Gas Leakage Monitoring and Alerting System

Date	23nd November 2022
Team ID	PNT2022TMID44495
Project Name	Gas Leakage Monitoring and Alerting System
Maximum Mark	4 marks

Dhineshkumar.R , Sundhareswaran.R, Gunasekaran.S ,Saravanakkumaar.S,Mohamedbasidh.S
(Department of Electronics and Communication Engineering)

CONTENTS

Abstract :-

Leakage of any kind of gas has been a concern in recent years, whether it is in a residential setting, a business, a cafe, or a canteen. In this paper development of an IoT based gas wastage monitoring, leakage detecting and alerting system is proposed. This paper elaborates design such an intelligent system that will help save gas and smartly prevent accidents. The system needs to be integrated with the cooker. The technology includes ultrasonic sensors that determine if the cooker is being utilized for cooking purposes or not. If it is discovered that the cooker is not in use, the system uses an

automatic switching off mechanism to cut off the gas supply. The moment gas leakage will probably be recognized, users will be informed via SMS through GSM, and so that user can solve the issue as soon as possible. The system will monitor flame and fire through flame sensor. When a fire is detected, the buzzer begins to sound. Aside from that, the system also has a cloud storage capability. The usage of gas for each user each day may be tracked with the aid of this cloud storage solution. At the end of the day, this procedure will assist in detecting peruser natural gas usage. The system has been tested and it is able to monitor gas wastage, leakage and send a SMS to the user. The resulting performance indicated its effectiveness toward saving a significant portion of the wasted gas in domestic.

1. INTRODUCTION

1.1 Project Overview:

The internet of Things is a developing topic of technical, social, and economic significance. The usage of the gas brings great problems in the domestic as well as working places. The inflammable gas, which is excessively used in the work places (Industries). The leakage of the gas causes destructible impact to the lives and as well as to the heritage of the people. Most of the societies have fire safety mechanism. But it can use after the fire exists. As a result, a system for detecting and monitoring gas leaks is required. Through a flame sensor, the system will sense fire and flame. The buzzer begins to ring when a fire is detected. Tests have shown that the system can keep track of the wastage of gas and leaks and notify the user. The performance that was produced showed that it was successful in reducing the amount of gas that was wasted.

1.2 Purpose:

The sensor-enabled solution **helps prevent the high risk of gas explosions and affecting any casualties within and outside the**

premises. The gas sensors help detect the concentration of the gases present in the atmosphere to avoid hazardous consequenceslike fire breakouts.

2. LITERATURE SURVEY

2.1 Existing problem:

Gas leakage is nothing but the leak of any gaseous molecule from a pipeline, or cylinder etc in the industries. Gas Leakages in open or closed areas can prove to be dangerous .This can occur either purposefully or even unintendedly. As we are aware that these kinds of leaks are dangerous to our health, and when it becomes explosive it could cause great danger to the people, industry and the environment. Therefore, we have used IoT technology to make a Gas Leakage Detector for society which has Smart Alerting techniques involving sending a text message to the concerned authority and the ability to perform data analytics on sensor readings. Our main aim is to propose a gas leakage system for a society where each flat has gas leakage detector hardware. This will detect the harmful gases in the environment and alerting to society members through the alarm and sending notifications.

2.2 References:

1. @articlemalipatillpg, title=LPG Gas Measurement and Detection using GPS, author=Malipatil Somashekhar, Shilpa, Jayasudha, journal=International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN, p. 2278 - 3075
2. Crossref
3. Rameswari Siddharth, Keerthana Gayathri, Kavin Sanjaya

4.Smart gas assistant for a perfect kitchen

4. International Journal of Intellectual Advancements and Research in Engineering Computations (IJIAREC, issue 2, p. 2348 - 2079
5. Nagesh Anusha, Venkata Sai, Rupalin Srikanth, Nanda
6. LPG Leakage Detection and Booking System with Customer SMS Alerts. International Journal for Modern Trends in Science and Technology (IJMTST)
7. Posted: 2020-05
8.
9. @articlechafekar2018implementation, title=Implementation of automatic gas accident prevention system using arduino, author=Chafekar, Zamir Khan, Mohd Husain, Lakra,

Kuldeep Dhonde, SB, journal=International Journal of Computer Applications, volume=180, number=47, p. 5 - 7

10. Posted: 2018

11. Swapnil System, Sumit More, Prathamesh Borkar, Gailwad

12. @inproceedingsfatkiyah2019early, title=Early Detection of Leaks on Gas Cylinders, author=Fatkiyah, Persada, Andayati, D, booktitle=Journal of Physics: Conference Series, volume=1413, number=1, pages=012030, year=2019

13. Jebamalar Author=leavline, Singh, Antony Asir, Gnana

14. @articleleavline2017lpg, title=LPG Gas Leakage Detection and Alert System, volume 9

15. Naresh Naik, Nanda Siva Nagendra Reddy, Tharun Kishore, Kumar ReddyArduino Based LPG gas Monitoring and Automatic Cylinder booking with Alert System

