Team ID:PNT2022TMID43114

SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITAN CITIES

INTRODUCTION

1.1 Project Overview

Today big cities around the world are facing a common problem, managing the city waste effectively without making city unclean. Today's waste management systems involve a large number of employees being appointed to attend a certain number of dumpsters this is done every day periodically. This leads to a very inefficient and unclean system in which some dumpsters will be overflowing some dumpsters might not be even half full. This is caused by variation in population density in the city or some other random factor this makes it impossible to determine which part needs immediate attention. Here a waste management system is introduced in which each dumpster is embedded in a monitoring system that will notify the corresponding personal if the dumpster is full. In this system, it is also possible to separate wet and dry waste into two separate containers. This system provides an effective solution to the waste management problem

1.2 Purpose

A waste management system is the strategy an organization uses to dispose, reduce, reuse, and prevent waste. Possible waste disposal methods are recycling, composting, incineration, landfills, bioremediation, waste to energy, and waste minimization.

LITERATURE SURVEY

2.1 Existing Problem

Indiscriminate disposal of solid waste is a major issue in urban centers of most developing countries and it poses a serious threat to healthy living of the citizens. Access to reliable data on the state of solid waste at different locations within the city will help both the local authorities and the citizens to effectively manage the menace.

2.2 Reference

1. Municipal Solid Waste Collection Problems: A Literature Review, Jeroen Beliën, Liesje De Boeck, Jonas Van Ackere
2. Nuortio,T., Kytojoki, J., Niska, H., Braysy, O.: Improved route planning and scheduling of waste collection and transport. Journal of Expert Systems with Applications 30(2), 223– 232 (2006) CrossRef
3. Zamorano, M., Molero, E., Grindlay, A., Rondriquez, M.L., Hurtado, A., Calvo, and F.J.: A planning scenario for the application of geographical information systems in municipal Waste
4. sigrenEa. Connectivity for smart recycling.
5. Moba corporation. Smart Waste Management with Sensor Technology 4.0

2.3 Problem Statement Definition

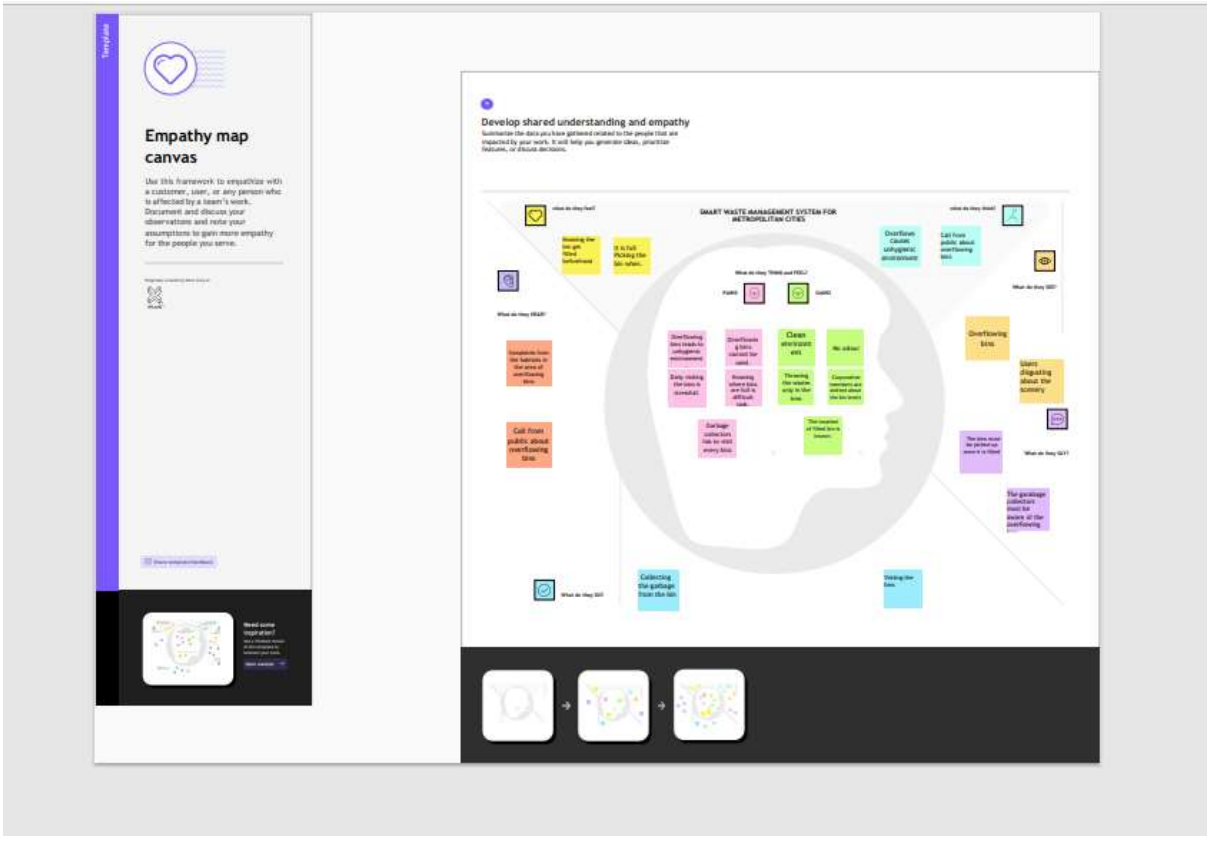
Depending on the fill level, the system sends appropriate notification message to alert relevant authorities and concerned citizen for necessary action. Also, the fill level is monitored on ThingSpeak in real-time. The system performance shows that the proposed solution may be found useful for efficient waste management in smart and connected communities.

