IBM PROJECT

GAS LEAKAGE MONITORING AND ALERTING SYSTEM FOR INDUSTRIES

Team ID:PNT2022TMID02488

Team Leader:

RAGHUL R

Team Members:

- PRAVEENKUMAR S
- SARUHASSAN T
- GOKUL M

CONTENTS

1. INTRODUCTION

1.1 Project Overview:

This project helps the industries in monitoring the emission of harmful gases. In several areas, the integration of gas sensors helps in monitoring the gas leakage. If any gas leakage is detected the admins will be notified along with the location. In the web application, admins can view the sensor parameters. The Internet of things (IoT) is the system of gadgets, vehicles, and home machines that contain hardware, programming, actuators, and network which enables these things to interface, collaborate and trade information. IoT includes broadening Internet network past standard device, for example, work areas, workstations, cell phones and tablets, to any scope of generally stupid or non-web empowered physical device and ordinary articles. Installed with innovation, these gadgets can convey and connect over the Internet, and they can be remotely observed and controlled. The meaning of the Internet of things has advanced because of union of numerous innovations, ongoing examination, AI, ware sensors, and implanted frameworks. Conventional fields of installed frameworks, remote sensor systems, control frameworks computerization (counting home and building mechanization), and others all add to empowering the Internet of things. A gas spill alludes to a hole of petroleum gas or different vaporous item from a pipeline or other regulation into any territory where the gas ought not be available. Since a little hole may steadily develop a hazardous convergence of gas, spills are perilous. Notwithstanding causing flame and blast dangers, holes can slaughter vegetation, including huge trees, and may discharge amazing ozone harming substances to the environment. Keywords: IOT, MQ5 sensor, Arduino module, GSM networks.

1.2 Purpose:

Inhaling concentrated gas can lead to asphyxia and possible death.To overcome these disasters, we designed a system for monitoring and alerting the leakage of those harmful gases. This makes the industrialists get rid of the fear of any disasters caused by the gases.

2. LITERATURE SURVEY

2.1 Existing Problem:

The number of sensors is unpredictable and the positioning of equipment is improper and also the affordable of the system is high and the systems are sometimes causing heavy disasters.

2.2 References:

1) IoT based Gas leakage detection system with database logging,prediction and smart alerting system.

AUTHOR*:* Chaitali Bagwe, Vidya Ghadi, Vinayshri Naik, Neha Kunte

CONTENT*:* The system provides constant monitoring and detection of gas leakage with storage of data in database for predictions and analysis. The IOT components used helps in making the system much more cost effective in comparison with traditional Gas detector systems.

*2)*Internet of things (IoT) based gas leakage monitoring and alerting system with Mq-6 Sensor.

AUTHOR: Rohan Chandra Pandey , Manish Verma , Lumesh Kumar Sahu , Saurabh Deshmukh

CONTENT: An overall conclusion IOT based toxic gas detector is it has become more efficient, more applicable to today's applications and smarter.

*3)*Internet of Things (IOT) Based Gas Leakage Monitoring and Alerting System with Mq-2 Sensor.

AUTHOR: Rohan Chandra Pandey , Manish Verma , Lumesh Kumar Sahu

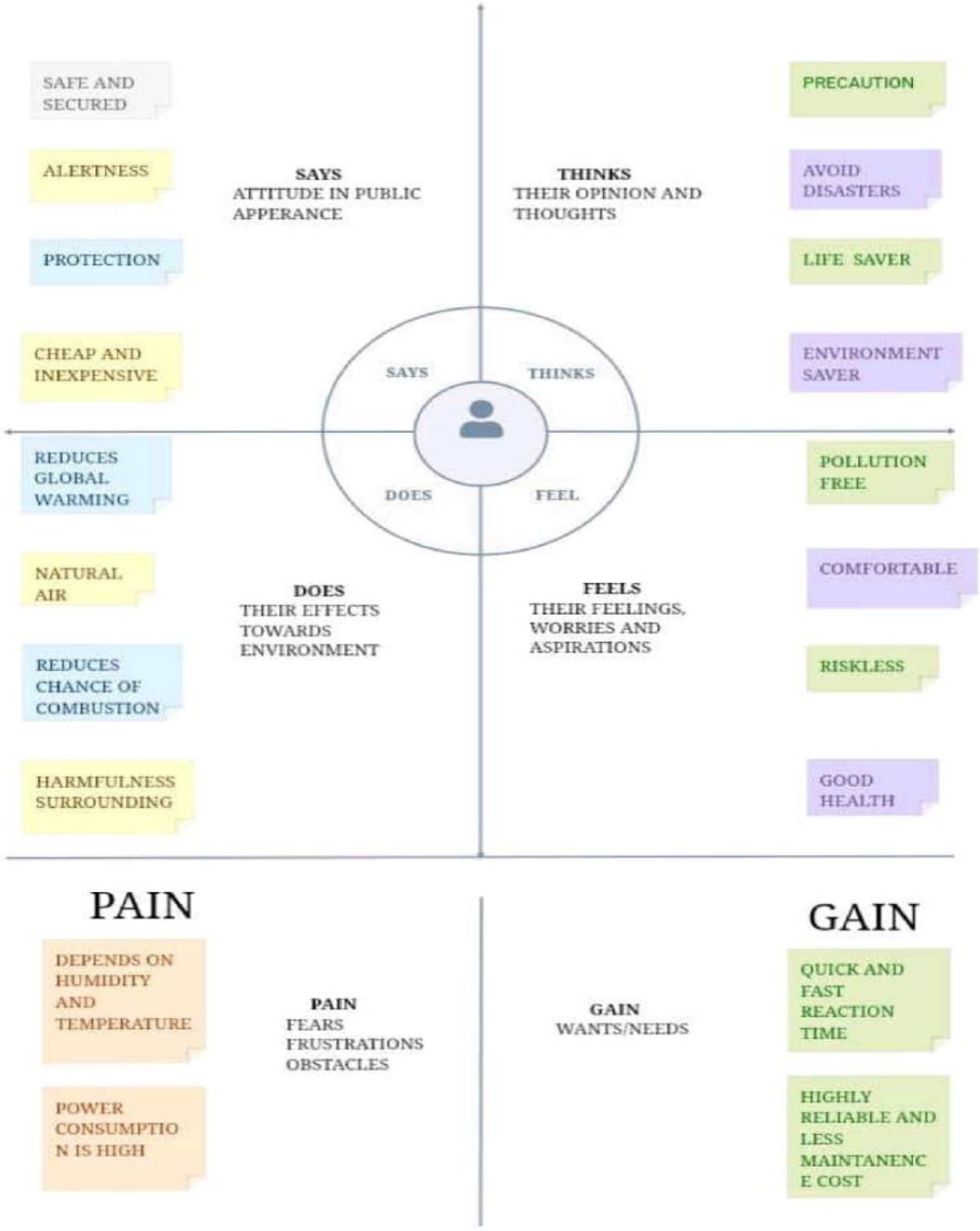
CONTENT: This paper choice of using a real time gas leakage monitoring and Sensing the output levels of gas has been clearly observed by the help of this system

