HAZARDOUS AREA MONITORING FOR INDUSTRIAL PLANT BY IOT

Team id:PNT2022TMID39386

1.INTRODUCTION

1.1 PROJECT OVERVIEW

The devices which were invented with the new emerging technologies related to the small, low power and viable sensors consists of a power monitor and remote communications capabilities. The server receives the natural phenomenon data i.e. Temperature, light, and pressure which is collected by the sensors monitoring and control of the physical environment from remote locations is performed by battery powered nodes. Wireless multisensory networks have met their applications ion medical, military, industrial, agricultural and environmental monitoring. Current, voltage, temperature and water level are the traceable parameters. An android app is developed for the control of industrial automation automatically using cloud. The manager of the android application has his/her own unique and specific username and password by which the app can be controlled manually. Wireless communication is the best technology and it is widely used in industry for automation purposes. Now-a-days data acquisition system is widely popular in the industry for remote monitoring and controlling of system status. The aim of the project is to design a system that uses various physical parameters such as light, fire, humidity of an industry that can be monitored through a PC using the Arduino. This parameter contains various data from different sensors. It covers the range and performance of the system.

1.2 PURPOSE

Internet of Things(IoT) has made great revolutions in industries.IOT is a technology that helps us with controlling of the physical devices over the internet. In this paper, we have designed a paradigm which can help us control and monitor all the industrial parameters from all over the world. For monitoring of the parameters, we have used different sensors such as fire, temperature, gas, humidity, voltage and current sensor. Also for alerting the workers, we have used a voice module which gives them voice alerts.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

LITERATURE 1: Hazard Monitoring System in Industry Using IOT

ABSTRACT:

In spite of security and automation in plants, industrial environments quiet critical for machines and humans. Industrial surroundings are crucial for both machines and people, despite security and automation in facilities. With a safety in industrial condition, this paper is contracted. A method has been created to identify risky circumstances, such as breakdown, which is the primary cause of leakage current in substations, and assist in preventing them. In order to validate the method employed in this method, safety scenes were also constructed. This method is intended to safeguard a person. This method reveals the negative effects of thermally hazardous environments and people. Automation systems that monitor the system and flag any errors reduce the amount of work that must be done by humans. The GSM concept is employed in this study. It consists of a network of physical items or things with electronics, software, sensors, Here automation system will be used in industry for monitoring various parameters such as temperature, humidity, gas and fire.

LITERATURE 2: A Study On Computer Based Monitoring SystemFor Hazardous Area Safety Measurement UsingVirtual Instrumentation

ABSTRACT:

Today there is a great challenge in the development of industrial hazardous safety monitoring for the application of gas leaks, fire, smoke, radiation etc. In all related fields of investigation, a key matter is the need flexible and practical virtual instruments, a way to easily expose the multi- sensors to the hazardous levels in risk concentration. The implementation of wireless sensor network provides an alternative solution by deploying a larger number of disposable sensor nodes. The Sensor data may consist of industrial environmental parameters like critical temperature, gas leakage, radiation, fire, smoke and the dynamic variations of these physical quantities. This software platform is in the terms of virtual instruments developed under Lab VIEW programming environment and integrated with computer controlled system.

LITERATURE 3: An overview of industrial alarm systems: Main causes for alarm overloading, research status, and open problems

ABSTRACT:

Alarm systems are crucial to the secure and effective running of contemporary industrial facilities. However, the majority of industrial alarm systems currently in use perform poorly, most notably having an excessive number of alarms that cannot be managed by operators in control rooms. Such alarm overloading is particularly damaging to the crucial role served by alarm systems. An overview of industrial alarm systems is given in this document. Alarm overloading is attributed to four primary causes: chattering alarms brought on by noise and disturbance; poorly configured alarm variables; isolation of the alarm design from related variables; and abnormality propagation due to physical connections. As supplementary evidence, industrial examples from a sizable thermal power plant are given. By concentrating on existing research, the state of industrial alarm system research is summarized.However, industrial alarm systems are generally suffering from alarm overloading. This paper provides an overview of industrial alarm systems, by proposing main causes for alarm overloading, summarizing current research status and formulating open problems. In presenting this overview, we hope to attract direct attentions from more researchers and engineers into the study of industrial alarm systems.

LITERATURE 4: Sleep scheduling in industrial wireless sensor networks for toxic gas monitoring

ABSTRACT:

Early detection and resolution of issues leads to time and money savings as well as increased business productivity. Systems based on WSNs are not uniform or incompatible. Between regions and processes, there is a lack of coordinated communication and

transparency. SCADA systems, on the other hand, are pricy, rigid, and not scalable, and they deliver data slowly. The oil and gas sectors are suggested a revolutionary IoT-based architecture in this study to make data collection from connected objects as easy, secure, robust, reliable, and rapid as possible. The application of this design to any of the three categories of operations—upstream, midstream, and downstream—is also suggested. A variety of Internet of Things (IoT)-based smart objects (devices) and cloud-based technologies can be used to do this. Our proposed IoT architecture supports the functional and business requirements of upstream, midstream and downstream oil and gas value chain of geologists, drilling contractors, operators, and other oil field services.

LITERATURE 5: Wireless sensor networks for industrial environments ABSTRACT:

A difficult research field is the deployment of a large population of sensors for complex sensing and control in industrial and commercial infrastructures. Today, there is a lot of interest in the usage of wireless sensor networks (WSN) for industrial applications. Due to the unpredictability of changes in temperature, pressure, humidity, the presence of large equipment, etc., the industrial environment for wireless sensor networks is harsher than that of office networks. We offer an overview of wireless sensor networks for industrial applications in this study.

