IBM-Project-17749-1659675804

Industry-specific Intelligent Fire Management System

TEAM ID: PNT2022TMID31718 **TEAM SIZE: 4**

Team Leader: Sneha P

Team Member 1: Radhika T

Team Member 2: Sethu Harshini V

Team Member 3: Sandhya S

Title
1. INTRODUCTION
a. Project Overview
b. Purpose
2. LITERATURE SURVEY
a. Existing problem
b. References
c. Problem Statement Definition
3. IDEATION & PROPOSED SOLUTION
a. Empathy Map Canvas
b. Ideation & Brainstorming
c. Proposed Solution
d. Problem Solution fit
4. REQUIREMENT ANALYSIS
a. Functional requirement
b. Non-Functional requirements
5. PROJECT DESIGN
a. Data Flow Diagrams
b. Solution &Technical Architecture
c. User Stories
6. PROJECTPLANNING & SCHEDULING
a. Sprint Planning & Estimation
b. Sprint Delivery Schedule
c. Reports from JIRA
7. CODING & SOLUTIONING
a. Feature 1
b. Feature 2

INTRODUCTION

Project Overview

Nowadays Internet of things can be anything in the world that actually gathering or collecting everything in our world to basically connect all things to the internet. All connected things are then being used to make a group of information or sending information or it can be for both processes in this system. Safety is significant and it is vital that acceptable wellbeing framework be executed in the spots of all fields. This system is used in buildings and home dwellings for the fire detection and prevention purpose. It should be implemented in all the establishments where the risk of fire accident is very high. The sensor nodes are placed in important areas of building, which we create a network and the monitored data is transmitted to control unit through wireless sensor network and if the temperature or pressure reach above the threshold value and building damage is detected automatically, alerts the surroundings and take necessary precautions to prevent the disaster. This safety system that can be used in any constructing and constructed environments. The sensor node detects the maximum level that it can withhold, in the meantime it calculates where the damage is occuring and remaining time that the building can offer further resistance to damage.

Project Purpose

The objective of “Industry specific-intelligent fire management system” is to avoid the unintended fire accidents in industries and also take appropriate measures to avoid any mishap. The smart fire management system includes a Gas sensor, Flame sensor and temperature sensors to detect any changes in the environment. If any flame is detected the sprinklers will be switched on automatically. The model incorporates MQ2 gas sensor for detecting propaneand methane gases,flame is detectedby IR flame sensor module and LM35 Temperature Sensor for the measurement of the environment. These readings are monitored

continuously by IBM Watson IOT Platform and stored in Cloudant DB. Based on the temperature readings and if any Gases are present, the exhaust fans are powered ON. In case any variation occurs, the authorities and fire station will be alerted via Fast2SMS web service. Emergency alerts are notified to the authorities and Fire station.

LITERATURE SURVEY

Existing Problem

Smart buildings are among the most innovative solutions for engineers to ensure social and environmental respponsibility and provide safe and secure environments for occupants. Emerging technologies when aligned together to complement each other , can deliver the promise of enhanced fire safety, enabling the promise of smart buildings and cities that are safer.

References

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

2. Ondrej Krejcar, "Using of mobile device localization for several types ofapplications in intelligent crisis management",5th IEEE GCC Conference & Exhibition, IEEE, 2009

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

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

5. Al Mamari, A. R. M. H., Al Mamari, H., Kazmi, S. I. A., Pandey, J., & AlHinai, S. (2019). *IoT based*

Smart Parking and Traffic Management System for Middle East College.