16. IOSR Journal of Electronics and Communication Engineering, volume 11, issue 4, p. 6 - 12

17. Posted: 2016-08

18. Yusnita Huan Hui Yan, Rahayu Design and Development of Gas Leakage Monitoring System using Arduino and ZigBee

19. Proceeding of International Conference on Electrical Engineering, p. 20 - 21

20. International Journal of Research in Engineering, issue 1, p. 2581 - 579

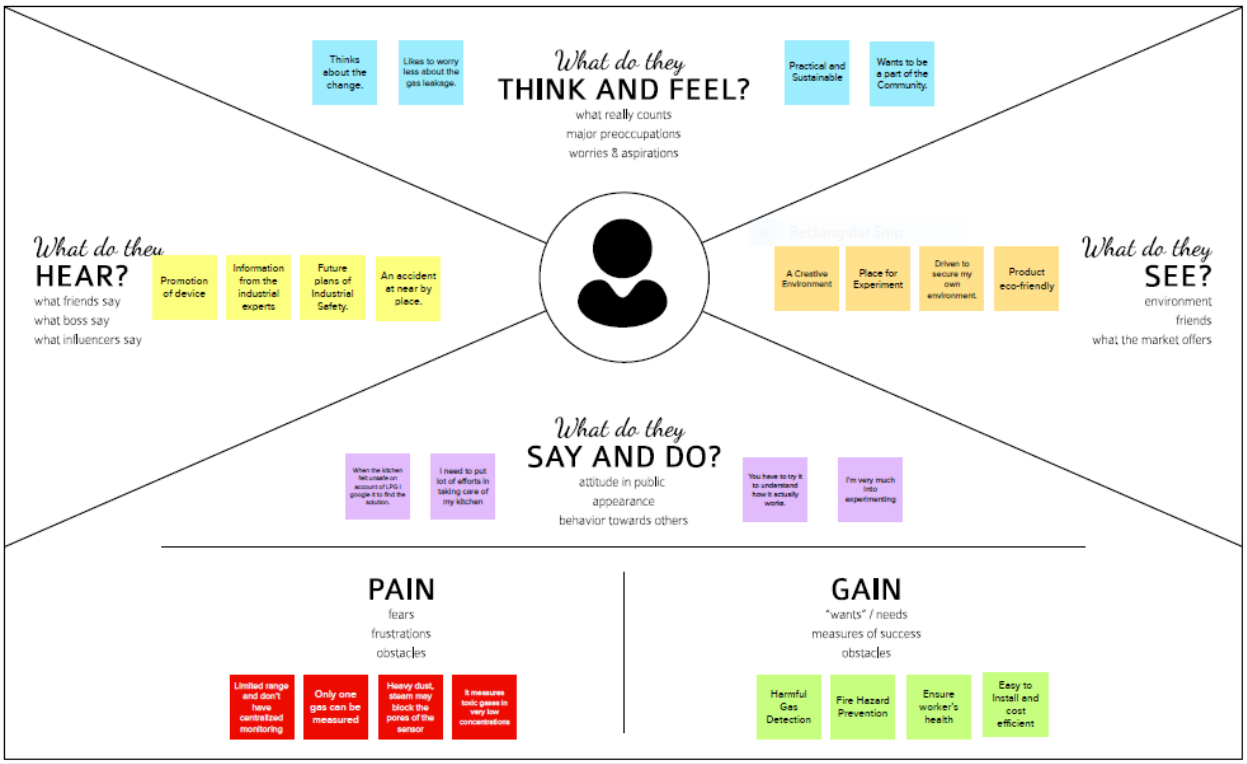
2.3 Problem Statement Definition :

Gas leakage is nothing but the leak of any gaseous molecule from a stove, or a pipeline, or cylinder etc. This can occur either purposefully or even unintendedly. As we are aware that these kinds of leaks are dangerous to our health, and when it becomes explosive it could cause great danger to the people, home, workplace, industry and the environment. Few of the major incidents that took place due to gas leakage include the Bhopal Disaster and the Vizag Gas leak. The Bhopal disaster is known to be the worst industrial accident ever. Approximately 45 tons of Methyl

Isocyanate was leaked from this insecticide plant. Methyl Isocyanate is an organic compound and a chemical that could come from the carbamate pesticides. This colorless, poisonous and flammable liquid is something that human beings have to be away from. Vizag Gas leak was a resultant of the escape of styrene that were unattended for a long period. This colorless oily liquid can spread in fumes. So, a detector must be made in such a way that could detect any kind of gas, fume, leak, smoke etc. However harmful and dangerous it can be, the detector could be attached with certain parameters that could help to prevent the issue.

3. IDEATION & PROPOSED SOLUTION

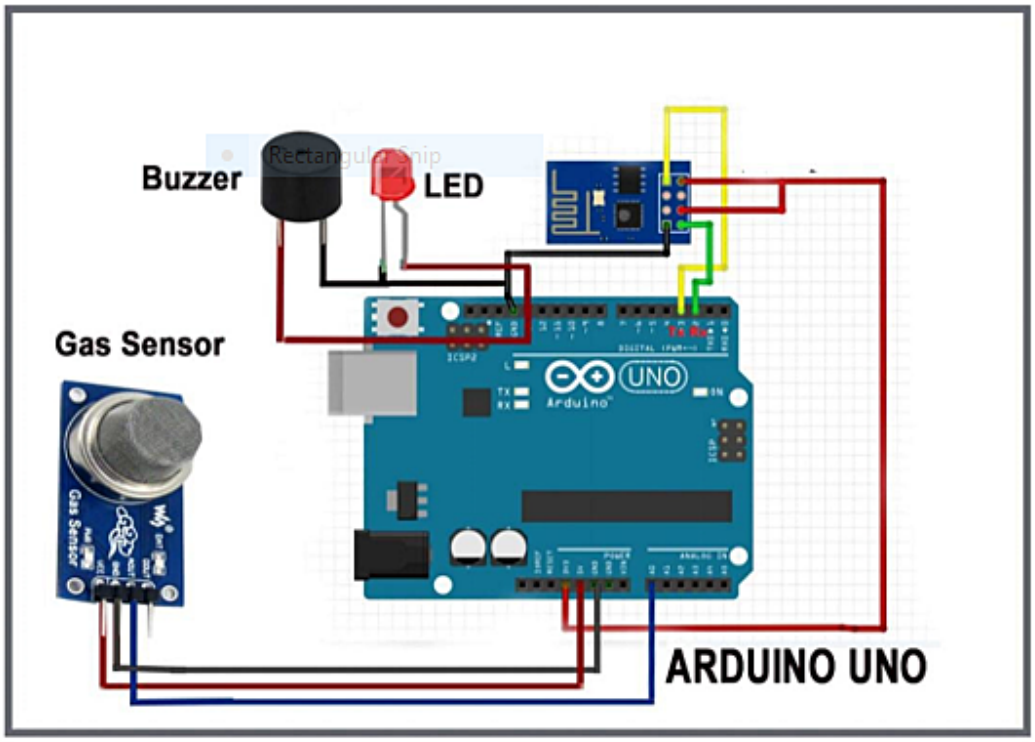
3.1 Empathy Map Canvas:



3.2 Ideation & Brainstorming:

The ideas are In case of higher gas leakage and fire accidents, a notification can be given to the fire station and hospital through software application.The level of gas in the industry can be informed through speakers periodically.When gas gets leaked, a notification can be passed to hospital.Sensor can be placed in the entrance for counting the workers who have been moved out in case of emergency.

In addition to alarm, a voice notes which alerts by saying the level of leakage can be designed.The alerting message can also be forwarded to the management of the industry.Sprinklers or extinguishers can be fixed which helps in case of inflammation by the leakage.Windows and gates can be opened automatically through sensors placed on that



I

used the IOT technology to make a Gas Leakage Detector for society which having Smart Alerting techniques involving sending text message to the concerned authority and an ability performing data analytics on sensor. This system will be able to detect the gas in environment using the gas sensors. This will prevent form the major harmful problem.

