Nalaiya Thiran

Batch No: B5 – 5M1E

Smart Waste Management System For Metropolitan Cities

Team ID: PNT2022TMID43463

Team Members:

720319106001	Aakash P
720319106002	Anitha C
720319106003	Aravind M
720319106005	Boopathi G

CHAPTER 1: INTRODUCTION

1.1 Project Overview

Smart waste management is an innovative approach to handling and collecting waste. Based on IoT (Internet of Things) technology, smart waste management provides data on waste generation patterns and behaviour.

Our Smart waste management solution uses sensors placed in garbage bins to measure fill levels and notifies city collection services when bins are ready to be emptied. There are load and ultrasonic sensors placed to continuously monitor the bins. This data is sent to the cloud (via a microcontroller that is connected to Wi-Fi) where it is stored after which it is processed further. When the levels exceed a certain limit, a notification is sent to the garbage collector via a web application.

Over time, historical data collected by sensors can be used to identify fill patterns, optimize driver routes and schedules, and reduce operational costs. The cost of these sensors is steadily decreasing, making IoT waste bins more feasible to implement and more attractive.

1.2 Purpose

Around 2.1 billion tonnes of municipal solid waste is generated annually around the globe. Population growth and rapid urbanization lead to a huge increase in waste generation, so the traditional methods of waste collection have become inefficient and costly. This system cannot measure the fullness levels of containers, and as a result, half-full containers can be emptied, and in contrast, pre-filled ones need to wait until the next collection period comes. Moreover, since drivers collect empty bins, predefined collection routes of the

system cause waste of time, an increase in fuel consumption, and excessive use of resources.

In today's ever-technological world, an innovative and data-driven approach is the only way forward, the waste sector needs a solution that empowers event-driven waste collection. The most efficient way this extraordinary amount of waste can be solved is through smart waste management without obsolete methods of waste collection. This empowers municipalities, cities, and waste collectors to optimize their waste operations, become more sustainable, and make more intelligent business decisions.

CHAPTER 2: LITERATURE SURVEY

2.1 Existing Problem

Around 80% of waste collections happen at the wrong time. Late waste collections lead to overflowing bins, unsanitary environments, citizen complaints, illegal dumping, and increased cleaning and collection costs. Early waste collections mean unnecessary carbon emissions, more traffic congestion, and higher running costs. The old way of doing waste management is highly inefficient. And in today's ever-technological world, an innovative and data-driven approach is the only way forward. Traditionally, municipalities and waste management companies would operate on a fixed collection route and schedule. This means that waste collection trucks would drive the same collection route and empty every single waste container – even if the waste container did not need emptying. This means high labour and fuel costs – which residents ultimately foot the bill for.

2.2 References

Paper Title	Author	Outcome

IOT based Smart Garbage System	1) T.Sinha 2) R.M. Sahuother	IoT Based Smart Garbage System which indicates directly that the dustbin is filled to a certain level by the garbage and cleaning or emptying them is a matter of immediate concern. This prevents lumping of garbage in the roadside dustbin which ends up giving foul smell and illness to people. The design of the smart dustbin includes a single by ultrasonic sensor which configured with Arduino Uno with this research, it is sending SMS to the Municipal Council that dustbin is to overflow.
Raspberry pi-based smart waste management system using Internet of Things.	1)Shaik Vaseem Akram 2)Rajesh Singh	Nowadays it is becoming a difficult task to distinguish wet and dry waste. The new waste management system covers several levels of enormous workforce. Every time, laborers must visit the garbage bins in the city area to check whether they are filled or not. The data communicates to the cloud server for real-time monitoring of the system. With the real-time fill level information collected via the monitoring platform, the system reduces garbage overflow by informing about such instances before they arrive

Smart Waste Management System.	1) Sanjiban Charkraborty	This Waste management is one of the serious challenges of the cities, the system now used in cities, we continue to use an old and outmoded paradigm that no longer serves the entail of municipalities, Still find over spilled waste containers giving off irritating smells causing serious health issues and atmosphere impairment.
Smart Solid Waste Management.	1) Mohd Helmy Abd Wahab	At the time of trash disposal, the material to be recycled could be identified using RFID technology.
Analysis of Load cell.	1) Ranjeet Kumar 2) Sandeep Chhabra	Load Cells 4.1 General Load Cell related information A load cell is meant to measure the size of a mass but actually is a force sensor which transforms force into an electrical signal. The load cell needs the earth gravity to work. Every mass is attracted by the earth gravimetric field, that force is named “load”.
Smart Waste Management using Wireless Sensor Network	1) Tarandeep Singh 2) Rita Mahajan 3) Deepak Bagai	In most of the places, garbage bins are not cleaned at periodic intervals, giving a hygienic issue. Thus, a system to manage bins, by using intelligent bins, gateway and remote base station is created. But this system is prone to attacks from hackers and complexity to build it is very high.

Smart Waste Management for Green Environment	1) T. P. Fei	The system is based on Bootstrap platform. This system works on the waterfall methodology which has 4 crucial phases: planning and analysis, system design, system implementation and system testing. Using this system, operators can get the information regarding collection from trash bins. The limitations of this approach are that the resultant product has a short life and uniformity is lost after a certain period.

