Anna University Regional Campus, Madurai

Nalaiya Thiran

executed by

Industry-specific intelligent fire management system

Project ID : PNT2022TMID07797

Team Lead : S.M.Sanjay Kumar

Team Members : G.Gughan
G.Pritheesh
R.Srinath
M.Sai Dhanush

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	this system can perform different parameter measurements early detection of building fires
2.	Idea / Solution description	This fire alarm system incorporates the heat and flame detector that are connected in parallel.The microcontroller is used as the heart of this fire alarm system that controls the entire operation involved . The fire alarm system is capable to locate and identified the place that is in fire where by its monitored using the monitoring system.
3.	Novelty / Uniqueness	In this paper, the installed Arduino device which was programmed with Android Studio receives gas smoke ,the temperature and humidity signal from the sensors . The sensor is connected to the input of the arduino with the help of connecting the cables or jumper cables . Further the circuit goes toward output where the buzzer is connected. If we differ the value of the buzzer then we get a variation in the buzzer sound.
4.	Social Impact / Customer Satisfaction	This product has huge social impact as presentation of the industry workers from fire related accidents.Prevention of the industry fire accident can also increases the industrial financial status
5.	Business Model (Revenue Model)	This product can be utilized by a industries .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 studio that takes received signals from sensors . This recognizes the fire from each area in turn assuming there is fire in other area the framework can not distinguish . So this item will be introduced in each required area independently.

1. INTRODUCTION

1.1 Project Overview

- The smart fire management system includes a Gas 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.

1.2 Purpose

The purpose of the system is :

- To prevent life losses , assests damage and uncontrollable spread of fire.
- To ensure the safety of workers and alert the manager and fire department.
- To not to recklessly endanger the life of the fire workers. This can be done by taking the control measures automatically.

2. LITERATURE SURVEY

2.1 Existing problem

The existing problems of the system are:

- Cost of ownership : The fire management system shoulb be cost effective. In average, the fire management is expected to last 10 years. The biggest problem is when the system cannont be maintained any longer due to component non-avaliabilty or due to being unsupported by the manufacturer.
- Structural changes : The structure of the hospital changes over time. The fire management system should be easily able to upgrade and adaptable to the changing structure.
- Evaculation and fire stratergy : The alert and the control measures are taken immediately, so that the building can be completely evaculated.
- System performance changes within specific environments : The industry will have unique or specified condition at some time. The major problem caused is the false fire alarm.

2.2 References

[1] Gazi weldesyase, Bahta G/meskel, Mekonen Abreha, Solomon Baynes, "GSM Based Fire and Smoke Detection and Prevention System", on 08/10/2010, Adigrat, Tigray, Ethiopia.

[2] May Zaw Tun, Htay Myint, "Arduino based Fire Detection and Alarm System Using Smoke Sensor", Volume 6, Issue 4, on April – 2020, Myanmar.

[3] Nitin Galugade, Mahesh Jakka, Devika Nair, Madhur Gawas, "Fire Monitoring and Controlling System based on Iot", 2020, Mumbai, India.

