Gas leakage monitoring and alerting system for Industries

TEAM ID: PNT2022TMID04339

TEAM LEAD	DHARANESH S	737819CSR034
TEAM MEMBER 1	ARUN PRABU A S	737819CSR012
TEAM MEMBER 2	DHARAANESHWARAN S	737819CSR032
TEAM MEMBER 3	HARIHARAN G	737819CSR052
TEAM MEMBER 4	HARIHARAN T	737319CSR053

INDEX

1.INTRODUCTION

1.1 Project Overview

IoT is an expanding network of physical devices that are linked with different types of sensors and with the help of connectivity to the internet, they are able to exchange data. Through IoT, internet has now extended its roots to almost every possible thing present around us and is no more limited to our personal computers and mobile phones. Safety, the elementary concern of any project, has not been left untouched by IoT. Gas Leakages in open or closed areas can prove to be dangerous and lethal. The traditional Gas Leakage Detector Systems though have great precision, fail to acknowledge a few factors in the field of alerting the people about the leakage. Therefore, we have used the IoT technology to make a Gas Leakage Detector having Smart Alerting techniques involving calling, sending text message and an e-mail to the concerned authority and an ability to predict hazardous situation so that people could be made aware in advance by performing data analytics on sensor readings.

1.2 Purpose

Fire accidents have been taking place frequently and the threat to human lives and properties is growing in recent years. Some gases are highly inflammable and can burn even at some distance from the source of leakage. Most fire accidents are caused because of a poor-quality rubber tube or the regulator is not turned off when not in use. Therefore, developing the gas leakage alert system is very essential.

2. LITERATURE SURVEY

2.1 Existing problem

In the existing method, gas sensing technology is used. The LPG leakage is detected by the semiconductor sensor. The leakage of gas may happen due to the human error, false chemical reaction, lack of service done in the gas valve. In the existing method, periodic check done by manually and partial sensing methodology is used. When the leakage was happened, it leads to major fire accident Before controlling the fire major accident may happen which leads to heavy loss in industry as well as human life. In addition to that the leak of gas may spread in the atmosphere, it may affect all the living things in an around them. In the existing system MQ5 sensor is used to detect gas leakage. Exhaust fans are used to suck out the gases when the leakage occurs. In the existing method, it raises only alarm whenever Gas leaked or fire is detected at any place in a factory. Due to this alarm, people could start to run haphazardly. Fire Service truck vehicle only control the fire accident.

2.2 References

[1] Rajeev B. Ahuja, Jayant K. Dash, Prabhat Shrivastava, “A comparative analysis of liquefied petroleum gas (LPG) and kerosene related burns”, Burns, Volume 37, Issue 8, December 2011.

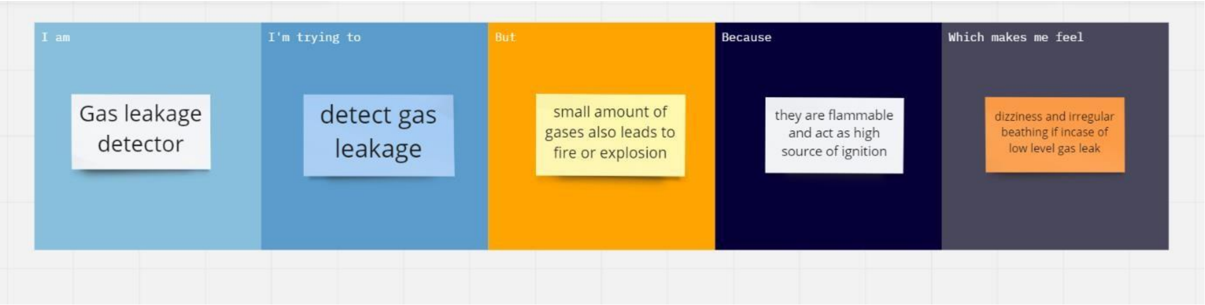
[2] Prof. Pankaj C. Warule, Shivam Upadhyay, Snehal S. Shelke, Sumitra K. Khandade, “LPG Detection, Metering and Control System Using Microcontroller”, IJARIIE, Volume 2, Issue 2, 2016.

[3] Ankit Sood, Babalu Sonkar, Atul Ranjan, Mr. Ameer Faisal, “Microcontroller Based LPG Gas Leakage Detector Using GSM Module”, International Journal of Electrical and Electronics Research, Volume 3, Issue2, April- June 2015.

[4] Ashish Shrivastava, Ratnesh Prabhakar, Rajeev Kumar, Rahul Verma, “GSM Based Gas Leakage Detection System”, International Journal of Technical Research and Applications”, Volume 1, Issue2, May- June 2013.

2.3 Problem Statement Definition

The risk of firing, explosion, suffocation all are based on their physical properties such flammability, toxicity etc. The number of deaths due to the explosion of gas cylinders has been increasing in recent years. The reason for such explosion is due to substandard cylinders, old valves, worn out regulators and lack of awareness using gas cylinders add to the risks. Inspections by oil companies found that many LPG consumers are unaware of safety checks of gas cylinders. Another reason is illegal filling of gas cylinder also causes accidents. There is a need for a system to detect and also prevent leakage of LPG.

