

SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITAN CITIES

DOMAIN – INTERNET OF THINGS (IoT)

A PROJECT REPORT

SUBMITTED BY

TEAM ID : PNT2022TMID25449

MANO B

MENTOR : A.N. Duraivel

JENSY J

NEELAVANNAK

ANNAN S

MUNIRAJ N

In the partial fulfillment of the requirements for the award of a degree of BACHELOR OF
ENGINEERING in ELECTRONIC AND COMMUNICATION ENGINEERING

KINGS ENGINEERING COLLEGE

IRUNGATTUKOTTAI

2022 – 2023

CONTENT

1. INTRODUCTION

- Project Overview
- Purpose

2. LITERATURE SURVEY

- Existing problem
- References
- Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

- Empathy Map Canvas
- Ideation & Brainstorming
- Proposed Solution
- Problem Solution fit

4. REQUIREMENT ANALYSIS

- Functional requirement
- Non-Functional requirements

5. PROJECT DESIGN

- Data Flow Diagrams
- Solution & Technical Architecture
- User Stories

6. PROJECT PLANNING & SCHEDULING

- Sprint Planning & Estimation
- Sprint Delivery Schedule
- Reports from JIRA

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

- Feature 1
- Feature 2
- Database Schema (if Applicable)

8. TESTING

- Test Cases
- User Acceptance Testing

9. RESULTS

- Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

- Source Code
- GitHub & Project Demo Link

1. INTRODUCTION

Internet of Things is nothing but the applications performing with the help of internet access.

IoT Communication over the internet has grown from user - user interaction to device – device interactions these days. The IoT concepts were proposed years back but still it's in the initial stage of commercial deployment. Home automation industry and transportation industries are seeing rapid growth with IoT. The basic project idea is to design a smart waste detection system which would automatically notify the officials about the current status of various garbage bins in the city, would have realtime monitoring capabilities, which would be remotely controlled using IoT techniques. This paper introduces you to the use of IoT on one such area, that is, Garbage Detection in smart ways using IoT and see how this can also be a major part of developing a city into a smart city.

Project Overview

A big challenge in the urban cities is that of waste management as there is a rapid growth in the rate of urbanization and thus there is a need of sustainable urban development plans. As the concept of smart cities is very much trending these days and the smart cities cannot be complete without smart waste management system. There needs to be system that gives prior information of the filling of the bin that alerts the municipality so that they can clean the bin on time and safeguard the environment. To avoid all such situations we intend to propose a solution for this problem "Smart Garbage Bin", which will alarm and inform the authorized person when the garbage bin is about to fill. Then message will be send to the authorized person to collect the garbage from the particular area. The authorized person will sends the message from his web application to the garbage collectors by sending a SMS .This system maintain a dry waste and a wet waste separately. This will help to reduce the overflow of the garbage bin and thus keeping the environment clean.

Purpose

This project helps the citizens to make their surroundings and environment clean , pollution free and lead a healthy life throughout. It avoids the possibility garbage overflow, unhygienic environment, air-borne and water-borne disease , etc...

2. LITERATURE SURVEY

EXISTING PROBLEM

In the existing system garbage is collected by the corporation weekly once or twice. Sometimes the garbage stinks and overflows from the bin and spread over the roads and pollutes the environment. This also produces a heavy air pollution and routes to various air-borne diseases. Many a times the street dogs and other animals eat these waste and scatter these waste around the surroundings which creates the spread of various diseases and situation of unclean environment.

Disadvantages of existing system:

- Time consuming and less effective.
- Overflow of waste from the bin.
- Unhygienic Environment and look of the city.
- Stinky smell and unpleasant situations.

PROPOSED SYSTEM :

In this proposed system there will be no issues repeated that of previous system. In this system the bin is designed in such a way that when the waste level reaches the threshold limit it

automatically closes the bin and intimates the alert to the admin . The bins are provided with low cost embedded device which helps in tracking the level of the garbage bins and a unique ID will be provided for every dustbin in the city .These details can be accessed by the concern authorities from their place with the help of internet and an immediate action can be made to clean the bin. The admin can monitor the level of the bin and can trace the location where it exists.

Advantages :

- Real time information on the fill level of the dustbin.
- Deployment of dustbin based on the actual needs.
- Cost Reduction and resource optimization.
- Improves Environment quality .

REFERENCES:

- [1] Ikuro Ihara; Nagaoka University of Technology; Ultrasonic Sensing: Fundamentals and Its Applications to Non-destructive Evaluation.
- [2] Arduino, "Available at <http://www.arduino.cc>," 2010.
- [3] M. Batty, "Smart Cities, Big Data," Environment and Planning B: Planning and Design 2012, vol. 39, pp. 191– 93.
- [4] Xu Li, Student Member, IEEE, Performance Evaluation of Vehicle-Based Mobile Sensor Networks for Traffic Monitoring.
- [5] Yusuf Abdullahi Badamasi, The Working Principle Of An Arduino, Electronics, Computer and Computation (ICECCO), 2014 11th International Conference on 29 Sept.-1 Oct. 2014.

Problem Statement

The waste management system provided earlier are not very reliable, efficient, cost effective and does not have any advanced processing features like automatic close of bin and alert intimations system .The following is a well articulated problem statement that allows you to find the ideal solution for the challenges faced.



Problem statement	I am (customer)	I'm trying to	But	Because	Which makes me feel
PS-1	People (smart dustbin)	Elimination of missed pickups	Irregular collection service	Refusal to learn about compliance	Environmental Degradation
PS-2	Smart dustbin	Sense all type of waste material it is in the form of solid or liquid	Requires of these type of sensors	Execution of new sensors for the problem	Uncleanness
PS-3	Tracker	Using container tracking your waste changes to smart pastures	Irregular collection coverage	Inappropriate Collection waste	Sanitary landfill and unhygienic condition

3. IDEATION & PROPOSED SOLUTION

EMPATHY MAP CANVAS



Ideation and Brainstorming

Brainstorm

Idea prioritization

Use this template in your own brainstorming sessions to your team. Use various idea-generation and start shaping concepts even if you're not sitting in the same room.

- 1. Problem to solve
- 2. Brainstorming
- 3. Prioritization

YOUR WORKSPACE

Step 1: Define the problem

Write a clear, concise statement of the problem you are trying to solve. This will be the focus of your brainstorming session.

Step 2: Brainstorming

Use the following prompts to generate ideas:

- What are the current challenges?
- What are the current solutions?
- What are the current users' needs?
- What are the current competitors' strengths and weaknesses?
- What are the current market trends?
- What are the current opportunities?
- What are the current threats?

Step 3: Prioritization

Use the following criteria to prioritize your ideas:

- Impact
- Feasibility
- Effort
- Time to market
- Resource requirements
- Risk
- Alignment with company goals

Step 4: Ideation

Use the following prompts to generate ideas:

- What are the current challenges?
- What are the current solutions?
- What are the current users' needs?
- What are the current competitors' strengths and weaknesses?
- What are the current market trends?
- What are the current opportunities?
- What are the current threats?

Step 5: Prioritization

Use the following criteria to prioritize your ideas:

- Impact
- Feasibility
- Effort
- Time to market
- Resource requirements
- Risk
- Alignment with company goals

Step 6: Ideation

Use the following prompts to generate ideas:

- What are the current challenges?
- What are the current solutions?
- What are the current users' needs?
- What are the current competitors' strengths and weaknesses?
- What are the current market trends?
- What are the current opportunities?
- What are the current threats?

