



Smart Waste Management System For Metropolitan Cities

A PROJECT REPORT

Submitted by

SENTHIL KUMAR S	(721819106007)
SATHYAKANTH R	(721819106006)
SANKAR R	(721819106005)
DINESH KUMAR M	(721819106003)

***in the partial fulfilment of the requirements for the award of a
degree of***

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

RATHINAM TECHNICAL CAMPUS, COIMBATORE.

ANNA UNIVERSITY : CHENNAI 600 025

CONTENT

1. **INTRODUCTION**
 - 1.1 Project Overview
 - 1.2 Purpose
2. **LITERATURE SURVEY**
 - 2.1 Existing problem
 - 2.2 References
 - 2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
 - 3.1 Empathy Map Canvas
 - 3.2 Ideation & Brainstorming
 - 3.3 Proposed Solution
 - 3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
 - 4.1 Functional requirement
 - 4.2 Non-Functional requirements
5. **PROJECT DESIGN**
 - 5.1 Data Flow Diagrams
 - 5.2 Solution & Technical Architecture
 - 5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
 - 6.1 Sprint Planning & Estimation
 - 6.2 Sprint Delivery Schedule
 - 6.3 Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
 - 7.1 Feature 1
 - 7.2 Feature 2
 - 7.3 Database Schema (if Applicable)
8. **TESTING**
 - 8.1 Test Cases
 - 8.2 User Acceptance Testing
9. **RESULTS**
 - 9.1 Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**
 - Source Code
 - GitHub 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.

1.1. 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.

1.2. 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

2.1 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 .

2.2. REFERENCES:

[1] Ikuo Ihara; Nagaoka University of Technology; Ultrasonic Sensing: Fundamentals and Its Applications to Non-destructive Evaluation.

[2] Arduino, "Available at <http://www.arduino.cc>," 2010.

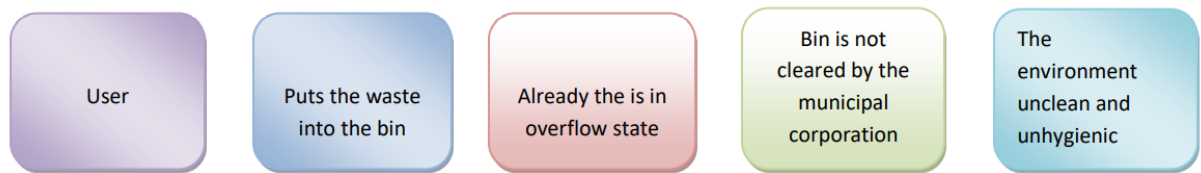
[3] M. Batty, "Smart Cities, Big Data," Environment and Planning B: Planning and Design 2012, vol. 39, pp. 191– 93.

[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.

2.3. 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 (PS)	I'm (User)	I'm trying to	But	Because	Which makes me feel
PS - 1	Colony resident member	Put the waste into the waste bin	bin is already in the state of overflow	bin is not cleared by the corporation cleaners	So disgusting to see the spoiled area that stays as a major reason for many air-borne diseases.
PS - 2	Street Walkers	Throw the waste into the bin	There is no waste bin at all	Bin is not provided for the area	The main reason to make the surrounding unclean and unhygienic.

3. IDEATION & PROPOSED SOLUTION

3.1. EMPATHY MAP CANVAS



3.2. Ideation and Brainstorming

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions as your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
15 minutes to collaborate
15 minutes to recapitulate

Before you collaborate

A little bit of preparation goes a long way with this exercise. Here's what you need to do to get going.

1. **Team ID:** PNT2022TMD15915

2. **Team ID:** PNT2022TMD15915

3. **Team ID:** PNT2022TMD15915

Define your problem statement

What problem are you trying to solve? Frame your problem as a single, clear statement. This will be the focus of your brainstorm.

Brainstorm

Write down any ideas that come to mind that address your problem statement.

SENTHIL KUMAR S

The proposed system will allow users to monitor the solid waste management process and management of the central collection process using IOT (Internet of Things)

SANKAR

Load cell on bottom of bins

when bins fill alert message to the authorized person

SATHYAKANTH

Placing Ultrasonic sensor to detect level of bins

Enable GPS function to locate bins easier

Waste generation analysis to understand cities usages

DINESH KUMAR

Place Arduino board at left side of bins

Visual fill status indicators on top of bins

using by GSM in bins achieve wireless communication with bins and managing center

solar panels for power supply for IOT devices

Group ideas

Take turns sharing your ideas while discussing similar or related topics as you go. Once all ideas have been shared, give each idea a number (1-10) based on how important and useful it is to you. Try to give ideas a number 1-10 based on how important and useful it is to you.

10 ideas

Smart garbage maintenance server

Collect only degradable and non-degradable wastes

Transparency and sustainable solution than normal garbage bins

IOT alert authorized person when bins going to fill

Optimized trash collection route

Prioritize

Your team should all be on the same page about what's important, what's feasible. Place your ideas on the grid to determine which ideas are important and which are feasible.