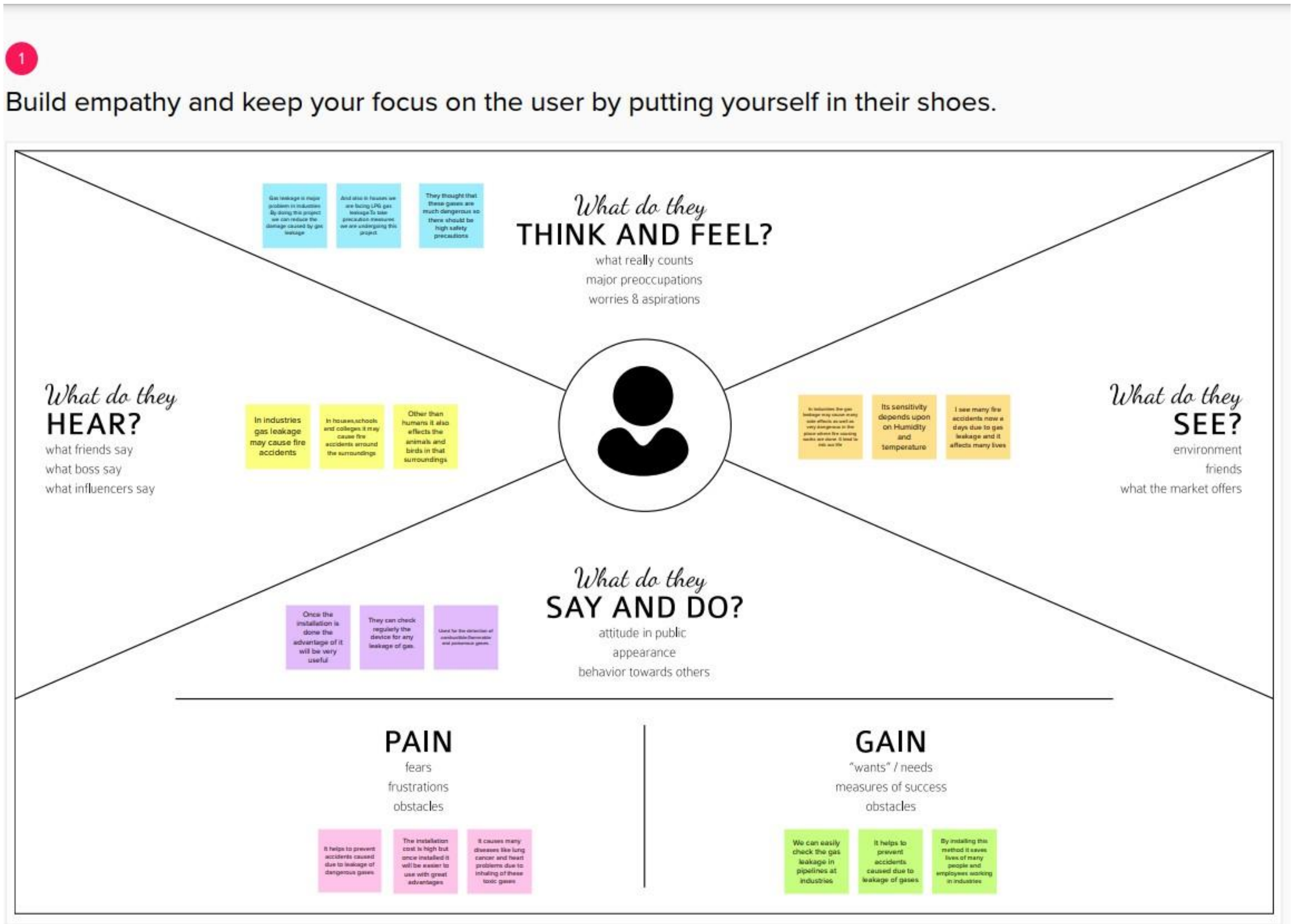


I am	I'm trying to	But	Because	Which makes me feel
Gas leakage detector	detect gas leakage	small amount of gases also leads to fire or explosion	they are flammable and act as high source of ignition	dizziness and irregular beathing if incase of low level gas leak

Problem Statement (PS)	**I am (Customer)**	**I'm trying to**	**But**	**Because**	**Which makes me feel**
PS-1	worker	Periodic checking of gas leakage	Small amount of gases also leads to fire or explosion	They are flammable and act as high source of ignition	Dizziness and irregular breathing if incase of low level gas leak
PS-2	Industry owner	Protect my industry from accidents	Don't have proper monitoring technology	Because of high initial setup cost	Fear

3. IDEATION & PROPOSED SOLUTION 3.1 Empathy

Map Canvas



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

Gas leakage is major problem in industries By doing this project we can reduce the damage caused by gas leakage

And also in houses we are facing LPG gas leakage.To take precaution measures we are undergoing this project

They thought that these gases are much dangerous so there should be high safety precautions

In industries gas leakage may cause fire accidents

In houses,schools and colleges it may cause fire accidents arround the surroundings

Other than humans it also effects the animals and birds in that surroundings

In industries the gas leakage may cause many side effects as well as very dangerous in the place where fire causing works are done. It lead to risk our life

Its sensitivity depends upon on Humidity and temperature

I see many fire accidents now a days due to gas leakage and it affects many lives

Once the installation is done the advantage of it will be very useful

They can check regularly the device for any leakage of gas.

Used for the detection of combustible/flammable and poisonous gases.

It helps to prevent accidents caused due to leakage of dangerous gases

The installation cost is high but once installed it will be easier to use with great advantages

It causes many diseases like lung cancer and heart problems due to inhaling of these toxic gases

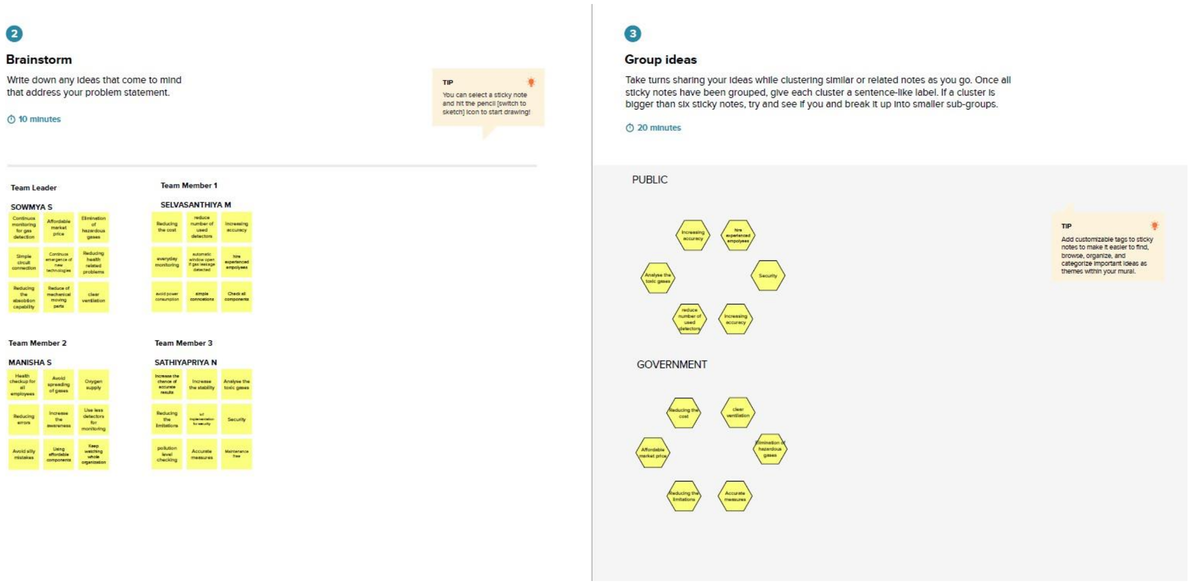
We can easily check the gas leakage in pipelines at industries

It helps to prevent accidents caused due to leakage of gases

By installing this method it saves lives of many people and employees working in industries

3.2

Ideation & Brainstorming



2

Brainstorm

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

10 minutes

TIP

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

Team Leader

SOWMYA S

Continuos monitoring for gas detection	Affordable market price	Elimination of hazardous gases
Simple circuit connection	Continuos emergence of new technologies	Reducing health related problems
Reducing the absobtion capability	Reduce of mechanical moving parts	clear ventilation

Team Member 1

SELVASANTHIYA M

Reducing the cost	reduce number of used detectors	Increasing accuracy
everyday monitoring	automatic window open if gas leakage detected	hire experienced empolyees
avoid power consumption	simple connections	Check all components

Team Member 2

MANISHA S

Health checkup for all employees	Avoid spreading of gases	Oxygen supply
Reducing errors	Increase the awareness	Use less detectors for monitoring
Avoid silly mistakes	Using affordable components	Keep watching whole organization

Team Member 3

SATHIYAPRIYA N

Increase the chance of accurate results	Increase the stability	Analyse the toxic gases
Reducing the limitations	iot implementation for security	Security
pollution level checking	Accurate measures	Maintenance free

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

PUBLIC

- Increasing accuracy
- hire experienced empolyees
- Analyse the toxic gases
- Security
- reduce number of used detectors
- Increasing accuracy