2.3 Problem statement definition:

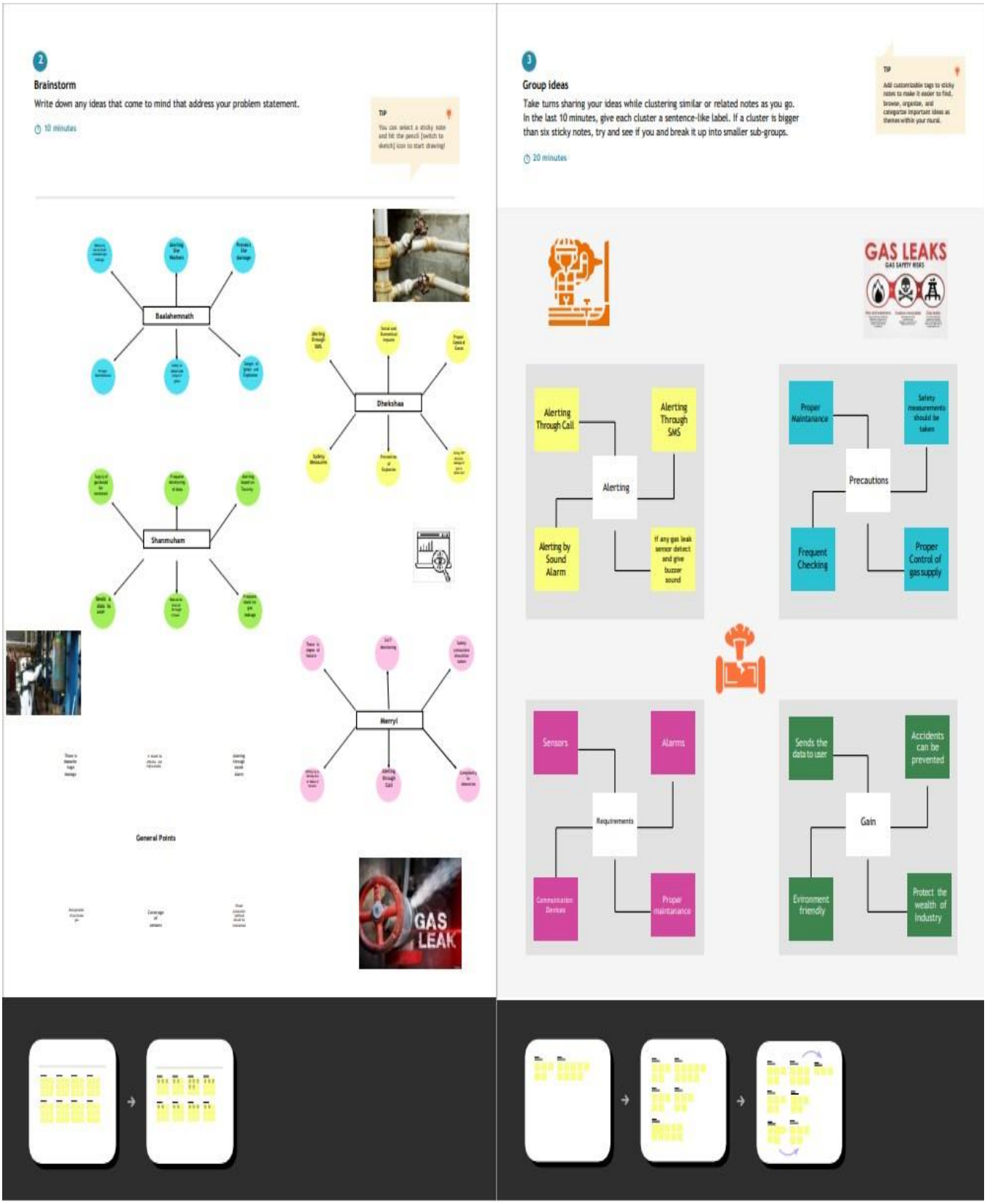
Since the number of sensors is unpredictable, the industrialists feel in secured in handling the gases. Also the cost price of the products and the complications in installing the systems are high. This makes the customers feel disappointed sometimes.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:



3.2 Ideation & Brainstorming:

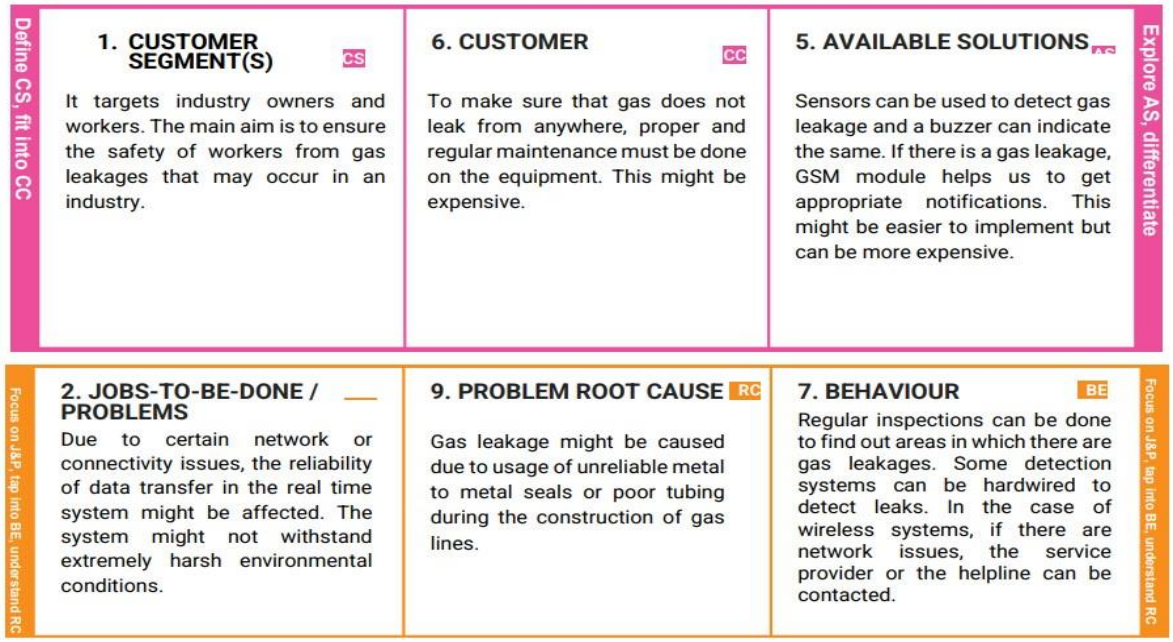


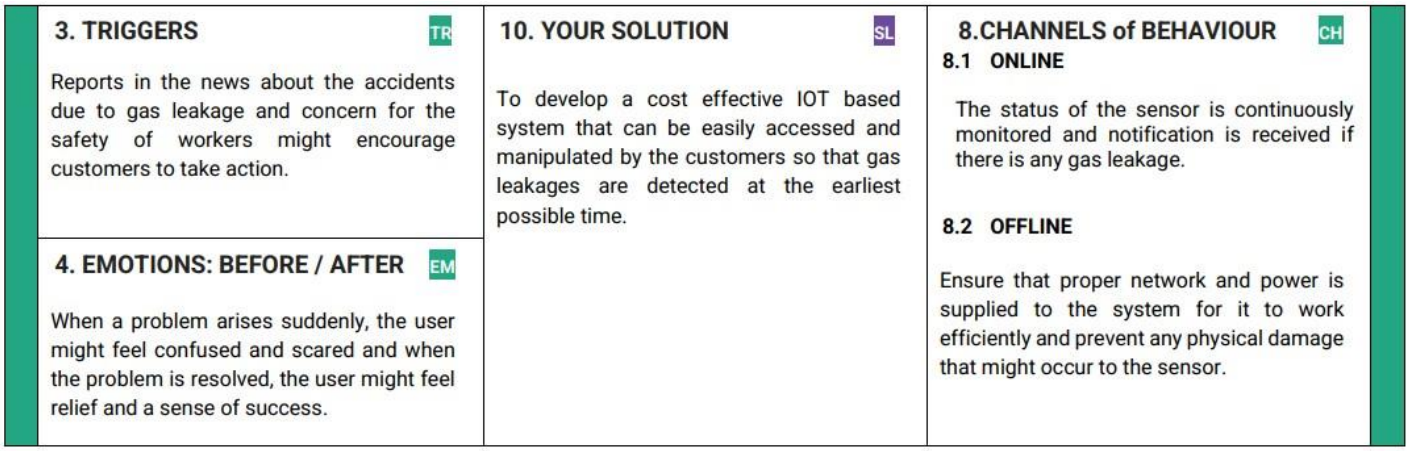
3.3 Proposed Solution:

S.No.	Parameter	Description
1	**Problem Statement**	Gas leakage is the main problem of the industrial sector, residential areas and gas-powered vehicles such as CNG (Compressed Natural Gas) buses and cars. Gas Leakage fires took an increasing toll on lives and property in recent years. Most Gases used for industrial activities are highly inflammable and will even catch fire from the source of the leak. The leakage of gases only may be detected by humans nearby and if there are not any humans nearby, it cannot be detected. But sometimes it cannot be detected by humans who have a coffee sense of smell. Therefore, in such cases Industries are in need of a Gas Leakage Detection System for gas leakage identification and to perform certain measures in order to scale back the impact incase of any fire and to alert people/workers about the gas leakage.
2	**Idea / Solution description**	To create a device which periodically monitors the level of the gas in the area of interest and updates the status in the server, which can be viewed using an application. It continuously listens to the level of gas in the atmosphere and provides a warning using a buzzer and provides the alert to the incharge people if any leakage occurs via the GSM network.
3	**Novelty / Uniqueness**	A safety has been a major issue in today's day to day life. LPG and CNG i.e. petroleum gas and compressed natural gas are most commonly used in residential and commercial places for cooking purpose and in various vehicles as a replacement for costly fuels like diesel, petrol.These gases are filled in cylinders which are easily un-damageable. But leakage can take place through pipes or regulators or knobs which may cause