2.2 REFERENCES

[1] Li Da Zu" Internet of Things in Industries: A Survey" IEEE Transactions on Industrial Informatics, vol. 10, no. 4, November 2014.

[2]Research Scholar, Research and Development Centre, Bharathiar University, Coimbatore -641046, India

[3]Emiliano Sisinni, Abusayeed Saifullah, Song Han, Ulf Jennehag, Mikael Gidlund

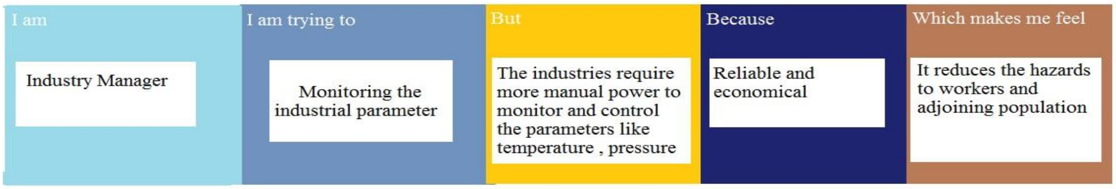
IEEE transactions on industrial informatics 14 (11), 4724-4734, 2018

[4]Mithun Mukherjee, Lei Shu, Likun Hu, Gerhard P Hancke, Chunsheng Zhu

IEEE Wireless Communications 24 (4), 106-112, 2017

[5] Gang Zhao "Wireless Sensor Networks for Industrial Process Monitoring and Control: A Survey", Network Protocols and Algorithms, 2011, Vol. 3, No.1, pp.4663