3.3 Proposed Solution:

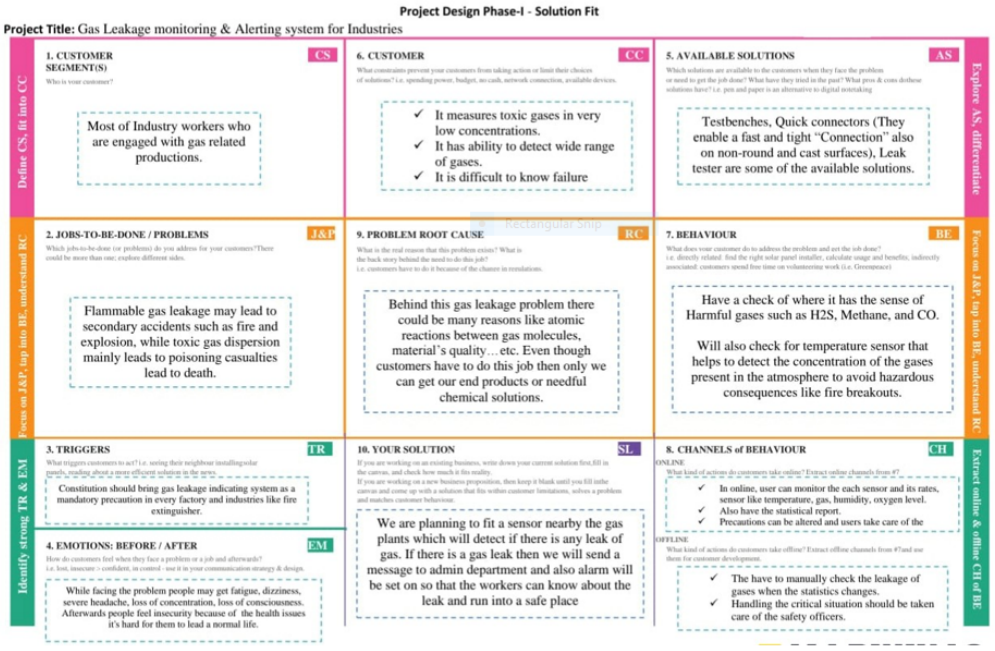
S.No.	Parameter	Description

1.	Problem Statement (Problem to be solved)	Workers who are engaged with a busy industries packed with gas either harmful or harmless needs a way to monitor their gas pipelines continuously and detect early if there is any leakage of gas in their surroundings so that they can work efficiently on major crises rather than worrying about monitoring or leakage of gas, this will indeed reduce the manpower of that industry and create a peaceful environment.
2.	Idea / Solution description	Workers who are engaged with a busy industries packed with gas either harmful or harmless needs a way to monitor their gas pipelines continuously and detect early if there is any leakage of gas in their surroundings so that they can work efficiently on major crises rather than worrying about monitoring or leakage of gas, this will indeed reduce the manpower of that industry and create a peaceful environment.

3.	Novelty / Uniqueness	Even though there are many existing solutions for this problem they failed to satisfy the needs of customer. Some of the solutions are only detecting some particular gases where some others failed to alert the main department and other solutions are with some delays. Our solution not only notify the industry person but also notify the fire fighters so that can take control over the situation and our solution will alert the workers even there is a small leak of gases.

4.	Social Impact / Customer Satisfaction	Our solution will be very helpful for the workers and the society which is associated or located nearby the industries. Our solution will prevent great disasters like Bhopal Gas Tragedy so that so many lives can be saved. Through this project the workers mental pressure will be reduced so that they can concentrate on other works or by relaxing them.
5.	Business Model (Revenue Model)	The main target of our solution is Industries so we have planned to visit industries and explain them about the benefits of our products. So that they can aware of the importance of this solution and use it.
6.	Scalability of the Solution	Our solution can be integrated for further future use because the solution we have provided will be lay on the basic or initial stage of any upgraded version.

3.4 Problem Solution fit:



4. REQUIREMENT ANALYSIS

4.1 Functional requiremen:

Functional Requirements

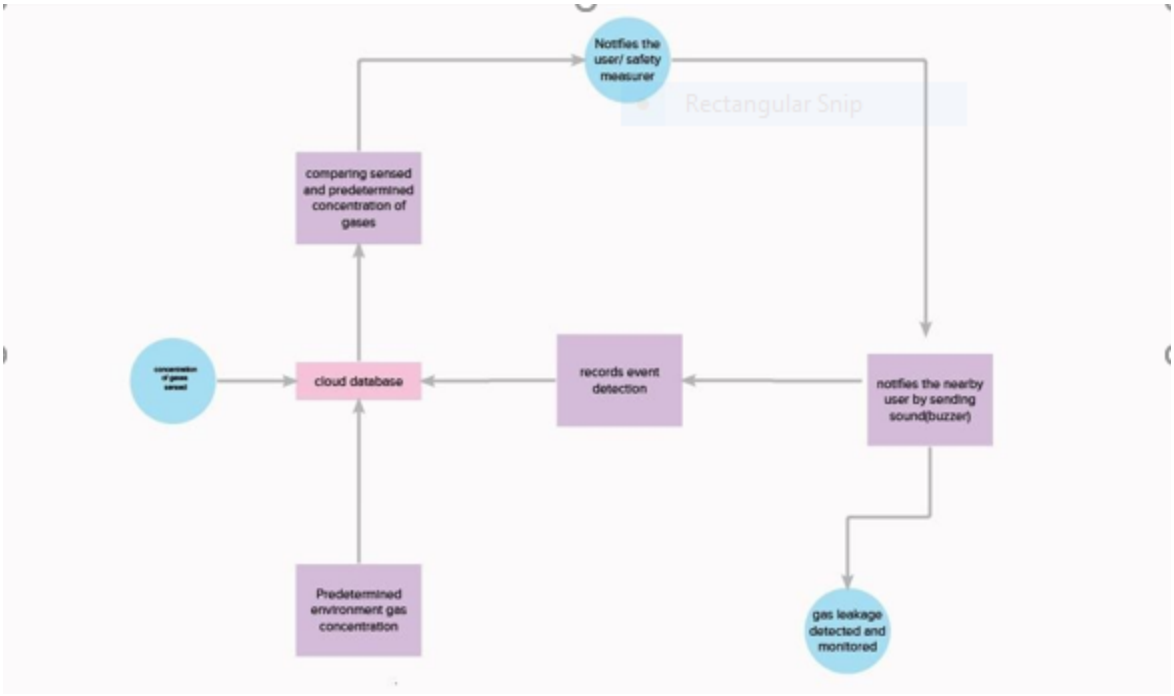
Business Requirements	User Requirements	Product Requirements
The said system can be deployed in homes, hotels, factory units, LPG cylinder storage areas, and so on. The main advantage of this IoT and Arduino-based application is that it can determine the leakage and send the data over to a site. It can 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 fulfill.