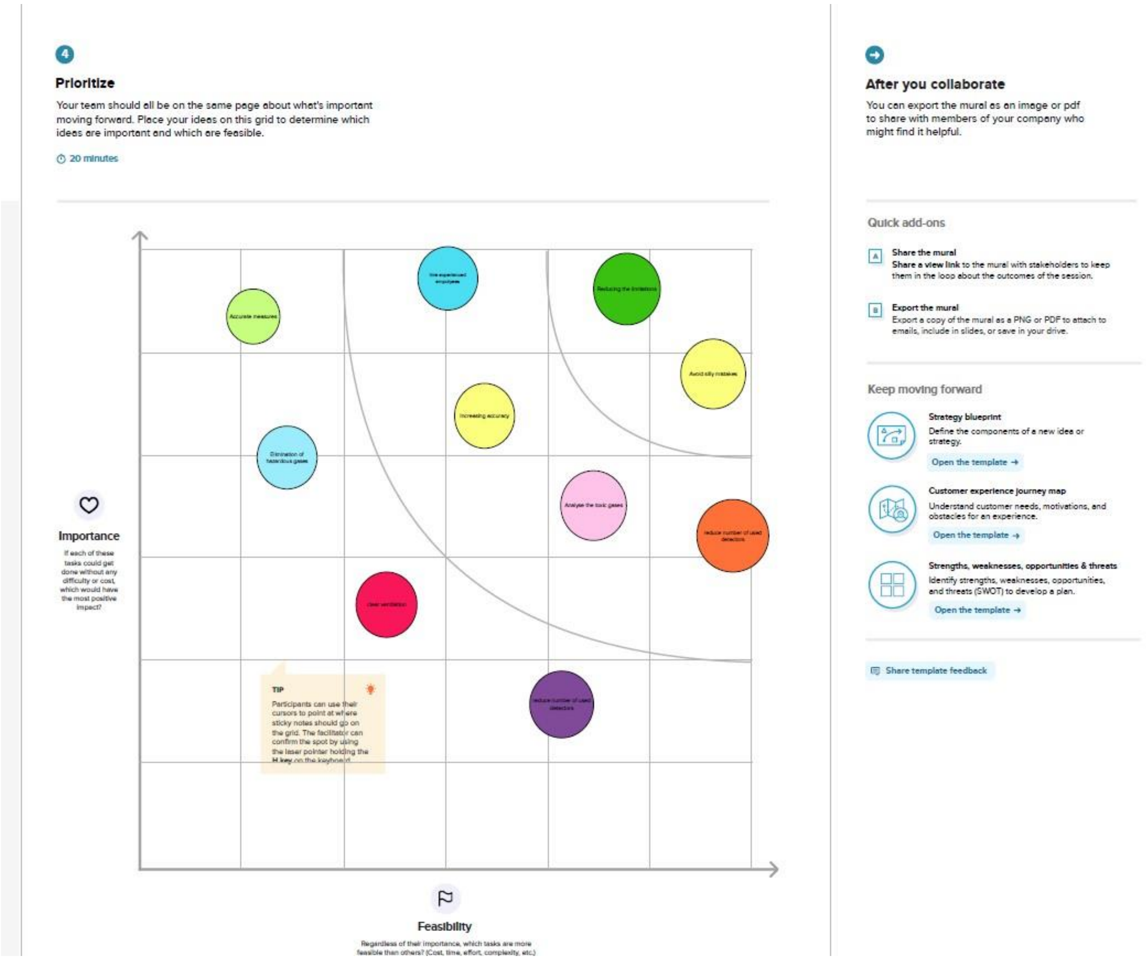
TIP

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

GOVERNMENT

- Reducing the cost
- clear ventilation
- Affordable market price
- Elimination of hazardous gases
- Reducing the limitations
- Accurate measures

3.3



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

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard.

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.)

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A Share the mural
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B Export the mural
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

Proposed Solution

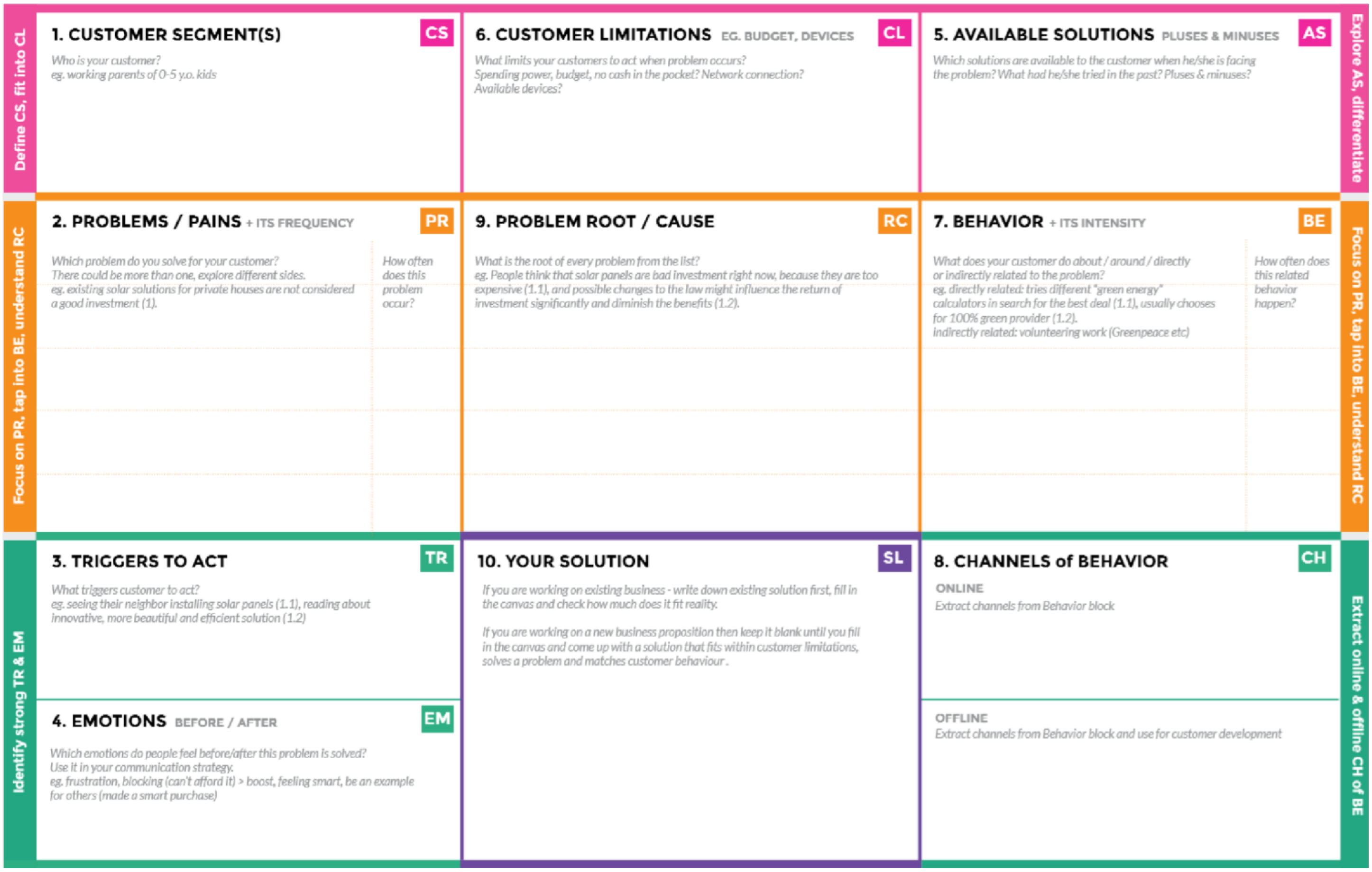
S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The leakage of gases only can be detected by human nearby and if there are no human nearby, it cannot be detected. But sometimes it cannot be detected by human that has a low sense of smell. Thus, this system will help to detect the presence of gas leakage.
2.	Idea / Solution description	If the system detects the level of gas in the air that exceeds the safety level it will activate the alarm which includes the buzzer to alert the users at industries of the

3.4

		abnormal condition and to take any necessary action.
3.	Novelty / Uniqueness	Reducing the cost of the gas leakage detector and increasing the accuracy percentage.
4.	Social Impact / Customer Satisfaction	These leaks *cause safety threats and secondary accidents* for those working in industry and the environment
5.	Business Model (Revenue Model)	The gas detector market is forecast to reach $2.96 billion by 2025, growing at a CAGR of 4% during 2019-2025.
6.	Scalability of the Solution	A wide range of *industrial* fixed *gas detectors* featuring flexible integration, simple installation, user-friendly operation

Problem Solution fit

3.5



Define CS, fit into CL

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
eg. working parents of 0-5 y.o. kids

6. CUSTOMER LIMITATIONS EG. BUDGET, DEVICES CL

What limits your customers to act when problem occurs?
Spending power, budget, no cash in the pocket? Network connection?
Available devices?

5. AVAILABLE SOLUTIONS PLUSES & MINUSES AS

Which solutions are available to the customer when he/she is facing the problem? What had he/she tried in the past? Pluses & minuses?

