Project Report

Smart Waste Management System in Metropolitan Cities

TEAM ID: PNT2022TMID52326

Team members

1. Sujith.R
2. Abins.A
3. Ajees.T
4. Akash.G
5. Binish.C
6. Ramesh.R

From
Stella Mary's College of Engineering, kanyakumari
In fulfilment of project in IBM-NALAYATHIRAN 2022

INDEX

1. INTRODUCTION

1.1 Project Overview

Waste has become a major worry for all of us due to the global population growth and industrialization of nations. Over the years, academics came to the conclusion that, in this age of globalization, waste management alone is insufficient for the efficient treatment and disposal of garbage. Researchers have developed IoT-based Smart Waste Management initiatives and solutions with the aid of technology, ensuring that the time and energy needed to deliver waste management services and lower the amount of waste generated is minimized. Unfortunately, a number of variables, including the socioeconomic context, prevent developing countries from implementing those current solutions. In order to assure effective household garbage disposal, collection, transportation, and recycling while using the fewest resources possible, we have focused our research on creating an intelligent Internet of Things-based waste management system for developing nations like INDIA.

1.2 Purpose

To efficiently establish a secure and sanitary workplace, we combine waste management with technology. Utilizing data and technology to make the trash business more effective is known as smart waste management. Smart trash management, which is based on Internet of Things (IoT) technology, aims to maximize resource allocation, lower operating costs, and improve the sustainability of waste services. This reduces the likelihood that any bin would be full for longer than a week while also enabling the trash collectors who empty the bins to plan more effective routes. The coordination between the waste collectors and the information provided by technology is good. This informs them of the current waste level and prompts them once the trash cans approach the threshold. Thus, smart waste management provides us with the most optimal way of managing the waste in an efficient manner using technology.

2. LITERATURE SURVEY

2.1 Existing problem

In local towns and cities all around the world, waste management has grown to be a serious problem. Municipalities frequently have overflowing local dumpsters without being aware of it. This has a variety of effects on the locals, from the unpleasant odour to the hazardous and unclean environment. Poor waste management, which includes everything from nonexistent collection infrastructure to inefficient disposal, contaminates the air, water, and land. Open and unclean environments can infect people, spread diseases, and lead to the contamination of drinking water. As they accumulate throughout the food chain, toxic substances like persistent organic pollutants (POPs) pose particularly serious dangers

to both human health and the ecosystem. Animals who consume polluted plants receive larger dosages of pollutants than those who are exposed to them directly. Hazardous elements from landfills, agricultural areas, feedlots, etc. will be absorbed by precipitation or surface water seeping through garbage and carried into surface and groundwater. Because it is frequently utilised for drinking, bathing, pleasure, as well as in agricultural and industrial processes, contaminated groundwater also offers a serious health danger. Various pests (insects, rodents, gulls, etc.) that seek out food in waste might be drawn to landfills and waste transfer terminals. These pests pose a threat to human health because they can transmit viruses and bacteria (such as salmonella and e-coli) that cause diseases.

2.2 References

"**IoT Based Smart Garbage alert system using Arduino UNO** ".

(References: Dr. N. Sathish Kumar, B. Vijayalakshmi, R. Jenifer Prarthana, A.Shankar, (2016))

This paper uses tremendous power of RFID technology and presents the development of an electronic monitoring (e-monitoring) system to overcome the problems in the conventional approach. The e-monitoring system is an embedded system that comprises of RFID technology interfaced with Arduino micro-controller and a web base which is completely computerized. For the verification process RFID tag (ID card of the cleaner) interrupts the RFID reader, the ultrasonic sensor checks the status of the dustbin and sends it to the web server. An android application is used to view the alerts and status at the server end.

"**RFID-based Real-time Smart Waste Management System**"

(References: Belal Chowdhury, Morshed U. Chowdhury, (2007) , Australasian Telecommunication Networks and Applications Conference, December, Christchurch, New Zealand.)

In this paper mainly consists of a smart waste (RFID) tag, a Reader, and a waste management IT system (i.e., WMITS). A load cell is used to record the weight of bulk waste from each waste bin. A reader device attached to the PDA (Personal Digital Assistant) or a smart phone placed in waste collector vehicle (garbage/recycling truck) enables the chip to transmit its unique identification to the reader device, allowing the bin to be remotely identified

"**Smart Recycle Bin a Conceptual Approach of Smart Waste Management with Integrated Web based System** ".

(References: Mohd Helmy Abd Wahab, Aeslina Abdul Kadir, Mohd Razali Tomari and Mohamad Hairol Jabbar (2014),)

This paper proposed a Smart Recycle Bin that caters for recycling glass, paper, aluminum can and plastic products. It automatically evaluates the value of the wastes thrown accordingly and provide 3R card. The recycle system enables collection of points for performing a disposal activity into designated recycle bins. Such system encourages recycling activities by allowing the points to be redeemable for products or services.

"**Smart bin: Smart Waste Management System**"

(References: Fachmin Folianto, Yong Sheng Low, Wai Leong Yeow, (2015)
, Tenth International Conference on Intelligent Sensors, Sensor Networks, and Information Processing (ISSNIP) Singapore.)