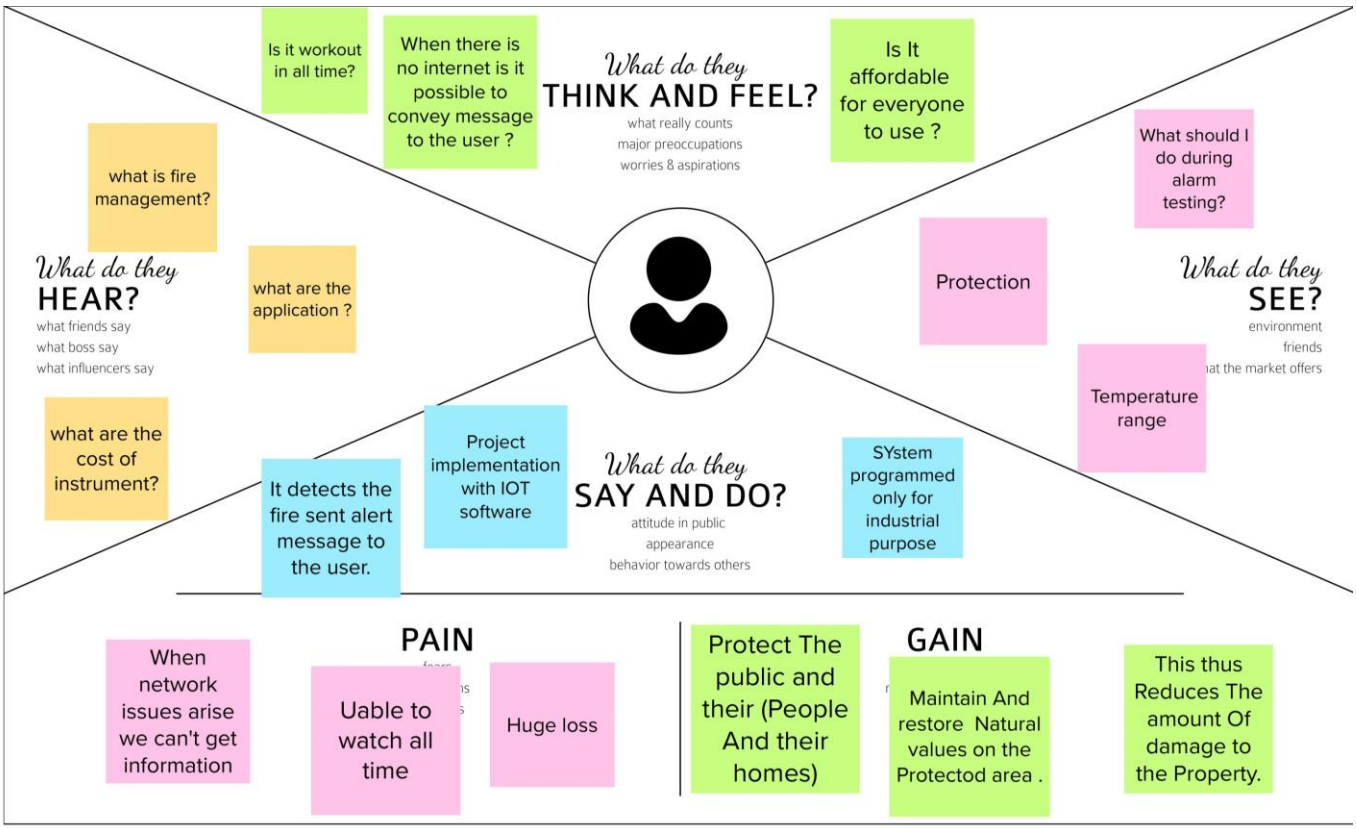
Problem statement Definition

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
Identify the chemical Explotion and fire in industry	A customer	Ensure the fire safety in my industry	I want to handle it easily	The fire explosion should be automatically detected without ant human monitoring	That my industry and my workers are safe
I can avoid the fire explosion in my industry and keep all my products safe from the fire spread	An industrialist	Work in chemical industry	It is not safe due to harmful effect	Fire and gas Leak safety Measure	To reduce the explosion

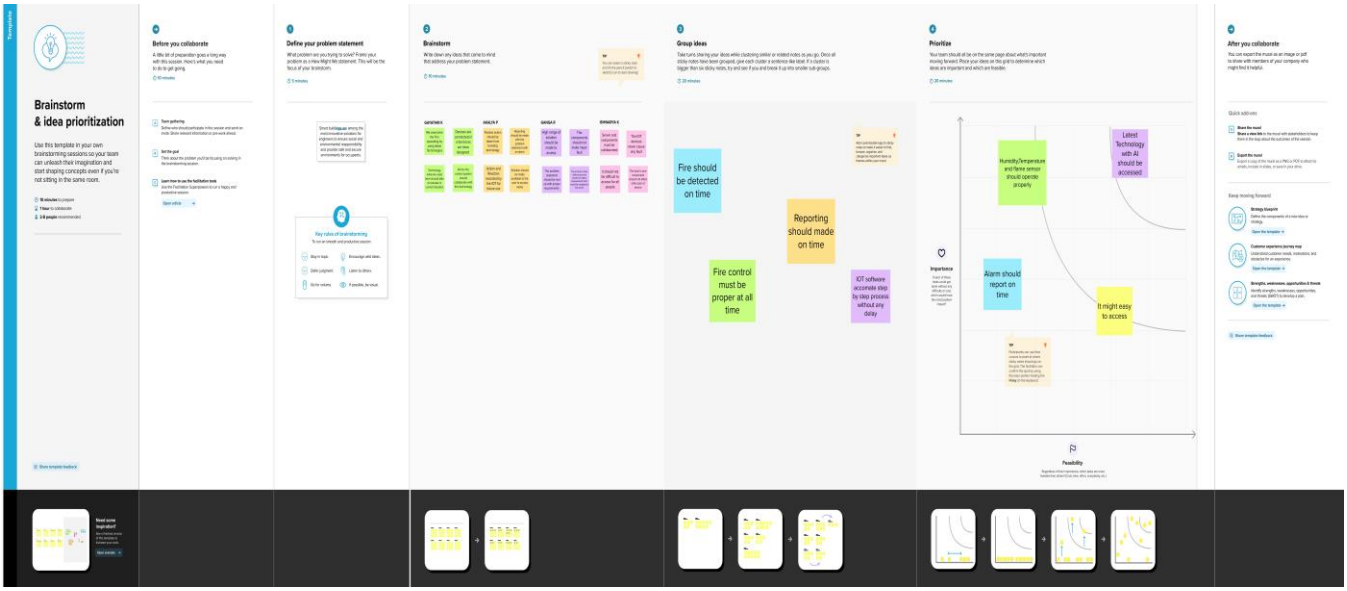
Fire detection Using Arduino uno and flame,gas, temperature sensors	A worker	To find the fire detection place/location	Some difficult in find location	The temperature Will increase Sometimes	While using GPS to find the locations

