Gas Leakage Monitoring &Alerting System For Industries

IBM-Project-45678-1660731582

NALAIYA THIRAN PROJECT BASED LEARNING ON PROFESSIONAL READLINESS FOR INNOVATION, EMPLOYNMENT AND ENTERPRENEURSHIP

A PROJECT REPORT

Submitted by

VIKASHINI A - 814419106006

VARSHA A - 814419106005

UDHAYAKUMAR S - 814419106301

RAM PRAKASH G – 814419106004

AKILAN G- 814419106302

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION

ENGINEERING

SUDHARSAN ENGINEERING COLLEGE –
PUDUKKOTTAI,622501

ANNA UNIVERSITY : CHENNAI 600 025

NOVEMBER 2022

ANNA UNIVERSITY : CHENNAI 600 025

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Project Overview

- Users can monitor the gas levels present in the surrounding environment using mobile application.
- Alertion is indicated with a buzzer, LED, and LCD Display when the gas level reaches the threshold value.
- Saving and Visualization of gas, temperature, humidity in IBM Watson IOT Platform.
- Connection with IOT device and MIT App Inventor is done by using Node Red Flow.

1.2 Purpose

To ensure the safety of workers in industries and to avoid any other accidental situations. The main aim is to provide them a safe working place and make them feel fear free.The purpose of this system is to detect gas leakage, neutralize it, and prevent any other explosion.This provides ability to predict hazardous situation so that worker could be made aware in advance by performing data analytics on sensor readings.This work modifies the existing safety model installed in industries and this system also be used in homes and offices.

2. LITERATURE SURVEY

2.1 Existing Problem

Normally in surrounding environment there is a presence of gases which is not harmful. But, the presence of hazardous gas leakage in a work place occurs from stored gases i.e container gas is not safe.The workers feels unsafe to work in that environment. This affects their mental strength and their works in industries.And this may also affect the lives of many people.In case of severe release it may lead to many hazardous damage to both industry and Human.

2.2 References

1." Design and Fabrication of a Carbon Dioxide Gas Leak Detection System for Oil and Gas Facilities",Chibuzor Amaobichukwu,2021.

2. "The monitoring system of leakage accidents in crude oil pipelines and LPG Gas leakage detection using GSM module",J Ding and J. Wang and Q.Pan ,2022.

3. "Internet of things (IOT) based gas leakage monitoring and alerting system with MQ-2 sensor ",Rohan Chandra Pandey, Manish Verma, Lumesh Kumar Sahu and Saurabh Deshmukh ,2017.

4. "Low-Cost Gas Leak Detection and Surveillance System for Single Family Homes", Ximena Perez-Palomino, Karina Rosas-Paredes, Jose Esquicha-Tejada,2022.

5. "Gas Leakage with Auto Ventilation and Smart Management System",AfsanaMim Anika, Ms.NasrinAkter, Md.Niamul Hasan,Jannatul Ferdous Shomaand Abdus Sattar, 2021.

6. "Leak Detection in Natural Gas Pipelines Using Intelligent Models",Oluwatoyin Akinsete and Adebayo Oshingbesan,2017.

7. "Gas Leakage Detector and Monitoring System",Yekini N.Asafe, Akinade O. Abigael, Oloyede A. Olamide,2022.

8. "Probabilistic Multiple Model Neural Network Based Leak Detection System", Mohammad Burhan Abdulla and RandaHerzalla,2015.

9. "A Secured Model of Iot Based Smart Gas Detection and Automatic Alarm System",Methila Farzana Woishe, Nila Sultana, Tamanna Zaman Bristy, Nafiza Anjum Khan, MD. Taimur Ahad,2022.

10. "An Intelligent Gas Leak Detection System Using Iot",Pankaj B Thote, Mohammad Ashar, Christie Anil Joseph, VishantG.Naik and Priya Bambal,2021.

2.3 Problem Statement Definition

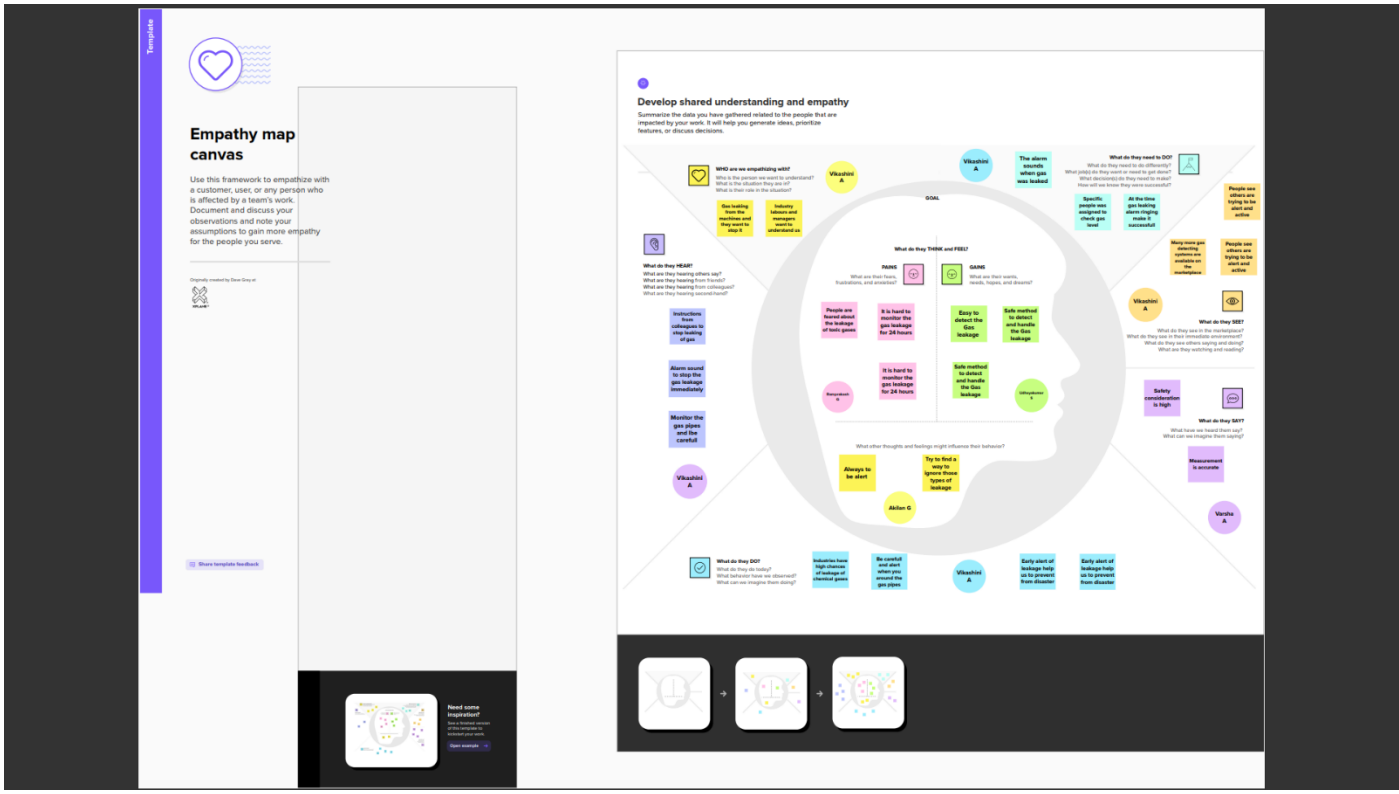
To identify and manage the gas leakage by detecting the toxic gas concentration and alerting the person to be get saved and reduce the gas leakage accidents. The high concentration of toxic gas from leakage can affect Central Nervous System and leads to several diseases who are working in industries.