Proposed Smart bin system has 3 –tier architecture. The ultra sound sensor installed in every Smart bin sense bin fullness and report readings and sensor statuses. The sensor reading is transmitted to the gateway nod which is installed in every sensor cluster. It forwardsthe information to the backend server. The bin sub-system sendsinformation to the workstation and it shows meaningful information to users through a graphical user interface.

"**Sustainable development of smart cities: A systematic review of the literature**"
(References: E. P. Trindade, M. P. F. Hinnig, E. M. da Costa, J. S. Marques, R. C. Bastos, and T. Yigitcanlar, J. Open Innov. Technol. Mark. Complex. vol. 3, no. 3, 2017.)

This paper discusses environmental sustainability and smart city concept. The SLR focuses on theoretical basis concepts of both sustainability and smart city, their relationships, issues, proposed works and strength and weaknesses of related works.

"**A Low Power IoT Sensor Node Architecture for Waste Management Within Smart Cities Context**,"
(References: M. Cerchecci, F. Luti, A. Mecocci, S. Parrino, G. Peruzzi, and A. Pozzebon, Seors, vol. 18, no. 4, p. 1282, Apr. 2018.)
proposed an IoT-based Waste Management System architecture using low-powered sensors as its nodes. This architecture design uses LoRa LPWAN (Low Power Wide Area Network) technology in order to reduce energy consumption thus extending the nodes' battery lifespan. The low power architecture is achieved by implementing no electrical grid connection in the smart bin side, instead, the nodes are expected to be running on batteries or energy storing cells such as solar panels
.

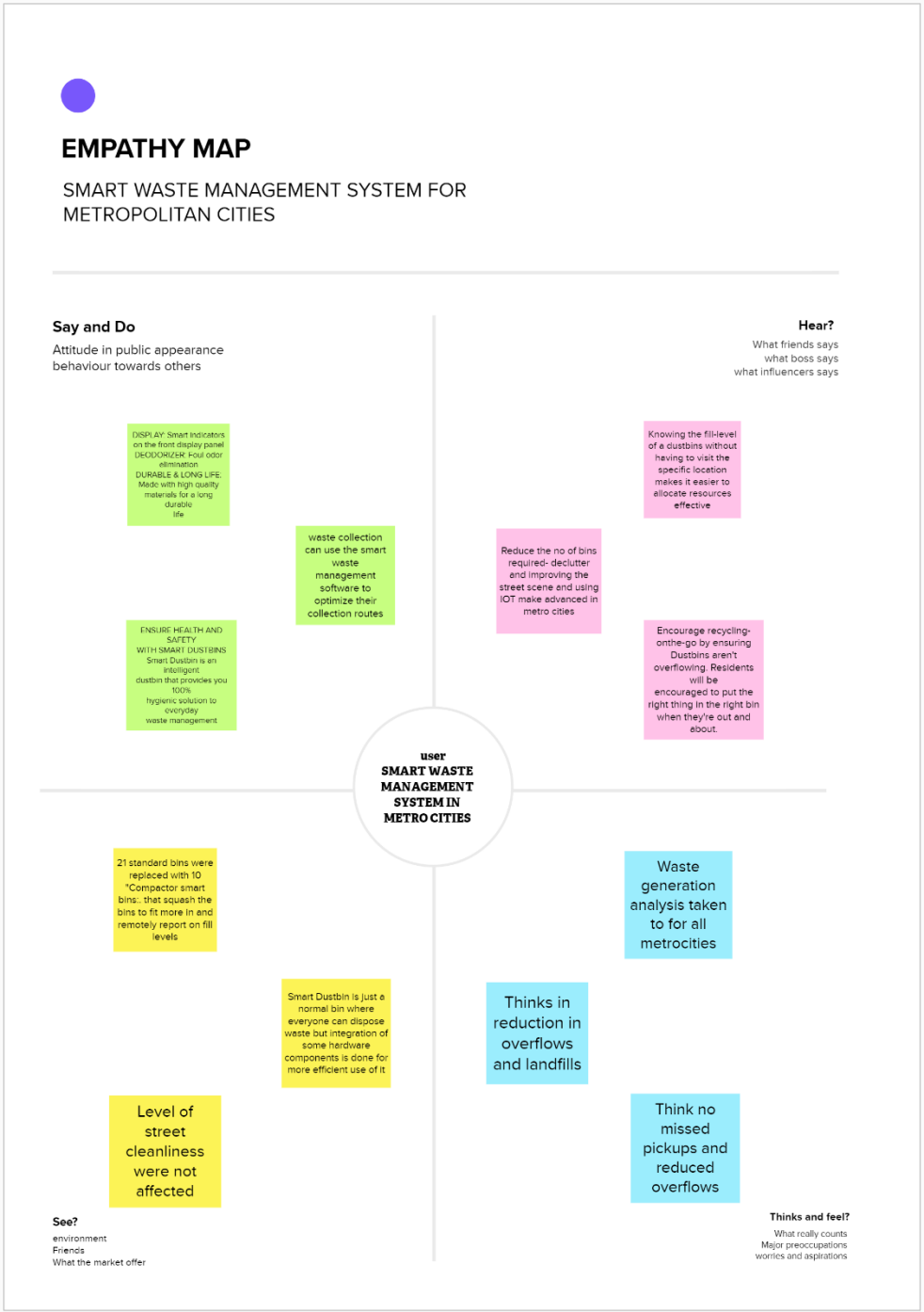
"Smart City Services Monitoring and Waste Collection,"
(References: S. A. Hassan, Near East University, 2016)
This paper proposing a smart city service for monitoring and waste collection using low-cost and open-source technologies. The proposed system is further divided into five subsystems which are Smart Waste System, Local Station, Smart Monitoring and Controlling, Smart Truck System and Smart Monitoring and Controlling Interface.