Customer Problem Statement:

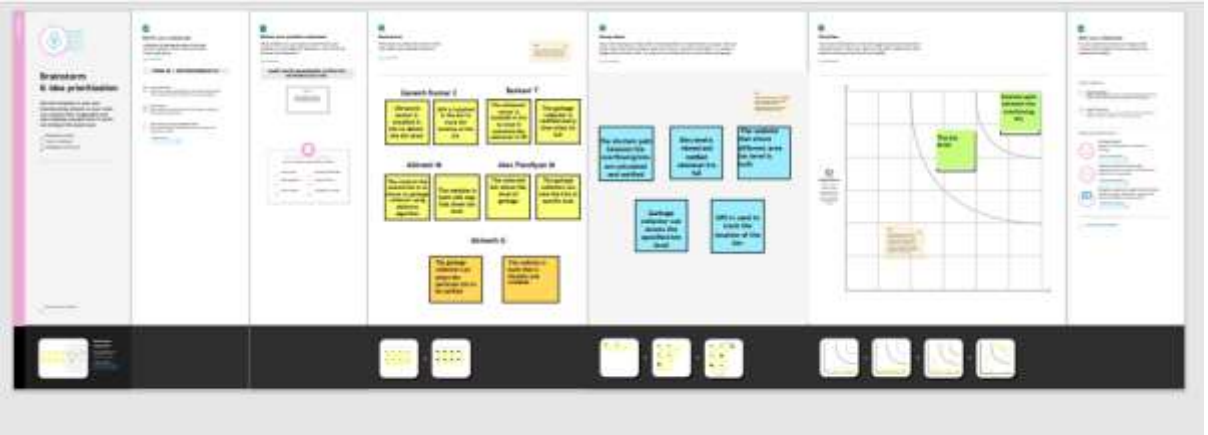
Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Garbage Collector	Collect Overflowing bins	Does not know where the nearest bin to collect	The area is wide to visit	Depressed
PS-2	Garbage collector	Know the different bin level	I have to visit the bins directly	The are is wide to visit	Tired

IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



3.3 Proposed Solution

Project Title: Smart waste management system for metropolitan cities Project Design Phase-I - Solution Fit Team ID: PNT2022TMID43114

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) (CS)
Garbage collector
Municipality officer

6. CUSTOMER CONSTRAINTS (CC)
The bins can't be visited directly daily
Overflowing bins can't be used
The waste of time by visiting the bins that are not full.

5. AVAILABLE SOLUTIONS (AS)
Checking the bin level
Checking the bin location
Notifying about the bin level.

Explore AS, differentiate

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

2. JOBS-TO-BE-DONE / PROBLEMS (J&P)
Overflowing of bins
Location of overflowing bins can't be known
Visiting every bins directly

9. PROBLEM ROOT CAUSE (RC)
Due to overflowing bins, the environment pollution is case
The location of overflowing bins is not known
Direct visiting consumes manpower.

7. BEHAVIOUR (BE)
Complaints from habitates around the overflowing bin.

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

3. TRIGGERS (TR)
Paying daily visit to the bin area
Bin level can't be known without.

10. YOUR SOLUTION (SL)
Automatically closing the bin once it filled.
Notifying garbage collector to pick the bin along with location
The nearest bin to be picked is notified.