Who does the problem affect?	The people who are working in the industries and factories of all generations are affected.
What are the boundaries of the problem?	Leakage diffusion characteristics and harmful boundary analysis of buried natural gas pipeline under multiple

	working conditions. Less accuracy, particular gas sensors and some instruments issues.
What is the issue?	These leaks can lead to physical symptoms and, in some cases, the gas can cause carbon monoxide poisoning in people and animals.
When does the issue occur?	When the concentration of toxic gases increases in the human body breathing difficulties such as dizziness, fatigue and irritation of eyes and throat.
Where is the issue occurring?	The gas leakage occurs mainly in factories and industries. This is due to any cracks occurring in gas storage and leads to accidents. It affects the whole body especially the respiratory system and the central nervous system. It even leads to death
Why is it important that we fix the problem?	Gas leak detection services are necessary mainly for health reasons. Combustible gases can start fires or cause explosions in homes and workplaces, injuring you, family members, friends, or workers. Gas leaks can also cause health issues in the long term from gradual exposure. Some gas leaks are even silent killers.

3. IDEATION & PROPOSED SOLUTION

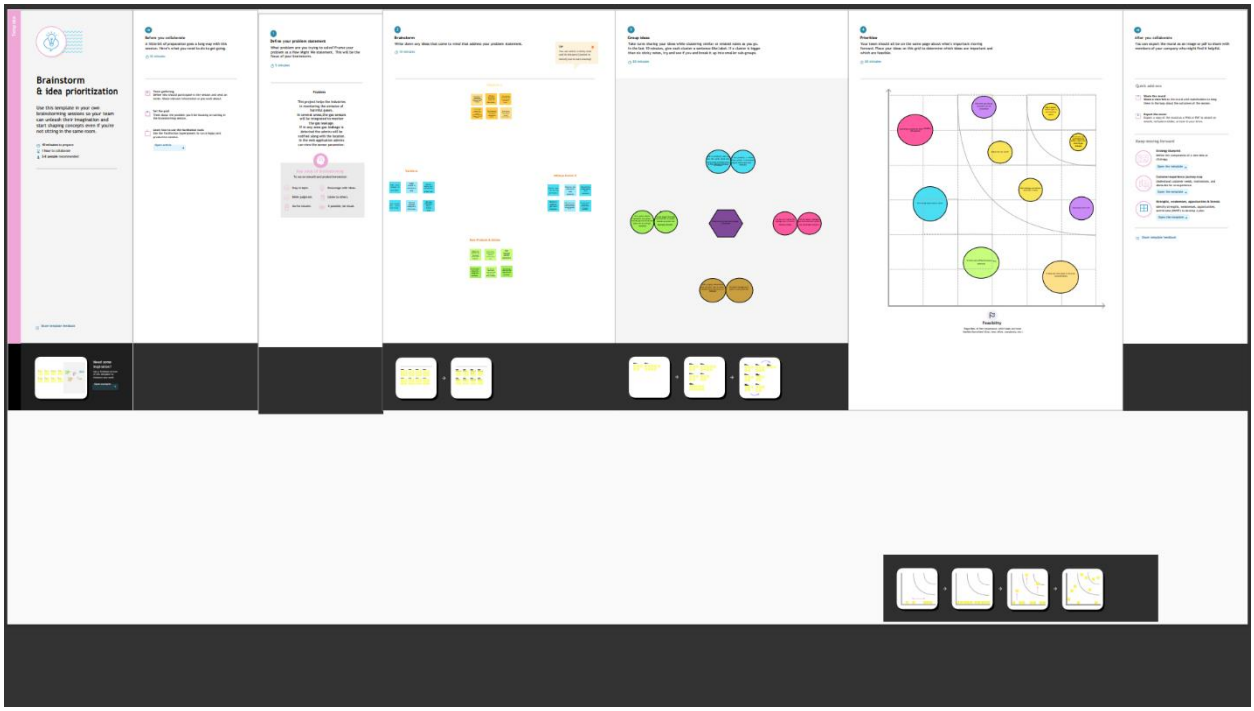
3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

The Internet of Things aims towards making life simpler by automating every small task around us. The increased concentration of certain gases in the atmosphere

