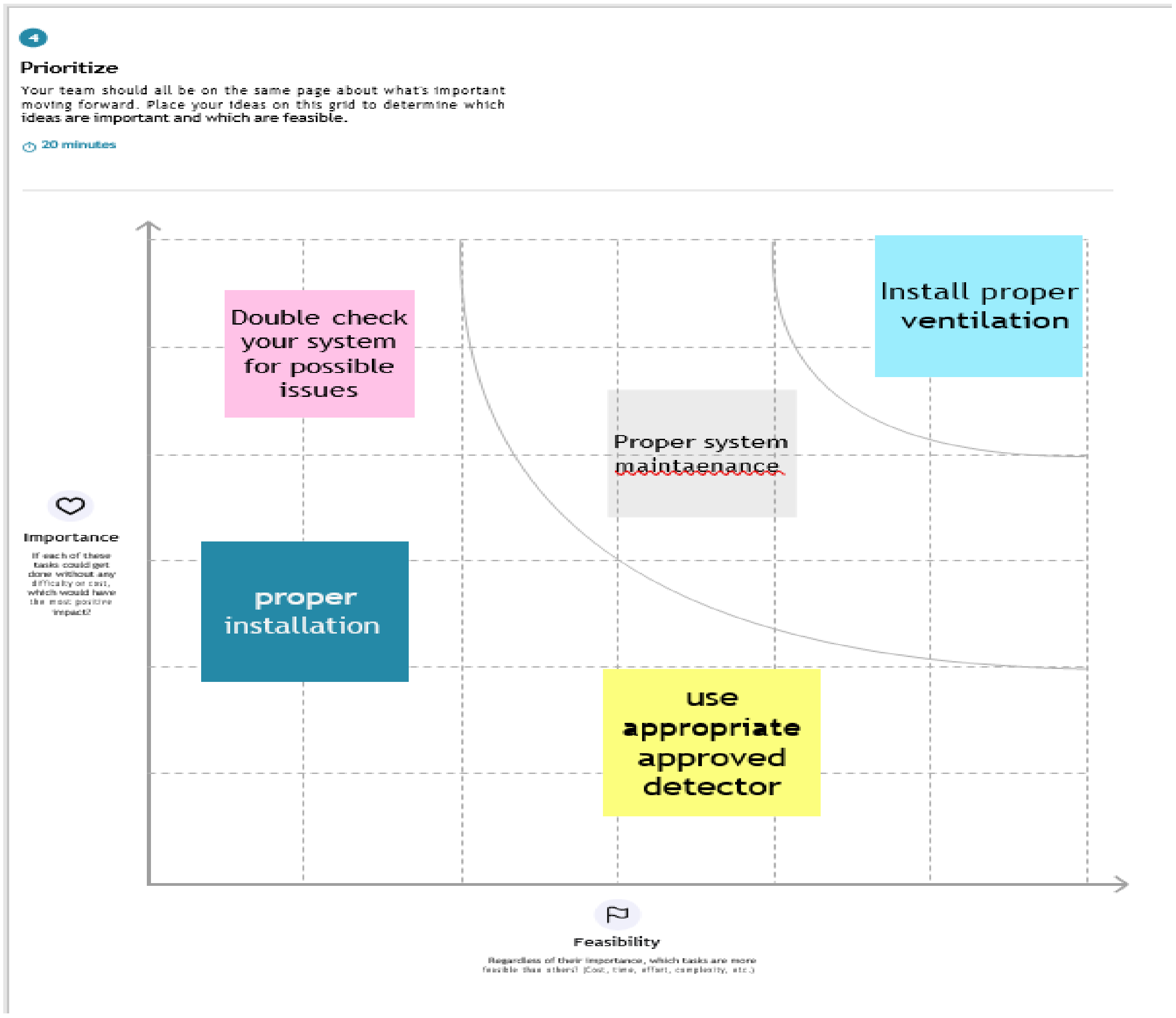
Proper installation

Testing and calibration of alarm sensors such as heat and smoke

Set the alarm sensitivity

Be proactive during any renovation like circumstances

<u>Step3</u>: Idea Prioritization



4

Prioritize

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 which are feasible.

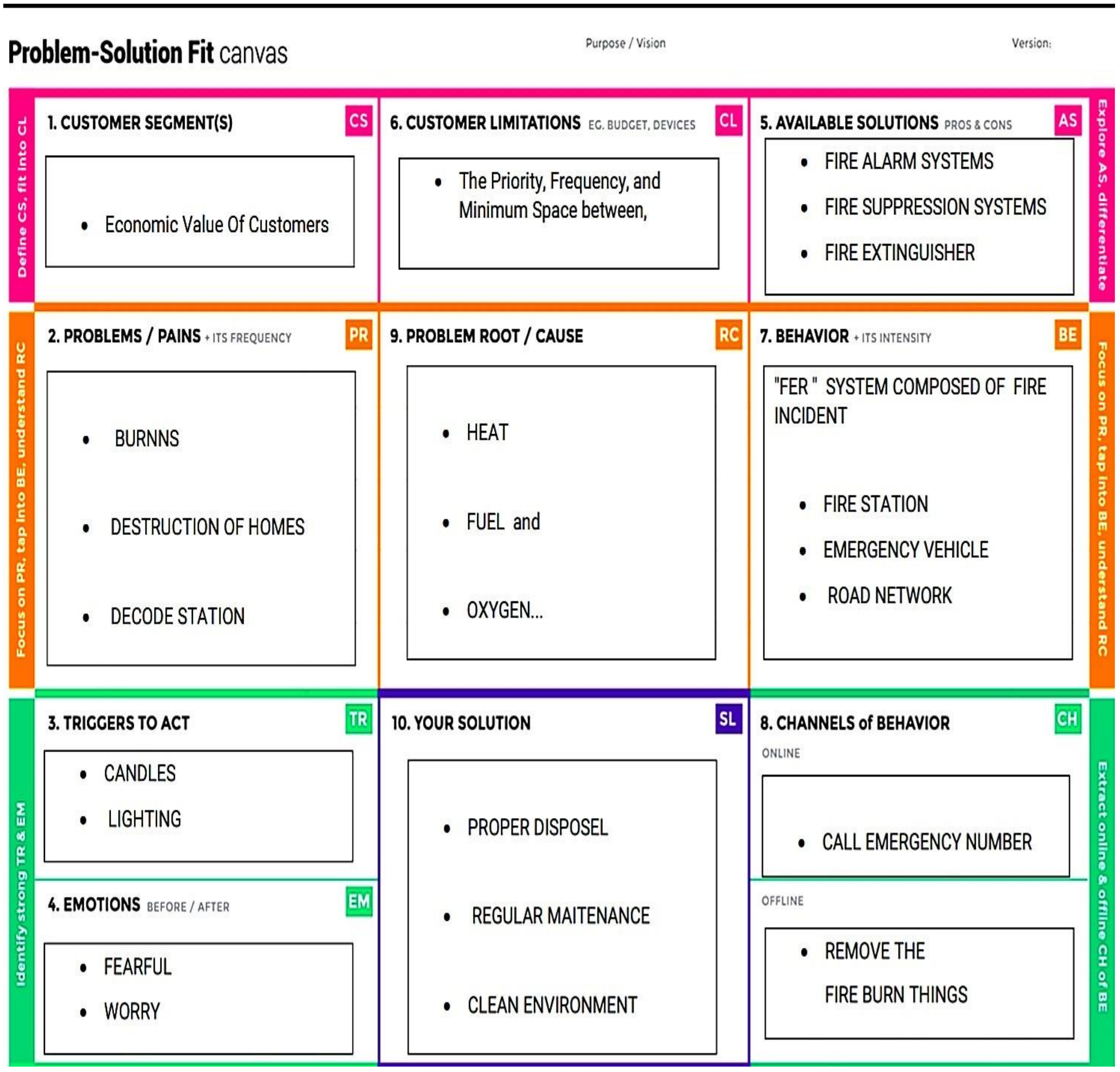
20 minutes

Double check your system for possible issues
Install proper ventilation
Proper system maintaenance
proper installation
use appropriate approved detector
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3Proposed Solution

S No	Parameter	Description
1.	Problem Statement (Problem to be solved)	To improve the safety management system in industries. Improving the safety management system against the fire incidents in industries.
2.	Idea / Solution description	To implement the fire safety management in industry based on IOT using Arduino uno board with fire detection and fire extinguisher system. And using some sensors (Humidity sensor, Flame sensor, smoke sensor) with GPS tracking system.
3.	Novelty / Uniqueness	An integrated system of temperature monitoring, gas monitoring, fire detection automatically fire extinguisher with accuracy of information about locations and response through SMS notification and call.
4.	Social Impact / Customer Satisfaction	It early prevents the accident cost by fire in industries. Nearby locations so maximum extend more accurate reliability Compatibility design integrated system
5.	Business Model (Revenue Model)	This product can be utilized by an industry. This can be thought of as a productive and helpful item as industries great many current rescuing people and machine from the fire accident.
6.	Scalability of the Solution	It is trying to execute this technique as we need to introduce an Arduino gadget which was modified with an Arduino that takes received signals from sensors. Easily operatable and can be maintained. Required low time for maintain. Cost is reasonable value

3.4 Proposed solution fit



Problem-Solution Fit canvas

Purpose / Vision

Version:

Define CS, fit into CL

1. CUSTOMER SEGMENT(S) CS

- Economic Value Of Customers

6. CUSTOMER LIMITATIONS EG. BUDGET, DEVICES CL

- The Priority, Frequency, and Minimum Space between,

5. AVAILABLE SOLUTIONS PROS & CONS AS

- FIRE ALARM SYSTEMS
- FIRE SUPPRESSION SYSTEMS
- FIRE EXTINGUISHER

Explore AS, differentiate

Focus on PR, tap into BE, understand RC

2. PROBLEMS / PAINS + ITS FREQUENCY PR

- BURNNS
- DESTRUCTION OF HOMES
- DECODE STATION

9. PROBLEM ROOT / CAUSE RC

- HEAT
- FUEL and
- OXYGEN...