Step 7: Prioritization

Use the following criteria to prioritize your ideas:

- Impact
- Feasibility
- Effort
- Time to market
- Resource requirements
- Risk
- Alignment with company goals

PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To maintain waste in the metropolitan cities and to maintain the clean environment which is polluted by the wastes and to dispose wastes.
2.	Idea / Solution description	-Software can be implemented -Fine can be implemented -Create awareness
3.	Novelty / Uniqueness	This system provides awareness among the people and safeguard the environment with the help of people.
4.	Social Impact / Customer Satisfaction	Clean cities Healthy environment Peaceful life
5.	Business Model (Revenue Model)	-Offering software as a service model to government. -Making use of useful wastes and making it us money.
6.	Scalability of the Solution	To help government to maintain clean environment.

PROBLEM SOLUTION FIT

<p>1. CUSTOMER SEGMENT(S) Who is your customer? i.e., working parents of 8-12 y.o. kids</p> <p>CS</p> <p>Define CS, fit into CC</p> <p>PEOPLE</p>	<p>6. CUSTOMER CONSTRAINTS What constraints prevent your customers from taking action or limit their choices of solutions? i.e., spending power, budget, no credit, network connectivity, availability of devices</p> <p>CC</p> <p>To avoid the non-degradable wastes</p>	<p>5. AVAILABLE SOLUTIONS Which solutions are available to the customer when they face the problem or need to get the job done? What have they tried in the past? What options come to their minds now? i.e., you and paper is an alternative to digital note taking</p> <p>AS</p> <p>Explore AS, differentiate</p> <p>1. Can create a software for monitoring wastages 2. Recycle the useful wastes</p>
<p>2. JOBS-TO-BE-DONE / PROBLEMS Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different angles.</p> <p>J&P</p> <p>Focus on J&P, map into BE, understand RC</p> <p>1. To avoid the use of Harmful wastes. 2. To protect the Environment.</p>	<p>9. PROBLEM ROOT CAUSE What is the real reason that this problem exists? What is the back story behind the need to do this job? i.e., customers have to do it because of the change in regulations.</p> <p>RC</p> <p>1. Due to lack of awareness. 2. Not taking responsibility in environmental issues by people.</p>	<p>7. BEHAVIOUR What do your customers do to solve the problem and get the job done? i.e., directly related: find the right software and install it, make it work and find indirectly associated activities: spend time on understanding software (i.e., 3 weeks)</p> <p>BE</p> <p>Focus on J&P, map into BE, understand RC</p> <p>Monitor the smart dustbin by using the software</p>
<p>3. TRIGGERS What triggers customers to act? i.e., seeing friends installing similar products, reading about a new app, client asking for the same.</p> <p>TR</p> <p>Identify strong TR & EM</p> <p>To create an awareness among the people</p> <p>4. EMOTIONS: BEFORE / AFTER How do customers feel when they face a problem or a job to be done? i.e., love, the sense of confidence, the control – use it in your communication strategy for sales.</p> <p>EM</p> <p>Safeguard the future generation</p>	<p>10. YOUR SOLUTION If you are working on an existing business, write down your current solution first. Fill in the gaps, and check for much fit to reality. If you are working on a new business proposal, then keep it blank until you fill in the canvas and come up with a solution that fits within customer need factors, address a problem or does the customer behaviour.</p> <p>SL</p> <p>1. Monitoring the wastage frequently 2. Harmful wastages must be banned</p>	<p>8. CHANNELS OF BEHAVIOUR ONLINE What kind of channels do customers use online? Extract until no channel is left!</p> <p>Uneducated people can not use this software</p> <p>BL2 OFFLINE What kind of activities do customers take offline? But not offline do similar to online? and is it a threat or an opportunity for your business?</p> <p>Due to human careless, waste disposal can be complicated</p> <p>Extract online & offline CH of BE</p>

4 . REQUIREMENT ANALYSIS

FUNCTIONAL REQUIREMENT

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Authentication	Collecting data from the transmitted signal
FR-2	IBM Watson IOT Platform	Stores the sensed data and alerts.
FR-3	Node RED	Designs the wireframing and connection of user interface
FR-4	Web User Interface	Created by Node RED service connected to IBM Watson IOT platform
FR-5	Database	Fetches data and updates it in the database
FR-6	Python script	Generates random data to the iot device and transmits to the Watson cloud.

