|  |
| --- |
|  |
|  |

SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITAN CITIES

**DOMAIN – INTERNET OF THINGS (IoT)**

# PROJECT REPORT

|  |
| --- |
| **TEAM ID :** PNT2022TMID50597  **TEAM MEMBERS :** KARTHIKA M (TL) - 952819106013  MARISELVI S -952819106016  SANGEETHA M -952819106701  DIVYA T -952819106004 |

**CHAPTER 1**

**INTRODUCTION**

**1.1 Project Overview**

Smart waste management is about using technology and data to create a more efficient waste industry. Based on IoT (Internet of Things) technology, smart waste management aims to optimize resource allocation, reduce running costs, and increase the sustainability of waste services.

The waste sector has traditionally been a static industry. And it is only in recent years that we have started to see modernizations to the ways of working in waste management. With the birth of IoT technology and new innovations becoming commercially available, waste authorities are increasingly looking to smart solutions, as a way of addressing budget cuts and ambitious sustainability targets.

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 labor and fuel costs – which residents ultimately foot the bill for this is also an unsustainable way of working - the more vehicles on the road carrying out

unnecessary collections means more carbon emissions are released into our planet’s atmosphere.

**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.

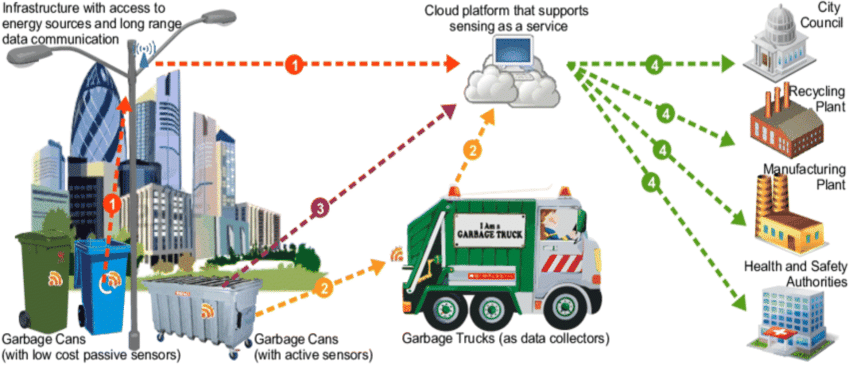


Fig 1: Smart waste management system

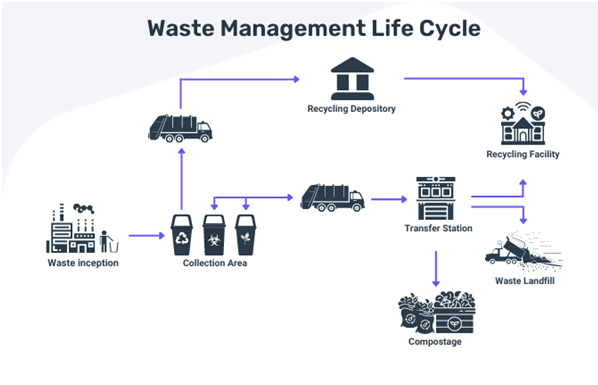


Fig 2: Waste management life cycle

As for waste management, it is the measures utilized to manage waste in its entire life cycle,

from waste generation to disposal or recovery.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 Existing Problem**

* Misunderstanding of the operations of smart sensors: Because this is a new and

emerging technology, there is a general misunderstanding of its operations

* Setting up the smart sensor
* Non-optimized truck routes
* Recycling
* Non-uniform waste distribution of waste in bins.

# IOT BASED SMART WASTE SYSTEM:

Our waste generation is constantly growing to form a Global garbage crisis .Even though we compromise on creating a more sustainable and green world with 2050 climate targets before too late, we still fail to recycle or handle our waste generation. Combining technology support with a vision of social, economic and environmental sustainability is the only way out of this problem.

# How Can Your Waste Operations Benefit from IoT?

[**IoT-based** **waste management for smart cities**](https://evreka.co/blog/sustainable-waste-management-with-big-data-ai/) has various examples around the world. These systems not only offer optimization for your operational plans but can also help reduce extra spending and ensure a more intelligent budget. In addition, they set an example for [**eco-friendly waste management**](https://evreka.co/blog/eco-friendly-solid-waste-management/), and the new resources created by advanced recycling practices make an important contribution to [**the circular economy**](https://evreka.co/circular-economy-solutions/). Overall, smart waste management using IoT can catch:

1. An operational standard and less managerial time

2)Wisely usage of scarce resources

3)Maximum revenue generation

Let’s give some examples of how smart waste management IoT relationship can Restructure your business.

# ****Smart Bin Sensors:****

Waste bins are one of the essential components of waste management Operations because they start the cycle of waste operations. [**IoT-based smart sensors**](https://evreka.co/smart-bin-technologies/) help you utilize smart bin sensor technology from the beginning.  One of the best types of smart bin sensors, the[**Fill Level Sensor**](https://evreka.co/solutions/fill-level-sensor/), supported by

IoT technology, you can:

1) Track the location with real-time data

2) View fullness levels for creating daily optimized routes for collection

3) Monitor the temperature of your smart bins

This way, you can reduce the number of missing containers with location trackingand reach advanced inventory management. Monitoring the temperatures will help prevent unwanted accidents like explosions and fires. Viewing the fullness levels will be one of

**To conclude , IoT Technology.**

To wrap up, Evreka can transform waste management operations to unprecedented levels with IoT technology. With the IoT based waste management for smart city, you can quickly achieve operational excellence. Utilizing above mentioned solutions in your operations will boost your revenue and bring your business to a respected place. But technology alone is not enough for gaining respect. When IoT and smart city are not combined with the sustainability vision, it may not produce the desired result.

**2.2 References**

[1] Shyam, Gopal Kirshna, Sunilkumar S. Manvi, and Priyanka Bharti. "Smart waste management using Internet-of-Things (IoT)." IEEE Computing and Communications Technologies (ICCCT), (2017) pp. 199-203.

[2] Kurre, Vishesh Kumar. "Smart Garbage Collection Bin overflows Indicator using IOT." International Research Journal of Engineering and Technology (IRJET) (2016).

[3] Folianto, Fachmin, Yong Sheng Low, and Wai Leong Yeow. "Smartbin: Smart waste management system." Tenth IEEE International conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), (2015).

[4] Vu, Dung, and Georges Kaddoum. "A waste city management system for smart cities applications." (2017).2017 Advances in Wireless and Optical Communications.

**2.3 Problem Statement Definition**

Solid waste management anywhere in the world is a problem that continually accelerates as a product of industrialization and population growth. As cities grow economically, greater

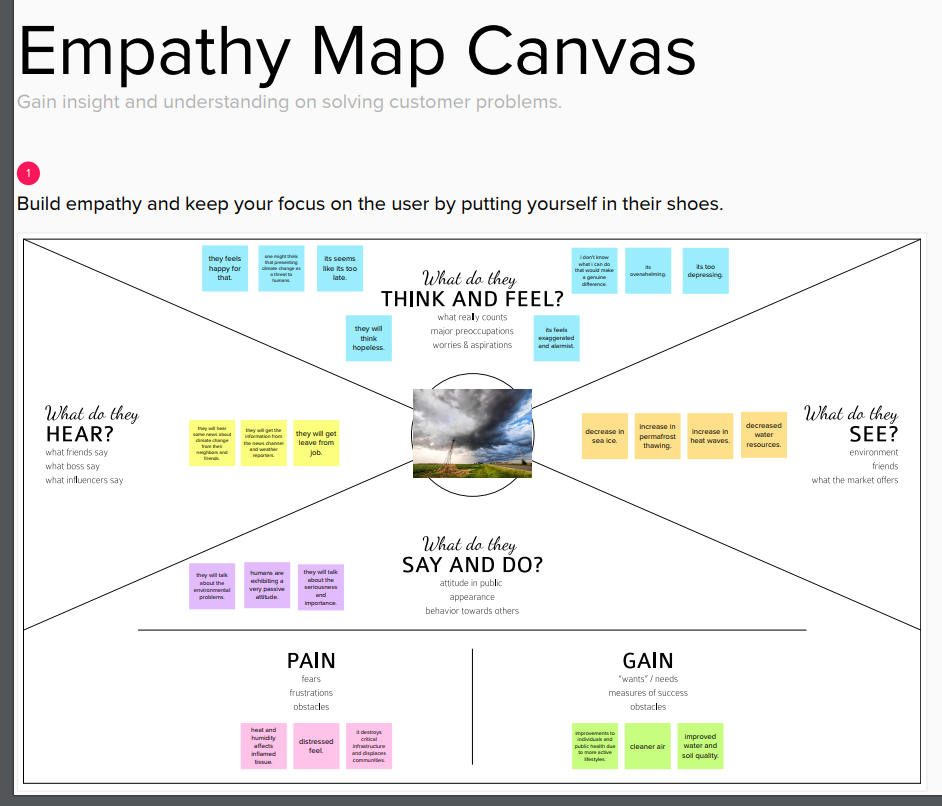
Business activity and diverse consumption patterns serve to drive up the solid waste quantities. Ghana has been principal victim of this disaster. Enough attention has been given to this challenge, but it seems to be a mere lip service that is played. A holistic and technical approach is needed since solid waste management is a complex challenge for the environment .Waste that are not properly managed are a serious health hazards leading to the spread of infectious diseases. Unattended waste lying around attracts flies, rats, and creatures that in turn spread diseases. WHO (2004) estimates that about 1.8 million people die annually from diarrheal diseases where 90% are children under five, mostly in developing countries. With the increasing influx of the people and the rapid urbanization, huge amount of human and small scale business waste of about 950 tones generate out of which 480 tones are collected representing 51%.