2.3 Problem Statement Definition

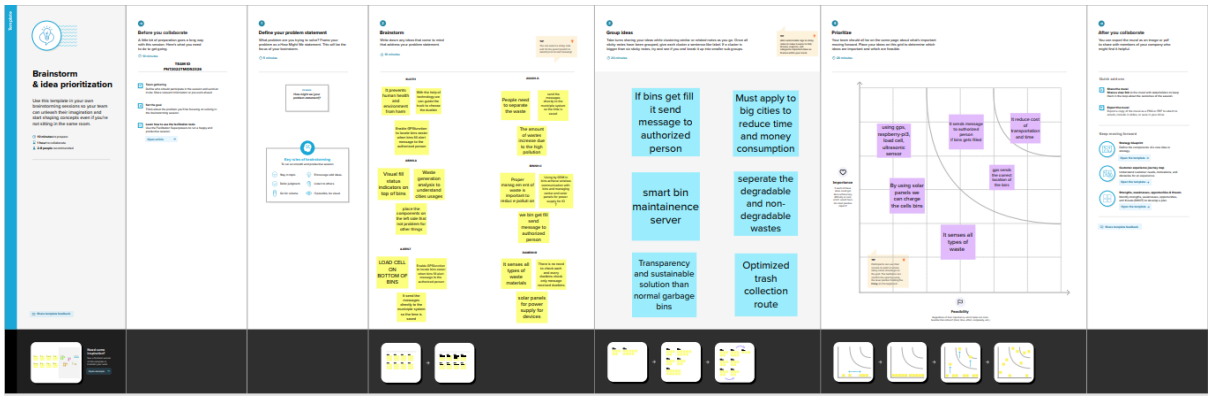
There are a few requirements that must be met for the waste to be managed effectively in our city. If the trash can is full, the garbage collector must arrive automatically to empty it using the weight of the can as well as the vacant space within.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