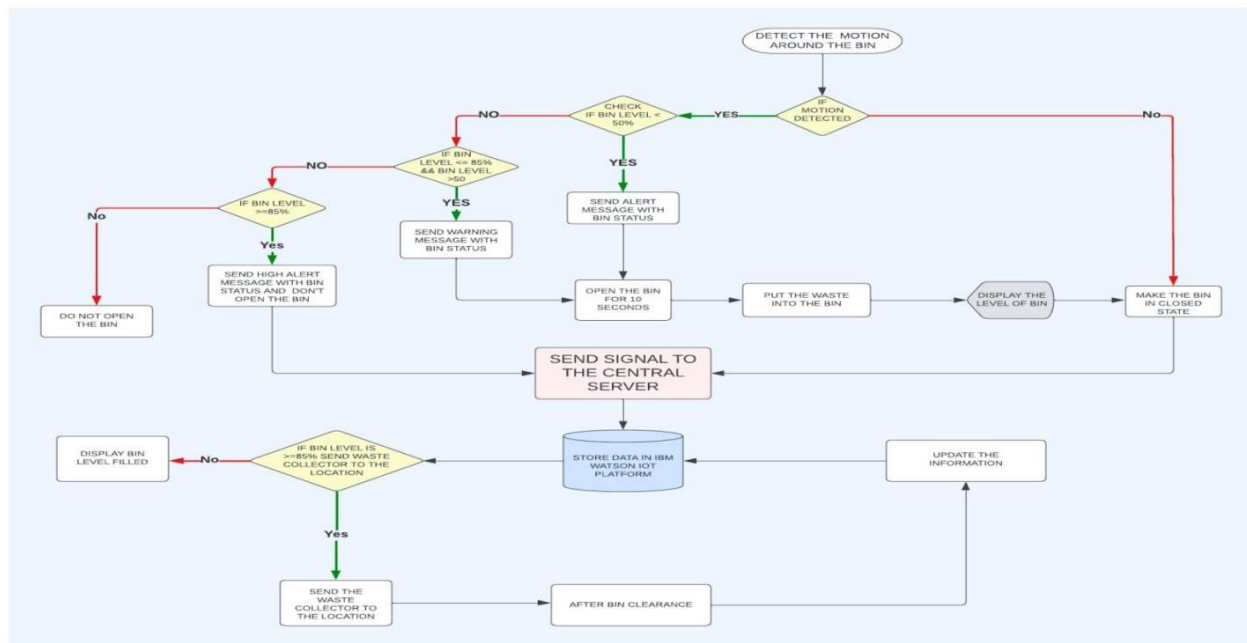
NON-FUNCTIONAL REQUIREMENT

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

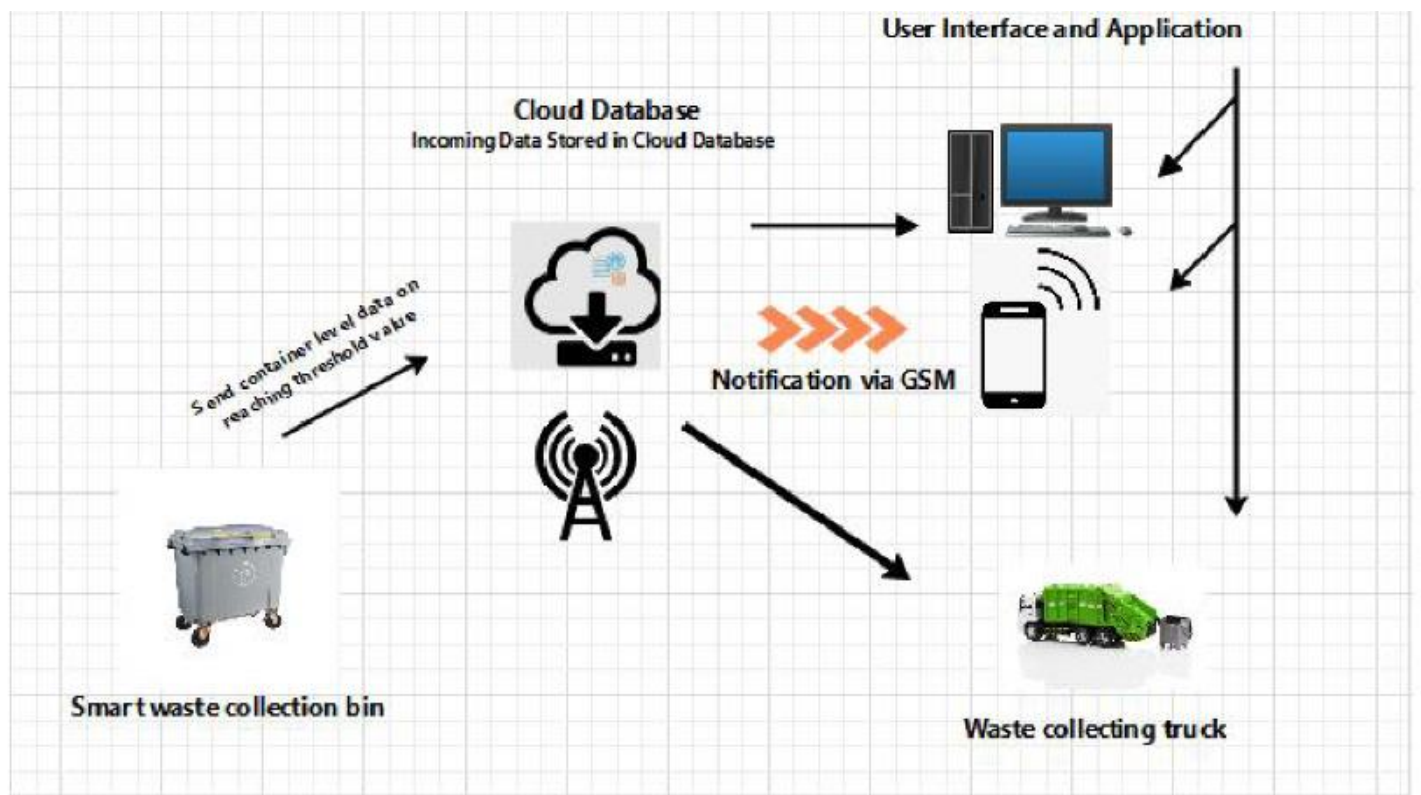
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Separate bins for recyclable and non
NFR-2	Security	Data fetched can be only accessed by authorized user
NFR-3	Reliability	Accurate data and availability is displayed
NFR-4	Performance	Detects and intimates alerts on reaching fixed limit.
NFR-5	Availability	Accessible through 24/7 by user and authorizer
NFR-6	Scalability	Holds vast accessibility by the user

5. PROJECT DESIGN

. DATA FLOW DIAGRAM



. SOLUTION ARCHITECTURE AND TECHNICAL ARCHITECTURE



USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user / Web user)	Signup	USN-1	User can signup using their email and password and confirm the details.	I can access my account / dashboard	High	Sprint-1
		USN-2	A confirmation mail is sent to the user.	I can receive confirmation email & click confirm	High	Sprint-1
	Login	USN-3	User can login using login credentials	User can log on to the website	High	Sprint-1
	Dashboard	USN-4	User can specify the location and area to check the availability of bins.	User can access dashboard and search for bins in specified areas	High	Sprint-2
		USN-5	User can post the queries and grievances in the report section	Options are provided to solve user issues	Medium	Sprint-2

6. PROJECT PLANNING AND SCHEDULING

. SPRINT PLANNING AND ESTIMATION

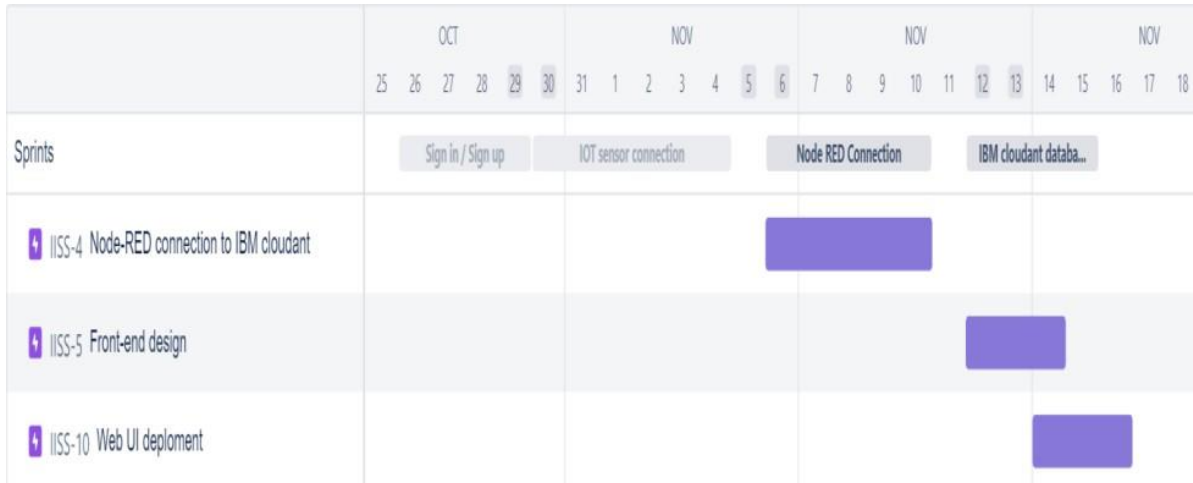
TITLE	DESCRIPTION
Literature Survey and Information Gathering	Surveying on the topic of selected project & gathering information by referring the, technical papers ,research publications etc.
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user pains & gains on particular issue.
Ideation	Jot down the ideas by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.
Proposed Solution	Prepare your proposed solution of the project which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.
Problem Solution Fit	Prepare problem - solution fit document.
Solution Architecture	Prepare solution architecture document.
Customer Journey Map	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit)
Functional Requirement	Prepare the functional requirement for the project.
Data Flow Diagrams	Draw the data flow diagrams to understand the flow of execution of the project.
Technology Architecture	Prepare the technology architecture diagram.
Milestone & Activity List	Prepare the milestones & activity list of the project.
Delivery of Sprints	Submit the coding development of the project and submit in sprints. Sprint -1 Sprint -2 Sprint -3 Sprint -4