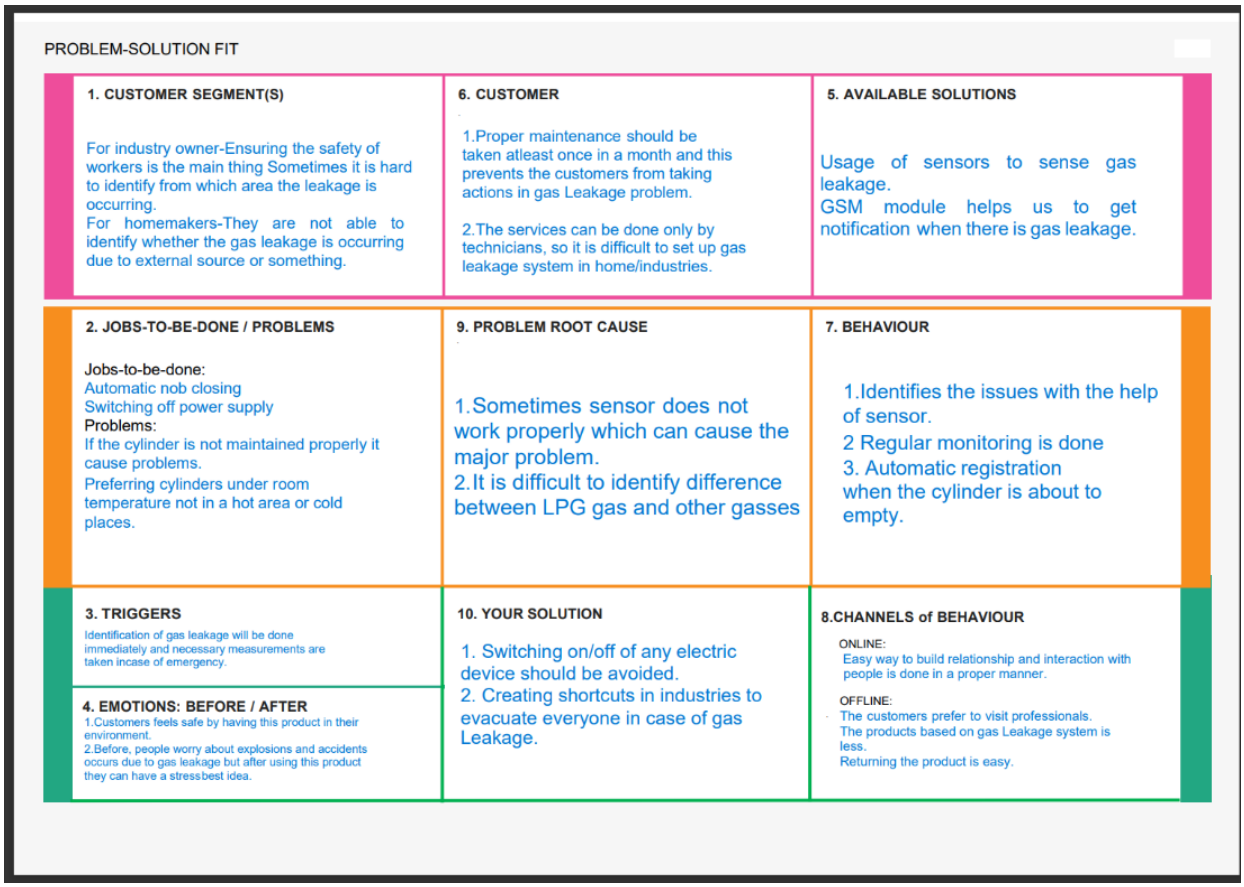
can prove to be extremely dangerous. These gases might be flammable at certain temperature and humidity conditions, toxic after exceeding the specified concentrations limits, or even a contributing factor in the air pollution of an area leading to problems such as smog and reduced visibility which can in turn cause severe accidents and have an adverse effect on the health of people. Most societies have a fire safety mechanism. But itcan use after the fire exists. In order to have control over such conditions we proposed a system that uses sensors that can detect the gases such as LPG and Alerts the people nearby.



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	➢ With a growing demand of industries in various sectors, Nowadays major accidents have been occurring mainly in industries and factories . ➢ The bitter reason is that it is mainly due to various gas leakages. ➢ This gas leakages causes the concentration of toxic gas in atmosphere to be increased and it leads to several respiratory diseases and cause even death.
2.	Idea / Solution description	➢ Understanding the industrialist and the workers needs to avoid any gas leakage accidents can be done using Gas leakage monitoring and alerting system for industries project. ➢ Most of the industries use this gas leakage monitor and detector. This system checks whether there is a gas leakage and monitors the concentration of toxic gas in atmosphere. ➢ When this system detects excess gases in atmosphere it alerts the industries to be get saved from any accidents.
3.	Novelty / Uniqueness	➢ The ultimate benefits of gas leakage monitor and detector includes Portable, battery-powered leak detector to identify gas leaks in analytical instruments. ➢ It also includes A single hand-held system with the bundle offering for two GC flow path monitoring. ➢ Due to use of smart gas leak detect, the industrialists or workers has a reduced issues of accuracies with wireless sensor.

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story/Sub-Task)
FR-1	Objective	The purpose of the system is to detect early gas leakage in the industries through the gas pipelines and alert the user with their location.
FR-2	Focus	To alert the user immediately if any gas leakage is sensed.
FR-3	Features	Gas leakage level will be indicated by the LED lights. It detects the different harmful gases like methane, LPG etc., by using the required sensors. It updates the sensor parameters in web applications.
FR-4	Essentiality	To prevent the industry workers from being exposed to toxic gases.
FR-5	Gas leakage location sent	Location sent to the web application through GPS module.

4.2 Non-Functional Requirement

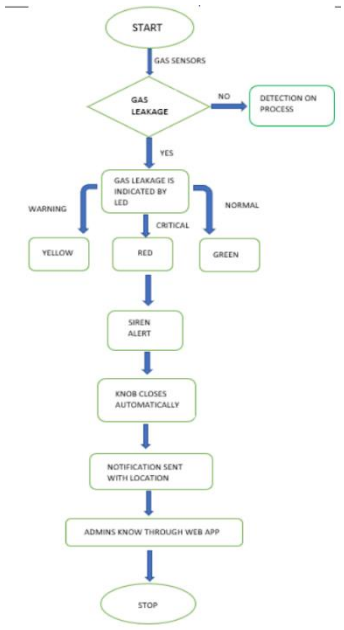
Non-Functional Requirements:

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

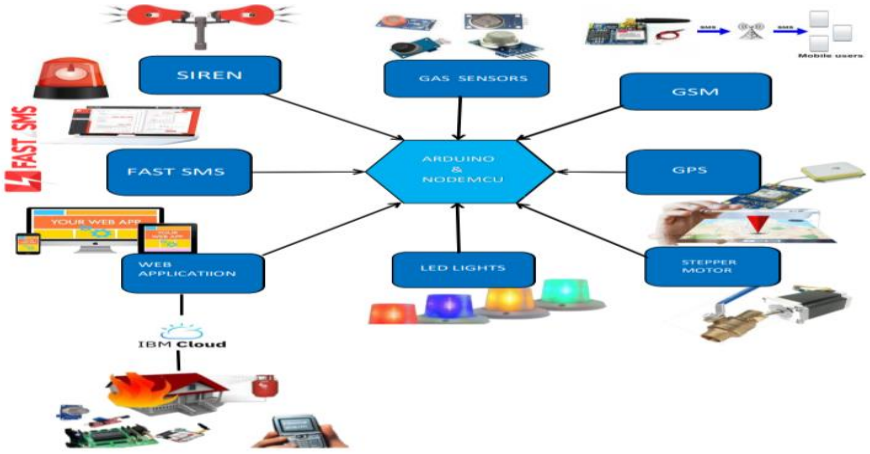
FR No.	Non-Functional Requirement	Description
FR-1	Usability	The web application is simple and easy to use. Efficiency is high.
FR-2	Reliability	The application runs accurately.
FR-3	Availability	The application can be accessed at any time and anywhere.
FR-4	Security	The web application is highly secure. Software is protected from unauthorized access.
FR-5	Scalability	Application is not limited to the users.