IDEATION & PROPOSED SOLUTION

Empathy Map



Ideation & Brainstorm

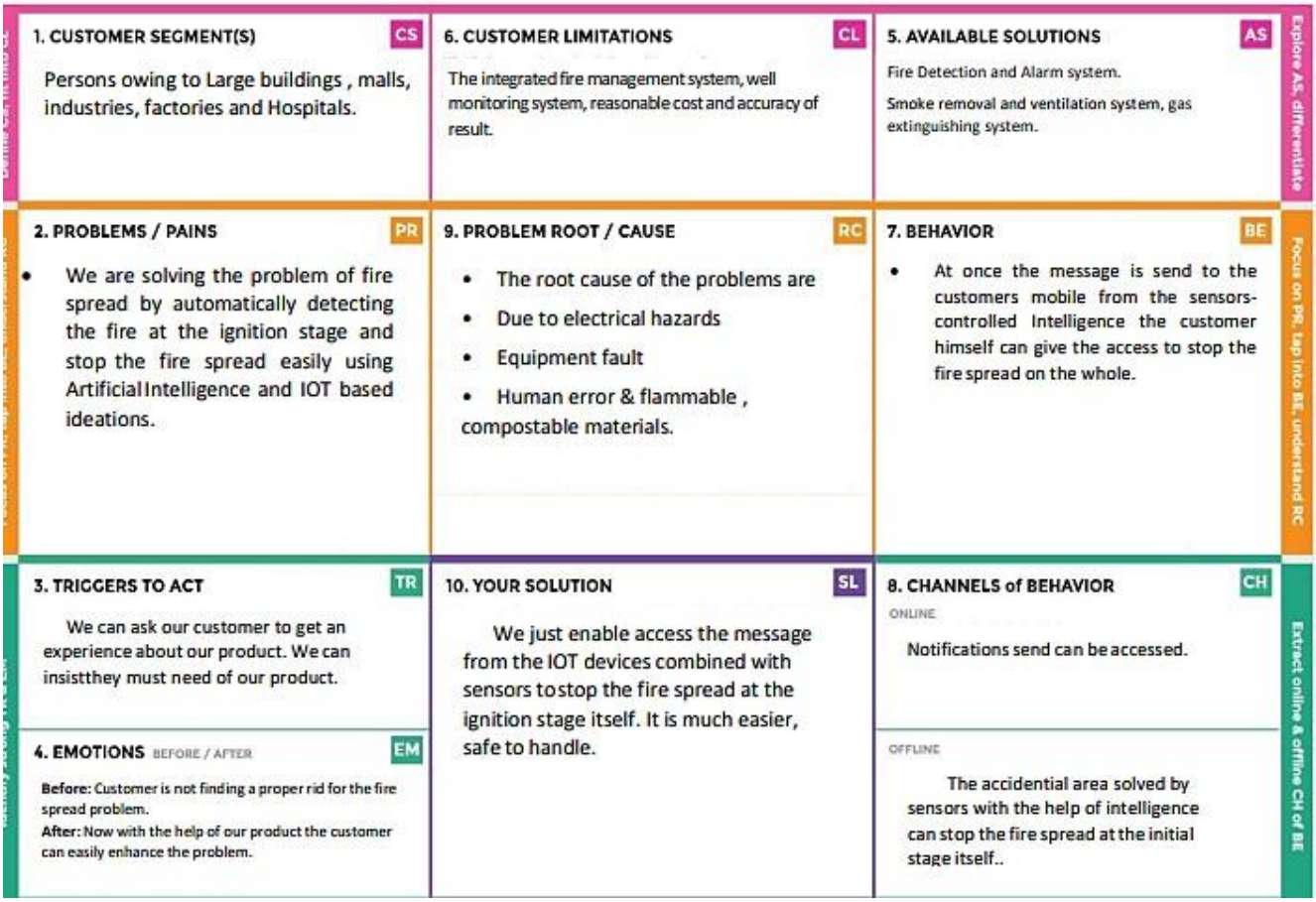


Proposed Solution

S.no	Parameter	Description
1	Problem Statement (Problem to be solved)	We are ready to solve the unintended fire explosion in the industry level.
2	Idea / Solution description	Idea for thisproblem titled Industry Specific Intelligent Fire Management System. The solution is to detect the fire before it becomes huge explosion in the industry as well as in public places.
3	Novelty / Uniqueness	By using Latest Technology Artificial Intelligence to answer and solve the fire explosion without Human presence.

