SMART WASTE MANAGEMENT FOR METROPOTIAN CITIES DOMAIN – INTERNET OF THINGS

PROJECT REPORT

SUBMITTED BY

TEAM ID : PNT2022TMID28979

VARSHA.N
VARSHINI.VP
VAISHNAVI.M
SALINI.R

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

PRINCE SHRI VENKATESHWARA PADMAVATHY ENGINEERING COLLEGE

CONTENT

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 ontime 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 aroundthe 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 IDwill 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 ItsApplications to Non-destructive Evaluation.

[2] Arduino, “Available at http://www.arduino.cc,” 2010.

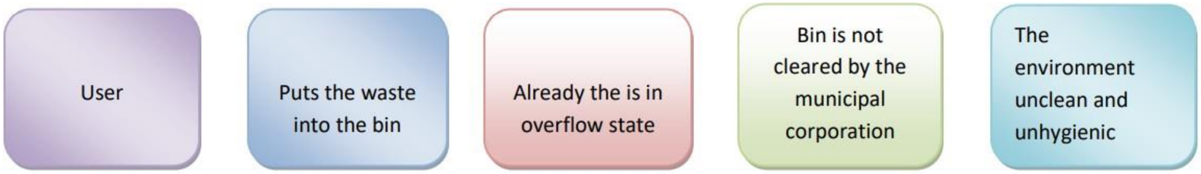
[3] M. Batty, “Smart Cities, Big Data,” Environment and Planning B: Planning and Design 2012,vol. 39, pp. 191– 93.

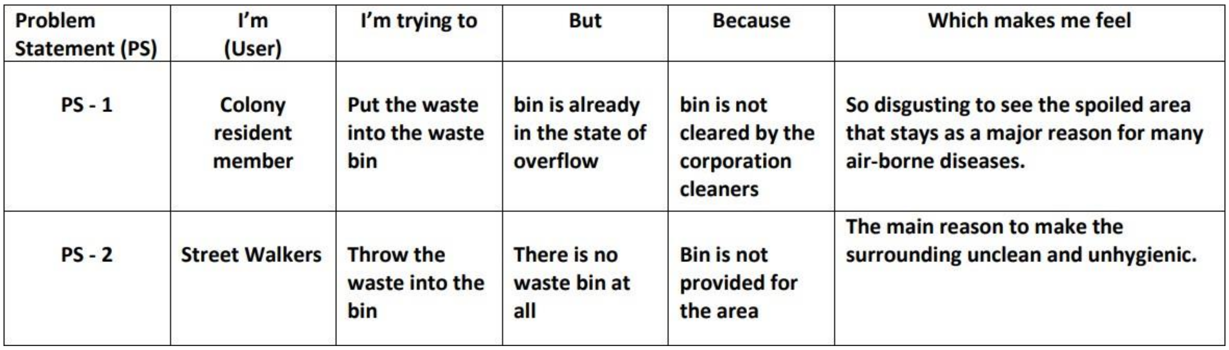
[4] Xu Li, Student Member, IEEE, Performance Evaluation of Vehicle-Based Mobile SensorNetworks for Traffic Monitoring.

[5] Yusuf Abdullahi Badamasi, The Working Principle Of An Arduino, Electronics, Computer andComputation (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 statementthat allows you to find the ideal solution for the challenges faced.


User
Puts the waste into the bin
Already the is in overflow state
Bin is not cleared by the municipal corporation
The environment unclean and unhygienic

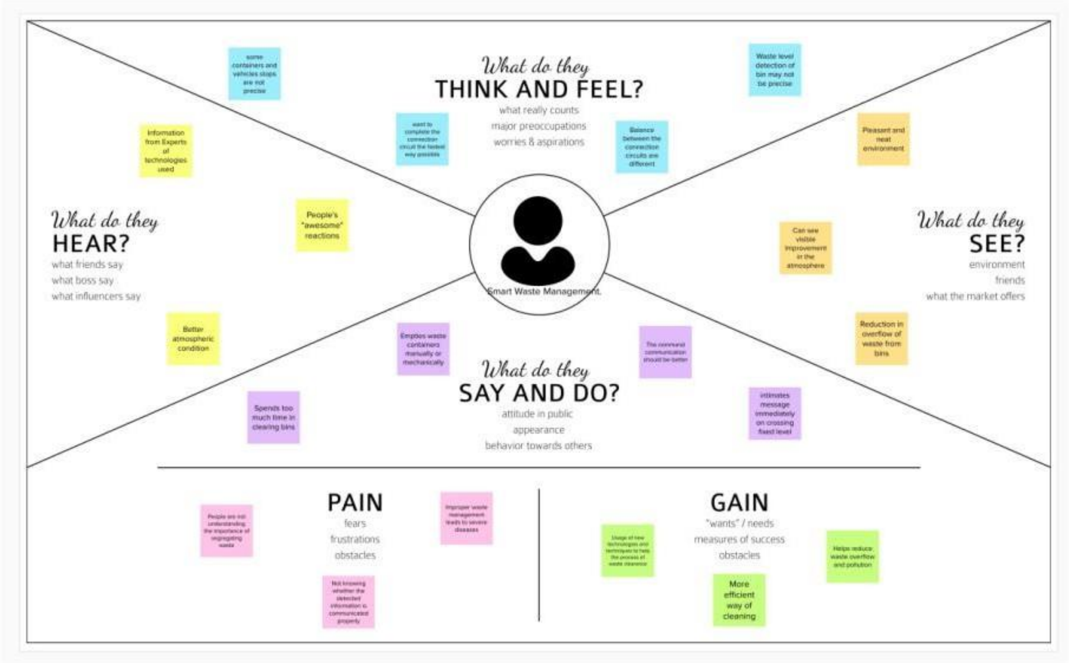


Problem Statement (PS)	I'm (User)	I'm trying to	But	Because	Which makes me feel
PS - 1	Colony resident member	Put the waste into the waste bin	bin is already in the state of overflow	bin is not cleared by the corporation cleaners	So disgusting to see the spoiled area that stays as a major reason for 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

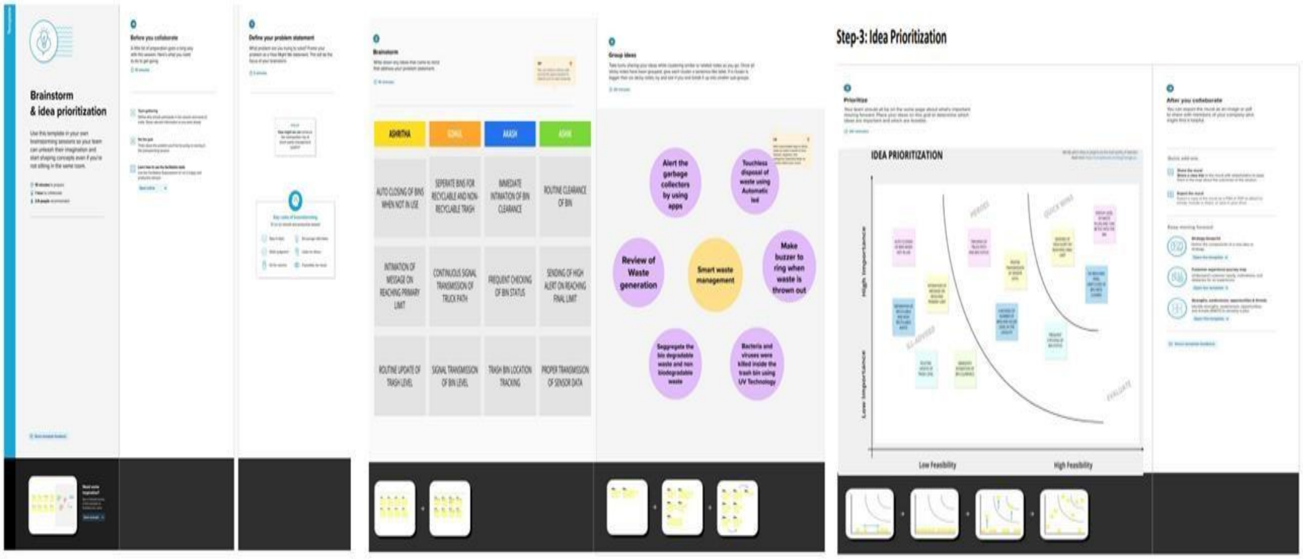
3.1. EMPATHY MAP CANVAS

Empathy map on Smart waste Management System for Metropolitan cities.


some containers and vehicles stops are not precise
What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Waste level detection of bin may not be precise
Information from Experts of technologies used
want to complete the connection circuit the fastest way possible
Balance between the connection circuits are different
Pleasant and neat environment
What do they
HEAR?
what friends say
what boss say
what influencers say
People's "awesome" reactions
Smart Waste Management.
Can see visible improvement in the atmosphere
What do they
SEE?
environment
friends
what the market offers
Better atmospheric condition
Empties waste containers manually or mechanically
The command communication should be better
Reduction in overflow of waste from bins
Spends too much time in clearing bins
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
intimates message immediately on crossing fixed level
PAIN
fears
frustrations
obstacles
People are not understanding the importance of segregating waste
Improper waste management leads to severe diseases
Not knowing whether the detected information is communicated properly
GAIN
"wants" / needs
measures of success
obstacles
Usage of new technologies and techniques to help the process of waste clearance
More efficient way of cleaning
Helps reduce waste overflow and pollution

Reference: https://www.mural.co/templates/empathy-map-canvas

3.2. Ideation and Brainstorming



ASHRITHA	[illegible]	AKASH	ASHIK
AUTO CLOSING OF BINS WHEN NOT IN USE	SEPERATE BINS FOR RECYCLABLE AND NON-RECYCLABLE TRASH	IMMEDIATE INTIMATION OF BIN CLEARANCE	ROUTINE CLEARANCE OF BIN
INTIMATION OF MESSAGE ON REACHING PRIMARY LIMIT	CONTINUOUS SIGNAL TRANSMISSION OF TRUCK PATH	FREQUENT CHECKING OF BIN STATUS	SENDING OF HIGH ALERT ON REACHING FINAL LIMIT
ROUTINE UPDATE OF TRASH LEVEL	SIGNAL TRANSMISSION OF BIN LEVEL	TRASH BIN LOCATION TRACKING	PROPER TRANSMISSION OF SENSOR DATA