10 ideas

Waste generation analysis to understand cities usages

Raspberry-pi with ultrasonic, GPS, Load cell, are configured

After you collaborate

You can export the final list of ideas or print it to share with members of your company who might find it helpful.

Quick solution

1. **Share the need**
Share a need that is important to your team. This will be the focus of your brainstorm.

2. **Brainstorm**
Brainstorm ideas that address the need. Write them down on sticky notes or index cards.

3. **Group ideas**
Take turns sharing your ideas while discussing similar or related topics as you go. Once all ideas have been shared, give each idea a number (1-10) based on how important and useful it is to you. Try to give ideas a number 1-10 based on how important and useful it is to you.

4. **Prioritize**
Place your ideas on the grid to determine which ideas are important and which are feasible.

5. **Export the final list**
You can export the final list of ideas or print it to share with members of your company who might find it helpful.

3.3. PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	<ul style="list-style-type: none">➤ Ravage and waste can cause environmental pollution.➤ Rotting garbage is the cause to produce harmful gases in the air and cause breathing problem to people.➤ Due to improper waste disposal, we may face several problems like unpleasant odour and severe health issues.
2.	Idea / Solution description	<ul style="list-style-type: none">➤ To solve this problem of waste disposal , we can design a smart garbage bin with inbuilt sensors and IOT devices such like Arduino UNO , Raspberry pi, etc....➤ Garbage level and location intimation mechanism .➤ AI Recycling Robots.
3.	Novelty / Uniqueness	<ul style="list-style-type: none">➤ Identify potential waste streams.➤ Create a waste management-focused community outreach plane.
4.	Social Impact / Customer Satisfaction	<ul style="list-style-type: none">➤ Neighbourhood of landfills to communities, breeding of pests and loss in property values.➤ The IOT solution uses the data and intimates the information to the local area of management department when it reaches the initial boundary line .
5.	Business Model (Revenue Model)	<ul style="list-style-type: none">➤ It generates revenue through the provision of various waste management and disposal services.➤ Recycling solutions to residential , commercial , industrial, and municipal clients.
6.	Scalability of the Solution	<ul style="list-style-type: none">➤ Installing separate bins for collecting recyclable and non-recyclable wastes.➤ Recycling not only save energy but also prevent the material from going to landfills & Incineration and provides raw materials for new products.

3.4. PROBLEM SOLUTION FIT

Project Title: Smart Waste Management For Metropolitan Cities

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID43771

Define CS, fit into CC	1. CUSTOMER SEGMENT(S) CS <p>Each and every individual in the society</p>	6. CUSTOMER CONSTRAINTS CC <p>Since metropolitan cities covers large area, implementation of this project requires more time and each and every individual should work for it. Also we are lack of development of technology in this field is a major issue.</p>	5. AVAILABLE SOLUTIONS AS <p>The garbage level in dustbins is monitored continuously monitored and once if the dustbin gets filled it automatically getslocked and the message will be sent to the municipality workers. This will be the best solution for this problem.</p>	Explore AS, differentiate
Focus on J&P, map into BE, understand RC	2. JOBS-TO-BE-DONE / PROBLEMS J&P <p>There are several issues faced by the residents of the flats. One of them is the disposal of solid waste. This is due to high housing demands which have drastically risen because of migration from villages to cities to find work.</p>	9. PROBLEM ROOT CAUSE RC <p>The main reason for this problem is increase in population in metropolitan cities. This is mainly due to development of the technology and people started to migrate form rural areas to urban due to unemployment.</p>	7. BEHAVIOUR BE <p>They should inform the municipality workers about the problem or they should voluntarily involve in waste management activities.</p>	Focus on J&P, map into BE, understand RC
Identify strong TR & EM	3. TRIGGERS TR <p>They get triggered after seeing the people who are affected with some diseases like cholera, typhoid andsome skin diseases due to the improper handling of waste. And also by seeing awareness programs about waste management.</p>	10. YOUR SOLUTION SL <p>Our solution for this problem is the garbage level of the dustbin is continuously monitored by the sensor and once if the dustbin get</p>	8. CHANNELS of BEHAVIOUR CH <p>Online: Advertising about the issue in internet, television, and creating some postures about the impact of the problem.</p>	Identify strong TR & EM
4. EMOTIONS: BEFORE / AFTER EM <p>Before: They get affected with some air borne diseases and the city is also more polluted.</p> <p>After: But it helps them to diagnose about the problem in early stage and they took some effective measures in waste management to avoid the problem.</p>		<p>filled, it automatically gets locked. Simultaneously message will be sentto the respective municipality worker andthen he can dispose the waste effectively.</p>	<p>Offline: Organising a campaign and awareness program in public places and schools.</p>	

4 . 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 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.