This leaves a substantial amount of back log that creates various kinds of inconveniences including health hazards to the people of Nkawkaw. Indiscriminate dumping of waste, irregular collection of waste generated and inadequate resources are the problems facing solid waste management in the municipal. Also lack of equipment and the absence of proper engineered final disposal sites delay the emptying of containers placed at vantage points. These containers overflow and litter scattered around it leading to the possible factors of diarrheal diseases. It therefore becomes necessary for this study to examine the problems of solid waste management.

**CHAPTER 3**

**IDEATION & PROPOSED SOLUTION**

**3.1 Empathy Map Canvas**

****

**Fig.3. Empathy map**

**3.2 Ideation and Brainstorming**

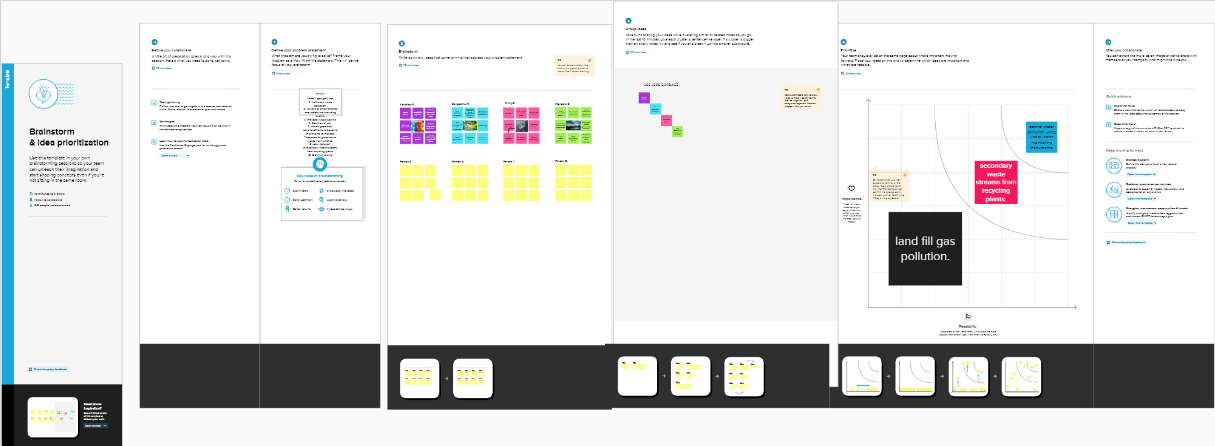


Fig.4. Ideation and Brainstorming

**3.3 Proposed Solution**

**1. Problem Statement (Problem to be solved):**

This project deals with the problem of waste management in smart cities,

where the garbage collection system is not optimized. This project enables the organizations to meet their needs of smart garbage management systems. This system allows the authorised

person to know the fill level of each garbage bin in a locality or city at all times, to give a cost-effective and time-saving route to the truck drivers.

**2. Idea / Solution description:**

The key research objectives are as follows:

• The proposed system would be able to automate the solid waste monitoring process and management of the overall collection process using IOT (Internet of Things).

• The Proposed system consists of main subsystems namely Smart Trash System(STS) and Smart Monitoring and Controlling Hut(SMCH). • In the proposed system, whenever the waste bin gets filled this is acknowledged by placing the circuit at the waste bin, which transmits it to the receiver at the desired place in the area or spot.

• In the proposed system, the received signal indicates the waste bin status at the monitoring and controlling system.

**3. Novelty / Uniqueness :**

We are going to establish SWM in our college but the real hard thing is that janitor (cleaner) don’t know to operate these thing practically so here our team planned to build a wrist band to them, that indicate via light blinking when the dustbin fill and this is Uniqueness we made here beside from project constrain.

**4. Social Impact / Customer Satisfaction:**

From the public perception as worst impacts of present solid waste disposal practices are seen direct social impacts such as neighbourhood of landfills to communities, breeding of pests and loss in property values

**5. Business Model (Revenue Model) :**

Waste Management organises its operations into two reportable business segments: Solid Waste, comprising the Company’s waste collection, transfer, recycling and resource recovery, and disposal services, which are operated and managed locally by the Company’s various subsidiaries, which focus on distinct geographic areas; and Corporate and Other, comprising the Company’s other activities, including its development and operation of landfill gas-to energy facilities in the INDIA, and its recycling brokerage services, as well as various corporate functions.

**6. Scalability of the Solution:**

In this regard, smart city design has been increasingly studied and discussed around the world to solve this problem. Following this approach, this paper presented an efficient IoT based and real-time waste management model for improving the living environment in cities, focused on a citizen perspective. The proposed system uses sensor and communication technologies where waste data is collected from the smart bin, in real-time, and then transmitted to an online platform where citizens can access and check the availability of the compartments scattered around a city.

**3.4 Problem Solution Fit**

**STEP 1:PROBLEM SOLVING CARDS:**

**-Basic question**

1) How will be collecting and handling?

2) What is the processing and resource recovery?

The Waste Framework Directive sets the basic concepts and definition related to waste management, including definitions of waste, recycling and recovery.

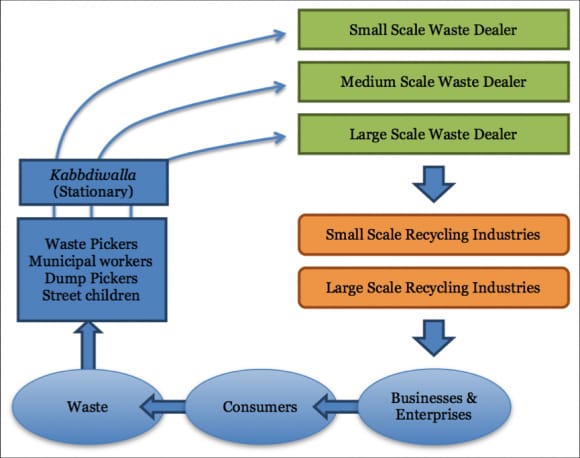


Fig .5. Problem Solving

Sustainable waste management means reducing and avoiding the amount of single-use plastic products along with increasing the amount recycled at the same time, which currently stands at just 8.5%.Waste management is intended to reduce adverse effects of waste on human health, the environment, planetary resources and aesthetics. The aim of waste management is to reduce the dangerous effects of such waste on the environment and human health. Waste management involves the processes of waste collection, transportation, processing, as well as waste recycling or disposal. Sustainable waste management systems include advanced management strategies to minimize environmental challenges and protect resources.

STEP 3: **IDEAS**

Problem solution

Fig.6 Ideas

* By minimizing the use of disposable items and promoting the use of recycled articles.
* Separating biodegradable and non-biodegradable waste before dumping them.
* Recycling the non-biodegradable waste material.