7. BEHAVIOR + ITS INTENSITY BE

"FER " SYSTEM COMPOSED OF FIRE INCIDENT

- FIRE STATION
- EMERGENCY VEHICLE
- ROAD NETWORK

Focus on PR, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TO ACT TR

- CANDLES
- LIGHTING

4. EMOTIONS BEFORE / AFTER EM

- FEARFUL
- WORRY

10. YOUR SOLUTION SL

- PROPER DISPOSEL
- REGULAR MAITENANCE
- CLEAN ENVIRONMENT

8. CHANNELS of BEHAVIOR CH

ONLINE

- CALL EMERGENCY NUMBER

OFFLINE

- REMOVE THE FIRE BURN THINGS

Extract online & offline CH of BE

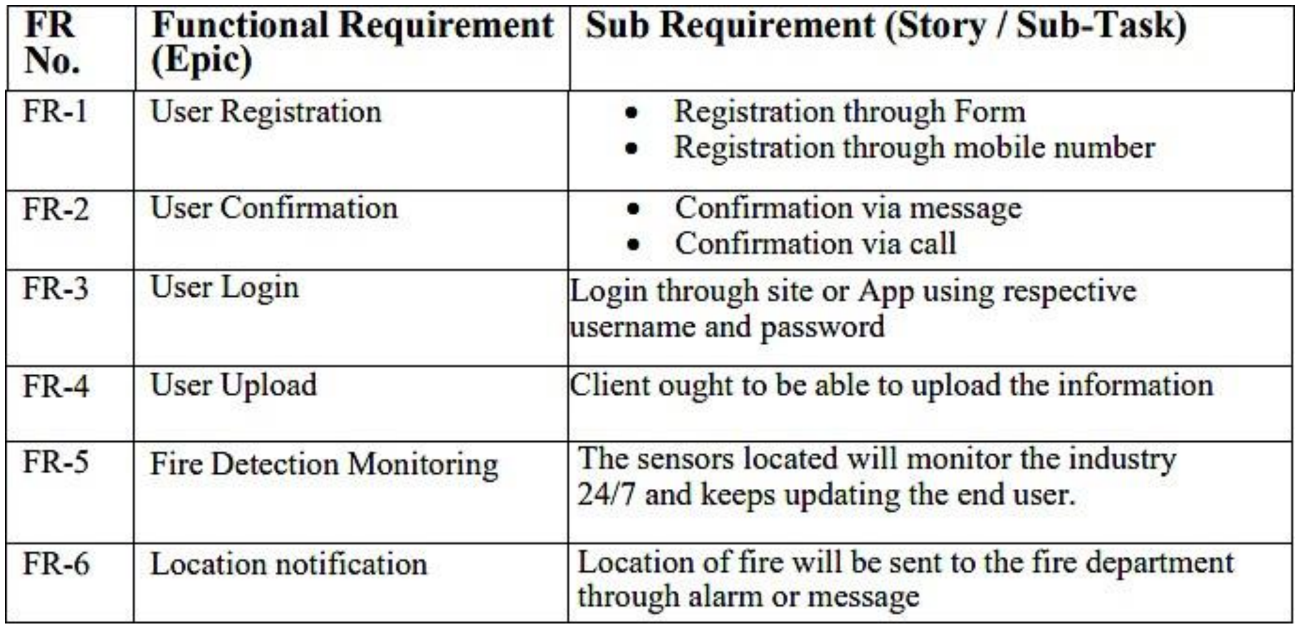
CHAPTER-4

REQUIREMENT ANALYSIS

4.1Functional Requirements

A functional requirement defines a system or component's function, where a function is

- Defined as a specification of behavior between inputs and outputs
- It defines "what the software system should do"
- Defined at the component level
- Usually simple to define
- Aids in testing the software's functionality



FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	• Registration through Form • Registration through mobile number
FR-2	User Confirmation	• Confirmation via message • Confirmation via call
FR-3	User Login	Login through site or App using respective username and password
FR-4	User Upload	Client ought to be able to upload the information
FR-5	Fire Detection Monitoring	The sensors located will monitor the industry 24/7 and keeps updating the end user.
FR-6	Location notification	Location of fire will be sent to the fire department through alarm or message

4.2 Non - Functional Requirements

4.2.1 A non-functional requirement defines a software system's quality attribute.

4.2.2 It limits "How should the software system fulfill the functional requirements?"

4.2.3 It is not required Applied to the entire system

4.2.4 Usually more difficult to define

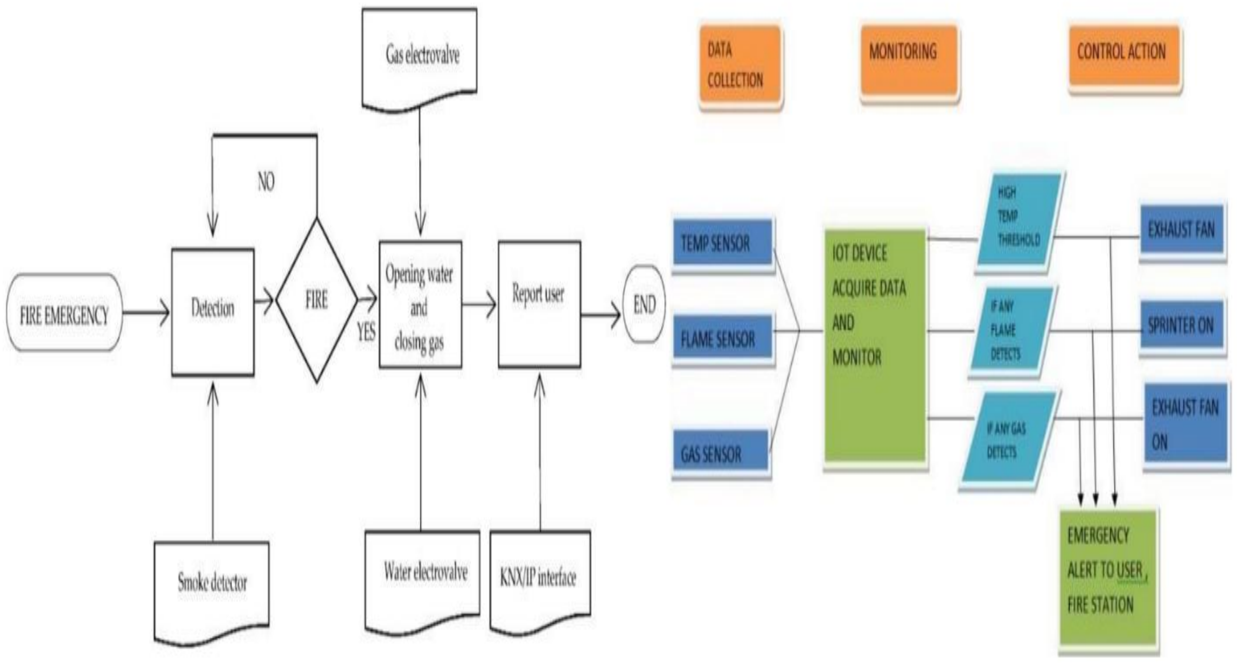
4.2.5 Aids in the verification of software performance

FR . No	Non - Functional Requirement	Description
NFR-1	Usability	● It is the simple and Economic ● Easy to use
NFR-2	Security	● The software remains resilient in the face of attacks ● The Web application is highly secured
NFR-3	Reliability	● Response timer will be faster ● It has high Reliability ● The application runs accurately
NFR-4	Performance	If Fire detected it will be immediately notified through the web application, and it also maintain track periodically.
NFR-5	Availability	We will be Monitoring the Industry by day and Night (24/7). In case of Fire detected we will beintimating the management rapidly.