5. PROJECT DESIGN

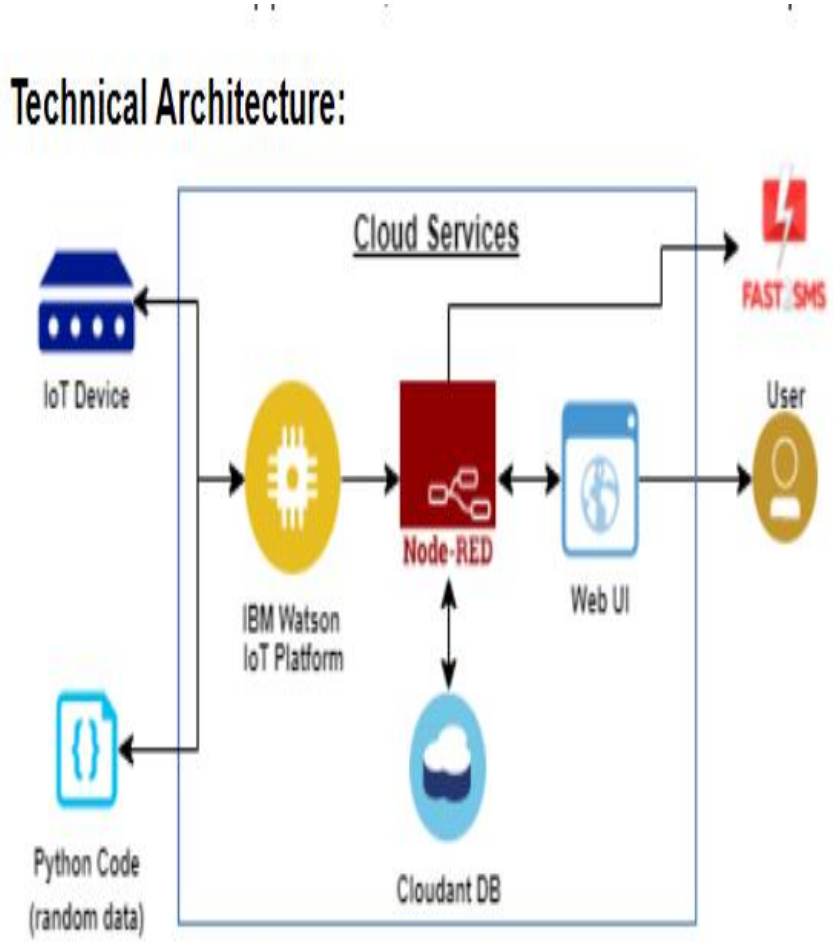
5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



Technical architecture



5.3 User Stories

User Stories

Use the below template to list all the user stories for the product.

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 register for the application by entering my email, 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
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by google.	I can access confirmation email.	High	Sprint-1
		USN-2	As a user, I can register for the application by firebox.	I can access confirmation Login.	low	Sprint-2
	Login	USN-3	As a user, I can register for the application through Gmail		Medium	Sprint-1
Administrator	Registration	USN-1	As a user, I can register for the application through Mobile app.	I can access confirmation My account	High	Sprint-1
		USN-2	As a user, I can register for the application through Mobile app.	I can access confirmation email	low	Sprint-2

6. PROJECT PLANNING & SCHEDULING

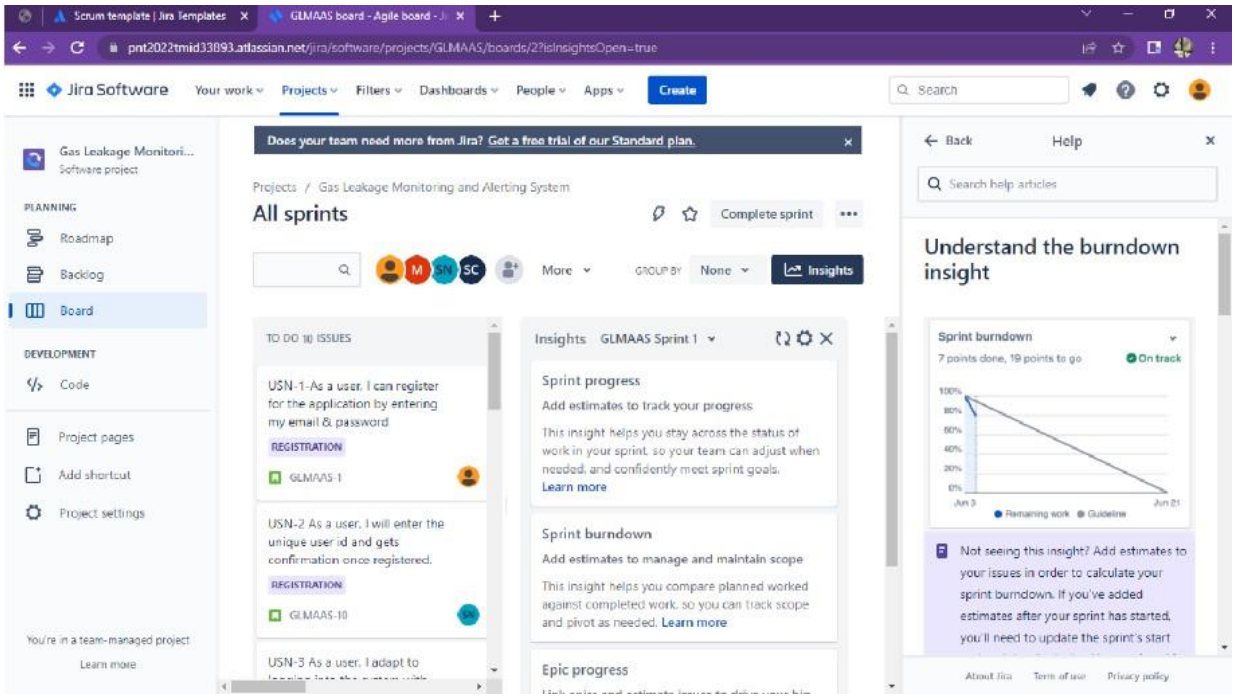
6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Resources Initialization	Create and initialize accounts in various public APIs like Open Weather Map API.	1	LOW	Vikashini , Varsha , Udhayakumar, Akilan , Ram Prakash
Sprint-1	Local Server/Software Run	Write a Python program that outputs results given the inputs like weather and location.	1	MEDIUM	Vikashini , Varsha , Udhayakumar, Akilan , Ram Prakash
Sprint-2	Push the server/software to cloud	Push the code from Sprint 1 to cloud so it can be accessed from anywhere	2	MEDIUM	Vikashini , Varsha , Udhayakumar, Akilan , Ram Prakash
Sprint-3	Hardware initialization	Integrate the hardware to be able to access the cloud functions and provide inputs to the same.	2	HIGH	Vikashini , Varsha , Udhayakumar, Akilan , Ram Prakash
Sprint-4	UI/UX Optimization & Debugging	Optimize all the shortcomings and provide better user experience.	2	LOW	Vikashini , Varsha , Udhayakumar, Akilan , Ram Prakash