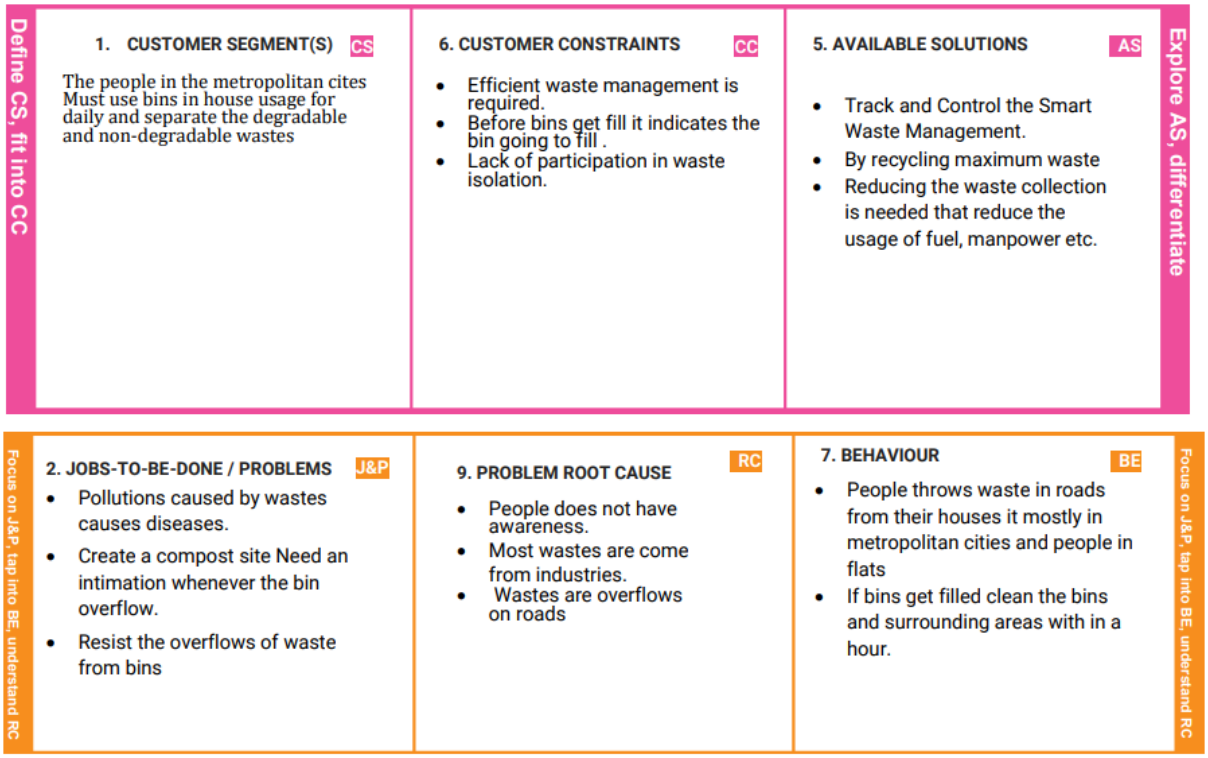


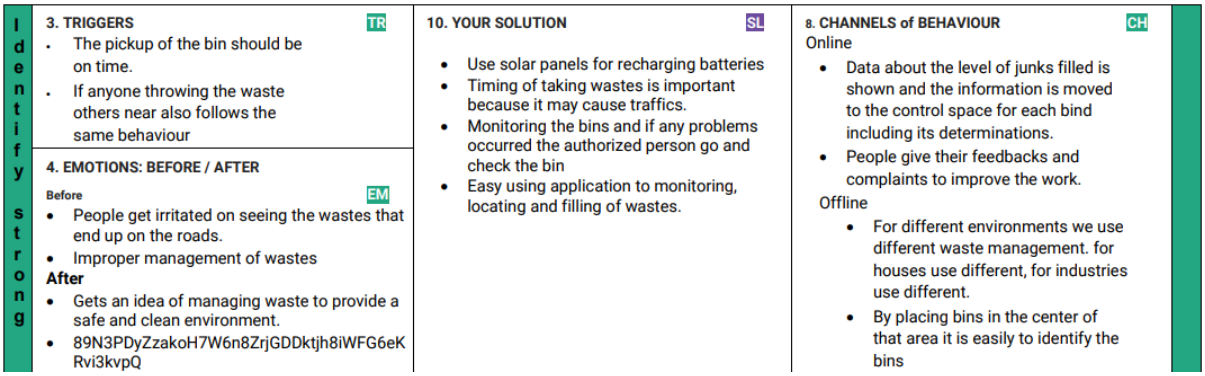
3.3 Proposed Solution

S.No.	Parameters	Description
1	Problem Statement (Problem to be solved)	The collection and disposal of garbage waste is in unordered, inefficient way which leads to overfilling of bins, rotting garbage smell and more fuel consumption of collecting trucks.
2	Idea / Solution description	• The proposed framework would have the option to Computerize the strong waste monitoring process what's more, the executives of the general assortment process utilizing IOT. • The purpose of this project is to focus on problems of detection of emptying of a recycling container using sensor measurements.
3	Novelty / Uniqueness	Using sensors, weighing machine; real time monitoring the level of waste in bins and information is gathered
4	Social Impact/ Customer Satisfaction	From the public insight as most horrendously terrible effects of present strong garbage removal rehearses are seen direct friendly effects, for

		example, neighabourhood of landfills to networks, reproducing of irritations and misfortune in property estimations
5	Business Model (Revenue Model)	Corporate and others, comprising the company's others activities, including its development and operation of landfill gas to energy facilities in the India, and its recycling brokerage services, as well as various corporate
6	Scalability of the Solution	• It helps the government to maintain a clean environment. • The need-driven waste collection eliminates unnecessary traffic blockage.