Explore AS, differentiate

Focus on PR, tap into BE, understand RC

2. PROBLEMS / PAINS + ITS FREQUENCY PR

Which problem do you solve for your customer?
There could be more than one, explore different sides.
eg. existing solar solutions for private houses are not considered a good investment (1).

How often does this problem occur?

9. PROBLEM ROOT / CAUSE RC

What is the root of every problem from the list?
eg. People think that solar panels are bad investment right now, because they are too expensive (1.1), and possible changes to the law might influence the return of investment significantly and diminish the benefits (1.2).

7. BEHAVIOR + ITS INTENSITY BE

What does your customer do about / around / directly or indirectly related to the problem?
eg. directly related: tries different "green energy" calculators in search for the best deal (1.1), usually chooses for 100% green provider (1.2).
indirectly related: volunteering work (Greenpeace etc)

How often does this related behavior happen?

Focus on PR, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TO ACT TR

What triggers customer to act?
eg. seeing their neighbor installing solar panels (1.1), reading about innovative, more beautiful and efficient solution (1.2)

4. EMOTIONS BEFORE / AFTER EM

Which emotions do people feel before/after this problem is solved?
Use it in your communication strategy.
eg. frustration, blocking (can't afford it) > boost, feeling smart, be an example for others (made a smart purchase)

10. YOUR SOLUTION SL

If you are working on existing business - write down existing solution first, fill in the canvas and check how much does it fit reality.

If you are working on a new business proposition then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour .

8. CHANNELS of BEHAVIOR CH

ONLINE
Extract channels from Behavior block

OFFLINE
Extract channels from Behavior block and use for customer development

Extract online & offline CH of BE

4. REQUIREMENT ANALYSIS

3.6

4.1 Functional requirement

FR No.	**Functional Requirement (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	User alert	Warnings must be sent to the user. Send the message as soon as possible
FR-2	User Understanding	The user could understand the amount and type of gas leaked and detect the location
FR-3	User controls	The user shall be able to turn off the electricity and other gadgets.
FR-4	User feasible	The user shall be able to notify the nearby fire station if gas leakage level is high.
FR-5	User location	The user shall be able to view the location of the gas leaked.

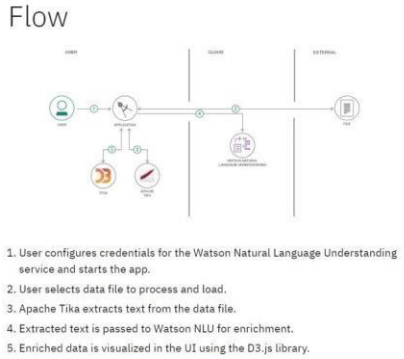
4.2 Non-Functional requirements

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Easy to operate and can be used effectively even by the uneducated people.
NFR-2	**Security**	The communication between the sensors and the simulator are secured using encryption.
NFR-3	**Reliability**	0% false alarming rate and able to get notifications through SMS, e-mail, or even through call.
NFR-4	**Performance**	Low latency and immediate response to the user and make immediate decision.
NFR-5	**Availability**	The system should work 24/7.
NFR-6	**Scalability**	The system can be used for domestic houses or even for large industry.

5. PROJECT DESIGN

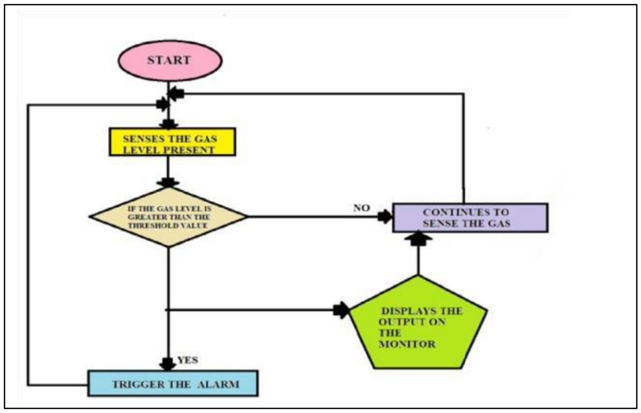
5.1 Data Flow Diagrams

Examp(Simplif)



Flow

1. User configures credentials for the Watson Natural Language Understanding service and starts the app.
2. User selects data file to process and load.
3. Apache Tika extracts text from the data file.
4. Extracted text is passed to Watson NLU for enrichment.
5. Enriched data is visualized in the UI using the D3.js library.


START
SENSES THE GAS LEVEL PRESENT
IF THE GAS LEVEL IS GREATER THAN THE THRESHOLD VALUE
NO
CONTINUES TO SENSE THE GAS
DISPLAYS THE OUTPUT ON THE MONITOR
YES
TRIGGER THE ALARM

5.2 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	User can install the mobile application	I can access my account / dashboard	High	Sprint-1
		USN-2	User can register their details like email and mobile number	I can receive confirmation email & click confirm	High	Sprint-1
	Login	USN-3	User can log on to the application using email and password.	I can Login into my application	High	Sprint-1
	Dashboard	USN-4	User can update their details like alternative mobile number etc..	I can view and change my details.	High	Sprint-2
		USN-5	User can view the gas level and the working condition of the sensors.	I can view the data given by the device	High	Sprint-2
Customer (Web user)	Usage	USN-1	User can register through the web	I can receive confirmation	High	Sprint-3
			page with e-mail or phone number.	email & click confirm		
		USN-2	User can log on to the web page using email and password.	I can Login into my application	High	Sprint-3
Customer	Working	USN-1	User can view the details	Act according to the alarm	Medium	Sprint-3
		USN-2	User can view the alert and turn off the power supply.	Act according to the alarm	High	Sprint-4
Customer Care Executive	Action	USN-1	Executer solves the user’s problem	I can solve the issues	High	Sprint-4
Administrator	Administration	USN-1	Periodic check the condition of sensors and Stores the user’s information	I can maintain the fault tolerance and error rate	High	Sprint-4

6.PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

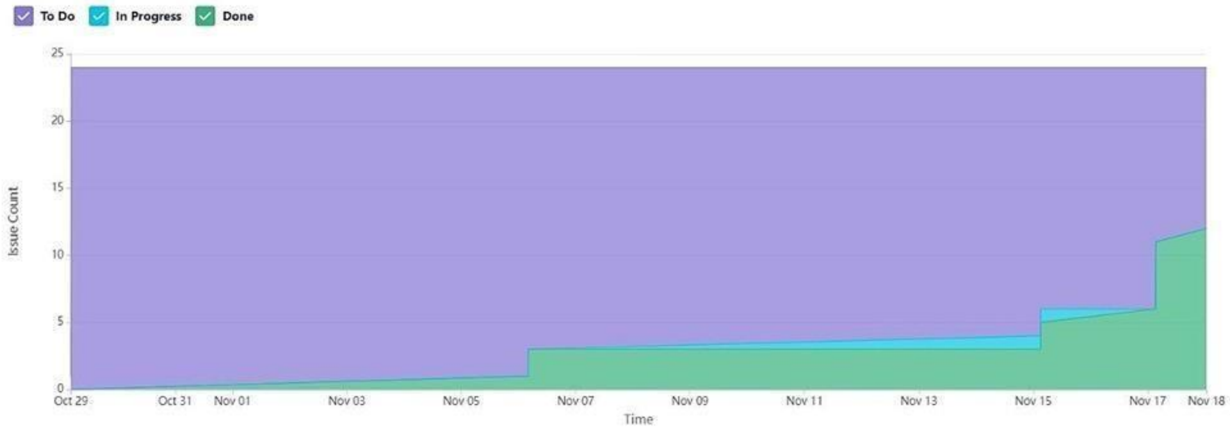
TITLE	DESCRIPTION	DUE DATE
Literature survey & Information gathering	Literature survey on the selected topic and collect information by referring to the related papers and research projects, journals etc.	3 September 2022
Prepare Empathy Map	Prepare empathy map canvas to understand about the user problems, pains and gains. From the empathised details, prepare the problem statements to be solved.	10 September 2022
Ideation	Conduct a brainstorming session with the teammates and discuss ideas to solve the problem. Prioritize the top 3 ideas based on feasibility.	17 September 2022
Proposed Solution	Prepare the proposed solution which includes the novelty, feasibility, revenue, social impact, scalability etc.	24 September 2022
Problem Solution Fit	Prepare the problem solution fit which includes the causes, problems and solutions of the problem.	1 October 2022
Solution Architecture	Prepare solution architecture that indicates the data flow from the user, model and the website.	1 October 2022