SPRINT DELIVERY SCHEDULE

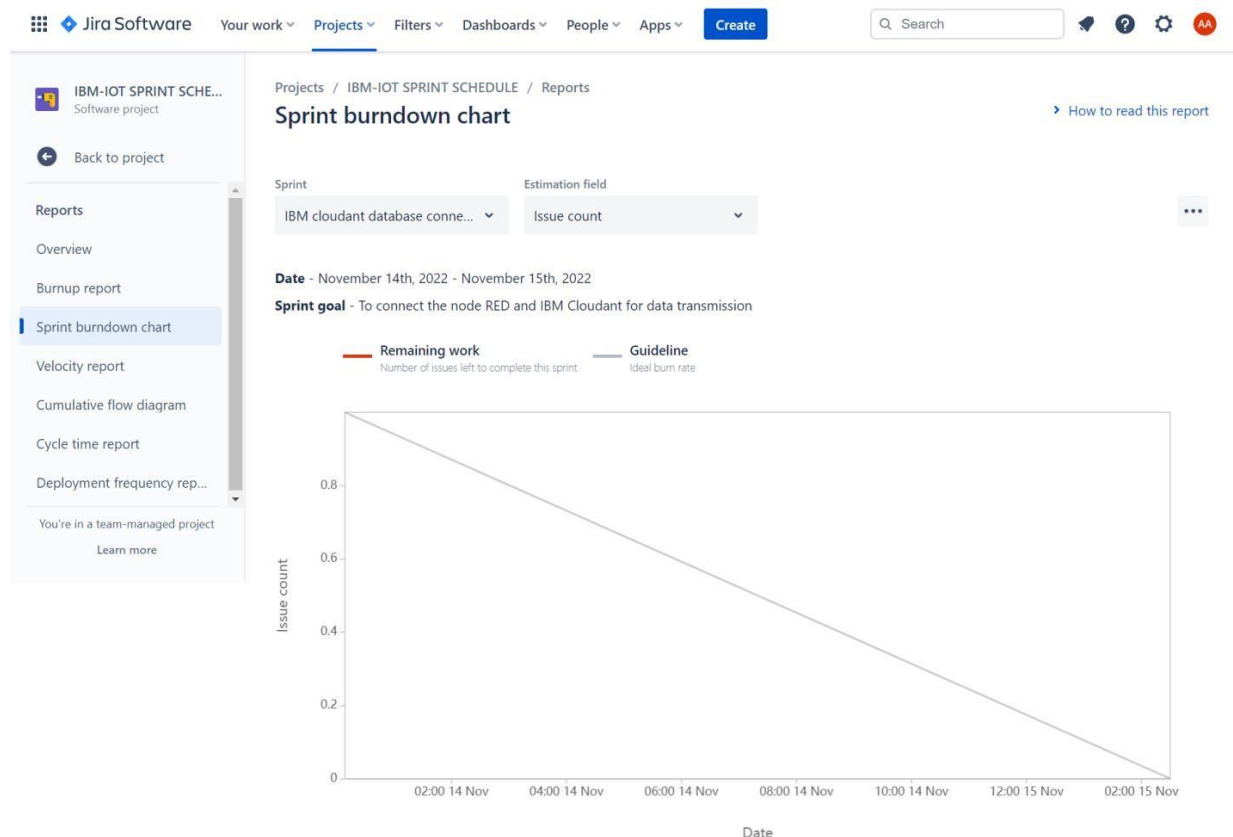
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Sign up / Sign in	USN - 1	User can signup using their email and password and confirm the details.	10	High
Sprint-1		USN - 2	A confirmation mail is sent to the user.	10	High
Sprint-2	Login	USN - 3	User can login using login credentials and is authenticated.	20	Low
Sprint-3	Dashboard	USN - 4	User can view the previous login activities of the account and updates.	10	Medium
Sprint - 4	Search Location	USN - 5	User can search for the bins available around the location.	10	High
Sprint - 4	Results / Grievances	USN - 6	User can post their grievances related to the bins and gets the results of bin status around the location from IBM Cloud.	10	High

JIRA REPORTS

ROADMAP



SPRINT BURNDOWN CHART



7. CODING AND SOLUTION

Wokwi code for Sensor transmission

```
#include <WiFi.h> // library for wifi
#include <PubSubClient.h> // library for MQTT
#include <LiquidCrystal_I2C.h>
#include <mjson.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);

//..... credentials of IBM Accounts .....

#define ORG "9gbe4w" // IBM organisation id
#define DEVICE_TYPE "SWMSMC" // Device type mentioned in ibm watson iot platform
#define DEVICE_ID "ibmproject" // Device ID mentioned in ibm watson iot platform
#define TOKEN "sUNA41tG6-Pq)0rk5X" // Token

//..... customise above values .....

char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // server name
char publishTopic[] = "iot-2/evt/data/fmt/json"; // topic name and type of event perform and format
in which data to be send
char topic[] = "iot-2/cmd/led/fmt/String"; // cmd Represent type and command is test format of
strings
char authMethod[] = "use-token-auth"; // authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //Client id

//.....

WiFiClient wifiClient; // creating instance for wificlient
PubSubClient client(server, 1883, wifiClient);

#define ECHO_PIN 12
#define TRIG_PIN 13
float dist;
String data3;
bool SealBin = true;
void setup()
{
  Serial.begin(115200);
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  //pir pin
  pinMode(34, INPUT);

  //ledpins
  pinMode(23, OUTPUT);
  pinMode(2, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(15, OUTPUT);

  lcd.init();
  lcd.backlight();
  lcd.setCursor(1, 0);
  lcd.print("");
  wifiConnect();
  mqttConnect();
}

float readcmCM()
{
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  int duration = pulseIn(ECHO_PIN, HIGH);
  return duration * 0.034 / 2;
}
```

```

void loop()
{

  lcd.clear();

  publishData();
  delay(500);
  if (!client.loop())
  {
    mqttConnect(); // function call to connect to IBM
  }
}

/* .....retrieving to cloud.....*/

void wifiConnect()
{
  Serial.print("Connecting to ");
  Serial.print("Wifi");
  WiFi.begin("Wokwi-GUEST", "", 6);
  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()
{
  float cm = readcmCM();

  if(digitalRead(34)) //pir motion detection
  {
    Serial.println("Motion Detected");
    Serial.println("Lid Opened");
    digitalWrite(15, HIGH);

    if(digitalRead(34)== true)
    {
      if(cm <= 100) //Bin level detection
      {
        digitalWrite(2, HIGH);
        Serial.println("High Alert!!!,Trash bin is about to be full");
      }
    }
  }
}