8. CHANNELS OF BEHAVIOUR
Online – checking out bin level through website
Offline – directly visiting the bins.

Identify strong TR & EM

4. EMOTIONS: BEFORE / AFTER (EM)
Before
Tired, sometimes stressed
After
Relaxed.

Identify strong TR & EM

3.4 Problem Solution Fit

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	• Developing a smart bin to detect the bin level
2.	Idea / Solution description	• Using Dijkstra's algorithm to find the nearest overflowing bin to be collected
3.	Novelty / Uniqueness	• Website with the map that shows bin level along with its location
4.	Social Impact / Customer Satisfaction	• The bin can be picked up once it filled and hence keep the environment clean
5.	Business Model (Revenue Model)	• The Website showing different bin level and notifying when it is full to get picked
6.	Scalability of the Solution	• Specific bin can be selected and details can be viewed.

REQUIREMENT ANALYSIS

4.1 Functional Requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR NO.	Functional Requirement (Epic)	Sub Requirement (Story/Sub- Task)
FR – 1	Website	Visiting website to Taking sensor reading from the sensor circuit. Visiting website to check the locations of the Bin. Visiting website to check overflowing bin locations
FR – 2	Searching	The Administrator Should be able to access the system as an administrator, add bins, view the bin dashboard, view all the bins on the map, View the fill levels of bins in real time and their locations on the google map.
FR – 3	Notifying	It sends an Message Alert to the garbage collector on duty when a bin is full.

4.2 Non-Functional Requirement

Non – functional Requirements:

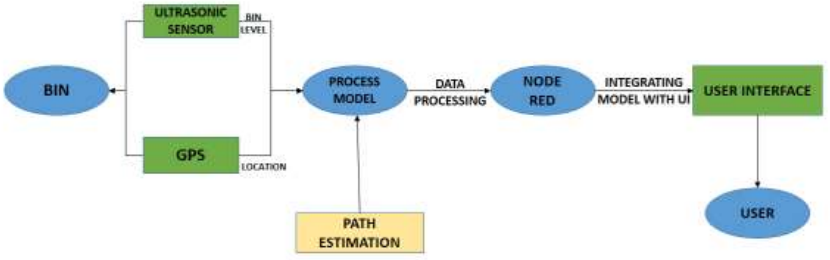
Following are the non – functional requirements of the proposed solution.

FR NO.	Non – functional Requirement	Description
NFR - 1	Usability	The UI should be simple enough for everyone to understand
NFR - 2	Security	The website must be secure enough to trust by the users
NFR - 3	Reliability	The UI should be able to withstand any errors in the data
NFR - 4	Performance	The live map for the bin in specified areas are shown
NFR - 5	Availability	The UI should respond to the users within 2 seconds
NFR - 6	Scalability	Dijkstra's Algorithm is used to estimate the nearest path

PROJECT DESIGN

5.1 Data Flow Diagrams

Data Flow Diagrams:



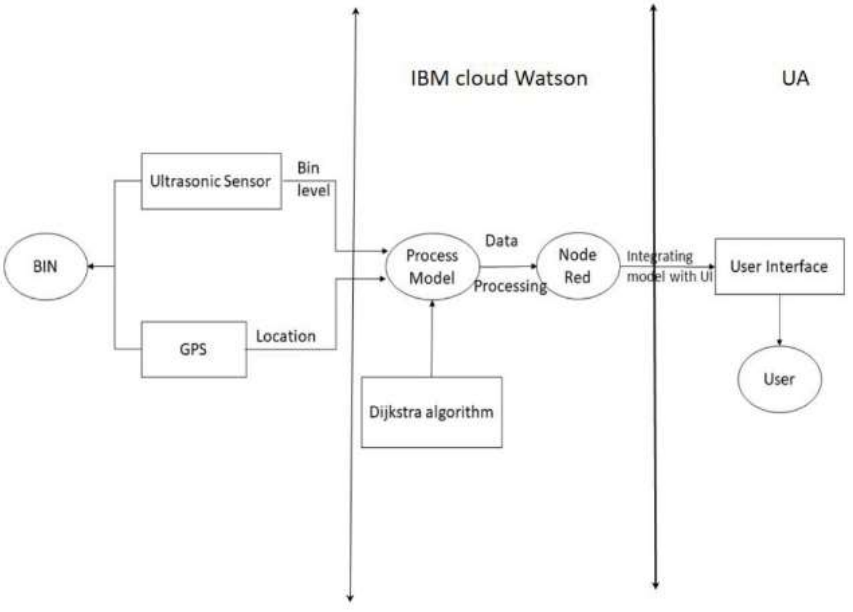
5.2 Solution & Technical Requirement

User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story/ Task	Acceptance Criteria	Priority	Release
Garbage collector	Visit website	USN – 1	As a user, I can visit website	I can visit the website	high	Sprint - 1
		USN - 2	As a user, I can monitor the bin level of different bins	I can get notification of overflowing bins	high	Sprint – 1
		USN – 3	AS a user, I can monitor the bin's location	I can get location of bins	high	Sprint – 1
		USN – 4	As a user, I can view the shortest path to the next bin.	I can see the next nearest bin to be picked.	high	Sprint - 2
Municipal officer	Visit website	USN – 5	As a user, I can monitor bin level and location of different bins	I can intimidate garbage collectors about bin levels	high	Sprint - 1