2.3 Problem Statement Definition

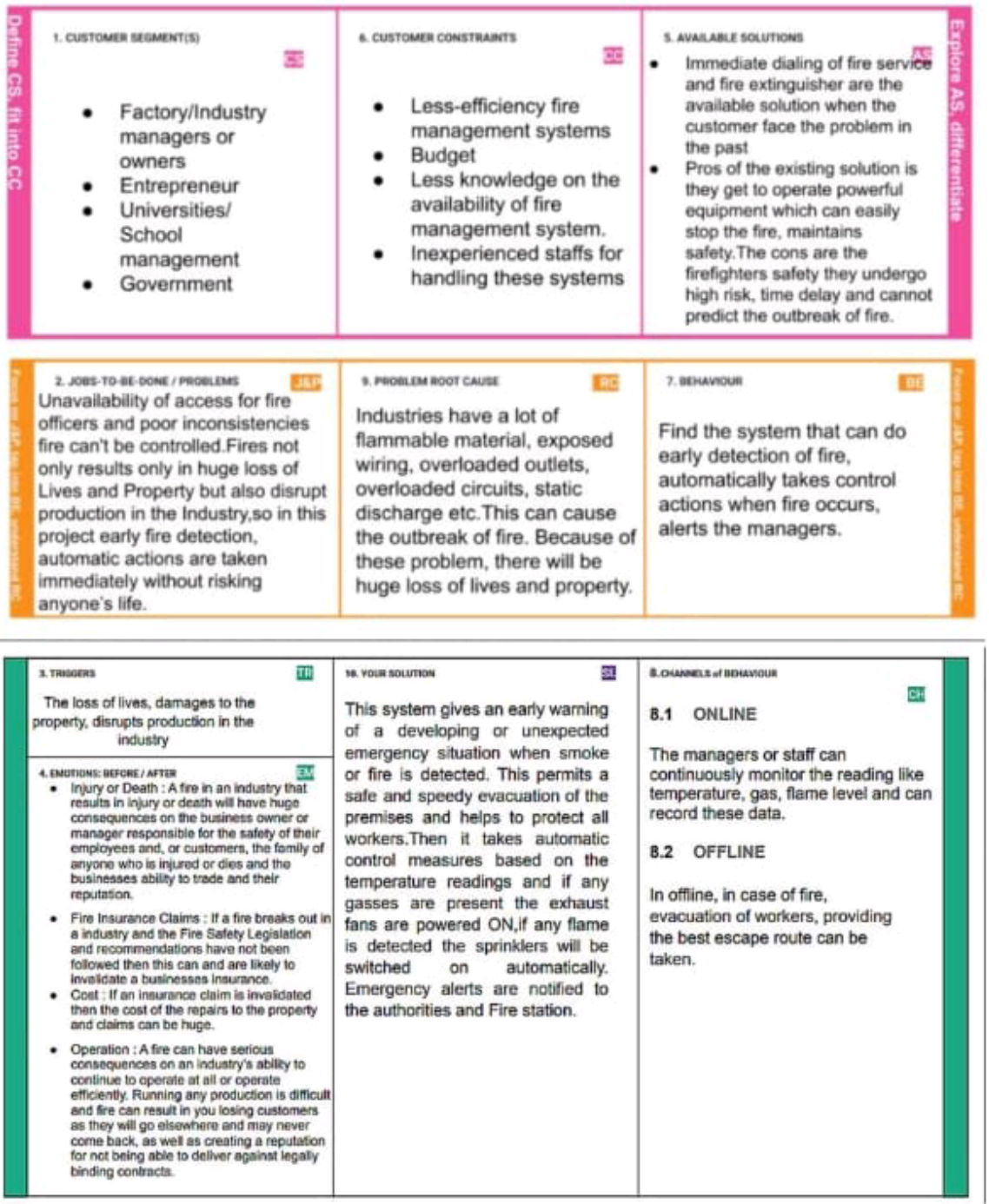
Background: Fire is the rapid oxidation of a material in the exothermic chemical process of combustion, releasing heat, light and various reaction products. Although it's a natural process, it can lead to great destruction. On average, everyday 35 people killed due to Fire-related accidents in the five years between 2016 and 2020, according to a report by Accidental Deaths and Suicides in India (ADSI), maintained by the National Crime Records Bureau. Fire is one of the major concerns when analyzing the potential risks on the building. Industrial Fires and Explosions cost companies and governments billions of Rupees every year apart from the loss oflife, which can't be described in monetary terms. These Fires not only results only in huge lossof Lives and Property but also disrupt production in the Industry. The Nilflisk says that the five major causes of industrial fires and explosions are Combustible dust, hot works, Flammable liquids and gasses, equipment and machinery and Electrical hazards.

Objective: The objective of this Industry-Specific Intelligent Fire Management System is to detect any changes in environment like detecting hazardous gas, flame detection and temperature that can lead to fire and exploitation incident. Based on the temperature readings and if any Gasses are present the exhaust fans should be powered ON automatically to replace contaminated and stale air with fresh, healthy air. If any flame is detected the sprinklers will be switched on automatically. Emergency alerts are notified to the authorities and Fire station. So that the authorities and Fire Fighters can control the situation.