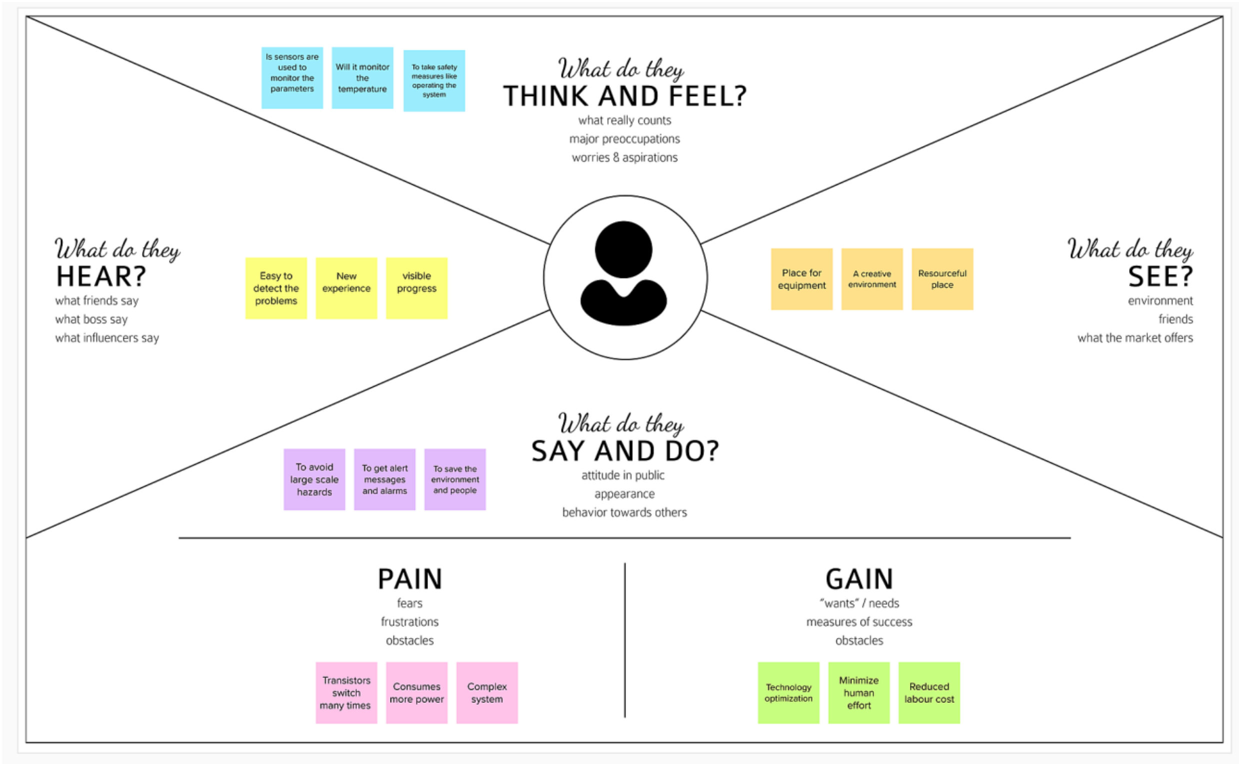
2.3 PROBLEM STATEMENT DEFINITION



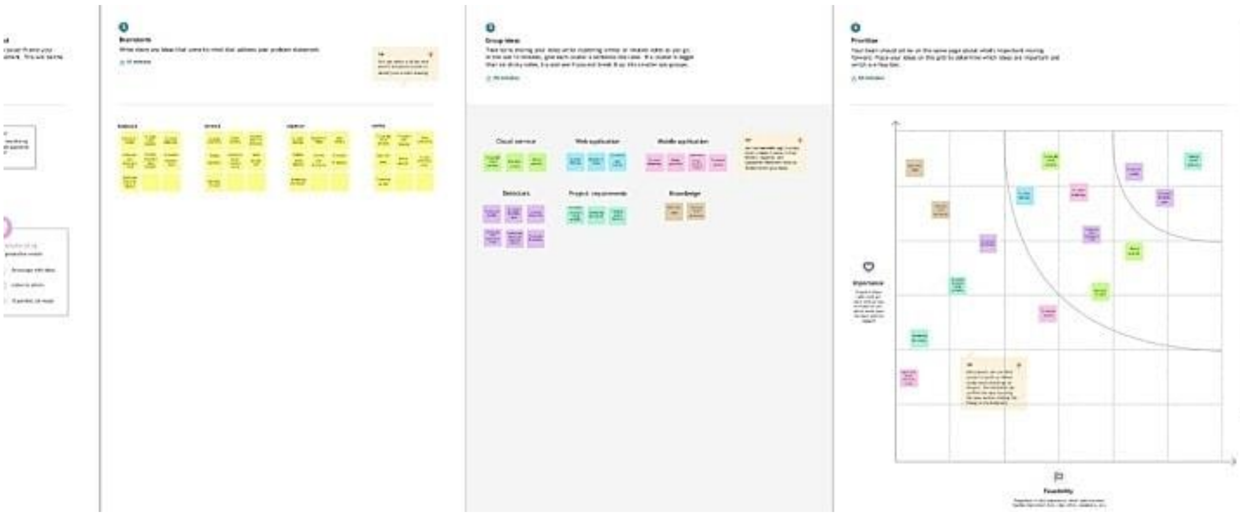
Problem Statement P S)	**I am(Customer)**	**I amtryingto**	**But**	**Because**	**Which makes mefeel**
PS-1	Industry Manager	Monitoring the industrial paramete	The industries require more manual power to monitor and control the parameters like temperature , pressure etc.,	Reliable and economical	It reduces the hazards to workers and adjoining population
PS-2	Industry Manager	Implementation of real WSN to measure industrial parameter	The parameters will not monitored and controlled properly at the time of emergency ,it leads to harmful situation.	Simple and low cost technology	The capability of sensing accurately with increased flexibility

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION & BRAINSTROMING



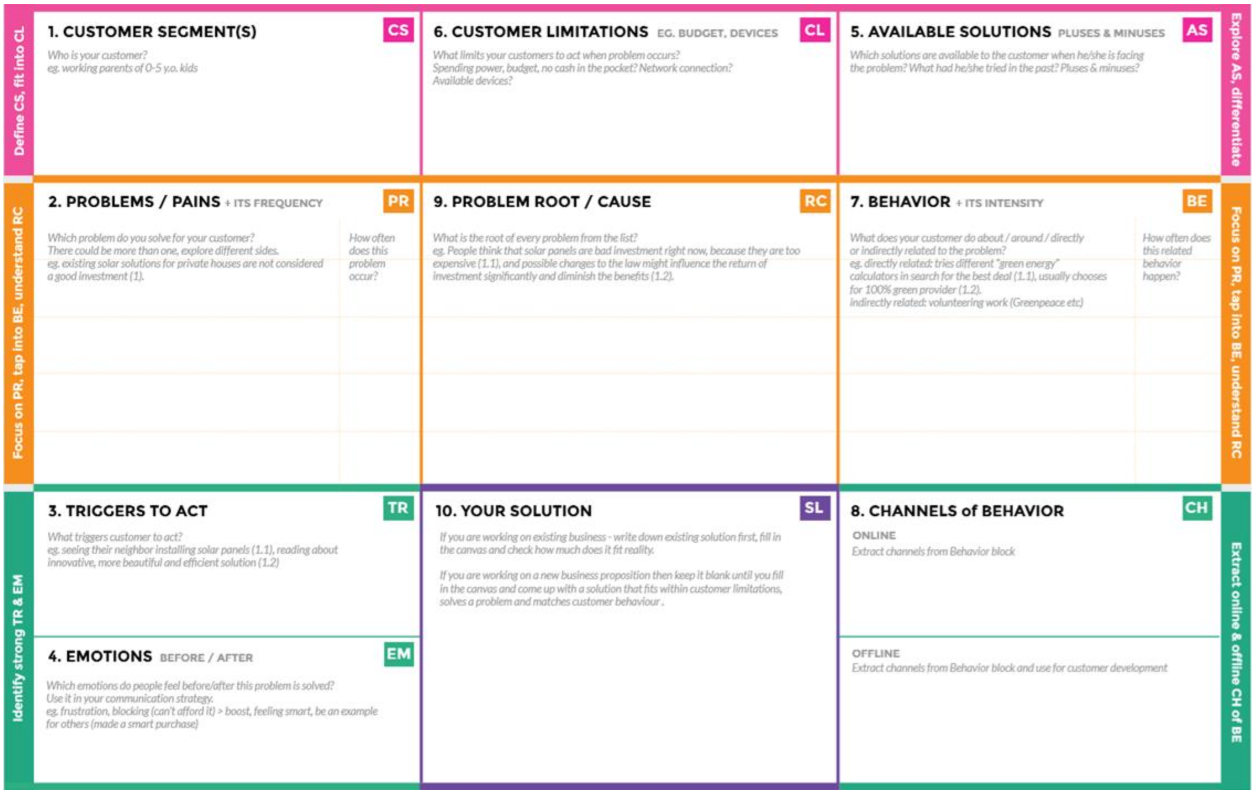
3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Hazardous Area Monitoring for Industrial Plant powered by IOT
2.	Idea / Solution description	Hazardous Area Monitoring for Industrial Plant powered by IOT is a project report that focuses on the necessity of the monitoring of hazardous areas in industrial plants.Industrial plants are the ones that contain both hazardous and non- hazardous areas. The monitoring of the hazardous areas in industrial plants important from time to time. If the damage that occurs in hazardous areas can result in the loss of property or lives. So monitoring of such areas can help in easy monitoring of the hazardous areas. There can be smart devices integrated at the hazardous areas that can help in detecting any fishy things that can occur in the particular area.

