PREDICTING THE ENERGY OUTPUT OF WIND TURBINE BASED ON WEATHER CONDITIONS

PROJECT REPORT

Submitted by

Gadhadharan G	193002028
Sujitha D	193002105
Suvethasri V	193002106
Vishal S R	193002122

EEC ELECTIVE

Department of Electronics and Communication Engineering

Sri Sivasubramaniya Nadar College of Engineering
(An Autonomous Institution, Affiliated to Anna University)
Rajiv Gandhi Salai (OMR), Kalavakkam – 603 110

ODD SEM 2022 – 2023

Project Report Format

CHAPTER - I

INTRODUCTION

1.1 Project Overview

Waste management is one of the primary problem that the world faces irrespective of the case of developed or developing country. The key issue in the waste management is that the garbage bin at public places gets overflowed well in advance before the commencement of the next cleaning process. Rapid increase in population, has led to the improper waste management in cities resulting in increased pests and spreading of diseases. Nowadays, the Garbage Collecting Vehicle (GCV) collects the waste twice or thrice in a week. So, the problem is over flowing of wastages on the roads. It in turn leads to various hazards such as bad odor & ugliness to that place which may be the root cause for spread of various diseases. It also causes land pollution. To avoid all such hazardous scenario and maintain public cleanliness and health this work is mounted on a smart garbage system. The main theme of the work is to develop a smart intelligent garbage alert system for a proper garbage management.

1.2 Purpose

With rapid increase in population, the issues related to sanitation with respect to garbage management are degrading immensely. It created unhygienic conditions for the citizens in the nearby surrounding, leading to the spread of infectious diseases and illness. Due to the increasing waste, the public bins which are used for collecting this waste are overflowing, the locality is jumbled of trash, causing not only malodorous streets but also a negative impact on the health and environment.

Due to civilians emptying their overloaded dustbins in open spaces, environmental pollution increases. The waste is a great hassle for our health and the environment it has many effects which are dreadful. Trash is breeding ground for bacteria, insects, flies these flies are the same that roam around the eatable and drop the off springs. Thus they increase the risk with food poisoning, typhoid, gastroenteritis, salmonella, the insects cause malaria dengue etc., beside these flies and insects other animals that prosper from the trash are the rats and the stray dogs spreading diseases. The garbage also causes various respiratory diseases the toxic contaminates such as co2, methane, nitrous oxide beside health issues adversely affect the environment causing air pollution water pollution. Disposal of hazardous waste like the electronic items, plastics in water affect the aquatic life and indirectly the human beings. Overflowing garbage is also a public hassle and eyesore. Everyone wants to visit fresh clean cities. A malodorous city with trash all around the place does not attract tourist thus losing the money revenue and the opportunities.

CHAPTER – II

LITERATURE SURVEY

2.1 Existing Problems

This paper proposes a smart alert system for garbage clearance by giving an alert signal to the municipal web server for instant cleaning of dustbin with proper verification based on level of garbage filling. It sends the alert to the municipal web server once if garbage is filled. It also enhances the smart garbage alert system by providing automatic identification of garbage filled in the dustbin and sends the status of clean-up to the server affirming that the work is done. The whole process is upheld by an embedded module and IOT Facilitation. The real time status of how waste collection is being done could be monitored and followed up by the municipality authority with the aid of this system.

2.2 References

[1] Pranay Mahajan, Avani Jain, " Survey Paper on Municipal Solid Liquid based smart waste management system – Zero waste using Internet of Things", IJESRT Journal, (PDF) SURVEY PAPER ON MUNICIPAL SOLID LIQUID BASED SMART WASTE MANAGEMENT SYSTEM-ZERO WASTE USING INTERNET OF THING | IJESRT Journal - Academia.edu

[2] N. Sathish Kumar; B. Vuayalakshmi; R. Jenifer Prarthana; A. Shankar, "IOT based smart garbage alert system using Arduino UNO", IEEE Xplore
IOT based smart garbage alert system using Arduino UNO IEEE Conference Publication | IEEE Xplore