Smart waste management

Alert the garbage collectors by using apps

Touchless disposal of waste using Automatic lid

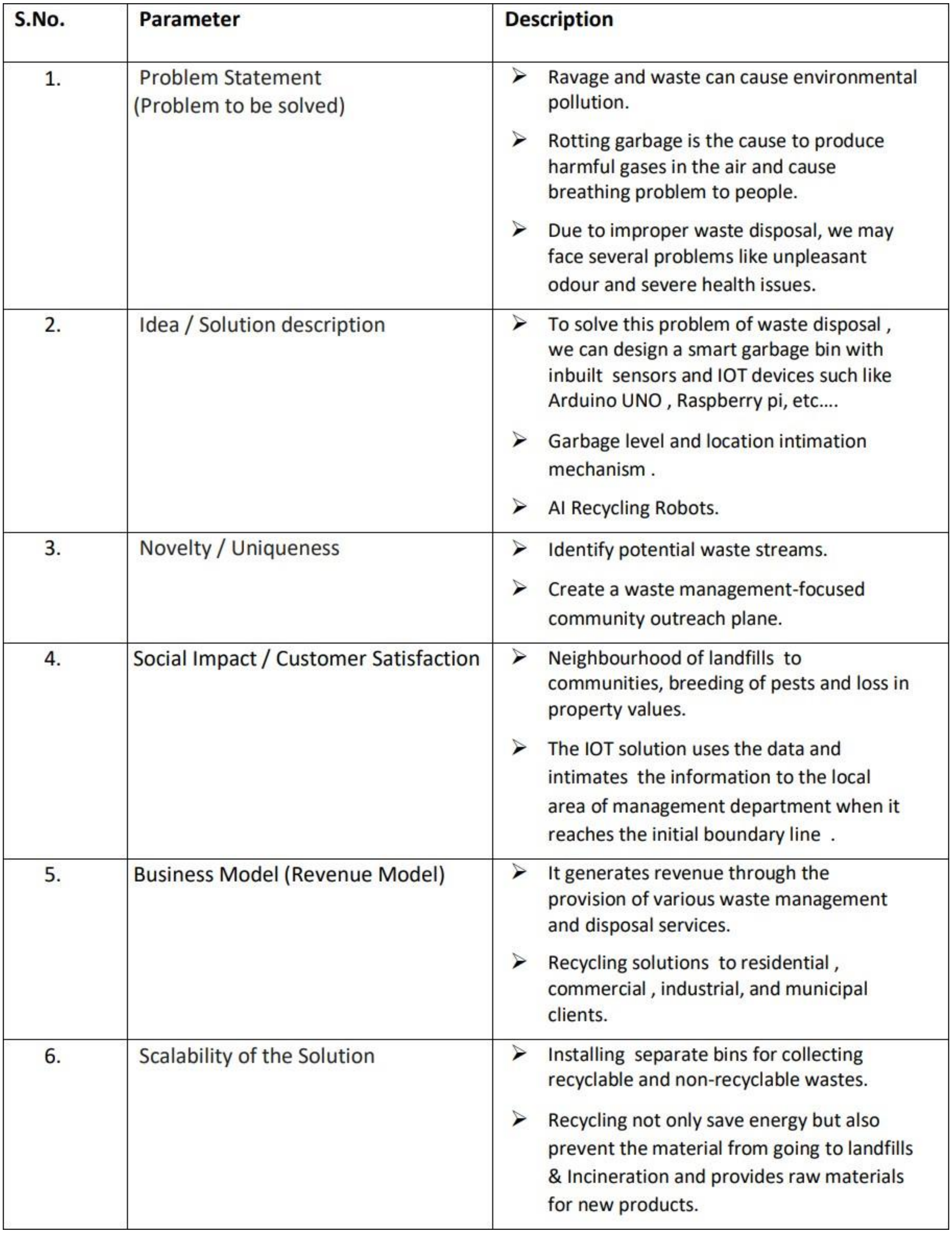
Review of Waste generation

Make buzzer to ring when waste is thrown out

Seggregate the bin degradable waste and non biodegradable waste

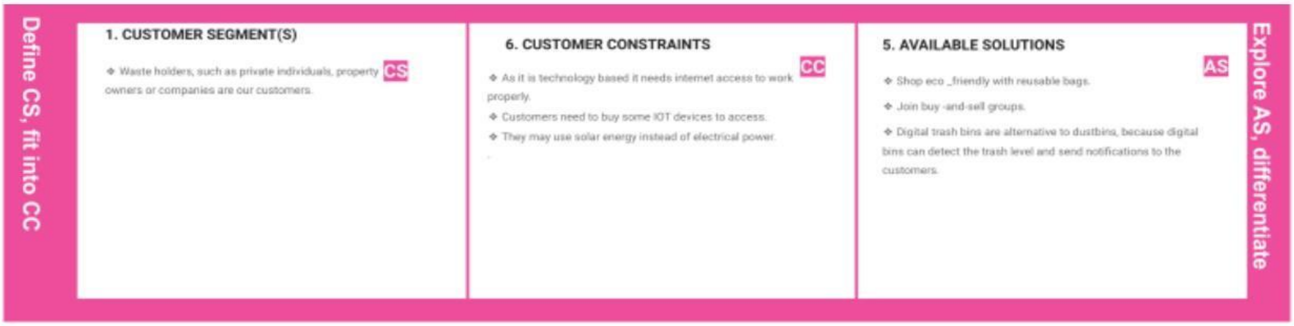
Bacteria and viruses were killed inside the trash bin using UV Technology

3.3. PROPOSED SOLUTION



S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	➢ 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	➢ 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	➢ Identify potential waste streams. ➢ Create a waste management-focused community outreach plane.
4.	Social Impact / Customer Satisfaction	➢ 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)	➢ 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	➢ 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



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

❖ Waste holders, such as private individuals, property owners or companies are our customers.

6. CUSTOMER CONSTRAINTS CC

❖ As it is technology based it needs internet access to work properly.

❖ Customers need to buy some IOT devices to access.

❖ They may use solar energy instead of electrical power.

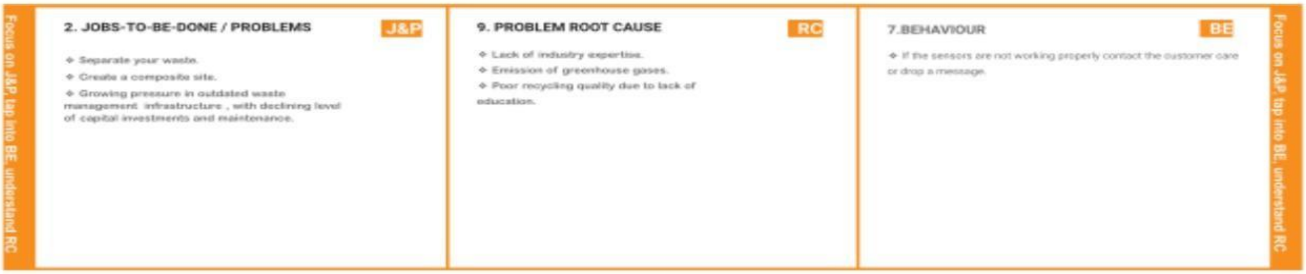
5. AVAILABLE SOLUTIONS AS

❖ Shop eco _friendly with reusable bags.

❖ Join buy -and-sell groups.

❖ Digital trash bins are alternative to dustbins, because digital bins can detect the trash level and send notifications to the customers.

Explore AS, differentiate



Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

❖ Separate your waste.

❖ Create a composite site.

❖ Growing pressure in outdated waste management infrastructure , with declining level of capital investments and maintenance.

9. PROBLEM ROOT CAUSE RC

❖ Lack of industry expertise.

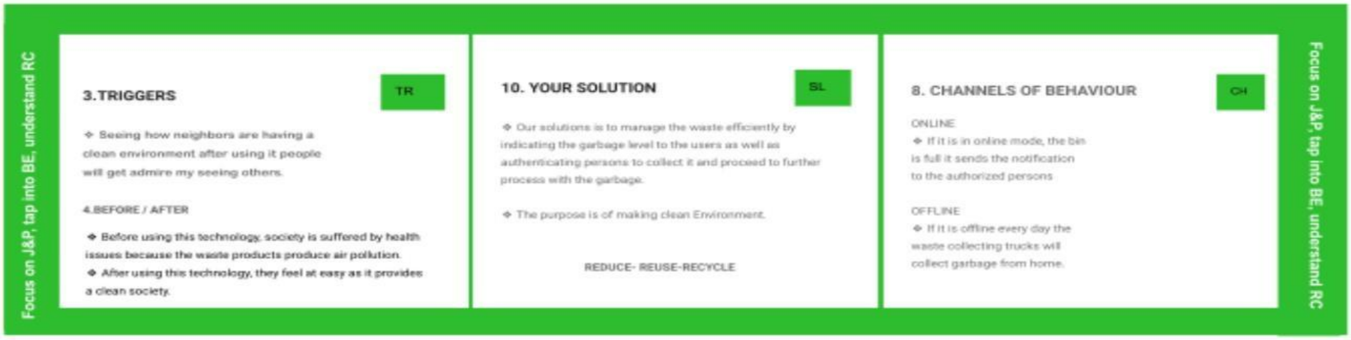
❖ Emission of greenhouse gases.

❖ Poor recycling quality due to lack of education.

7.BEHAVIOUR BE

❖ If the sensors are not working properly contact the customer care or drop a message.

Focus on J&P, tap into BE, understand RC



Focus on J&P, tap into BE, understand RC

3.TRIGGERS TR

❖ Seeing how neighbors are having a clean environment after using it people will get admire my seeing others.

4.BEFORE / AFTER

❖ Before using this technology, society is suffered by health issues because the waste products produce air pollution.

❖ After using this technology, they feel at easy as it provides a clean society.

10. YOUR SOLUTION SL

❖ Our solutions is to manage the waste efficiently by indicating the garbage level to the users as well as authenticating persons to collect it and proceed to further process with the garbage.

❖ The purpose is of making clean Environment.