CHAPTER-5

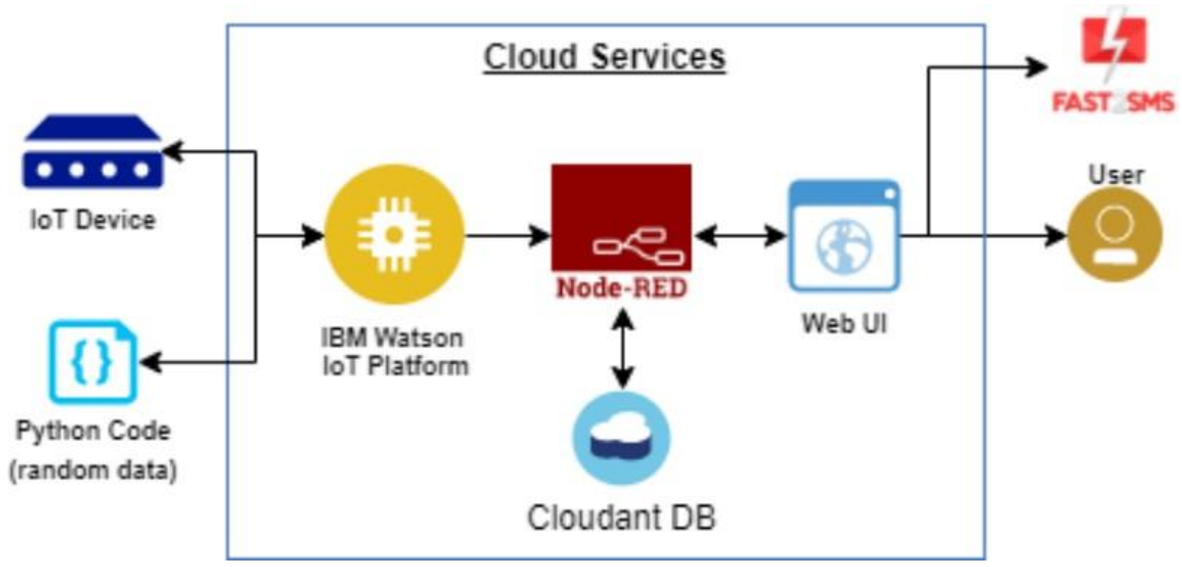
PROJECT DESIGN

5.1 Data flow Diagram

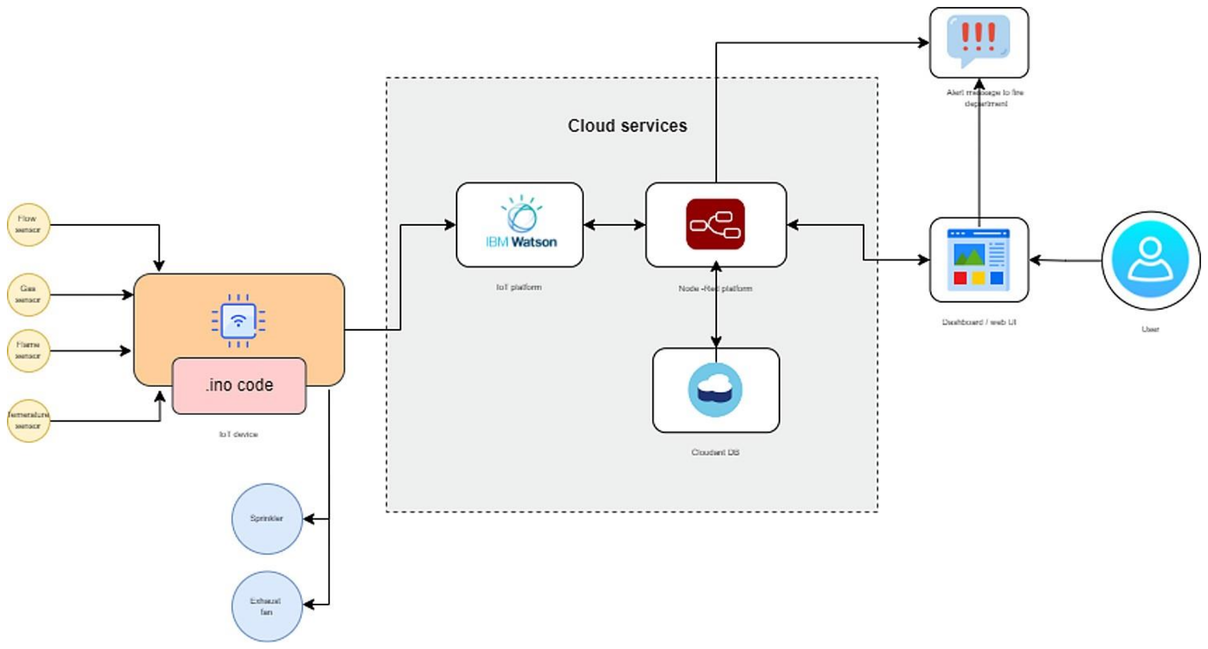


5.2 solution and technical architecture

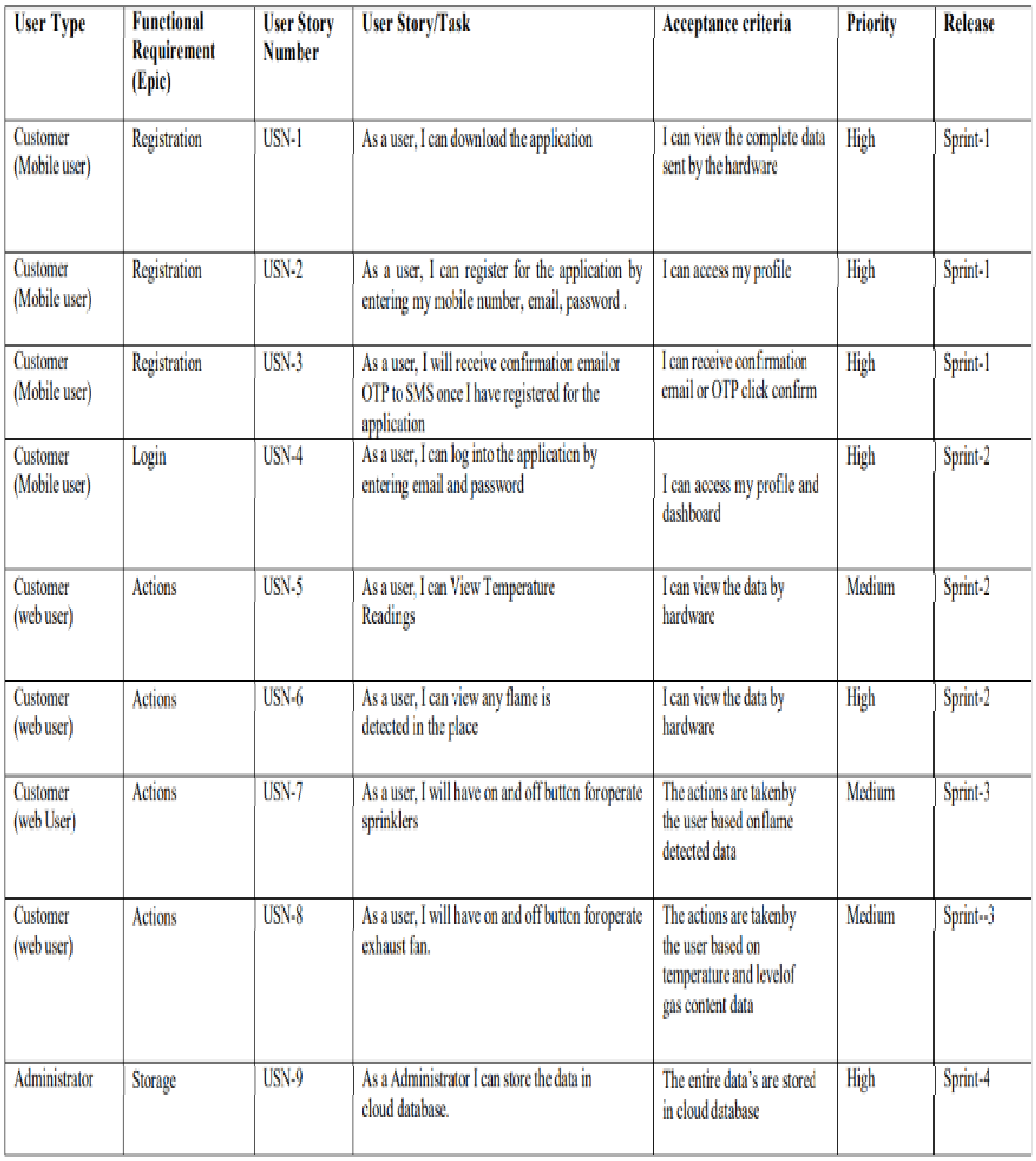
4.3 Solution Architecture



Technical Architecture



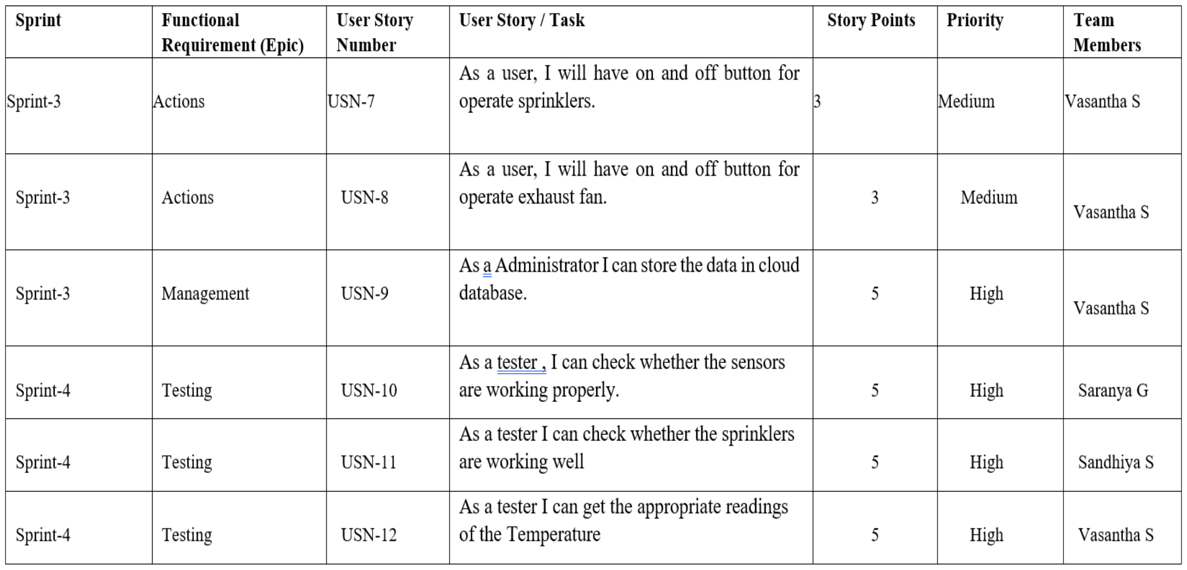
5.3 User Stories



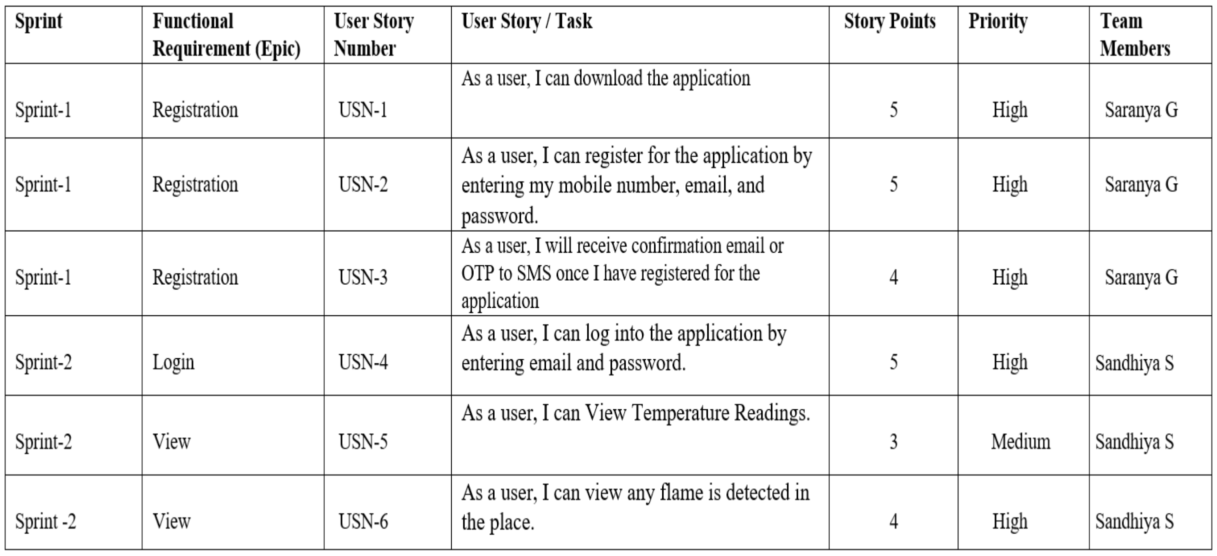
User Type	Functional Requirement (Epic)	User Story Number	User Story/Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can download the application	I can view the complete data sent by the hardware	High	Sprint-1
Customer (Mobile user)	Registration	USN-2	As a user, I can register for the application by entering my mobile number, email, password .	I can access my profile	High	Sprint-1
Customer (Mobile user)	Registration	USN-3	As a user, I will receive confirmation email or OTP to SMS once I have registered for the application	I can receive confirmation email or OTP click confirm	High	Sprint-1
Customer (Mobile user)	Login	USN-4	As a user, I can log into the application by entering email and password	I can access my profile and dashboard	High	Sprint-2
Customer (web user)	Actions	USN-5	As a user, I can View Temperature Readings	I can view the data by hardware	Medium	Sprint-2
Customer (web user)	Actions	USN-6	As a user, I can view any flame is detected in the place	I can view the data by hardware	High	Sprint-2