3.4 Problem Solution fit





4. REQUIREMENT ANALYSIS

4.1 Functional requirement

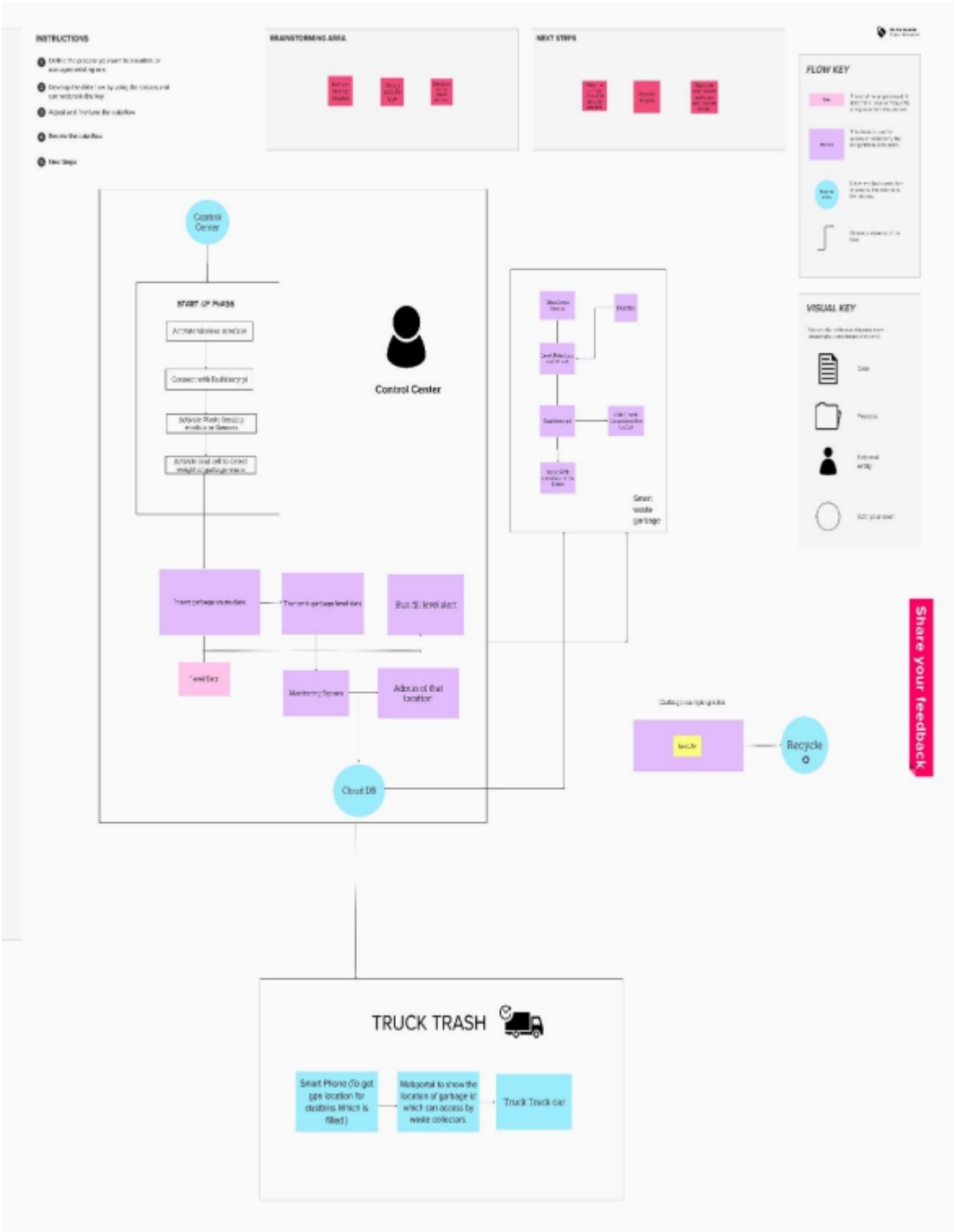
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Detailed bin inventory.	• The Google Street View function allows you to visit any of the monitored bins or stands at any time. They are all visible on the map. • On the map, bins or stands appear as green, orange, or red circles. • The Dashboard displays information about each bin, including its capacity, trash type, most recent measurement, GPS location, and pick-up schedule..
FR-2	Real time bin monitoring.	• The Dashboard shows real-time information on the amount of fill in bins being tracked by sophisticated sensors. • In addition to predicting the percentage of fill, the programme also forecasts when the bin will be full based on previous data, one of the features that even the best waste management software does not offer. • Sensors also recognise picks, allowing you to determine when the bin was last emptied. • You can get rid of the overflowing bins and cease collecting half-empty ones with real-time data and predictions.
FR-3	Expensive bins.	• We assist you in locating containers that increase collection prices. The tool determines a collection cost rating for each bin. • The tool takes local average depo-bin discharge into account. The tool determines the distance from depo-bin discharge and rates bins (1–10).
FR-4	Adjust bin distribution.	• Ensure the best possible bin distribution. Determine which regions have a dense or sparse distribution of bins. • Ensure that each type of waste has a representative stand. • You can make any necessary adjustments to bin position or capacity based on past data.

4.2 Non-Functional requirements

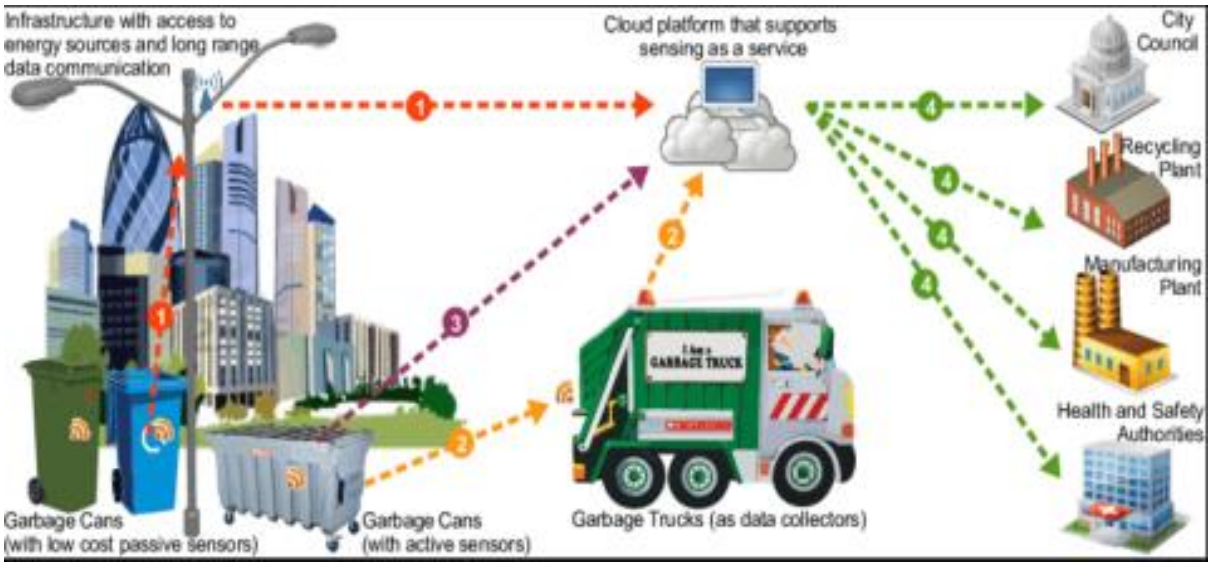
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Separate bins for recyclable and non
NFR-2	**Security**	Data fetched can be only accessed by authorized user
NFR-3	**Reliability**	Accurate data and availabity is displayed
NFR-4	**Performance**	Detects and intimates alerts on reaching 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 Diagrams