```

```

        Serial.println("Lid Closed");
        lcd.print("Full! Don't use");
        delay(2000);
        lcd.clear();
        digitalWrite(4, LOW);
        digitalWrite(23, LOW);
    }
    else if(cm > 100 && cm < 180)
    {
        digitalWrite(4, HIGH);
        Serial.println("Warning!!,Trash is about to cross 50% of bin level");
        digitalWrite(2, LOW);
        digitalWrite(23, LOW);

    }
    else if(cm > 180)
    {
        digitalWrite(23, HIGH);
        Serial.println("Bin is available");
        digitalWrite(2,LOW);
        digitalWrite(4, LOW);

    }

    delay(10000);
    Serial.println("Lid Closed");
}
else
{
    Serial.println("No motion detected");
    digitalWrite(2, LOW);
    digitalWrite(15, LOW);
    digitalWrite(4, LOW);
    digitalWrite(23, LOW);
}

}

else
{
    digitalWrite(15, LOW);

}

}

if(cm <= 100)
{
    digitalWrite(21,HIGH);
    String payload = "{\"High_Alert\":\"";
    payload += cm;
    payload += " }";
    Serial.print("\n");
    Serial.print("Sending payload: ");
    Serial.println(payload);

    if (client.publish(publishTopic, (char*) payload.c_str())) // if data is uploaded to cloud successfully,prints publish ok
    else prints publish failed
    {
        Serial.println("Publish OK");
    }
}
else if(cm <= 180)
{
    digitalWrite(22,HIGH);
    String payload = "{\"Warning\":\"";
    payload += cm ;
    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");
}
}
else if(cm > 180)
{
    digitalWrite(23,HIGH);
    String payload = "{";
    payload += cm;
    payload += " }";
    Serial.print("\n");
    Serial.print("Sending payload: ");
    Serial.println(payload);

    if (client.publish(publishTopic, (char*) payload.c_str())) // if data is uploaded to cloud successfully,prints publish ok
    else prints publish failed
    {
        Serial.println("Publish OK");
    }
}

float inches = (cm / 2.54); //print on lcd
lcd.setCursor(0,0);
lcd.print("Inches");
lcd.setCursor(4,0);
lcd.setCursor(12,0);
lcd.print("cm");
lcd.setCursor(1,1);
lcd.print(inches, 1);
lcd.setCursor(11,1);
lcd.print(cm, 1);
lcd.setCursor(14,1);
delay(1000);
lcd.clear();
}

//handles commands from user side

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++) {

        data3 += (char)payload[i];
    }
    Serial.println("data: "+ data3);

    const char *s =(char*) data3.c_str();
    double pincode = 0;

    const char *buf;
    int len;

    if (mjson_find(s, strlen(s), "$.command", &buf, &len)) // And print it
    {
        String command(buf,len);

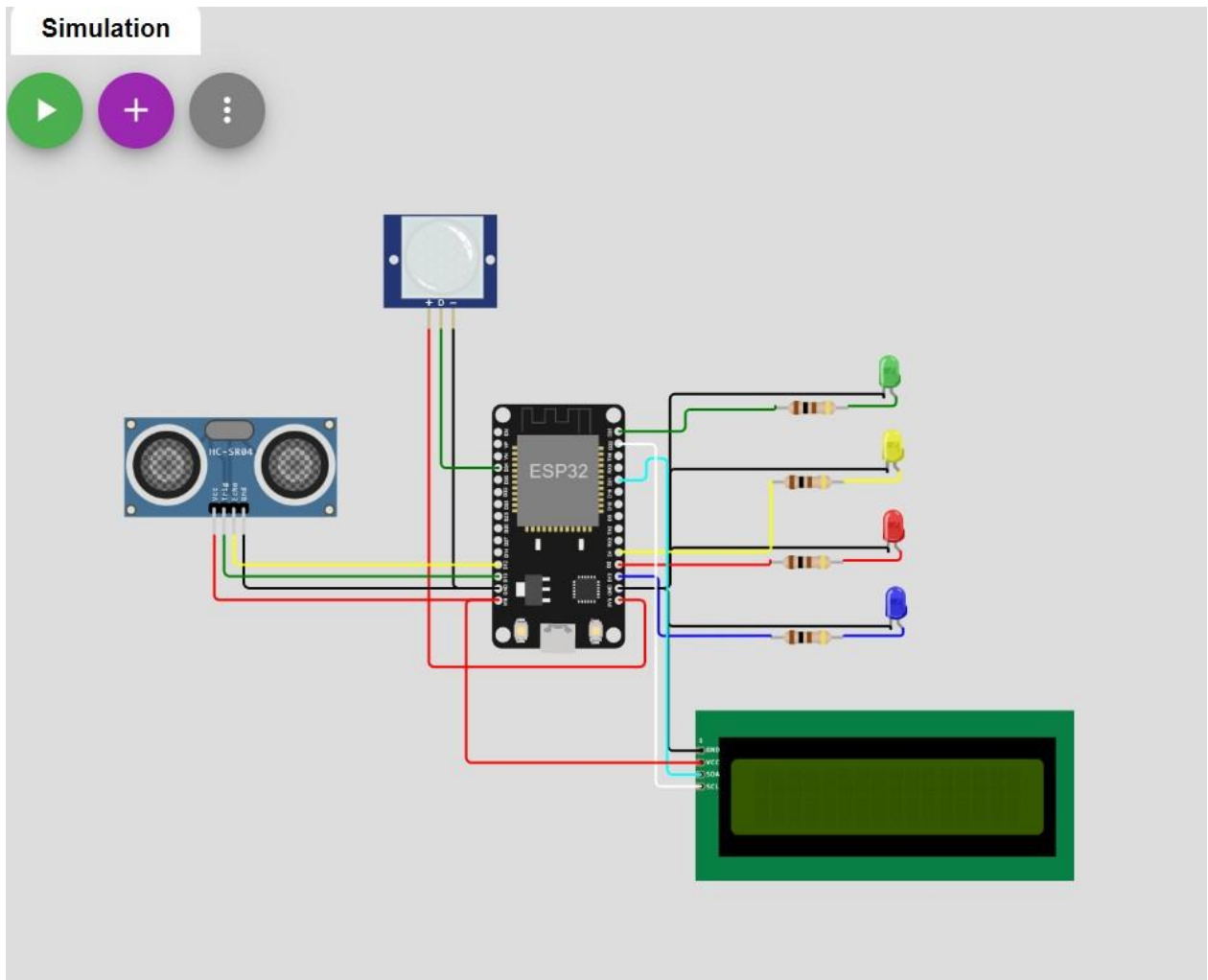
        if(command=="\"SealBin\"")
        {
            SealBin = true;
        }

    }

    data3="";
}

```

. Sensor Connection Setup



PHYSICAL COMPONENTS:

- PIR MOTION SENSOR
- ULTRASONIC DISTANCE SENSOR
- ESP32-ARDUINO MICROCONTROLLER

OUTPUT:

WOKWI SETUP

WOKWI

SAVE SHARE ESP32-IBMconnection[original] Docs

esp32-blink.ino diagram.json libraries.txt Library Manager

```

1 #include <WiFi.h> // library for wifi
2 #include <PubSubClient.h> // library for MQTT
3 #include <LiquidCrystal_I2C.h>
4 LiquidCrystal_I2C lcd(0x27, 20, 4);
5
6 //----- credentials of IBM Accounts -----
7
8 #define ORG "9gbe4w" // IBM organisation id
9 #define DEVICE_TYPE "SWMSMC" // Device type mentioned in
10 #define DEVICE_ID "ibmproject" // Device ID mentioned in ib
11 #define TOKEN "sUNA41tG6-Pq)0rk5X" // Token
12
13 //----- customise above values -----
14
15 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
16 char publishTopic[] = "iot-2/evt/data/fmt/json";
17 char topic[] = "iot-2/cmd/led/fmt/String";
18 char authMethod[] = "use-token-auth";
19 char token[] = TOKEN;
20 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
21
22 //-----
23
24 WiFiClient wificlient;
25 PubSubClient client(server, 1883, wificlient);
26
27 #define ECHO_PIN 12