6.2 Sprint Delivery Schedule

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

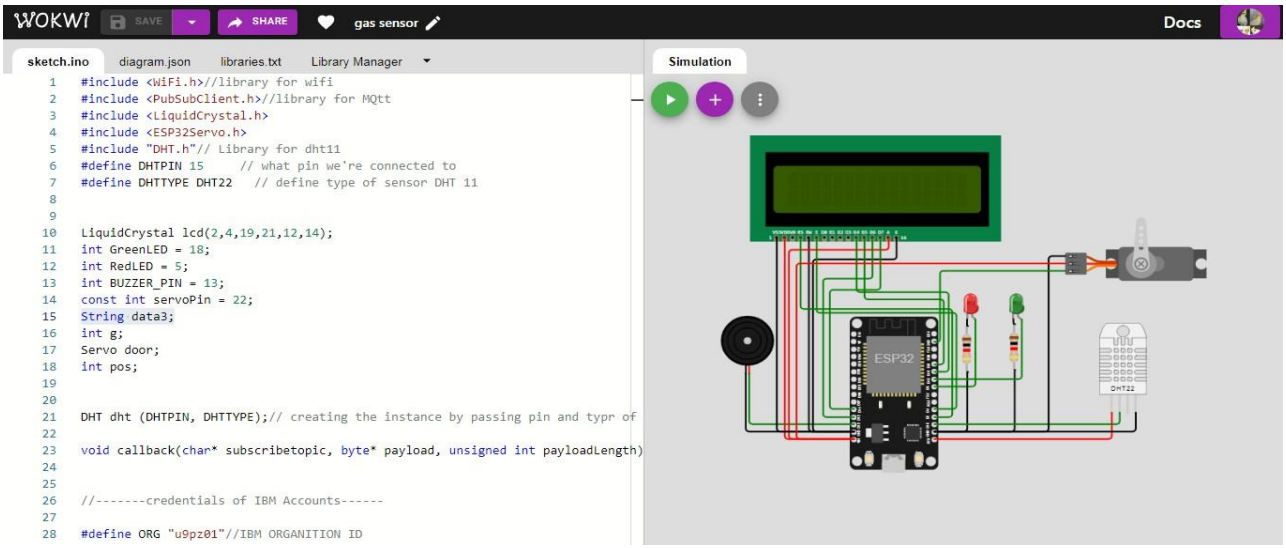
6.3 Reports from JIRA



7. CODING & SOLUTIONING

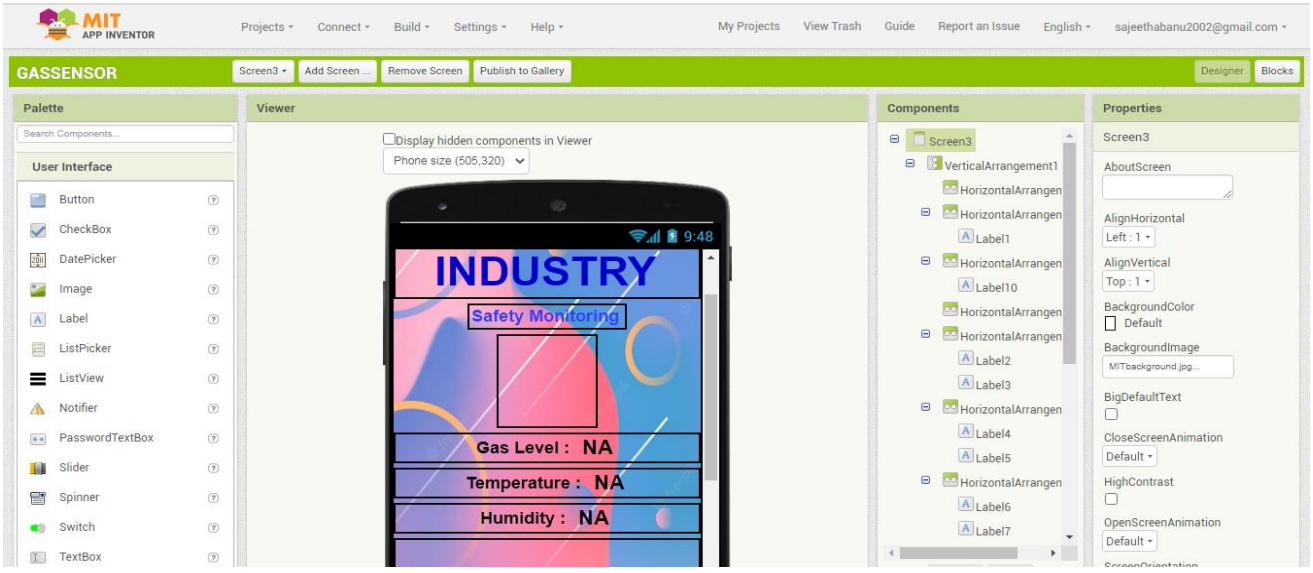
7.1 Feature 1

The Simulation is done using the Wokwi simulator.