3.	Novelty / Uniqueness	* A hazardous area is any area with an atmosphere containing, or potentially containing, gases, vapor or dust which are flammable or explosive. These areas are rigorously analyzed with condition monitoring when installing equipment to minimize the risk to individuals and assets. It is crucial that equipment operating in these conditions are effectively monitored preempt any issues before they occur. Unlike most industries, these issues not only result in downtime, but present significant safety risk. * Condition monitoring is integral in industrial operations to avoid downtime, implement maintenance and to reduce the risk of failure. Remote condition monitoring has previously been limited hazardous areas due to the lack of cost-effective and easy to install solutions – and the often-challenging environments , which this equipment exists. For example, equipment used in offshore operations cannot be monitored frequently or easily.
4.	Social Impact / Customer Satisfaction	1. Real time plant monitoring 2. Reduced risk of disasters 3. Automated detection 4. Excellent customer experience
5.	Business Model (Revenue Model)	Raspberry -Pi 3 Temperature Sensor DS18B20 Gas Sensor - MQ 5/9 Breadboard Raspbian OS (Running on Rpi-3) Simple push API Thing speak Cloud Platform

6.	Scalability of the Solution	This system can be deployed in many industrial areas like mining, underground factories, metal refineries, automatic welding factories and even heavy parts production lines. It will help to provide a safe and efficient working environment wich aroos, while also opening new paths to improve the safety parameters these places.

3.4 PROBLEM SOLUTION FIT



4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Data Gathering	The smart beacon must be able to detect and the temperature of a particular area in real.
FR-2	Location Detection	The smart beacon must be able to detect when a wearable devicehas entered an area near it.
FR-3	Beacon Data Syncing	The smart beacon must be able to share its stored data with both the wearable device and admin dashboard through the cloud.
FR-4	Wearable Device Display	The wearable device must be able to display the temperature of the area where the worker is currently present.
FR-5	SMS Notification	If the temperature of the area is found to reach dangerous levels, the worker should be informed via SMS to their phone instructing them to leave the area.
FR-6	Admin Dashboard	If the temperature of the area is found to reach dangerous levels the admin is informed via the dashboard and must take the necessary precautions.

4.2 NON-FUNCTIONAL REQUIREMENTS

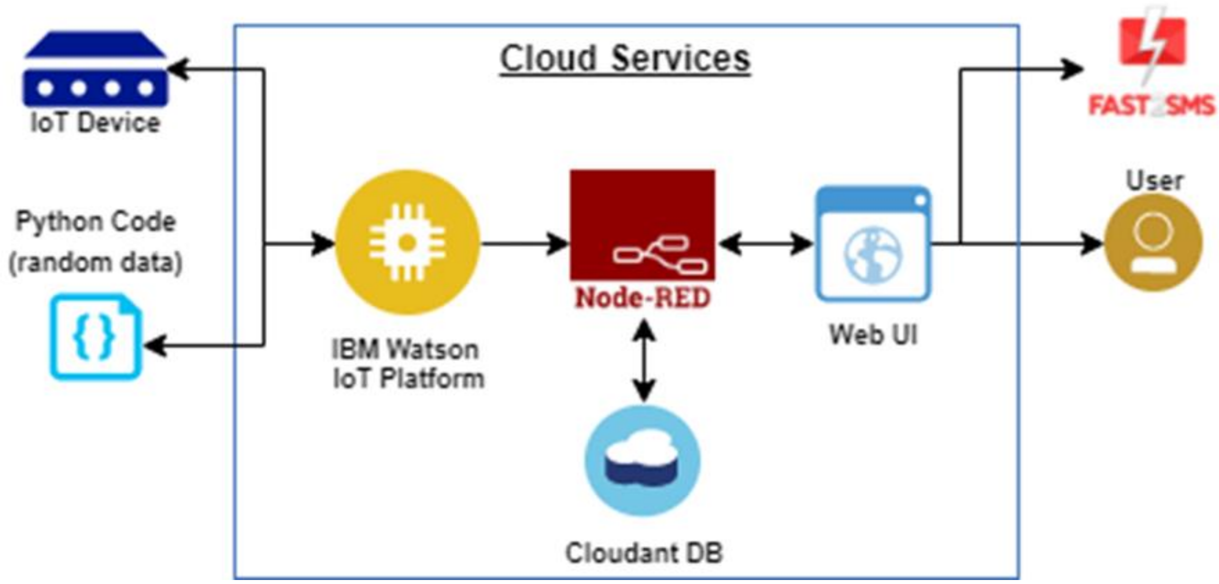
NFR NO.	Non-Functional Requirement	Description

NFR-1	Usability	1.The wearable device should be slim and not annoy or disturb the workers who are wearing them. 2.They should also reliably display the temperature without large delays and notifications should be clear in cases of detected danger.
NFR-2	Security	1.The connection of the beacons to the cloud and wearable devices should be secure. 2.The security of the database housing all the temperature data should also be bolstered.
NFR-3	Reliability	1.The wearable device should be able function without any faults even dangerous temperatures. 2.If a fault is detected it should notify The user and the admin to be repaired and replaced immediately.
NFR-4	Performance	1.The device should update temperature readings in real time and requires high end sensors and processors to do so. 2.The time to send data to the cloud and other devices should also be made as small as possible.