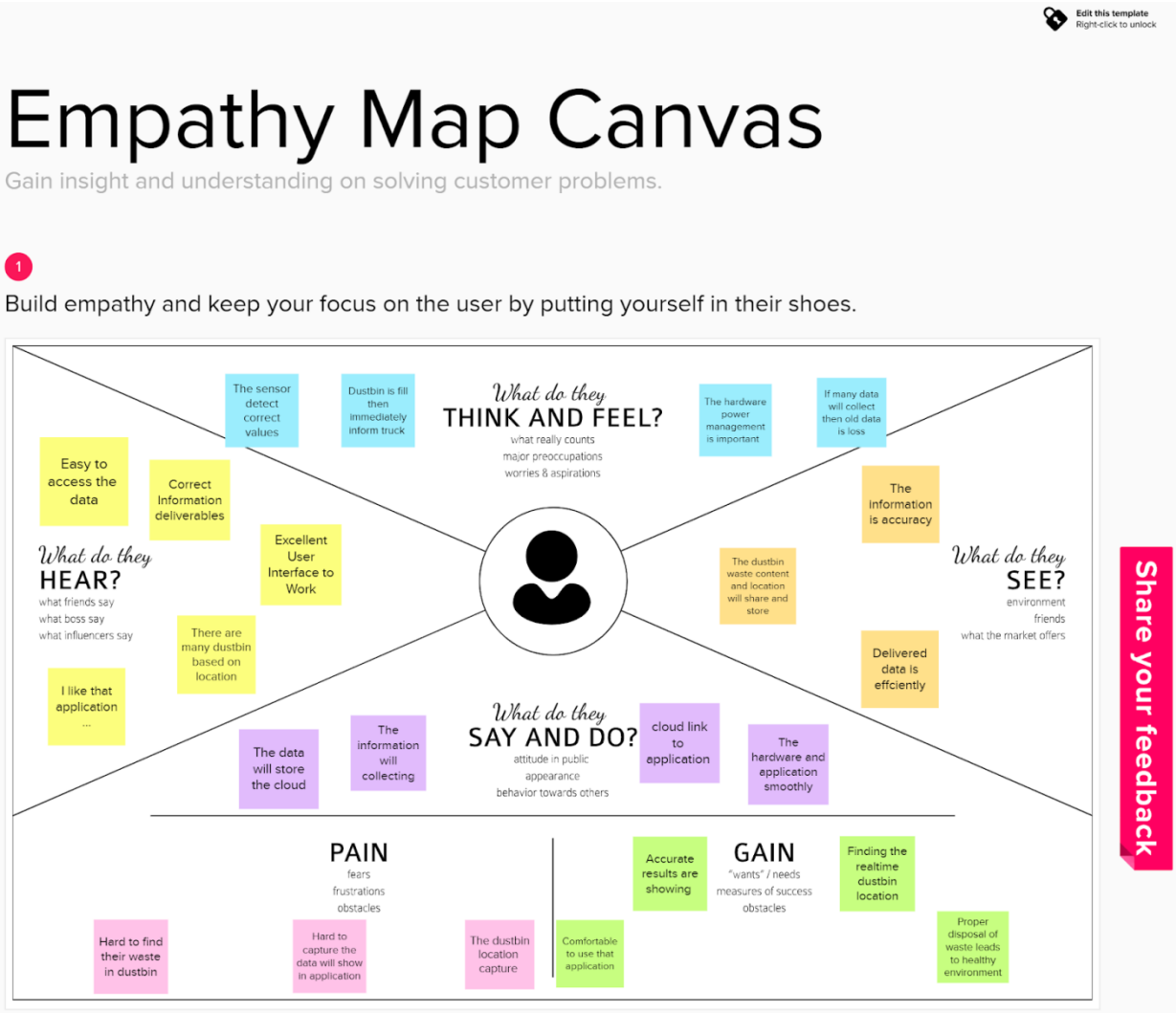
2.3 Problem Statement Definition

Urban India generates tonnes of wastes annually. Our country faces major challenges associated with waste management. Conventional garbage collection is not efficient since the authorities are not notified until the waste bin is full, and this leads to overflow of waste material. Efficient waste disposed of garbage is essential for a sustainable and clean India.

This project smart waste management using IOT based waste bin for collection and monitoring the level of waste inside the bin. The system is implemented using two ultrasonic sensors which are controlled by Node MCU. One of the ultrasonic sensors detects the level of the waste in the bin and another detects the person approaching the bin to dispose of the waste. This detection helps in automatic opening and closing of the lid. Servo motor is connected to the lid which serves the action of closing and opening of the lid. In this system, the level of waste in the bin will be sent to concerned authorities. The IOT data is stored and monitored using an app.

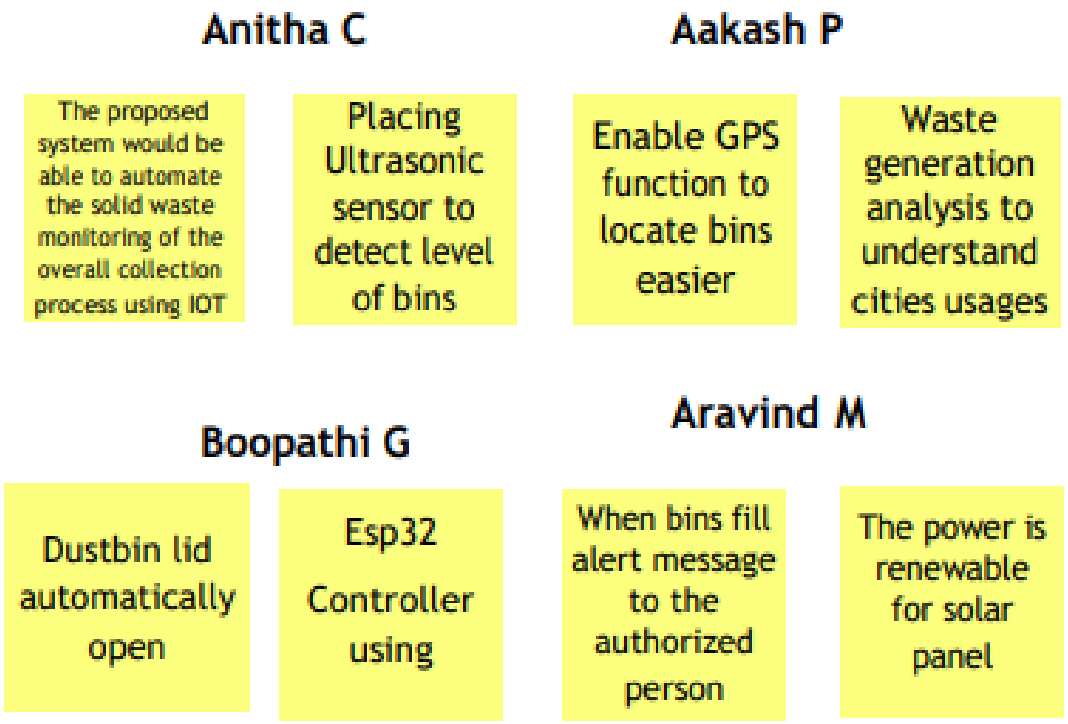
CHAPTER 3: IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