Industry-specific intelligent fire management system

3.2 Ideation & Brainstorming

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User visibility	Emergency alerts via Fast SMS.
FR-2	User reception	The data like amount of gas levels, smoke content and temperature are received via SMS.
FR-3	User Understanding	Based on the data, the user understands that if any of the data is above the threshold value, then there is a fire burst.
FR-4	User action	In case of fire bursts, the user needs to take actions like find the best escape route, evacuate the workers and take necessary actions to control the fire.

4.2 Non-functional Requirements:

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

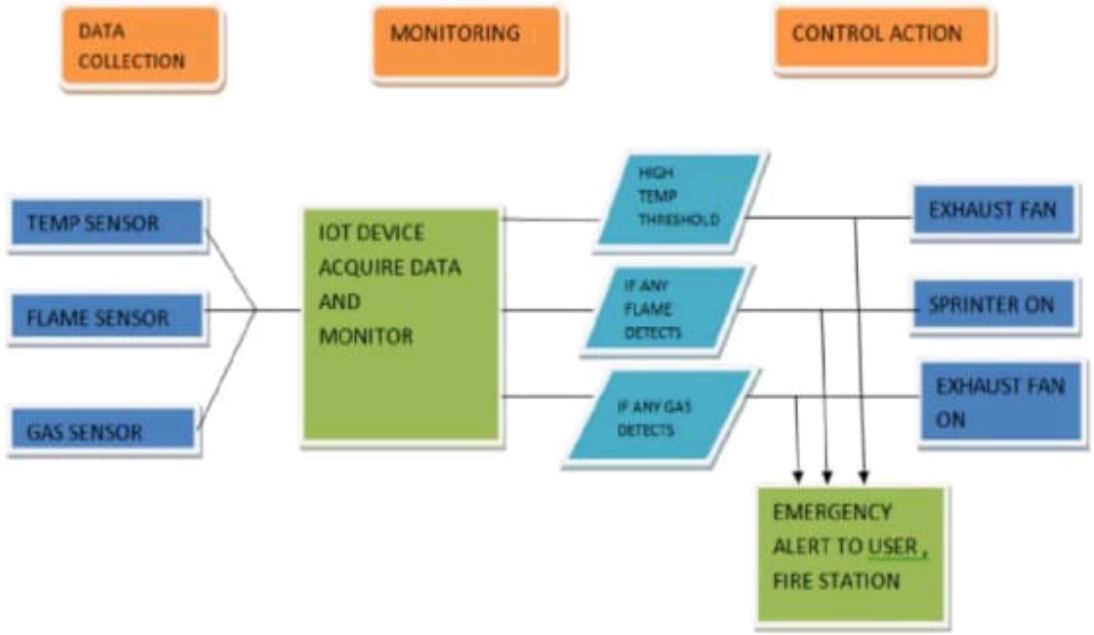
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It ought to have the option to caution inhabitants of the structure the utilization of every perceptible and apparent alert.
NFR-2	**Security**	It ought to be utilized to guarantee the insurance of both important properties, as well as human existence.
NFR-3	**Reliability**	It might have a capacity to recognize the smoke accurately and doesn't give a false caution or signal.

NFR-4	**Performance**	It ought to have Programmed fire sprinklers combined with identification which distinguishes the flames, yet in addition smother the flames in the underlying stage itself.

NFR-5	**Availability**	It could be accessible for day in and day out hours so it tends to be useful for individuals.
NFR-6	**Scalability**	The sensors and boards utilized in this framework ought to have the option to effortlessly change overhaul concurring to change and need in requirements