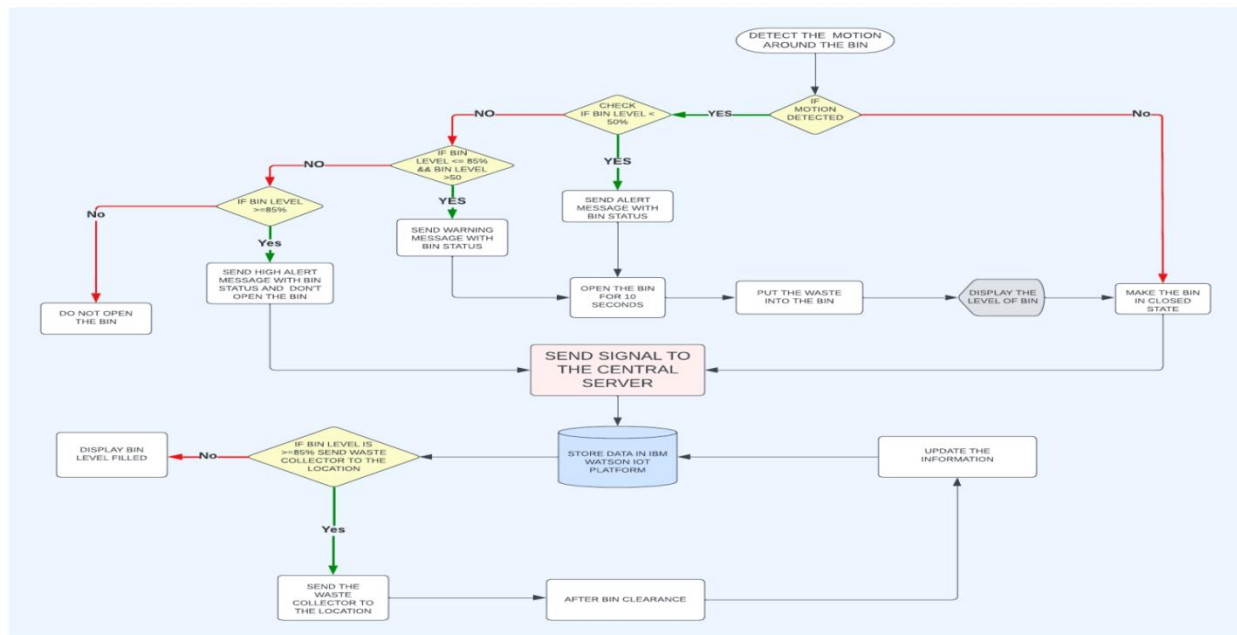
4.2 NON-FUNCTIONAL REQUIREMENT

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

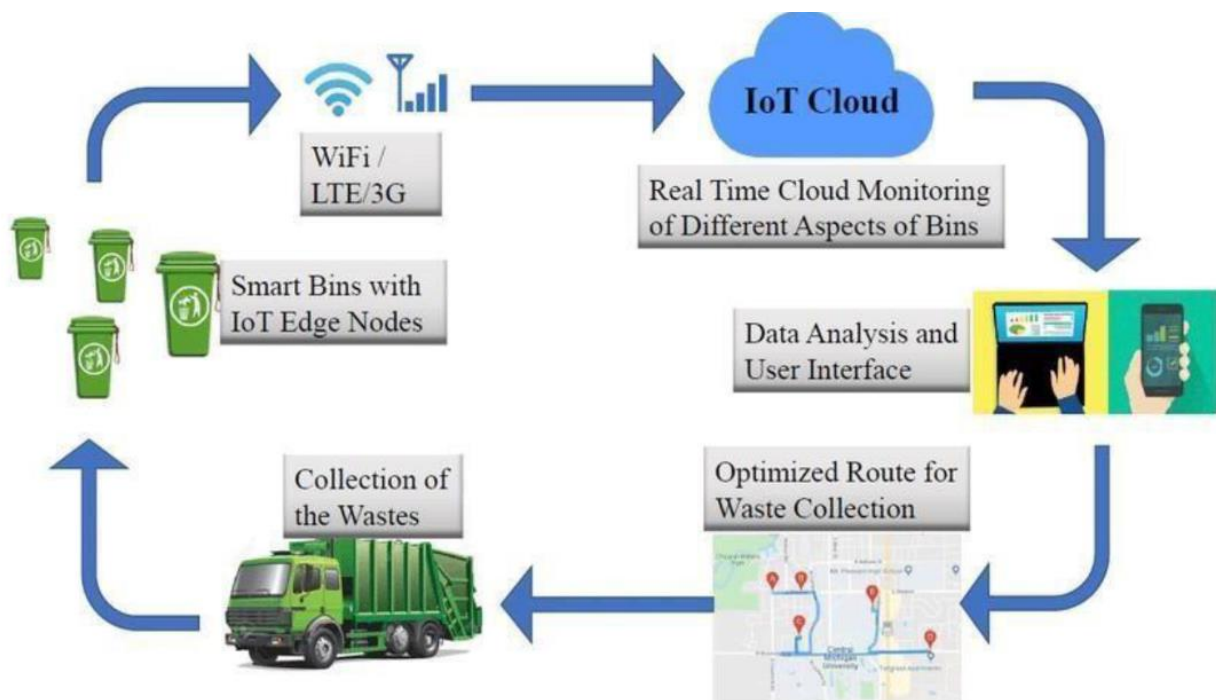
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Separate bins for recyclable and non
NFR-2	Security	Data fetched can be only accessed by authorized user
NFR-3	Reliability	Accurate data and availability is displayed
NFR-4	Performance	Detects and intimates alerts on reaching fixed limit.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