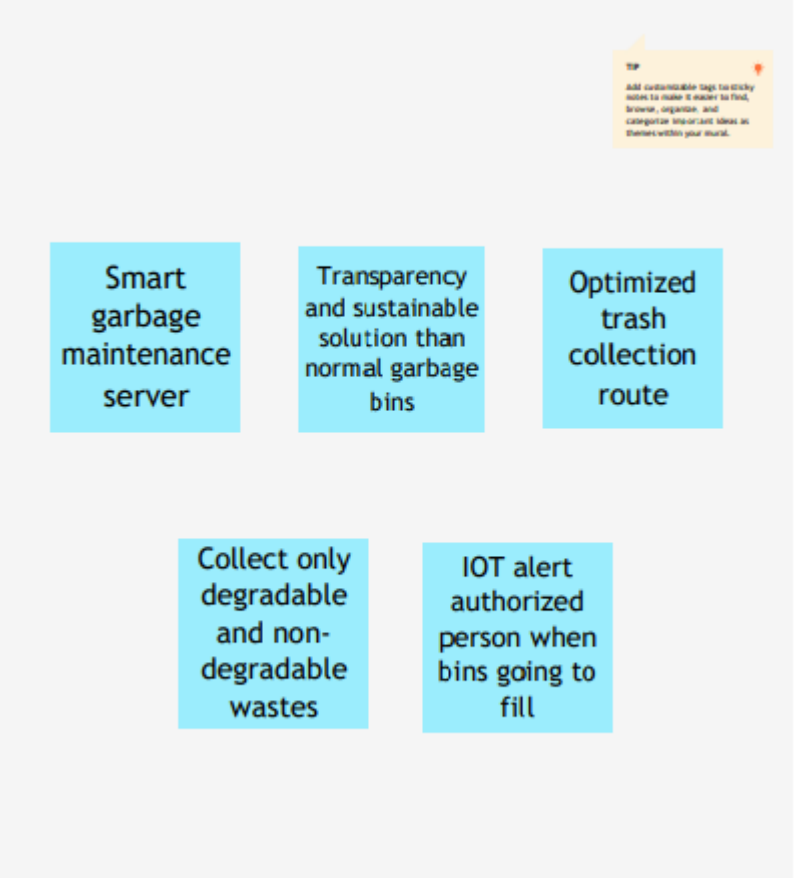


3.2 Ideation & Brainstorming

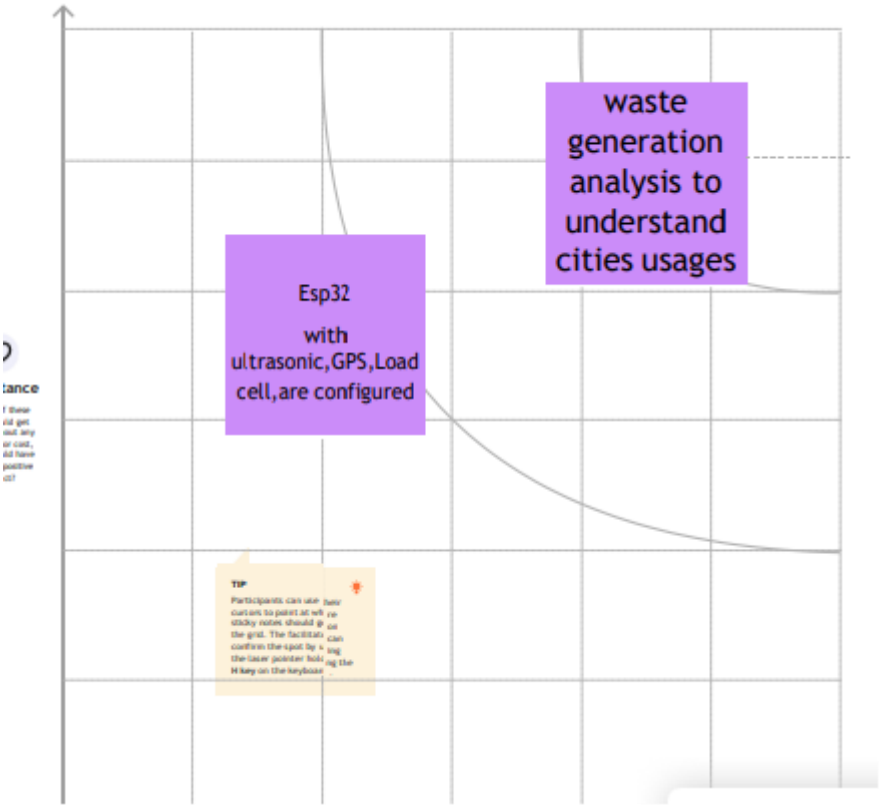
3.2.1 Brainstorm by team members



3.2.2 Group ideas



3.2.3 Prioritize

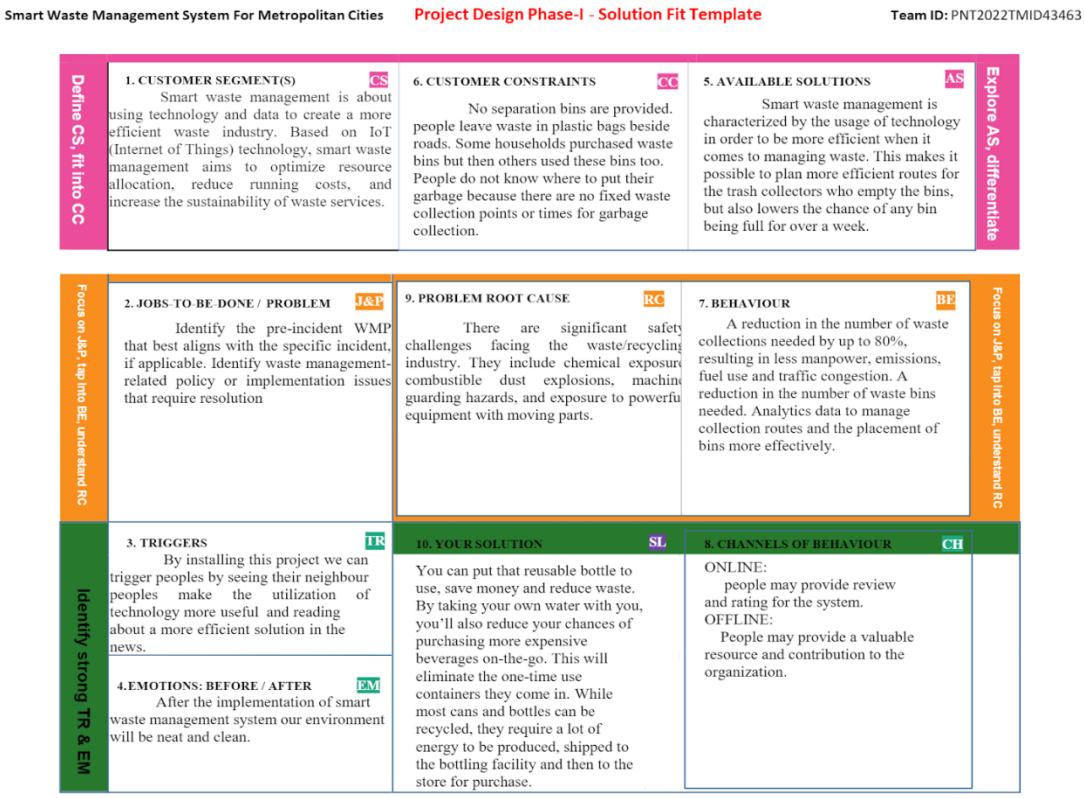


3.3 Proposed Solution

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Solid waste disposal is a significant problem in metropolitan areas of most developing nations, and it seriously jeopardizes the residents' ability to live a healthy lifestyle. Both the local authorities and the public will benefit from having access to trustworthy information on the state of solid trash at various sites throughout the city for managing the threat.
2.	Idea / Solution description	A 24/7 monitoring system is designed for the bins consisting of a microcontroller and embedded Wi-Fi module and has sensors for detecting the weight and level of the waste. When the bin is full, information is sent to the cloud which is forwarded to the users via the web application. Once the message is sent, garbage collection is initiated for all the bins whose level has risen more than the threshold value.
3.	Novelty / Uniqueness	Garbage collection is made simple and efficient with the help of this web application, which can be used to monitor the bins throughout the city. The fill level of the bin is also displayed which makes it easier for garbage collectors. Additionally, it alerts the location of the bin to the garbage collectors.
4.	Social Impact / Customer Satisfaction	Large overflowing bins are a potential threat because they not only pollute the air nearby but also serve as a breeding ground for contagious diseases. Also, the waste collection process is more effective for the garbage collector. Normally, they might see a bin that is overflowing and is difficult for them to collect or one that is

		only partially filled, but with the help of this application, this problem is resolved.
5.	Business Model (Revenue Model)	Without any financial advantages, the primary goal of this solution is to assist locals and government employees. Recycling dry waste and composting wet waste, which could then be sold, are two ways to generate income if it is necessary.
6.	Scalability of the Solution	The waste collection for an entire major metropolitan area should be supported by the platform. The implementation of this will consider various implementation-related factors, including the storage and security of data in the cloud.