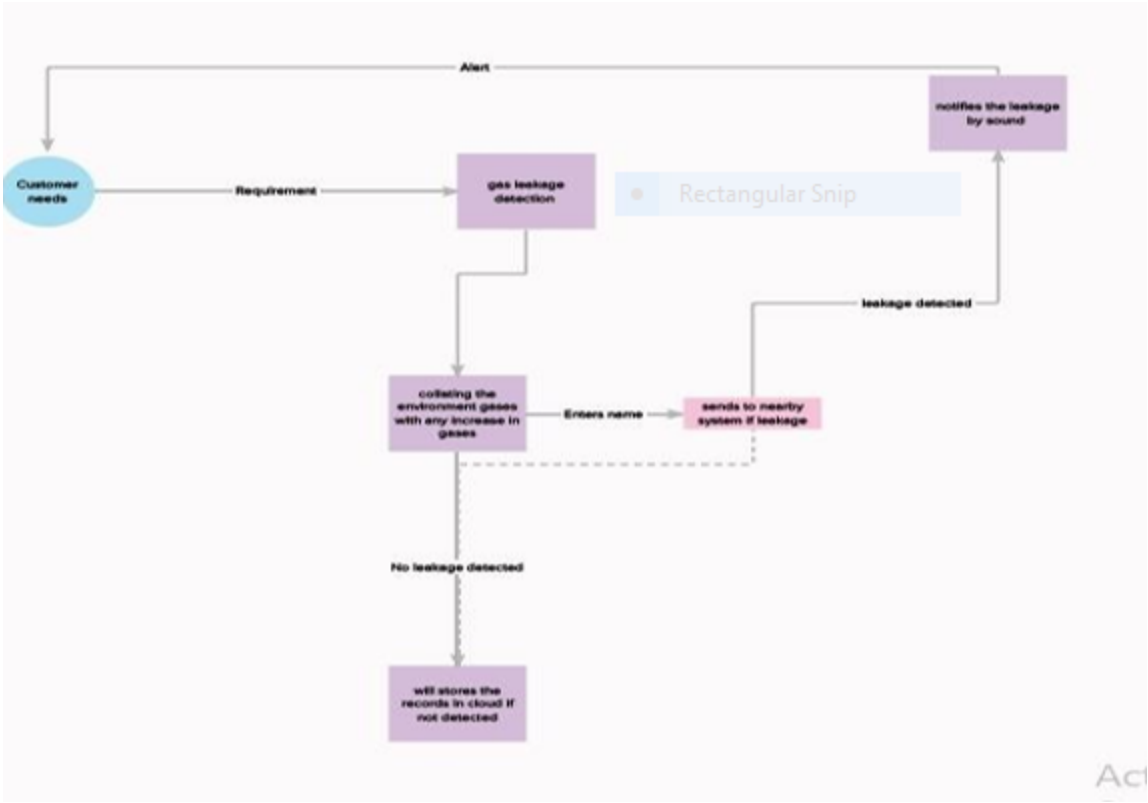
4.2 Non-Functional requirements:

FR No.	Non-Functional Requirement	Description
NFR- 1	**Usability**	Easy user interface with alerting notifications and location of the defect gas cylinder.
NFR- 2	**Security**	1. Secure Cloud database is used. 2. Notify only the registered and verified users. 3. Multiple deployments across the potential sources can help industries to avoid any industrial accident and protect workplace safely.
NFR- 3	**Reliability**	1. Gas exposure will measured with ± 25% of the true concentration of the target analyte with 95% certainty. 2. Robust device that can withstand harsh industrial conditions and provide real-time gas leakage detection.
NFR- 4	**Performance**	1. Accurate data monitoring system enables periodic analysis of the air quality.

