SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITAN CITIES

A.Bindo(210219106010),K.Logesh(210219106024), K.B.Kiran Babu(210219106023), P.Pandi Judhisti(210219106302).

Department of Electronics Communication Engineering, Apollo Engineering College, Poonamallee, Chennai.

TEAM ID	PNT2022TMID36751
PROJECT NAME	*SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITIAN CITIES*
TEAM LEADER	*A.Bindo*
TEAM MEMBER 1	*K.Logesh*
TEAM MEMBER 2	*K.B.Kiran Babu*
TEAM MEMBER 3	*P.Pandi Judhisti*

1. INTRODUCTION

1.1 PROJECT OVERVIEW

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. In this paper, an intelligent solid waste monitoring system is developed using Internet of Things (IoT) and cloud computing technologies. The fill level of solid waste in each of the containers, which are strategically situated across the communities, is detected using ultrasonic sensors. A Wireless Fidelity (Wi-Fi) communication link is used to transmit the sensor data to an IoT cloud platform known as ThingSpeak. Depending on the fill level, the system sends appropriate notification message (in form of tweet) to alert relevant authorities and concerned citizen(s) for necessary action.

1.2 PURPOSE

The current state of technology in the field of smart waste management involves the use of sensors that measure the fill level of the trash bin. Measured data is sent to the Cloud for further processing and analysis. By exploiting this data, trash collection can be planned as well as truck routes can be optimized. Despite this solution being an improvement of the conventional (sensorless) solution, it suffers from major drawbacks, poor sensor performance, and lack of any contribution to the trash sorting at the recycling facility.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

The current state of technology in the field of smart waste management involves the use of sensors that measure the fill level of the trash bin. Measured data is sent to the Cloud for further processing and analysis. By exploiting this data, trash collection can be planned as well as truck routes can be optimized. Despite this solution being an improvement of the conventional (sensorless) solution, it suffers from major drawbacks, poor sensor performance, and lack of any contribution to the trash sorting at the recycling facility.

.

2.2 REFERENCES

1.L. Banica, E. Butrescu and F. Enescu, "The Impact of Internet-of-Things in Higher Education. Scientific Bulletin – Economic Sciences", vol. 16, no. 1, pp. 53-59, 2017, 2.3

2. T. Ramos, C. de Moraisa and A. Barbosa-Póvoa, "The smart waste collection routing problem: Alternative operational management approaches", Expert Systems With Applications, vol. 103, pp. 146-158, 2018.

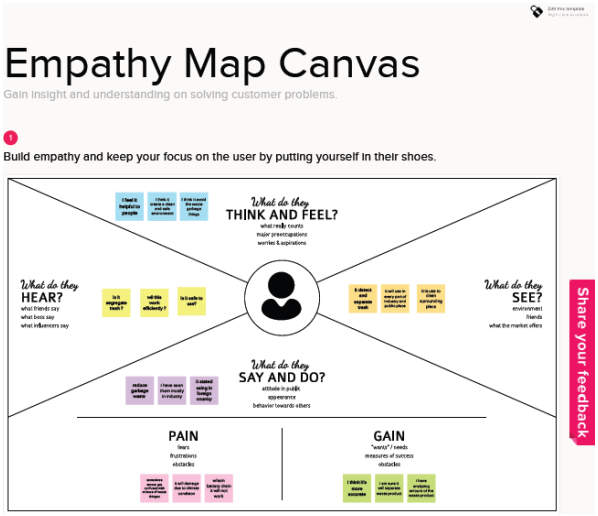
PROBLEM STATEMENT DEFINITION

The current process of waste management starts with the waste being created by people in the cities and disposed in trash bins near its creation point. The disposed trash is collected by municipality or private company trucks at the predefined times and transferred to temporary collection centers. The trash at the collection centers is then sent for recycling..