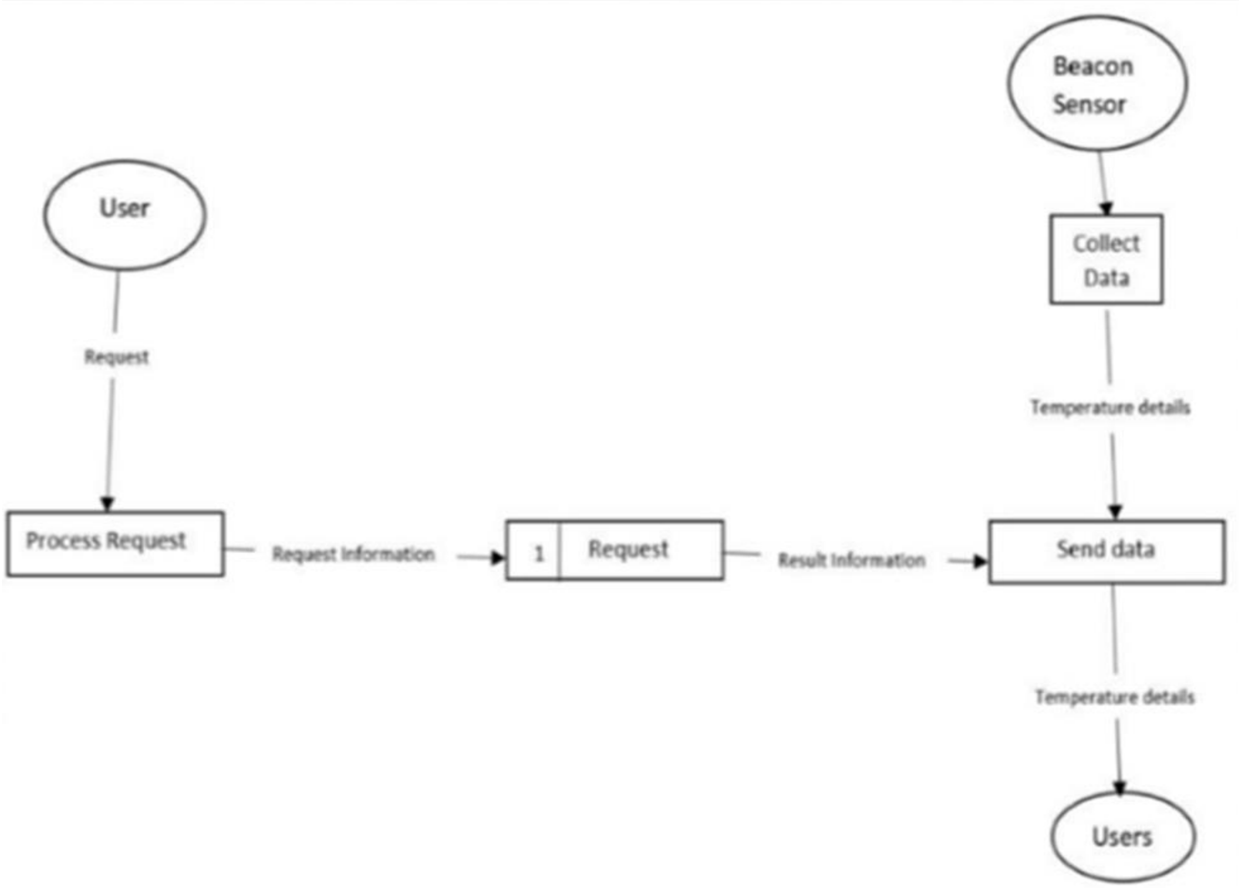
5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

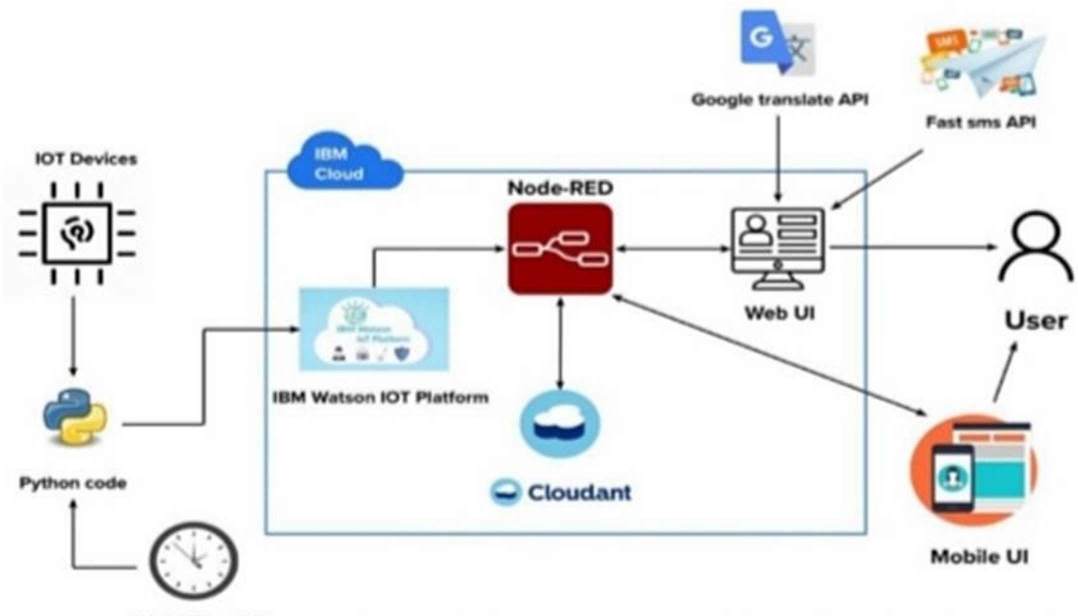
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