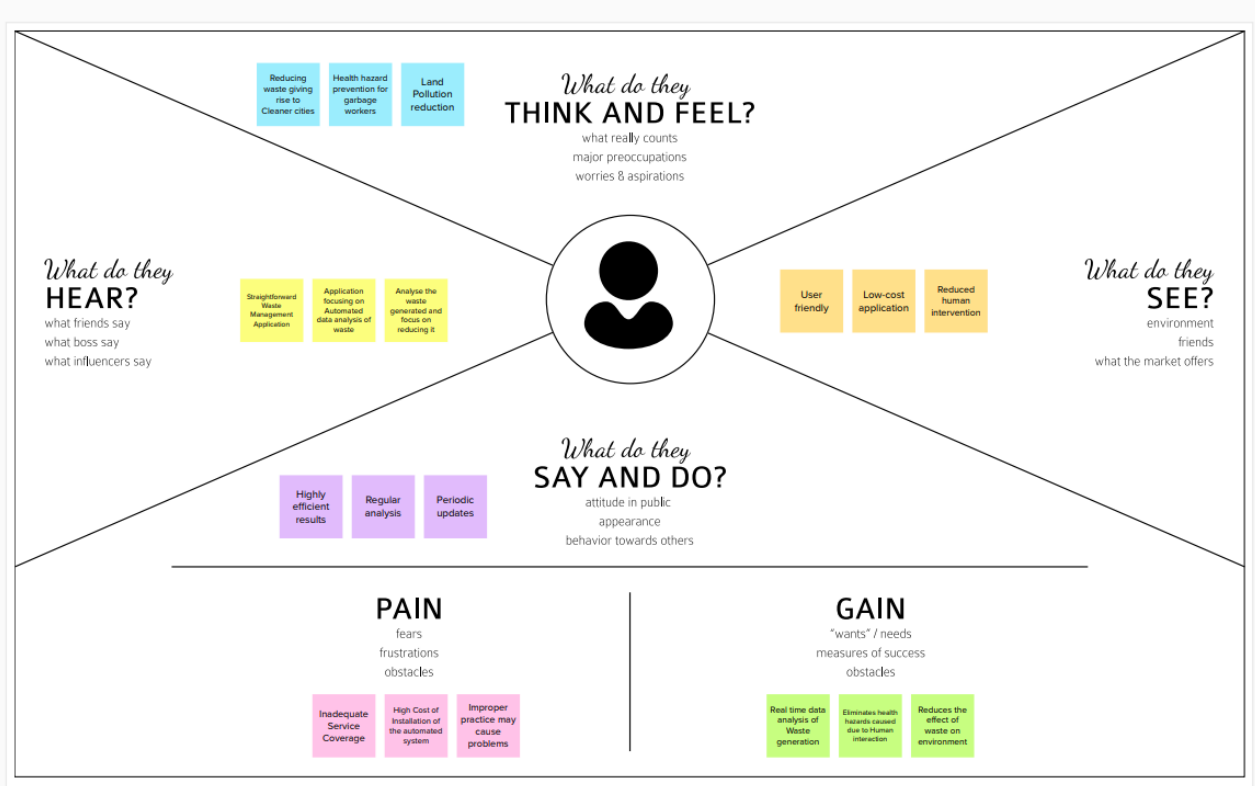
2.3 Problem Statement Definition

One of the most critical drawback in windmill farms is the uncertainty of the power it can generated thus reducing the reliability of the system. The power generated is directly proportional to the wind speed. Since, wind speed is affected by weather it is necessary to model the relationship between weather conditions and power generated.