4	Social Impact / Customer Satisfaction	The AI detects and senses the fire using many sensors that we use and it helps the customers to access with the immediate notification and the timely access.
5	Business Model (Revenue Model)	This model is used to calculate the probability of the ignition level and check how long it spread across a landscape.
6	Scalability of the Solution	The System is completely modular make it expandable and business efficiency in customized fire detection , with affordable price.

Problem Solution Fit



REQUIREMENT ANALYSIS

Solution Requirements (Functional & Non-functional)

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 is done through Gmail which is available in the playstore .
FR-2	User Confirmation	Confirmation via Email as a invitation Confirmation via OTP through user's mobile number.
FR-3	User Login	It is necessary to Login through website or App using the respective username and password given by the user.
FR-4	User Access	User might allow all the requirements for better experience.
FR-5	User Guide	Guides the basic steps of using the application.
FR-6	User Upload	User should be able to send the data
FR-7	User Solution	Data report should be generated and delivered to user for per every 24 hours
FR-8	User Data Sync	API interface to increase to invoice system

Non-functional Requirements:

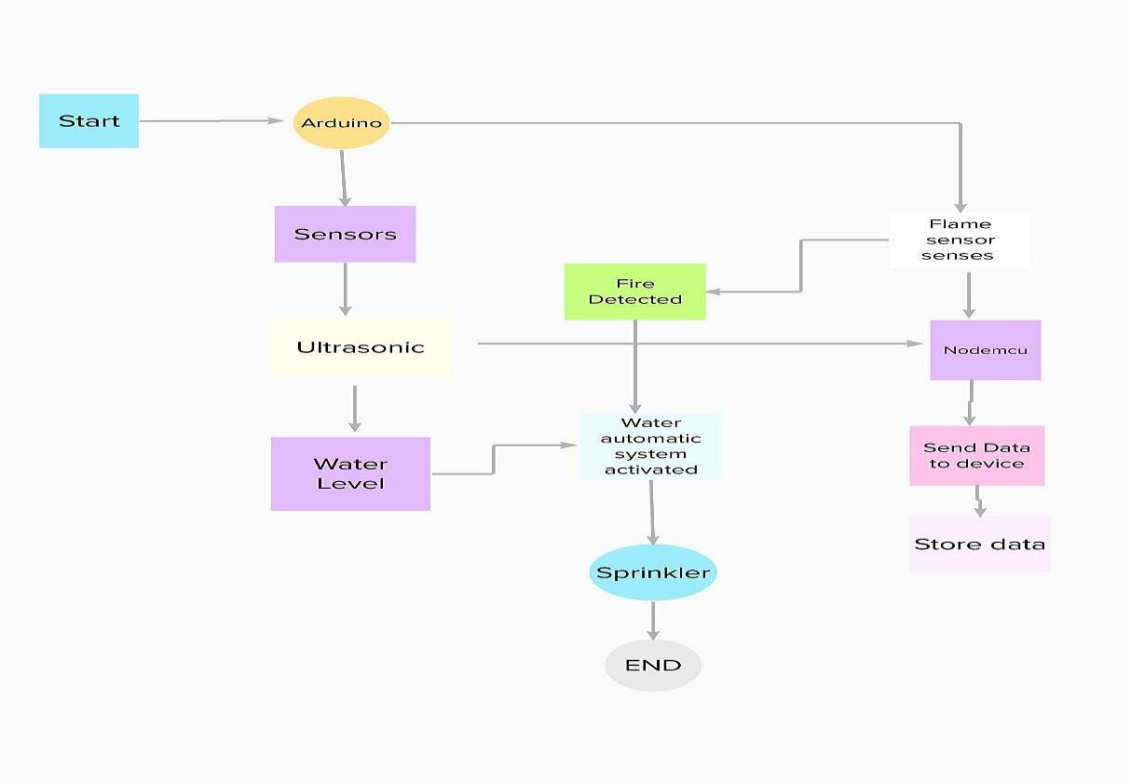
Following are the non-funconal requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Usability requirements includes barriers in language and localization
		tasks. Easy for everyone to access.

NFR-2	Security	Access permissions for the particular system information may only be changed by the system's data administrator.
NFR-3	Reliability	The database update process must roll back all related updates when any update fails.
NFR-4	Performance	The front-page load time must be no more than 2 seconds for users that access the website using an VoLTE mobile connection.
NFR-5	Availability	New module deployment must not impact front page, product pages, and check out pages availability and mustn't take longer than one hour.
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.