REDUCE- REUSE-RECYCLE

8. CHANNELS OF BEHAVIOUR CH

ONLINE

❖ If it is in online mode, the bin is full it sends the notification to the authorized persons

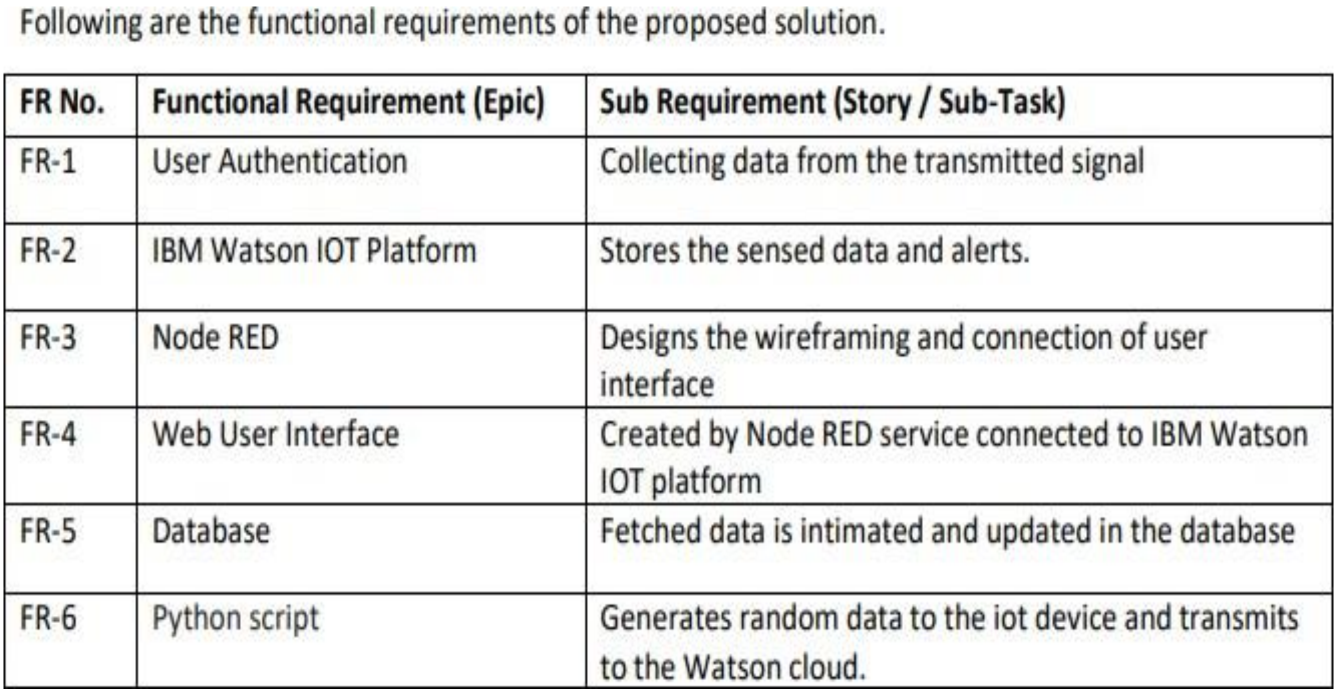
OFFLINE

❖ If it is offline every day the waste collecting trucks will collect garbage from home.

Focus on J&P, tap into BE, understand RC

4 . REQUIREMENT ANALYSIS

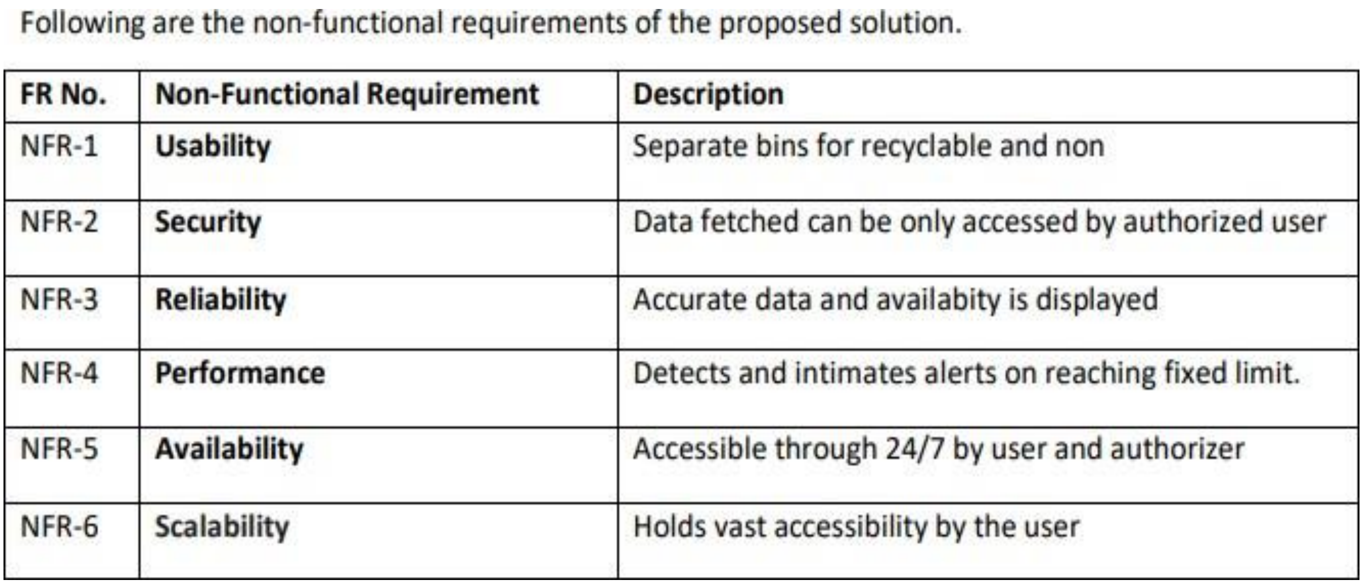
4.1 FUNCTIONAL REQUIREMENT



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	Fetched data is intimated and updated 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

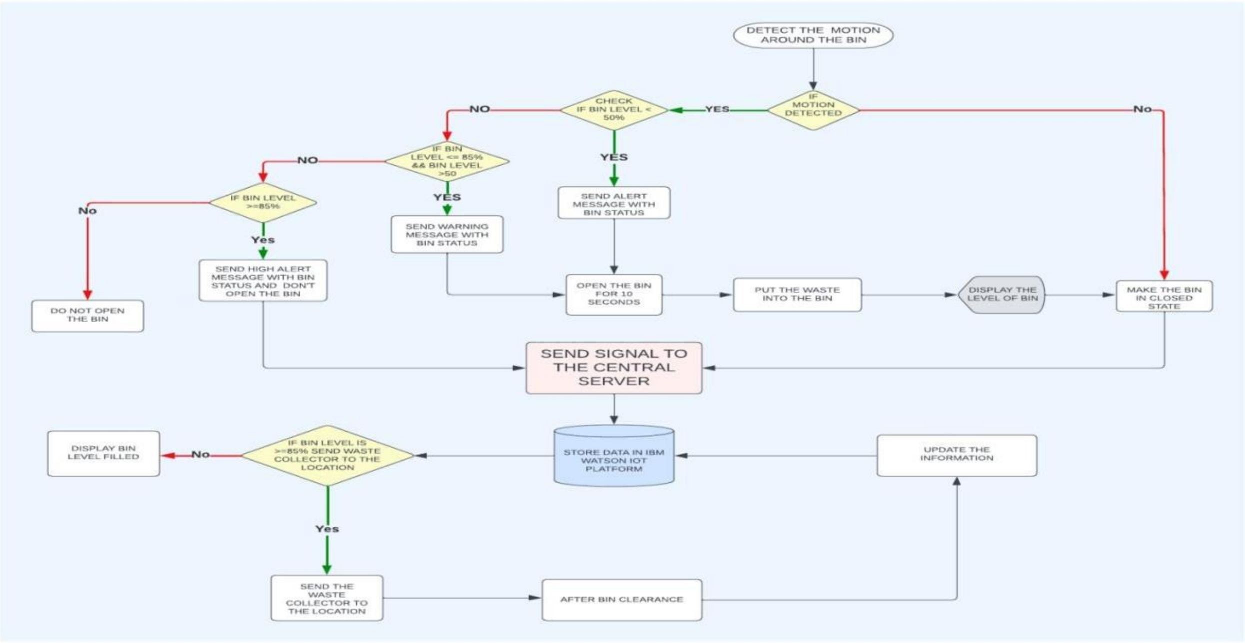


Following are the non-functional requirements of the proposed solution.

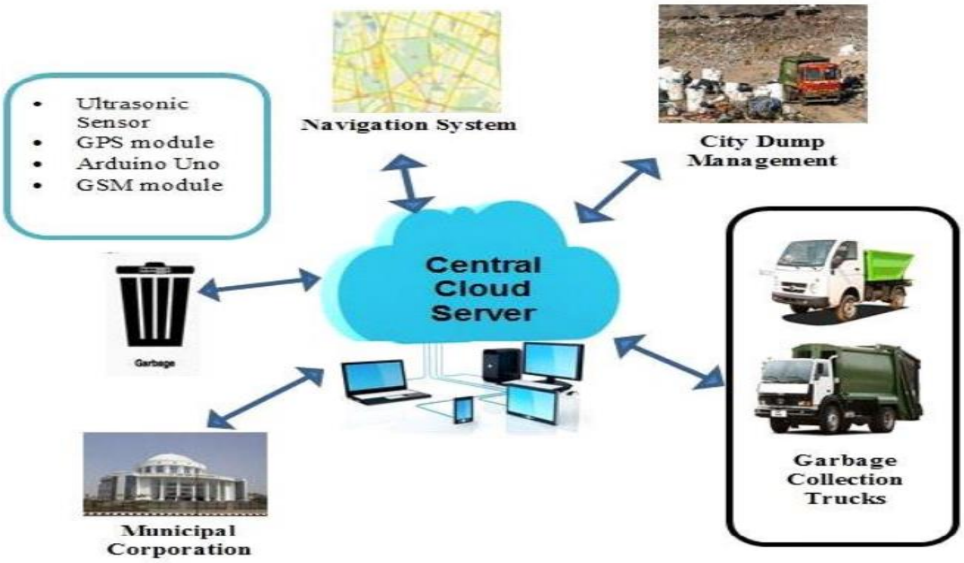
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Separate bins for recyclable and non
NFR-2	**Security**	Data fetched can be only accessed by authorized user
NFR-3	**Reliability**	Accurate data and availabity is displayed
NFR-4	**Performance**	Detects and intimates alerts on reaching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