CHAPTER – III

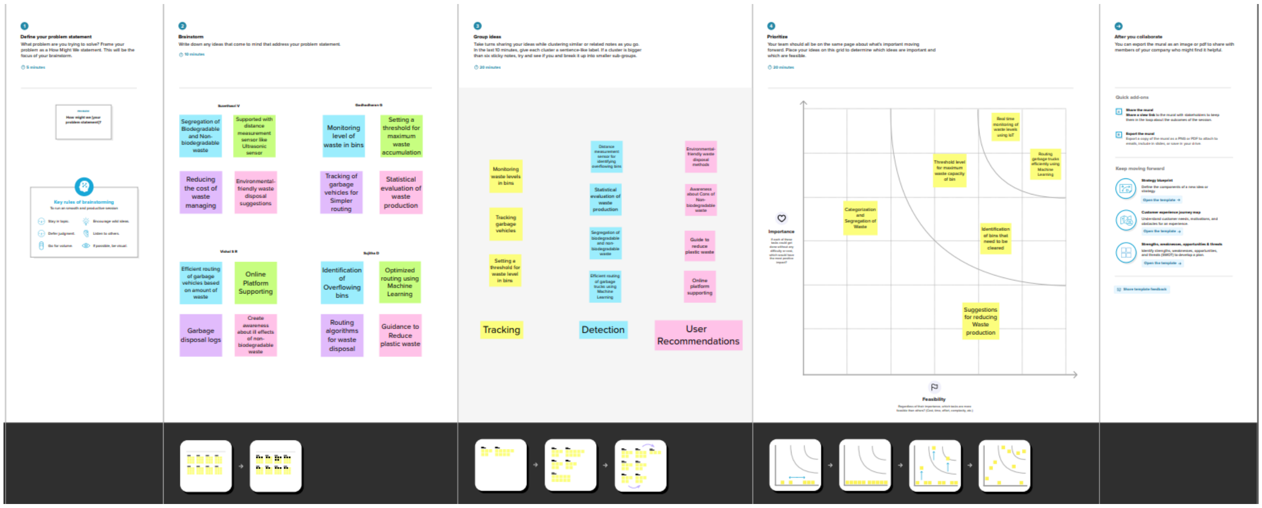
IDEATION AND PROPOSED SOLUTION

3.1 Empathy map canvas:


Reducing waste giving rise to Cleaner cities
Health hazard prevention for garbage workers
Land Pollution reduction
What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
What do they
HEAR?
what friends say
what boss say
what influencers say
Straightforward Waste Management Application
Application focusing on Automated data analysis of waste
Analyse the waste generated and focus on reducing it
User friendly
Low-cost application
Reduced human intervention
What do they
SEE?
environment
friends
what the market offers
Highly efficient results
Regular analysis
Periodic updates
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
PAIN
fears
frustrations
obstacles
Inadequate Service Coverage
High Cost of Installation of the automated system
Improper practice may cause problems
GAIN
"wants" / needs
measures of success
obstacles
Real time data analysis of Waste generation
Eliminates health hazards caused due to Human interaction
Reduces the effect of waste on environment

3.2 Ideation and brainstorming:

In this brainstorming session we defined our problem statement, addressed various challenges involved in making our model ready for use. We have also prioritized our ideas based on feasibility and importance.


Define your problem statement
Brainstorm
Write down any ideas that come to mind that address your problem statement.
10 minutes
Segregation of Biodegradable and Non-biodegradable waste
Supported with distance measurement sensor like Ultrasonic sensor
Monitoring level of waste in bins
Setting a threshold for maximum waste accumulation
Reducing the cost of waste managing
Environmental-friendly waste disposal suggestions
Tracking of garbage vehicles for Simpler routing
Statistical evaluation of waste production
Efficient routing of garbage vehicles based on amount of waste
Online Platform Supporting
Identification of Overflowing bins
Optimized routing using Machine Learning
Garbage disposal logs
Create awareness about ill effects of non-biodegradable waste
Routing algorithms for waste disposal
Guidance to Reduce plastic waste
Group ideas
Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, 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
Monitoring waste levels in bins
Tracking garbage vehicles
Setting a threshold for waste level in bins
Distance measurement sensor for identifying overflowing bins
Statistical evaluation of waste production
Segregation of biodegradable and non-biodegradable waste
Efficient routing of garbage trucks using Machine Learning
Environmental-friendly waste disposal methods
Awareness about Cons of Non-biodegradable waste
Guide to reduce plastic waste
Online platform supporting
Tracking
Detection
User Recommendations
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
Real time monitoring of waste levels using IoT
Threshold level for maximum waste capacity of bin
Routing garbage trucks efficiently using Machine Learning
Categorization and Segregation of Waste
Identification of bins that need to be cleared
Suggestions for reducing Waste production
Importance
Feasibility
After you collaborate
Quick add-ons
Keep moving forward

3.3 Proposed solution:

❖ Problem statement:

Garbage level detection in bins-

• Improper waste collection leads to significant land pollution, especially in urban areas. This is due to reduced availability of garbage workers.

• This project aims to automate the Garbage collection scheduling and optimize it.

• The status of the bins are constantly monitored, along with their location remotely via a Web Application and the data is collected and stored on a cloud platform.

• It alerts the admins to empty the bin when they are full.

- Idea/ Solution description:

  Main Solution-

  • The GPS coordinates and the garbage level of the garbage bin will be sent to the IBM IoT platform by developing a python script.