Customer (web User)	Actions	USN-7	As a user, I will have on and off button foroperate sprinklers	The actions are takenby the user based onflame detected data	Medium	Sprint-3
Customer (web user)	Actions	USN-8	As a user, I will have on and off button foroperate exhaust fan.	The actions are takenby the user based on temperature and levelof gas content data	Medium	Sprint--3
Administrator	Storage	USN-9	As a Administrator I can store the data in cloud database.	The entire data's are stored in cloud database	High	Sprint-4

CHAPTER-6
PROJECT DESIGN AND PLANNING

6.1 Sprint Planning and Estimation

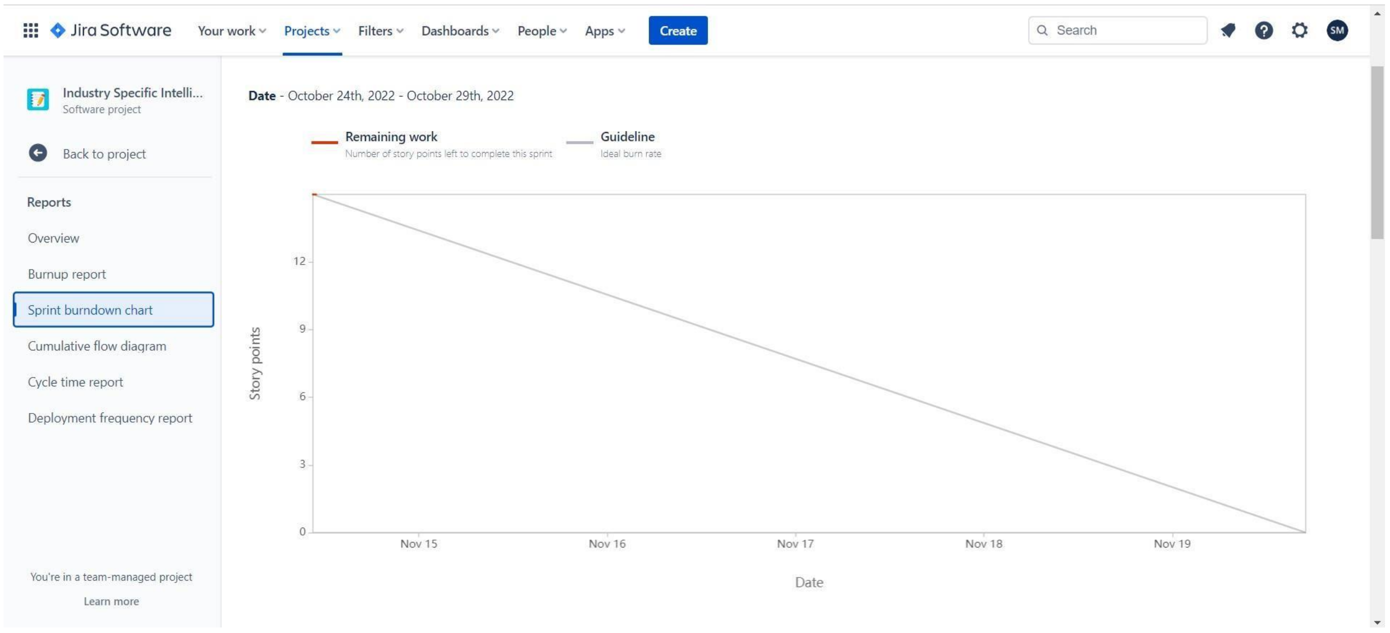


Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	Actions	USN-7	As a user, I will have on and off button for operate sprinklers.	3	Medium	Vasantha S
Sprint-3	Actions	USN-8	As a user, I will have on and off button for operate exhaust fan.	3	Medium	Vasantha S
Sprint-3	Management	USN-9	As a Administrator I can store the data in cloud database.	5	High	Vasantha S
Sprint-4	Testing	USN-10	As a tester , I can check whether the sensors are working properly.	5	High	Saranya G
Sprint-4	Testing	USN-11	As a tester I can check whether the sprinklers are working well	5	High	Sandhiya S
Sprint-4	Testing	USN-12	As a tester I can get the appropriate readings of the Temperature	5	High	Vasantha S