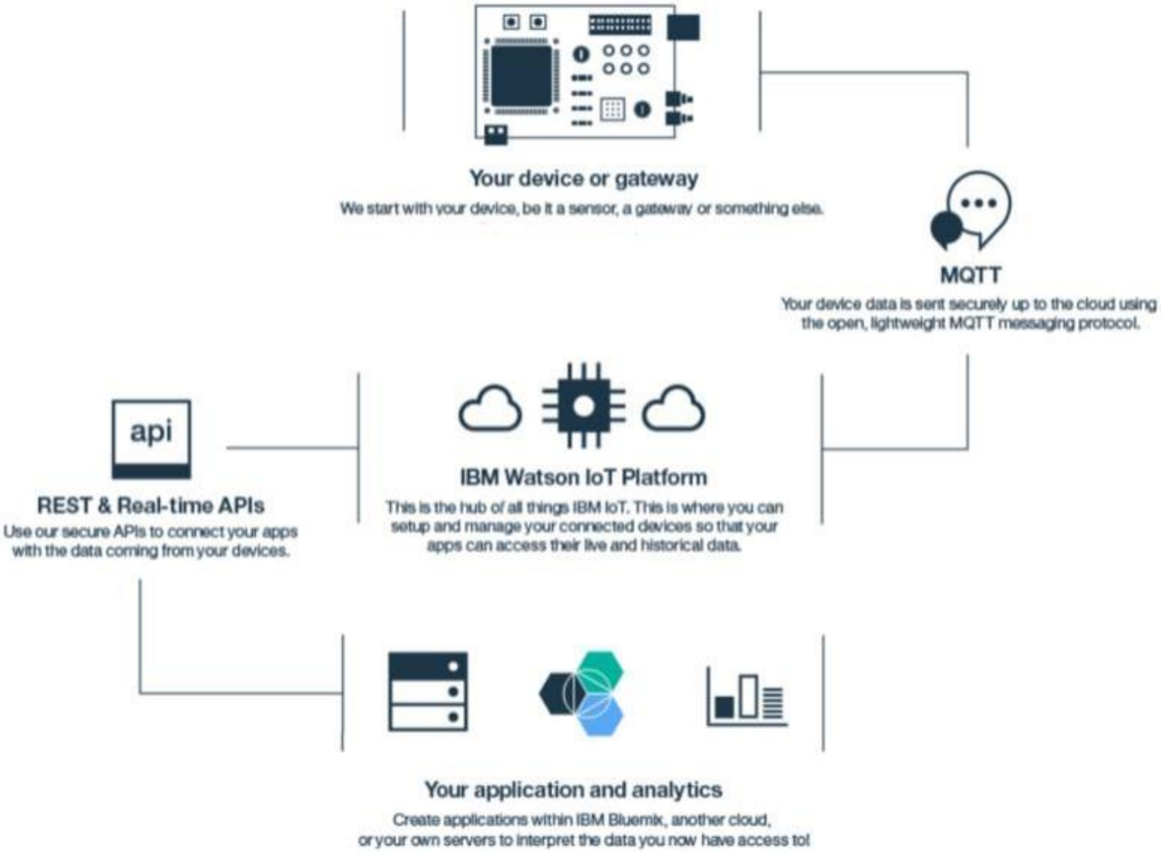
5. PROJECT DESIGN

5.1 . DATA FLOW DIAGRAM


DETECT THE MOTION AROUND THE BIN
IF MOTION DETECTED
YES
No
CHECK IF BIN LEVEL < 50%
NO
YES
SEND ALERT MESSAGE WITH BIN STATUS
IF BIN LEVEL <= 85% && BIN LEVEL >50
NO
YES
SEND WARNING MESSAGE WITH BIN STATUS
IF BIN LEVEL >=85%
No
Yes
SEND HIGH ALERT MESSAGE WITH BIN STATUS AND DON'T OPEN THE BIN
DO NOT OPEN THE BIN
OPEN THE BIN FOR 10 SECONDS
PUT THE WASTE INTO THE BIN
DISPLAY THE LEVEL OF BIN
MAKE THE BIN IN CLOSED STATE
SEND SIGNAL TO THE CENTRAL SERVER
STORE DATA IN IBM WATSON IOT PLATFORM
UPDATE THE INFORMATION
IF BIN LEVEL IS >=85% SEND WASTE COLLECTOR TO THE LOCATION
No
DISPLAY BIN LEVEL FILLED
Yes
SEND THE WASTE COLLECTOR TO THE LOCATION
AFTER BIN CLEARANCE

5.2. SOLUTION ARCHITECTURE AND TECHNICAL ARCHITECTURE





Your device or gateway

We start with your device, be it a sensor, a gateway or something else.

MQTT

Your device data is sent securely up to the cloud using the open, lightweight MQTT messaging protocol.

IBM Watson IoT Platform

This is the hub of all things IBM IoT. This is where you can setup and manage your connected devices so that your apps can access their live and historical data.

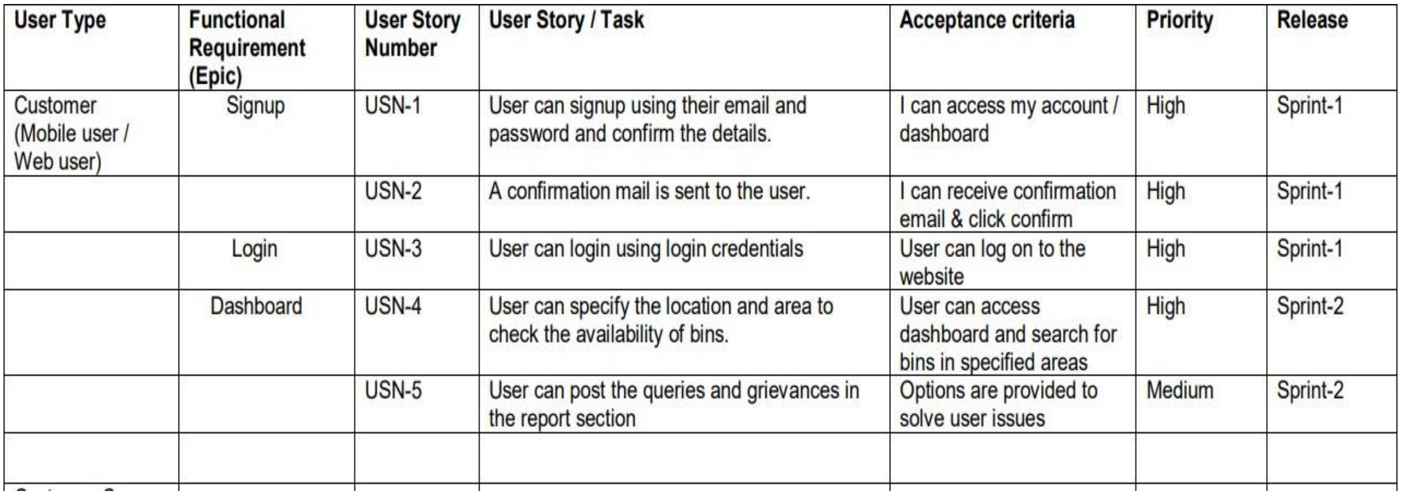
REST & Real-time APIs

Use our secure APIs to connect your apps with the data coming from your devices.

Your application and analytics

Create applications within IBM Bluemix, another cloud, or your own servers to interpret the data you now have access to!

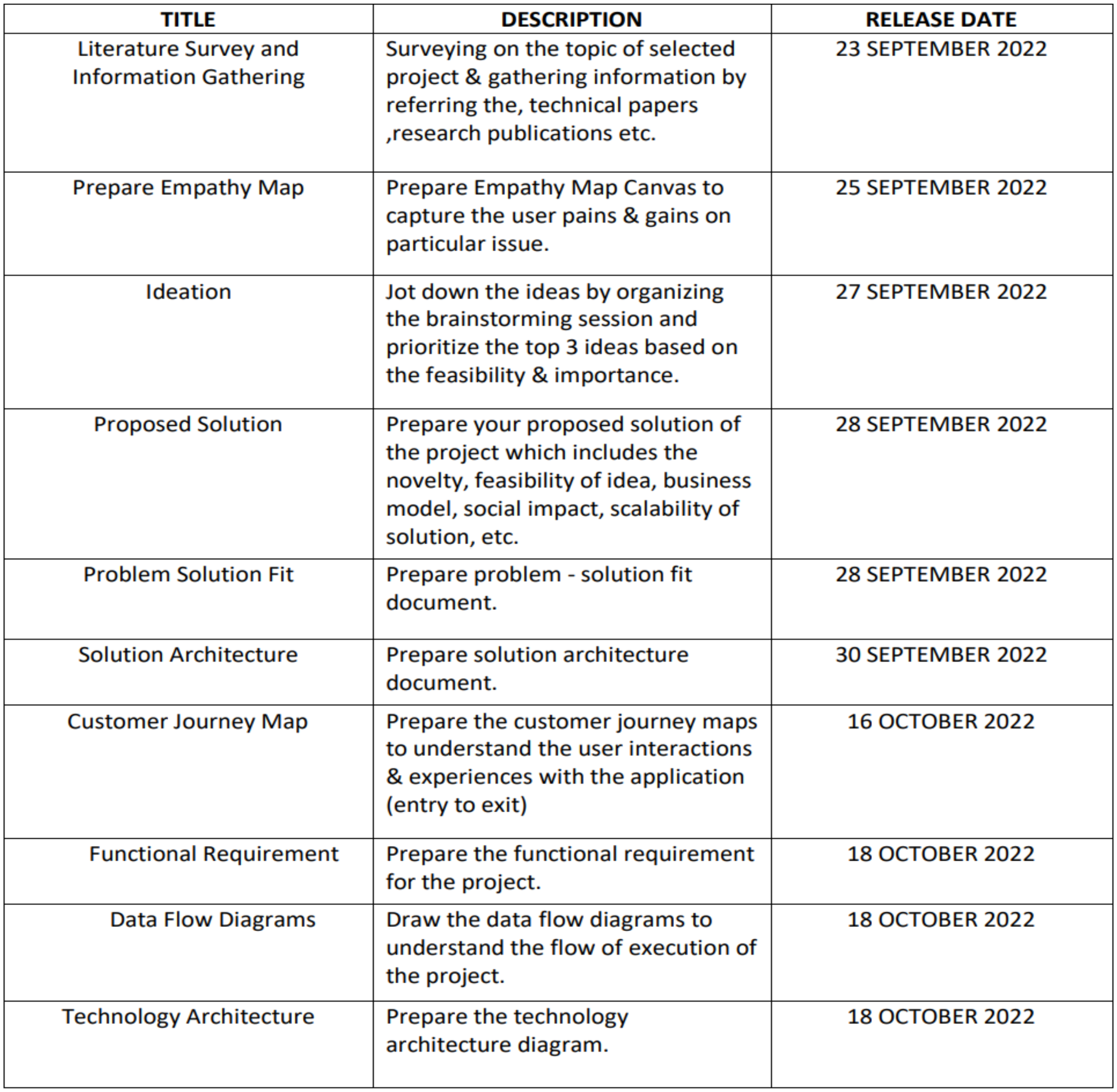
5.3 USER STORIES

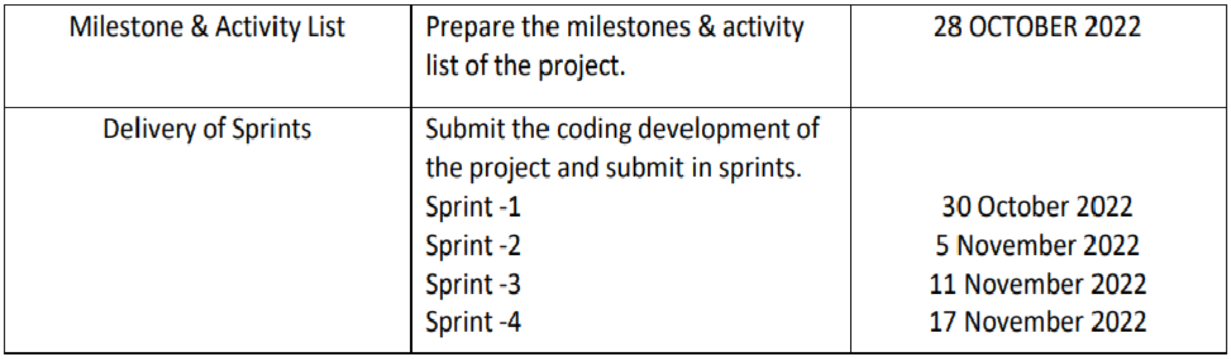


User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user / Web user)	Signup	USN-1	User can signup using their email and password and confirm the details.	I can access my account / dashboard	High	Sprint-1
		USN-2	A confirmation 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