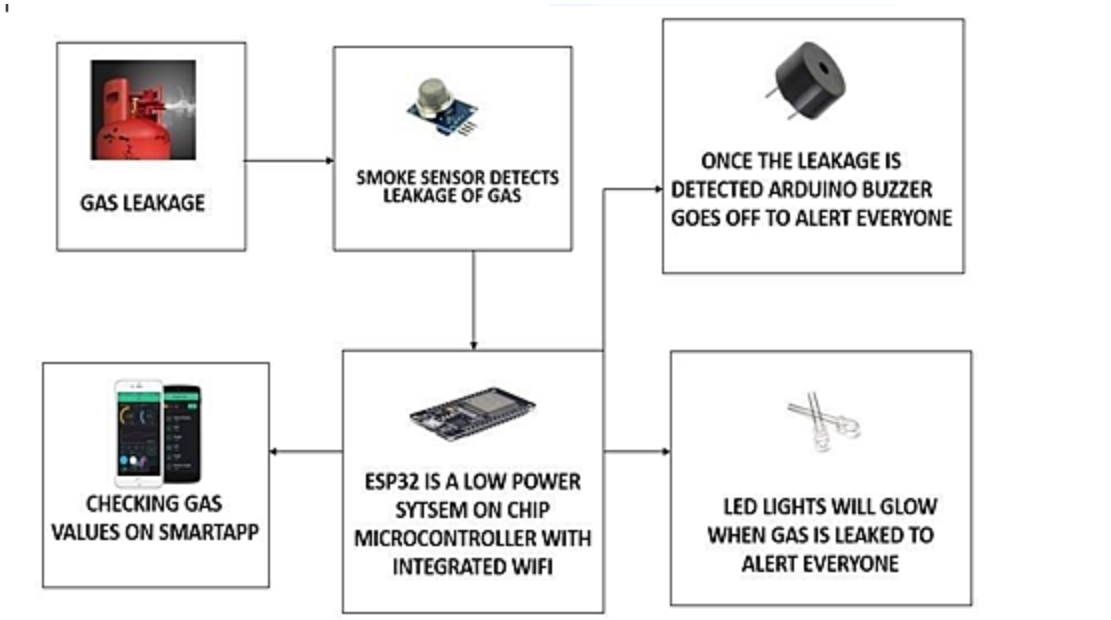
5. PROJECT DESIGN

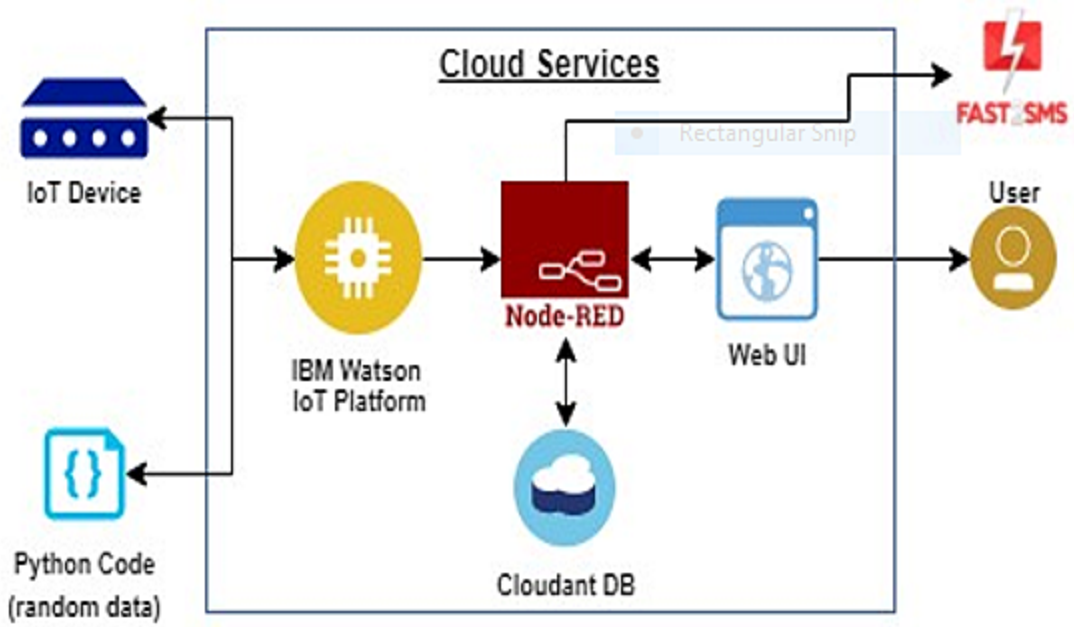
5.1 Data Flow Diagrams:





5.2 Solution & Technical Architecture:





6.PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation:

- SPRINT PLAN
- ANALYZE THE PROBLEM
- PREPARE an ABSTRACT, PROBLEM STATEMENT
- LIST A REQUIRED OBJECT NEEDED
- CREATE A PROGRAM CODE AND RUN IT
- MAKE A PROTOTYPE TO IMPLEMENT
- TEST WITH THE CREATED CODE AND CHECK THE DESIGNED PROTOTYPE

6.2 Sprint Delivery Schedule:

Sprint	Functional Requirement (Epic)	User Story	User Story / Task	Story Point	Priority
Sprint-1	Create	US-1	Create the IBM Cloud services which are being used in this project.	5	High

Sprint-1	Configure	US-2	Configure the IBM Cloud services which are being used in completing this project.	1	Medium
Sprint-1	Create	US-3	IBM Watson IoT platform acts as the mediator to connect the web application to IoT devices, so create the IBM Watson IoT platform.	1	Medium
Sprint-1	Configure	US-4	Configure the IBM Watson IoT which are being used to display the output.	13	High
Sprint-2	Create	US-1	In order to connect the IoT device to the IBM cloud, create a device in the IBM Watson IoT platform and get the device credentials.	13	High

Sprint-2	Configure	US-2	Configure a device in the IBM Watson IoT platform and get the device credentials.	3	Medium
Sprint-2	Create	US-3	Create a Node-RED service.	3	High
Sprint-2	Configure	US-4	Configure the connection security and create API keys that are used in the Node- RED service for accessing the IBM IoT Platform.	1	Medium

Sprint-3	Develop	US-1	Develop a python script to publish random sensor data such as temperature, Flame level and Gas level to the IBM IoTplatform	1 3	High

Sprint-3	Configure	US-2	After developing python code and commands just run the code	1	Medium
Sprint-3	Print	US-3	Print the statements which represent the control of the devices.	1	Low
Sprint-3	Publish	US-4	Publish Data to The IBM Cloud	5	High
Sprint-4	Create	US-1	Create Web UI in Node- Red	5	High
Sprint-4	Configure	US-2	Configure the Node-RED flow to receive data from the IBMIoT platform	5	High
Sprint-4	Publish	US-4	Publish the received data in webapplication	5	High

7.CODING & SOLUTIONING

```
 #     Importing     Required
modules import time import
sys
import wiotp.sdk.device# IBM IoT Watson Platform
Module import ibmiotf.device import tkinter as tk #
Python GUI Package from tkinter import ttk #
Python GUI import time from threading import
Thread


organization = "9s9m43" # Organization ID
deviceType = "NodeMCU" # Device
type deviceId = "gasleakage" # Device
ID authMethod = "token"       #
Authentication Method authToken =
"1234589123" #Replace the authtoken


# Tkinter root window root = tk.Tk()
root.geometry('350x300') # Set size of root
window root.resizable(False, False) # root
window non- resizable root.title('Gas
Leakage Monitoring And
Alerting System for Industries
(PNT2022TMID00378)')


# Layout Configurations root.columnconfigure(0,
weight=1) root.columnconfigure(1, weight=3)

current_gas = tk.DoubleVar()

def get_current_gas(): # function returns current gas level value
return '{: .2f}'.format(current_gas.get())
```