Example: DFD Level 0 (Industry Standard)



5.2 SOLUTION & TECHNICAL ARCHITECTURE



S.NO	Component	Description	Technology
1.	User Interface	Web UI, Mobile App, SMS service and Wearable devices	Node-RED, Fast sms and MIT App inventor
2.	Application Logic-1	Getting input from smart beacons	Embedded C and Python
3.	Application Logic-2	Process data in cloud	IBM Watson IOT platform, Cloudant DB and Node-RED
4.	Application Logic-3	Display data to the user	Web UI, Fast sms and Mobile application
5.	Database	Real time database	Cloudant DB
6.	Cloud Database	Database Service on Cloud	IBM Cloudant
7.	External API-1	To send sms to user	Fast sms API
8.	External API-2	Language for the website is writt to be dynamic	Google translate API
9.	External API-3	To access time	World time API

10.	Smart Beacon	To monitor the area and update the stats in the cloud	Node MCU and Sensors
11.	Infrastructure (Server Cloud)	Application Deployment on Cloud	IBM Cloud

Application Characteristics :

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	The Node-RED open source frameworks are used build the web application as well as to communicate with the mobile application and to handle alert sms	Node-RED framework
2.	Scalable Architecture	The 3 – tier architecture used with a separate user interface, application tier and data tier makes it easily scalabl	IBM Watson Studio

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story/ Task	Acceptance criteria	Priorit y	Releas e

Technician	Installation	USN-1	The technician mustinstall the smartbeacons at points to ensurethe entire area ofthe plant iscovered.	A beacon can be found in every areaof the plant.	High	Sprint-1
	Data Gathering	USN-2	The beacons obtainthe temperature of their respective area using sensors.	The temperature of areas within the plant is obtained.	High	Sprint-1
	Data Sync	USN-3	The beacons sendtheir data to thecloud in thereal time which is in turn sent to nearby wearable devices and the administrators dashboard.	Data is sent to the cloud successfully and synced with other devices.	High	Sprint-1
Worker	Wearable device display	USN-4	The wearable devices shoulddisplay the datasent by beacons withinthe area.	The user can seethe temperature of the area ontheir device.	High	Sprint-1
	Wearable device adjustments	USN-5	The user can adjust the size of the wearabledevice to better suit them.	The user can make adjustments to the deviceto make working with it more comfortable.	Low	Sprint-2
	Wearable display customization	USN-6	The user can adjust the device display to suit theirneeds on the device itself.	The user can modify thedisplay of the device to increase readability.	Medium	Sprint-2
	SMS Notifications	USN-7	The user is sent a notification to their phone from the wearable devicethrough an API whenthe area theyare in reaches dangerous temperatures.	The user is informed of potential danger via SMS as soon as it is detected by the beacons.	High	Sprint-1
Administrator	Admin Dashboard	USN-8	The beacons send thedata through the cloudtoa dashboard which is run by the administrator.	The data of all the beaconscan be viewed by the administrator of the plant.	High	Sprint-1

	Dashboard Customization	USN-9	The dashboard can becustomized by the admin to suit theirpersonal requirements andpriorities.	The admin can customize theUI for theirdashboard .	Medium	Sprint-2

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

6.3 Reports From JIRA

7.CODING & SOLUTIONING

7.1 Feature 1

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <DHT.h>
WiFiClient wifiClient;
String data3;
#define DHTTYPE DHT11
#define DHTPIN 9
DHT dht(DHTPIN, DHTTYPE);

#define ORG "v6wg8x"
#define DEVICE_TYPE "nodeMcu"
#define DEVICE_ID "NodeMCU"
#define TOKEN "123456789"
#define speed 0.034

void callback(char* topic, byte* playload, unsigned int payloadLength);

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char topic[] = "iot-2/cmd/test/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, callback , wifiClient);
void publishData();
```

```
String command;
String data = "";

long duration;
float dist;



void setup()
{
  Serial.begin(115200);
  dht.begin();
  wifiConnect();
  mqttConnect();
}

void loop() {
  publishData();
  delay(500);

  if (!client.loop()) {
    mqttConnect();
  }
}

void wifiConnect() {
```

```
  Serial.print("Connecting to "); Serial.print("Wifi");
  WiFi.begin("SSID","Passord");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.print("WiFi connected, IP address: "); Serial.println(WiFi.localIP());
}

void mqttConnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting MQTT client to "); Serial.println(server);
    while (!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }
    initManagedDevice();
    Serial.println();
  }
}

void initManagedDevice() {
  if (client.subscribe(topic)) {
    Serial.println("IBM subscribe to cmd OK");
  } else {
```

```
    Serial.println("subscribe to cmd FAILED");
  }
}

void publishData()
{
  int sensorValue = analogRead(34); //MQT 135 connected to GPIO 34 (Analog ADC1_CH6)
  Serial.print("AirQua=");
  Serial.print(sensorValue, DEC);
  Serial.println(" PPM");
  float humid = dht.readHumidity();
  float temp = dht.readTemperature(true);
  float airQty = sensorValue/4095;
  String payload = "{\"Temperature\":";
  payload += temp;
  payload += "}";
  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish OK");
  }
  payload = "{\"Air Quality\":";
  payload += airQty;
  payload += "%}";
  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish OK");
  }
```