3. IDEATION & PROPOSED SOLUTION

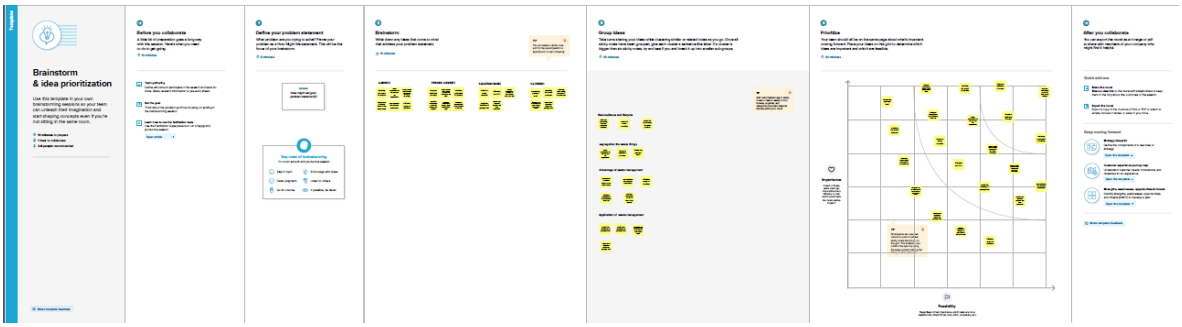
3.1 EMPATHY MAP CANVAS

An Empathy map is a collaborative tool teams can use to gain into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. Our empathy map canvas is shown as smart waste management monitoring and control system.



3.2 IDEATION & BRAINSTORMING

Ideation refers to the hole crreative process of coming up with and communicating new ideas.It can take many different forms,from coming up with a totally new idea to combining multiple existing ideas to create a new process or organizational system.Ideation is similar to a pratice known as brainstorming



3.3 PROPOSED SOLUTION

Proposed Solution means the technical solution to be provide by the implementation agency in response to the requirements and the objectives of the project. he following information may be useful to you in completing this portion of your team's work. Skim this section, then refer back to it as necessary.

Project Design Phase-I
Proposed Solution Template

Date	24 September 2022
Team ID	PNT2022TMID36751
Project Name	Smart waste management system for metropolitan cities
Maximum Marks	2 Marks

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)	Some trash bins are overfilled while others are underfilled by the trash collection, overfilled trash bins create unhygienic. Unoptimized truck routes result in excessive fuel usage and environmental pollution
2.	Idea / Solution description	Create awareness among the people about the waste management. Use the proper sensor to monitoring trash bins. Separating bio and non bio degradable waste
3.	Novelty / Uniqueness	This system provides the awareness among the people and making the environment safeguard with people
4.	Social Impact / Customer Satisfaction	By the system our environment be clean and safe. It gives healthy and peaceful life to the people.
5.	Business Model (Revenue Model)	Offering software as service model to the government. collecting useful waste and make it us money. Recycling old product to new product
6.	Scalability of the Solution	To helping the government to maintain clean and healthy environment

3.4 PROBLEM SOLUTION FIT

Problem solving is the act of defining a problem; determining the cause of the problem; identifying, prioritizing, and selecting alternatives for a solution; and implementing a solution. In order to effectively manage and run a successful organization, leadership must guide their employees and develop problem-solving techniques. Finding a suitable solution for issues can be accomplished by following the basic four-step problem-solving process and methodology outlined below.