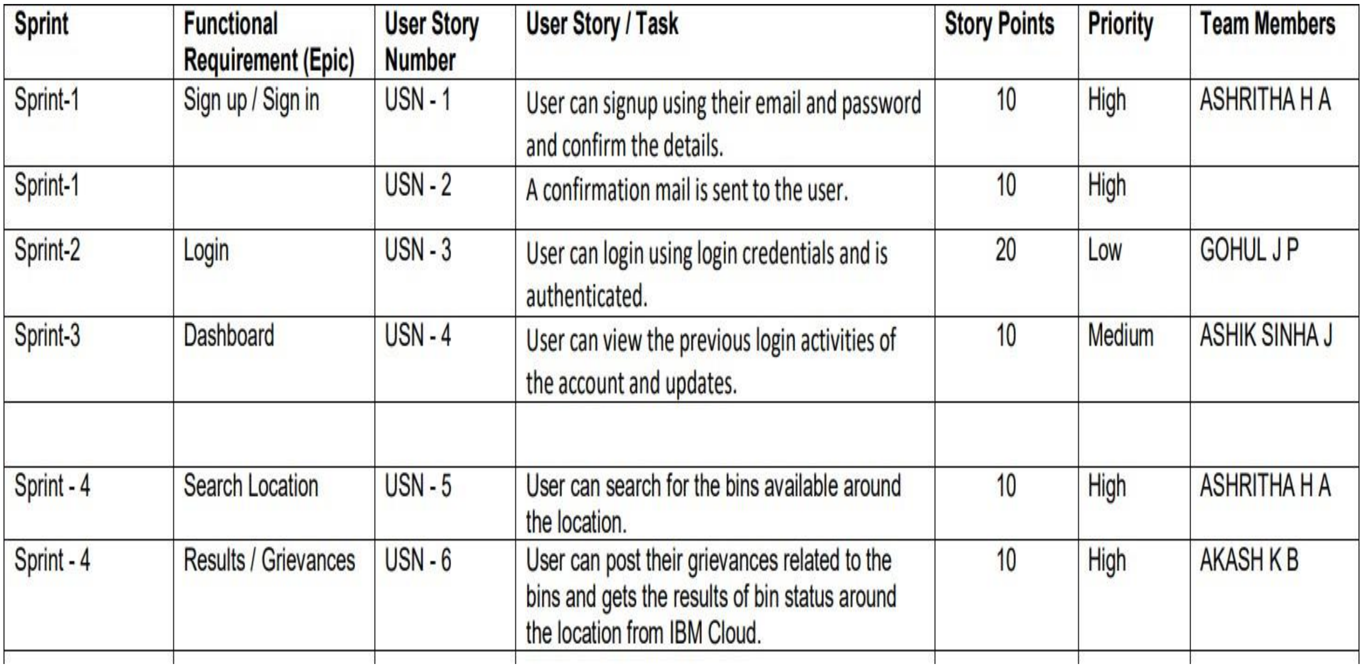
6.1 . SPRINT PLANNING AND ESTIMATION



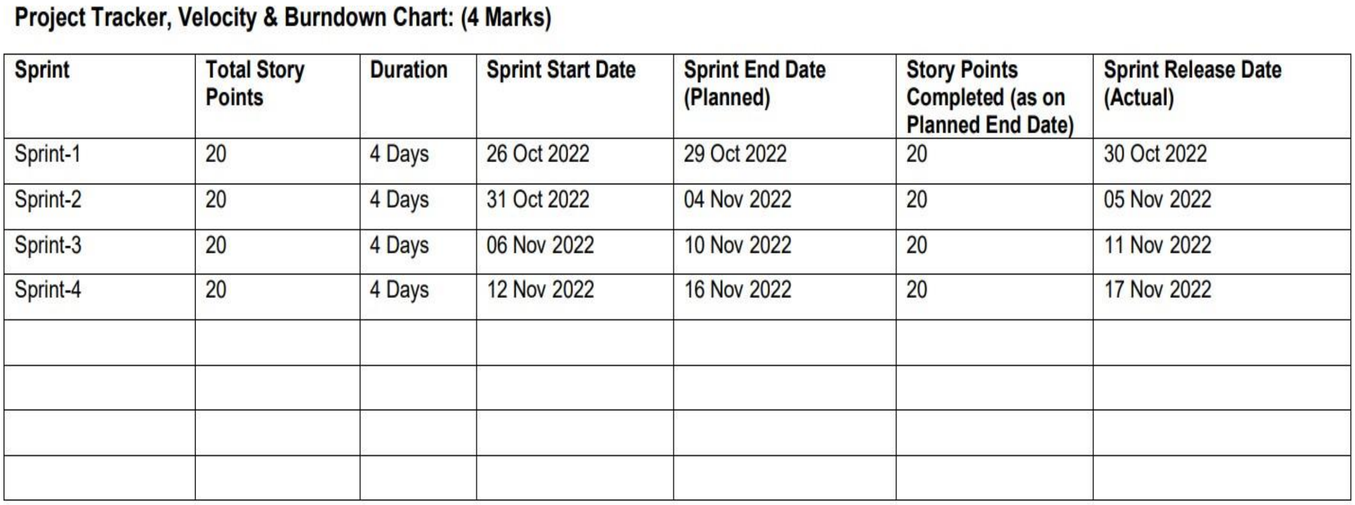


TITLE	DESCRIPTION	RELEASE DATE
Literature Survey and Information Gathering	Surveying on the topic of selected project & gathering information by referring the, technical papers ,research publications etc.	23 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user pains & gains on particular issue.	25 SEPTEMBER 2022
Ideation	Jot down the ideas by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	27 SEPTEMBER 2022
Proposed Solution	Prepare your proposed solution of the project which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	28 SEPTEMBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	28 SEPTEMBER 2022
Solution Architecture	Prepare solution architecture document.	30 SEPTEMBER 2022
Customer Journey Map	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit)	16 OCTOBER 2022
Functional Requirement	Prepare the functional requirement for the project.	18 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams to understand the flow of execution of the project.	18 OCTOBER 2022
Technology Architecture	Prepare the technology architecture diagram.	18 OCTOBER 2022
Milestone & Activity List	Prepare the milestones & activity list of the project.	28 OCTOBER 2022
Delivery of Sprints	Submit the coding development of the project and submit in sprints. Sprint -1 Sprint -2 Sprint -3 Sprint -4	30 October 2022 5 November 2022 11 November 2022 17 November 2022

6.2. SPRINT DELIVERY SCHEDULE

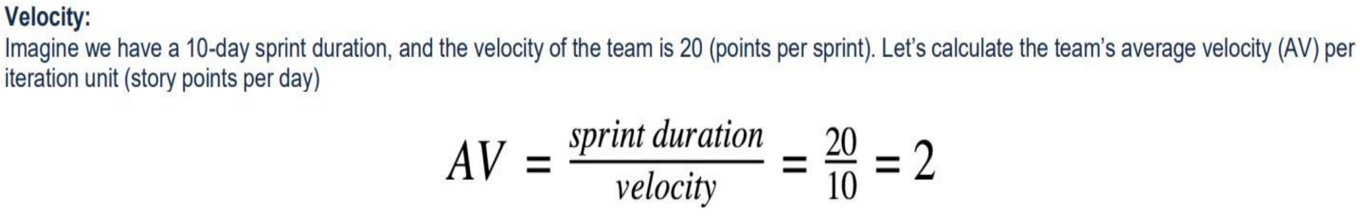


Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Sign up / Sign in	USN - 1	User can signup using their email and password and confirm the details.	10	High	ASHRITHA H A
Sprint-1		USN - 2	A confirmation mail is sent to the user.	10	High	
Sprint-2	Login	USN - 3	User can login using login credentials and is authenticated.	20	Low	GOHUL J P
Sprint-3	Dashboard	USN - 4	User can view the previous login activities of the account and updates.	10	Medium	ASHIK SINHA J
Sprint - 4	Search Location	USN - 5	User can search for the bins available around the location.	10	High	ASHRITHA H A
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	AKASH K B



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	4 Days	26 Oct 2022	29 Oct 2022	20	30 Oct 2022
Sprint-2	20	4 Days	31 Oct 2022	04 Nov 2022	20	05 Nov 2022
Sprint-3	20	4 Days	06 Nov 2022	10 Nov 2022	20	11 Nov 2022
Sprint-4	20	4 Days	12 Nov 2022	16 Nov 2022	20	17 Nov 2022



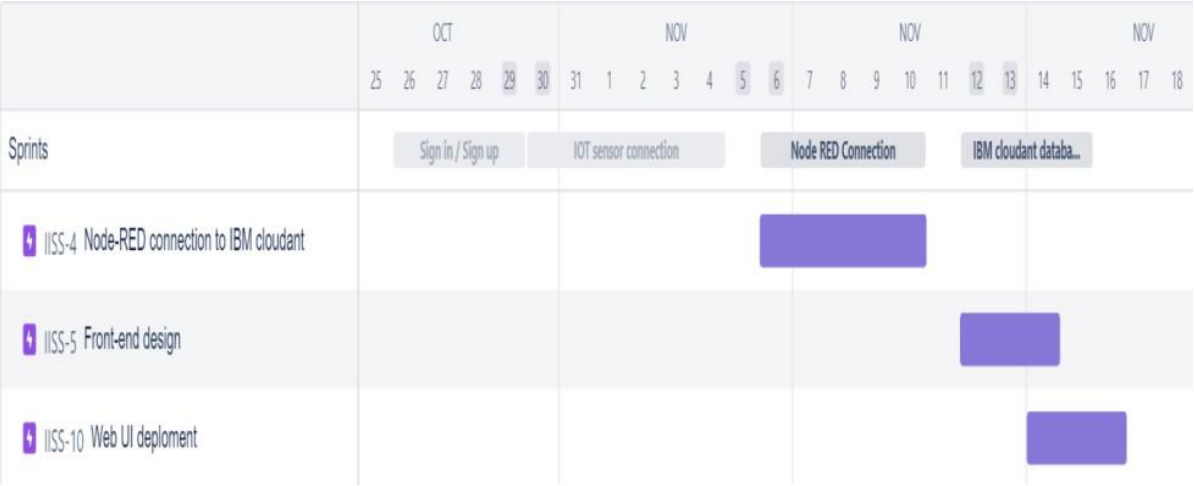
Velocity:

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)