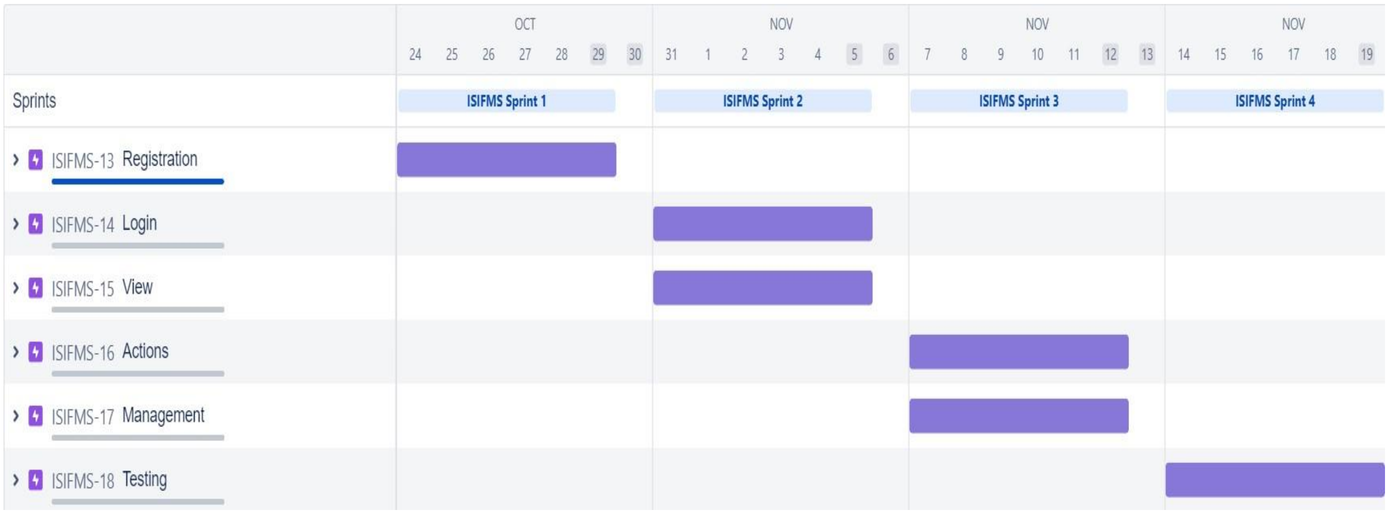


Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can download the application	5	High	Saranya G
Sprint-1	Registration	USN-2	As a user, I can register for the application by entering my mobile number, email, and password.	5	High	Saranya G
Sprint-1	Registration	USN-3	As a user, I will receive confirmation email or OTP to SMS once I have registered for the application	4	High	Saranya G
Sprint-2	Login	USN-4	As a user, I can log into the application by entering email and password.	5	High	Sandhiya S
Sprint-2	View	USN-5	As a user, I can View Temperature Readings.	3	Medium	Sandhiya S
Sprint -2	View	USN-6	As a user, I can view any flame is detected in the place.	4	High	Sandhiya S

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	14	6 Days	24 Oct 2022	29 Oct 2022	20	31 oct 2022
Sprint-2	12	6 Days	31 Oct 2022	05 Nov 2022	20	07 Nov 2022
Sprint-3	11	6 Days	07 Nov 2022	12 Nov 2022	20	13 Nov 2022
Sprint-4	15	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022



ROADMAP:



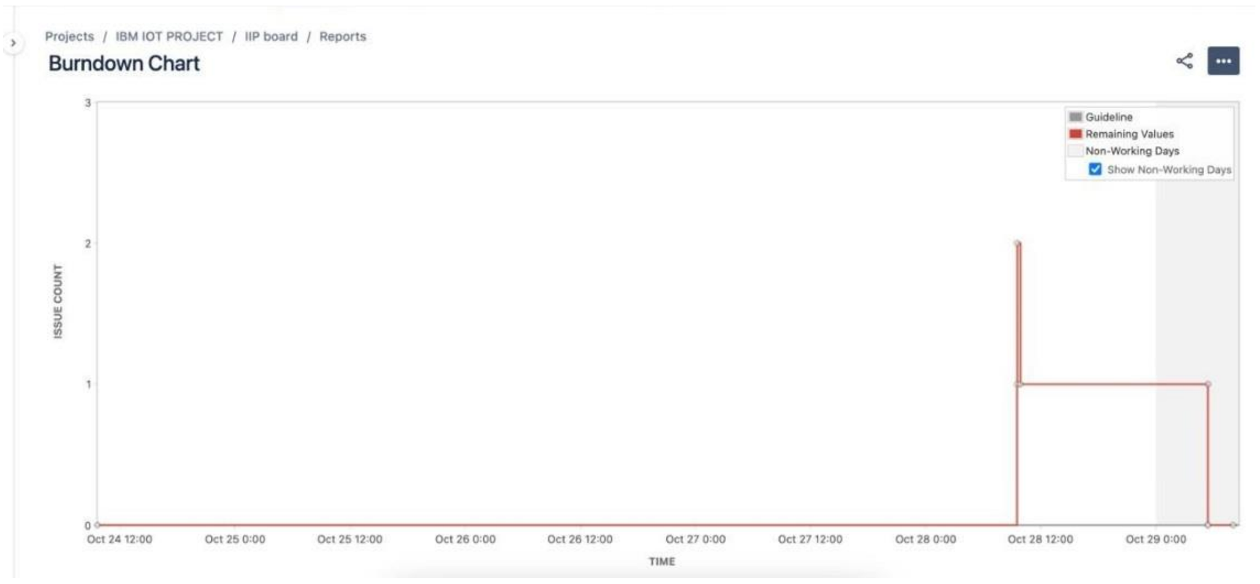
6.2 MILESTONE AND ACTIVITES

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	28 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	24 SEPTEMBER 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	25 SEPTEMBER 2022

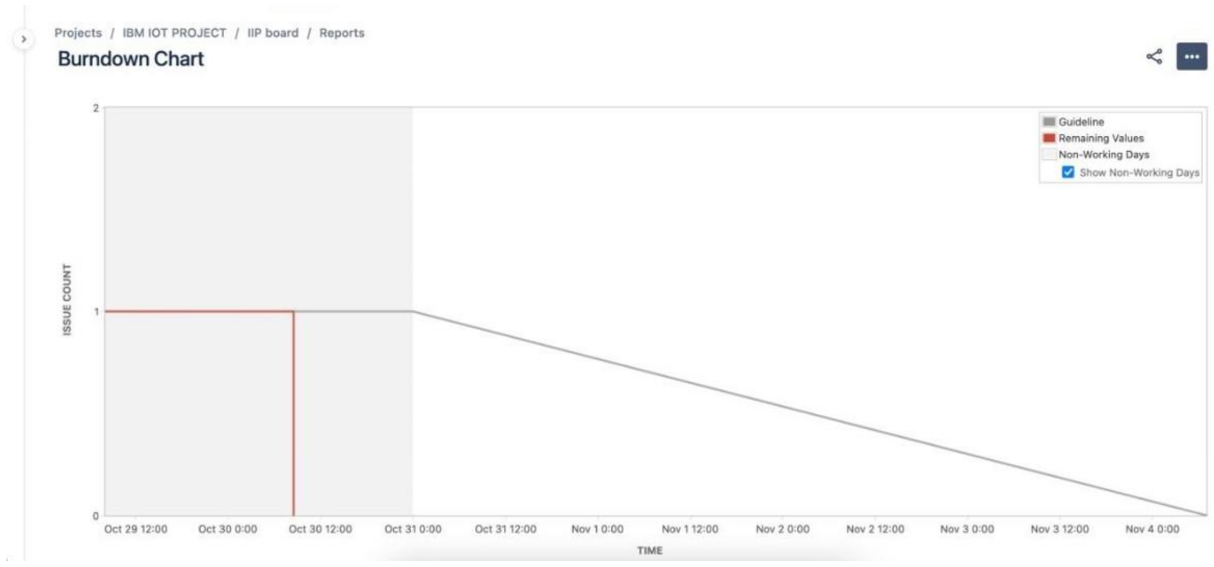
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	23 SEPTEMBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	30 SEPTEMBER 2022
Solution Architecture	Prepare solution architecture document.	28 SEPTEMBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	20 OCTOBER 2022
Functional Requirement	Prepare the functional requirement document.	8 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	9 OCTOBER 2022
Technology Architecture	Prepare the technology architecture diagram.	10 OCTOBER 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	22 OCTOBER 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it.	IN PROGRESS..