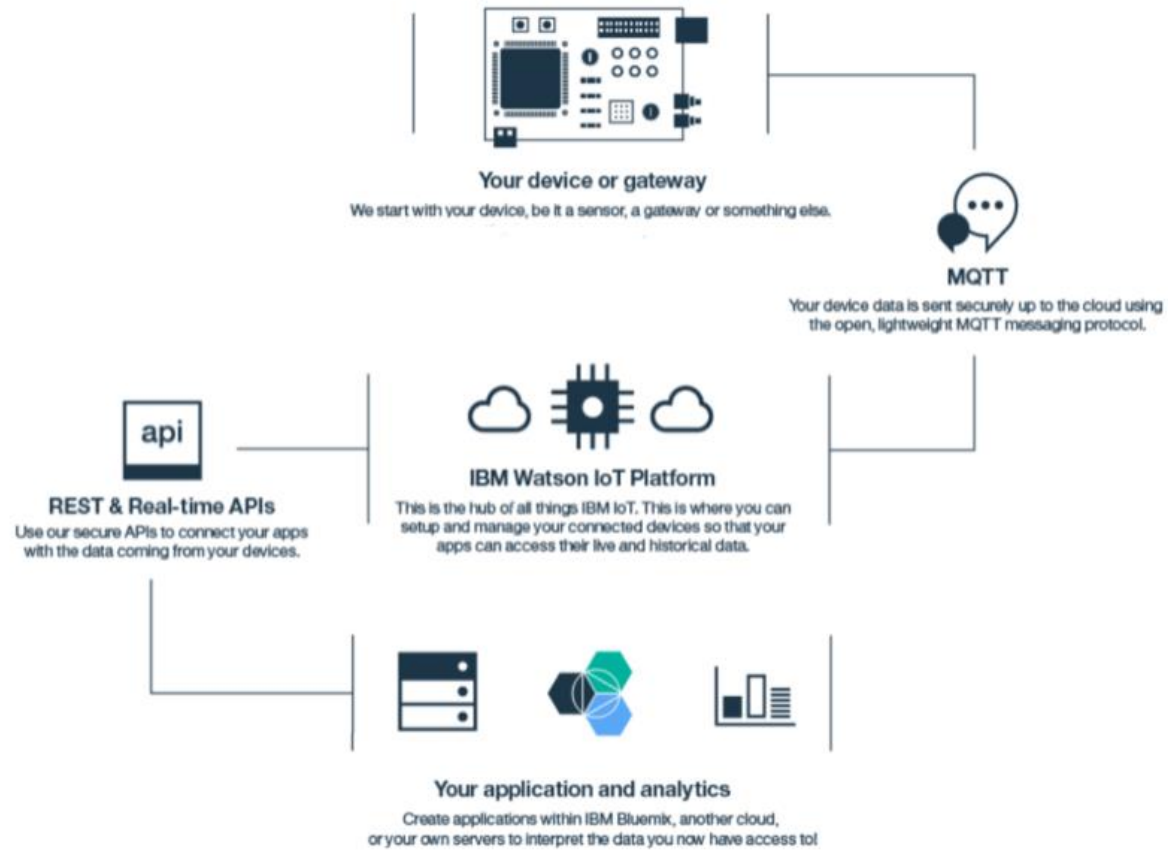
5. PROJECT DESIGN

5.1 . DATA FLOW DIAGRAM



5.2. SOLUTION ARCHITECTURE AND TECHNICAL ARCHITECTURE





5.3 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

6.1 . SPRINT PLANNING AND ESTIMATION

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	28 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	24 SEPTEMBER 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	25 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	23 SEPTEMBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	30 SEPTEMBER 2022
Solution Architecture	Prepare solution architecture document.	28 SEPTEMBER 2022

Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	20 OCTOBER 2022
Functional Requirement	Prepare the functional requirement document.	8 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	9 OCTOBER 2022
Technology Architecture	Prepare the technology architecture diagram.	10 OCTOBER 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	22 OCTOBER 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it.	IN PROGRESS..