5 PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered

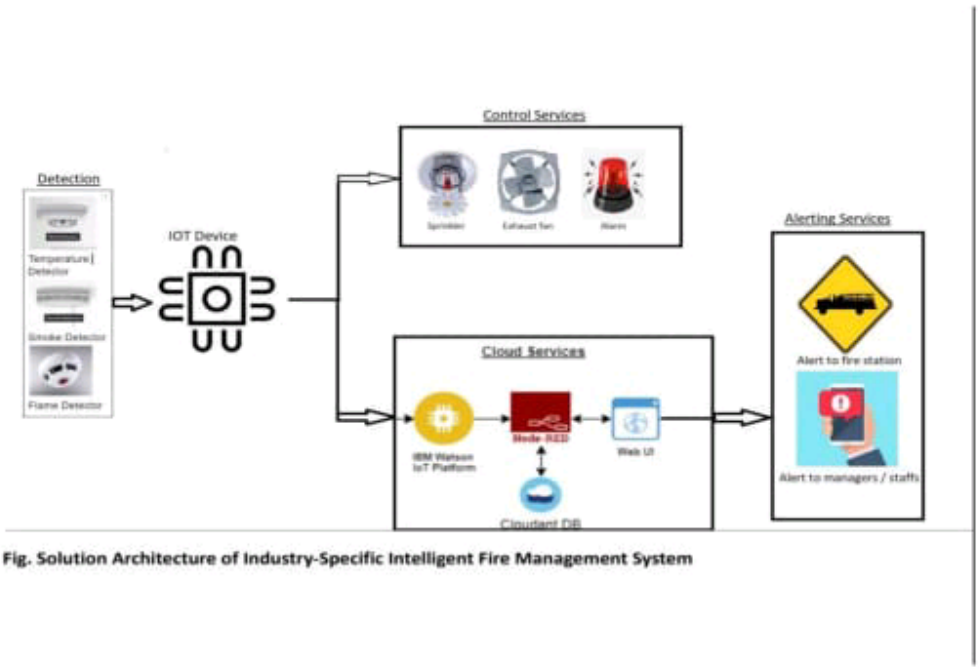


Fig. Solution Architecture of Industry-Specific Intelligent Fire Management System

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 data sent by the hardware.	High	Sprint-3
Customer (Web user)	Registration	USN-1	As a user, I can view the application web page	I can view the data sent by the hardware.	High	Sprint-3
Customer (Data types)	Data viewing	USN-1	As a user, I can view Temperature readings	Data by the hardware	High	Sprint-1
		USN-2	As a user, I can view level of gas content	Data by the hardware	High	Sprint-1
		USN-3	As a user, I can view if any flame is detected	Data by the hardware	High	Sprint-1
Customer	Actions	USN-1	As a user, I will have exhaust fan on and off button	Based on temperature and level of gas content data, actions are taken by the user	Medium	Sprint-2
		USN-2	As a user, I will have sprinkler on and off button	Based on the flame detected data, actions are taken by the user.	Medium	Sprint-2
Administrator	Storage	USN-1	As an administrator, I will store the data in Cloud database	All the data are stored in cloud database.	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1		US-1	Create the IBM Cloud services which are being used in this project.	6	High	S.Bhumika K.janani E.keerthana
Sprint-1		US-2	Configure the IBM Cloud services which are being used in completing this project.	4	Medium	S.bhumika A.B.Saritha E.keerthana
Sprint-1		US-3	IBM Watson IoT platform acts as the mediator to connect the web application to IoT devices, so create the IBM Watson IoT platform.	5	Medium	S.Bhumika E.keerthana K.janani
Sprint-1		US-4	In order to connect the IoT device to the IBM cloud, create a device in the IBM Watson IoT platform and get the device credentials	5	High	K.Janani A.B.Saritha
Sprint-2		US-1	Configure the connection security and create API keys that are used in the Node-RED service for accessing the IBM IoT Platform.	10	High	s.bhumika E.keerthana
Sprint-2		US-2	Create a Node-RED service.	10	High	S.Bhumika K.janani
Sprint-3		US-1	Develop a python script to publish random sensor data such as temperature, Flame level and Gas level to the IBM IoT platform	7	High	S.bhumika k.janani A.B.saritha E.Keerthana

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3		US-2	After developing python code, commands are received just print the statements which represent the control of the devices.	5	Medium	S.bhumikaa A.B.saritha K janani E.keerthana
Sprint-3		US-3	Publish Data To The IBM Cloud	8	High	S.bhumika k.janani E keerthana
Sprint-4		US-1	Create Web UI in Node- Red	10	High	S.bhumika k.janani A.B.saritha
Sprint-4		US-2	Configure the Node-RED flow to receive data from the IBM IoT platform and also use Cloudant DB nodes to store the received sensor data in the cloudant DB	10	High	S.bhumika k.janani E keerthana A.B.saritha

6.2 Sprint Delivery Schedule

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	03 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

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

6.3 Reports from JIRA

https://pnt2022tmid47460.atlassian.net/jira/software/projects/ISIFMS2/boards/2/roadmap

7. CODING & SOLUTIONING

7.1 Feature 1

Python script for generating the random sensor values - Temperature, Flame Level and Gas Level to the IBM Watson IoT Platform.