  • The location of the bins along with bin status can be viewed in the Web Application created using Node-RED service.

  • This data is stored in Cloudant Database.

  • It alerts the admin if the bin value crosses the threshold value, i.e., when the bins are full.

  Additional benefit-

  • Remote monitoring of the garbage bin location and status.

  • Automated notification sent to admin when Bin is full.

  • Both of these methods reduce unnecessary human involvement and interference.

- Novelty:

  • User friendly web application

  • Completely automated and can be remote

- Social impact:

  • Reduction in Land pollution
  • Reduction in unnecessary travel by Garbage workers

- Business model:

  It can be used universally to detect garbage levels and give orders to dispose the garbage automatically.

- Scalability:

  • User friendly User Interface to notify admins, report problems or provide feedbacks

  • Improved accuracy using high quality sensors

CHAPTER – IV

REQUIREMENT ANALYSIS

4.1 Functional requirement:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login through Google Login through Email
FR-4	Access the Location and Bin status in the Web Application	Information to be displayed on the dashboard
FR-5	Monitoring Bin Status	By setting a threshold value above which the Bin is considered full
FR-6	Notifying Admin if Bin is full	Notification to be sent to the Admin if threshold is crossed

4.1 Non-Functional requirement:

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

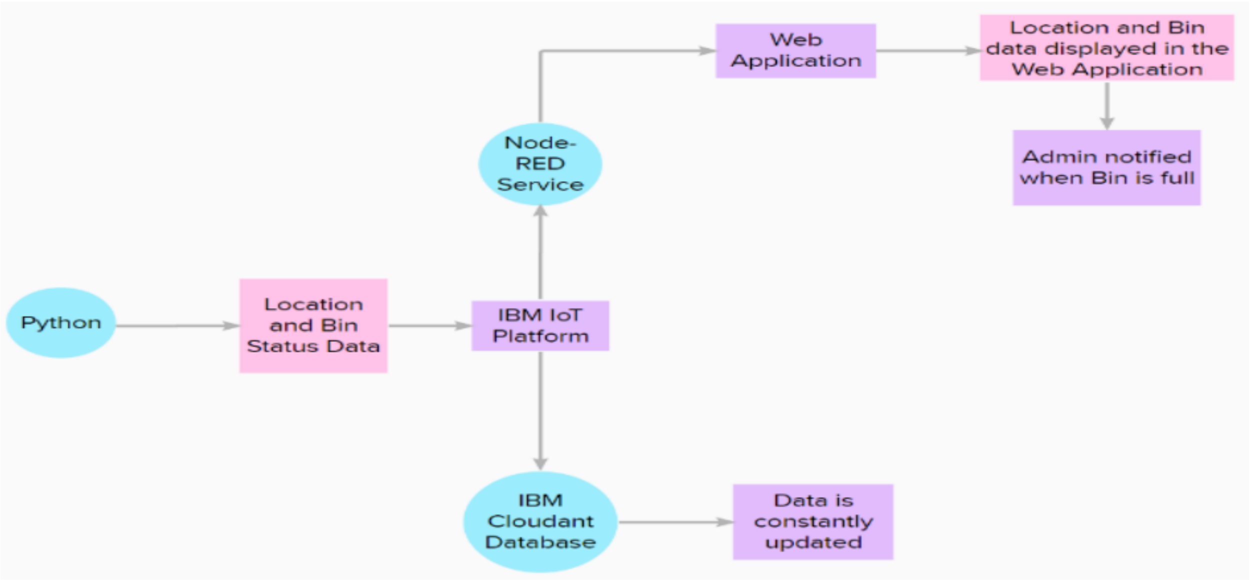
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It has a user-friendly and interactive User Interface
NFR-2	**Security**	It secures the personal information provided by the users
NFR-3	**Reliability**	Results are constantly instantaneous with very low failure rate
NFR-4	**Performance**	Consistent Performance with admins being immediately notified for higher efficiency
NFR-5	**Availability**	Remote monitoring is constantly happening
NFR-6	**Scalability**	It can be installed easily and monitored remotely