6.2. SPRINT DELIVERY SCHEDULE

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Login	USN-1	As a Administrator, I need to give user id and passcode for ever workers over there in municipality	10	High	SENTHIL
Sprint-1	Login	USN-2	As a Co-Admin, I'll control the waste level by monitoring them vai real time web portal. Once the filling happens, I'll notify trash truck with location of bin with bin ID	10	High	DINESH
Sprint-2	Dashboard	USN-3	As a Truck Driver, I'll follow Co-Admin's Instruction to reach the filling bin in short roots and save time	20	Low	SANKAR
Sprint-3	Dashboard	USN-4	As a Local Garbage Collector, I'll gather all the waste from the garbage, load it onto a garbage truck, and deliver it to Landfills	20	Medium	SATHYA KANTH
Sprint-4	Dashboard	USN-5	As a Municipality officer, I'll make sure everything is proceeding as planned and without any problems	20	High	DINESH

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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

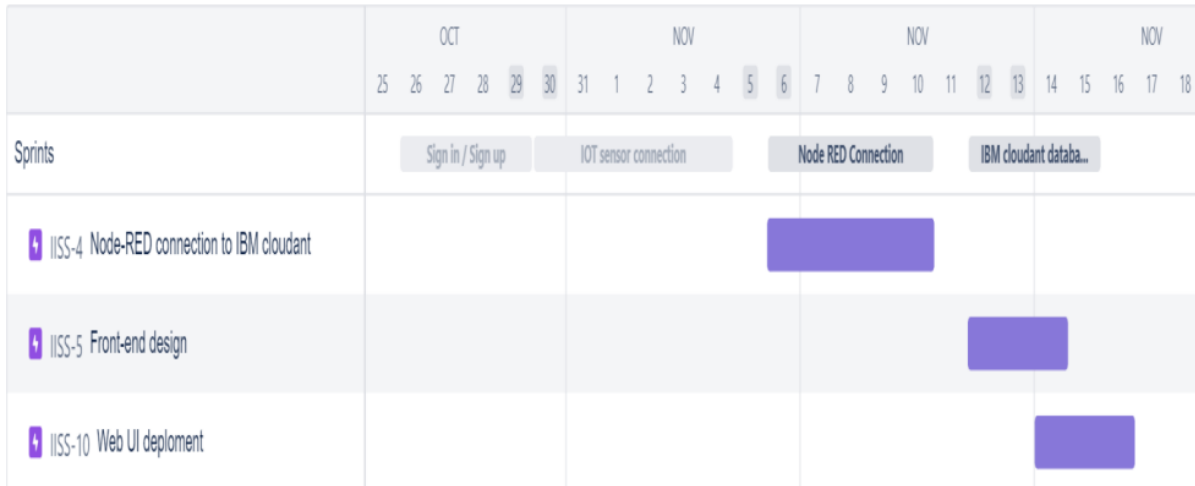
Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

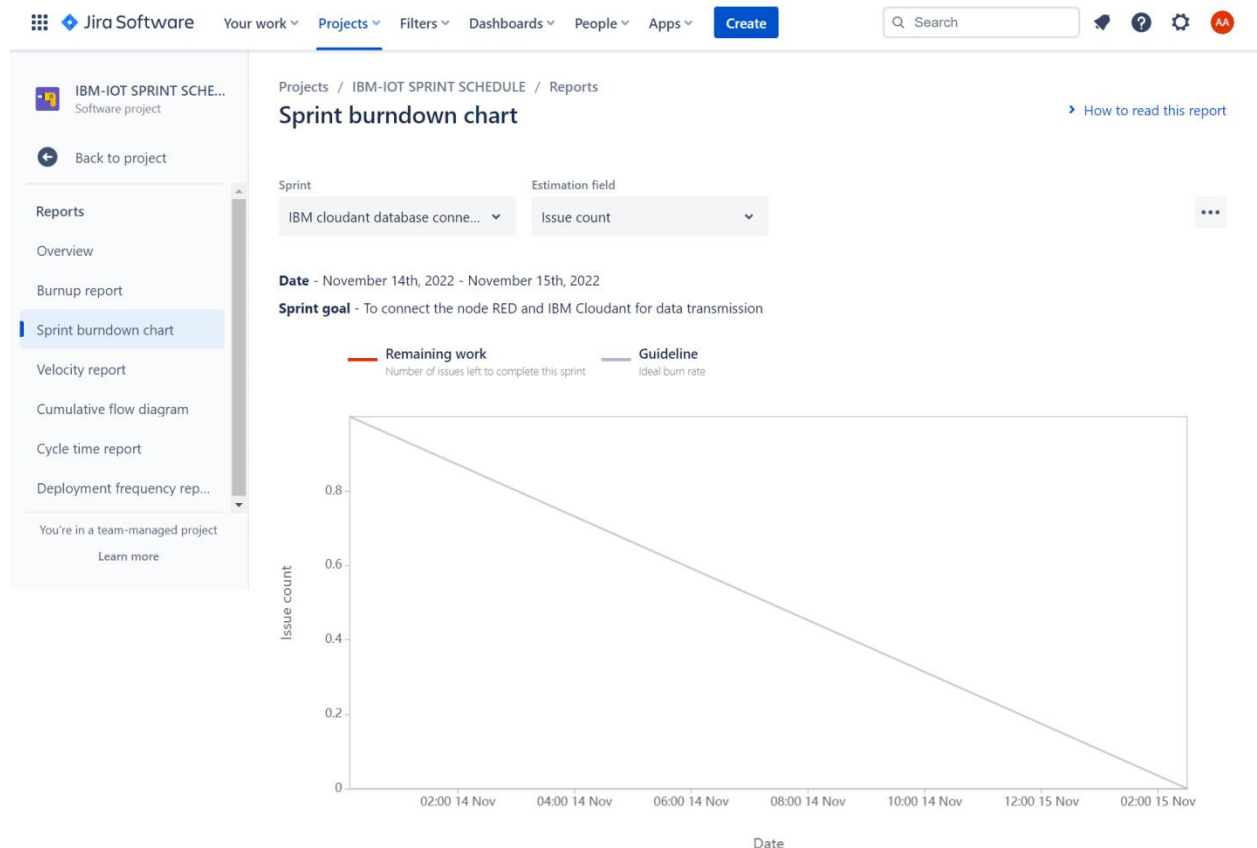
$$AV = \frac{\text{sprint duration}}{\text{velocity}} = \frac{20}{10} = 2$$