		accidents like suffocation, uneasiness or sometimes may catch fire and short circuit as well.
4	**Social Impact / Customer Satisfaction**	Explosions caused by unidentified gas leaks are dangerous for the workers who are exposed to a dangerous atmosphere. For increased safety, it becomes necessary to implement smart systems to precisely identify combustible, flammable, and hazardous gases as well as detect oxygen depletion in industrial buildings. A gas detection system is a fundamental necessity for safety in the oil and gas, hospital, and hotel industries, as well as other settings where dangerous gases are frequently employed.
5	**Business Model (Revenue Model)**	The product can be made compact, cost efficient and easily installable so that all the industries from small scale to large scale can afford to buy the product which creates more profit. It can even be used for domestic purposes for LPG gases.
6	**Scalability of the Solution**	A mobile application can be developed that can provide details about the amount of gas present in the region, set reminders to check gas levels, and anticipate gas leaks by providing values. To increase safety, relay motors can be added to the system. In the event that the gas concentration exceeds a certain threshold, these motors have the ability to turn off the main power and gas supplies.

3.4 Problem Solution Fit:





4. REQUIREMENT ANALYSIS

4.1 Functional Requirement:

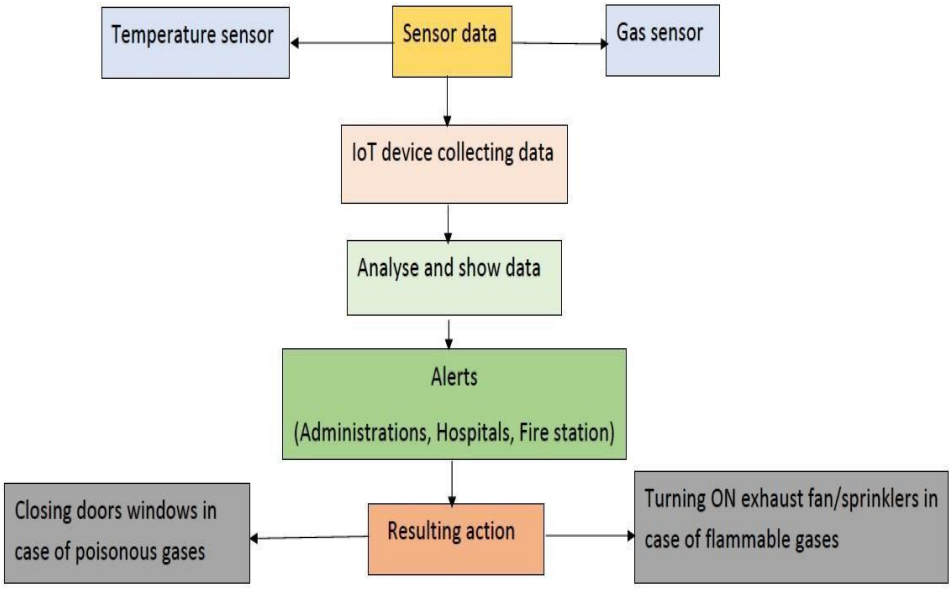
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Visibility	The level of gas can be monitored by users if there is any leakage, alerts can be sent through messages.
FR-2	User Reception	The data like the level of gas can be sent through messages
FR-3	User Understanding	The user can monitor the level of gas with the help of the data. If there is an increase in gas level, then the alert will be given. They also get notified by the alert.
FR-4	User Convenience	Through messages we can easily get data of gas level and in case of gas leakage, it can directly send notifications to nearby police stations and hospitals.
FR-5	User Performance	When the user gets notified, he could turn on the exhaust fan/sprinkler.