**CHAPTER 4**

**REQUIREMENT ANALYSIS**

**4.1 Functional Requirement**

Following are the functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | 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 distance depo-bin- discharge 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. |

**4.2 Non- Functional Requirement**

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

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-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. |
| 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 data 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 town , cities coz we able to monitor . |

**CHAPTER 5**

**PROJECT DESIGN**

* 1. **Data Flow Diagram**

• A data flow diagram system is implemented throughout the city.This is a centrally controlled system which finds locations of several waste bins.

• System is implemented throughout the city. This is a centrally controlled system which finds locations of several waste bins.

• These sensors constitute a smart waste bin system to send information like level of smart waste bin and locations of the smart waste bins the flow of collected data is shown.

• Send waste collector at that location Store data band analysis found to be above than a prescribed danger level, an alert system is generated and the subsequent message is circulated and broadcasted to all the concerned cities.

• The implementation of the model consisting of sensors.

• Predominantly composition of Information and Communication Technologies (ICT) and the Internet of Things (IoT) is an innovative city environment.

• It develops, deploys, and promotes sustainable development to address growing urbanization challenges in smart applications such as computational power, communication bandwidth, and others.

• ICT is an intelligent network of connected objects and machines; it has an essential role in communicating the data to the cloud with various wireless technology such as Wi-Fi, Bluetooth, Global Position System (GPS), and Cellular Communication environment. Waste management involves the processes of waste collection, transportation, processing, as well as waste recycling or disposal. Sustainable waste management systems include advanced management strategies to minimize environmental challenges and protect resources.

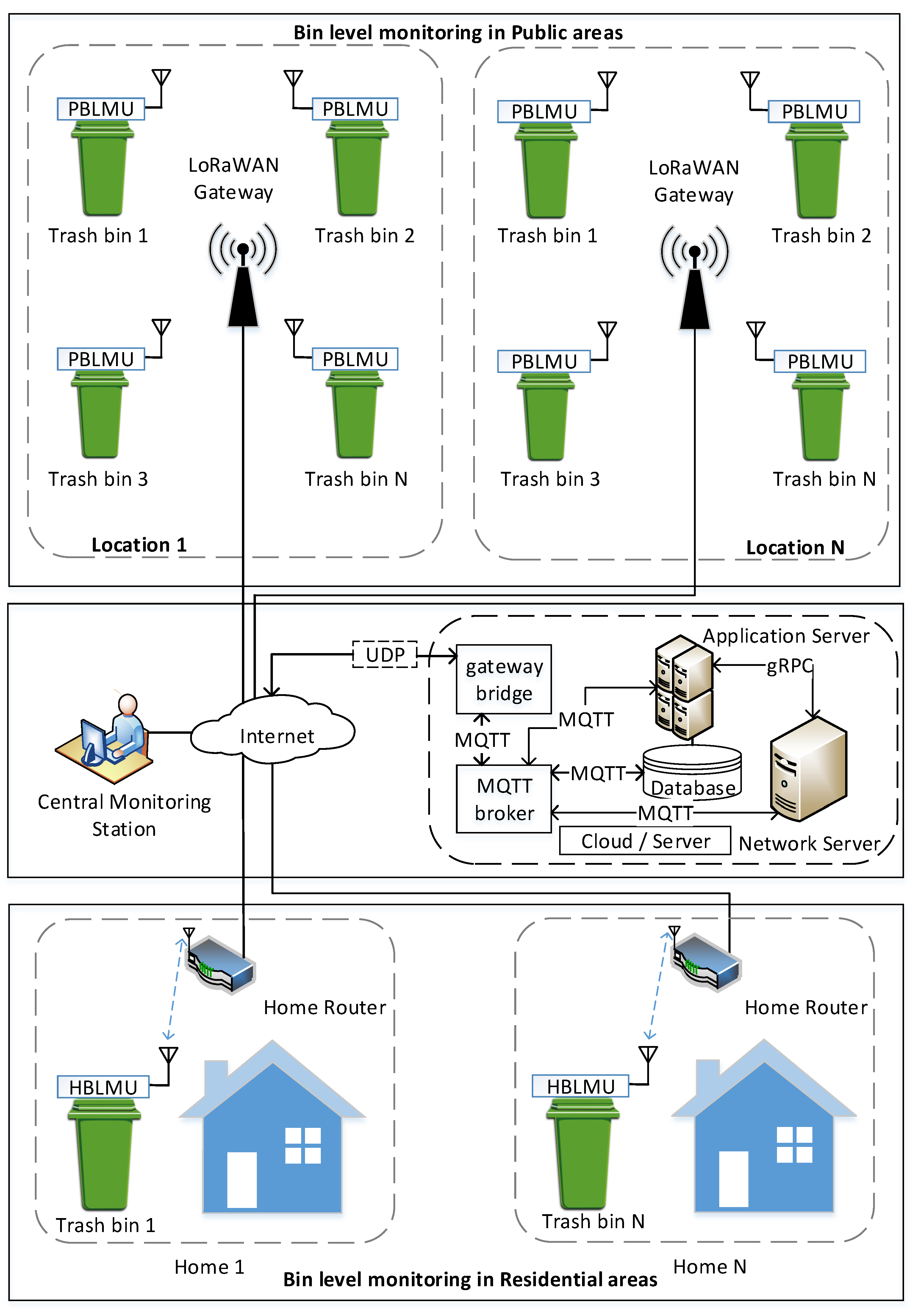
****

Fig.7. Bin level monitoring in residential areas

* 1. **Solution and Technical Architecture**

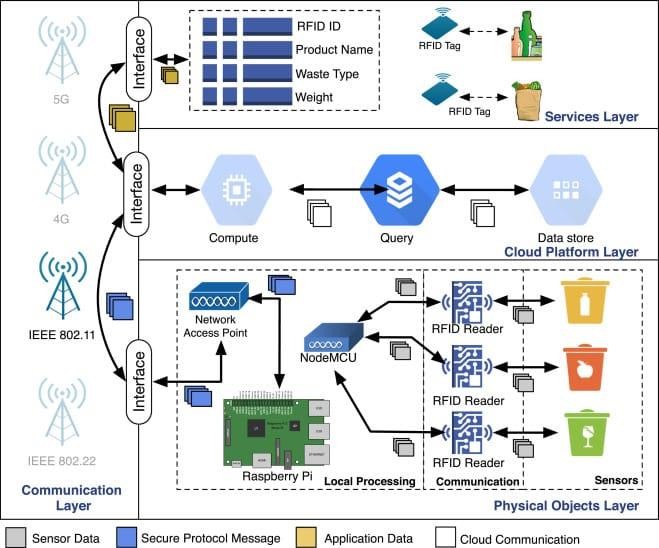
****

Fig.8. Technical Architecture

At the core of smart bin technology is a series of wireless ultrasonic sensors that detect fill levels. The IoT devices communicate waste data to sanitation department workers in real-time. Workers can then use the insights from this data to make decisions on waste management adjustments. The adoption of these smart waste bins will promote a cleaner and safer environment for companies, citizens, and visitors.

Ultimately, this technology will utilize garbage trucks where they are needed most and reduce wasted trips, lowering emissions and, thus, creating a more eco-friendly solution. Smart trash bins have already proven to be highly efficient with just a small number of early adopters. What makes these dumpsters smart is that IoT devices have been installed to measure trash levels and other metrics associated with waste removal.

As trash bins fill up, sanitation workers are notified, allowing them to route pick-up vehicles more strategically. Additionally, emptying dumpers before they are excessively filled is more efficient than allowing them to overflow, which can be a community health hazard.

* 1. **User stories**

Smart waste management is about using technology and data to create a moreefficient waste industry. Based on IoT (Internet of Things) technology, smart waste management aims to optimize resource allocation, reduce running costs, and increase the sustainability of waste services.

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 behavior. This empowers municipalities, cities, and waste collectors to optimize their waste operations, become more sustainable, and make more intelligent business decisions. The waste sector has traditionally been a static industry.

And it is only in recent years that we have started to see modernizations to the ways of working in waste management. With the birth of IoT technology and new innovations becoming commercially available, waste authorities are increasingly looking to smart solutions, as a way of addressing budget cuts and ambitious sustainability targets.