CHAPTER – V

PROJECT DESIGN

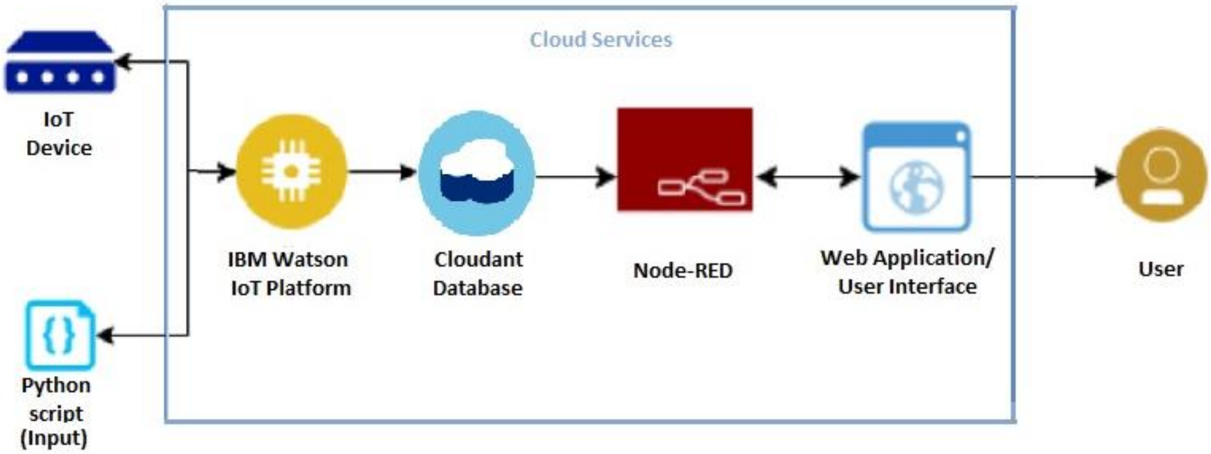
5.1 Data flow diagrams:

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.


Web Application
Location and Bin data displayed in the Web Application
Node-RED Service
Admin notified when Bin is full
Python
Location and Bin Status Data
IBM IoT Platform
IBM Cloudant Database
Data is constantly updated

5.2 Solution & Technical Architecture:

Solution architecture diagram:



Technical architecture:

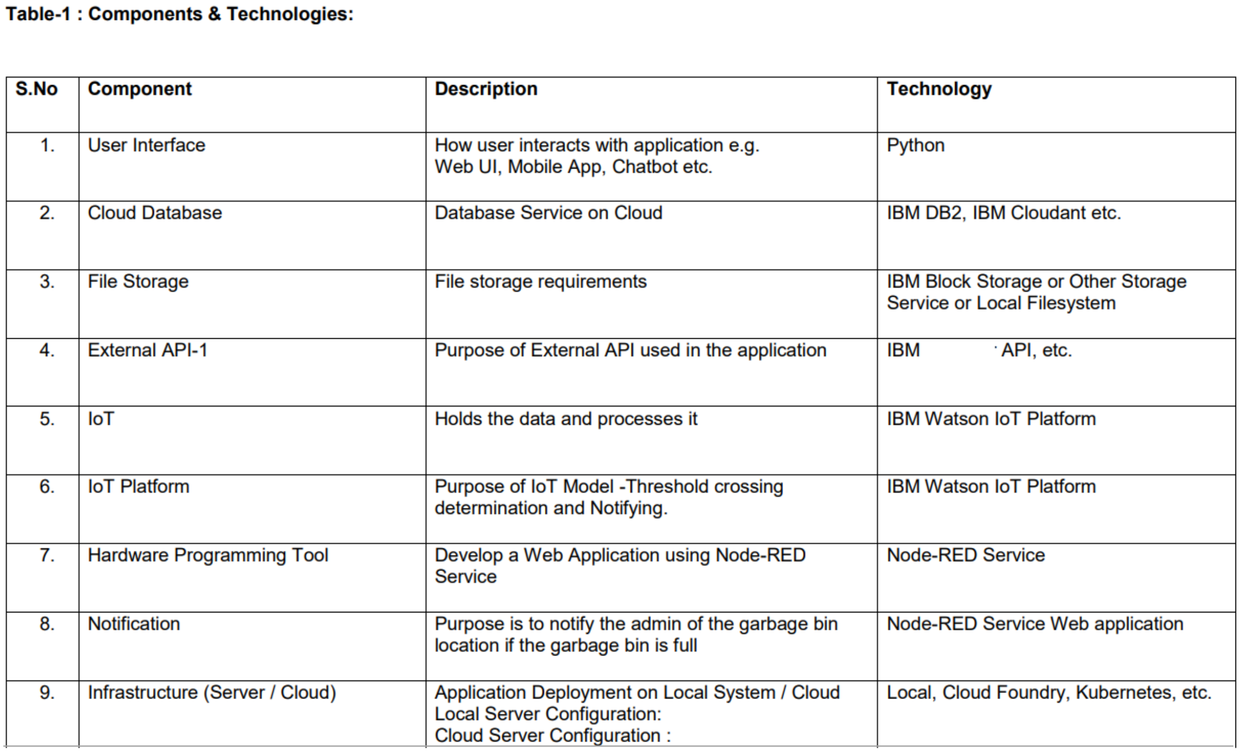


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	Python
2.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
3.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
4.	External API-1	Purpose of External API used in the application	IBM · API, etc.
5.	IoT	Holds the data and processes it	IBM Watson IoT Platform
6.	IoT Platform	Purpose of IoT Model -Threshold crossing determination and Notifying.	IBM Watson IoT Platform
7.	Hardware Programming Tool	Develop a Web Application using Node-RED Service	Node-RED Service
8.	Notification	Purpose is to notify the admin of the garbage bin location if the garbage bin is full	Node-RED Service Web application
9.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

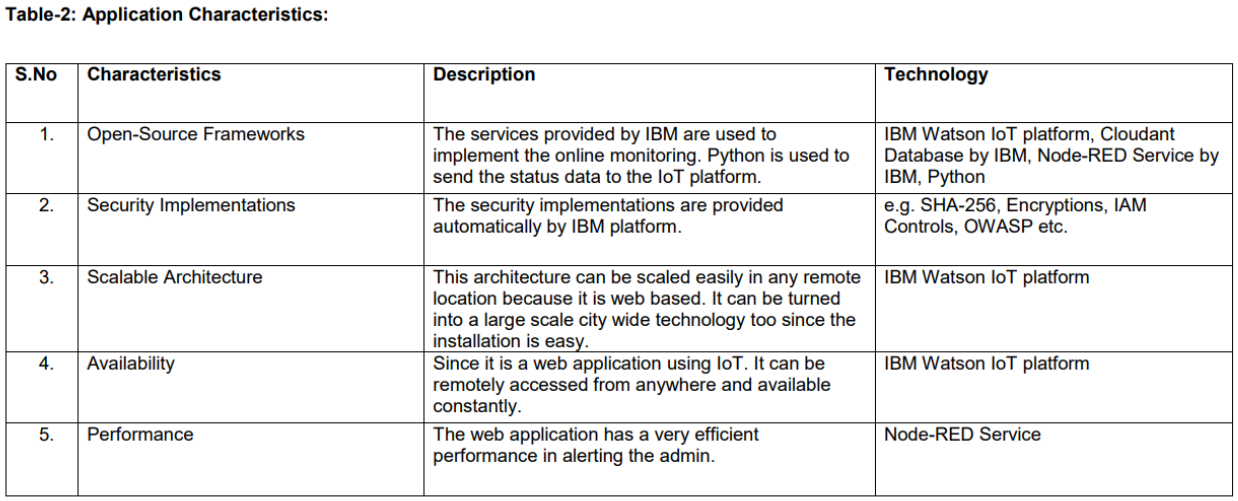
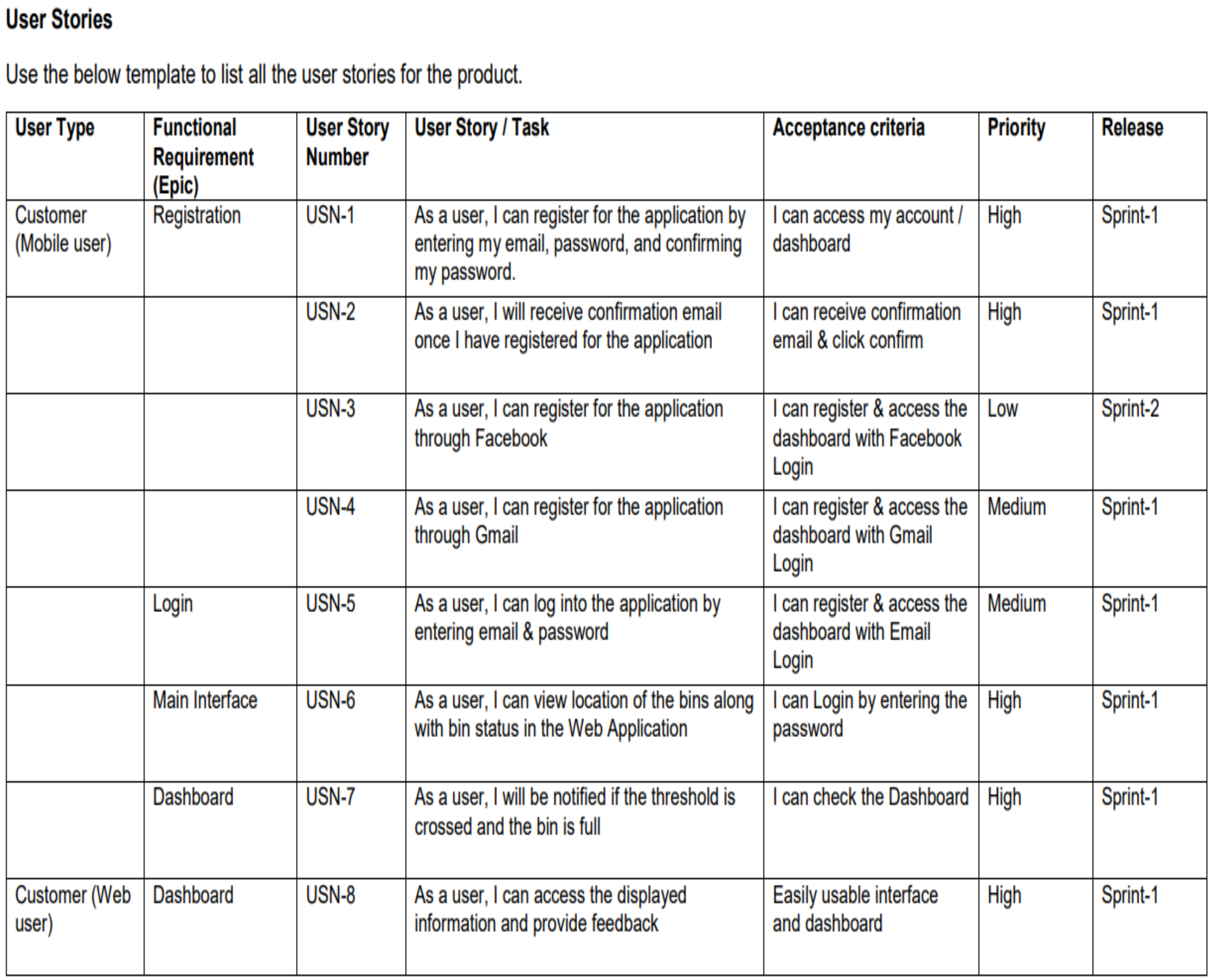


Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	The services provided by IBM are used to implement the online monitoring. Python is used to send the status data to the IoT platform.	IBM Watson IoT platform, Cloudant Database by IBM, Node-RED Service by IBM, Python
2.	Security Implementations	The security implementations are provided automatically by IBM platform.	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	This architecture can be scaled easily in any remote location because it is web based. It can be turned into a large scale city wide technology too since the installation is easy.	IBM Watson IoT platform
4.	Availability	Since it is a web application using IoT. It can be remotely accessed from anywhere and available constantly.	IBM Watson IoT platform
5.	Performance	The web application has a very efficient performance in alerting the admin.	Node-RED Service

5.3 User Stories:



User Stories

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

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail Login	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can register & access the dashboard with Email Login	Medium	Sprint-1
	Main Interface	USN-6	As a user, I can view location of the bins along with bin status in the Web Application	I can Login by entering the password	High	Sprint-1
	Dashboard	USN-7	As a user, I will be notified if the threshold is crossed and the bin is full	I can check the Dashboard	High	Sprint-1
Customer (Web user)	Dashboard	USN-8	As a user, I can access the displayed information and provide feedback	Easily usable interface and dashboard	High	Sprint-1

CHAPTER – VI

CODING AND SOLUTIONING

6.1 Code

```
#IBM
Watson
IOT
Platform
           #pip install wiotp-sdk
           import wiotp.sdk.device
           import time
           import random
           myConfig = {
               "identity": {
                   "orgId": "hj5fmy",
                   "typeId": "NodeMCU",
                   "deviceId":"12345"
               },
               "auth": {
                   "token": "12345678"
               }
           }


           def myCommandCallback(cmd):
               print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
               m=cmd.data['command']


           client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
           client.connect()


           while True:
               temp=random.randint(-20,125)
               hum=random.randint(0,100)
               myData={'temperature':temp, 'humidity':hum}
```

```
client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

6.2 Main Code