```

Simulation

00:10.649 61%

No motion detected

Sending payload: {"High Alert!!":"52.96left" }

Publish OK

Sending distance: 52.96

Publish OK

WEB UI

node-red-wjldy-2022-11-05-syd.mybluemix.net/ui/#/0?socketid=DIVoouriSJNRRVU-AAAJ

Log

Control

Bin in use

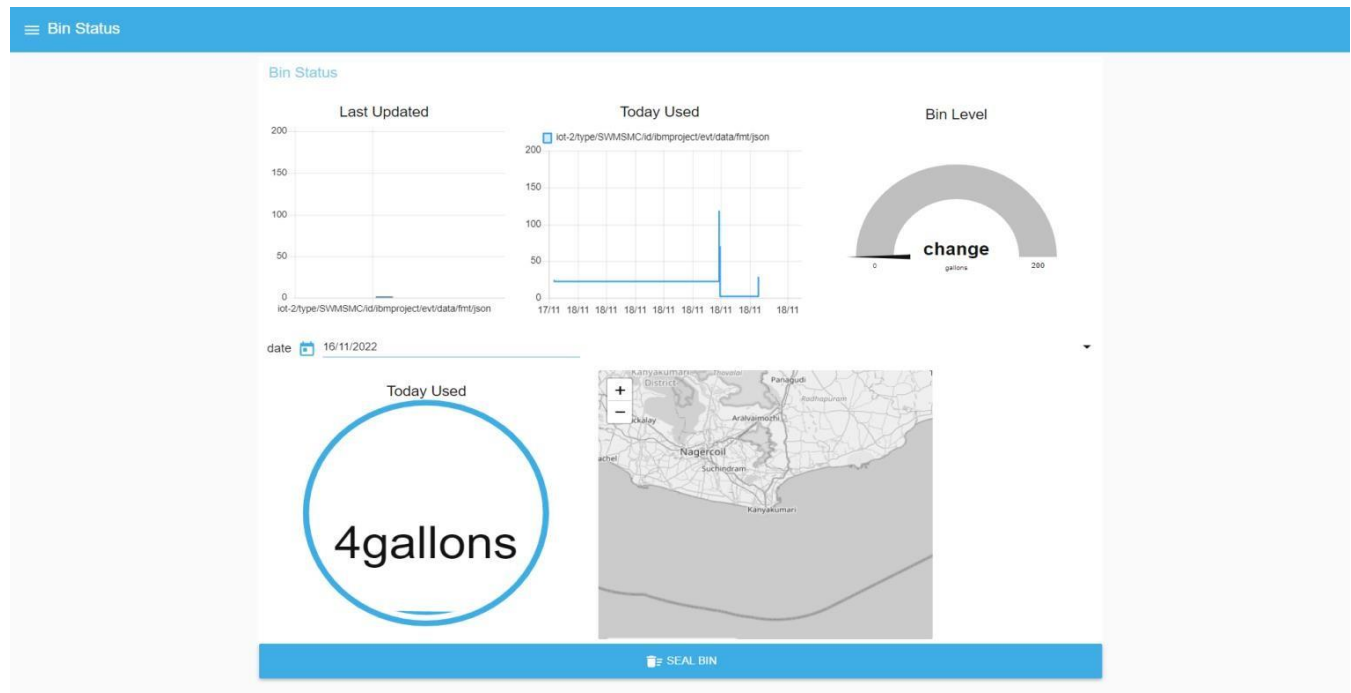
Range

Today Used 0

Enter Location
change

date 15/11/2022

VIEW STATUS



The admin gets notification when the bin detects motion and if the bin level crosses 50 percent it indicates warning and if it crosses 90 percent it gives a High alert and closes the bin. If the admin wants to seal the bin the admin can command seal bin until it is accessed for cleaning.

Test Case:

Maximum Size of Bin : 200 cm

Safe limit: below 100 cm

Minimum threshold limit of bin: 100 cm

Maximum threshold limit of bin: 180 cm

S.no	Bin Level (cm filled)	Bin Status	Location
1	45	Safe	Kanyakumari
2	78	Safe	Coimbatore
3	112	Warning	Trichy
4	169	Warning	Chennai
5	186	Warning	Ooty
6	193	High_Alert	Tirunelveli
8	0	Safe	Chengalpattu
9	35	Safe	Madurai
10	101	Warning	Salem
11	132	Warning	Thanjavore
12	158	Warning	Vellore
13	93	High_Alert	Erode
14	93	High_Alert	Karur
15	93	High_Alert	Cuddalore
16	30	Safe	Kumbakonam
17	110	Warning	Ambur
18	180	Warning	Sivakasi
19	195	High_Alert	Neyveli
20	80	Safe	Krishnagiri

Note: The bin location provided above is default. When the user access the bin , the location and status of the bin displayed to the admin.

USER ACCEPTANCE TESTING

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Smart Waste Management System project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	3	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	78

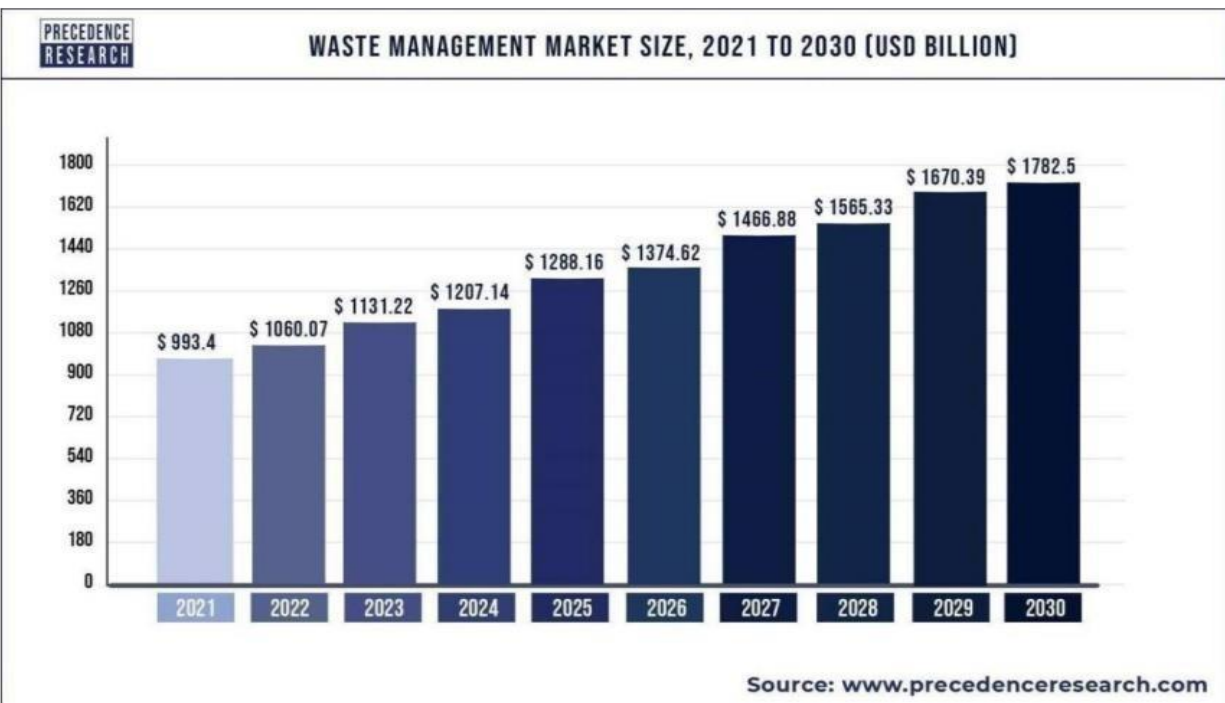
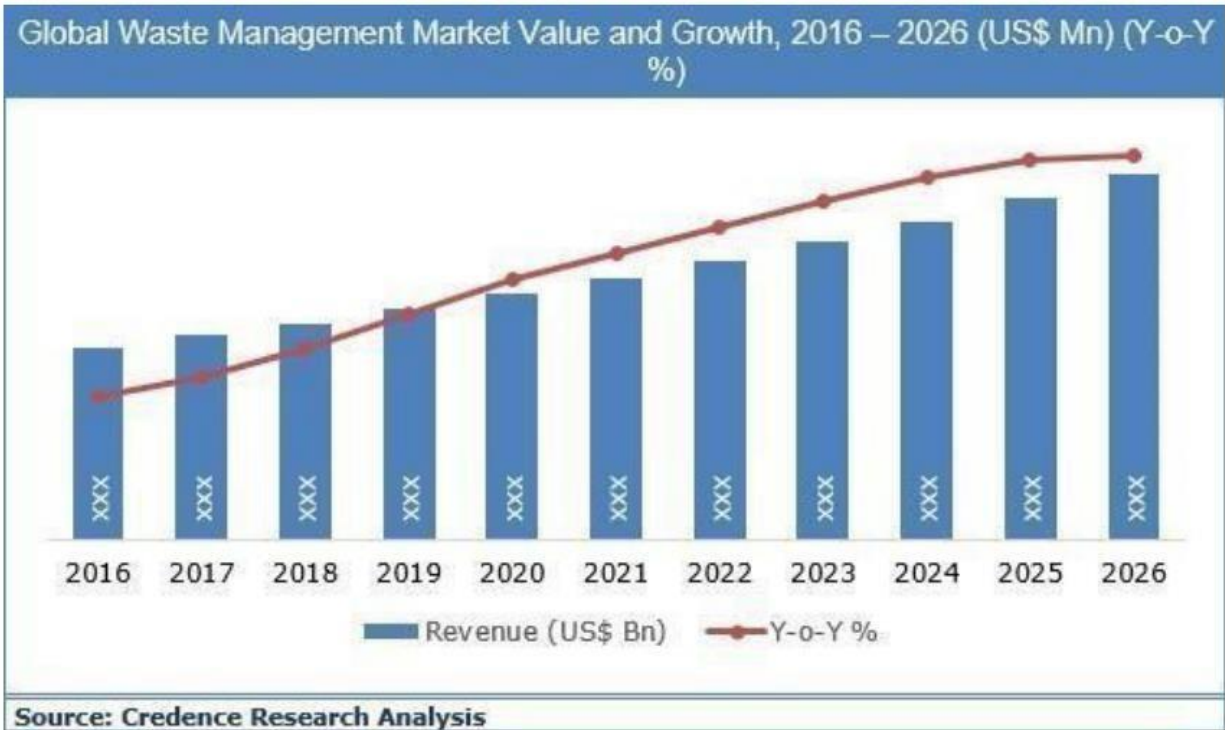
3. TEST CASE ANALYSIS