4. REQUIREMENT ANALYSIS

Requirement analysis also called as requirement engineering is the process of determining user expectation for a new or modified product. These features are called requirement must be qualifiable relevant and detailed. Its classified as two major type. They are;

4.1 FUNCTIONAL REQUIREMENT

4.2 NON-FUNCTIONAL REQUIREMENT

Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Date	22 October 2022
Team ID	PNT2022TMID36751
Project Name	SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITAN CITIES
Maximum Marks	4 Marks

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Detailed bin inventory.	All monitored bins and stands can be seen on the map, and you can visit them at any time via the Street View feature from Google. Bins or stands are visible on the map as green, orange or red circles. You can see bin details in the Dashboard – capacity, waste type, last measurement, GPS location and collection schedule or pick recognition.
FR-2	Real time bin monitoring.	The Dashboard displays real-time data on fill-levels of bins monitored by smart sensors. In addition to the % of fill-level, based on the historical data, the tool predicts when the bin will become full, one of the functionalities that are not included even in the best waste management software. Sensors recognize picks as well, so you 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.
FR-3	Expensive bins.	We help you identify bins that drive up your collection costs. The tool calculates a rating for each bin in terms of collection costs. The tool considers the average distances depo-bin&bin-depo in the area. The tool assigns bin a rating (1-10) and calculates distance from depo-bin discharge.
FR-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. Based on the historical data, you can adjust bin capacity or location where necessary.
FR-5	Eliminate unefficient 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 . The report shows how full the bin was when picked. You immediately see any inefficient picks below 80% full.
FR-6	Plan waste collection routes.	The tool semi-automates 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.

Non-functional Requirements:

Following are the non-functional requirements of proposed solution

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	IoT device verifies that usability is a special and important perspective to analysis 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, behaviou r and experience.
NFR-2	**Security**	Use a reusable bottles Use reusable grocery bags Purchase wisely and recycle Avoid single use food and drink containers.
NFR-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.
NFR-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 das to Sensoneo 's Smart Waste Management Software System, a powerful cloud-based platform, for data driven daily operations, available also as a waste management app. 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%.
NFR-5	**Availability**	By developing & deploying resilient hardware and beautiful software we empower cities, businesses, and countries to manage waste smarter.
NFR-6	**Scalability**	Using smart waste bins reduce the number of bins inside the town , cities coz we able to monitor the garbage 24/7 more cost effect and scalability when we moves to smarter.

5. PROJECT DESIGN

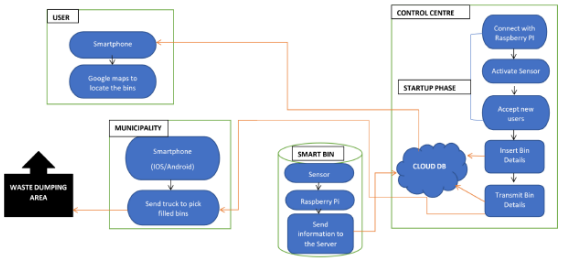
5.1 DATA FLOW DIAGRAM

Data flow diagram is a way of representing a flow of data through a processer or a system. DFD also provide information about the input and output of each entity and the process itself. A data flow diagram as no control flow they are no decision rules and no loops.

Project Design Phase-II
Data Flow Diagram & User Stories

Date	22 October 2022
Team ID	PNT2022TMID36751
Project Name	Project –Smart Waste Management for Metropolitan cities
Maximum Marks	4 Marks

Data Flow Diagram:

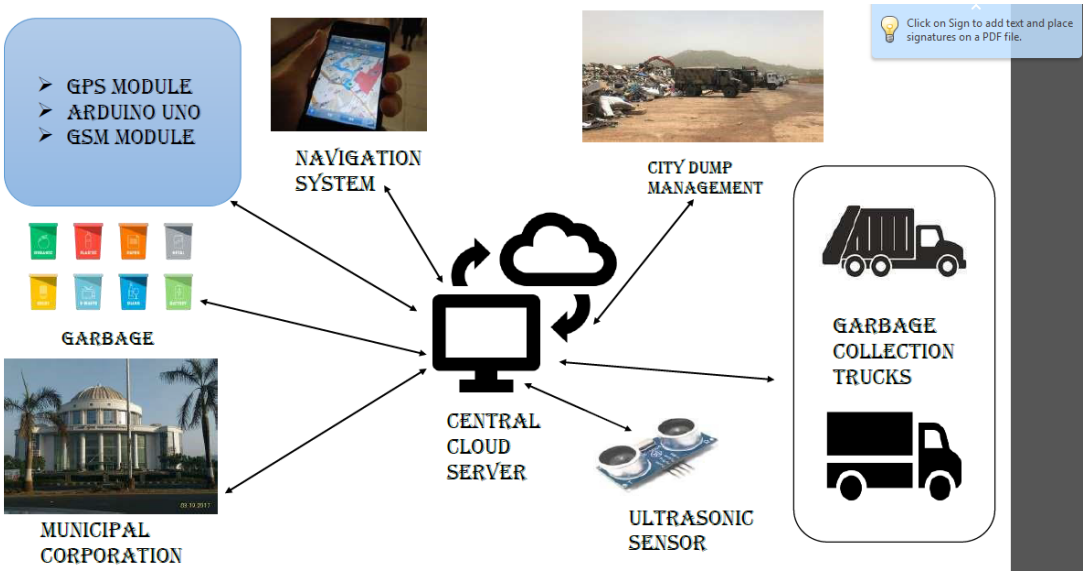


User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Mobile user	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account.	High	sprint
Admin	Registration and login	USN-2	As an Admin, I will manage the details entered by the user.	I can manage the account	High	sprint
Co Admin	Login	USN-3	As a Co Admin, I will manage bin details and I will send the information to the municipality.	I can handle bin details.	High	sprint
Truck Driver	Login	USN-4	As a Truck driver, I will collect the trash from the filled bins.	I can reach the bin location.	Medium	sprint
Municipality	Login	USN-5	As a Municipality, I will monitor the entire process.	I can manage the entire process.	High	sprint

5.2 SOLUTION & TECHNICAL ARCHITECTURE



A Solution architecture is an architectural description of a specific solution. SAs combine guidance from different enterprise architectural viewpoints (business, information and technical) as well as from the enterprise solution architecture (ESA).

5.3 USER STORIES

Waste management **reduces the effect of waste on the environment, health, and so on. It can also help reuse or recycle resources, such as; paper, cans, glass, and so on**. There is various type of waste management that include the disposal of solid, liquid, gaseous, or hazardous substances. Recycling saves energy, helps keep materials out of landfills and incinerators, and provides raw materials for the production of new products. When waste cannot be prevented, recycling is the next best option

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Sprint planning is an event in scrum that kicks of the sprint. The purpose of sprint planning is to define what we can deliver in the sprint and how that work will be achieved. Sprint planning is done in collaboration with the whole scrum team.

PROJECT PLANNING PHASE

Milestone and Activity List

Date	24 October 2022
Team ID	PNT2022TMID36751
Project Name	Smart Waste Management for Metropolitan Cities

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	22 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	21 SEPTEMBER 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	23 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	24 SEPTEMBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	18 OCTOBER 2022
Solution Architecture	Prepare solution architecture document.	22 OCTOBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	22 OCTOBER 2022
Functional Requirement	Prepare the functional and non functional requirement document.	22 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	22 OCTOBER 2022
Technology Architecture	Prepare the technology architecture diagram.	22 OCTOBER 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	24 OCTOBER 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it.	IN PROGRESS.

6.2 SPRINT DELIVERY SCHEDULE

The deliverables of the sprint are not as predictable as they are for the other project. Sprint participance have produced sketches and drawing, writing, photograph, comic, strip, video and fully coded working prototypes

Project Planning Phase

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Date	24 October 2022
Team ID	PNT2022TMID36751
Project Name	Project – Smart Waste Management System for metropolitan cities
Maximum Marks	8 Marks