```
#include
<ESP8266WiFi.h>

                  #include <PubSubClient.h>

                  WiFiClient wifiClient;



                  //Enter your network credentials below in ssid and password

                  const char* ssid = " ";

                  const char* password = " ";



                  //Provide your IBM IOT Platform credentials

                  #define ORG ""

                  #define DEVICE_TYPE ""

                  #define DEVICE_ID ""

                  #define TOKEN ""



                  char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

                  char publishTopic[] = "iot-2/evt/Data/fmt/json";
                  char topic[] = "iot-2/cmd/home/fmt/String";  // cmd  REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING

                  char authMethod[] = "use-token-auth";

                  char token[] = TOKEN;

                  char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;



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

                  PubSubClient client(server, 1883, callback, wifiClient);



                  int publishInterval = 5000; // 30 seconds

                  long lastPublishMillis;

                  String data;



                  void setup()

                  {

                    Serial.begin(9600);

                    pinMode(D0, OUTPUT);

                    wifiConnect();

                    mqttConnect();

                  }



                  void loop() {
                    if (millis() - lastPublishMillis > publishInterval)
                    {
                      publishData();
                      lastPublishMillis = millis();
                    }

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

                  void wifiConnect() {
                    Serial.print("Connecting to "); Serial.print(ssid);
                    WiFi.begin(ssid, password);
                    while (WiFi.status() != WL_CONNECTED) {
                      delay(500);
                      Serial.print(".");
                    }
                    Serial.print("nWiFi 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(client.subscribe(topic));
                      Serial.println("subscribe to cmd OK");
                    } else {
                      Serial.println("subscribe to cmd FAILED");
                    }
                  }

                  void callback(char* topic, byte* payload, unsigned int payloadLength) {
                    Serial.print("callback invoked for topic: ");
                    Serial.println(topic);

                    for (int i = 0; i < payloadLength; i++) {
                      //Serial.print((char)payload[i]);
                      data += (char)payload[i];
                    }

                    Serial.println("Data: " + data );
                    if (data == "lon") {
                      digitalWrite(D0, HIGH);
                    }
                    else if (data == "loff") {
                      digitalWrite(D0, LOW);
                    }
                    data = "";
                  }
                  void publishData()
                  {
                    int a = 10;
                    Serial.print("Sample Value: ");
                    Serial.println(a);
                    String payload = "{\"d\":{\"data\":";
                    payload += a;
                    payload += "}}";

                    Serial.print("\n");
                    Serial.print("Sending payload: ");
                    Serial.println(payload);

                    if (client.publish(publishTopic, (char*) payload.c_str())) {
                      Serial.println("Publish OK");
                    } else {
                      Serial.println("Publish FAILED");
                    }
                  }
```

CHAPTER VII

ADVANTAGES AND DISADVANTAGES

Advantages:

We can obtain a real time data analysis of Waste generation in each and every geographical location where it is installed. It eliminates health hazards due to the unhygienic surrounding caused by the overflowing wastes. Land pollution is reduced. Overall effect of waste on the neighbouring environment is reduced. The application developed is user friendly, low cost and allows remote monitoring. It gives highly efficient results with regular periodic updates.

Disadvantages:

Installation of sensors takes time and cost of labour for this is high. Faulty sensors need to be replaced in person. Proper Service and Internet coverage is required for efficient results.

CHAPTER VIII

CONCLUSION

A smart intelligent garbage alert system for a proper garbage management is developed. The key issue in the waste management is that the garbage bin at public places gets overflowed well in advance before the commencement of the next cleaning process. Various hazards, mosquito breeding and land pollution occurs. This is prevented by this remotely monitored system and makes the job easier for civilians and Garbage collectors.

CHAPTER IX

FUTURE SCOPE

In the future, we plan to concentrate on getting the data by placing ultrasonic sensors to monitor the garbage level in the bins so that the app can be used by organisations responsible for garbage collection.

CHAPTER – X

APPENDIX

GitHub repository link : IBM-EPBL/IBM-Project-30743-1660185446: Smart Waste Management System For Metropolitan Cities (github.com)