PROJECT DESIGN

Data Flow Diagram



USER STORIES

User Type	Functional requirement	User story number	User story/ta sk	Acceptan ce criteria	Priority	Release

Customer (Mobile user, web user; care executive,Administrator)	Registration	USN-1	As a user, I can registerfor the application by entering my mail, password, and confirming my password	I can access my account/dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email& click confirm	High	Sprint-1

	Dashboard	USN-3	As a user, I can register for the application through internet	I can register & access the dashboard with Internet login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can confirm the registeration in Gmail	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email& password	I can login with my id and password	High	Sprint-1

PROJECT PLANNING & SCHEDULING

Sprint Planning & Scheduling

Sprint	Functional Requir ement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Login	USN-1	As a customer , I might ensure login credential through gmail ease manner for the purpose of sending alert messageto the owner.	2	High	Gayathri
Sprint-1	Registration	USN-2	As a user , I have to registered my details and tools details in a simple and easy manner in case of fire incident, this registered system sends notification to the industrialist.	2	High	Akalya
Sprint-2	Dashboard	USN-3	As a user, In case of Fire in the industry I need the sprinkler to spray wateron the	2	Low	Ganga

			existing fire automatically.			
Sprint-1	Dashboard	USN-4	As a user , I need to safeguard my properties as well as and it will be better to send alertmessage to the fire department.	2	Medium	Ishwarya
Sprint-1	Dashboard	USN-5	As a user , Its good to havea IOT basedsystem to extinguish the fire without humanpresence.	2	High	Gayathri

Sprint Delivery Schedule

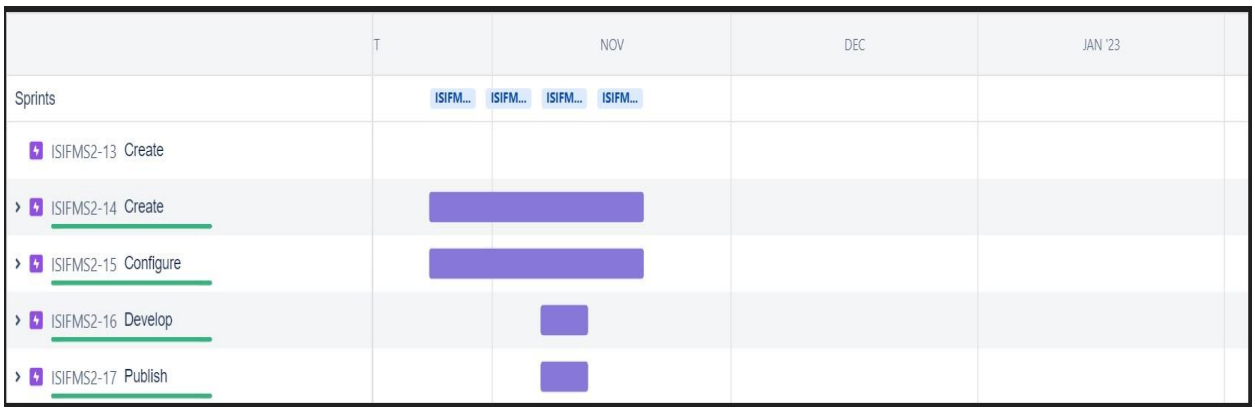
Sprint	**Total Story Points**	**Durati on**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story Points Completed (as on Planned End Date)**	**Sprint Release Date (Actual)**
Sprint-1	20	6 Days	24Oct 2022	29 Oct2022	20	29 Oct2022

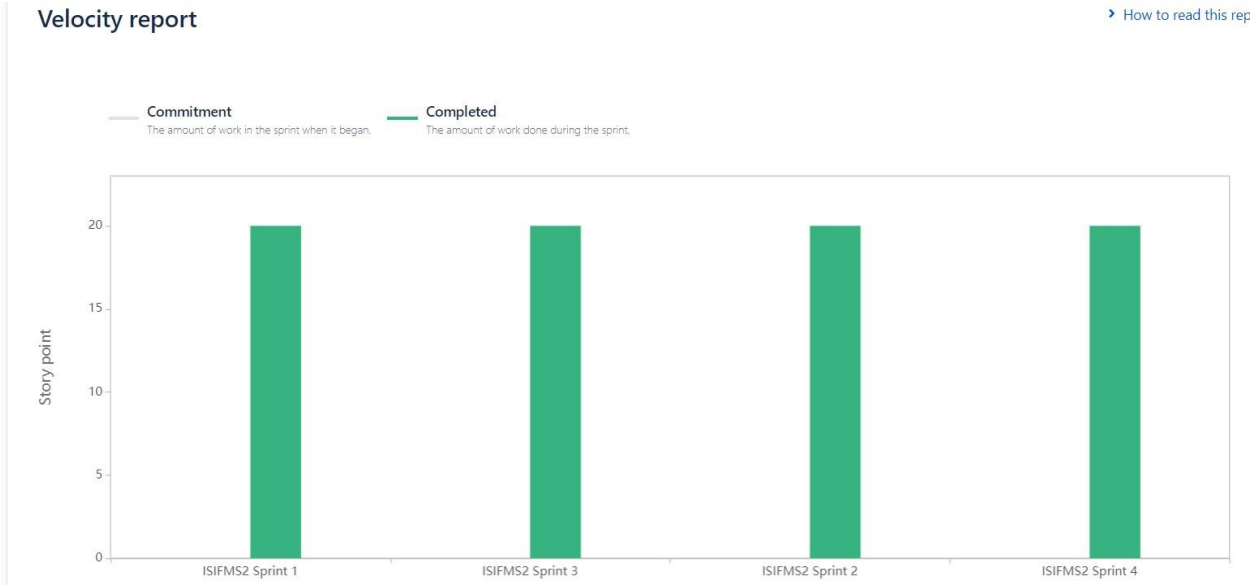
Sprint-2	20	6 Days	31Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07Nov	12 Nov 2022	20	12 Nov 2022
			2022			
Sprint-4	20	6 Days	14Nov 2022	19 Nov 2022	20	19 Nov 2022