INSTRUCTIONS
BRAINSTORMING AREA
NEXT STEPS
FLOW KEY
VISUAL KEY
Control Center
START UP PHASE
Control Center
Recycle
Share your feedback
Cloud DB
Admin of that location
TRUCK TRASH
Truck Truck car

5.2 Solution & 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 & SCHEDULING

6.1 Sprint Planning & Estimation

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	8th SEPTEMBER
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem	9th SEPTEM BER
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	15th SEPTEMBER
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model,	19th SEPTEMBER 2022

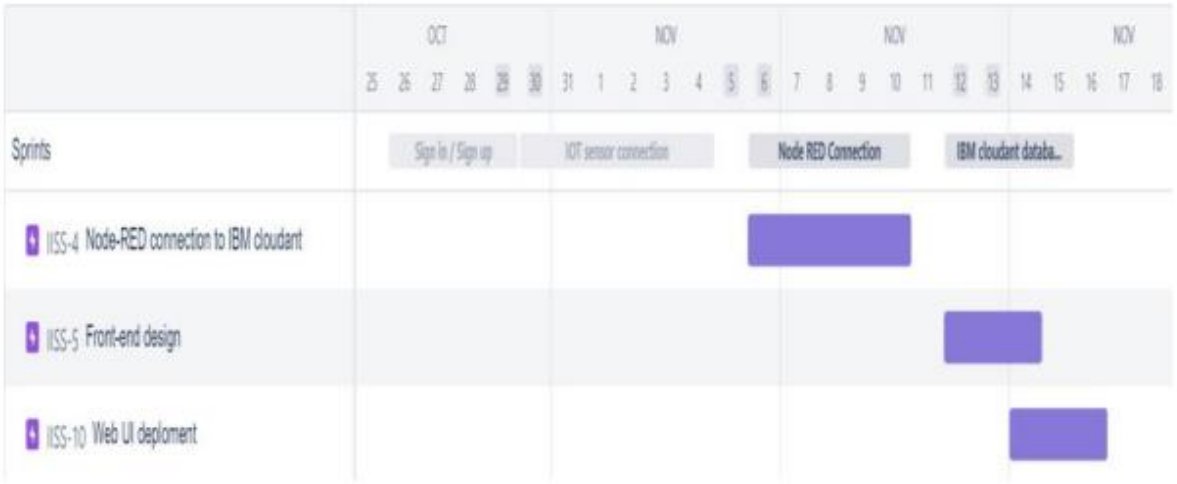
	social impact, scalability of solution, etc.	
Problem Solution Fit	Prepare problem - solution fit document.	1 OCTOBER 2022
Solution Architecture	Prepare solution architecture document.	3 OCTOBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application	21 OCTOBER 2022
Functional Requirement	Prepare the functional requirement document	29 0CTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review	1 NOVEMBER 2022
Technology Architecture	Prepare the technology architecture diagram	5 NOVEMBER 2022
Prepare Milestone & Activity	Prepare the milestones & activity list of the project	9 NOVEMBER 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

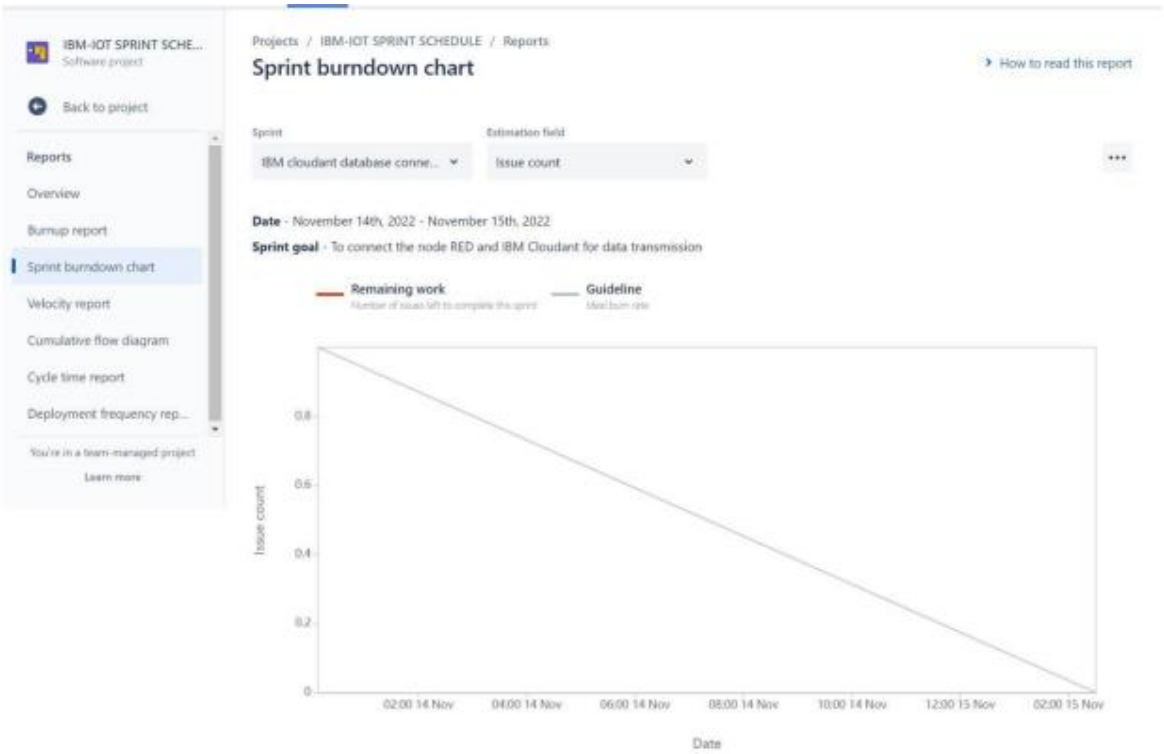
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	4 Days	26 Oct 2022	29 Oct 2022	20	30 Oct 2022
Sprint-2	20	4 Days	31 Oct 2022	04 Nov 2022	20	05 Nov 2022
Sprint-3	20	4 Days	06 Nov 2022	10 Nov 2022	20	11 Nov 2022
Sprint-4	20	4 Days	12 Nov 2022	16 Nov 2022	20	17 Nov 2022