```
}
void callback(char* subscribeTopic, byte* payload, unsigned int payloadLength) {
  Serial.print("callback invoked for topic:");
  Serial.println(subscribeTopic);
  for (int i = 0; i < payloadLength; i++) {
    dist += (char)payload[i];
  }
  Serial.println("data:" + data3);
  if (data3 == "lighton") {
    Serial.println(data3);
  }
  data3 = ""
}
```

7.2 Feature 2

```
float temp;
float vout;
float vout1;
int LED = 13;
int gasSensor;
int piezo = 7;
void setup()
{
pinMode(A0,INPUT);
pinMode(A1,INPUT);
pinMode(LED,OUTPUT);
```

```
pinMode(piezo,OUTPUT);
pinMode(5,OUTPUT);

Serial.begin(9600);
}
void loop()
{
vout=analogRead(A1);
vout1=(vout/1023)*5000;
temp=(vout1-500)/10;
gasSensor=analogRead(A0);
if (temp>=65)
{
digitalWrite(LED,HIGH);
digitalWrite(5,HIGH);
}
else
{
digitalWrite(LED,LOW);
digitalWrite(5,LOW);
}
if (gasSensor>=100)
{
digitalWrite(piezo,HIGH);
}
```

```
else
{
digitalWrite(piezo,LOW);
}
Serial.print("in DegreeC= ");
Serial.print(" ");
Serial.print(temp);
Serial.print("\t");
Serial.print("GasSensor=");
Serial.print(" ");
Serial.print(gasSensor);\
delay(1000);
}
```

8.TESTING

8.1 Test Cases

Test case ID	Feature Type	Component	Test Scenario
TC_OO1	Functional	IBM cloud	Create the IBM Cloud services which are being used in this project.
TC_OO2	Functional	IBM Cloud	Configure the IBM Cloud services which are being used in completing this project.

TC_OO3	Functional	IBM Watson IoT Platform	IBM Watson IoT platform acts as the mediator to connect the web application to IoT devices, so create the IBM Watson IoT platform.
TC_OO4	Functional	IBM Watson	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.
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 Platform.

TC_OO6	Functional	Node Red	Create a Node-RED service.
TC_OO7	Functional	Python 3.7.0	Develop a python script to publish random sensor data such as temperature, humidity level, soil moisture to the IBM IoT platform
TC_OO8	Functional	Python 3.7.0	After developing python code, commands are received just print the statements which represent the control of the devices.
TC_OO9	Functional	IBM Cloudant DB	Publish Data to The IBM Cloud

TC_O10	Web UI	Node Red & MIT Inventor	Create Web UI in Node- Red
TC_O11	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

Pre-Requisite	Steps To Execute	Test Data
IBM Cloud Login ID & Password	1.Go to IBM Cloud signup page 2.Enter e-mail id and other credentials 3.Enter a password	https://cloud.ibm.com/login
IBM Cloud Login ID & Password	1.Go to Cloud login 2.Enter user ID & Password 3.Verify login by the popup display	https://cloud.ibm.com/login

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 and click Sign in IBM Watson Platform	https://eynrcc.internetofthings.ibmcloud.com/dashboard/devices/browse/add
IBM Watson IoT Platform Login ID & Password	1.Login to IBM Watson Platform 2. Click Add Device 3.Enter the details and click Finish. Create Device ID & Device type 4.Turn on Device Simulator and click simulation running. Enter the values of temperature, Humidity , Soil moisture 5.Click Send & Save. Verify the displayed result of the levels	Temperature, Humidity , sensor values are generated randomly in simulation
Node Red Installation	1.Install node red and open node red in command prompt 2.Select IBM input in IoT	https://cloud.ibm.com/developer/appservice/create-app?starterKit=59c9d5bd-4d31-3611-897a-f94eea80dc9f&defaultLanguage=undefined

Node Red Installation	1.Select IBM IoT input in Node. In IBM IoT Watson Platform, go to apps and click on generate API keys. 2.Copy & paste generated 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 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", "Soil moisture" &"Humidity" 5.Finally add light ON/OFF buttons to the IBM IoT and debug. Verify the output from NODE RED using Local host link	Values of sensors and button for light ON/OFF is displayed

Python 3.7.0(64 bit) installation	1.Download and install Python 3.7.0 2.Develop python code	https://www.python.org/downloads/release/python-370/
Python 3.7.0(64 bit) installation	1.Downlinstall Python 3.7.0 2.After python code	Get the output from the code
IBM Cloud Login ID & Password	1.Run the python code 2.Verify the displayed output	Publishment of python code
MIT Inventor Login ID & password	1.Go to Node Red. Select http in & http response. Add functions and select another http in and http response. Connect them to IBM IoT output and function.Print the command statements such as light ON/OFF and sensor 2.Go to MIT app inventor and create frontend using buttons,horizontal arrangement, text bar, etc. Add blocks and so on to create back end. Verify the output	Sensors values and command values can be seen in the mobile application
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 is connected by NODE RED

	cloudant and verify the results

Expected Result	Actual Result	Status	Comments	TC for Automation (Y/N)	BUG ID	Executed By
User should sign up IBM cloud and details should be verified	Working as expected	Pass	Results verified	No		Kalasurya.A , Saranya.K, Suganya.S, Switha.D
User login to IBM Cloud and should be navigated to IBM Cloud dashboard page	Working as expected	Pass	Results verified	No		Kalasurya.A , Saranya.K, Suganya.S, Switha.D
User should be navigated to IBM IoT Watson Platform	Working as expected	Pass	Results verified	No		Kalasurya.A , Saranya.K, Suganya.S, Switha.D
Temperature, Humidity , sensor values should be randomly generated	Working as expected	Pass	Results verified	No		Kalasurya.A , Saranya.K, Suganya.S, Switha.D
User should be able to see the Node Red page	Working as expected	Pass	Results verified	No		Kalasurya.A , Saranya.K, Suganya.S, Switha.D

Values of sensors and button for light ON/OFF should be displayed	Working as expected	Pass	Results verified	No		Kalasurya.A , Saranya.K, Suganya.S, Switha.D
User should be able to develop a python code	Working as expected	Pass	Results verified	No		Sidharthini K.S,Abarna A,Kayathri V,Mathumitha A
User should be able to get the results from the developed code	Working as expected	Pass	Results verified	No		Kalasurya.A , Saranya.K, Suganya.S, Switha.D
User should be able to publish the code	Working as expected	Pass	Results verified	No		Kalasurya.A , Saranya.K, Suganya.S, Switha.D

Sensors values and command values should be seen in the mobile application	Working as expected	Pass	Results verified	No		Kalasurya.A , Saranya.K, Suganya.S, Switha.D
User should be able to connect the Cloudant and Node Red	Working as expected	Pass	Results verified	No		Kalasurya.A , Saranya.K, Suganya.S, Switha.D

8.2 User Acceptance Testing

1. Defect Analysis

This reportshows the numberof resolved or closed bugs at each severity level,and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	7	2	2	3	14
Duplicate	1	0	3	0	4
External	2	4	0	1	7
Fixed	15	2	5	18	40
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	3	4
Won't Fix	0	7	2	1	10
Totals	25	15	14	26	80

2. Test Case Analysis

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
ClientApplication	31	0	0	31
Security	16	0	0	16
Outsource Shipping	9	0	0	9
Exception Reporting	10	0	0	10
Final ReportOutput	7	0	0	7
Version Control	5	0	0	5

9.RESULTS

9.1 Performance Metrics

NFT - Risk Assessment

S.NO	Project Name	Scope/feature	FunctionalChanges	HardwareChanges	Software Changes	Load/VolumeChanges	Risk Score	Justification
1.	Motor ON/OFF	Existing	Moderate	No Changes	Moderate	>10 to 30%	ORANGE	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.	Python script	Python coding	https://www.python.org/psf/sponsors/#heroku	Depend on the delivered code
2.	Node-red	Sensor & command values	https://nodered.org/	Sensor values
3.	MIT App Inventor	Motor control/Sensors notification	https://appinventor.mit.edu/about/termsofservice	Notifications

10.ADVANTAGES

- Automation of sensors leads to better monitoring of devices.

- Use of sensors for monitoring reduces human effort.
- When the temperature increases, the alarm is automatically turned ON.
- Once fire is detected, the sprinkler immediately extinguishes the fire.

11.CONCLUSION

Currently, IoT is present and gaining more traction in a lot of fields, and one of the most important field is industrial applications. There are a huge number of ways in which industries can make use of IoT to improve working conditions, efficiency, cutting costs and improving the overall growth of the sector. However, hazard monitoring and mitigation is often overlooked in industrial areas.

Therefore, this project specifically aims to make use of IoT to actively monitor and analyze various factors in a typical heavy industrial zone like temperature and levels of gases in the environment. If, the above parameters exceed the recommended safe values. The system can track the same issue alerts. Also the data generated in real time can provide important information about how smoothly the work is going on in different zones.

This system can be deployed in many industrial areas like mining, underground factories, metal refineries, automatic welding factories and even heavy parts production lines. It will help to provides a safe and efficient working environment in such areas, while also opening new paths to improve the safety parameters of these places.

12.FUTURE SCOPE

Firstly, providing service engineers and manage remote access to industrial machines.

Secondly, allowing web-based virtual network connection to manage and observe HMI functions on the IOT platform.

In addition, it offers predictive analytics for maintaining machines and identifying potential problems.

Most importantly, it controls, monitors, and manages data from multiple systems in various locations. Simultaneously storing the collected data at a central cloud application. Hence, realtime machine data and analysis are easily accessible using industrial communication networks.

13.APPENDIX

Source Code