5.3 User Stories

Technical Architecture:



PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

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

Use the below template to create product backlog and sprint schedule

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

6.2 Sprint Delivery Schedule

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

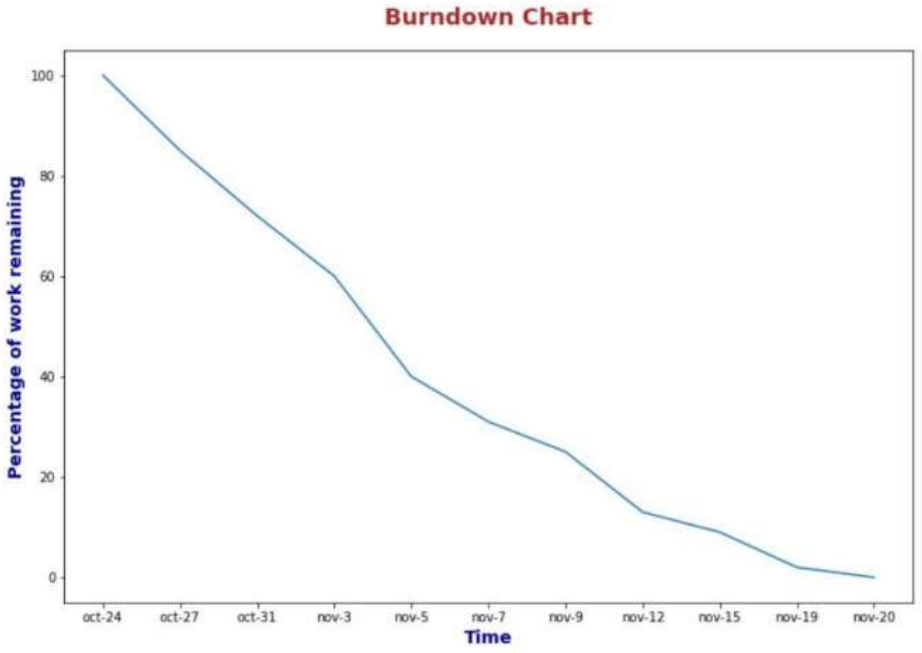
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	27 Oct 2022	02 Nov2022	20	02 Nov 2022
Sprint-2	20	6 Days	03 Nov 2022	08 Nov 2022	20	08 Nov 2022
Sprint-3	20	6 Days	08 Nov 2022	13 Nov 2022	20	13 Nov 2022
Sprint-4	20	6 Days	13 Nov 2022	18 Nov 2022	20	18 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 Report From JIRA



CODING & SOLUTIONING

7.1 Feature 1

```
import time
import sys
import ibmiotf.application
import ibmiotf.device

#provide your IBM Watson Device Credentials
Organization="80bbqy"
Device Type="cse2019"
deviceId="ganesh1601"
authMethod="token"
authToken="!8-vkyUJA+QxM-l+uf"

#Initialize GPIO

def myCommandCallback(cmd);
print("Command received:%s" %cmd.data['command'])
```

```
status=cmd.data['command']
if status=="lighton":
        print("led is on")
else
        print("led is off")

#print(cmd)
try:
        deviceOptions={"org": organization, "type":deviceType,"id":
device,"auth-method": authMethod, "auth-token": authToken}
deviceCli =ibmiotf.device.Client(deviceOptions)
#.....

Except Exception as e:
        Print("Caught exception connecting device: %s" %str(e))
        Sys.exit()

#Connect and send an datapoint "hello" with value "world" into the cloud as an event type "greeting" 10 times

deviceCli.connect()
while true:
        #Get sensor data fromDHT11
        level=random.randint(0,100)
        weight=random.randint(0,100)
        data={'level':level,'weight':weight}
#print data
        Def myOnPublishCallback():
        Print("Published level=%sC"%level,"weight=%s%%"%weight,"to IBM Watson")
        Success=deviceCli.publishEvent("IoTSensor","json",data,qos=0, on publish=myOnPublishCallback)
        if not success:
                print("Not Connected to IoTF")
        time.sleep(1)
        deviceCli.commandCallback = myCommandCallback
```

if(level>=75):

print(“Full LED ON”)