Customer Journey	Prepare the customer journey map to understand the user needs and experience with the application.	8 October 2022
Functional Requirement	Prepare the functional requirement which includes all the features that will be available in the application.	15 October 2022
Technology Architecture	Prepare the technology architecture that defines about the technologies and the IBM cloud features used in the application.	15 October 2022
Data Flow Diagrams	Draw the data flow diagram to indicate the data flow from the user, during the model building and while predicting the result,	15 October 2022
Prepare Milestone & Activity List	Split the entire project into simpler tasks and prepare milestones and activity list of the project.	22 October 2022
Sprint Delivery Plan	Prepare a delivery plan of the project with specific due dates to complete each sprint consisting of a set of functional requirements.	22 October 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop, test and submit the code.	19 November 2022

6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points
Sprint-1	Resources Initialization	USN-1	Create and set up accounts for several open APIs, such as the Open Weather Map API.	1
Sprint-1	Local Server/Software Run	USN-1	Write a Python program that generates outputs in response to inputs such as location and weather.	1
Sprint-2	Push the server/software to cloud	USN-2	Push the code from Sprint 1 to cloud so it can be accessed from anywhere	2
Sprint-3	Hardware initialization	USN-3	Integrate the hardware so you may use it to access cloud services and provide them input.	2
Sprint-4	UI/UX Optimization & Debugging	USN-4	Optimize all the shortcomings and provide better user experience.	2

6.3 Reports from JIRA



7. CODING & SOLUTIONING

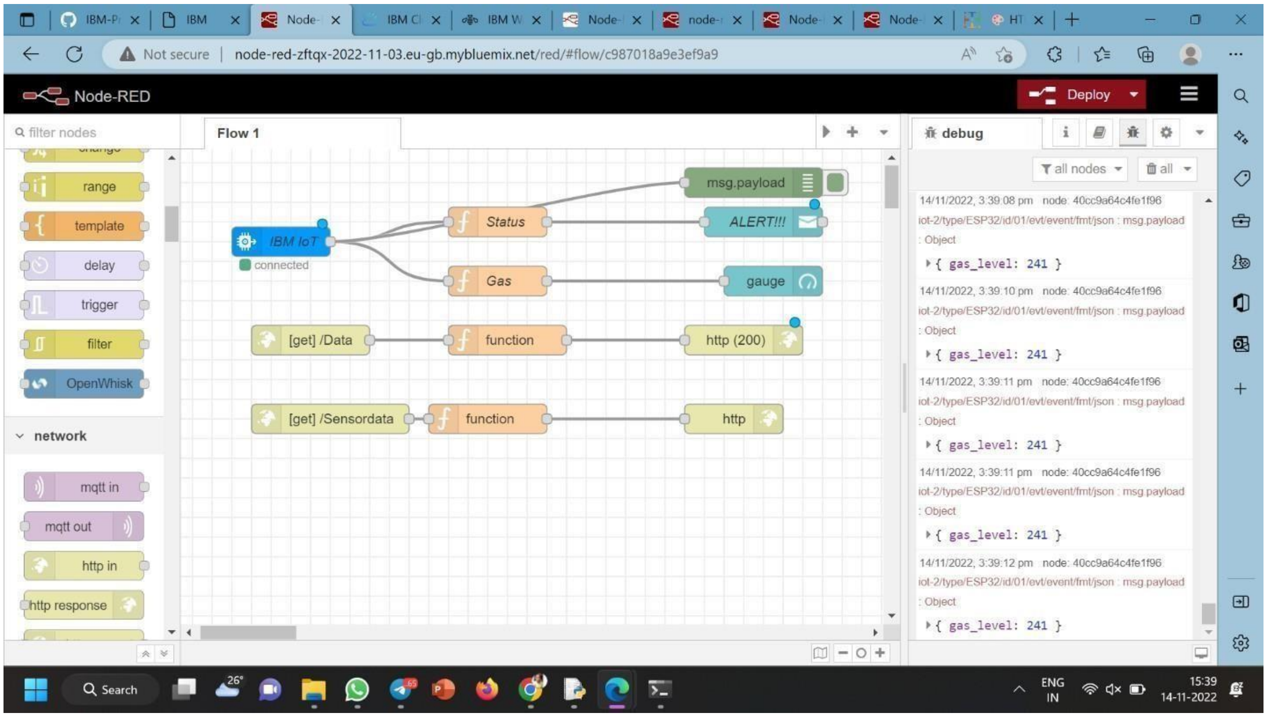
7.1 Feature 1

Fire alert using red light and web alert.

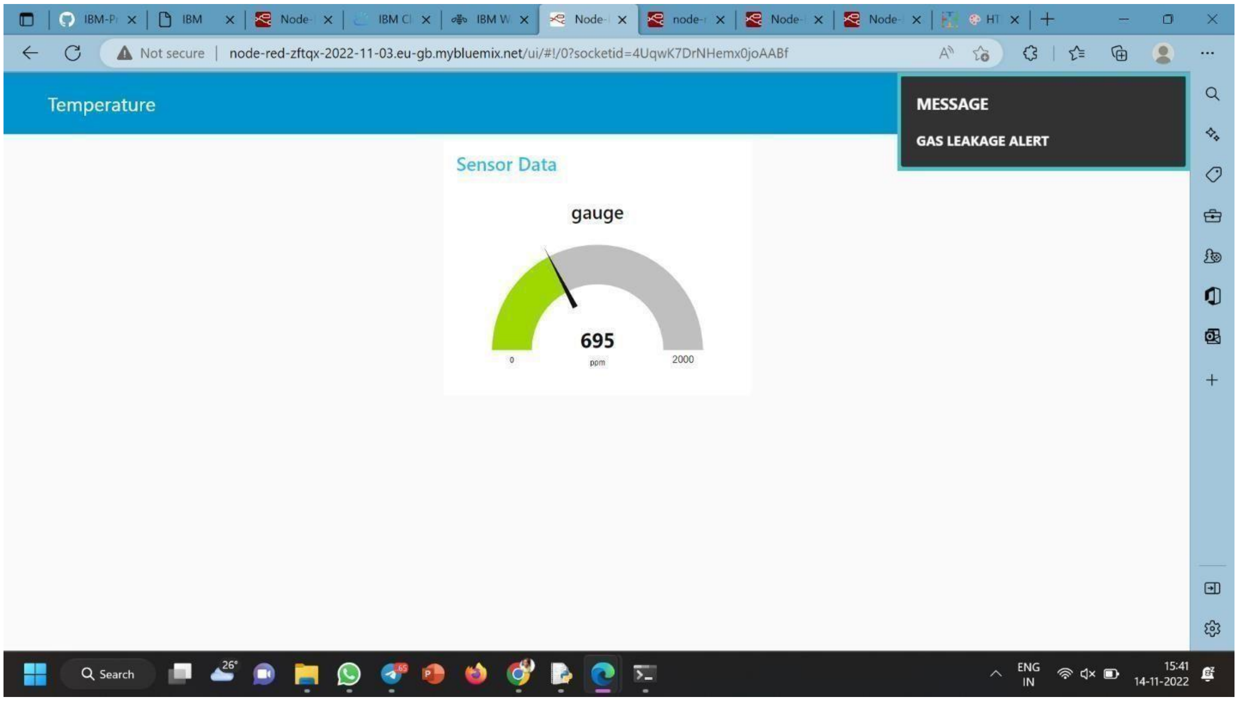
```
#include <LiquidCrystal.h>
LiquidCrystal lcd(5,6,8,9,10,11);

int redled = 2;  int
greenled = 3;  int
buzzer = 4;  int sensor
= A0;  int sensorThresh
= 400;


void setup()
{
pinMode(redled, OUTPUT);
pinMode(greenled,OUTPUT);
pinMode(buzzer,OUTPUT);
pinMode(sensor,INPUT);
Serial.begin(9600);
lcd.begin(16,2);
}

void loop()
{
  int analogValue = analogRead(sensor);
Serial.print(analogValue);    if(analogValue>sensorThresh)
  {
    digitalWrite(redled,HIGH);
digitalWrite(greenled,LOW);
tone(buzzer,1000,10000);
lcd.clear();
lcd.setCursor(0,1);
lcd.print("ALERT");
delay(1000);      lcd.clear();
lcd.setCursor(0,1);
lcd.print("EVACUATE");
delay(1000);
  }
else
  {
    digitalWrite(greenled,HIGH);
digitalWrite(redled,LOW);
noTone(buzzer);      lcd.clear();
lcd.setCursor(0,0);
lcd.print("SAFE");
delay(1000);
    lcd.clear();
lcd.setCursor(0,1);
lcd.print("ALL CLEAR");
delay(1000);
  }

}
```