3.4 Proposed Solution fit



CHAPTER 4: REQUIREMENT ANALYSIS

4.1 Functional Requirements

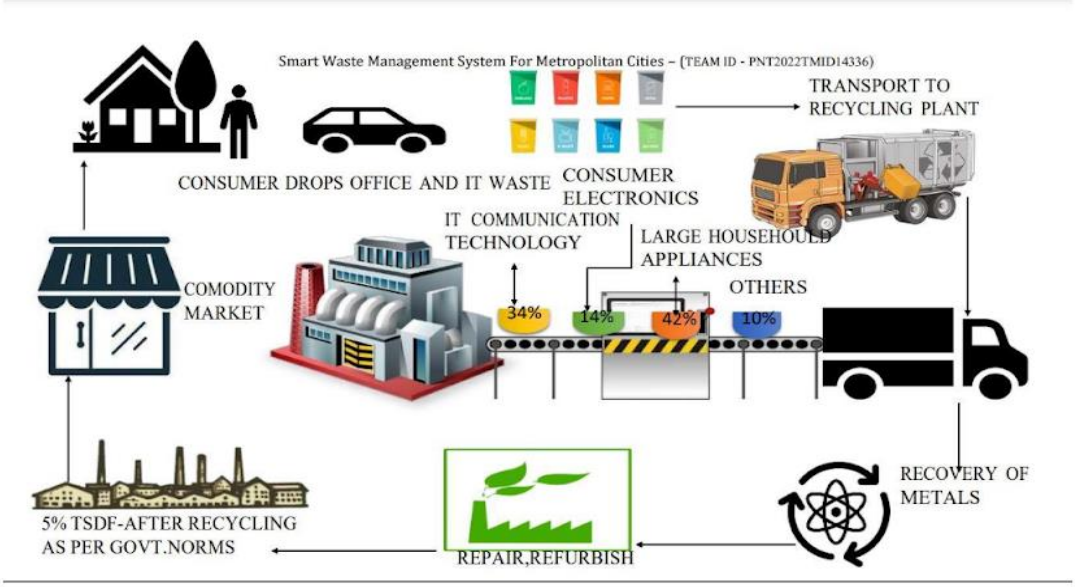
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
1	Detailed bin inventory.	All monitored bins can be seen on the map, such that the route can be optimized for the garbage collectors. Bins or stands are visible on the map as green, orange, or red circles. You can see bin details such as – capacity, last measurement, etc.
2	Real time bin monitoring.	The amount of fill is displayed in %, based on the garbage level and the tool predicts when the bin will become full, which is one of the functionalities not included in the best waste management software. Sensors recognize picks as well; so, we can check when the bin was last collected. With real-time data and predictions, you can eliminate the overflowing bins and stop collecting half-empty ones.
3	Plan waste collection routes.	The tool semi-automated waste collection route planning. Based on current bin fill-levels and predictions of reaching full capacity, you are ready to respond and schedule waste collection. You can compare planned vs. executed routes to identify any inconsistencies.
4	Adjust bin distribution.	Ensure the most optimal distribution of bins. Identify areas with either dense or sparse bin distribution. Make sure all trash types are represented within a stand.
5	Eliminate inefficient picks.	Eliminate the collection of half-empty bins. The sensors recognize picks. By using real-time data on fill-levels and pick recognition, we can show you how full the bins you collect are.

4.2 Non-functional Requirements

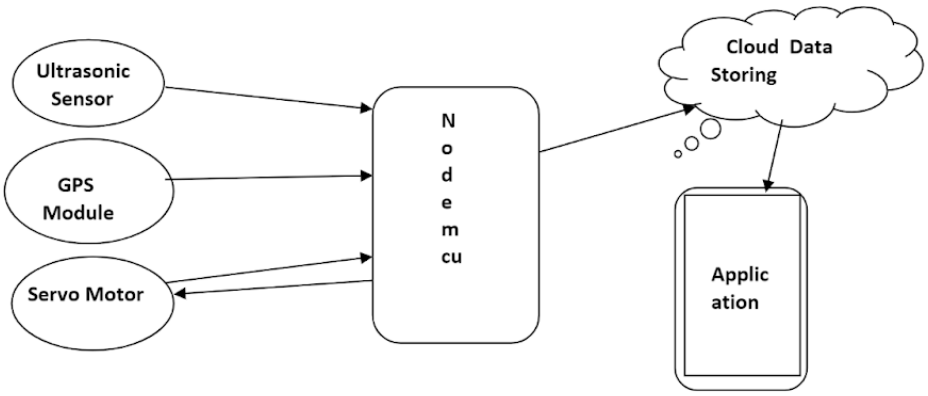
NFR No.	Non-Functional Requirement	Description
1	Usability	IoT device verifies that usability is a special and important perspective to analyze user requirements, which can further improve the design quality. In the design process with user experience as the core, the analysis of users' product usability can indeed help designers better understand users' potential needs in waste management, behavior and experience.
2	Security	● Use of reusable bottles ● Use of reusable grocery bags ● Purchase wisely and recycle ● Avoid single use food and drink containers
3	Reliability	Smart waste management is also about creating better working conditions for waste collectors and drivers. Instead of driving the same collection routes and servicing empty bins, waste collectors will spend their time more efficiently, taking care of bins that need servicing.
4	Performance	The Smart Sensors use ultrasound technology to measure the fill levels (along with other data) in bins several times a day. Using a variety of IoT networks (NB-IoT, GPRS), the sensors send the data to IBM Watson, that contains all the devices. Customers are hence provided data-driven decision making, and optimization of waste collection routes, frequencies, and vehicle loads resulting in route reduction by at least 30%

CHAPTER 5: PROJECT DESIGN

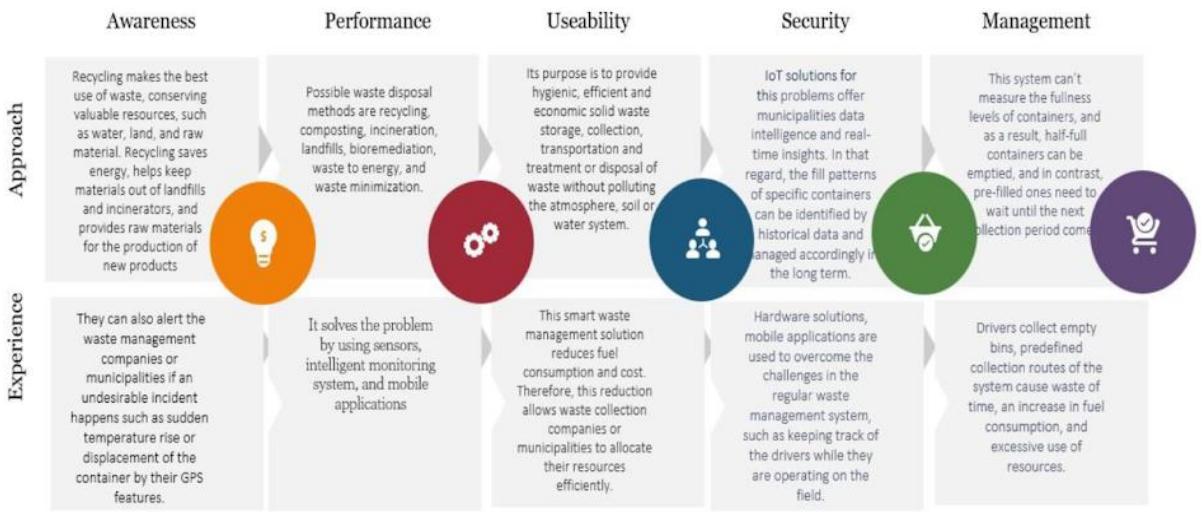
5.1 Data Flow Diagram:



5.2 Solution and Technical Architecture



5.3 User Stories



CHAPTER 6: PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation:

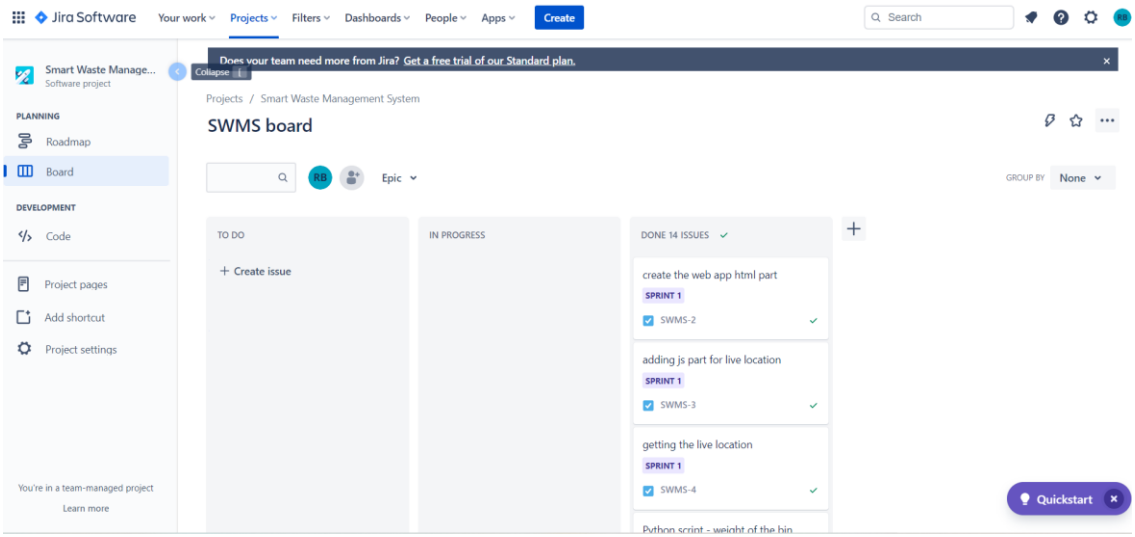
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Hardware+software	USN-1	The hardware & sensors connect to ibm iot Watson	20	High	Aakash P
Sprint-2	Mit App,webapp	USN-2	Create mit app and the cloud app connect the mit app.show that value	20	High	Anitha C
Sprint-3	Dashboard	USN-3	It will find shortest path between truck and bin.	20	Medium	Aravind M
Sprint-4	Dashboard	USN-4	As a authority officer ,I'll make sure everything is proceeding as planned and without any problem.	20	Medium	Boopathi G

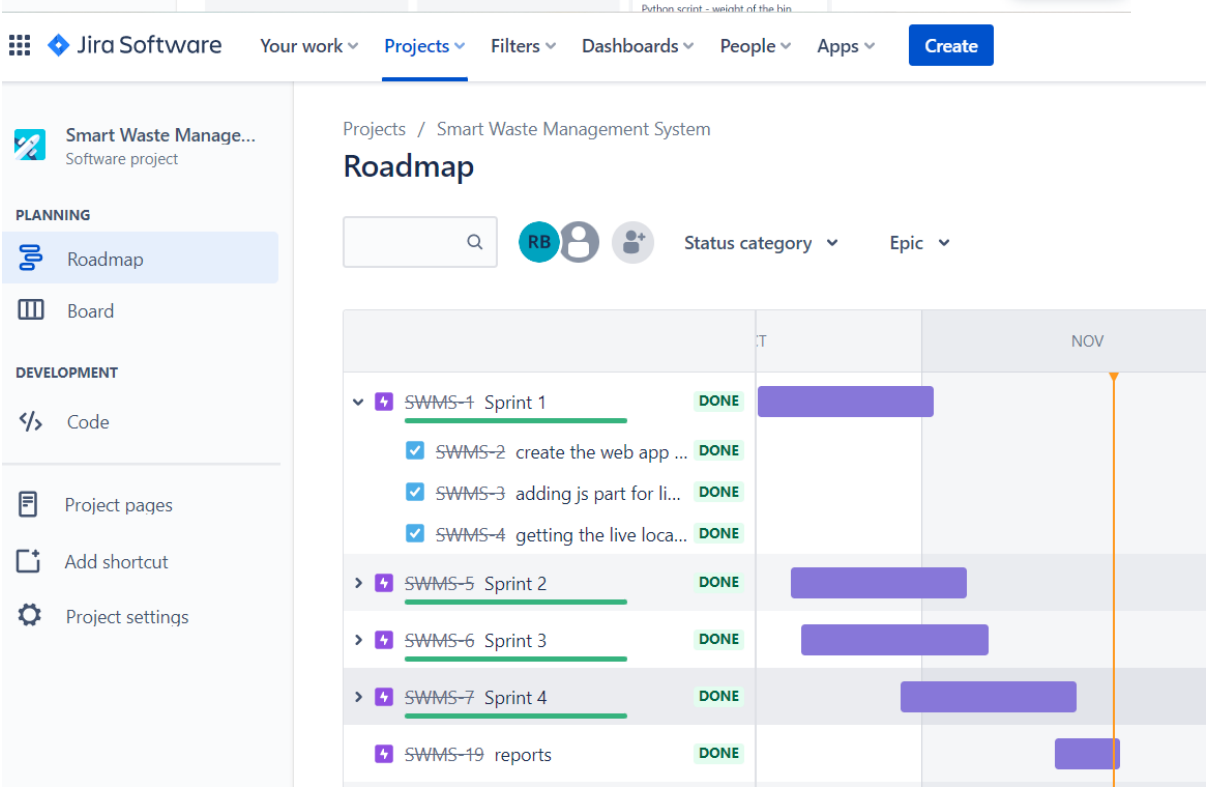
6.2 Sprint Delivery Schedule:

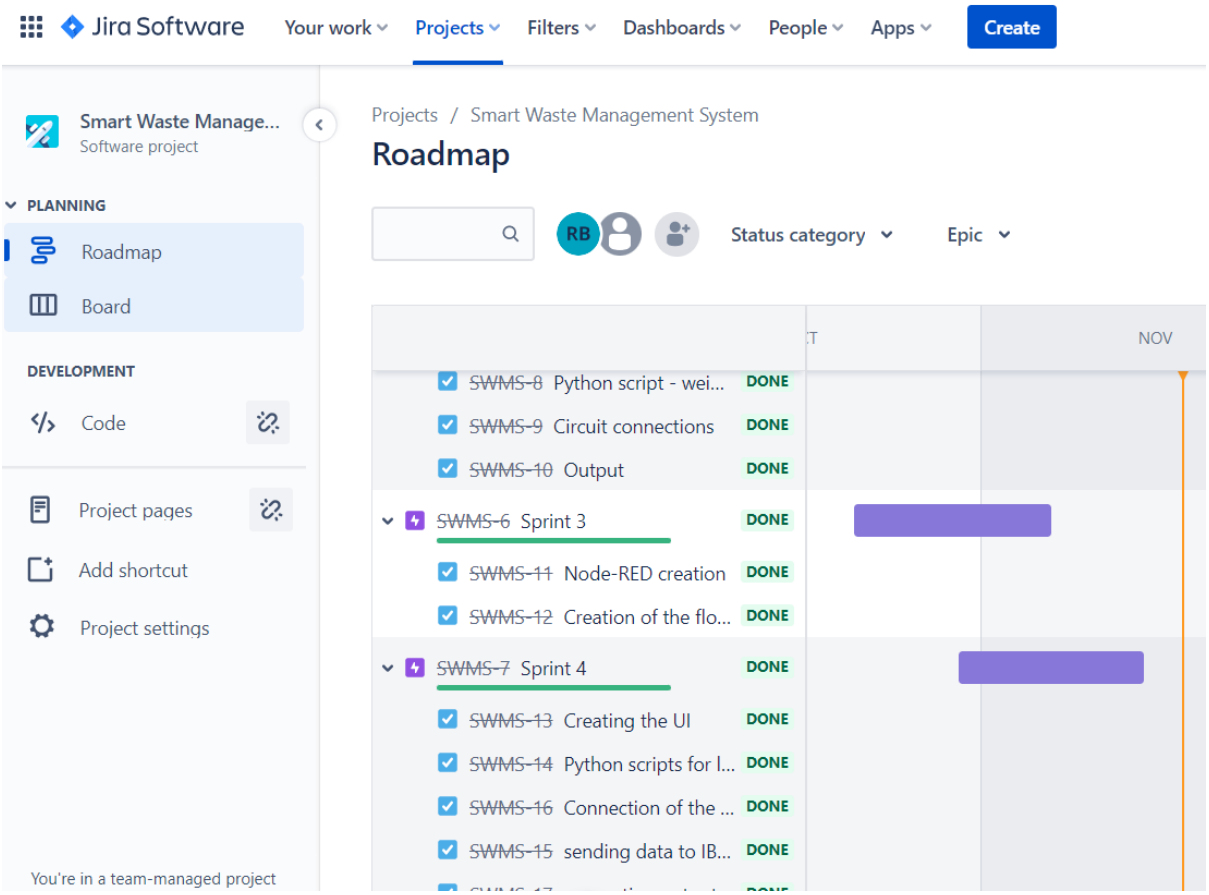
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022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

Sprint-4	**20**	**6 Days**	**14 Nov 2022**	**19 Nov 2022**	**20**	**19 Nov 2022**

6.3 Reports from JIRA:






Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Smart Waste Manage...
Software project
PLANNING
Roadmap
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Projects / Smart Waste Management System
Roadmap
RB
Status category
Epic
NOV
SWMS-8 Python script - wei... DONE
SWMS-9 Circuit connections DONE
SWMS-10 Output DONE
SWMS-6 Sprint 3 DONE
SWMS-11 Node-RED creation DONE
SWMS-12 Creation of the flo... DONE
SWMS-7 Sprint 4 DONE
SWMS-13 Creating the UI DONE
SWMS-14 Python scripts for l... DONE
SWMS-16 Connection of the ... DONE
SWMS-15 sending data to IB... DONE

CHAPTER 7: CODING AND SOLUTIONING

Code:

```
#include <WiFi.h>
#include <PubSubClient.h>
#include<Servo.h>

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

//-------credentials of IBM Accounts------

#define ORG "woaev0"//IBM ORGANITION ID
#define DEVICE_TYPE "abcde"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "112345"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "1a2b3c4d"
String data3;
float d,d1,w;

Servo Myservo;
int pos;

#define trigpin    18
#define echopin    5

#define trigpin1 25
#define echopin1 33

//-------- Customise the 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 subscribetopic[] = "iot-2/cmd/display/fmt/String";// cmd  REPRESENT
command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id


//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the
predefined client id by passing parameter like server id,portand
wificredential


void setup()// configureing the ESP32
{
```

```
  Serial.begin(115200);
  Serial.println();

   pinMode(trigpin, OUTPUT);
  pinMode(echopin, INPUT);
  Myservo.attach(26);
  pinMode(trigpin1, OUTPUT);
  pinMode(echopin1, INPUT);

  wificonnect();
  mqttconnect();
}

void loop()// Recursive Function
{

   digitalWrite(trigpin1,LOW);
  digitalWrite(trigpin1,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin1,LOW);
  float duration=pulseIn(echopin1,HIGH);
  d1=(duration/(58*4));
  if(d1<20){
    Myservo.write(180);
    delay(15);
    Serial.println("Putin waste here");
  }
  else{
    Myservo.write(0);
  }


  digitalWrite(trigpin,LOW);
  digitalWrite(trigpin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin,LOW);
 float dura=pulseIn(echopin,HIGH);
  d=(dura/(58*4));

w=random(5,100);
String s="Kinathukadavu,Coimbatore";
String status="";
  Serial.print("WasteLevel:");
  Serial.println(d);
  Serial.print("WasteWeight:");
  Serial.println(w);
  Serial.println("Location:");
  Serial.println(s);
```

```
  PublishData(d, w);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}



/*.....................................retrieving to
Cloud.............................*/

void PublishData(float dis, float wei) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String status="";

  String payload = "{\"DustbinContent\":";
  payload += dis;
  payload += "," "\"WateWeight\":";
  payload += wei;
  payload += "," "\"Location\":";
  payload += "\"Kinathukadavu,Coimbatore\"";
  payload += "}";


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


  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it sucessfully upload data on the cloud
then it will print publish ok in Serial monitor or else it will print publish
failed
  } else {
    Serial.println("Publish failed");
  }

}


void mqttconnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting client to ");
    Serial.println(server);
    while (!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
```

```
      delay(500);
    }

     initManagedDevice();
     Serial.println();
  }
}
void wificonnect() //function defination for wificonnect
{
  Serial.println();
  Serial.print("Connecting to ");

  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish
the connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void initManagedDevice() {
  if (client.subscribe(subscribetopic)) {
    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}

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++) {
    //Serial.print((char)payload[i]);
    data3 += (char)payload[i];
  }
  Serial.println("data: "+ data3);
  if(data3=="lighton")
  {
Serial.println(data3);
  }
```

```
data3="";
}
```

Output:

```
Sending payload:
{"DustbinContent":53.23,"WateWeight":89.00,"Location":"Kinath
ukadavu,Coimbatore"}
Publish ok
WasteLevel:53.23
WasteWeight:88.00
Location:
Kinathukadavu,Coimbatore
Sending payload:
{"DustbinContent":53.23,"WateWeight":88.00,"Location":"Kinath
ukadavu,Coimbatore"}
Publish ok
WasteLevel:53.23
WasteWeight:32.00
Location:
Kinathukadavu,Coimbatore
Sending payload:
{"DustbinContent":53.23,"WateWeight":32.00,"Location":"Kinath
ukadavu,Coimbatore"}
Publish ok
```