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)

Reports from JIRA


T
NOV
DEC
JAN '23
Sprints
ISIFM...
ISIFM...
ISIFM...
ISIFM...
ISIFMS2-13 Create
ISIFMS2-14 Create
ISIFMS2-15 Configure
ISIFMS2-16 Develop
ISIFMS2-17 Publish


Velocity report
How to read this rep
Commitment
The amount of work in the sprint when it began.
Completed
The amount of work done during the sprint.
Story point
20
15
10
5
0
ISIFMS2 Sprint 1
ISIFMS2 Sprint 3
ISIFMS2 Sprint 2
ISIFMS2 Sprint 4

CODING & SOLUTIONING

Feature 1

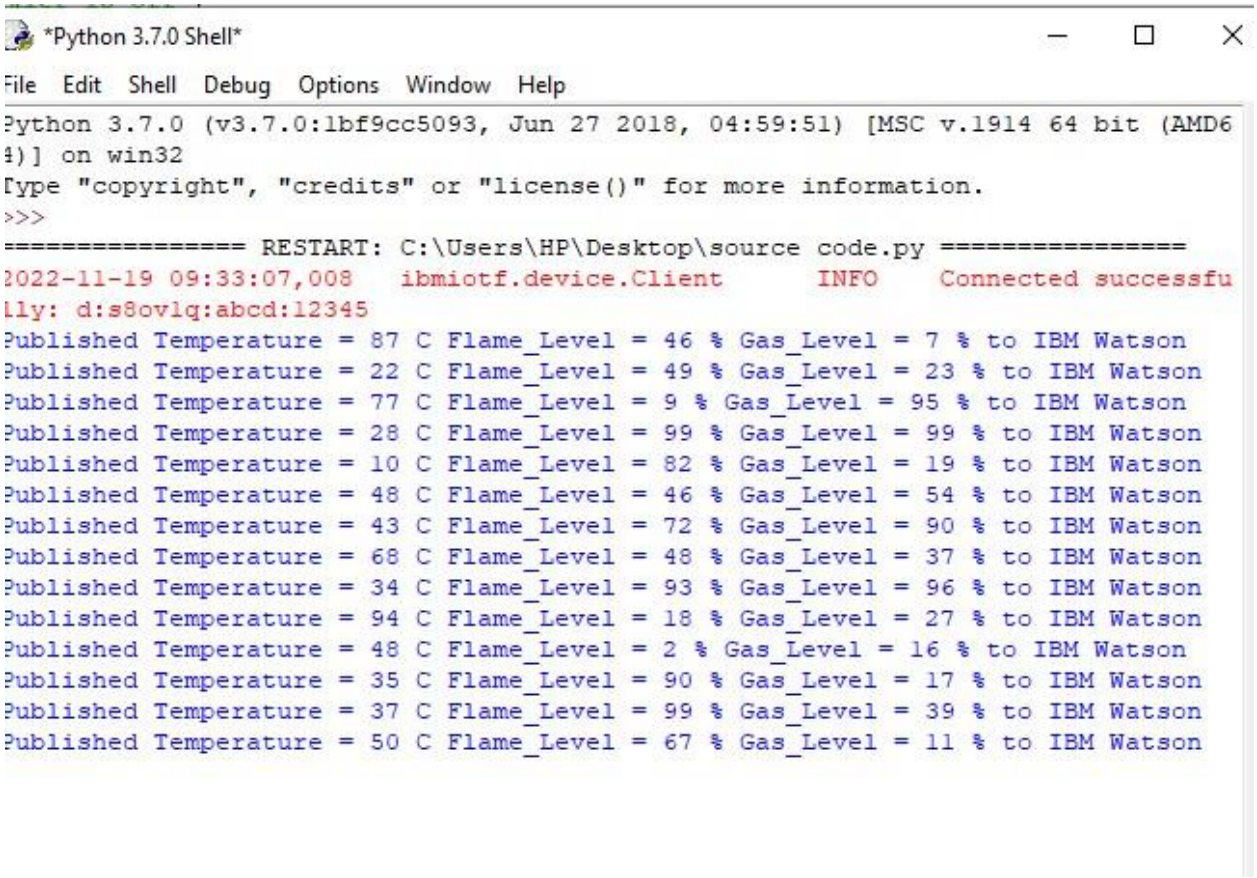
source code.py - C:\Users\HP\Desktop\source code.py (3.7.0)