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 can have total access to all the Co-Admin and Truck driver and monitor the waste.	20	High	A.BINDO
Sprint-2	Login In	USN-2	As a Co-Admin, I'll control the waste level by monitoring them via IBM Iot. Once the filling happens, I'll notify trash truck with location of bin with bin ID.	20	High	A.BINDO
Sprint-3	Dashboard	USN-3	As a Co-Admin, I will set the Notification process and other management are done.	20	High	P.PANDI JUDHISTI
Sprint-4	Dashboard	USN-4	As a Truck Driver, I can able to see the filled dustbin in my Dashboard and empty them.	10	Medium	K.LOGESH
Sprint-4	Dashboard	USN-5	As a Municipality officer I can view all the process is proceeding without any problems.	10	High	K.B.KIRAN BABU

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

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Now2022	20	12 Now2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity:

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

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

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

7.1Feature 1

```
#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 "dxjch6"                          // IBM organisation id
#define DEVICE_TYPE "sprint4"                  // Device type mentioned in ibm
watson iot platform
#define DEVICE_ID "77777"                 // Device ID mentioned in ibm watson iot
platform
#define TOKEN "78lTED7u9Kf52Q(iX)"            // Token

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

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";                //
server name
char publishTopic[] = "iot-2/evt/data/fmt/json";                               //
topic name and type of event perform and format in which data to be send
char topic[] = "iot-2/cmd/led/fmt/String";                                     //
cmd Represent type and command is test format of strings
char authMethod[] = "use-token-auth";                                          //
authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":"
DEVICE_ID;                      //Client id

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

WiFiClient wifiClient;                                                        //
creating instance for wificlient
PubSubClient client(server, 1883, wifiClient);

#define ECHO_PIN 12
#define TRIG_PIN 13
float dist;
String data3;
bool SealBin = true;
void setup()
{
  Serial.begin(115200);
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  //pir pin
  pinMode(34, INPUT);

  //ledpins
  pinMode(23, OUTPUT);
```

```
  pinMode(2, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(15, OUTPUT);


  lcd.init();
  lcd.backlight();
  lcd.setCursor(1, 0);
  lcd.print("");
  wifiConnect();
  mqttConnect();
}

float readcmCM()
{
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  int duration = pulseIn(ECHO_PIN, HIGH);
  return duration * 0.034 / 2;
}

void loop()
 {

  lcd.clear();

  publishData();
  delay(500);
  if (!client.loop())
    {
      mqttConnect();                                    // function call to
connect to IBM
    }
}


/* ---------------------------------retrieving to cloud----------------------------
--------------------------*/

void wifiConnect()
{
  Serial.print("Connecting to ");
```

```
  Serial.print("Wifi");
  WiFi.begin("Wokwi-GUEST", "", 6);
  while (WiFi.status() != WL_CONNECTED)
    {
      delay(500);
      Serial.print(".");
    }
  Serial.print("WiFi connected, IP address: ");
  Serial.println(WiFi.localIP());
}
void mqttConnect()
  {
    if (!client.connected())
      {
        Serial.print("Reconnecting MQTT client to ");
        Serial.println(server);
        while (!client.connect(clientId, authMethod, token))
          {
            Serial.print(".");
            delay(500);
          }
        initManagedDevice();
        Serial.println();
      }
  }
void initManagedDevice()
  {
    if (client.subscribe(topic))
      {
        Serial.println("IBM subscribe to cmd OK");
      }
    else
      {
        Serial.println("subscribe to cmd FAILED");
      }
  }
void publishData()
{
  float cm = readcmCM();

  if(digitalRead(34))                                  //pir motion detection
  {
    Serial.println("Motion Detected");
    Serial.println("Lid Opened");
    digitalWrite(15, HIGH);
```

```
 if(digitalRead(34)== true)
 {
  if(cm <= 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 if(cm > 120)
{
```

```
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

import time import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials organization = "dxjch6" deviceType = "sprint4" deviceId = "77777" authMethod = "token" authToken = "78lTED7u9Kf52Q(iX)"

Initialize GPIO

```
def myCommandCallback(cmd):     print("Command received: %s" % cmd.data['command'])
status=cmd.data['command']     if status=="binfull":         print ("-----EMPTY THE BIN
IMMEDIATELY----")             #print(cmd)
```