Reports from JIRA

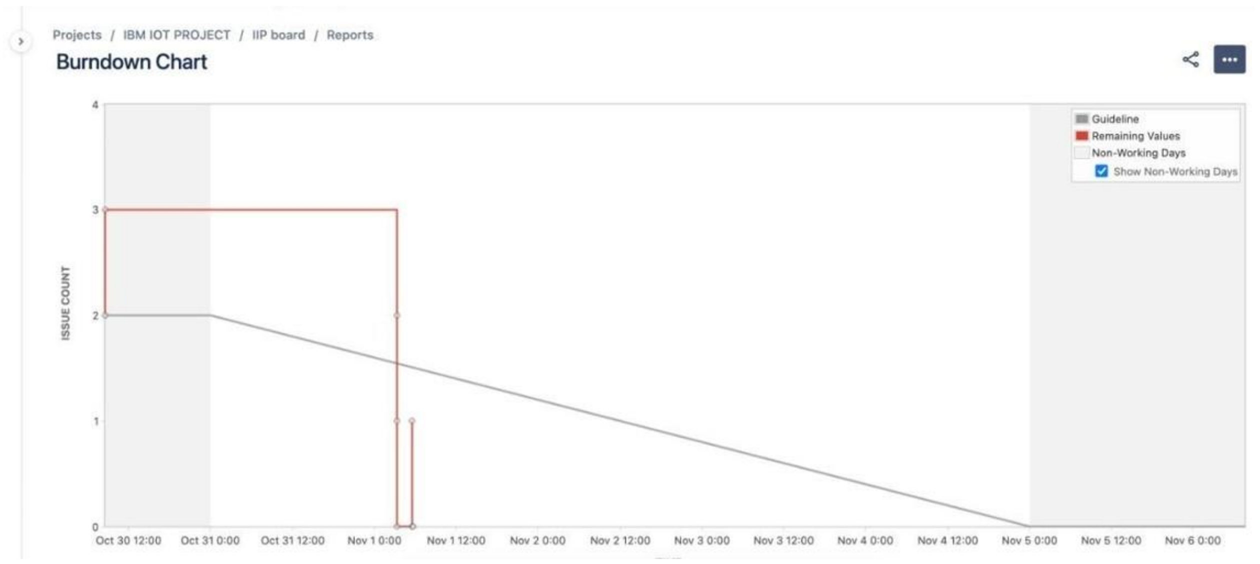
Sprint 1



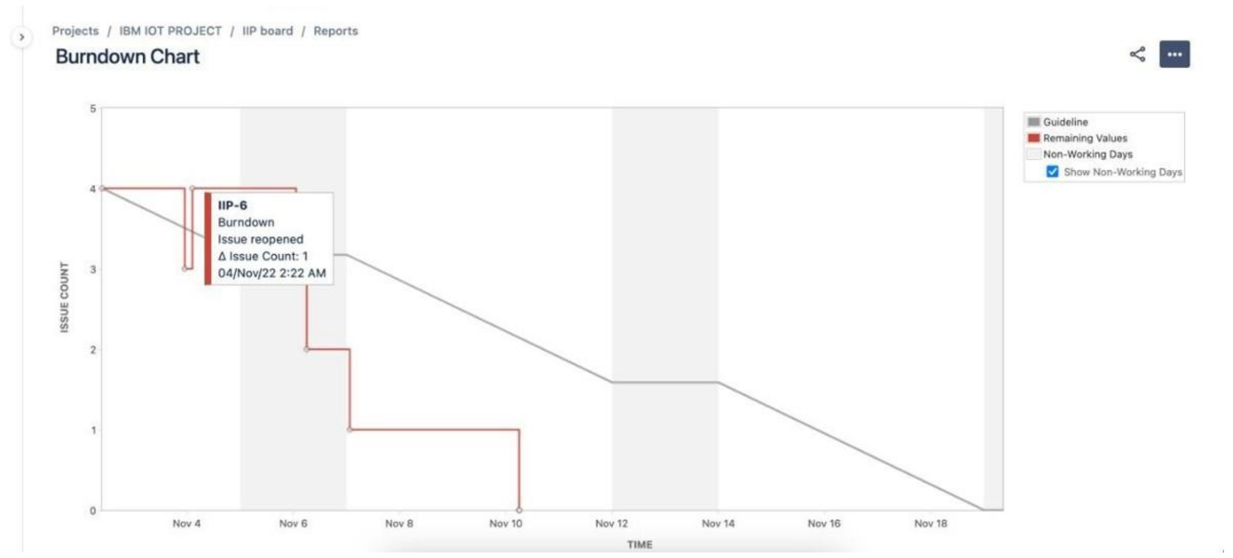
Sprint 2



Sprint 3



Sprint 4



CHAPTER-7

CODING & SOLUTIONING

7.1 Feature 1: False Alarm Checking

```
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";
 }
```

Explanation

- This set of code checks the false alarms and sets the current status
- It also handles the permission management of whether a device will work or not

Feature 2

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

mqttconnect();

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
}
else
{
Serial.println("Publish failed");
}
}
```

Explanation

- It sends the data to IBM Watson Platform

Feature 3

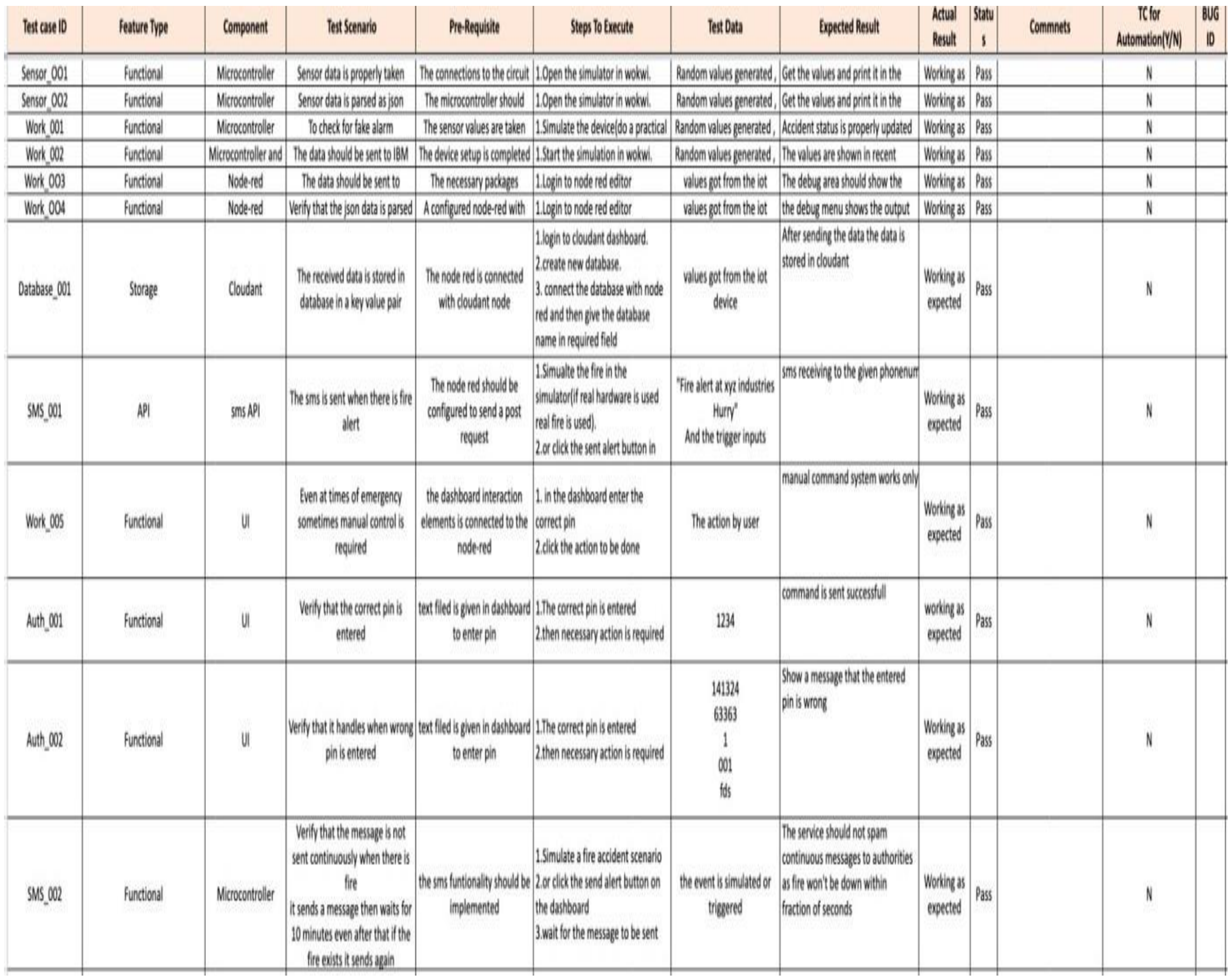
```
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)

{

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), "$.pin", &pincode)){
if(((int)pincode)==137153){ const char *buf;
int len;
if (mjson_find(s, strlen(s), "$.command", &buf, &len))

{

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

}

else if(command=="\"cantsprink\""){ cansprinkoperate = !cansprinkoperate;
}else if(command=="\"sentalert\""){ resetcooldown();
}
}
}
 }
data3="";
}
```

Explanation

- The user's action is received as a command and stored in a buffer
- The event in the device is performed in accordance with the command
- It searches for a secret encrypted pin to perform that event

CHAPTER-8

8.TESTING



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)	BUG ID
Sensor_001	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_002	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_003	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_004	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	

CHAPTER-9

ADVANTAGES & DISADVANTAGES

Advantages:

- Active detection of gas leaks and fire outbreaks
- SMS alerting of administrators and fire authorities
- Turning on/off sprinklers and exhaust fans automatically
- To manually turn on/off sprinklers and exhaust fans, as well as send SMS alerts, authentication is required
- It detects false fire outbreaks automatically, reducing unnecessary panic
- We can confirm that the sprinkler system is functioning properly by using flow sensors
- A dashboard can display the status of any device
- The dashboard can be viewed by users via a web application
- The dashboard can be viewed by users via a web application

Disadvantages:

- Always require an internet connection [only to send the SMS alert]
- If the physical device fails, the entire operation fails
- Because a large amount of data is stored in the cloud database every second, a large database is required

CHAPTER-10

10.CONCLUSION

So we conclude that, our problem premise is solved using IoT devices by developing a smart management system that solves many inherent problems in traditional fire management systems, such as actively monitoring for fire breakouts and gas leakage and sending SMS alerts to administrators and fire authorities.

CHAPTER-11

11.FUTURE SCOPE

The existing devices can be modified to work in various specialized environments, as well as scaled to house use to large labs [Because fire accidents can cause significant loss of human lives in homes to large industries], as well as used in public places and vehicles.

CHAPTER-12

12.APPENDIX

ESP32 - Microcontroller:

The ESP32 is a low-cost, low-power system-on-a-chip microcontroller family with integrated Wi-Fi and dual-mode Bluetooth.

- Memory: 320 KiB SRAM
- CPU: Tensilica Xtensa LX6 Microprocessor @ 160 or 240 MHz
- Power: 3.3 VDC
- Manufacturer: Espressif Systems
- Predecessor: ESP8266

Sensors:

DHT22 - Temperature & Humidity Sensor:

The DHT22 is a simple and inexpensive digital temperature and humidity sensor. It measures the surrounding air with a capacitive humidity sensor and a thermistor and outputs a digital signal on the data pin (no analog input pins needed).

Flow Sensors:

A flow sensor (also known as a "flow meter") is an electronic device that measures or controls the flow rate of liquids and gases through pipes and tubes.

MQ5 - Gas Sensor:

Gas sensors (also referred to as gas detectors) are electronic devices that detect and identify various types of gasses. They are frequently used to detect toxic or explosive gases as well as to measure gas concentration.

Flame Sensor:

A flame-sensor is a type of detector that is intended to detect and respond to the occurrence of a fire or flame. The response to flame detection can be affected by 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>
#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 "sms611"

#define DEVICE_TYPE "1406"
#define DEVICE_ID "14"
#define TOKEN "123456789"

String data3 = "";

String accidentstatus ="";
String sprinkstatus = "";
float temp =0;
bool isfanon = false;

 bool issprinkon = false;
bool cansprinkoperate = true;
bool canfanoperate = 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 sceanarios the time can be
reduced or neclected

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())) {
```