7.2 Feature 2

Mobile app notification



```
14/11/2022, 3:39:08 pm  node: 40cc9a64c4fe1f96
iot-2/type/ESP32/id/01/evt/event/fmt/json : msg.payload
: Object
▸{ gas_level: 241 }
14/11/2022, 3:39:10 pm  node: 40cc9a64c4fe1f96
iot-2/type/ESP32/id/01/evt/event/fmt/json : msg.payload
: Object
▸{ gas_level: 241 }
14/11/2022, 3:39:11 pm  node: 40cc9a64c4fe1f96
iot-2/type/ESP32/id/01/evt/event/fmt/json : msg.payload
: Object
▸{ gas_level: 241 }
14/11/2022, 3:39:11 pm  node: 40cc9a64c4fe1f96
iot-2/type/ESP32/id/01/evt/event/fmt/json : msg.payload
: Object
▸{ gas_level: 241 }
14/11/2022, 3:39:12 pm  node: 40cc9a64c4fe1f96
iot-2/type/ESP32/id/01/evt/event/fmt/json : msg.payload
: Object
▸{ gas_level: 241 }
```


node-red-zftqx-2022-11-03.eu-gb.mybluemix.net/ui/#!/0?socketid=4UqwK7DrNHemx0joAABf
Temperature
Sensor Data
gauge
695
ppm
0
2000

MESSAGE

GAS LEAKAGE ALERT

8.TESTING

8.1 Test Cases

Test case ID	**Feature Type**	**Component**	**Test Scenario**	**PreRequisit e**	**Steps To Execute**	**Expected Result**	**Actual Result**	**Status**	**links**	**Executed By**
TC_OO1	Functional	IBM cloud	Create the IBM Cloud services which are being used in this project.	IBM Clou d Logi n ID & Password	1.Go to IBM Cloud signup page 2.Enter e-mail id and other credentials 3.Enter a password	User shoul d sign up IBM cloud and detail s should be verifie d	Working as expected	Pass	https://cloud.ibm.com/login	Hariharan G
TC_OO2	Functional	IBM Cloud	Configure the	IBM Clou d	1.Go to Cloud login	User login to	Working as exp	Pass	https://cloud.ibm.com/login	Hariharan T

			IBM Cloud services which are being used in completing this project.	Login ID & Password	2.Enter user ID & Password 3.Verify login by the popup display	IBM Cloud and should be navigated to IBM Cloud dashboard page	ected			
TC_OO3	Functional	IBM Watson IoT Platform	IBM Watson on IoT platform acts as the mediator to connect the web application to IoT devices,	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	User should be navigated to IBM IoT Watson Platform	Working as expected	Pass	https://oyi7sh.internetofthings.ibmcloud.com/dashboard/	Dharanesh

			so create the IBM Watson IoT platform.		and click Sign in IBM Watson Platfor m					

TC_ OO 4	Func tion al	IBM Wats on	In order to conn ect the IoT devic e to the IBM cloud , creat e a devic e in the IBM Wats on IoT platf orm and get the devic e cred entia ls.	IBM Wat son IoT Platf orm Logi n ID & Pass wor d	1.Login to I BM Watson Platfor m 2. Click Add Device 3.Enter the details and click Finish. Create Device ID & Device type 4.Turn on Device Simulat or an d click simulati on running. Enter the values of g a s,	Temp eratur e, Humi dity and Gas senso r value s shoul d be rando mly gener ate	Wo rkin g as exp ect ed	Pa ss	Temperature, Humidity and Gas sensor values are generated randomly in simulation	Arun prabu AS

					temperature & humidity level 5.Click Send & Save. Verify the displayed result of the levels					
TC_OO5	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 Platf	Node Red Installation	1.Install node red and open node red in command prompt 2.Select IBM input in IoT	User should be able to see the Node Red page	Working as expected	Pass	https://cloud.ibm.com/developer/appservice/cre ate-app?starterKit=59c9d5bd -4d31-3611-897af94eea80dc9f&defaultLanguage=undefined	Arun prabu AS

			orm.							

TC_OO6	Functional	Node Red	Create a Node-RED service.	Node Red Installation	1.Select IBM IoT input in Node. In IBM IoT Watson Platfor m, go to apps and click on generate API keys. 2.Copy & paste generat ed 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 rename as Msg.payload and	Value s of senso rs and butto n for Alarm & Sprinkler ON/OFF should be displayed	Working as expected	Pass	Values of sensors and button for Alarm & Sprinkler ON/OFF is displayed	Arun prabu AS

					click on done. Click gauge from the dashboard and fill the details & add functions to the gauge. Check the generated values from the debug message. 4.Edit function node, connect them, add another gauge and functions, name them as "Temperature", "Gas" &"Humidity"						

					5.Finally add alarm ON/OFF and Sprinkler ON/OFF buttons to the IBM IoT and debug. Verify the output from NODE RED using Local host link					
TC_OO7	Functional	Python 3.7.0	Develop a python script to publish random sensor data such as temperature, humidity level and	Python 3.7.0 (64 bit) installation	1.Download and install Python 3.7.0 2.Develop python code	User should be able to develop a python code	Working as expected	Pass	https://www.python.org/downloads/release/python-370/	Dharaaneshwaran

			Gas level to the IBM IoT platform							

TC_OO8	Functional	Python 3.7.0	After developing pytho n code, commands are received just print the statements which represent the contr ol of the devic es.	Python 3.7.0 (64 bit) installation	1.Downl install Python 3.7.0 2.After python code	User should be able to get the results from the developed code	Working as expected	Pass	Get the output from the code	Dharaaneshwaran
TC_OO9	Functional	IBM Cloudant DB	Publis h Data to The IBM Cloud	IBM Cloud Login ID & Password	1.Run the python code 2.Verify the displayed output	User should be able to publish the code	Working as expected	Pass	Publishment of python code	Dharaaneshwaran
TC_OO10	Web UI	Node Red	Creat e Web	MIT Inve	1.Go to Node	Sensors	Working as	Pass	Sensors values and command values can be seen in the mobile	Dharaaneshwaran

		& MIT Inventor	UI in Node - Red	ntor Login ID & password	Red. Select http in & http response. A dd functions and select another http in and http response. Connect them to IBM IoT output and function .Print the command statements such as Sprinkler ON/OFF , Alarm ON/OFF and sensor 2.Go to MIT app inventor and create fronten	value s and comm and values should be seen in the mobile application	expected		application	

					d using buttons, horizontal arrangement, text bar, etc. Add blocks and so on to create back end. Verify the output					