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random


#Provide your IBM Watson Device Credentials

organization = "4aqwut"
deviceType = "12345678dt"
deviceId = "12345678did"
authMethod = "token"
authToken = "*PrtsGAO?B@_tTPEKT"


# Initialize GPIO

def myCommandCallback(cmd):
   print("Command received: %s" % cmd.data['command'])
   status=cmd.data['command']
   if status=="sprinkleron":
      print ("Sprinkler is on")
   elif status == "sprinkleroff":
      print ("Sprinkler is off")
   elif status == "exhaustfanon":
      print ("Exhaust Fan ON")
   elif status == "exhaustfanoff":
      print ("Exhaust Fan 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


    temp=random.randint(0,100)
    flame_level=random.randint(0,100)
    gas_level = random.randint(0,100)


    data = { 'Temperature' : temp, 'Flame_Level' : flame_level, 'Gas_Level' : gas_level }

    #print data

    def myOnPublishCallback():
      print ("Published Temperature = %s C" % temp, "Flame_Level = %s %%" % flame_level,
"Gas_Level = %s %%" %gas_level ,"to IBM Watson")

    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
    if not success:
      print("Not connected to IoTF")
    time.sleep(1)

    deviceCli.commandCallback = myCommandCallback


# Disconnect the device and application from the cloud

deviceCli.disconnect()
```

7.2 Feature 2

Output :

Published Temperature = 3 C Flame_Level = 88 % Gas_Level = 30 % to IBM Watson
Published Temperature = 22 C Flame_Level = 51 % Gas_Level = 16 % to IBM Watson
Published Temperature = 80 C Flame_Level = 32 % Gas_Level = 88 % to IBM Watson
Published Temperature = 98 C Flame_Level = 81 % Gas_Level = 34 % to IBM Watson
Command received: sprinkleroff
Sprinkler is off
Command received: exhaustfanoff
Exhaust Fan OFF
Command received: sprinkleron
Sprinkler is on
Published Temperature = 93 C Flame_Level = 77 % Gas_Level = 43 % to IBM Watson
Command received: exhaustfanon
Exhaust Fan ON
Published Temperature = 18 C Flame_Level = 37 % Gas_Level = 88 % to IBM Watson
Published Temperature = 61 C Flame_Level = 53 % Gas_Level = 65 % to IBM Watson
Published Temperature = 95 C Flame_Level = 76 % Gas_Level = 90 % to IBM Watson
Published Temperature = 56 C Flame_Level = 14 % Gas_Level = 27 % to IBM Watson
Published Temperature = 34 C Flame_Level = 33 % Gas_Level = 51 % to IBM Watson
Published Temperature = 9 C Flame_Level = 56 % Gas_Level = 80 % to IBM Watson
Published Temperature = 42 C Flame_Level = 51 % Gas_Level = 18 % to IBM Watson

8. TESTING

8.1 Test Cases

Date	[illegible]
Team ID	[illegible]
Project Name	Project - Industry Specific Intelligent [illegible]
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
TC_001	Functional	IBM Cloud	Create the IBM Cloud services which are being used in this project	IBM Cloud Login ID & Password	1.Go to IBM Cloud signup page 2.Enter e-mail id and other credentials 3.Enter a password	https://cloud.ibm.com/login	Should be able to create the IBM Cloud account	Working as expected	Pass	Results verified	No		Bhumika Janani
TC_002	Functional	IBM Cloud	Configure the IBM Cloud services which are being used in completing this project	IBM Cloud Login ID & Password	1.Go to Cloud login 2.Enter user ID & Password 3.Verify login by the [illegible]	https://cloud.ibm.com/login	Should able login to IBM Cloud and navigated to IBM Cloud dashboard page	Working as expected	Pass	Results verified	No		Saritha bhumika
TC_003	Functional	IBM Watson IoT Platform	IBM Watson IoT platform acts as the mediator to connect the web application to IoT devices, so create the IBM Watson IoT platform	IBM Watson IoT Platform Login ID & Password	1.Login to IBM Cloud 2.Click Catalog 3.Search IoT and click create 4.Go to resource list and search Internet of Things platform 5.Press launch and click Sign in IBM Watson Platform	[illegible]	Should be able to navigate to IBM IoT Watson Platform	Working as expected	Pass	Results verified	No		bhumika Keerthana janani