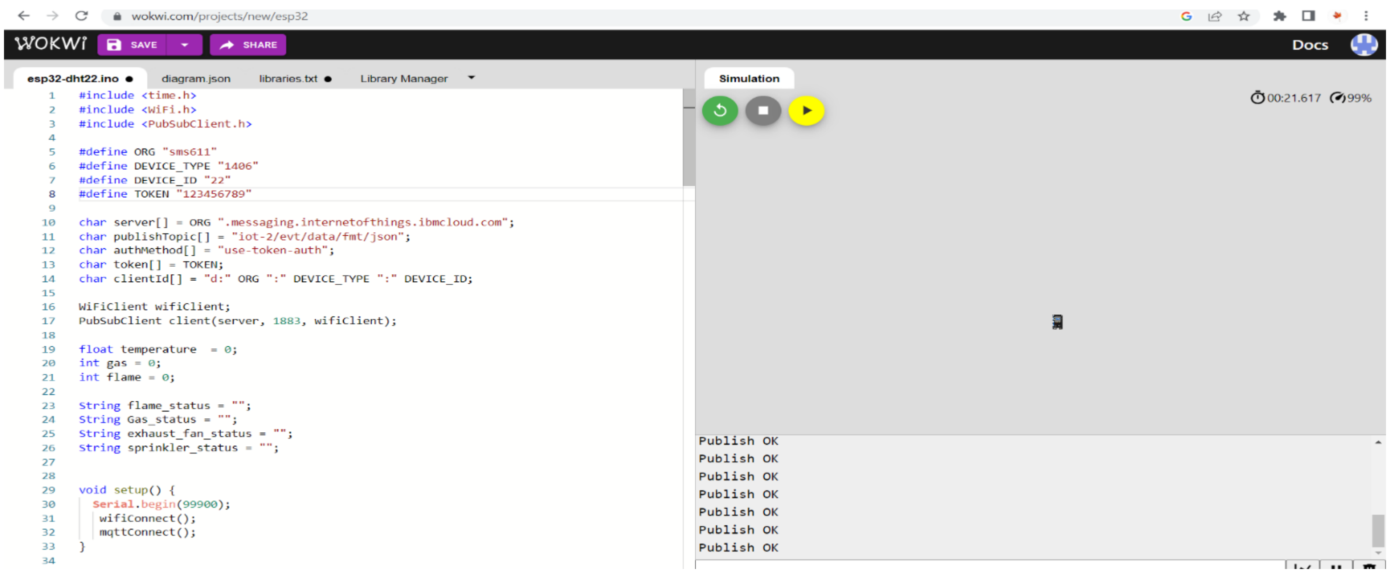
```
Serial.println("Publish ok");     // if it sucessfully upload data on the cloud then it will print publish ok in Serial
monitor or else it will print publish failed

} 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), "$.pin", &pincode))
{
if(((int)pincode)==137153)
{
const char *buf; int len;
if (mjson_find(s, strlen(s), "$.command", &buf, &len))        // And print it

{

String command(buf,len);
 if(command=="\"cantfan\""){
//this works when there is gas sensor reads high value and if there should be a

//manual trigger else it will be automate canfanoperate = !canfanoperate;
}

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

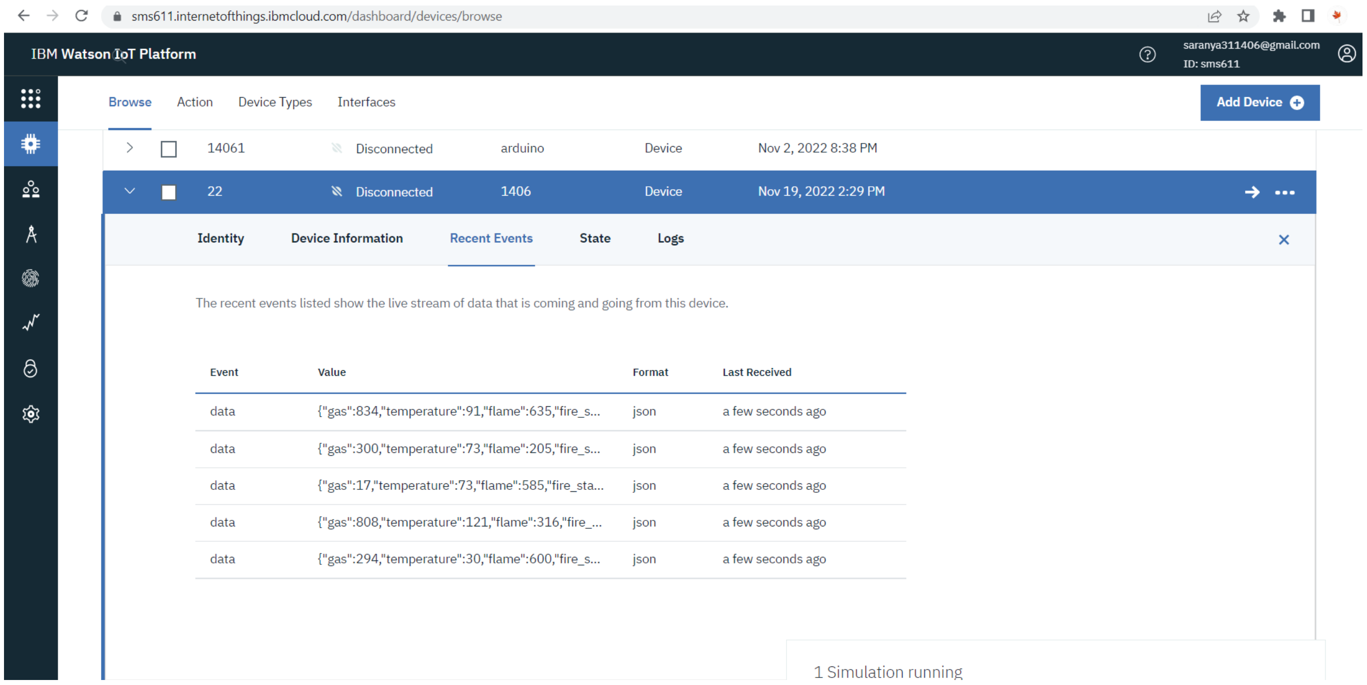
```
//this works when there is accident status is severe and if there should be a

//manual trigger else it will be automate resetcooldown();
}

}

}

}

data3="";

}

void resetcooldown()
{
cooldown = 0;
}

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

OUTPUT:



```
#include <time.h>
#include <WiFi.h>
#include <PubSubClient.h>

#define ORG "sms611"
#define DEVICE_TYPE "1406"
#define DEVICE_ID "22"
#define TOKEN "123456789"

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

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

float temperature  = 0;
int gas = 0;
int flame = 0;

String flame_status = "";
String Gas_status = "";
String exhaust_fan_status = "";
String sprinkler_status = "";


void setup() {
  Serial.begin(99900);
   wifiConnect();
   mqttConnect();
}

```

```
Publish OK
Publish OK
Publish OK
Publish OK
Publish OK
Publish OK
Publish OK
```



14061	Disconnected	arduino	Device	Nov 2, 2022 8:38 PM
22	Disconnected	1406	Device	Nov 19, 2022 2:29 PM

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
data	{"gas":834,"temperature":91,"flame":635,"fire_s...	json	a few seconds ago
data	{"gas":300,"temperature":73,"flame":205,"fire_s...	json	a few seconds ago
data	{"gas":17,"temperature":73,"flame":585,"fire_sta...	json	a few seconds ago
data	{"gas":808,"temperature":121,"flame":316,"fire_...	json	a few seconds ago
data	{"gas":294,"temperature":30,"flame":600,"fire_s...	json	a few seconds ago