6.3. JIRA REPORTS

ROADMAP



SPRINT BURNDOWN CHART



7. CODING AND SOLUTION

7.1 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);

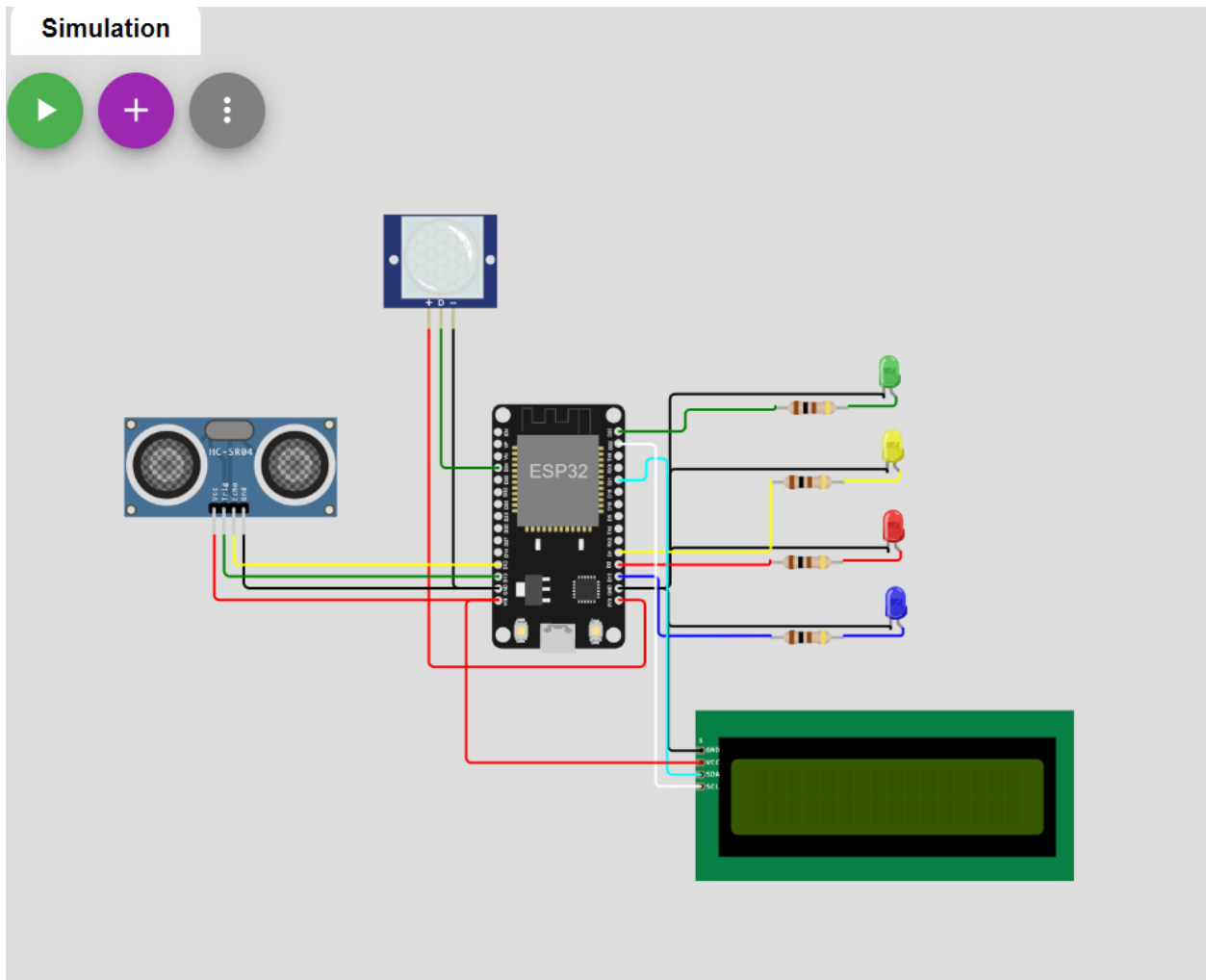
        if(command=="\SealBin\")
        {
            SealBin = true;
        }