File Edit Format Run Options Window Help

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


#Provide your IBM Watson Device Credentials
organization = "s8ovlq"
deviceType = "abcd"
deviceId = "12345"
authMethod = "token"
authToken = "12345678"

# 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()
```

Feature 2


Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\HP\Desktop\source code.py ==========
2022-11-19 09:33:07,008 ibmiotf.device.Client INFO Connected successfully: d:s8ovlq:abcd:12345
Published Temperature = 87 C Flame_Level = 46 % Gas_Level = 7 % to IBM Watson
Published Temperature = 22 C Flame_Level = 49 % Gas_Level = 23 % to IBM Watson
Published Temperature = 77 C Flame_Level = 9 % Gas_Level = 95 % to IBM Watson
Published Temperature = 28 C Flame_Level = 99 % Gas_Level = 99 % to IBM Watson
Published Temperature = 10 C Flame_Level = 82 % Gas_Level = 19 % to IBM Watson
Published Temperature = 48 C Flame_Level = 46 % Gas_Level = 54 % to IBM Watson
Published Temperature = 43 C Flame_Level = 72 % Gas_Level = 90 % to IBM Watson
Published Temperature = 68 C Flame_Level = 48 % Gas_Level = 37 % to IBM Watson
Published Temperature = 34 C Flame_Level = 93 % Gas_Level = 96 % to IBM Watson
Published Temperature = 94 C Flame_Level = 18 % Gas_Level = 27 % to IBM Watson
Published Temperature = 48 C Flame_Level = 2 % Gas_Level = 16 % to IBM Watson
Published Temperature = 35 C Flame_Level = 90 % Gas_Level = 17 % to IBM Watson
Published Temperature = 37 C Flame_Level = 99 % Gas_Level = 39 % to IBM Watson
Published Temperature = 50 C Flame_Level = 67 % Gas_Level = 11 % to IBM Watson

TESTING

Test Cases

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)
TC_001	Functional	IBM cloud	Create the IBM Cloud services which are being used in this project	IBM Cloud login ID &password	Go to IBM cloud signup page.Enter e-mail id and other credential.Enter a password	https://cloud.com/login	Login/Signup popup should display	Working as expected	Pass	results verified	No
TC_002	Functional	IBM cloud	Configure the IBM cloud services which are being used in completing this project	IBM Cloud login ID &password	Go to Cloud login.Enter user id & password.verify the login by the popup display	https://cloud.com/login	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create account link e.Last password? Recovery password link	Working as expected	Pass	results verified	No
TC_003	Functional	IBM Watson IOT Platform	IBM WAtson IOT platform acts as mediator to connect the web application to IOT devices, so create the IBM WAtson	IBM Watson IOT Platform login id &password	login to IBM cloud . click catalog, search IOT and click create.Go to resourse list and search IOT platform	https://vq4nsy.internetofthings.ibmcloud.com/dashboard/	User should navigate to user account homepage	Working as expected	Pass	results verified	No
TC_004	Functional	IBM Watson IOT Platform	To create a devicein the IBM Watson IOT platform and get the device credential	IBM Watson IOT Platform login id &password	login to IBM Watson platform click Add Device Enter the details and click finish . Note down the Device ID,device name,authentication key,organisation name	Device credentials	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass	results verified	No
TC_005	Functional	IBM cloud	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	search node-red in catalog wait for some time to completely configure the node-red	https://cloud.ibm/developer/appservice/create-app?starterkit=59c9d5bd-4d31-3611-897a-f84eea80bc9f&defaultLanguage=undefined	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass	results verified	No
TC_006	Functional	Node Red	create a Node-Red service	Node-RED installation	select IBM IOT input in node in IBM IOT watson platform go to apps and click on generate api keys copy and paste generated api key and token in IBM IOT input after after entering all details click the done button	values of sensors and button for alarm & sprinkler ON/OFF is displayed	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass	results verified	No

User Acceptance Testing

Purpose of Document : The purpose of this documentis to briefly explain the test coverageand 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	TotalCases	NotTested	Fail	Pass
Print the Sensor values	7	0	0	7
Client MobileApplication	51	0	0	51
Security	2	0	0	2