#Disconnect the device and application from the cloud

deviceCli.disconnect()

7.2 Feature 2

```
#include <WiFi.h>                    // library for wifi #include
<PubSubClient.h>                     // library for MQTT
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
//---------------------- credentials of IBM Accounts ----------------------------
#define ORG "80bbqy"                 // IBM organisation id
#define DEVICE_TYPE "cse2019"        // Device type mentioned in ibm watson iot platform
#define DEVICE_ID "ganesh1601"       // Device ID mentioned in ibm watson iot platform
#define TOKEN "12345678"         // 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;
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 <= 60)                                    //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 > 60 && cm < 120)
 {
  digitalWrite(4, HIGH);
  Serial.println("Warning!!,Trash is about to cross 50% of bin level");
  digitalWrite(2, LOW);
  digitalWrite(23, LOW);

 }
 else if(cm > 120)
 {
  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 <= 60)
{
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 <= 120)
{
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
{
Serial.println();
```

```
}
 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();
}
```

8.Testing

8.1 Test cases:

Date				18-Nov-22						
Team ID				PNT2022TMID43114						
Project Name				Smart Waste Management System for Metropolitan Cities - IOT						
Maximum Marks				4 marks						
Feature Type-Bin Level	**Component**	**Test Case Scenario**	**Pre-Requisite**	**Availability**	**Test Condition**	**Expected Result**	**Actual Result**	**Status**	**Comments**	**Accessed By**
Empty	Ultrasonic Sensor	When Bin is empty	Ultrasoncic sensor , Garbage Bins	Bin is accessible to users	Bin Level == 0	Displays Bin level and space left	Working as expected	Pass		User
Accessible	Ultrasonic Sensor	When bin level is below 50 %	Ultrasoncic sensor , Garbage Bins	Bin is accessible to users	Bin Level < 50	Displays Bin level and space left	Working as expected	Pass		User
Accessible	Ultrasonic Sensor	When bin level is above 50	Ultrasoncic sensor , Garbage Bins	Bin is accessible to users and the admin gets warning about the bin level	Bin Level > 50	Displays Bin level and space left	Working as expected	Pass		User
Accessible	Ultrasonic Sensor	When bin level is below 75 %	Ultrasoncic sensor , Garbage Bins	Bin is accessible to users and the admin gets warning about the bin level	Bin Level < 75	Displays Bin level and space left	Working as expected	Pass		User
Limit exceeded	Ultrasonic Sensor	When bin level is above 75 %	Ultrasoncic sensor , Garbage Bins	Bin is not accessible to the users, the admin recieves High alert and seals the the bin to avoid overflow.	Bin Level > 75	Displays Bin is FULL and Seals the bin.	Working as expected	Pass	The system starts to sense the level once the Bin is emptied partially	User/Admin

8.2 User Acceptance Testing:

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

9.Result:

9.1 Performance metrics:

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

10. Advantages and disadvantages:

10.1 Advantages

- Improve Productivity and Performance.
- Increase Profitability.
- Boost Sustainability.
- Superior Customer Engagement.
- Become a Smart City.
- Enhance Safety.

10.2 Disadvantages

- Process is not always cost-effective
- The resultant Product has a short life
- Need More Global Buy-in
- The site are often dangerous

11.Conclusion

In the system advocated above, the fusion of sensors, identification technology, and internet connectivity will lead to a uniquely smart disposal trash bin. Together with the cloud, these trash bins would become irreplaceable elements in the waste management cycle where the collection, transportation, storage, and recycling of waste could be automated. The use of RFID technology in waste collection services not only increases the efficiency of waste management through automation but also increases environmental responsibility which is one of the pillars of the Smart City.

12.Future Scope

Total of approximately 143,449 MT of municipal waste is generated daily. However, only 35,062 tons of waste is treated. A report from MNRE says that waste generation is expected to reach.Under the business-as-usual scenario, the amount of E-Waste will be more than doubled by 2050; “The complicated nature of production, use, and disposal of electronics require significant changes in order for the processes to become sustainable”.

Putting efforts into optimising collection, processing and treatment of recyclables and waste is not enough; we need to reach out to the production sector and help them create products that are optimally recyclable to the highest possible extent once consumers decide to generate their end-of-life products as waste.

13.Appendix

13.1. GitHub Account Link:

https://github.com/IBM-EPBL/IBM-Project-37921-1660364834

13.2 Project Demo Link:

https://drive.google.com/file/d/1VxL0FVF5wKa5zprFjk1PA9-nKAXIgzyI/view?usp=sharing