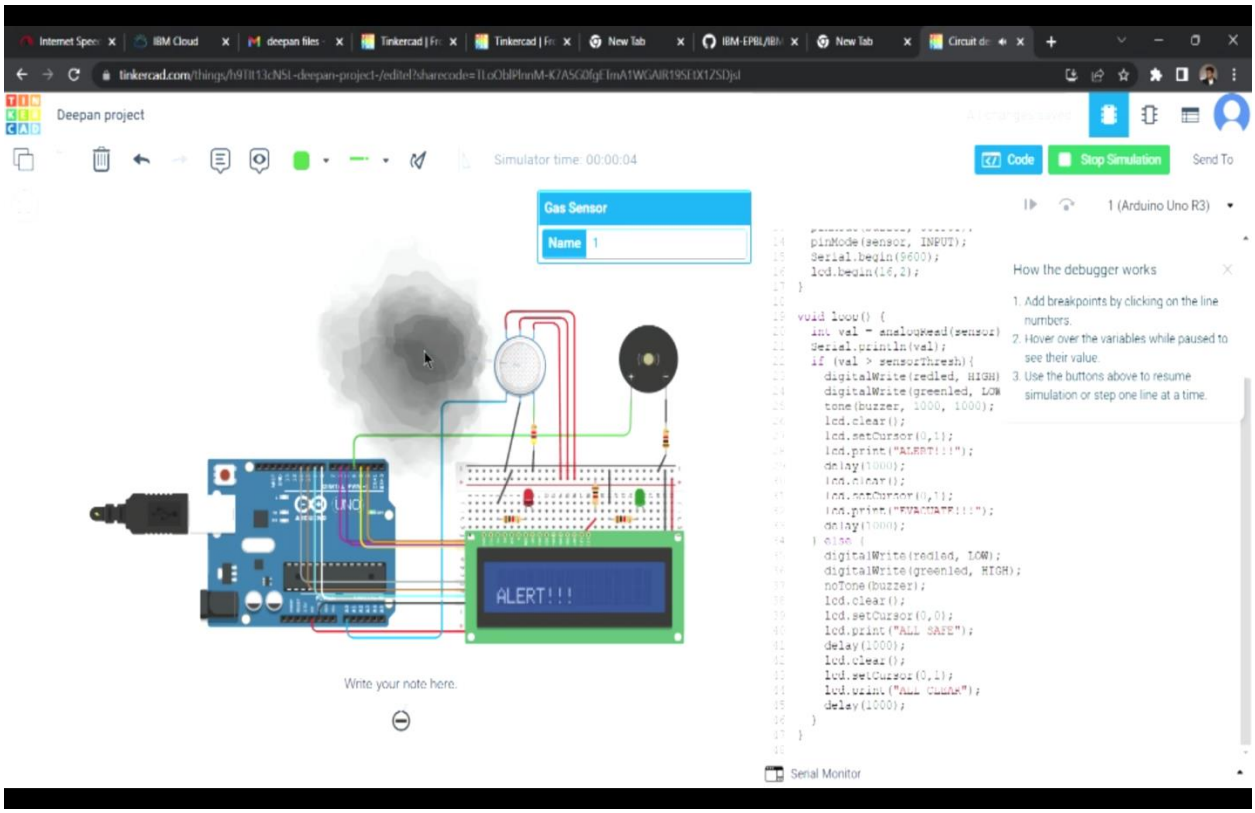


7.2 Feature 2

The Final Controlling is done using the MIT App Inventor.



7.3 Schematic Diagram



Code:

```
#include <WiFi.h>                    //library for wifi
#include <PubSubClient.h>            //library for MQtt
`  #include <LiquidCrystal.h>
#include <ESP32Servo.h>
#include "DHT.h"                     // Library for dht11
#define DHTPIN 15                    // what pin we're connected to
#define DHTTYPE DHT22                // define type of sensor DHT 11
LiquidCrystal lcd(2,4,19,21,12,14);
int GreenLED = 18;
int RedLED = 5;
int BUZZER_PIN = 13;
const int servoPin = 22;
String data3;
int g;
Servo door;
int pos;
DHT dht (DHTPIN, DHTTYPE);//creating the instance by passing pin and type of
dht connected
```

```
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
//-------credentials of IBM Accounts------
#define ORG "u9pz01"//IBM ORGANITION ID
#define DEVICE_TYPE "gassense"//Device type mentioned in IBM Watson IOT Platform
#define DEVICE_ID "sensor"//Device ID mentioned in IBM Watson IOT Platform
#define TOKEN "12345678" //Token
float h, t;
//-------- Customize the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id
//.....
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback, wifiClient); //calling the predefined client id by passing parameter like server id,port and wificredential
void setup() {
Serial.begin(115200);
dht.begin();
pinMode(GreenLED, OUTPUT);
pinMode(RedLED, OUTPUT);
pinMode(BUZZER_PIN, OUTPUT);
lcd.begin(16,2);
lcd.setCursor(1,0);
lcd.print(("GAS DETECTION"));
door.attach(servoPin, 500, 2400);
Serial.println();
wificonnect();
mqttconnect();
}
void loop() {
g =random(0,100);
```

```
Serial.print("Gas Level in Percentage :");
Serial.println(g);
h = dht.readHumidity();
t = dht.readTemperature();
Serial.print("temp:");
Serial.println(t);
Serial.print("Humid:");
Serial.println(h);
condition(g);
PublishData(t, h ,g);
delay(1000);
if (!client.loop()) {
mqttconnect();
}
delay(5000);
}
// Condition for buzzer
void myTone( int pin)
{
ledcAttachPin(pin, 0); // pin, channel
ledcWriteNote(0, NOTE_F, 4); // channel, frequency, octave
}
void myNoTone( int pin)
{
ledcDetachPin(pin);
}
// Condition for Gas level
void condition(int g)
{
if(g > 50)
{
myTone(BUZZER_PIN);
digitalWrite(RedLED, HIGH);
digitalWrite(GreenLED, LOW);
delay(500);
lcd.setCursor(0,1);
lcd.print("ALERT!!");
delay(300);
```

```
lcd.setCursor(0,1);
lcd.print("HAZARDOUS LEVEL!");
}
else
{
myNoTone(BUZZER_PIN);
digitalWrite(RedLED, LOW);
digitalWrite(GreenLED, HIGH);
delay(500);
lcd.setCursor(0,1);
lcd.print("NORMAL GAS LEVEL");
}
}
/*.....retrieving to Cloud.....*/
void PublishData(float temp, float Humid, int Gas) {
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 += "," "\"Humid\":";
payload += Humid;
payload += "," "\"Gas\":";
payload += Gas;
payload += "}";
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str())) {
Serial.println("Publish ok");// if it successfully 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 definition for wificonnect
{
Serial.println();
Serial.print("Connecting to ");
WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the
connection
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}
void initManagedDevice() {
if (client.subscribe(subscribetopic)) {
Serial.println((subscribetopic));
Serial.println("subscribe to cmd OK");
} else {
Serial.println("subscribe to cmd FAILED");
}
}
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{
Serial.print("callback invoked for topic: ");
Serial.println(subscribetopic);
```

```
for (int i = 0; i < payloadLength; i++) {
//Serial.print((char)payload[i]);
data3 += (char)payload[i];
}
Serial.println("data: "+ data3);
if(data3=="dooropen")
{
Serial.println(data3);
pos = 180; //open the door
door.write(pos);
}
else
{
Serial.println(data3);
pos = 0; // closing the door
door.write(pos);
}
data3="";
}
```