**CHAPTER 6**

**PROJECT PLANNING AND SCHEDULING**

* 1. **Sprint Planning and Estimation**

When a team refines the backlog, the most important is to understand what is wanted in each backlog item. The need to estimate is a way to make everyone understand what is involved on that product increment. If I have to define how complex is a functionality, I have to understand it first.

Two examples on how this process can be helpful:

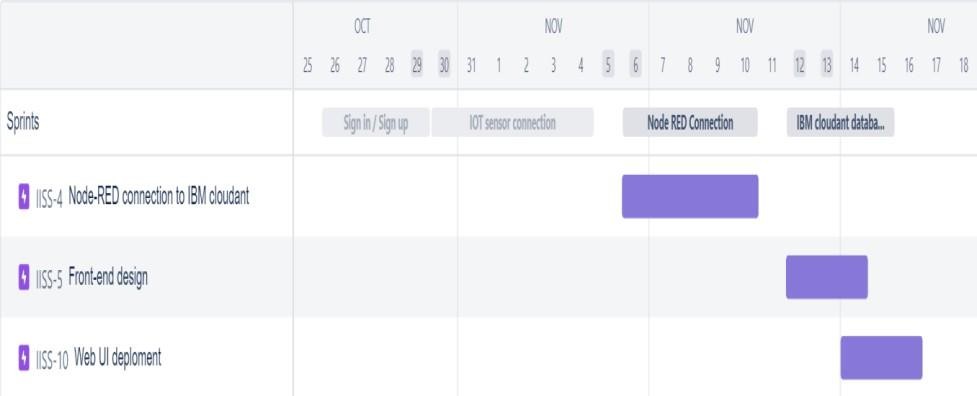
* If the teams give different values to define the size of a backlog item, it shows that there’s various levels of understanding on that subject and the team should discuss to build the common view, levelling the knowledge and diminishing the uncertainties.
* If the increment is too big or too complex, then the team should break that item in smaller ones, keeping in mind that each new backlog item should be **I**ndependent, **N**egotiable , **V**aluable, **E**stimable, **S**mall and **T**estable (the known INVEST acronym created by Mike Cohn).

For this phase, I find great consensus on the value of this process, as a way for having a good product backlog with a quite accurate midterm predictability for new features implementation, using the team’s Velocity to evaluate what can be done on the following sprints. The Team can use Story Points, T-shirt sizes or other kind of values to define backlog items size/complexity. Some Teams are more comfortable on using an Effort value to define a high level estimate of time for that work. Others just break the work so each Backlog Item can have the same size (#NoEstimates). The importance of this refinement process resides on getting a clear understanding of whats need to be done and, by taking advantage of that process, to have a size of each wanted product increment for future calculation of team velocity.

Still on the refinement sessions, could we do a more detailed estimation of the work needed? We could, but when we are refining the backlog items we don’t know when the work will be done, nor what will be the product state when the item is to be done. That increment may even loose priority and/or feasibility, and may never be done. So, if we do a more detailed evaluation of the work needed on an early phase, it can just be a waste of time.

* 1. **Sprint Delivery Schedule**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirement(Epic)** | **User Story Number** | **User Story/Task** | **Story Points** | **Priority** | **Team Members** |
| Sprint-1 | Login | USN-1 | As a Administrator ,I need to give user id and pass code for ever workers over there in municipality | 10 | High | Karthika M |
| Sprint-1 | Login | USN-2 | As a Co-Admin, I’ll control the waste level bymonitoringthemvairealtimewebportal.Oncethefillinghappens,I’llnotifytrashtruckwith  Location of bin with bin ID | 10 | High | Mariselvi S |
| 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 | Sangeetha M |
| Sprint-3 | Dashboard | USN-4 | As a Local Garbage Collector, I’II gather all the  waste from the garbage, load it onto a garbage truck ,and deliver it to Landfills | 20 | Medium | Divya T |

**6.3 JIRA REPORT**

**CHAPTER 7**

**CODING AND SOLUTIONING**

* 1. **Feature 1**

## Wokwi code for Sensor transmission

#include <WiFi.h> // library for wifi

#include <PubSubClient.h> // library for MQTT #include <LiquidCrystal\_I2C.h>

#include <mjson.h> LiquidCrystal\_I2C lcd(0x27, 20, 4);

// credentials of IBM Accounts #define ORG "9gbe4w" // IBM organisation id

#define DEVICE\_TYPE "SWMSMC" // Device type mentioned in ibm watson iot platform

#define DEVICE\_ID "ibmproject" // Device ID mentioned in ibm watson iot platform #define TOKEN "sUNA41tG6-Pq)0rk5X" // Token

// customise above values char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // server name

char publishTopic[] = "iot-2/evt/data/fmt/json"; // topic name and type of event perform and format

in which data to be send

char topic[] = "iot-2/cmd/led/fmt/String"; // cmd Represent type and command is test format of strings

char authMethod[] = "use-token-auth"; // authentication method char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID; //Client id

//

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

#define ECHO\_PIN 12

#define TRIG\_PIN 13 float dist;

String data3;

bool SealBin = true; void setup()

{

**Serial**.begin(115200); pinMode(LED\_BUILTIN, OUTPUT); pinMode(TRIG\_PIN, OUTPUT); pinMode(ECHO\_PIN, INPUT);

//pir pin pinMode(34, INPUT);

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

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

}

float readcmCM()

{

digitalWrite(TRIG\_PIN, LOW); delayMicroseconds(2); digitalWrite(TRIG\_PIN, HIGH); delayMicroseconds(10); digitalWrite(TRIG\_PIN, LOW);

int duration = pulseIn(ECHO\_PIN, HIGH); return duration \* 0.034 / 2;

}

void loop()

{

lcd.clear();

publishData(); delay(500);

if (!client.loop())

{

mqttConnect(); // function call to connect to IBM

}

}

/\* retrieving to cloud \*/ void wifiConnect()

{

**Serial**.print("Connecting to "); **Serial**.print("Wifi"); WiFi.begin("Wokwi-GUEST", "", 6); while (WiFi.status() != WL\_CONNECTED)

{

delay(500);

**Serial**.print(".");

}

**Serial**.print("WiFi connected, IP address: ");

**Serial**.println(WiFi.localIP());

}

void mqttConnect()

{

if (!client.connected())

{

**Serial**.print("Reconnecting MQTT client to ");

**Serial**.println(server);

while (!client.connect(clientId, authMethod, token))

{

**Serial**.print("."); delay(500);

}

initManagedDevice();

**Serial**.println();

}

}

void initManagedDevice()

{

if (client.subscribe(topic))

{

**Serial**.println("IBM subscribe to cmd OK");

}

else

{

**Serial**.println("subscribe to cmd FAILED");

}

}

void publishData()

{

float cm = readcmCM();

if(digitalRead(34)) //pir motion detection

{

**Serial**.println("Motion Detected"); **Serial**.println("Lid Opened"); digitalWrite(15, HIGH);

if(digitalRead(34)== true)

{

if(cm <= 100) //Bin level detection

{

digitalWrite(2, HIGH);

**Serial**.println("High Alert!!!,Trash bin is about to be full");

**Serial**.println("Lid Closed"); lcd.print("Full! Don't use"); delay(2000);

lcd.clear(); digialWrite(4, LOW); digitalWrite(23, LOW);

}

else if(cm > 100 && cm < 180)

{

digitalWrite(4, HIGH);

**Serial**.println("Warning!!,Trash is about to cross 50% of bin level"); digitalWrite(2, LOW);

digitalWrite(23, LOW);

}

else if(cm > 180)

{

digitalWrite(23, HIGH); **Serial**.println("Bin is available"); digitalWrite(2,LOW); digitalWrite(4, LOW);

}

delay(10000);

**Serial**.println("Lid Closed");

}

else

{

**Serial**.println("No motion detected"); digitalWrite(2, LOW); digitalWrite(15, LOW); digitalWrite(4, LOW); digitalWrite(23, LOW);

}

}

else

{

digitalWrite(15, LOW);

}

if(cm <= 100)

{

digitalWrite(21,HIGH);