CHAPTER 8: TESTING

8.1 Test cases:

Unit testing

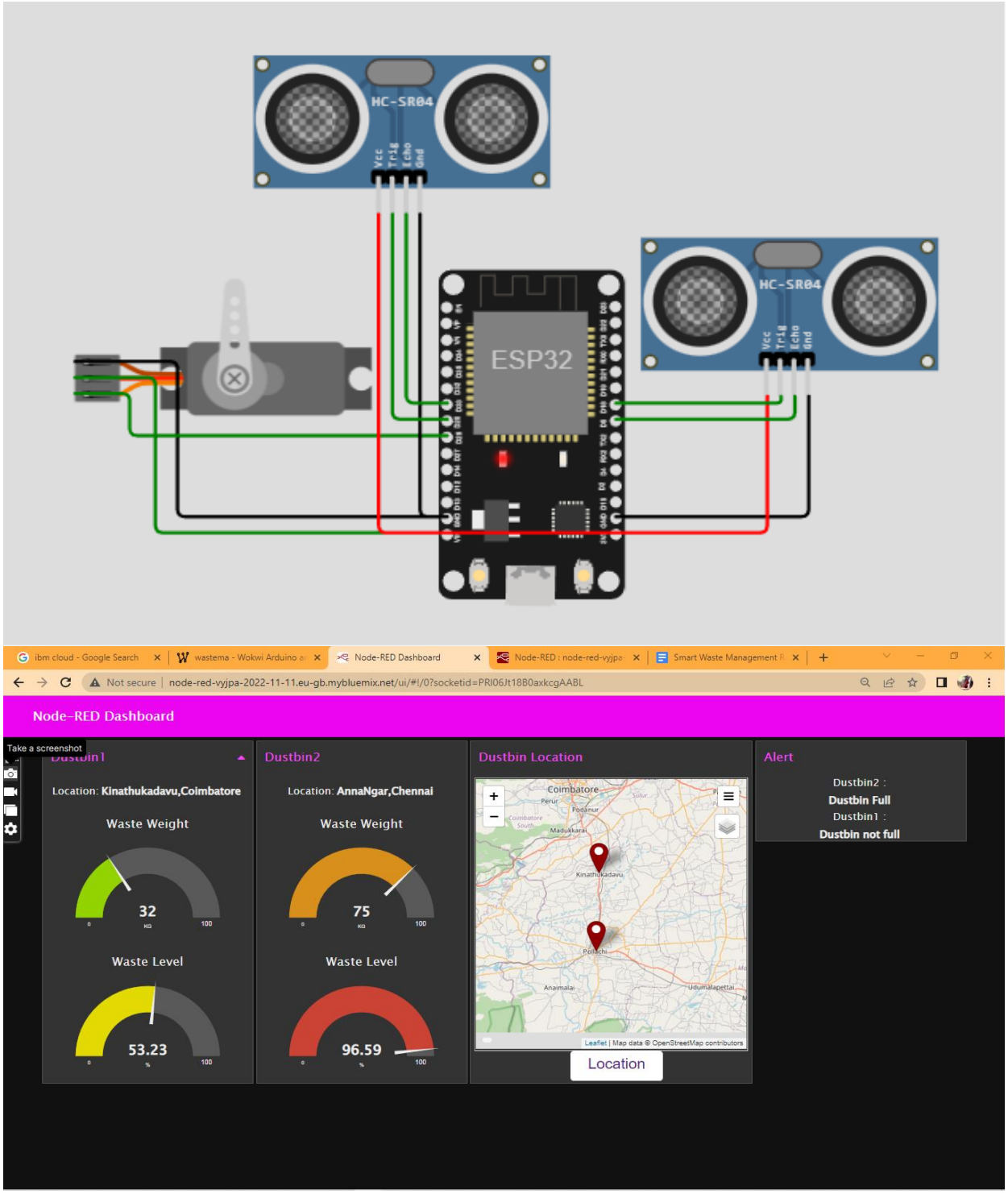
Test case no.	Sensor/Stage	Input	Expected output	Obtained output	Status
1.	Ultrasonic	Garbage level in bin i)Null ii)Full iii)Range in %	Correct level or distance	As expected	Pass
2.	ESP – 32	Microcontroller to process the input data	To collect the data from sensor	As expected	Pass
3.	Gauge	To display the tares	Display the level for collection	As expected	Pass
4.	HX710	Weight of the bin (in kg)	Measure the weight	As expected	Pass

8.2 User Acceptance testing

Acceptance testing - is the final phase of product testing prior to public launch. A level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

CHAPTER 9: RESULTS

Outputs:



Output:

```
Sending payload:
{"DustbinContent":53.23,"WateWeight":89.00,"Location":"Kinath
ukadavu,Coimbatore"}
Publish ok
WasteLevel:53.23
WasteWeight:88.00
Location:
Kinathukadavu,Coimbatore
Sending payload:
{"DustbinContent":53.23,"WateWeight":88.00,"Location":"Kinath
ukadavu,Coimbatore"}
Publish ok
WasteLevel:53.23
WasteWeight:32.00
Location:
Kinathukadavu,Coimbatore
Sending payload:
{"DustbinContent":53.23,"WateWeight":32.00,"Location":"Kinath
ukadavu,Coimbatore"}
Publish ok
```

CHAPTER 10: ADVANTAGES AND DISADVANTAGES

10.1 Advantages:

- Intelligent compaction of waste by monitoring fill level in real-time using sensors.
- It keeps our surroundings clean and keeps free from bad odour.
- Reduces manpower requirement to handle the garbage collection
- Emphasizes of healthy environment and keep the cities cleaner and more beautiful.
- It reduces infrastructure, operating and maintenance costs by upto 30%.
- Increases recycling rate of waste.

10.2 Disadvantages:

- Initial large-scale implementation takes cost.
- System requires more number waste bins for separate waste collection.

- Wireless technologies used should have proper connections as they have shorter range and lower data speed
- Training programs should be provided to people involving in the ecosystem of smart waste management.
- Sensors may encounter damage so it should be kept under protective ambience to prevent the damage.
- Replacement of sensors require knowledgeable people and thus acknowledgement of malfunction of sensor.

CHAPTER 11: CONCLUSION

Improper disposal and improper maintenance of domestic waste create issues in public health and environment pollution thus this paper attempts to provide practical solution towards managing the waste collaborating it with the use of IOT. by using the smart waste management system, we can manage waste properly we are also able to sort the Bio-degradable and non-Biodegradable waste properly which reduces the pollution in the environment. Various waste management initiatives taken for human well-being and to improve the TWM practices were broadly discussed in this chapter. The parameters that influence the technology and economic aspects of waste management were also discussed clearly. Different types of barriers in TWM, such as economic hitches, political issues, legislative disputes, informative and managerial as well as solutions and success factors for implementing an effective management of toxic organic waste within a globular context, were also discussed giving some real examples. The effect of urbanization on the environmental degradation and economic growth was also discussed. The proposed system will help to overcome all the serious issues related to waste and keep the environment clean.

CHAPTER 12: FUTURE WORK

Based on the real-time and historical data collected and stored in the cloud waste collection schedules and routes can be optimized. Predictive analytics could be used to make decisions ahead of time and offers insight into waste bin locations. Graph theory optimization algorithms can be used to manage waste collection strategies dynamically and efficiently. Every day, the workers can receive the newly calculated routes in their navigation devices. The system can be designed to learn from experience and to make decisions not only on the daily waste level status but also on future state forecast, traffic congestion, balanced cost-efficiency functions, and other affecting factors that a priori humans cannot foresee.

