Nalaiya Thiran

Vivekanandha College of Technology for Women

Department of Computer Science and Engineering

Smart Waste Management System For Metropolitan Cities

Team ID: PNT2022TMID30606

Team Members:

613019104091	S.Valarmathi
613019104089	R.P.Umadevi
613019104065	V.Sakthi
613019104501	D.Sathiya

Project Guide:

Industry mentor: Mr. Dinesh

Faculty Mentor: Mr. Vivek Pandiya Raj

INDEX

1. INTRODUCTION

1.1 Project Overview

Rapid increase in population, has led to the improper waste management in cities resulting in increased pests and spreading of diseases. Nowadays, the Garbage Collecting Vehicle (GCV) collects the waste twice or thrice in a week. So, the problem is over flowing of wastages on the roads. Hence, to overcome this limitation, in this paper a scheme on smart waste management using Wireless Sensor Networks (WSN) and IoT (Internet of Things) is proposed. The garbage bins are deployed with sensors and are networked together using WSN. The sensors deployed in the garbage bins collect the data for every determined interval. Once the threshold is reached, it raises a request to the GCA (Garbage Collector Agent). This agent collects the requests of all the filled vehicles and communicate using IoT framework. The experimental simulation is done in proteus tool. A hardware prototype is developed for the proposed framework. Analysis of the proposed scheme provides better results in waste management.

1.2 Purpose

Most of the metro cities globally poses a challenge on effective solid waste management and maintenance of the waste bins. An IOT enabled Smart Waste Bin with real time monitoring is designed and implemented using ultra sonic sensor, parallel plate capacitance and status of the bins are communicated effectively using the cloud and Node MCU. The results prove the efficiency of the designed smart bins qualitatively. A smart waste management system incorporating robotic smart bins, where the smart bins has the mobility to move to the waste dockyard by localizing itself in the environment is proposed. This system could find an application in smart building where the waste management could be practiced autonomously in a smarter way.

2. LITERATURE SURVEY

2.1 Existing Problem

The waste management in metropolitan cities has serious environmental impacts like water pollution, methane emissions, and soil degradation. The average density of Indian municipal waste at the point of collection varies from 400 to 600 kg per cubic meter. At the landfill site, however, the density is much higher because of compaction and putrefaction. Waste incineration (including Waste to Energy) and other thermal processes are local sources of air pollution, constituting additional health risk factors to city dwellers, who often already have to cope with serious air contamination issues. Ecosystems vary widely from location to location. However, one of the most outsize consequences of our global waste problem manifests itself in relation to our marine life and waterways. Simply put, it affects the people who depend on the ocean for their livelihoods

2.2 References

Paper Title	Author	Outcome
IOT Based Waste Management for Smart City	1) Parkash Tambare 2) Prabu Venkatachalam	In the current situation, we frequently observe that the trash cans or dust cans that are located in public spaces in cities are overflowing due to an increase in the amount of waste produced each day. We are planning to construct "IoT Based Waste Management for Smart Cities" to prevent this from happening because it makes living conditions for people unsanitary and causes unpleasant odours in the

		surrounding area. There are numerous trash cans scattered throughout the city or on the campus that are part of the proposed system. Each trash can is equipped with a low-cost embedded device that tracks the level of the trash cans and an individual ID that will enable it to be tracked and identified.
Design a Smart Waste Bin for Smart Waste Management	1) Aksan Surya Wijaya 2) Zahir Zainuddin 3) Muhamm ad Niswar	In this paper, we presented the smart wastebin that can managed the waste in a smart city project. The system consist of sensors to measure the weight of waste and the level of waste inside the bin. The system also adapt with network environment, to manage all information from waste management. As the result we proposed a prototype of smart waste-bin that suitable for many kind of conventional waste-bin.
Waste Management Initiatives in India For Human Wellbeing	1) Dr. Raveesh Agarwal 2) Mona Chaudhary 3) Jayveer Singh	The objective of this paper is to examine the present methods used in India for the welfare of its people in different waste management efforts. The other goal is to offer advice on how to make Indian municipalities' trash disposal procedures better. On secondary research, this essay is founded. The system is improved by looking at the reports that have already been written about waste management and the