8. TESTING

8.1 Test Cases

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
App installation_TC_001	Functional	MIT App inventor	Verify user is able to install the app for any device	Knowledge about App creation	1. Search the application. 2.Enter the URL. 3.Click and Go. 4.Install the application.	http://ai2.appinventor.mit.edu/#6560580513955840	Installation app page will be displayed	Expected result appeared	Pass	-	Yes	-	GOOGLE
App installation_TC_002	UI	MIT App inventor	Verify user is able to install the app for any device	Knowledge about App development	1. Search the application. 2.Enter the URL. 3.Click and Go. 4.Install the application. 5.Downloading the android app file	http://ai2.appinventor.mit.edu/#6560580513955840	Downloading the required app.	Expected result appeared	Pass	-	Yes	-	MIT APP INVENTOR
Import to device_TC_003	Functional	Device/Phone	Verify user is able to install the app for any device	Knowledge about App import	1. Open the downloaded apk file. 2. Select the location where it has to be placed. 3. Import the downloaded file.	http://ai2.appinventor.mit.edu/#6560580513955840	Import of apk file to device.	Expected result appeared	Pass	-	Yes	-	MOBILE PHONE
App running_TC_004	Functional	Device/Phone	Verify user is able to run the app	Access of App	1. Search the app from where it is located. 2.Click and open the app. 3.Home page is displayed.	Opening of app	Whether the app is opened without any delay	Expected result appeared	Pass	-	Yes	-	MOBILE PHONE

Login page_TC_005	Functional	Home page	Verify user is able to see the Login/Signup popup when user clicked on My account button	Access of App	1. Open the app 2. Verify Login and signup Popup displayed or not.	Login of App	Login/Signup popup should display	Expected result appeared	Pass	-	Yes	-	APPLICATION
Login page_TC_006	UI	Home page	Verify the UI elements in Login/Signup popup	Access of App with Login Credentials	1. Open the app 2. Verify Login and signup Popup with below UI elements. a. User name textbox b. Password textbox c. Login textbox d. Sign up textbox	Login of App	Application should show below UI elements: a.email text box b.password text box c.Login button d.New customer? Create account link	Expected result appeared	Pass	-	Yes	-	APPLICATION
Login page_TC_007	Functional	Home page	Verify user is able to log into application with Valid credentials	Access of App with Login Credentials	1.Open the App. 2.Enter Valid username in User Name text box 3.Enter valid password in password text box 4.Click on login button	Username: Sarss password: Test123	User should navigate to user account homepage	Expected result appeared	Pass	-	Yes	-	APPLICATION
Login page_TC_008	Functional	Login page	Verify user is able to log into application with Valid credentials	Access of App with Login Credentials	1.Open the App. 2.Enter InValid username in User Name text box 3.Enter valid password in password text box 4.Click on login button	Username: Cutie password: Test123	Application should show 'Incorrect email or password ' validation message.	Expected result appeared	Pass	-	Yes	-	APPLICATION
LoginPage_TC_009	Functional	Login page	Verify user is able to log into application with Valid credentials	Access of App with Login Credentials	1.Open the App. 2.Enter Valid username in User Name text box 3.Enter Invalid password in password text box 4.Click on login button	Username: Sarss password: Test1234	Application should show 'Incorrect email or password ' validation message.	Expected result appeared	Pass	-	Yes	-	APPLICATION
LoginPage_TC_010	Functional	Login page	Verify user is able to log into application with Valid credentials	Access of App with Login Credentials	1.Open the App. 2.Enter InValid username in User Name text box 3.Enter Invalid password in password text box 4.Click on login button	Username: Cutie password: Test1234	Application should show 'Incorrect email or password ' validation message.	Expected result appeared	Pass	-	Yes	-	APPLICATION
Sensing gas_TC_011	Functional	Wokwi Simulator	Verify user for sensing the required parameters.	Knowledge about sensor components	1.Add the elements for sensing. 2.Place it in the right position. 3.Make sensing is done for required parameters. 4.Make the wired connection.	Gas, Temperature, Humidity values	Correct sensing is done with desired result	Expected result appeared	Pass	-	Yes	-	WOKWI SIMULATOR
Sensing gas_TC_012	Functional	Wokwi Simulator	Verify user for sensing the required parameters.	Knowledge about sensor components	1.Add the elements for sensing. 2.Place it in the right position. 3.Make sensing is done for required parameters. 4.Make the wired connection.	Various Gas values	Measuring various gases	Certain gas are not sensed	Fail	Specified gas sensors measures certain gases	Yes	BUG ID_1	WOKWI SIMULATOR
Alertion Indication_TC_013	Functional	Wokwi Simulator	Verify user gets alertion message.	Knowledge about alert components	1.Add the elements for alertion. 2.Connect LED, Buzzer, LCD for alertion. 3.Alerting message occurs when reaches the desired condition.	LED ON/OFF, Buzzer sound, Display of Alert message	LED glowing, Buzzer tone and Display message is to be obtained	Expected result appeared	Pass	-	Yes	-	WOKWI SIMULATOR

8.2 User Acceptance Testing

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the **gas leakage monitoring and alerting system** project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

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

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
By Design	5	4	2	0	11
Duplicate	3	2	0	0	5
External	2	3	0	1	6
Fixed	11	2	4	2	19
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	2	1	0	0	3
Totals	23	12	6	3	44

3. Test Case Analysis

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

Section	TotalCases	Not Tested	Fail	Pass
Print Engine	7	0	1	6
Client Application	51	0	0	51
Security	2	0	0	2
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

As the Result the project helps in the stress free and fear free lives of many people.