4.2 Non-Functional Requirement:

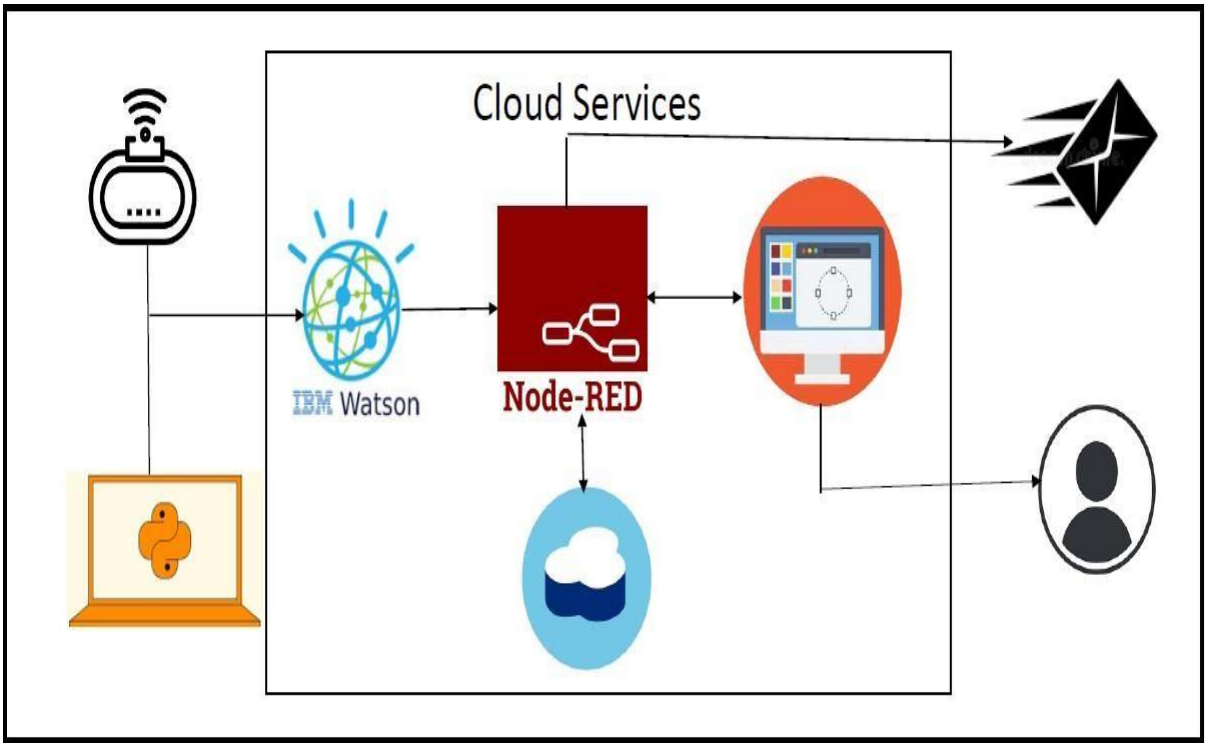
Business Requirements	User Requirements	Product Requirements
The said system can be deployed in homes, hotels, factory units, LPG cylinder storage areas, and so on. The main advantage of this IoT and Arduino-based application is that it can determine the leakage and send the data over to a site. It can be monitored, and preventive measures can be taken to avoid any disaster.	The gas leakage detection system can be optimized for detecting toxic gasses along with upgrading them with smoke and fire detectors to identify the presence of smoke and fire. Ensuring worker safety is important but making using of the right technology is even more vital.	Detecting gasses is necessary regardless of your business role or individual purpose. Certain technologies at play make such IoT devices what they are, and if you want to indulge in IoT application development, you must know what they are and what purpose they can fulfil.

5. PROJECT DESIGN

5.1 Data Flow Diagrams:



5.2 Solution & Technical Architecture:



5.3 User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	User can enter into the web application	I can access my account / dashboard	High	Sprint-1
		USN-2	Users can register their credentials like email id and password	I can receive confirmation email and click confirm	High	Sprint-1
	Login	USN-3	User can log in to the application by entering email and password	I can login to my account	High	Sprint-1
	Dashboard	USN-4	User can view the temperature	I can view the data given by the device	High	Sprint-2
		USN-5	User can view the level of gas	I can view the data given by the device	High	Sprint-2
Customer (Web user)	Usage	USN-1	User can view the webpage and get the information	I can view the data given by the device	High	Sprint-3
Customer	Working	USN-1	User act according to the alert given by the device	I can get the data work according to it	High	Sprint-3
		USN-2	User turns ON the exhaust fan/sprinkler when the leakage occurs	I can get the data work according to it	High	Sprint-4
Customer Care Executive	Action	USN-1	User solve the problems when someone faces any usage issues	I can solve the issues when someone fails to understand the procedure	High	Sprint-4
Administrator	Administration	USN-1	User stores every information	I can store the gained information	High	Sprint-4