Garbage collectors could access the application on their mobile phone/tablets using the internet. Real-time GPS assistance can be used to direct them to the pre-decided route. As they go collecting the garbage from the containers, the management is also aware of the progress as the vehicle, as well as the garbage containers, are traced in real-time. The management staff gets their own personalized administration panel over a computer/tablet which gives them a bird eye view over the entire operations.

An alternative solution using image processing and camera as a passive sensor could be used. But, the cost of those image processing cameras is higher as compared to the ultrasonic sensors, which leads to high solution implementation cost.

CHAPTER 13: APPENDIX

13.1 Source Code:

Dustbin1:

```
#include <WiFi.h>
#include <PubSubClient.h>
#include<Servo.h>

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

//-------credentials of IBM Accounts------

#define ORG "woaev0"//IBM ORGANITION ID
#define DEVICE_TYPE "abcde"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "112345"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "1a2b3c4d"
String data3;
float d,d1,w;

Servo Myservo;
int pos;

#define trigpin    18
#define echopin    5

#define trigpin1 25
#define echopin1 33

//-------- Customise the 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 subscribetopic[] = "iot-2/cmd/display/fmt/String";// cmd  REPRESENT
command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id


//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the
predefined client id by passing parameter like server id,portand
wificredential
```

```
void setup()// configureing the ESP32
{
  Serial.begin(115200);
  Serial.println();

   pinMode(trigpin, OUTPUT);
  pinMode(echopin, INPUT);
  Myservo.attach(26);
  pinMode(trigpin1, OUTPUT);
  pinMode(echopin1, INPUT);

  wificonnect();
  mqttconnect();
}

void loop()// Recursive Function
{

   digitalWrite(trigpin1,LOW);
  digitalWrite(trigpin1,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin1,LOW);
  float duration=pulseIn(echopin1,HIGH);
  d1=(duration/(58*4));
  if(d1<20){
    Myservo.write(180);
    delay(15);
    Serial.println("Putin waste here");
  }
  else{
    Myservo.write(0);
  }


  digitalWrite(trigpin,LOW);
  digitalWrite(trigpin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin,LOW);
 float dura=pulseIn(echopin,HIGH);
  d=(dura/(58*4));

w=random(5,100);
String s="Kinathukadavu,Coimbatore";
String status="";
  Serial.print("WasteLevel:");
  Serial.println(d);
  Serial.print("WasteWeight:");
  Serial.println(w);
  Serial.println("Location:");
```

```
  Serial.println(s);

  PublishData(d, w);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}



/*.....................................retrieving to
Cloud.............................*/

void PublishData(float dis, float wei) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String status="";

  String payload = "{\"DustbinContent\":";
  payload += dis;
  payload += "," "\"WateWeight\":";
  payload += wei;
  payload += "," "\"Location\":";
  payload += "\"Kinathukadavu,Coimbatore\"";
  payload += "}";


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


  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it sucessfully upload data on the cloud
then it will print publish ok in Serial monitor or else it will print publish
failed
  } else {
    Serial.println("Publish failed");
  }

}


void mqttconnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting client to ");
    Serial.println(server);
```

```
    while (!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }

     initManagedDevice();
     Serial.println();
  }
}
void wificonnect() //function defination for wificonnect
{
  Serial.println();
  Serial.print("Connecting to ");

  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish
the connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void initManagedDevice() {
  if (client.subscribe(subscribetopic)) {
    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}

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++) {
    //Serial.print((char)payload[i]);
    data3 += (char)payload[i];
  }
  Serial.println("data: "+ data3);
  if(data3=="lighton")
  {
Serial.println(data3);
```

```
  }

data3="";
}
```

Dustbin-2:

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include<Servo.h>

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

//-------credentials of IBM Accounts------

#define ORG "7wpj5u"
#define DEVICE_TYPE "smarth"
#define DEVICE_ID "1213213e2e"
#define TOKEN "23456782345"

String data3;
float d,d1,w;

Servo Myservo;
int pos;

#define trigpin    18
#define echopin    5

#define trigpin1 25
#define echopin1 33

//-------- Customise the 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 subscribetopic[] = "iot-2/cmd/display/fmt/String";// cmd  REPRESENT
command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id


//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
```

```
PubSubClient client(server, 1883, callback ,wifiClient); //calling the
predefined client id by passing parameter like server id,portand
wificredential


void setup()// configureing the ESP32
{
  Serial.begin(115200);
  Serial.println();

   pinMode(trigpin, OUTPUT);
  pinMode(echopin, INPUT);
  Myservo.attach(26);
  pinMode(trigpin1, OUTPUT);
  pinMode(echopin1, INPUT);

  wificonnect();
  mqttconnect();
}

void loop()// Recursive Function
{

   digitalWrite(trigpin1,LOW);
  digitalWrite(trigpin1,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin1,LOW);
  float duration=pulseIn(echopin1,HIGH);
  d1=(duration/(58*4));
  if(d1<20){
    Myservo.write(180);
    delay(15);
    Serial.println("Putin waste here");
  }
  else{
    Myservo.write(0);
  }


  digitalWrite(trigpin,LOW);
  digitalWrite(trigpin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin,LOW);
 float dura=pulseIn(echopin,HIGH);
  d=(dura/(58*4));

w=random(5,100);
String s="Kinathukadavu,Coimbatore";
  Serial.print("WasteLevel:");
```

```
  Serial.println(d);
  Serial.print("WasteWeight:");
  Serial.println(w);
  Serial.println("Location:");
  Serial.println(s);

  PublishData(d, w);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}



/*.....................................retrieving to
Cloud.............................*/

void PublishData(float dis, float wei) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String payload = "{\"DustbinContent\":";
  payload += dis;
  payload += "," "\"WateWeight\":";
  payload += wei;
  payload += "," "\"Location\":";
  payload += "\"AnnaNgar,Chennai\"";
  payload += "}";


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


  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it sucessfully upload data on the cloud
then it will print publish ok in Serial monitor or else it will print publish
failed
  } else {
    Serial.println("Publish failed");
  }

}


void mqttconnect() {
  if (!client.connected()) {
```

```
    Serial.print("Reconnecting client to ");
    Serial.println(server);
    while (!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }

     initManagedDevice();
     Serial.println();
  }
}
void wificonnect() //function defination for wificonnect
{
  Serial.println();
  Serial.print("Connecting to ");

  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish
the connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void initManagedDevice() {
  if (client.subscribe(subscribetopic)) {
    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}

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++) {
    //Serial.print((char)payload[i]);
    data3 += (char)payload[i];
  }
  Serial.println("data: "+ data3);
  if(data3=="lighton")
```

```
  {
Serial.println(data3);
  }

data3="";
}
```

13.2 Project Links:

Github Link:
https://github.com/IBM-EPBL/IBM-Project-35239-1660283010

DemoLink:
https://drive.google.com/drive/folders/1cr-12jiyuDG-qAm8cACYX4ZCtKDeDW24

Nodes Dashboard UI Link:
http://node-red-vyjpa-2022-11-11.eu-gb.mybluemix.net/ui/