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

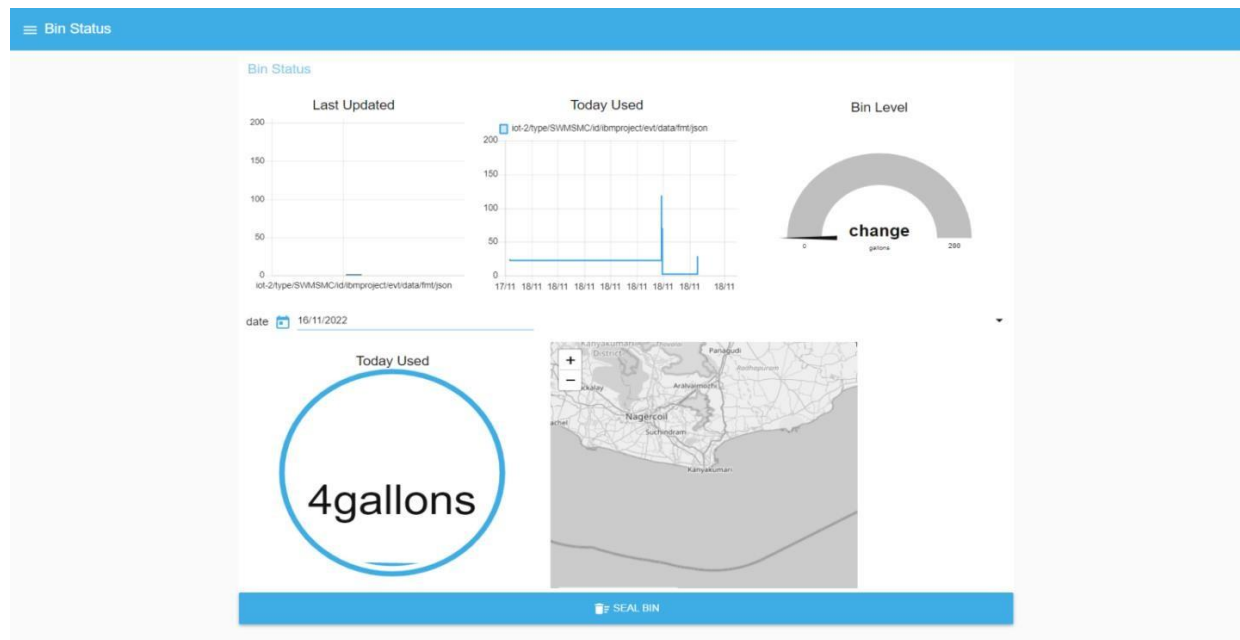
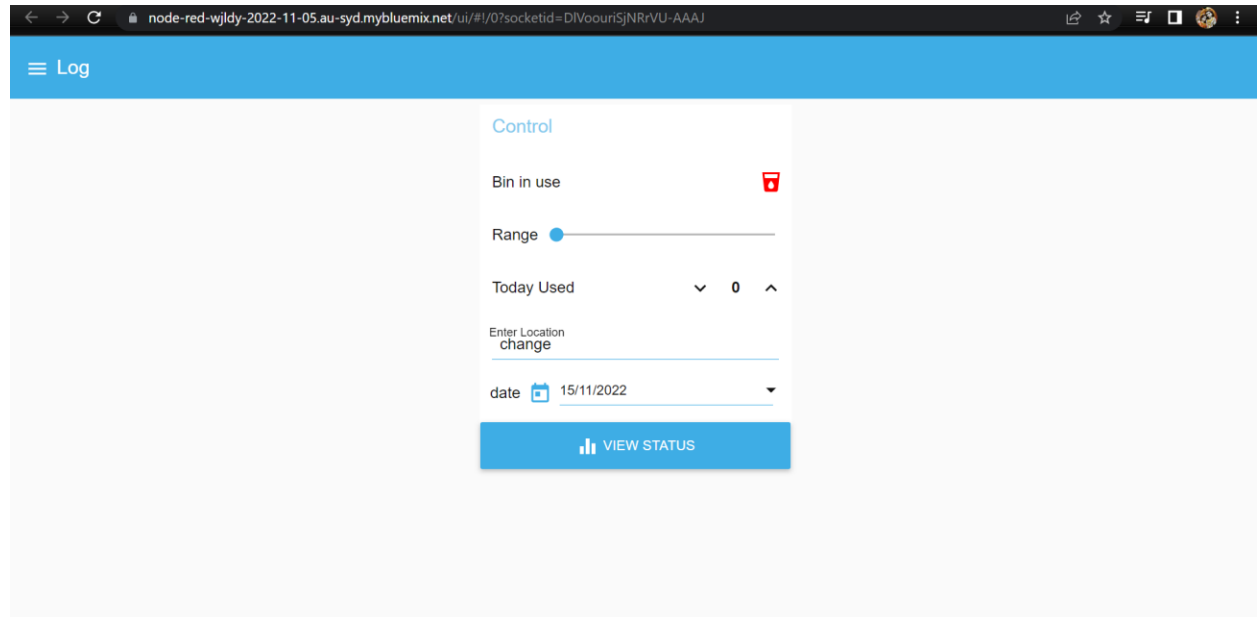
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9. RESULTS

Performance Metrics



. Admin Web UI



10. ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- Reduction in Collection Cost
- No Missed Pickups
- Reduced Overflows
- Waste Generation Analysis
- CO2 Emission Reduction

DISADVANTAGES

- System requires a greater number of waste bins for separate waste collection as per population in the city.
- This results into high initial cost due to expensive smart dustbins compare to other methods. Sensor nodes used in the dustbins have limited memory size.