6. PROJECT PLANNING AND SCHEDULING

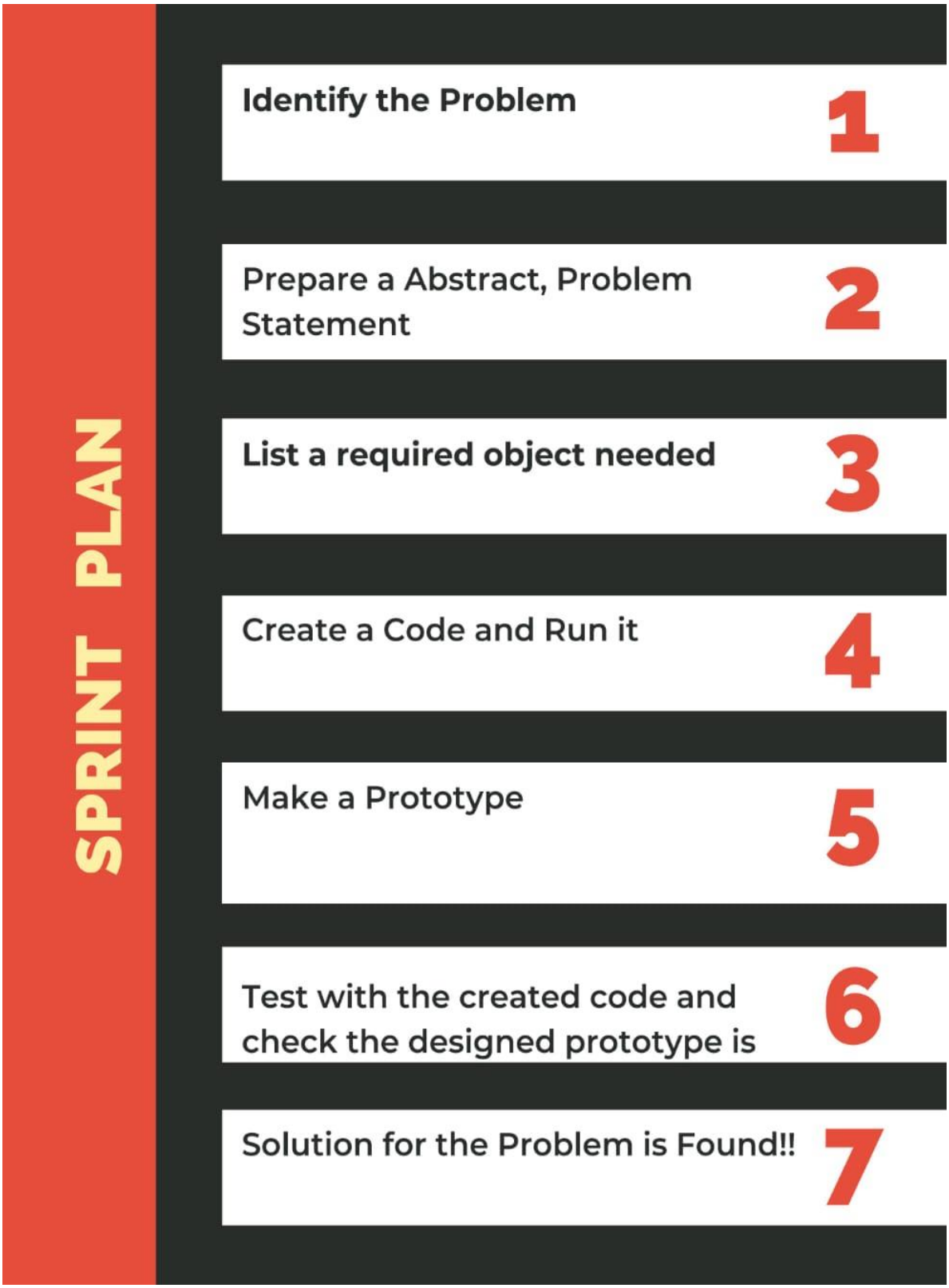
6.1 Sprint Planning & Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Objective	USN-1	As a system, the gas sensor should detect the gas	8	High
Sprint-1	Features	USN-2	As a system, the gas sensor values should be displayed on an LCD screen	2	Low
Sprint-1	Features	USN-3	As a system, as soon as the detected gas reaches the threshold level, the red colored LED should be turned ON.	5	High
Sprint-1	Features	USN-4	As a system, as soon as the detected gas reaches the threshold level, the buzzer should be turned ON.	5	High
Sprint-2	Focus	USN-5	As a system, it should the send the location at which the gas is detected	8	High
Sprint-2	Focus	USN-6	As a system, it should also send the alerting SMS to the registered phone number	2	Low

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-2	Features	USN-7	As a system, the gas leakage pipe should beclosed automatically once it attains the threshold value	5	Medium
Sprint-2	Features	USN-8	As a system, it should indicate that the gas leakage pipe is closed in the LCD screen and send SMS to the registered mobile number.	5	Medium
Sprint-3	Data Transfer	USN-9	As a program, it should retrieve the API key of the IBM cloud to send the details of the system.	2	Low
Sprint-3	Data Transfer	USN-10	As a system, it should send the data of sensor values along with latitudes and longitudes to theIBM cloud	5	Medium
Sprint-3	Data Transfer	USN-11	As a cloud system, the IBM cloud should sendthe data to NodeRed	2	Medium
Sprint-3	Data Transfer	USN-12	As a system, it should collect the data from NodeRed and give it to the backend of the MIT app.	3	Medium
Sprint-3	Data Transfer	USN-13	As an application, it should display the details ofthe gas level and other details to the user through the frontend of the MIT app.	8	High
Sprint-4	Registration	USN-14	As a user, I must first register my email andmobile number in the website	2	High