TC_OO11	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	User should be able to connect the Cloudant and Node Red	Working as expected	Pass	Cloudant is connected by NODE RED	Dharanesh

8.2 User Acceptance Testing

1. *Defect Analysis*

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	6	3	2	2	13
Duplicate	1	0	3	0	4
External	2	2	0	1	5
Fixed	7	3	4	5	19
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	16	13	13	10	52

2. *Test Case Analysis*

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	5	0	0	5
Client Application	9	0	0	9
Security	3	0	0	3
Outsource Shipping	1	0	0	1
Exception Reporting	2	0	0	2
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9.RESULTS

9.1 Performance Metrics

S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Load/Volume Changes	Risk Score	Justification
1	Alarm ON/OFF	Existing	Low	No Changes	Low	>5 to 10%	GREEN	Changes occurs less
2	Sensor values	Existing	Moderate	No Changes	Moderate	>10 to 30%	ORANGE	Some changes occurs

S.No	Project Overview	NFT Test approach	Approvals/SignOff	Assumptions/Dependencies/Risks
1	.ino(ardunio)	ino coding	wokwi.com	Depend on the delivered code
2	Node Red	Sensor & command values	https://nodered.org/	Sensor values
3	MIT Inventor	Alarm/Sprinkler/Sensors notification	https://appinventor.mit.edu/about/termsofservice	Notifications

S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NOGO decision	Identified Defects (Detected/Closed/Open)	Recommendations	Approvals/SignOff
1	.ino(ardunio)	ardunio coding	Met	Pass	GO	Closed	Efficient code	wokwi.com
2	Node Red	Sensors&command values	Met	Pass	GO	Closed	Sensing the values perfectly	https://nodered.org/

3	MIT Inventor	Alarm/Sprinkler/Sensors notification	Met	Pass	GO	Closed	Notifies the users at correct time	https://appinventor.mit.edu/about/termsofservice

10.ADVANTAGES & DISADVANTAGES

Advantages:

- Detect the concentration of the gases
- The sensor-enabled solution helps prevent the high risk of gas explosions and affecting any casualties within and outside the premises. ● Get real-time alerts about the gaseous presence in the atmosphere
- Prevent fire hazards and explosions
- Supervise gas concentration levels
- Ensure worker’s health
- Real-time updates about leakages
- Cost-effective installation
- Data analytics for improved decisions
- Measure oxygen level accuracy
- Get immediate gas leak alerts **Disadvantages:**
- Only one gas can be measured with each instrument.
- When heavy dust, steam or fog blocks the laser beam, the system will not be able to take measurements

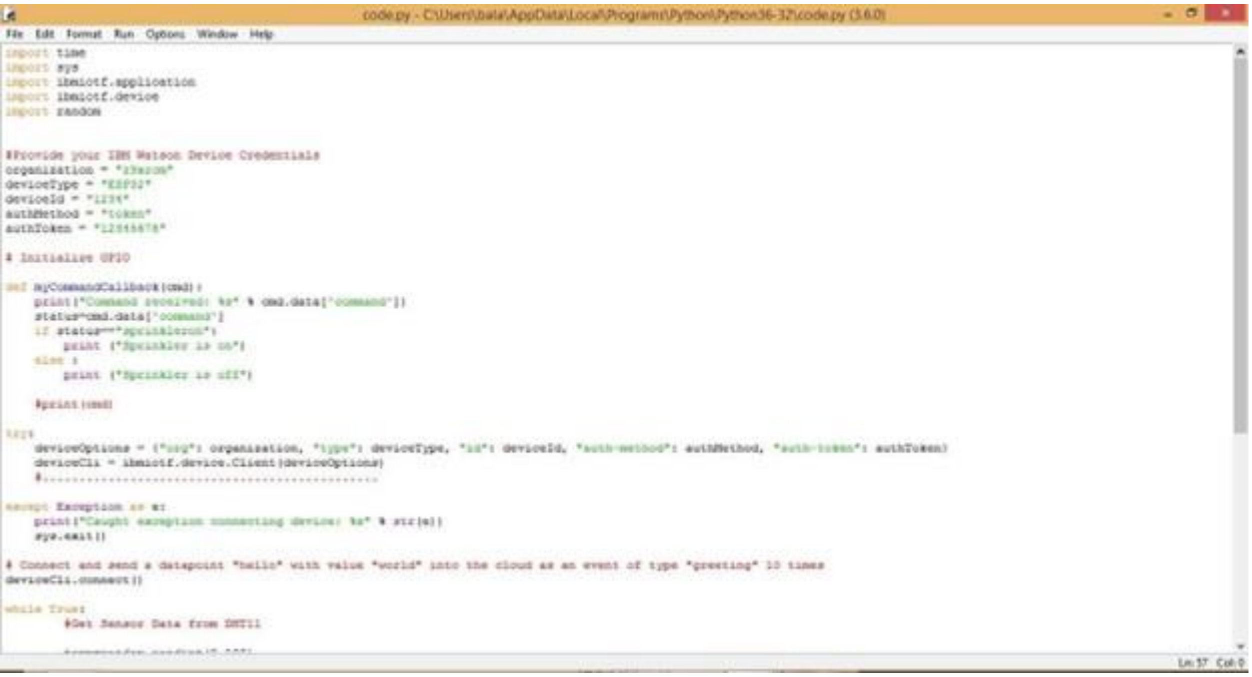
11.CONCLUSION

Gas leaks cause serious disasters that result in property damage and human injuries. The main causes of gas leaks are poor equipment upkeep and a lack of public awareness. As a result, detecting LPG leaks is critical for avoiding accidents and saving human lives. This paper discussed a system for detecting and alerting LPG leaks. Whenever LPG leakage is detected, this device activates an LED and a buzzer to inform people. This approach is straightforward but dependable. Internet of Things has gained its wide popularity in recent days due to its various streams of applications which has paved way for smooth, safe and easier mode of living style for human beings. One such area of applications includes gas booking and gas leakage detection for both domestic and commercial purposes. Though, several techniques is existing for the same, yet gas leakage detection is one major concern and a challenge.