11. CONCLUSION :

A Smart Waste Management system that is more effective than the one in use now is achievable by using sensors to monitor the filling of bins. Our conception of a "smart waste management system" focuses on monitoring waste management, offering intelligent technology for waste systems, eliminating human intervention, minimizing human time and effort, and producing a healthy and trash-free environment. The suggested approach can be implemented in smart cities where residents have busy schedules that provide little time for garbage management. If desired, the bins might be put into place in a metropolis where a sizable container would be able to hold enough solid trash for a single unit. But these may price bit high.

12. FUTURE SCOPE:

There are several future works and improvements for the proposed system, including the following:

- Changes the system of user authentication and atomic lock of bins, which would aid in protecting the bin from damage or theft.

- The concept of green points would encourage the involvement of residents or end users, making the idea successful and aiding in the achievement of collaborative waste management efforts, thus fulfilling the idea of 'Swachh Bharath'.
- Having case study or data analytics on the type and times waste is collected on different days or seasons, making the bin level predictable and remove the reliance on electronic components, and fixing the coordinates.
- Improving the Server's and Android's graphical interfaces

13. APPENDIX

- **Esp32 - Microcontroller :**

ESP32 is a series of low-cost, low-power system on a chip microcontrollers with integrated Wi-Fi and dual-mode Bluetooth.

- Memory: 320 KiB
- SRAM CPU: Tensilica Xtensa LX6 microprocessor @ 160 or 240 MHz
- Power: 3.3 V DC
- Manufacturer: Espressif Systems
- Predecessor: ESP8266

- **Sensors :**

- PIR motion sensor: PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range.
- Ultrasonic Distance Sensor : Ultrasonic Sensors measure the distance to the target by measuring the time between the emission and reception.

. Source code

```
#include <WiFi.h> // library for wifi
#include <PubSubClient.h> // library for MQTT
#include <LiquidCrystal_I2C.h>
#include <mjson.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);

//..... credentials of IBM Accounts .....

#define ORG "9gbe4w" // IBM organisation id
#define DEVICE_TYPE "SWMSMC" // Device type mentioned in ibm watson iot platform
#define DEVICE_ID "ibmproject" // Device ID mentioned in ibm watson iot platform
#define TOKEN "sUNA41tG6-Pq)0rk5X" // Token

//..... customise above values .....

char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // server name
char publishTopic[] = "iot-2/evt/data/fmt/json";
// topic name and type of event perform and format in which data to be send
char topic[] = "iot-2/cmd/led/fmt/String";
// cmd Represent type and command is test format of strings
char authMethod[] = "use-token-auth";
// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //Client id

//.....

WiFiClient wifiClient; // creating instance for wificlient
PubSubClient client(server, 1883, wifiClient);

#define ECHO_PIN 12
#define TRIG_PIN 13
float dist;
String data3;
bool SealBin = true;
void setup()
{
  Serial.begin(115200);
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  //pir pin
  pinMode(34, INPUT);

  //ledpins
  pinMode(23, OUTPUT);
  pinMode(2, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(15, OUTPUT);

  lcd.init();
  lcd.backlight();
  lcd.setCursor(1, 0);
  lcd.print("");
  wifiConnect();
  mqttConnect();
}

float readcmCM()
{
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  int duration = pulseIn(ECHO_PIN, HIGH);
  return duration * 0.034 / 2;
}

void loop()
```

```

{
    lcd.clear();

    publishData();
    delay(500);
    if (!client.loop())
    {
        mqttConnect(); // function call to connect to IBM
    }
}

/* .....retrieving to cloud.....*/

void wifiConnect()
{
    Serial.print("Connecting to ");
    Serial.print("Wifi");
    WiFi.begin("Wokwi-GUEST", "", 6);
    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()
{
    float cm = readcmCM();

    if(digitalRead(34) //pir motion detection
    {
        Serial.println("Motion Detected");
        Serial.println("Lid Opened");
        digitalWrite(15, HIGH);

        if(digitalRead(34)== true)
        {
            if(cm <= 100) //Bin level detection
            {
                digitalWrite(2, HIGH);
                Serial.println("High Alert!!!,Trash bin is about to be full");
                Serial.println("Lid Closed");
                lcd.print("Full! Don't use");
                delay(2000);
            }
        }
    }
}

```

```

        lcd.clear();
        digitalWrite(4, LOW);
        digitalWrite(23, LOW);
    }
    else if(cm > 100 && cm < 180)
    {
        digitalWrite(4, HIGH);
        Serial.println("Warning!!,Trash is about to cross 50% of bin level");
        digitalWrite(2, LOW);
        digitalWrite(23, LOW);

    }
    else if(cm > 180)
    {
        digitalWrite(23, HIGH);
        Serial.println("Bin is available");
        digitalWrite(2,LOW);
        digitalWrite(4, LOW);

    }
    delay(10000);
    Serial.println("Lid Closed");
}
else
{
    Serial.println("No motion detected");
    digitalWrite(2, LOW);
    digitalWrite(15, LOW);
    digitalWrite(4, LOW);
    digitalWrite(23, LOW);
}

}
else
{
    digitalWrite(15, LOW);

}

if(cm <= 100)
{
    digitalWrite(21,HIGH);
    String payload = "{\"High_Alert\":\":";
    payload += cm;
    payload += " }";
    Serial.print("\n");
    Serial.print("Sending payload: ");
    Serial.println(payload);

    if (client.publish(publishTopic, (char*) payload.c_str())) // if data is uploaded to cloud successfully,prints publish ok
    else prints publish failed
    {
        Serial.println("Publish OK");
    }
}
else if(cm <= 180)
{
    digitalWrite(22,HIGH);
    String payload = "{\"Warning\":\":";
    payload += cm ;
    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");
    }
}

```



```

}
else if(cm > 180)
{
digitalWrite(23,HIGH);
String payload = "{";
payload += cm;
payload += " }";
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload);

if (client.publish(publishTopic, (char*) payload.c_str())) // if data is uploaded to cloud successfully,prints publish ok
else prints publish failed
{
Serial.println("Publish OK");
}

}

float inches = (cm / 2.54); //print on lcd
lcd.setCursor(0,0);
lcd.print("Inches");
lcd.setCursor(4,0);
lcd.setCursor(12,0);
lcd.print("cm");
lcd.setCursor(1,1);
lcd.print(inches, 1);
lcd.setCursor(11,1);
lcd.print(cm, 1);
lcd.setCursor(14,1);
delay(1000);
lcd.clear();
}

//handles commands from user side

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++) {
data3 += (char)payload[i];
}
Serial.println("data: "+ data3);

const char *s=(char*) data3.c_str();
double pincode = 0;

const char *buf;
int len;

if (mjson_find(s, strlen(s), "$.command", &buf, &len)) // And print it
{
String command(buf,len);

if(command=="\SealBin\"){
SealBin = true;
}

}

data3="";
}

```

. GITHUB LINK

LINK : <https://github.com/IBM-EPBL/IBM-Project-39740-1660494184>

VIDEO DEMO LINK: https://drive.google.com/drive/folders/1DF_l6rklGaHQnNWV_QZP3mCag5S1S82u