String payload = "{\"High\_Alert\":"; payload += cm;

payload += " }"; **Serial**.print("\n"); **Serial**.print("Sending payload: "); **Serial**.println(payload);

if (client.publish(publishTopic, (char\*) payload.c\_str())) // if data is uploaded to cloud successfully,prints publish ok else prints publish failed

{

**Serial**.println("Publish OK");

}

}

else if(cm <= 180)

{

digitalWrite(22,HIGH);

String payload = "{\"Warning\":"; payload += cm ;

payload += " }"; **Serial**.print("\n"); **Serial**.print("Sending payload: "); **Serial**.println(payload);

if(client.publish(publishTopic, (char\*) payload.c\_str()))

{

**Serial**.println("Publish OK");

}

Else

{

**Serial**.println("Publish FAILED");

}

}

else if(cm > 180)

{

digitalWrite(23,HIGH); String payload = "{"; payload += cm;

payload += " }"; **Serial**.print("\n"); **Serial**.print("Sending payload: "); **Serial**.println(payload);

if (client.publish(publishTopic, (char\*) payload.c\_str())) // if data is uploaded to cloud successfully,prints publish ok else prints publish failed

{

**Serial**.println("Publish OK");

}

}

float inches = (cm / 2.54); //print on lcd lcd.setCursor(0,0);

lcd.print("Inches"); lcd.setCursor(4,0); lcd.setCursor(12,0); lcd.print("cm"); lcd.setCursor(1,1); lcd.print(inches, 1); lcd.setCursor(11,1); lcd.print(cm, 1); lcd.setCursor(14,1); delay(1000); lcd.clear();

}

//handles commands from user side

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

{

**Serial**.print("callback invoked for topic: ");

**Serial**.println(subscribetopic);

for (int i = 0; i < payloadLength; i++) {

data3 += (char)payload[i];

}

**Serial**.println("data: "+ data3);

const char \*s =(char\*) data3.c\_str(); double pincode = 0;

const char \*buf; int len;

if (mjson\_find(s, strlen(s), "$.command", &buf, &len)) // And print it

{

String command(buf,len);

if(command=="\"SealBin\"")

{

SealBin = true;

}

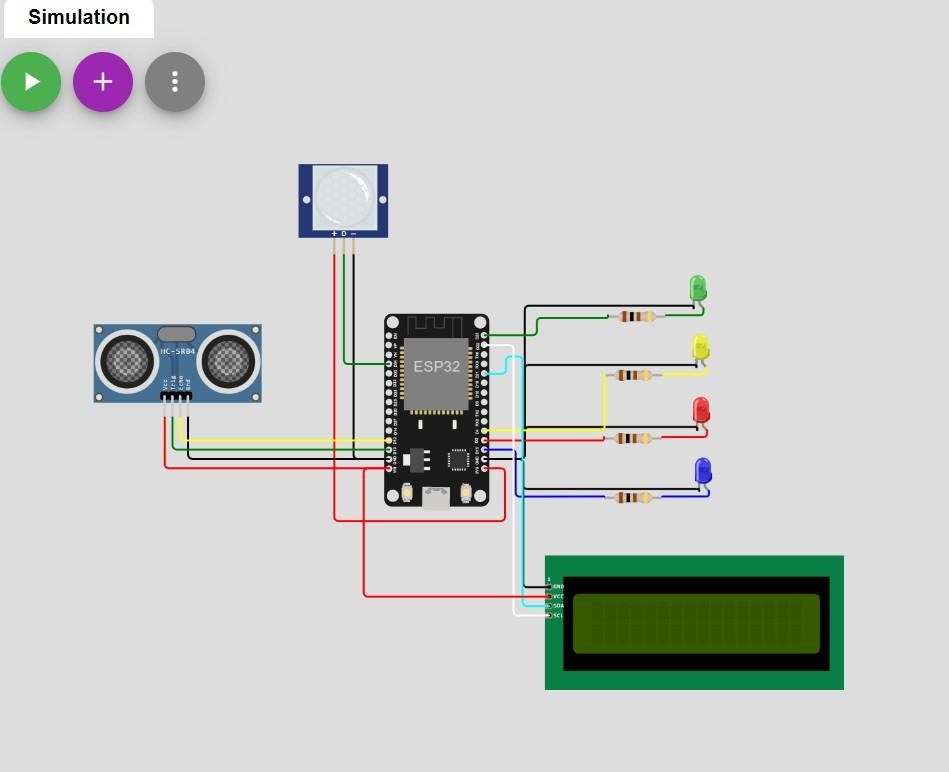
}

data3="";