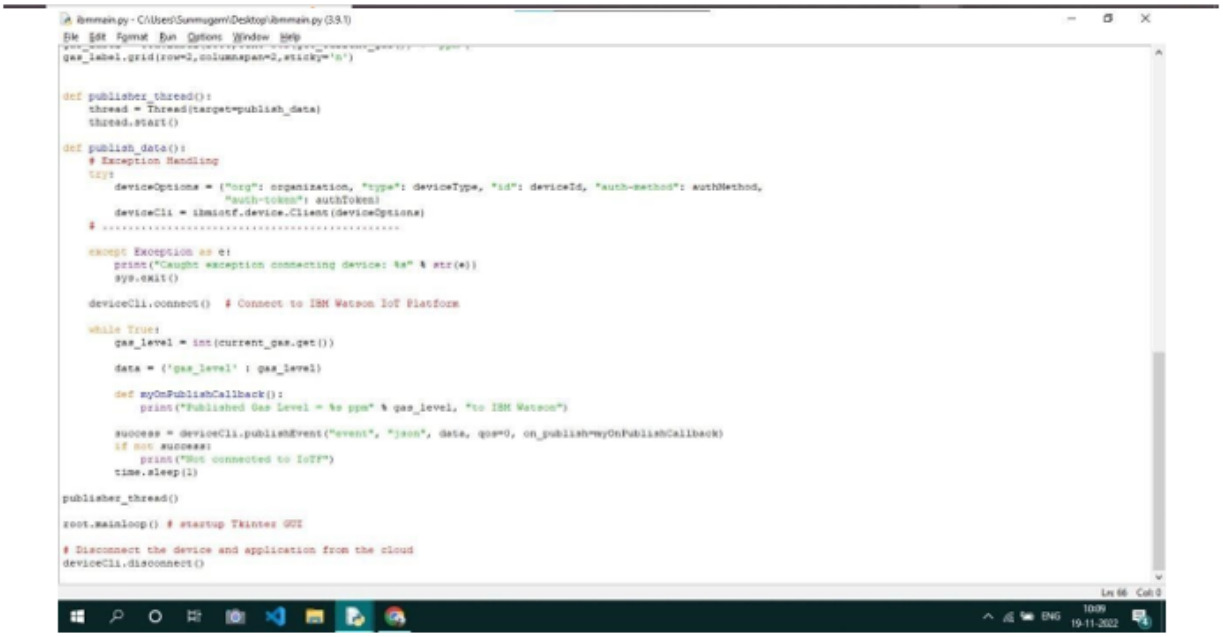
```
def slider_changed(event): # Event Handler for changes in sliders      print('------
--')
     print('Gas Level: {: .2f}'.format(current_gas.get()))      print('------
--')
gas_label.configure(text=str(get_current_gas()) +" ppm") # Displays current gas level as
                                                                                label
content


# Tkinter Labels
# label for the gas level slider slider_gas_label =
ttk.Label(root,text='Set Gas Level:')
slider_gas_label.grid(column=0,row=0,sticky='w')


# Gas Level slider slider_gas
=                                    ttk.Scale(root,from_=0,to=3000,orient='horizontal',
command=slider_changed,variable=current_
gas)
slider_gas.grid(column=1,row=0,sticky='we')


# current gas level label
current_gas_label = ttk.Label(root,text='Current Gas Level:')
current_gas_label.grid(row=1,columnspan=2,sticky='n',ipadx=10,ipady=10)


# Gas level label (value gets displayed here)
gas_label                                    =
ttk.Label(root,text=str(get_current_gas()) +" ppm")
gas_label.grid(row=2,columnspan=2,sticky='n')

 def publisher_thread():      thread =
Thread(target=publish_data)
   thread.start()

 def publish_data():     # Exception Handling      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()

   deviceCli.connect()  #  Connect  to  IBM  Watson  IoT

   Platform

   while   True:                    gas_level   =
int(current_gas.get())

       data = {'gas_level' : gas_level}
        def myOnPublishCallback():              print("Published Gas Level = %s
ppm" % gas_level, "to IBM Watson")

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

root.mainloop() # startup Tkinter GUI

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

code:

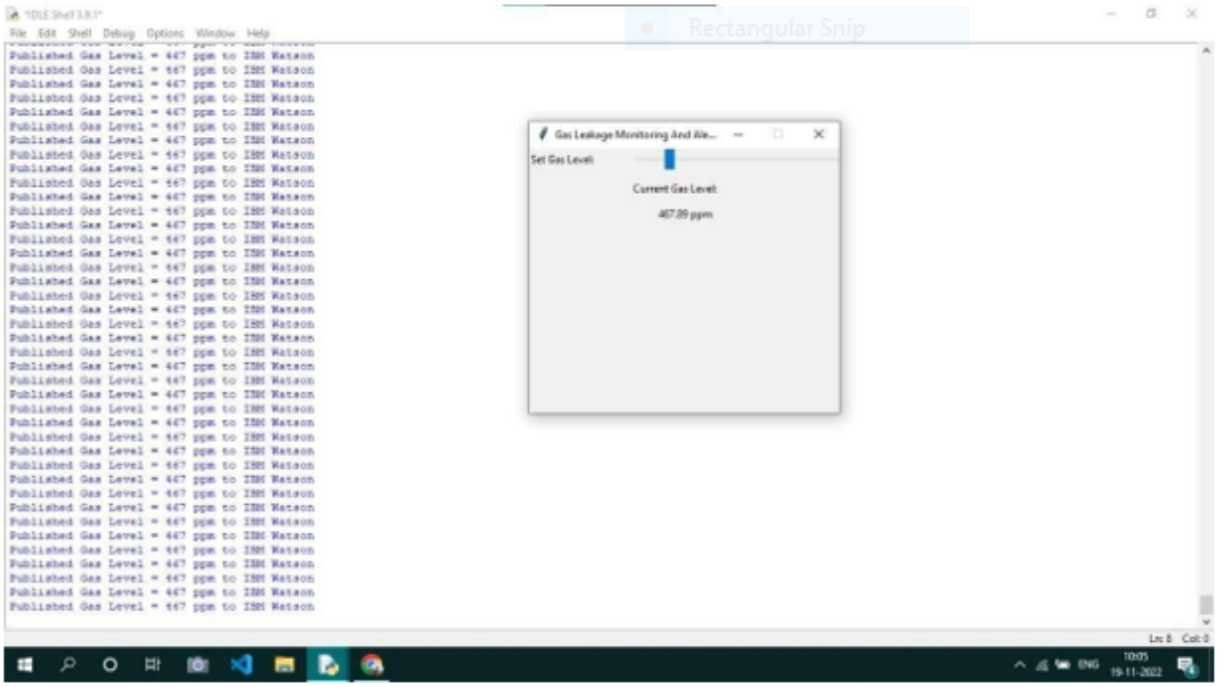
CODE:

```
# Importing Required modules import time
import sys
import wiotp.sdk.device# IBM IoT Watson Platform Module import ibmiotf.device
import tkinter as tk # Python GUI Package from tkinter import ttk # Python GUI

import time
from threading import Thread


organization = "nd8val" # Organization ID deviceType = "GASLEAKAGE"     # Device type deviceId = "10042002"     # Device ID
authMethod = "token"    # Authentication Method authToken = "10042002" #Replace the authtoken


# Tkinter root window root = tk.Tk()
root.geometry('350x300') # Set size of root window root.resizable(False, False) # root window non-resizable root.title('Gas Leakage Mc


# Layout Configurations root.columnconfigure(0, weight=1) root.columnconfigure(1, weight=3)


current_gas = tk.DoubleVar()

def get_current_gas(): # function returns current gas level value return '{: .2f}'.format(current_gas.get())

def slider_changed(event): # Event Handler for changes in sliders print('       ')
print('Gas Level: {: .2f}'.format(current_gas.get())) print('    ')
gas_label.configure(text=str(get_current_gas()) +" ppm") # Displays current gas level as label content