    }

    data3="";
}

```

7.2 . Sensor Connection Setup



PHYSICAL COMPONENTS:

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

OUTPUT:

WOKWI SETUP

WOKWI

SAVE

SHARE

ESP32-IBMconnection[original]

Doc

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

No motion detected

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

Publish OK

Sending distance: 52.96

Publish OK

WEB UI

Log

Control

Bin in use

Range

Today Used

Enter Location change

date

VIEW STATUS

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

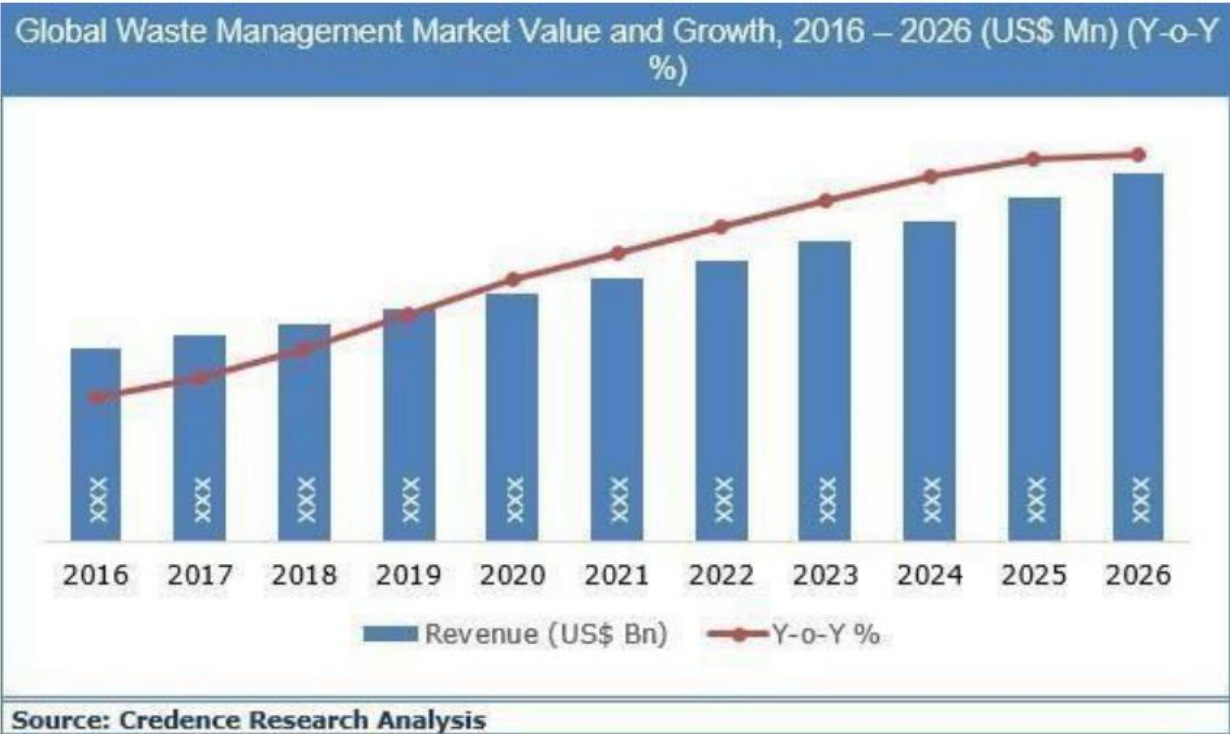
3. TEST CASE ANALYSIS