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

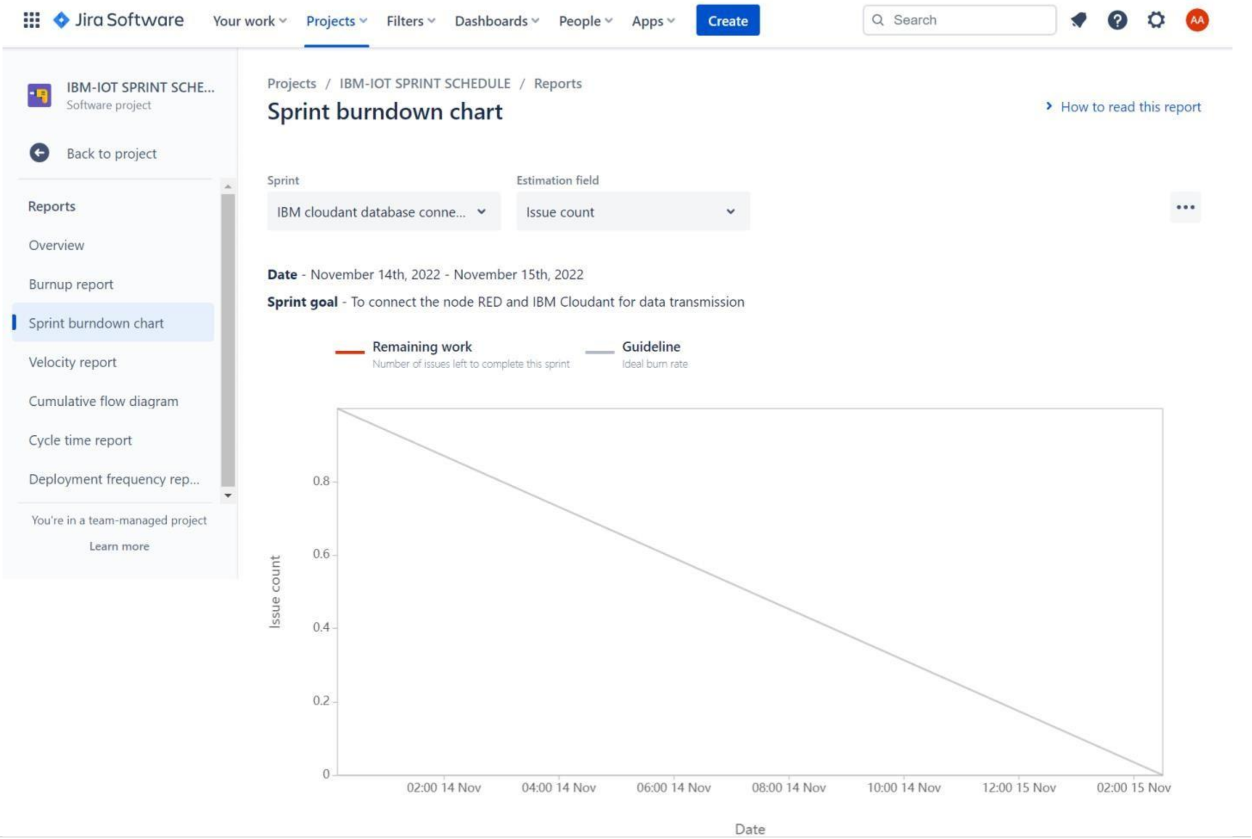
6.3. JIRA

REPORTS



ROADMAP

SPRINT BURNDOWN CHART



Projects / IBM-IOT SPRINT SCHEDULE / Reports

Sprint burndown chart

Sprint: IBM cloudant database conne...

Estimation field: Issue count

Date - November 14th, 2022 - November 15th, 2022

Sprint goal - To connect the node RED and IBM Cloudant for data transmission

7. CODING AND SOLUTION

7.1 Wokwi code for Sensor transmission

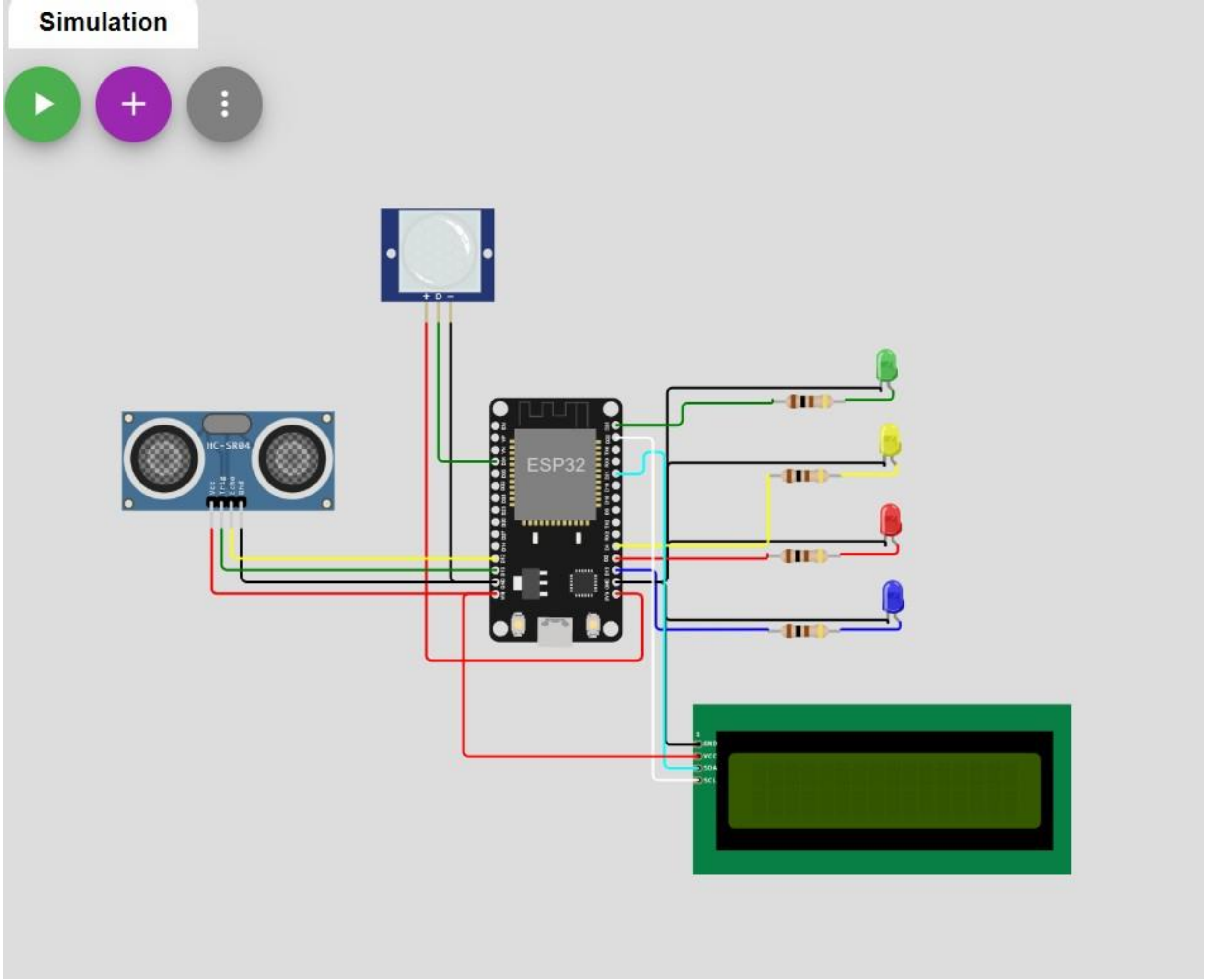
```
#include <WiFi.h>                                   // library for wifi
#include <PubSubClient.h>                           // library for MQTT
#include <LiquidCrystal_I2C.h>
#include <mjson.h> LiquidCrystal_I2C
lcd(0x27, 20, 4);

// ...................................... 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 formatin which data to
be send
char topic[] = "iot-2/cmd/led/fmt/String";                                     // cmd Represent type and command is test format ofstrings
char authMethod[] = "use-token-auth";                                          // authentication methodchar
token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;                     //Client id

// ..............................................................................................................................................................

WiFiClient wifiClient;                                                        // creating instance for wificlientPubSubClient
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 okelse 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 okelse 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



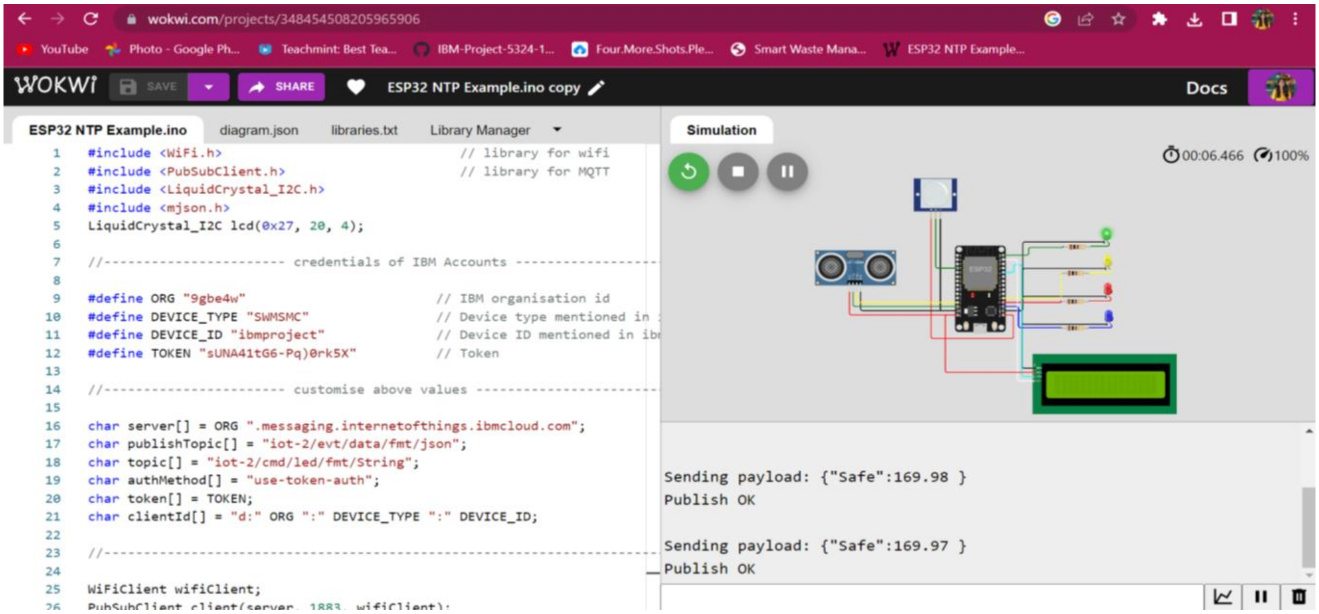
PHYSICAL COMPONENTS:

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

OUTPUT:

WOKWI

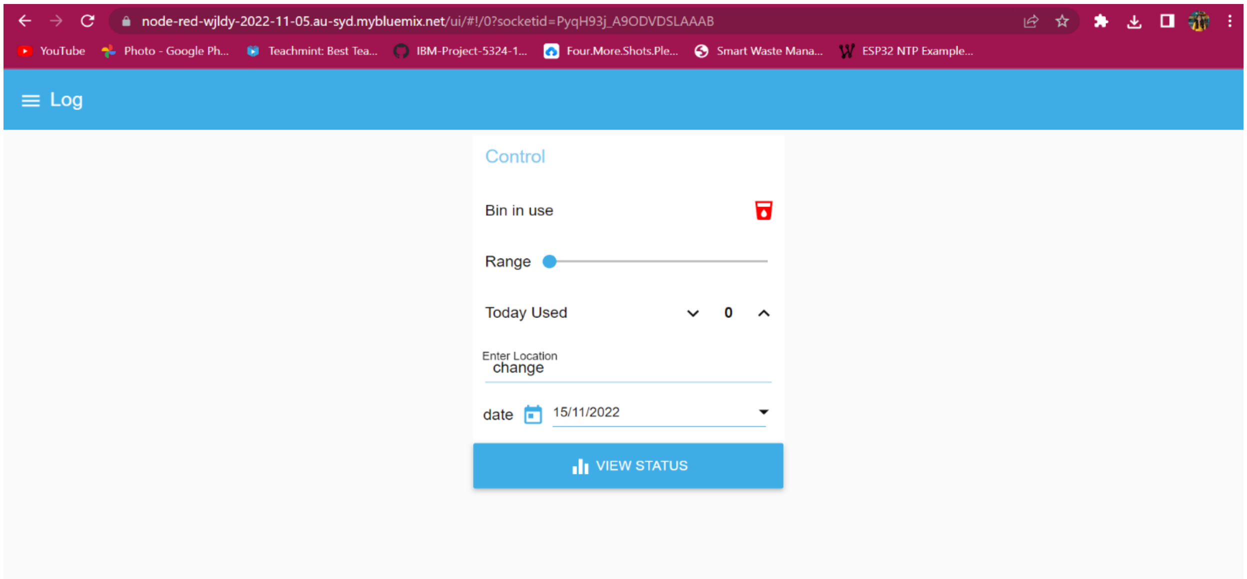
SETUP