6.3 Reports from JIRA

ROADMAP



SPRINT BURNDOWN CHART



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

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))
{
}
else
{
}
}
Serial.println("IBM subscribe to cmd OK");
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");
18 | P a g e
Serial.println("Lid Closed");
lcd.print("Full! Don't use");
delay(2000);
```

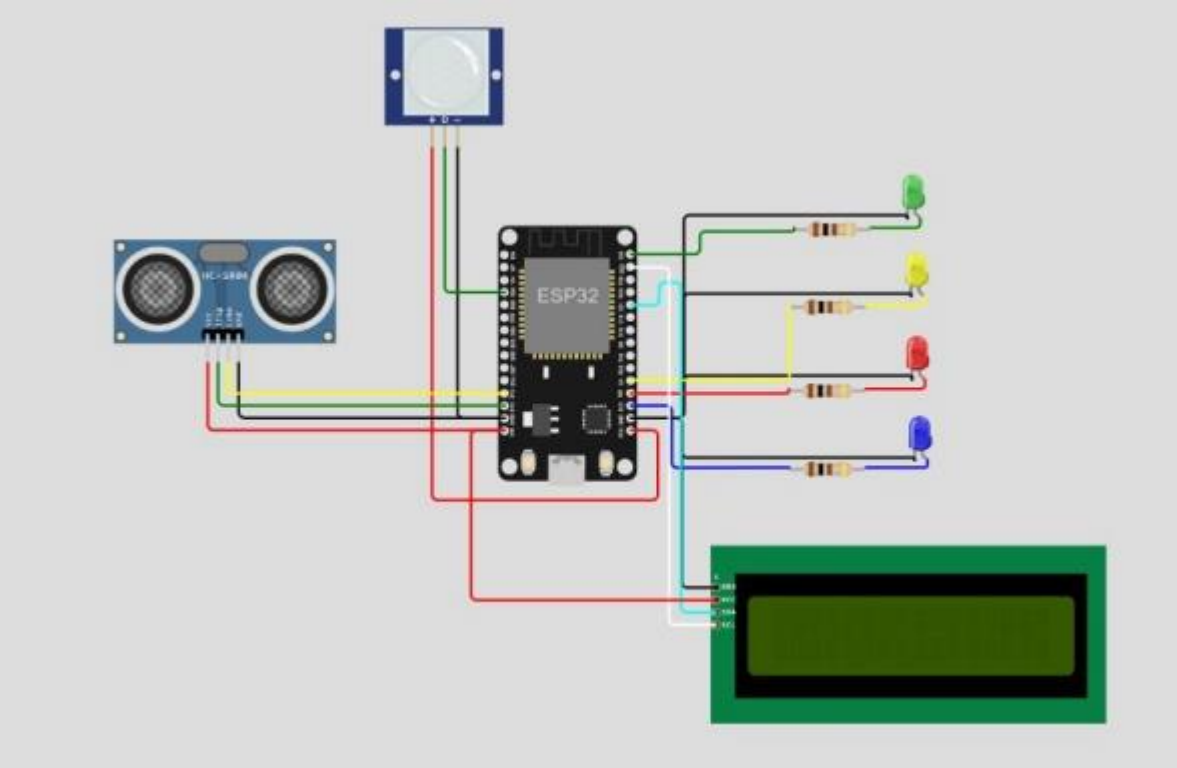
```
lcd.clear();
digitalWrite(4, LOW);
digitalWrite(23, LOW);
}
else if(cm > 100 && cm < 180)
{
digitalWrite(4, HIGH);
Serial.println("Warning!!,Trash is about to cross 50% of bin level");
digitalWrite(2, LOW);
digitalWrite(23, LOW);
}
else if(cm > 180)
{
digitalWrite(23, HIGH);
Serial.println("Bin is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



8. TESTING

8.1 Test Cases

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

8.2 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).