```
float temp;
float vout;
float vout1;
int LED = 13;
int gasSensor;
int piezo = 7;
void setup()
{
pinMode(A0,INPUT);
pinMode(A1,INPUT);
pinMode(LED,OUTPUT);
pinMode(piezo,OUTPUT);
pinMode(5,OUTPUT);

Serial.begin(9600);
}
void loop()
{
vout=analogRead(A1);
vout1=(vout/1023)*5000;
temp=(vout1-500)/10;
gasSensor=analogRead(A0);
if (temp>=65)
```

```
{
digitalWrite(LED,HIGH);
digitalWrite(5,HIGH);
}
else
{
digitalWrite(LED,LOW);
digitalWrite(5,LOW);
}
if (gasSensor>=100)
{
digitalWrite(piezo,HIGH);
}
else
{
digitalWrite(piezo,LOW);
}
Serial.print("in DegreeC= ");
Serial.print(" ");
Serial.print(temp);
Serial.print("\t");
Serial.print("GasSensor=");
Serial.print(" ");
Serial.print(gasSensor);\
delay(1000);
```

}

GitHub Link :

https://github.com/IBM-EPBL/IBM-Project-43561-1660718031

Project Demo Link:

https://drive.google.com/file/d/1zWBk2FGpwUGG_CpXMvf8rjhP-Tz5rT0t/view?usp=drivesdk