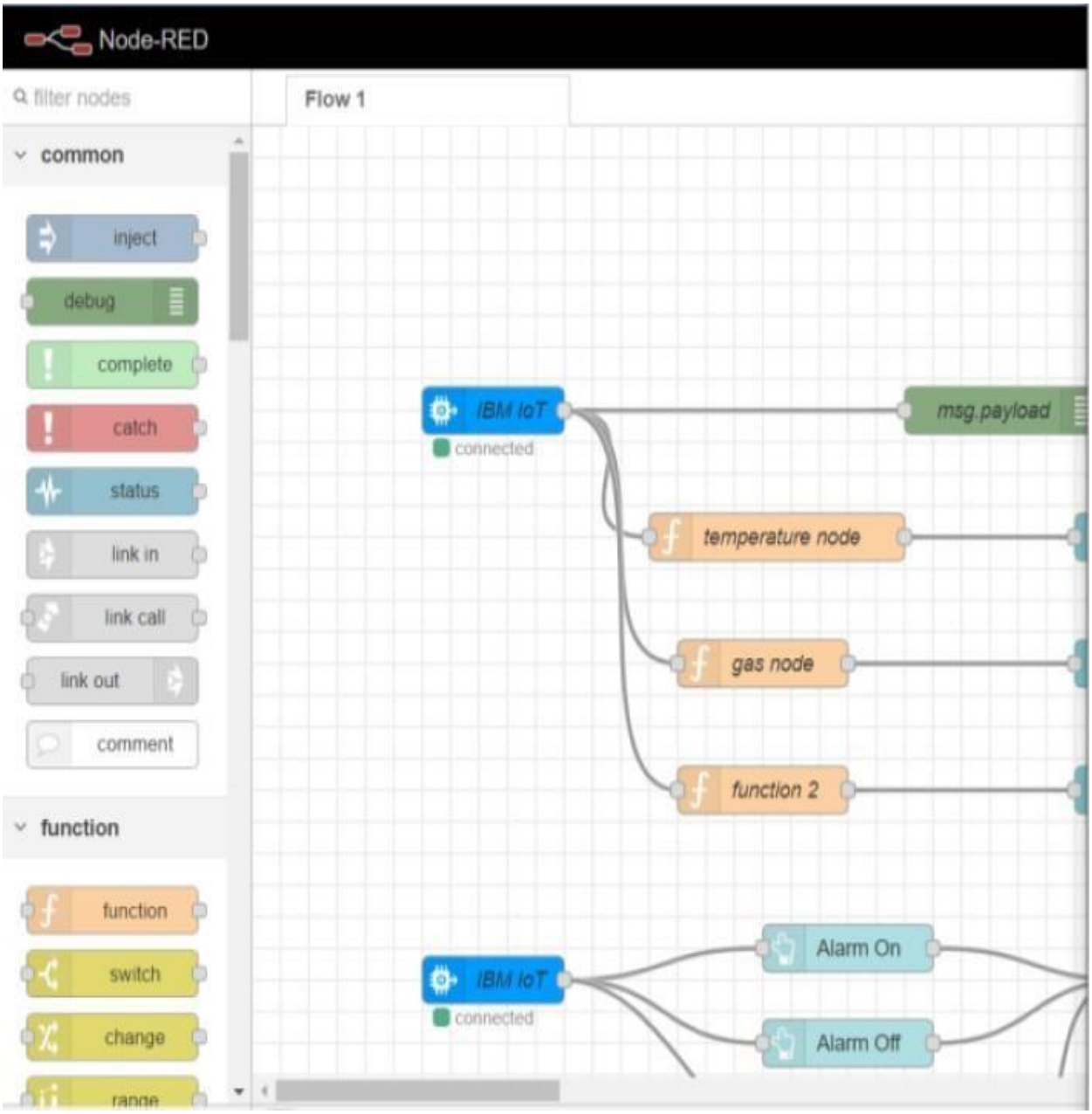
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-4	Registration	USN-15	As a user, I must receive confirmation mail and SMS on registration	2	Medium
Sprint-4	Login	USN-16	As a user, I can login into the web application through email and password.	3	High
Sprint-4	Dashboard	USN-17	As a user, I can access the dashboard andmake use of available resources.	2	Medium
Sprint-4	Focus	USN-18	As a user, I must receive an SMS once the leakage is detected.	5	High
Sprint-4	Allocation	USN-19	As an admin, I must receive information about theleakage along with location and it should share exact location and route to the person.	3	High
Sprint-4	Allocation	USN-20	As an admin, I must allot a particular person to look after the leakage in a particular location.	3	High

6.2 Sprint Delivery Schedule:



7. CODING AND SOLUTIONING

7.1 Feature 1(Node Red Output)



7.2 Feature 2: (Python Output)

```
File  Edit  Shell  Debug  Options  Window  Help
Published Temperature = 72 C Humidity = 38 % Gas_Level = 93 % to IBM Watson
Published Temperature = 29 C Humidity = 58 % Gas_Level = 63 % to IBM Watson
Published Temperature = 71 C Humidity = 14 % Gas_Level = 87 % to IBM Watson
Published Temperature = 5 C Humidity = 32 % Gas_Level = 92 % to IBM Watson
Published Temperature = 51 C Humidity = 20 % Gas_Level = 82 % to IBM Watson
Published Temperature = 87 C Humidity = 10 % Gas_Level = 62 % to IBM Watson
Published Temperature = 35 C Humidity = 14 % Gas_Level = 19 % to IBM Watson
Published Temperature = 8 C Humidity = 28 % Gas_Level = 81 % to IBM Watson
Published Temperature = 69 C Humidity = 90 % Gas_Level = 50 % to IBM Watson
Published Temperature = 39 C Humidity = 0 % Gas_Level = 51 % to IBM Watson
Published Temperature = 88 C Humidity = 62 % Gas_Level = 27 % to IBM Watson
Published Temperature = 76 C Humidity = 89 % Gas_Level = 98 % to IBM Watson
Published Temperature = 99 C Humidity = 90 % Gas_Level = 12 % to IBM Watson
Published Temperature = 93 C Humidity = 36 % Gas_Level = 7 % to IBM Watson
Published Temperature = 98 C Humidity = 23 % Gas_Level = 40 % to IBM Watson
Published Temperature = 32 C Humidity = 72 % Gas_Level = 62 % to IBM Watson
Published Temperature = 55 C Humidity = 7 % Gas_Level = 80 % to IBM Watson
Published Temperature = 100 C Humidity = 74 % Gas_Level = 29 % to IBM Watson
Published Temperature = 64 C Humidity = 86 % Gas_Level = 13 % to IBM Watson
Published Temperature = 55 C Humidity = 5 % Gas_Level = 17 % to IBM Watson
Published Temperature = 72 C Humidity = 28 % Gas_Level = 37 % to IBM Watson
Published Temperature = 10 C Humidity = 54 % Gas_Level = 65 % to IBM Watson
Published Temperature = 30 C Humidity = 82 % Gas_Level = 82 % to IBM Watson
Published Temperature = 40 C Humidity = 95 % Gas_Level = 57 % to IBM Watson
Published Temperature = 28 C Humidity = 18 % Gas_Level = 17 % to IBM Watson
Published Temperature = 47 C Humidity = 66 % Gas_Level = 50 % to IBM Watson
Published Temperature = 58 C Humidity = 86 % Gas_Level = 50 % to IBM Watson
Published Temperature = 98 C Humidity = 19 % Gas_Level = 87 % to IBM Watson
Published Temperature = 12 C Humidity = 81 % Gas_Level = 40 % to IBM Watson
Published Temperature = 32 C Humidity = 79 % Gas_Level = 75 % to IBM Watson
Published Temperature = 37 C Humidity = 80 % Gas_Level = 24 % to IBM Watson
Published Temperature = 73 C Humidity = 59 % Gas_Level = 40 % to IBM Watson
Published Temperature = 51 C Humidity = 69 % Gas_Level = 34 % to IBM Watson
Published Temperature = 96 C Humidity = 13 % Gas_Level = 68 % to IBM Watson
Published Temperature = 28 C Humidity = 62 % Gas_Level = 7 % to IBM Watson
Published Temperature = 86 C Humidity = 69 % Gas_Level = 34 % to IBM Watson
Published Temperature = 48 C Humidity = 5 % Gas_Level = 40 % to IBM Watson
Published Temperature = 20 C Humidity = 51 % Gas_Level = 78 % to IBM Watson
Published Temperature = 60 C Humidity = 2 % Gas_Level = 91 % to IBM Watson
Published Temperature = 42 C Humidity = 86 % Gas_Level = 64 % to IBM Watson
Published Temperature = 95 C Humidity = 47 % Gas_Level = 99 % to IBM Watson
Published Temperature = 49 C Humidity = 16 % Gas_Level = 84 % to IBM Watson
Published Temperature = 59 C Humidity = 25 % Gas_Level = 66 % to IBM Watson
Published Temperature = 85 C Humidity = 100 % Gas_Level = 56 % to IBM Watson
Published Temperature = 65 C Humidity = 73 % Gas_Level = 13 % to IBM Watson
Published Temperature = 48 C Humidity = 38 % Gas_Level = 38 % to IBM Watson
```