TC_004	Functional	IBM Watson	To create a device in the IBM Watson IoT platform and get the device credentials	IBM Watson IoT Platform Login ID & Password	1.Login to IBM Watson Platform 2.Click Add Device 3.Enter the details and click Finish. Note down the Device ID, Device Name, Authentication key, Organization name	Device credentials	Should be able to get Device details	Working as expected	Pass	Results verified	No		bhumika
TC_005	Functional	IBM Cloud Node Red	Configure the connection security and create API keys that are used in the Node-RED service for accessing the IBM IoT Platform	Node Red installation	1. Search "Node red" in catalog 2. Wait for some time to completely configure the Node-Red	[illegible]	Should be able to open Node-Red service	Working as expected	Pass	Results verified	No		Saritha A b
TC_006	Functional	Node Red	Create a Node-RED service	Node Red installation	1.Select IBM IoT input in Node- in IBM IoT Watson Platform, go to apps and click on generate API key. 2.Copy & paste generated API key and token in the IBM IoT input. After entering all details, click the done button. 3.Add debug to the IBM IoT and [illegible] click on done. Click gauge from the dashboard and fill the details & add functions to the gauge. Check [illegible]	Values of sensors and button for alarm & Sprinkler ON/OFF is displayed	Values of sensors and button for alarm & Sprinkler ON/OFF should be displayed	Working as expected	Pass	Results verified	No		bhumika.s

Date	[illegible]
Team ID	[illegible]
Project Name	Project - Industry Specific Intelli[illegible]
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
TC_007	Functional	Python 3.7.0	Develop a python script to publish random sensor data such as temperature, humidity level and Gas levels to the IBM IoT platform	Python 3.7 IBM IoT installation	1.Download and install Python 3.7.0 2.Develop python code	https://www.python.org/downloads/release/python-370	Should be develop a python script that can randomly generate and send Temperature, Gas level and Flame level values to the IBM IoT Watson Platform	Working as expected	Pass	Results verified	No		S.bhumika
TC_008	Functional	Python 3.7.0	After developing python code, commands are received just print the statements which represent the control of the [illegible]	Python 3.7 IBM IoT installation	1.Download and install Python 3.7.0 2. Open Node-Red to MIT mobile app	Verifie the output from the code	Should be able to display the commands like Sprinkler ON, Sprinkler OFF, Exhaust Fan ON, Exhaust Fan OFF	Working as expected	Pass	Results verified	No		K.janani A b Saritha
TC_009	Functional	IBM Cloudant DB	Store the sensor values - Temperature, Flame Level and Gas Level in the Cloud	IBM Cloud Account	1.Run the python code 2.Verify the displayed output	Output from the python code	Should be able to store the sensor values generated by the python script in the cloud	Working as expected	Pass	Results verified	No		S.Bhumika
TC_010	Web UI	Node Red & MIT Inventor	Create Web UI in the Node Red	MIT Inventor Login ID & password	[illegible]	Sensors values and command values displayed in the Debug window and in the mobile application	Sensors values and command values is displayed in the Debug window and the User should be able to view these data in the MIT Mobile APP and be able to press the buttons if any value exceeds the threshold value	Working as expected	Pass	Results verified	No		Keerthana bhumika janani
TC_011	Functional	IBM Cloudant DB	Configure the Node-RED flow to receive data from the IBM IoT platform and also use Cloudant DB nodes to store the received sensor data in the cloudant DB	IBM Cloud Login ID & Password	1.Go to IBM cloud, search Cloudant in Catalog. Add new dashboard, go to Node Red 2.Connect to cloudant and verify the results	Cloudant is connected to the NODE RED	User should be able to connect the Cloudant and Node Red and be able to see the created cloud database with the sensor values	Working as expected	Pass	Results verified	No		S Bhumika

8.2 User Acceptance Testing

Purpose of Document : The purpose of this document is to briefly explain the test coverage and open issues of the Industry-specific intelligent fire management system project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis :

Section	Total Cases	Not Tested	Fail	Pass
Print the Sensor values	7	0	0	7