```
#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
#define DEVICE_ID "ibmproject"                // Device ID mentioned in ib
#define TOKEN "sUNA41tG6-Pq)0rk5X"            // Token

//----------------------- customise above values ----------------------

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/data/fmt/json";
char topic[] = "iot-2/cmd/led/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

//--------------------------------------------------------------------

WiFiClient wifiClient;
PubSubClient client(server, 1883, wifiClient);
```

```
Sending payload: {"Safe":169.98 }
Publish OK

Sending payload: {"Safe":169.97 }
Publish OK
```

WEB UI



Control

Bin in use

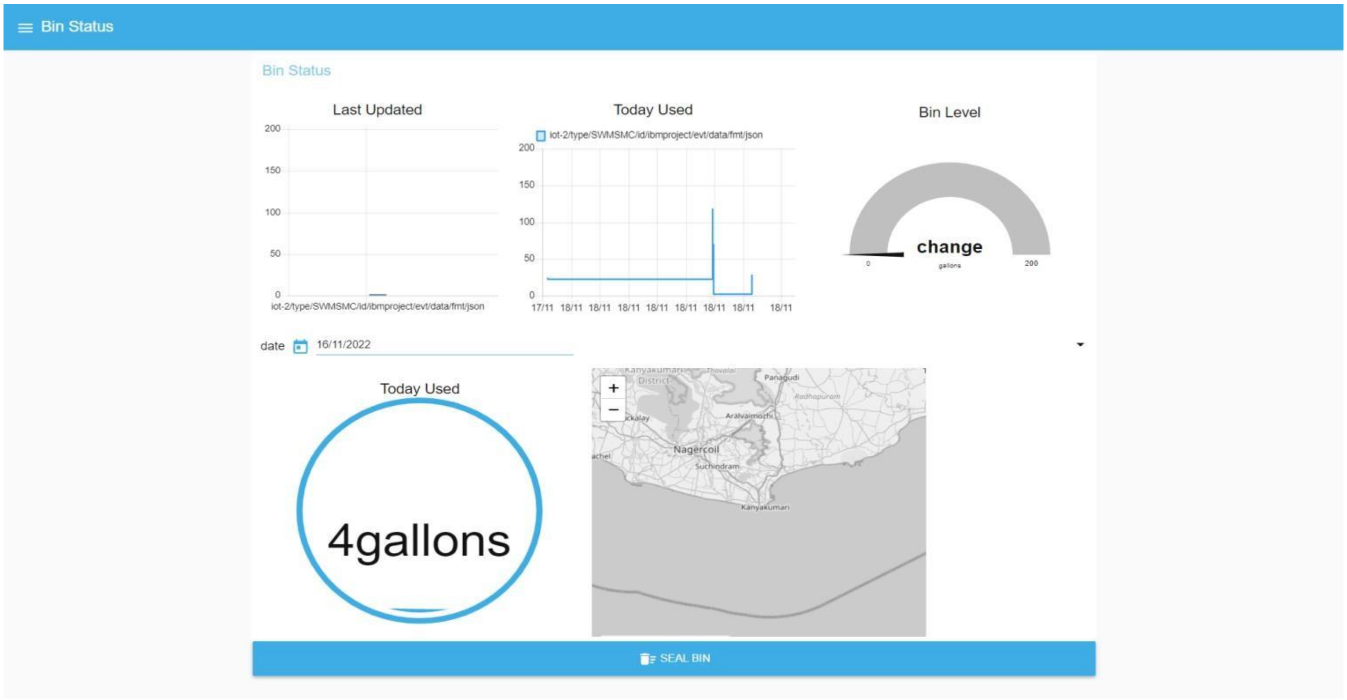
Range

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
cmMinimum 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.

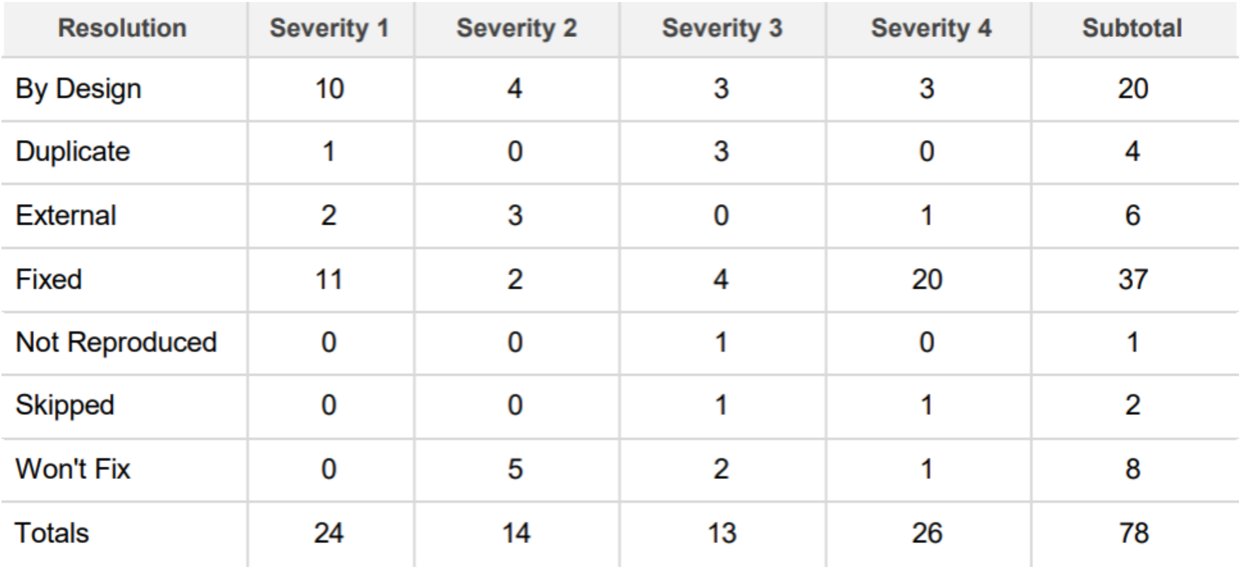
USER ACCEPTANCE TESTING

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

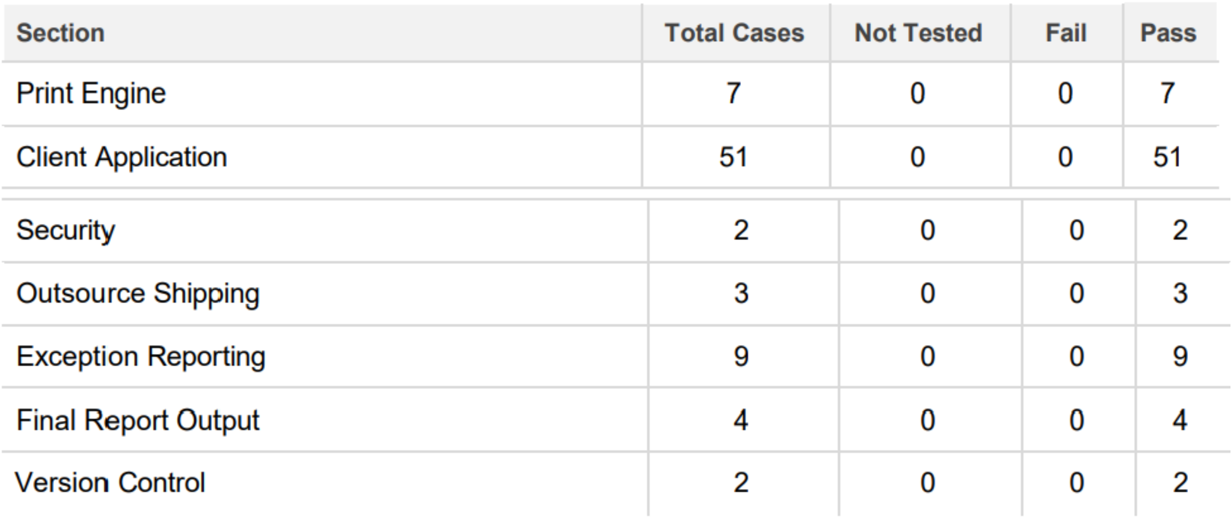
This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.



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

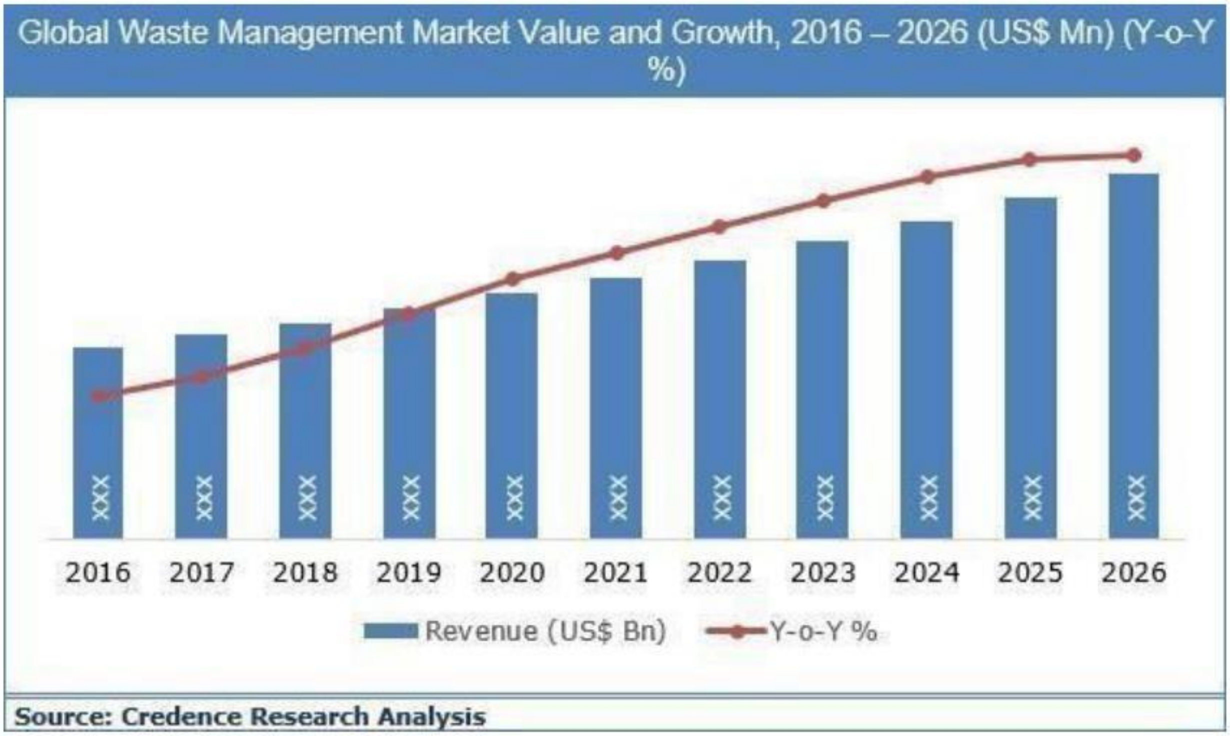
This report shows the number of test cases that have passed , failed and untested.



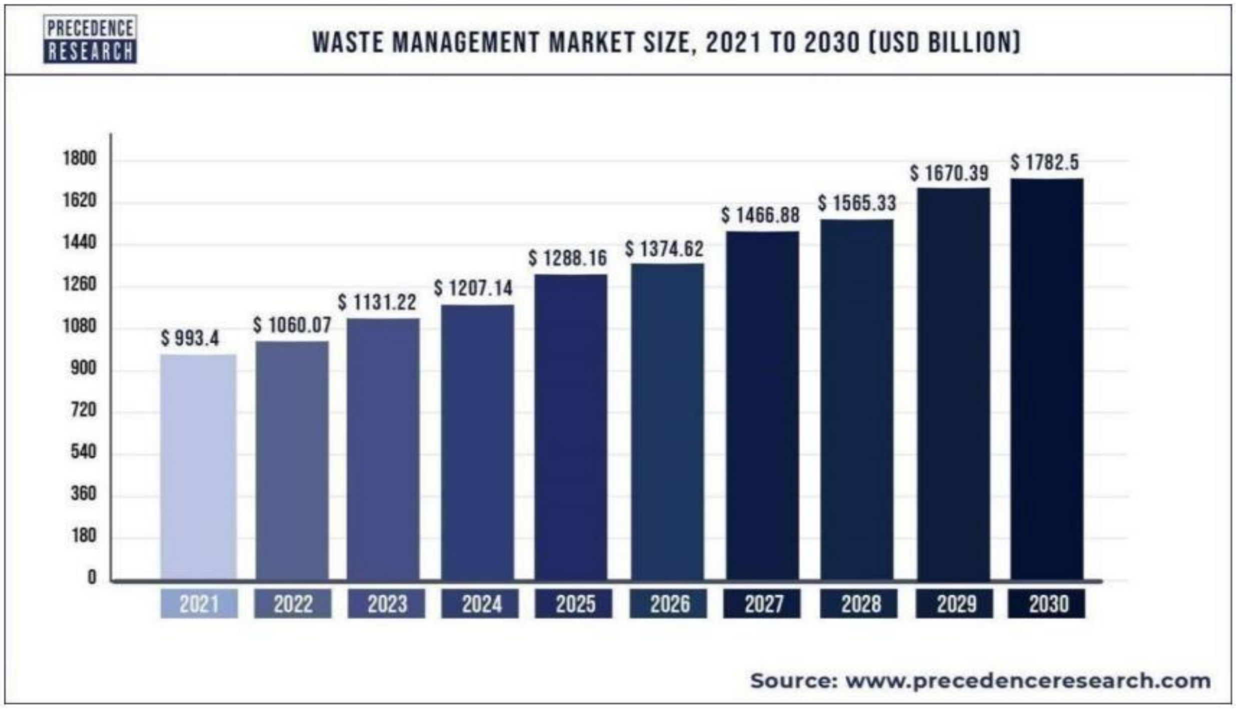
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9. RESULTS