1 Simulation running

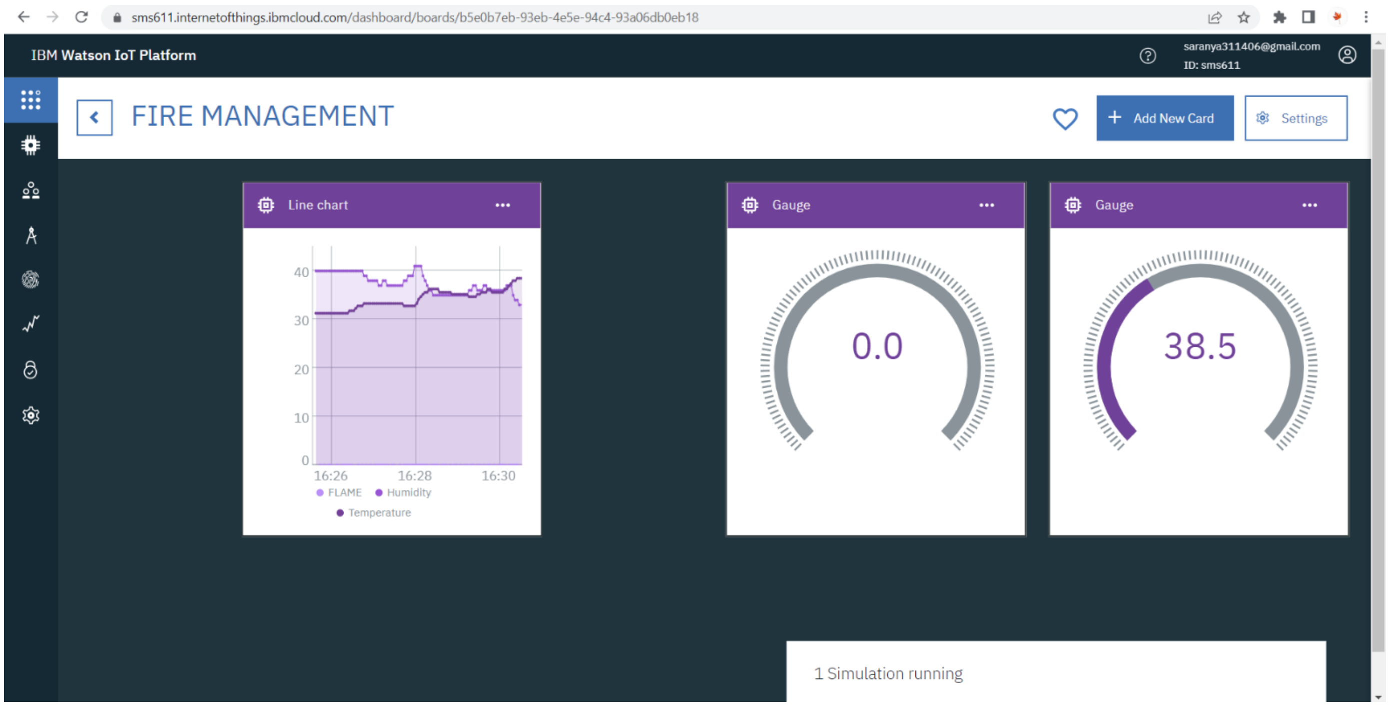
SOURCE CODE:

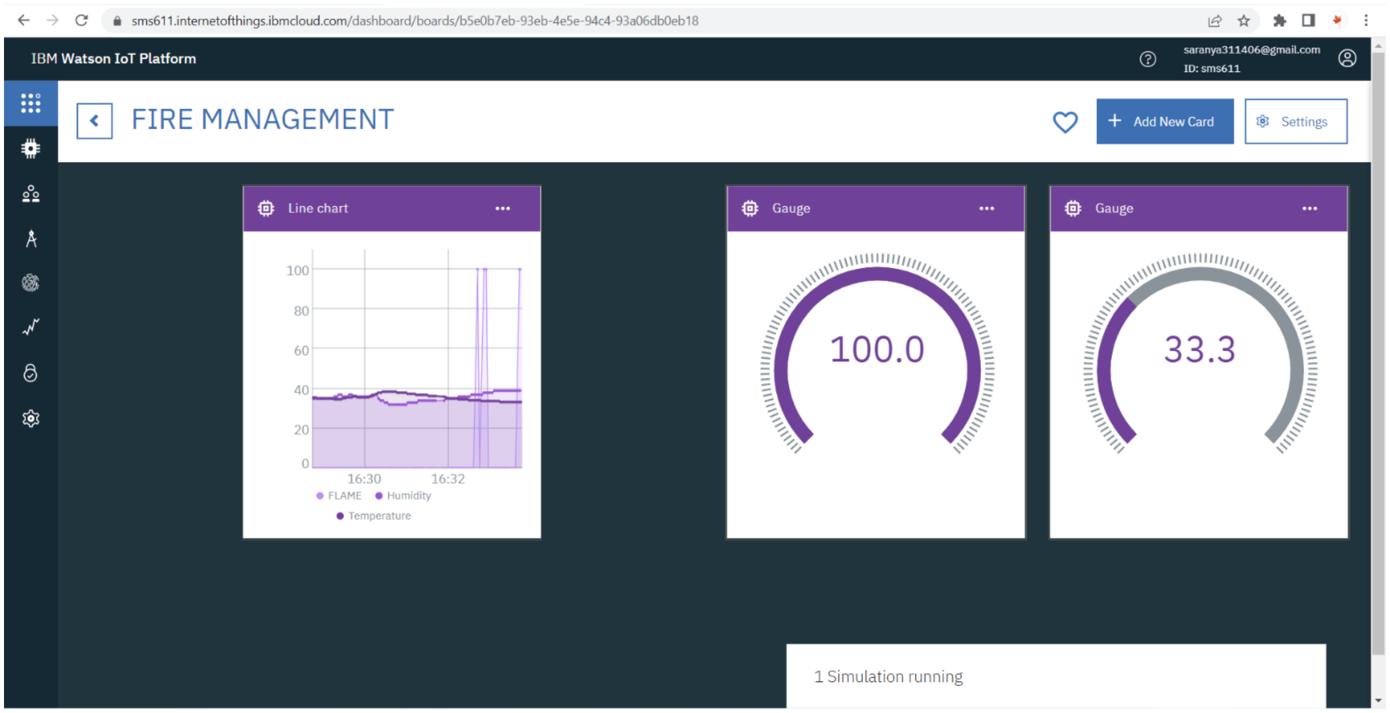
```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <PubSubClient.h>
#include "DHT.h"

const char* ssid = "SMART-G";
const char* password = "10112019";

#define DHTPIN D6
#define G D0
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

#define ID "sms611"
#define DEVICE_TYPE "ESP8266"
#define DEVICE_ID "TEST"
#define TOKEN "TEST-12345"

char server[] = ID ".messaging.internetofthings.ibmcloud.com";
char publish_Topic1[] = "iot-2/evt/Data1/fmt/json";
char publish_Topic2[] = "iot-2/evt/Data2/fmt/json";
char publish_Topic3[] = "iot-2/evt/Data2/fmt/json";
char publish_Topic4[] = "iot-2/evt/Data2/fmt/json";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ID ":" DEVICE_TYPE ":" DEVICE_ID;

WiFiClient wifiClient;
PubSubClient client(server, 1883, NULL, wifiClient);

void setup() {
  pinMode(D0,OUTPUT);
  digitalWrite(D0,HIGH);
    pinMode(D2,OUTPUT);
  digitalWrite(D2,HIGH);
    Serial.begin(115200);
    dht.begin();
    Serial.println();
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
      delay(500);
      Serial.print(".");
    }
    Serial.println("");
    Serial.println(WiFi.localIP());

    if (!client.connected()) {
        Serial.print("Reconnecting client to ");
        Serial.println(server);
        while (!client.connect(clientId, authMethod, token)) {
            Serial.print(".");
            delay(500);
        }
        Serial.println("Connected TO IBM IoT cloud!");
    }
}

long previous_message = 0;
void loop() {
    client.loop();
    long current = millis();
    if (current - previous_message > 3000) {
        previous_message = current;
         float hum = dht.readHumidity();
         float temp = dht.readTemperature();
         float MOI =  map(analogRead(A0), 0, 1023, 0, 100);
         float bi = map(digitalRead(D1), 0, 1, 100, 0 );
         if (isnan(hum) || isnan(temp)  ){
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }

  Serial.print("Temperature: ");
  Serial.print(temp);
  Serial.print("°C");
  Serial.print(" Humidity: ");
  Serial.print(hum);
  Serial.print("%");
  Serial.print("GAS: ");
  Serial.print(MOI);
  Serial.print("FLAME: ");
  Serial.print(bi);
  if(MOI>=80)
  {
      digitalWrite(D0,LOW);

    }
    else
    {
      digitalWrite(D0,HIGH);
    }

      if(bi>=80 ||temp>=35)
  {
      digitalWrite(D0,LOW);

    }
    else
    {
      digitalWrite(D0,HIGH);
    }
```

```
       String payload = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
              payload += ",\"Temperature\":";
              payload += temp;
              payload += "}}";

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

        if (client.publish(publish_Topic1, (char*) payload.c_str())) {
            Serial.println("Published successfully");
        } else {
            Serial.println("Failed");
        }
        String payload1 = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
              payload1 += ",\"Humidity\":";
              payload1 += hum;
              payload1 += "}}";
              Serial.print("Sending payload: ");
              Serial.println(payload1);
              Serial.println('\n');

         if (client.publish(publish_Topic2, (char*) payload1.c_str())) {
            Serial.println("Published successfully");
        } else {
            Serial.println("Failed");
        }


       String payload3 = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
              payload3 += ",\"GAS\":";
              payload3 += MOI;
              payload3 += "}}";

        Serial.print("Sending payload: ");
        Serial.println(payload3);

        if (client.publish(publish_Topic3, (char*) payload3.c_str())) {
            Serial.println("Published successfully");
        } else {
            Serial.println("Failed");
        }



String payload4 = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
              payload4 += ",\"FLAME\":";
              payload4 += bi;
              payload4 += "}}";

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

```
    Serial.println(payload4);

    if (client.publish(publish_Topic4, (char*) payload4.c_str())) {
      Serial.println("Published successfully");
    } else {
      Serial.println("Failed");
    }




  }
}
```

OUTPUT:





Github Link: https://github.com/IBM-EPBL/IBM-Project-15468-1659599011

DEMO LINK: https://youtu.be/3rZTuWpLLsM