This reportshows the numberof 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 reportshows the number of test cases that have passed,failed, and untested

Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final ReportOutput	4	0	0	4

RESULTS

Performance Metrics

NFT - Risk Assessment

S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Voluem Changes	Risk Score	Justification
1	Receiving sensor va	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/OFI	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	sumptions/Dependencies/Ri	Approvals/SignOff
1	Python 3.7.0	Developing Python Scr	Depends on the code	https://www.python.org/psf/sponsors/#heroku
2	IBM Watson IoT Platform	Creating and configuri	Depends on the Device Cred	https://4aqwut.internetofthings.ibmcloud.com/dashboard/
3	Node-Red	Creating Web-UI	Depends on the sensor valu	https://nodered.org/
4	MIT App Devoloper	Developing Mobile ap	Depends on the Sensor valu	https://appinventor.mit.edu/about/termsofservice
5	Cloudant DB	Storing Sensor values	Depends on the Sensor valu	https://2587b83c-debe-4618-8ea6-c3bdd6111fb4-bluemix.cloudant.com/dashboard.html

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	ame sensor and ter	This is done by devel	Met	Pass	GO	Code working properly	Closed	https://www.python.org/psf/sponsors/#heroku
2	Based on the temp	This is done by creat	Met	Pass	GO	Sprinkler is turning on and o	Closed	http://159.122.183.108:32627/red/#flow/51cd2ad32ac08578
3	If any flame is dete	This is done by creat	Met	Pass	GO	Exhaust fan is turning on an	Closed	http://159.122.183.108:32627/red/#flow/51cd2ad32ac08578
4	Emergency alerts are notified to the auth		Met	Pass	GO	Emergency alerts are send vi	Closed	https://www.fast2sms.com/dashboard/sms/bulk

ADVANTAGES & DISADVANTAGES

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

1. The user need not require expertise knowledge to controlthis system. This system is simple. The user can easily view the sensor values and take controlactions.
2. The control actions are taken automatically.
3. If it is implemented in hardware, then the cost of implementation will be affordable.
4. As we are sensing the sensor values continuously, any slight changein the environment is detected
5. This system is in User-Friendly format.

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

6. This systemwill not be able to detect the origin of fire.
7. This systemwill not providethe escape route if thereis fire outbreak.
8. If the industryhas specific changesin the environment, then this system will gives falsealarm.

CONCLUSION

An understanding and having Fire Management system in the industry is of utmost importance.This project is a fire management systemthat can be user in the industrybased on IOT.This system creates a simulation device credentials 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 accuracy .This Industry- Specific Intelligent Fire Management ensures the protection of property, asset and the processes are cost effective and the automatic measures are in control.

FUTURE SCOPE

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

APPENDIX

Solution Architecture Diagram:

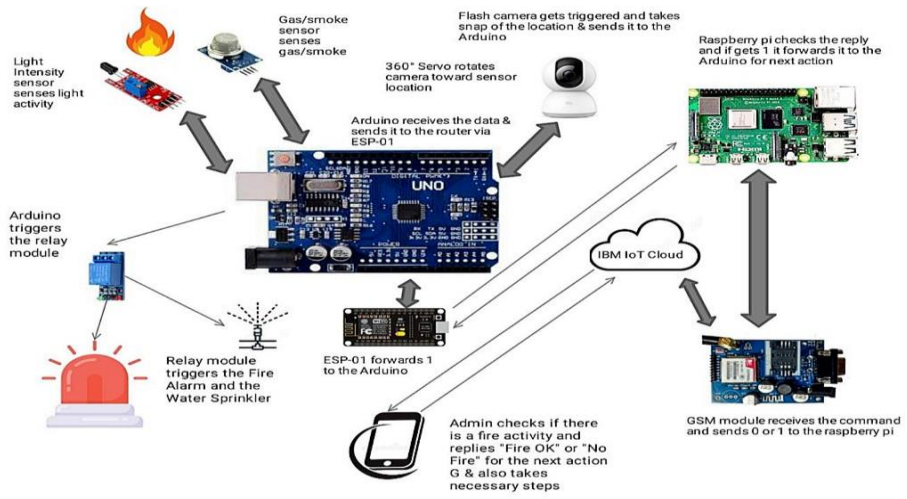


Fig : Technology architecture of our project

Source Code

```
import me import sys

import ibmio .applica on

import ibmio .device

import random



#Provide your IBM Watson Device Creden als

organiza on = "s8ov1q" deviceType = "abcd"

deviceId = "12345" authMethod = "token"

authToken = "12345678"



# Ini alize 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 == "exhaus anon":         print ("Exhaust Fan ON")
elif status == "exhaus anoff":         print ("Exhaust Fan
OFF")

   #print(cmd)


try:
          deviceOp ons = {"org": organiza on, "type": deviceType, "id": deviceId, "auth-method":
authMethod, "auth-token": authToken}
          deviceCli = ibmio .device.Client(deviceOp ons)
          #.....


except Excep on as e:
          print("Caught excep on connec ng device: %s" % str(e))
sys.exit()


# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "gree ng"
10 mes
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")
me.sleep(1)


        deviceCli.commandCallback = myCommandCallback


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

GitHub & Project Demo Link

GitHub

IBM-Project-17749-1659675804

Project Demo Link

https://drive.google.com/drive/folders/1KCmqGy6QU4vGOVYNlXmbZkJjDOFf8vdB?usp=share_link