}

* 1. **Feature 2**

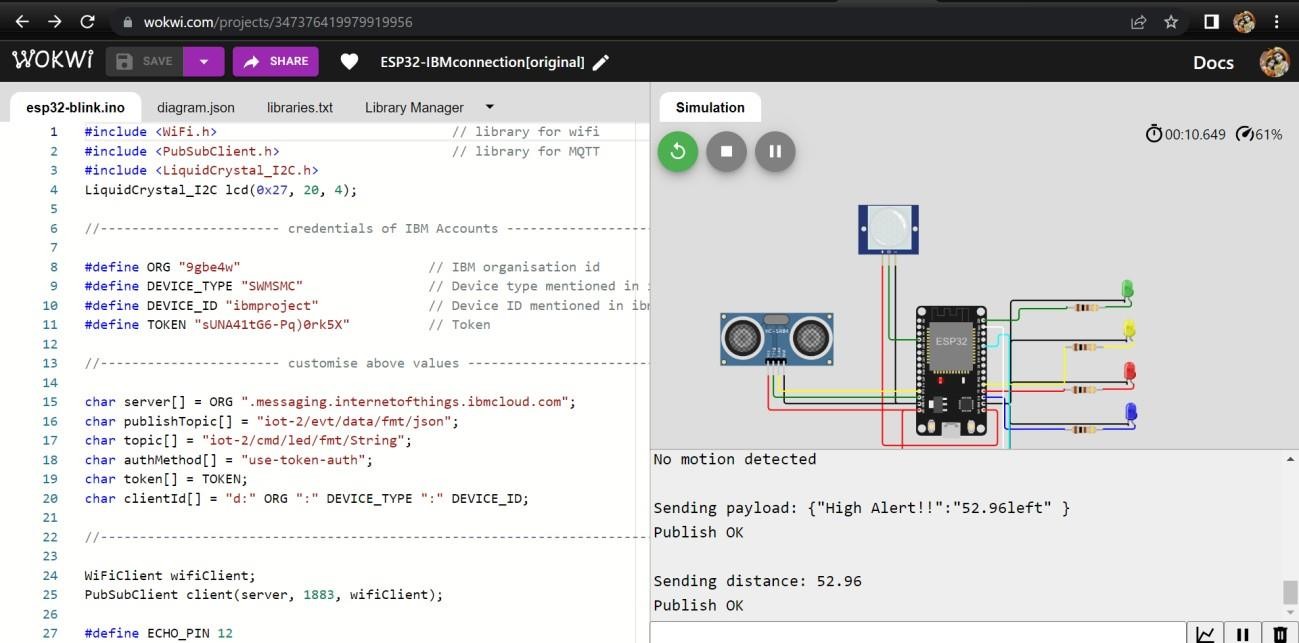
# . Sensor Connection Setup



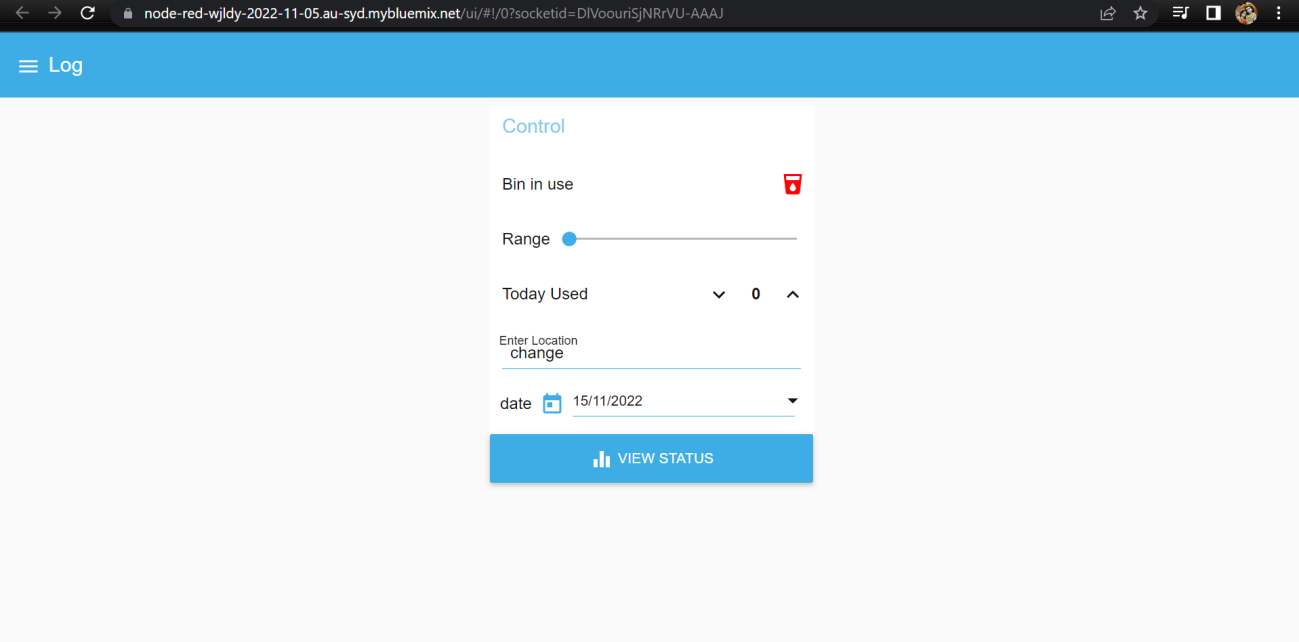
PHYSICAL COMPONENTS:

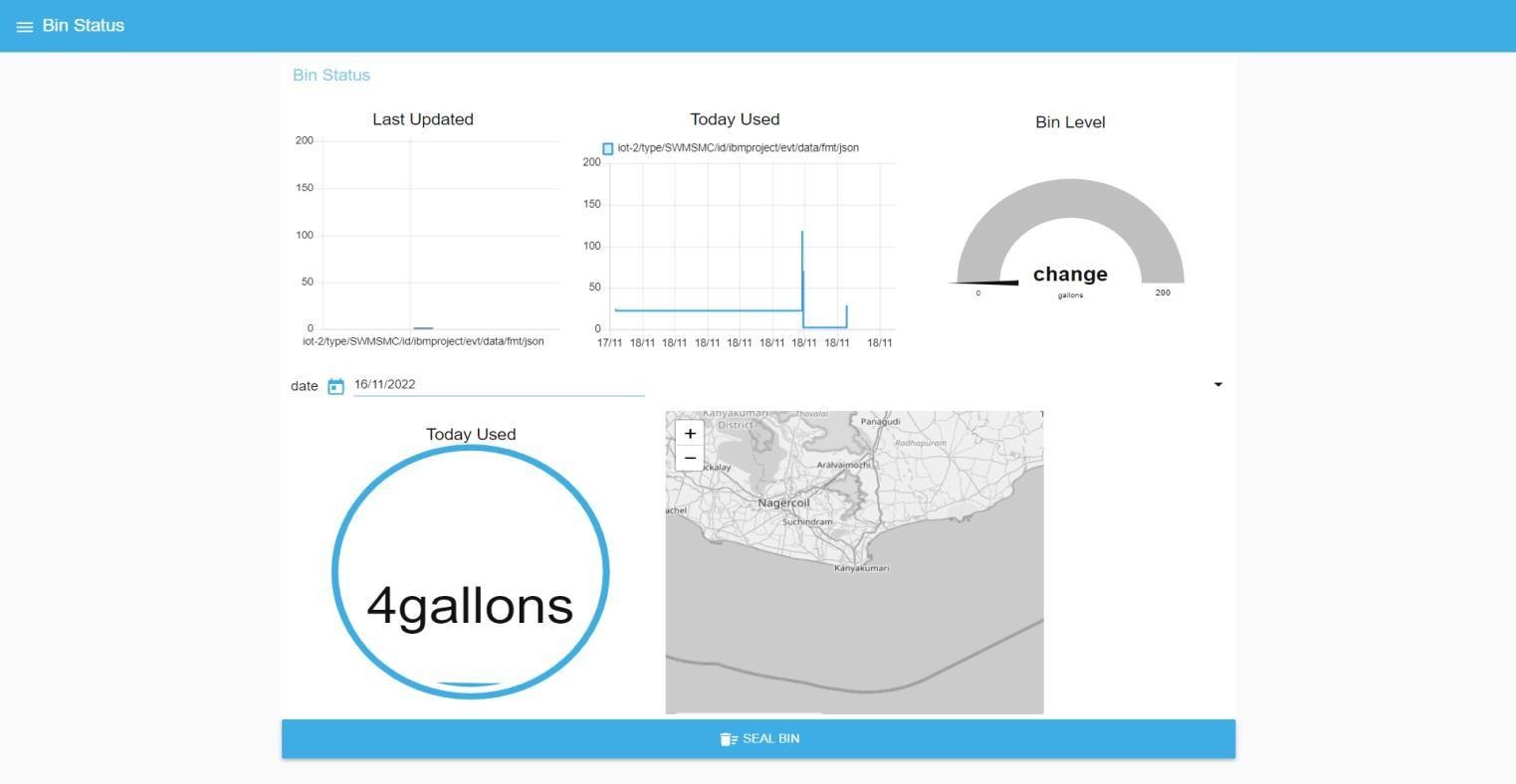
* + - PIR MOTION SENSOR
    - ULTRASONIC DISTANCE SENSOR
    - ESP32-ARDUINO MICROCONTROLLER

OUTPUT: WOKWI SETUP



WEB UI





The admin gets notification when the bin detects motion and if the bin level crosses 50 percent it indicates warning and if it crosses 90 percent it gives a High alert and closes the bin. If the admin wants to seal the bin the admin can command seal bin until it is accessed for cleaning.

## 

**CHAPTER 8**

**TESTING**

**8.1 Test Cases**

Maximum Size of Bin : 200 cm Safe limit: below 100 cm Minimum threshold limit of bin: 100 cm

Maximum threshold limit of bin: 180 cm

|  |  |  |  |
| --- | --- | --- | --- |
| **S.no** | **Bin Level**  **(cm filled)** | **Bin Status** | **Location** |
| 1 | 45 | Safe | Kanyakumari |
| 2 | 78 | Safe | Coimbatore |
| 3 | 112 | Warning | Trichy |
| 4 | 169 | Warning | Chennai |
| 5 | 186 | Warning | Ooty |
| 6 | 193 | High\_Alert | Tirunelveli |
| 8 | 0 | Safe | Chengalpattu |
| 9 | 35 | Safe | Madurai |
| 10 | 101 | Warning | Salem |
| 11 | 132 | Warning | Thanjavore |
| 12 | 158 | Warning | Vellore |
| 13 | 93 | High\_Alert | Erode |
| 14 | 93 | High\_Alert | Karur |
| 15 | 93 | High\_Alert | Cuddalore |
| 16 | 30 | Safe | Kumbakonam |
| 17 | 110 | Warning | Ambur |
| 18 | 180 | Warning | Sivakasi |
| 19 | 195 | High\_Alert | Neyveli |
| 20 | 80 | Safe | Krishnagiri |

**Note:** The bin location provided above is default. When the user access the bin , the location and status of the bin displayed to the admin.

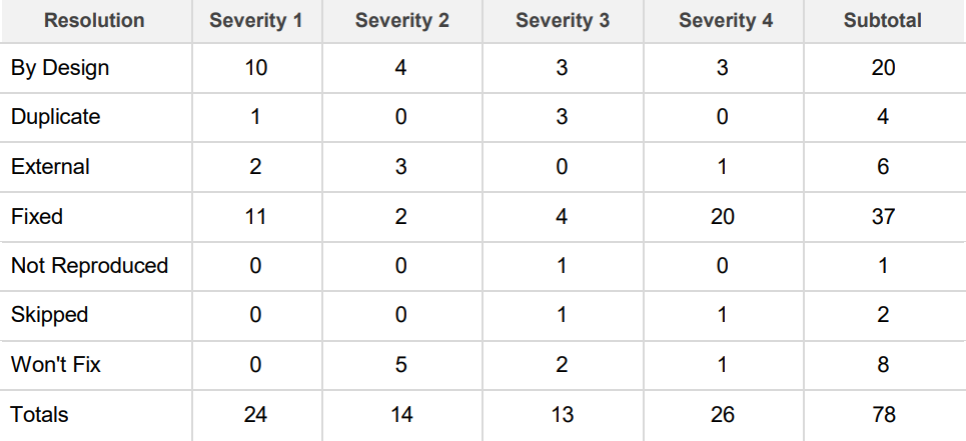
# USER ACCEPTANCE TESTING

# Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Smart Waste Management System project at the time of the release to User Acceptance Testing (UAT).