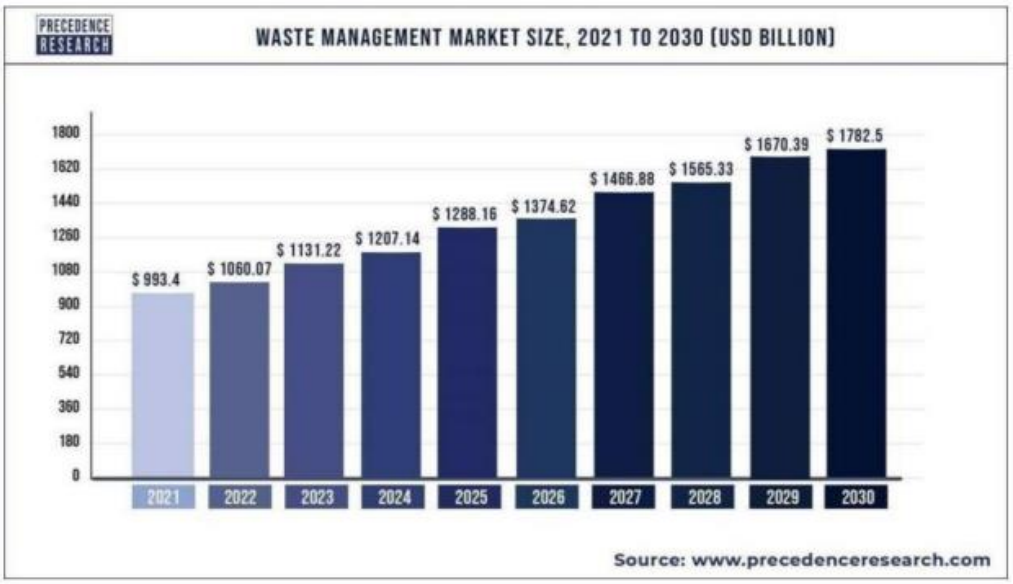
2. Defect Analysis

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

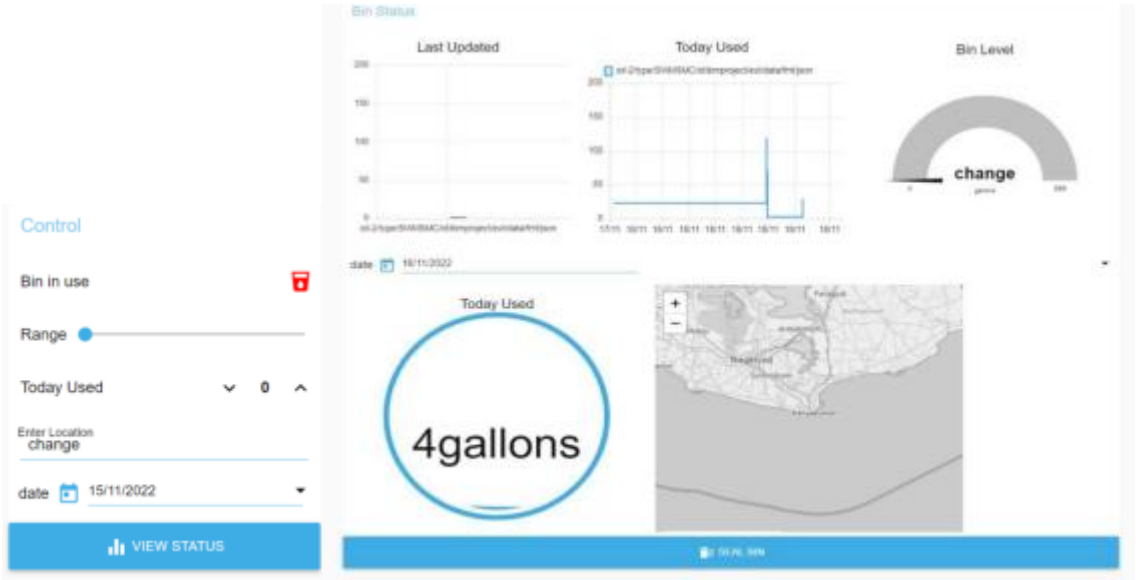
resolved.

9. RESULTS

9.1 Performance Metrics



9.2 ADMIN WEB UI



ADVANTAGES

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

DISADVANTAGES

- According to the city's population, the system needs more trash cans for separate garbage collection.
- This has a high initial cost because smart dustbins are more expensive than other techniques. The dustbin sensor nodes only have a little amount of memory.

11. CONCLUSION

By employing sensors to track the filling of bins, a Smart Waste Management system that is more effective than the one now in use can be created. Our idea of a "smart waste management system" focuses on tracking waste management, providing intelligent technology for waste systems, doing away with human intervention, reducing human time and effort, and creating a clean, healthy environment. In smart cities where citizens have hectic schedules that provide little time for garbage management, the suggested solution can be put into practise. If desired, the bins might be placed in a city where a big enough container could carry enough solid waste for one unit. The cost could be very 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.

28 | P a g e

• 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.

1. Memory: 320 KiB
2. SRAM CPU: Tensilica Xtensa LX6 microprocessor @ 160 or 240 MHz
3. Power: 3.3 V DC
4. Manufacturer: Espressif Systems
5. Predecessor: ESP8266

• Sensors :

a. 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.

b. 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))
{
}
else
{
}
}
Serial.println("IBM subscribe to cmd OK");
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 & Project Demo Link

Link: https://github.com/IBM-EPBL/IBM-Project-51826-1660985525