It saves the lives of many people.It alerts the people by using the LCD Display and Buzzer in advance. This principle of Operation is done by Wokwi and MIT App Inventor.

9.1 Performance Metrices

		NFT - Risk Assessment						
Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Volume Change	Risk Score	Justification
Gas Leakage Monitering and Alerting system for industries	New	Moderate	No Changes	Moderate	17-4	>70 to 100%	ORANGE	As we have seen changes

NFT - Detailed Test Plan				
S.No	Project Overview	NFT Test approach	Assumptions/Dependencies/Risks	Approvals/SignOff
1	Indication of gas alert	Done	In case of absense it may lead to improper detection /It Depent on the application owner/It may lead to risk live sof many people	Approved

End Of Test Report							
Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Recommendations	(Detected/Closed/Open)	Approvals/SignOff
Indication of Gas Alert	Done	Done	Detects the gas,Temperature and Humudity level and alerts the User	GO	It is Recommended to send the alert message directly to the users mobile.	Closed	Approved

10. ADVANTAGES

- The sensor-enabled solutions helps prevent the high risk of gas explosion and affecting any casualties within and outside the premises.
- Prevent fire hazards and explosions.
- Ensure worker's health.
- Real-time updates about leakages.
- Get immediate gas leak alerts.

DISADVANTAGES

- Poor stability.
- Only one gas can be measured with each instrument.
- Greater environmental impact.
- It can be poisoned by lead, chlorine and silicon.
- It requires more user expertise.

11. CONCLUSION

Gas leakage occurs mainly due to poor maintenance of equipment's and inadequate awareness of the people. Hence, Gas leakage detection is essential to prevent accidents and to save human lives. This paper presented is based on gas leakage monitoring and alerting system. This system triggers LED and buzzer to alert people when gas leakage is detected. This system is very simple yet reliable.

12. FUTURE SCOPE

Future scope could be including a Automatic Shut-off device which will turn off the gas supply whenever it will detect any gas leakage. This system can be implemented in Industries, Hotels and wherever the LPG cylinders are used..

13. APPENDIX

Source Code(Simulation Code):

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include <LiquidCrystal.h>
#include <ESP32Servo.h>
#include "DHT.h"// Library for dht11
#define DHTPIN 15 // what pin we're connected to
#define DHTTYPE DHT22 // define type of sensor DHT 11
LiquidCrystal lcd(2,4,19,21,12,14);
int GreenLED = 18;
int RedLED = 5;
int BUZZER_PIN = 13;
const int servoPin = 22;
String data3;
int g;
Servo door;
int pos;
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 "u9pz01"//IBM ORGANITION ID
#define DEVICE_TYPE "gassense"//Device type mentioned in IBM Watson IOT Platform
```

```
#define DEVICE_ID "sensor"//Device ID mentioned in IBM Watson IOT Platform
#define TOKEN "12345678" //Token
float h, t;
//-------- Customize the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and
format in which data
to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd REPRESENT command
type AND COMMAND IS TEST OF
FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id
//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback, wifiClient); //calling the predefined client id by
passing parameter
like server id,port and wificredential
void setup() {
Serial.begin(115200);
dht.begin();

pinMode(GreenLED, OUTPUT);
pinMode(RedLED, OUTPUT);
pinMode(BUZZER_PIN, OUTPUT);
lcd.begin(16,2);
lcd.setCursor(1,0);
lcd.print(("GAS DETECTION"));
door.attach(servoPin, 500, 2400);
Serial.println();
wificonnect();
mqttconnect();
}
void loop() {
g =random(0,100);
Serial.print("Gas Level in Percentage :");
Serial.println(g);
h = dht.readHumidity();
```

```
t = dht.readTemperature();
Serial.print("temp:");
Serial.println(t);
Serial.print("Humid:");
Serial.println(h);
condition(g);

PublishData(t, h ,g);
delay(1000);
if (!client.loop()) {
mqttconnect();
}
delay(5000);
}
// Condition for buzzer
void myTone( int pin)
{
ledcAttachPin(pin, 0); // pin, channel
ledcWriteNote(0, NOTE_F, 4); // channel, frequency, octave
}
void myNoTone( int pin)
{
ledcDetachPin(pin);
}
// Condition for Gas level
void condition(int g)
{
if(g > 50)

{
myTone(BUZZER_PIN);
digitalWrite(RedLED, HIGH);
digitalWrite(GreenLED, LOW);
delay(500);
lcd.setCursor(0,1);
lcd.print("ALERT!!");
delay(300);
lcd.setCursor(0,1);
lcd.print("HAZARDOUS LEVEL!");
}
```

```
else
{
myNoTone(BUZZER_PIN);
digitalWrite(RedLED, LOW);
digitalWrite(GreenLED, HIGH);
delay(500);
lcd.setCursor(0,1);
lcd.print("NORMAL GAS LEVEL");
}
}
/*.....retrieving to Cloud..... */

void PublishData(float temp, float Humid, int Gas) {
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 += "," "\"Humid\":";
payload += Humid;
payload += "," "\"Gas\":";
payload += Gas;
payload += "}";
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str())) {
Serial.println("Publish ok");// if it successfully 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 definition for wificonnect
{
Serial.println();
Serial.print("Connecting to ");
WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection
}

while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}
void initManagedDevice() {
if (client.subscribe(subscribetopic)) {
Serial.println((subscribetopic));
Serial.println("subscribe to cmd OK");
} else {
Serial.println("subscribe to cmd FAILED");
}
}
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)

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

```
Serial.println("data: "+ data3);
if(data3=="dooropen")
{
Serial.println(data3);
pos = 180; //open the door
door.write(pos);
}
else
{
Serial.println(data3);
pos = 0; // closing the door
door.write(pos);
}
data3="";
}
```

GITHUB LINK :

GitHub Repositories :

https://github.com/IBM-EPBL/IBM-Project-45678-1660731582