Client Mobile Application	51	0	0	51
Security	2	0	0	2

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	0	0	1	8
Totals	24	14	13	26	70

Test Case Analysis

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

Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4

Version Control	2	0	0	2

9. RESULTS

9.1 Performance Metrics

NFT - Risk Assessment

S.No	Project Name	Scope/Feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Volume Changes	Risk Score	Justification
1	Receiving sensor v	Existing	Moderate	No Changes	Moderate	No	>5 to 10%	ORANGE	As we have seen the changes
2	Sprinkler ON/OFF	Existing	Low	No Changes	Low	No	>5 to 10%	GREEN	As we have seen the changes
3	Exhaust Fan ON/Off	Existing	Low	No Changes	Low	No	>5 to 10%	GREEN	As we have seen the changes
4	Fast SMS	New	Low	No Changes	No Changes	No	>5 to 10%	GREEN	As we have seen the changes
5	Cloudant Database	New	No Changes	No Changes	No Changes	No	>5 to 10%	GREEN	As we have seen the changes

NFT - Detailed Test Plan

S.No	Project Overview	NFT Test approach	umptions/Dependencies/Ri	Approvals/SignOff
1	Python 3.7.0	Developing Python Scr	Depends on the code	[illegible]
2	IBM Watson IoT Platform	Creating and configur	Depends on the Device Cred	[illegible]
3	Node-Red	Creating Web-UI	Depends on the sensor valu	https://nodered.org/
4	MIT App Developer	Developing Mobile ap	Depends on the Sensor valu	[illegible]
5	Cloudant DB	Storing Sensor values	Depends on the Sensor valu	[illegible]

End Of Test Report

S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Recommendations	Identified Defects (Detected/Closed/Open)	Approvals/SignOff
1	Janel sensor and te	This is done by deve	Met	Pass	GO	Code working properly	Closed	[illegible]
2	Based on the temp	This is done by creat	Met	Pass	GO	Sprinkler is turning on and o	Closed	[illegible]
3	If any flame is dete	This is done by creat	Met	Pass	GO	Exhaust fan is turning on an	Closed	[illegible]
4	Emergency alerts are notified to the auth		Met	Pass	GO	Emergency alerts are send vi	Closed	[illegible]

10. ADVANTAGES & DISADVANTAGES

The Advantages of this Industry-Specific Intelligent Fire Managment system are as follows

- The user need not require expertise knowlege to control this system. This system is simple. The user can easily view the sensor values and take control actions.
- The control actions are taken automatically.
- If it is implemented in hardware, then the cost of implemention will be affordable.
- As we are sensing the sensor values continously, any slight change in the environment is detected
- This system is in User-Friendly format.

The Disadvantage of this Industry-Specific Intelligent Fire Managment system are as follows

- This system will not be able to detect the orgin of fire.
- This system will not provide the escape route if there is fire outbreak.
- If the industry has specific changes in the environment, then this system will gives false alarm.

11. CONCLUSION

An understanding and having Fire Managment system in the industry is of utmost importance.This project is a fire management system that can be user in the industry based on IOT.This system creates a simulation device cedentials in IBM WATSON IOT PLATFORM.In node-red,necessary nodes are installed and used.These nodes are installed and used.These nodes are deployed and the data is collected.In the event of fire, this system can issue sprinkler on,exhaust fan on.This remote user monitoring system can monitor the system status of each node in real time.This system monitors the data continuously so that the any slight change in the environment can be easily detected.This ensures good control accracy .This Industry-SpecificIntelligent Fire Managment ensures the protection of property, asset and the processes are costeffective and the automatic measures are in control.

12. FUTURE SCOPE

The future scope of this project is to add additonal features like triggering the extinguisher automatically, predict the escape route if the fire outbreaks and to implement this system in real time using hardware.

13.APPENDIX

Source Code

https://github.com/IBM-EPBL/IBM-Project-51318-1660978079

GitHub & Project Demo Link

THANK YOU