8. TESTING

8.1 Test Case Report

8.2 User Acceptance Test:

9. RESULTS

9.1 Performance Testing Report :

10. ADVANTAGES AND 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 alert

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 leakage leads to severe accidents resulting in material losses and human injuries.Gas leakage occurs due to poor maintenance of equipment and inadequate awareness of the people. Hence, gas leakage detection is essential to prevent accidents and to save human lives. This paper presented LPG leakage detection and alert system. This system triggers buzzer and notification to alert people when gas leakage is detected. This system is basic yet reliable**.**

12. FUTURE SCOPE

Major cities of India are pushing Smart Home application, gas monitoring system is a part of Smarthome application. Enhancing Industrial Safety using IoT. This system can be implemented in Industries, Hotels and wherever the gas cylinders are used. This system can be used in industries involving applications such as Furnace, Boilers, Gas welding, Gas cutting, Steel Plants, Metallurgical industries, Food processing Industries, Glass Industries, Plastic industries, Pharmaceuticals, Aerosol manufacturing. As hospitals require to provide maximum possible safety to patients, this system can be used to keep track of all the cylinders used in it. Some of the cylinders used are Oxygen cylinder, Carbon dioxide cylinder, Nitrous oxide cylinder. As many students are naive the risk of causing accidents is high. Hence, our system can also be used in schools, colleges. Many colleges have well established labs including chemistry lab and pharmaceutical labs where gas burners are used. Several medical equipment requires gas cylinders.

13. APPENDIX

Source Code:

```
 #include <LiquidCrystal.h>
LiquidCrystal lcd(6, 7, 8, 9, 10, 11);
float gasPin = A0;
float gasLevel;
int ledPin = 2;
int buttonPin = 3;
int buzzPin = 4;
int buttonState;
int fan = 5;

void setup(){
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
  pinMode(gasPin,INPUT);
  pinMode(fan,OUTPUT);
  Serial.begin(9600);
  lcd.begin(16, 2);
  lcd.setCursor(0,0);
  lcd.print(" Welcome");
  lcd.setCursor(0,2);
  lcd.print(" Youtube");
  delay(500);
  lcd.clear();
}

void loop(){
  // Read the value from gas sensor and button
  gasLevel = analogRead(gasPin);
  buttonState = digitalRead(buttonPin);

  // call the function for gas detection and button work
  gasDetected(gasLevel);
  buzzer(gasLevel);
  exhaustFanOn(buttonState);
}

// Gas Leakage Detection & Automatic Alarm and Fan ON
void gasDetected(float gasLevel){
  if(gasLevel >= 300){
   digitalWrite(buzzPin,HIGH);
    digitalWrite(ledPin,HIGH);
    digitalWrite(fan,HIGH);
```

```
    lcd.setCursor(0,0);
   lcd.print("GAS:");
    lcd.print(gasLevel);
   lcd.setCursor(0,2);
   lcd.print("FAN ON");
   delay(1000);
   lcd.clear();
  }else{
   digitalWrite(ledPin,LOW);
    digitalWrite(buzzPin,LOW);
    digitalWrite(fan,LOW);
    lcd.setCursor(0,0);
   lcd.print("GAS:");
    lcd.print(gasLevel);
   lcd.setCursor(0,2);
   lcd.print("FAN OFF");
   delay(1000);
   lcd.clear();
  }
}
//BUZZER
void buzzer(float gasLevel){
if(gasLevel>=300)
  {
  for(int i=0; i<=30; i=i+10)
  {
  tone(4,i);
  delay(400);
  noTone(4);
  delay(400);
  }
  }
}
// Manually Exhaust FAN ON
void exhaustFanOn(int buttonState){
  if(buttonState == HIGH){
    digitalWrite(fan,HIGH);
    lcd.setCursor(0,0);
   lcd.print("Button State:");
    lcd.print(buttonState);
    lcd.setCursor(0,2);
    lcd.print("FAN ON");
    delay(10000);
    lcd.clear();
  }
}
```

GitHub Link : https://github.com/Raghul339