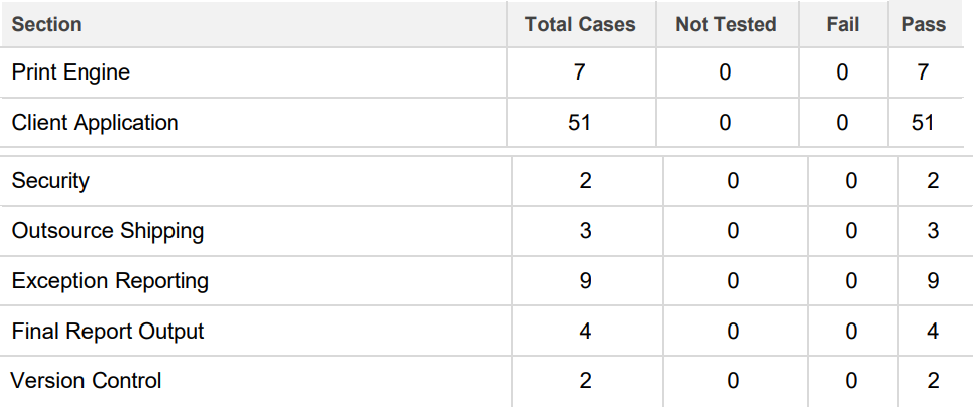
#### 1.Defect Analysis

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



#### TEST CASE ANALYSIS

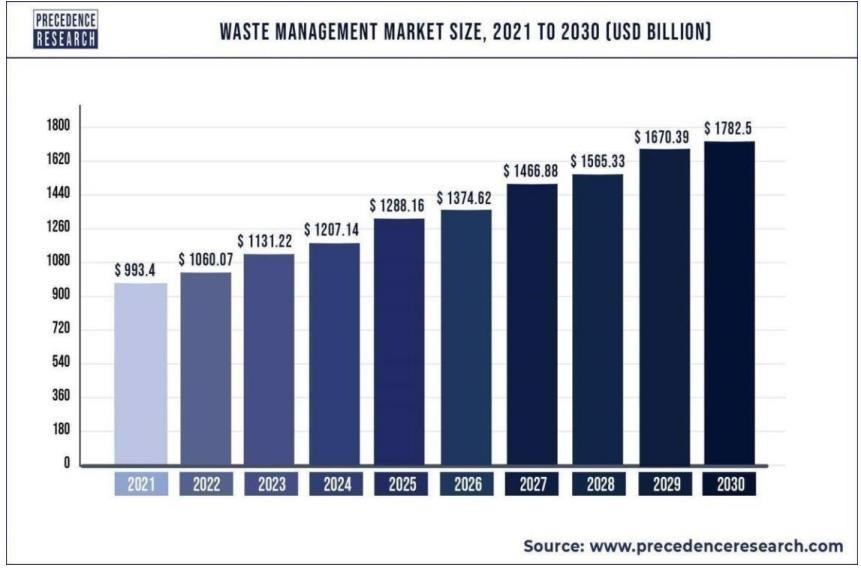
This report shows the number of test cases that have passed , failed and untested.