9.1. Performance Metrics

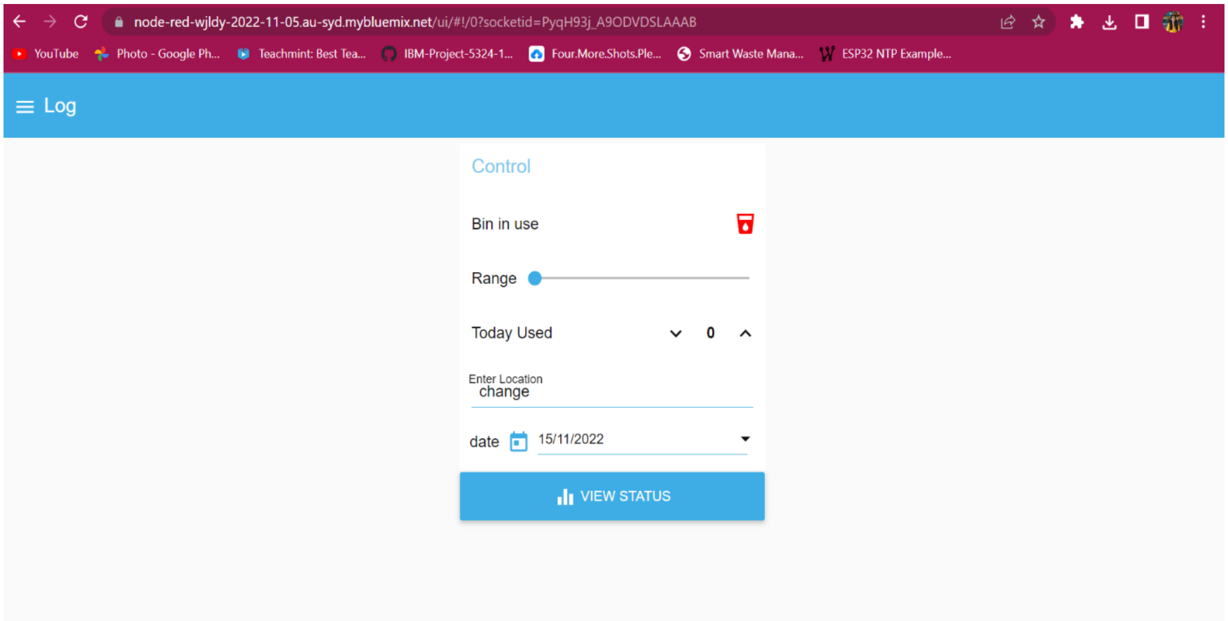


Source: Credence Research Analysis



Source: www.precedenceresearch.com

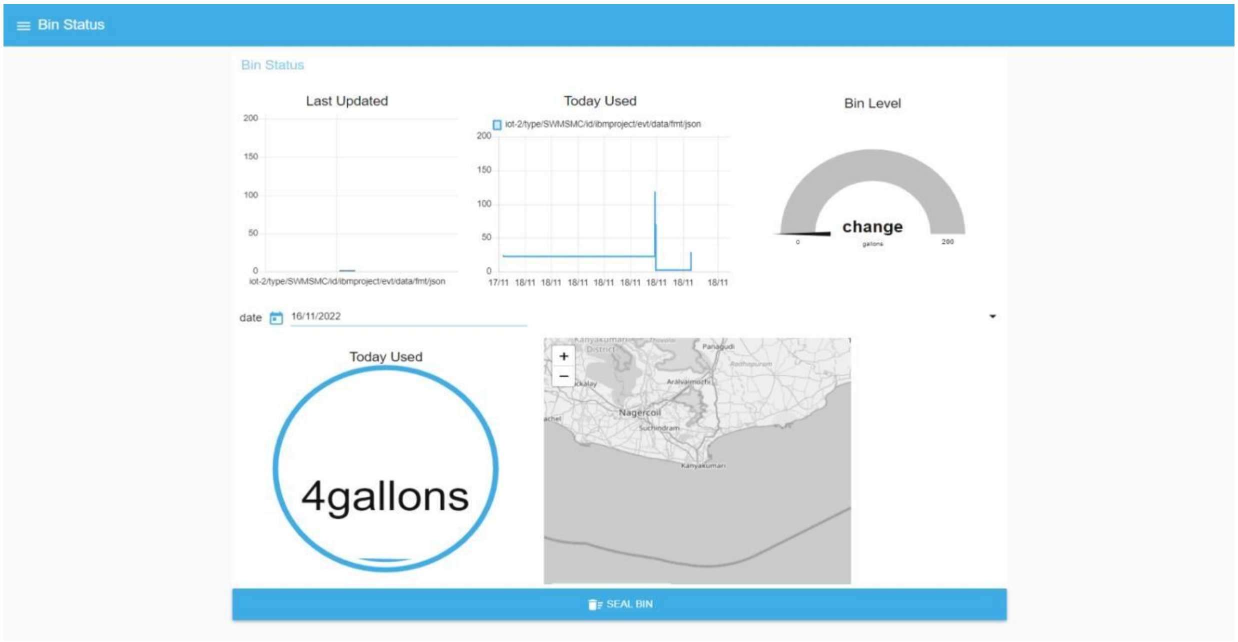
9.2 . Admin Web UI



Today Used 0

Enter Location: change

date 15/11/2022



date 16/11/2022

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 anddual-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 namechar
publishTopic[] = "iot-2/evt/data/fmt/json";
  // topic name and type of event perform and format in which data to be sendchar topic[] = "iot-
2/cmd/led/fmt/String";
  // cmd Represent type and command is test format of stringschar authMethod[] =
"use-token-auth";
  // authentication methodchar
token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;                                   //Client id

// ..........................................................................................................................................................................

WiFiClient wifiClient;                                                                      // creating instance for wificlientPubSubClient
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 okelse 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 okelse 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-5324-1658757871
DEMO LINK : https://youtu.be/nWBZKgQDPpY