```
try:  deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "authmethod":
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 a datapoint "hello" with value "world" into the cloud as an event of type
"greeting" 10 times deviceCli.connect()

while True:         #Get Sensor Data from DHT11
binlevel=random.randint(10,100)         locationId=random.randint(1,5)
district="Tirunelveli"         state="Tamilnadu"         country="India"         if locationId == 1:
latitude=8.7060581            longitude=77.7633162            village="VM Chathiram"         elif
locationId == 2:            latitude=8.7066676            longitude=77.732578
village="Perumalpuram"         elif locationId == 3:            latitude=8.7199159
         longitude=77.725674            village="Palayamkottai"         elif locationId == 4:
latitude=8.7282671            longitude=77.7180244            village="Vannarpettai"         elif
locationId == 5:            latitude=8.7289086            longitude=77.6745726            village="Nellai
Town"         else:            print("No location Found!!")                    data = { 'latitude' : latitude,
'longitude': longitude,'binlevel': binlevel,'village':village,'district':district,'state':stat
e,'country':country }         #print data         def myOnPublishCallback():            print ("Published
Latitude = %s " % latitude, "Longitude = %s %%" % longitude, "Binlevel = %s" %
binlevel,"village = %s " % village,"district= %s" % district,"state = %s" % state,"country = %s "
% country, "to IBM Watson\n")         if binlevel >= 90:               data={'Latitude':latitude,
'Longitude':longitude, 'Binlevel':binlevel, 'Village':village, 'District':district,
'State':state,'Country':country}              print("!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!BIN IS FULL
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!\n")             print("---------------------------------EMPTY
THE BIN IMMEDIATELY--------------------------------------------\n")
deviceCli.commandCallback = myCommandCallback              time.sleep(5)        else:
print("BIN IS IN NORMAL LEVEL\n")           time.sleep(5)                        success =
deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
if not success:            print("Not connected to IoTF")        time.sleep(1)