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

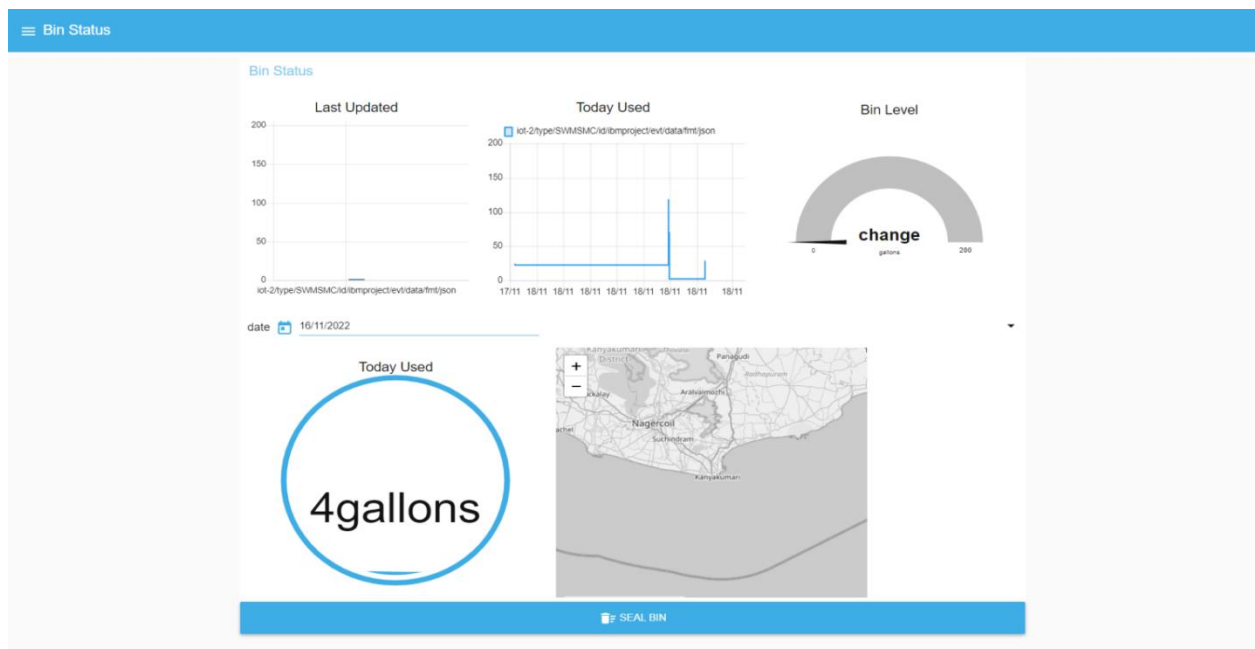
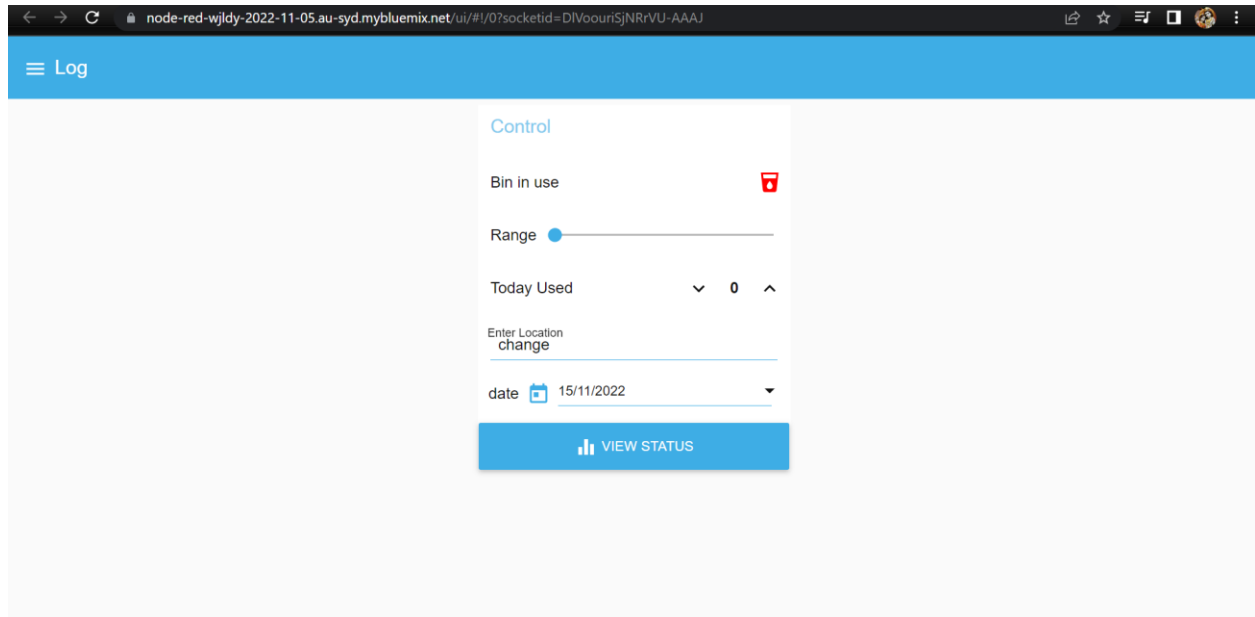
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9. RESULTS

9.1. Performance Metrics



9.2 . Admin Web UI



10. ADVANTAGES AND DISADVANTAGES

10.1. ADVANTAGES

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

10.2 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

14. 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.

13.1. 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="";
}

```


13.2 . GITHUB LINK

LINK : <https://github.com/IBM-EPBL/IBM-Project-46679-1660753203>