# Tkinter Labels

# label for the gas level slider
slider_gas_label = ttk.Label(root,text='Set Gas Level:') slider_gas_label.grid(column=0,row=0,sticky='w')


# Gas level slider
```

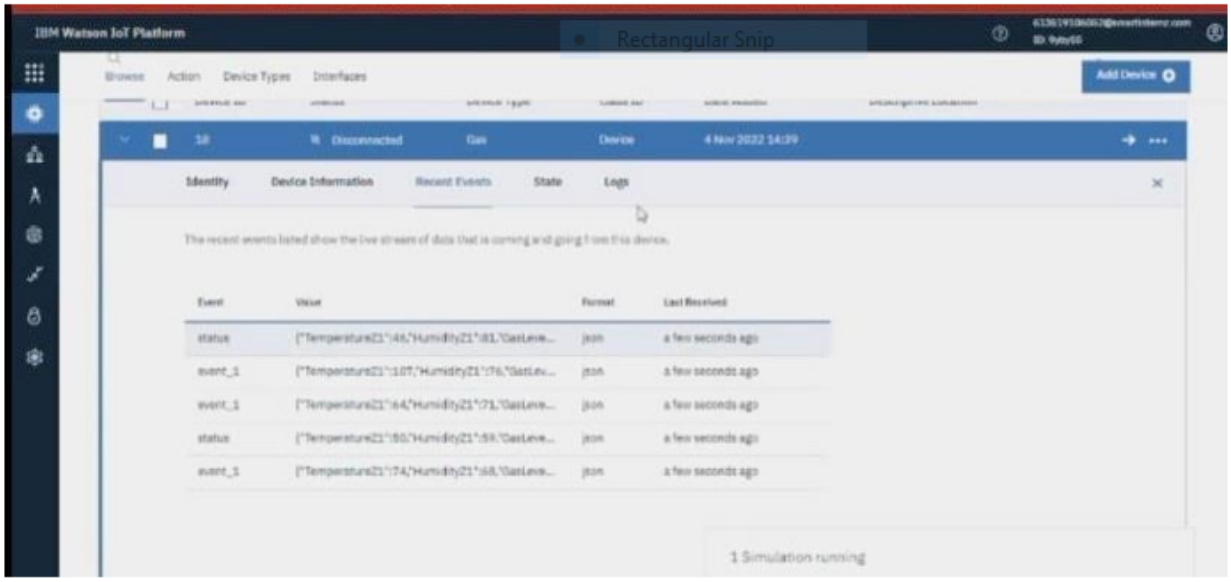


```
gas_label.grid(row=2,columnspan=2,sticky='n')


def publisher_thread():
    thread = Thread(target=publish_data)
    thread.start()

def publish_data():
    # Exception Handling
    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()

    deviceCli.connect()  # Connect to IBM Watson IoT Platform

    while True:
        gas_level = int(current_gas.get())

        data = {'gas_level' : gas_level}

        def myOnPublishCallback():
            print("Published Gas Level = %s ppm" % gas_level, "to IBM Watson")

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

publisher_thread()

root.mainloop() # startup Tkinter GUI

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

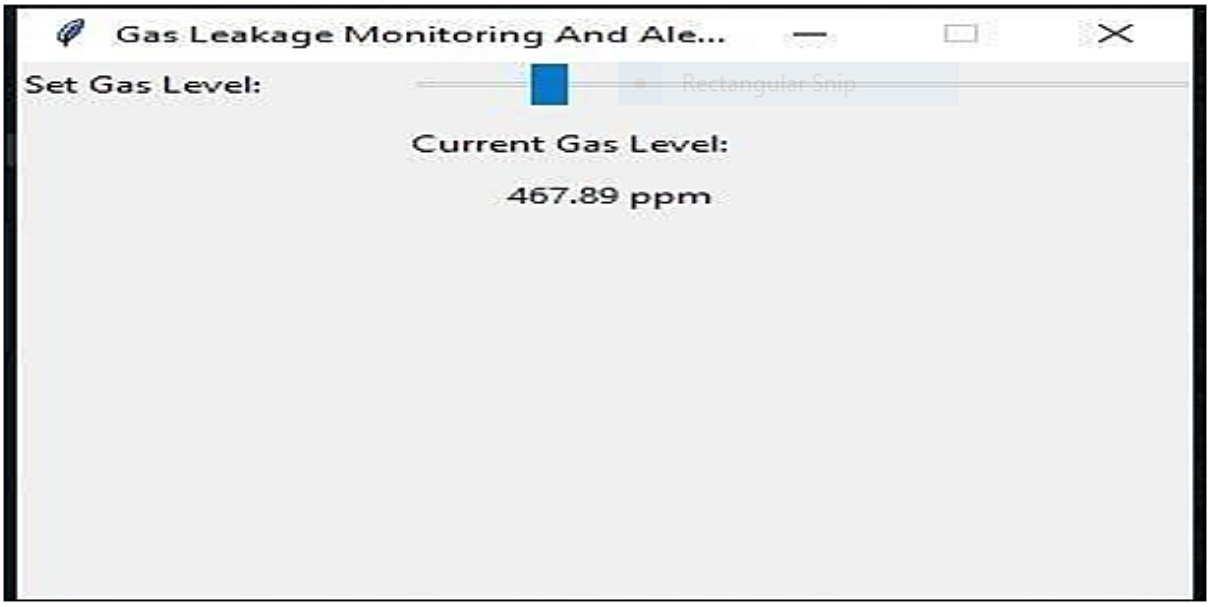
OUTPUT:

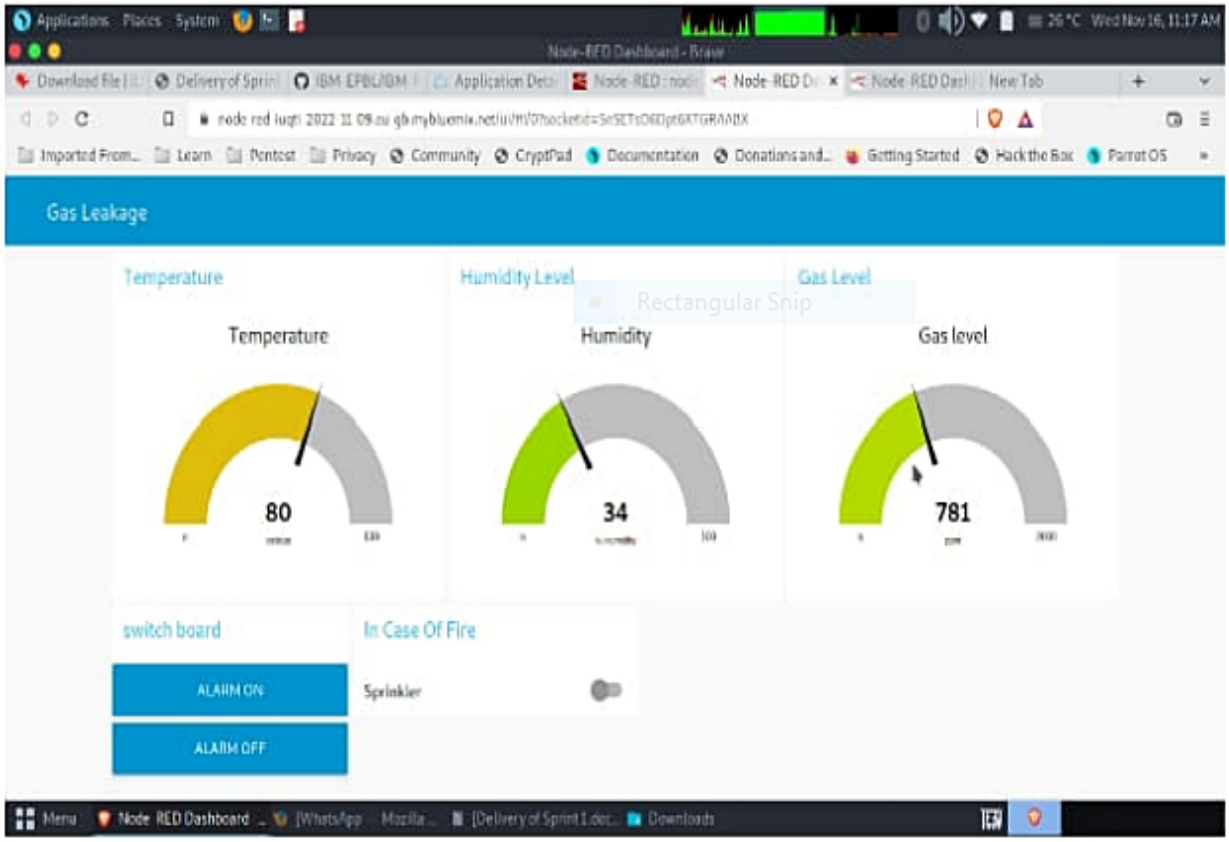


8. TESTING




Zone 1
Zone 1
Humidity
GasLevel
Pressure


Gas Leakage Monitoring And Ale...
Set Gas Level:
Current Gas Level:
467.89 ppm



UAT TESTING :

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

Test Case Analysis :

Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

8.Result:

The system can be taken as a small attempt in connecting the existing primary gas detection methods to a mobile platform integrated with IoT platforms. The gases are sensed in an area of 1m radius of the rover and the sensor output data are continuously transferred to the local server. The accuracy of sensors is not up to the mark thus stray gases are also detected which creates an amount of error in the outputs of the sensors, especially in case of methane. Further the availability and storage of toxic gases like hydrogen sulphide also creates problems for testing the assembled hardware. As the system operates outside the pipeline, the complication of system maintenance and material selection of the system in case of corrosive gases is reduced. Thus, the system at this stage can only be use data primary indicator of leakage inside a plant.

9.Advantages/Disadvantages

9.1 Advantages:

1. Get real-time alerts about the gaseous presence in the atmosphere.
2. Prevent fire hazards and explosions.
3. Supervise gas concentration levels.
4. Ensure worker’s health.
5. Real-time updates about leakages.

6. Cost-effective installation.
7. Data analytics for improved decisions.

8. Measure oxygen level accuracy.
9. Get immediate gas leak alerts.

9.2 Disadvantages:

1. It requires air or oxygen to work.
2. It gets reacted due to heating of wire.
3. It can be poisoned by lead, chlorine and silicon

10.CONCLUSION

This gas leak detector system contains two features, this includes the SMS Gateway feature for only sending warning information regarding the gas leak to user, and the alarm for the warning alert. There is some improvement which can be applied for the future work, such as regarding the SMS Gateway, it need to enhance with feature such as notifying the user whenever the remaining credit balance is insufficient. Another thing which can be enhanced is regarding the sensor, the sensors in this module do not include somewhat notification for notifying the user whenever the sensor not working properly or not connected to the micro-controller for some cases, therefore, it is recommended to add this kind of features in the future work for better refinement.

11.FUTURE SCOPE

We propose to build the system using an MQ6 gas detection sensor and interface it with an Aurdino Uno microcontroller along with an LCD Display. This system uses the gas sensor to detect any gas leakages. The gas sensor sends out a signal to the microcontroller as soon as it encounters a gas leakage. The microcontroller processes this signal and a message is displayed on the LCD to alert the user.

12. APPENDIX

Source Code

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


#Provide your IBM Watson Device Credentials organization =
"pi0ywk" deviceType = "Gas_Geakage_Detector" deviceId =
"Udayakpr007" authMethod = "token" authToken =
"9952356299"

# Initialize GPIO


def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status == "alarmon":
        print ("Alarm is on please all Evacuate Fans On")
    elif status == "alarmoff":
        print ("Alarm is off and Fans Off")
    elif status == "sprinkleron":
        print ("Sprinkler is On Evacuate Faster")
    elif status == "sprinkleroff":
        print("Sprinkler is Off")
    else:
        print("Please send proper command")
    #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 random function

      temp=random.randint(0,120)
      Humid=random.randint(0,100)
      gas=random.randint(0,1500)
      data={'temp':temp,'Humid':Humid,'gas':gas}
      #print data
      def myOnPublishCallback():
         print (" Published Temperature = %s C" % temp, "Humidity = %s
%%" % Humid, "Gas_Level = %s ppm" %gas, "to IBM Watson")

      success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
      if not success:
         print("\n Not connected to IoTF")
      if temp>60 :
         print("\n Fire Detected due to gas Leak ! Alarm ON! Sprinkler
ON! Call The Fire Police \n")
      elif gas>350:
         print("\n Gas is Leaking \n")
```

```
        time.sleep(10)

        deviceCli.commandCallback = myCommandCallback

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

GitHub & Project Demo Link:

https://github.com/IBM-EPBL/IBM-Project-49938-1660884893

https://github.com/IBM-EPBL/IBM-Project-49938-1660884893

https://github.com/IBM-EPBL/IBM-Project-49938-1660884893#:~:text=Explore-,IBM%2DEPBL,IBM%2DProject%2D49938%2D1660884893,-Public