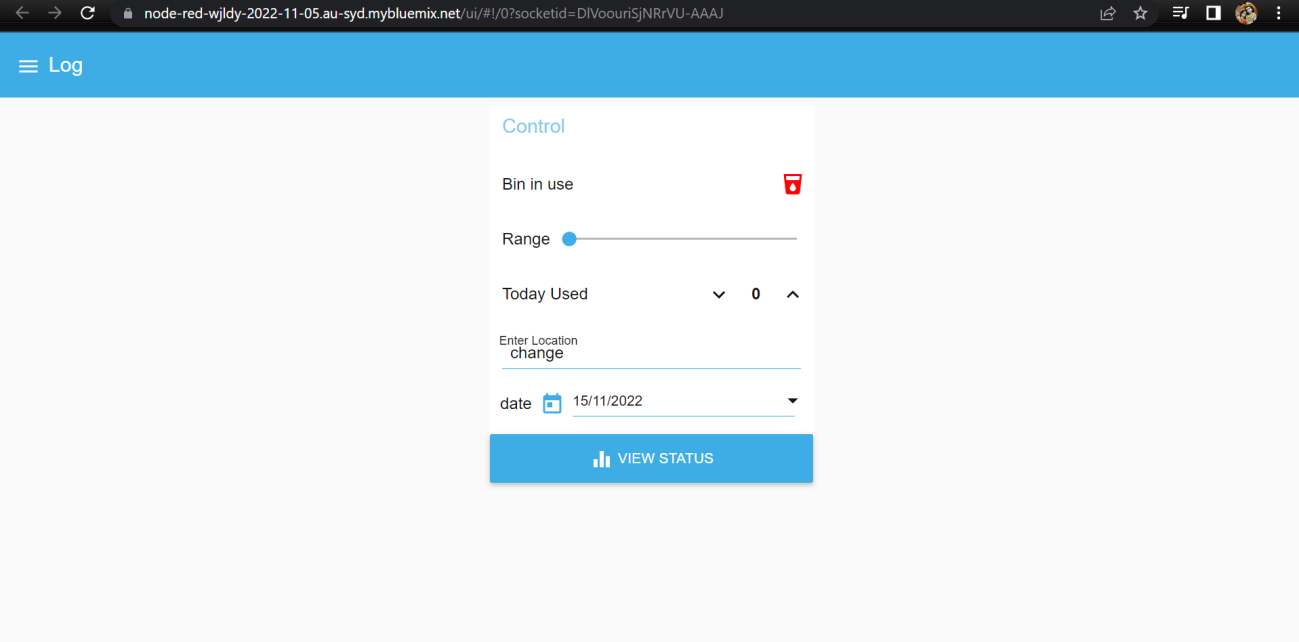
**CHAPTER 9**

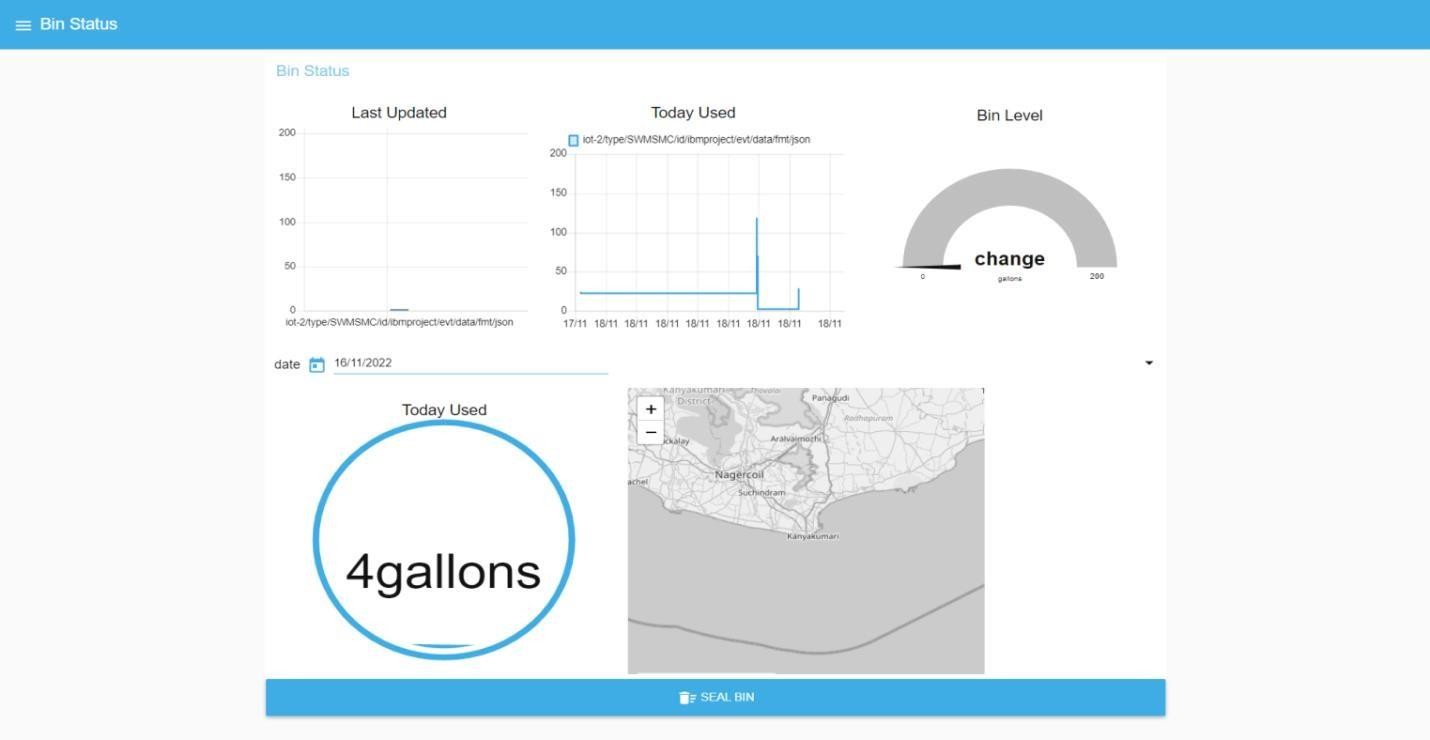
**RESULTS**

## Performance Metrics



**9.2 . Admin Web UI**





**CHAPTER 10**

## ADVANTAGES AND DISADVANTAGES

**10.1 ADVANTAGES**

1] Reduction in Collection Cost

2] No Missed Pickups

3] Reduced Overflows

4] Waste Generation Analysis

5] CO2 Emission Reduction

## 10.2 DISADVANTAGES

* System requires a greater number of waste bins for separate waste collection as per population in the city.
* This results into high initial cost due to expensive smart dustbins compare to other methods. Sensor nodes used in the dustbins have limited memory size.

According to the author there may be several disadvantages such as increasing cost of thedustbin. For example, if there are three different levels then three sensors has to be placed; one sensor for each level. Also rough action and usage of the user may cause damages to the sensors.

# CHAPTER 11

# CONCLUSION

A Smart Waste Management system that is more effective than the one in use now is achievable by using sensors to monitor the filling of bins. Our conception of a "smart waste management system" focuses on monitoring waste management, offering intelligent technology for waste systems, eliminating human intervention, minimizing human time and effort, and producing a healthy and trash- free environment. The suggested approach can be implemented in smart cities where residents have busy schedules that provide little time for garbage management. If desired, the bins might be put into place in a metropolis where a sizable container would be able to hold enough solid trash for a single unit. But these may price bit high.

# CHAPTER 12

# FUTURE SCOPE

# FUTURE SCOPE:

There are several future works and improvements for the proposed system, including the following:

* Changes the system of user authentication and atomic lock of bins, which would aid in protecting the bin from damage .
* Having case study or data analytics on the type and times waste is collected on different days or seasons, making the bin level predictable and remove the reliance on electronic components, and fixing the coordinates.
* Improving the Server's and Android's graphical interfaces

# CHAPTER 13

# APPENDIX

### Esp32 - Microcontroller :

ESP32 is a series of low-cost, low-power system on a chip microcontrollers with integrated Wi-Fi and dual-mode Bluetooth.

* + Memory: 320 KiB
  + SRAM CPU: Tensilica Xtensa LX6 microprocessor @ 160 or 240 MHz
  + Power: 3.3 V DC
  + Manufacturer: Espressif Systems
  + Predecessor: ESP8266
* **Sensors :**
  + PIR motion sensor: PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range.
  + Ultrasonic Distance Sensor : Ultrasonic Sensors measure the distance to the target by measuring the time between the emission and reception.

## 13.1. Source code

#include <WiFi.h> // library for wifi

#include <PubSubClient.h> // library for MQTT #include <LiquidCrystal\_I2C.h>

#include <mjson.h> LiquidCrystal\_I2C lcd(0x27, 20, 4);

// credentials of IBM Accounts #define ORG "9gbe4w" // IBM organisation id

#define DEVICE\_TYPE "SWMSMC" // Device type mentioned in ibm watson iot platform

#define DEVICE\_ID "ibmproject" // Device ID mentioned in ibm watson iot platform #define TOKEN "sUNA41tG6-Pq)0rk5X" // Token

// customise above values char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // server 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 <= 100) //Bin level detection

{

digitalWrite(2, HIGH);

**Serial**.println("High Alert!!!,Trash bin is about to be full");

**Serial**.println("Lid Closed"); lcd.print("Full! Don't use"); delay(2000);

lcd.clear(); digitalWrite(4, LOW); digitalWrite(23, LOW);

}

else if(cm > 100 && cm < 180)

{

digitalWrite(4, HIGH);

**Serial**.println("Warning!!,Trash is about to cross 50% of bin level"); digitalWrite(2, LOW);

digitalWrite(23, LOW);

}

else if(cm > 180)

{

digitalWrite(23, HIGH); **Serial**.println("Bin is available"); digitalWrite(2,LOW); digitalWrite(4, LOW);

}

delay(10000);

**Serial**.println("Lid Closed");

}

else

{

**Serial**.println("No motion detected"); digitalWrite(2, LOW); digitalWrite(15, LOW); digitalWrite(4, LOW); digitalWrite(23, LOW);

}

}

else

{

digitalWrite(15, LOW);

}

if(cm <= 100)

{

digitalWrite(21,HIGH);

String payload = "{\"High\_Alert\":"; payload += cm;

payload += " }"; **Serial**.print("\n"); **Serial**.print("Sending payload: "); **Serial**.println(payload);

if (client.publish(publishTopic, (char\*) payload.c\_str())) // if data is uploaded to cloud successfully,prints publish ok else prints publish failed

{

**Serial**.println("Publish OK");

}

}

else if(cm <= 180)

{

digitalWrite(22,HIGH);

String payload = "{\"Warning\":"; payload += cm ;

payload += " }"; **Serial**.print("\n"); **Serial**.print("Sending payload: "); **Serial**.println(payload);

if(client.publish(publishTopic, (char\*) payload.c\_str()))

{

**Serial**.println("Publish OK");

}

else

{

**Serial**.println("Publish FAILED");

}

}

else if(cm > 180)

{

digitalWrite(23,HIGH); String payload = "{"; payload += cm;

payload += " }"; **Serial**.print("\n"); **Serial**.print("Sending payload: "); **Serial**.println(payload);

if (client.publish(publishTopic, (char\*) payload.c\_str())) // if data is uploaded to cloud successfully,prints publish ok else prints publish failed

{

**Serial**.println("Publish OK");

}

}

float inches = (cm / 2.54); //print on lcd lcd.setCursor(0,0);

lcd.print("Inches"); lcd.setCursor(4,0); lcd.setCursor(12,0); lcd.print("cm"); lcd.setCursor(1,1); lcd.print(inches, 1); lcd.setCursor(11,1); lcd.print(cm, 1); lcd.setCursor(14,1); delay(1000); lcd.clear();

}

//handles commands from user side

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

{

**Serial**.print("callback invoked for topic: ");

**Serial**.println(subscribetopic);

for (int i = 0; i < payloadLength; i++) {

data3 += (char)payload[i];

}

**Serial**.println("data: "+ data3);

const char \*s =(char\*) data3.c\_str(); double pincode = 0;

const char \*buf; int len;

if (mjson\_find(s, strlen(s), "$.command", &buf, &len)) // And print it

{

String command(buf,len); if(command=="\"SealBin\"")

{

SealBin = true;

}

}

data3="";

}

## 13.2 . GITHUB LINK

**LINK :** https://github.com/IBM-EPBL/IBM-Project-42941-1660711529