# Disconnect the device and application from the cloud deviceCli.disconnect()
```

8. TESTING

8.1 Test Cases

Test case description	Required input	Information and related requirements	Test case status indicating pass or fail
The user or concerned service provider should register with the required details	User input details for registration	User Name, Email ID, Phone Number, and Security Password	Pass
The user or concerned service provider tried to log in to the monitoring portal with registered details	User login details	User Name, Security Password	Pass or Fail
Monitoring website portal indicating home, user, SGB status	User monitoring home screen should be display	The developed prototype for Smart Garbage Bin must be kept 'ON.'	Pass

9. RESULTS

9.1 Performance Metrics
We a successful built a web based UI and integrated all the services using node red
UI :https://node-red-kjery-2022-11-16.eu-gb.mybluemix.net/ui/#!/2?socketid=kG2lQ6ln2cuv4VSeAABk

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES

A reduction in the number of waste collections needed by up to 80%, resulting in less manpower, emissions, fuel use and traffic congestion. A reduction in the number of waste bins needed. Analytics data to manage collection routes and the placement of bins more effectively
.

DISADVANTAGES

- Air emissions. Air emissions are mainly produced by fumes from the burning of waste and also landfill gases. ...
- Health impact. ...
- Ecosystem services in danger. ...
- Soil contamination. ...
- Surface and groundwater. ...
- Marine pollution. ...
- Odour and littering. ...
- Pests.

11. CONCLUSION

Components of waste management are not just with respect to waste collected from the household, commercial establishments, etc. Waste is more of a serious concern which impacts the health and sanitation factor of every citizen of the city. Cleanliness is directly related to waste management and sanitation both. So it is not just the Smart cities which have become the enablers for Solid Waste Management but several other schemes like Swachh

Bharat Mission have also played a vital and significant role in bringing public awareness on health factor related to waste

12. FUTURE SCOPE

(1) an infrastructure for proper collection of product lifecycle data to facilitate full visibility throughout the entire lifespan of a product, (2) a set of new business models relied on product lifecycle data to prevent waste generation, and (3) an intelligent sensor-based infrastructure for proper upstream waste separation and on-time collection.The proposed framework highlights the value of product lifecycle data in reducing waste and enhancing waste recovery and the need for connecting waste management practices to the whole product life-cycle. An example of the use of tracking and data sharing technologies for investigating the waste management issues has been discussed. Finally, the success factors for implementing the proposed framework and some thoughts on future research directions have been discussed.

13. APPENDIX

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 "dxjch6"                              // IBM organisation id
#define DEVICE_TYPE "sprint4"                      // Device type mentioned in ibm
watson iot platform
#define DEVICE_ID "77777"                     // Device ID mentioned in ibm watson iot
platform
#define TOKEN "78lTED7u9Kf52Q(iX)"                // Token

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

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";                 //
server name
char publishTopic[] = "iot-2/evt/data/fmt/json";                                //
topic name and type of event perform and format in which data to be send
char topic[] = "iot-2/cmd/led/fmt/String";                                      //
cmd Represent type and command is test format of strings
```

```
char authMethod[] = "use-token-auth";                                    //
authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":"
DEVICE_ID;                    //Client id

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

WiFiClient wifiClient;                                                   //
creating instance for wificlient
PubSubClient client(server, 1883, wifiClient);

#define ECHO_PIN 12
#define TRIG_PIN 13
float dist;
String data3;
bool SealBin = true;
void setup()
{
  Serial.begin(115200);
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  //pir pin
  pinMode(34, INPUT);

  //ledpins
  pinMode(23, OUTPUT);
  pinMode(2, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(15, OUTPUT);


  lcd.init();
  lcd.backlight();
  lcd.setCursor(1, 0);
  lcd.print("");
  wifiConnect();
  mqttConnect();
}

float readcmCM()
{
  digitalWrite(TRIG_PIN, LOW);
```

```
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  int duration = pulseIn(ECHO_PIN, HIGH);
  return duration * 0.034 / 2;
}

void loop()
 {

  lcd.clear();

  publishData();
  delay(500);
  if (!client.loop())
    {
      mqttConnect();                                    // function call to
connect to IBM
    }
}


/* ---------------------------------retrieving to cloud---------------------------
--------------------------*/

void wifiConnect()
{
  Serial.print("Connecting to ");
  Serial.print("Wifi");
  WiFi.begin("Wokwi-GUEST", "", 6);
  while (WiFi.status() != WL_CONNECTED)
    {
      delay(500);
      Serial.print(".");
    }
  Serial.print("WiFi connected, IP address: ");
  Serial.println(WiFi.localIP());
}
void mqttConnect()
  {
    if (!client.connected())
      {
        Serial.print("Reconnecting MQTT client to ");
        Serial.println(server);
```

```
        while (!client.connect(clientId, authMethod, token))
          {
            Serial.print(".");
            delay(500);
          }
        initManagedDevice();
        Serial.println();
      }
  }
void initManagedDevice()
  {
    if (client.subscribe(topic))
      {
        Serial.println("IBM subscribe to cmd OK");
      }
    else
      {
        Serial.println("subscribe to cmd FAILED");
      }
  }
void publishData()
{
  float cm = readcmCM();

  if(digitalRead(34))                                    //pir motion detection
  {
    Serial.println("Motion Detected");
    Serial.println("Lid Opened");
    digitalWrite(15, HIGH);



 if(digitalRead(34)== true)
 {
  if(cm <= 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 if(cm > 120)
{
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

https://github.com/IBM-EPBL/IBM-Project-26275-1660023155

demo link: https://github.com/IBM-EPBL/IBM-Project-26275-1660023155/blob/main/Final%20Deliverables/final%20deliverables%20video.mp4