12.FUTURE SCOPE

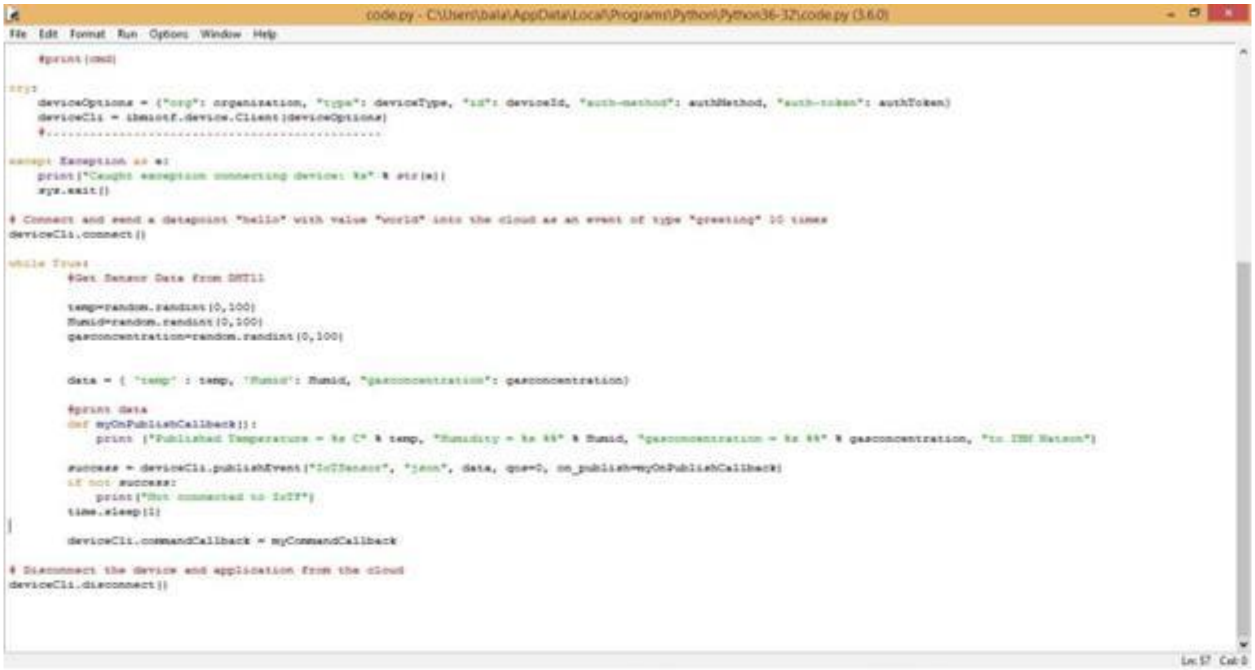
In the future, instead of using AC power, the gas leakage detecting system might be created using photovoltaic panels with a battery as a backup power supply to give a continuous supply, as opposed to the current use of AC power. The protection system employs a combination of MQ6 gas sensors, DHT22 temperature sensors, load sensors, smoke and flame sensors, and PIR sensors. A number of sensors must be calculated, taking into account the room's volume, installation position, and other factors. This system assures that if a gas leak happens, it can be tracked more effectively and that occupants may be notified ahead of time, regardless of whether the leak is visible or not, whether the house is vacant or occupied. The best recommendation for a monitoring system is to utilize a WiFi module that allows the user to monitor the gas level in real-time and automate direct management of the safety device system if an unanticipated occurrence occurs. Finally, the safety device employed was the most vital and important aspect. We also suggested that a tripper circuit be built, which would automatically turn off the (MSB) in the event of a fire, and turn off the gas regulator valve via a solenoid valve either from the cylinder from the main switchboard, If an incident occurs, it automatically can switch on the exhaust fan to suck gas to the outside house and sound an alarm and audio buzzer to inform the user or persons around and the user can opening the window, This device monitors the gas and detects any leaks in order to keep people safe.

13.APPENDIX Source Code

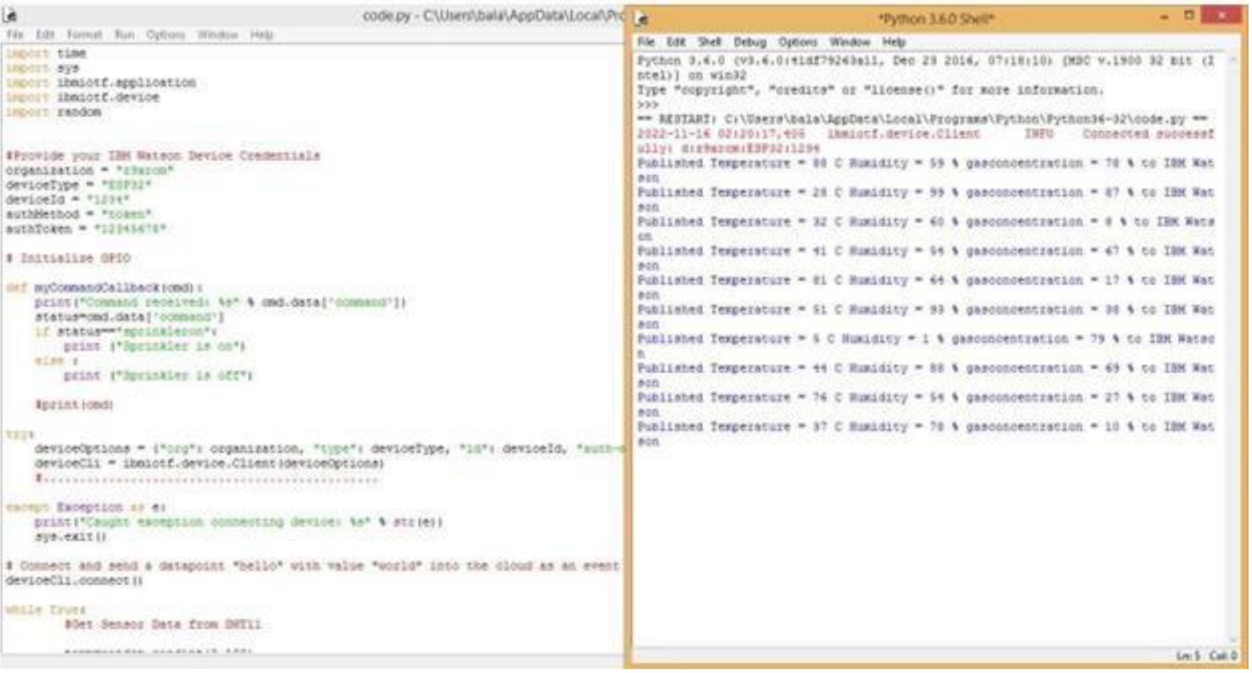


code.py - C:\Users\bala\AppData\Local\Programs\Python\Python36-32\code.py (3.6.0)
File Edit Format Run Options Window Help
Ln: 37 Col: 0

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


#Provide your IBM Watson Device Credentials
organization = "i3hzcm"
deviceType = "ESP32"
deviceId = "1234"
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")
    else :
        print ("Sprinkler is off")

    #print(cmd)

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

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

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

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


code.py - C:\Users\bala\AppData\Local\Programs\Python\Python36-32\code.py (3.6.0)
File Edit Format Run Options Window Help
Ln: 57 Col: 0

```
    #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)
        Humid=random.randint(0,100)
        gasconcentration=random.randint(0,100)


        data = { 'temp' : temp, 'Humid': Humid, "gasconcentration": gasconcentration}

        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "gasconcentration = %s %%" % gasconcentration, "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()
```


code.py - C:\Users\bala\AppData\Local\Pro
File Edit Format Run Options Window Help
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Ln: 5 Col: 0

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


#Provide your IBM Watson Device Credentials
organization = "z9arom"
deviceType = "ESP32"
deviceId = "1234"
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")
    else :
        print ("Sprinkler is off")

    #print(cmd)

try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-m
    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
deviceCli.connect()

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

```
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900 32 bit (I
ntel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:\Users\bala\AppData\Local\Programs\Python\Python36-32\code.py ==
2022-11-16 02:20:17,406   ibmiotf.device.Client      INFO    Connected successf
ully: d:z9arom:ESP32:1234
Published Temperature = 80 C Humidity = 59 % gasconcentration = 78 % to IBM Wat
son
Published Temperature = 28 C Humidity = 99 % gasconcentration = 87 % to IBM Wat
son
Published Temperature = 32 C Humidity = 60 % gasconcentration = 8 % to IBM Wats
on
Published Temperature = 41 C Humidity = 54 % gasconcentration = 47 % to IBM Wat
son
Published Temperature = 81 C Humidity = 64 % gasconcentration = 17 % to IBM Wat
son
Published Temperature = 51 C Humidity = 93 % gasconcentration = 38 % to IBM Wat
son
Published Temperature = 5 C Humidity = 1 % gasconcentration = 79 % to IBM Watso
n
Published Temperature = 44 C Humidity = 88 % gasconcentration = 69 % to IBM Wat
son
Published Temperature = 76 C Humidity = 54 % gasconcentration = 27 % to IBM Wat
son
Published Temperature = 37 C Humidity = 78 % gasconcentration = 10 % to IBM Wat
son
```

14.GitHub & Project Demo Link

Github Link : https://github.com/IBM-EPBL/IBM-Project-10096-1659093755

Demonstration video Link : https://drive.google.com/drive/folders/1gMmlE_5pzynHEzsrv0VOf-JopfoTBHNz