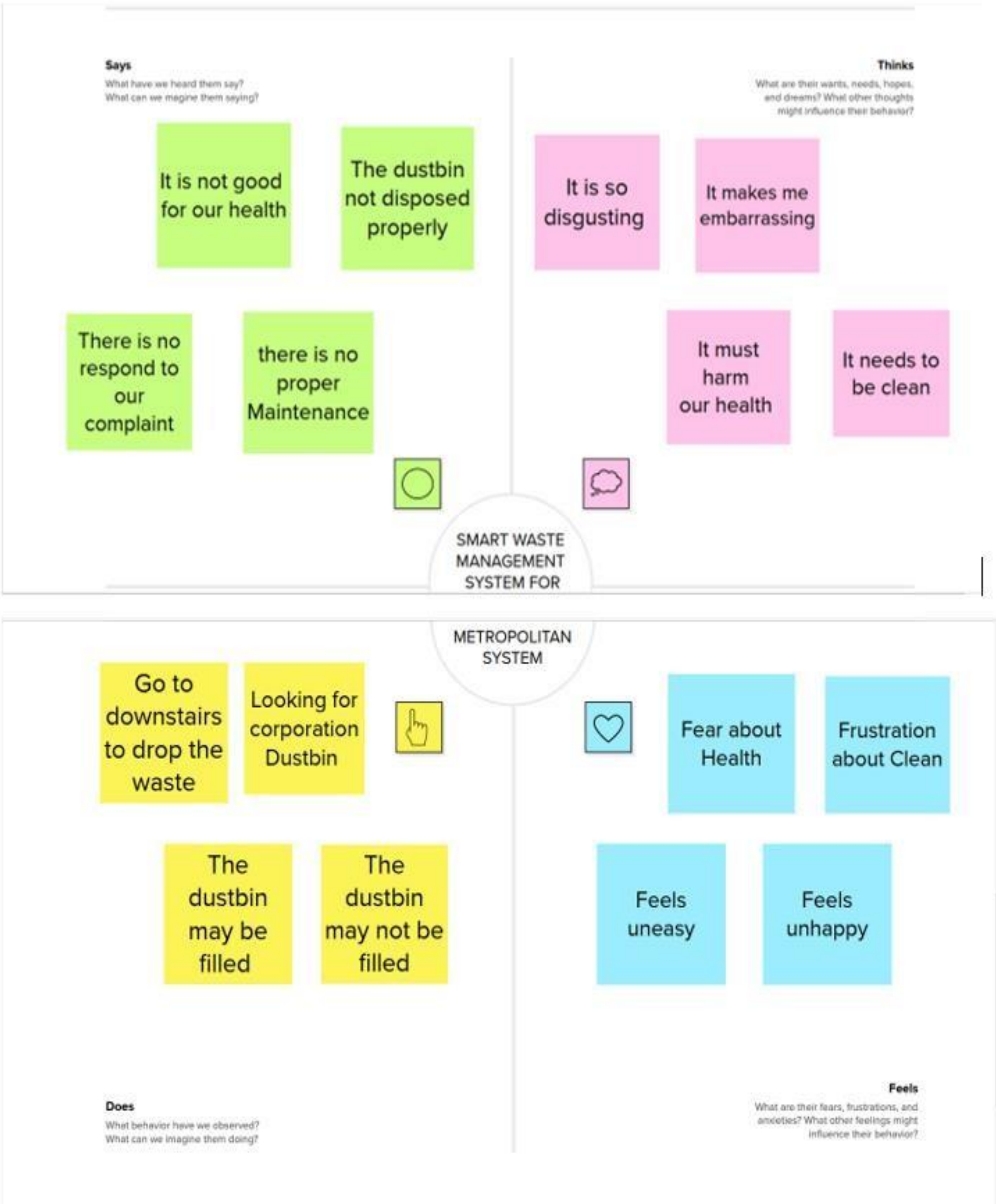
		suggestions made for improvement by planners, NGOs, consultants, government accountability organisations, and important business leaders. It provides in-depth understanding of the various waste management programmes in India and identifies areas where waste management might be improved for societal benefit.
Smart Solid Waste Management.	1) Mohd Helmy Abd Wahab	At the time of trash disposal, the material to be recycled could be identified using RFID technology.
Analysis of Load cell.	1) Ranjeet Kumar 2) Sandeep Chhabra	Load Cells 4.1 General Load Cell related information A load cell is meant to measure the size of a mass but actually is a force sensor which transforms force into an electrical signal. The load cell needs the earth gravity to work. Every mass is attracted by the earth gravimetric field, that force is named "load".
IoT based smart garbage collection system	1) Rahul Kumar Borah 2) Sahana Shetty 3) Rahul Patidar 4) Anisha Raniwala 5) Kratee Jain	To create an effective and dynamic waste management system, the smart trash container is crucial. One of the most significant challenges for municipal organisations across the world is managing waste from its inception to transfer. Due to the daily growth in garbage, dustbins placed across finished urban areas and placed in open areas are overflowing, creating unsanitary circumstances for the residents. To maintain a

		crucial barrier from such a situation, we have proposed a remote strong waste management prototype for smart urban groups. This prototype enables common associations to remotely monitor the status of trash cans, complete web server, and profitably maintain urban areas clean by increasing the cost and time required for it.
Smart Waste Management for Green Environment	1) T. P. Fei	The system is based on Bootstrap platform. This system works on the waterfall methodology which has 4 crucial phases: planning and analysis, system design, system implementation and system testing. Using this system, operators can get the information regarding collection from trash bins. The limitations of this approach are that the resultant product has a short life and uniformity is lost after a certain period.

2.3 Problem Statement Definition

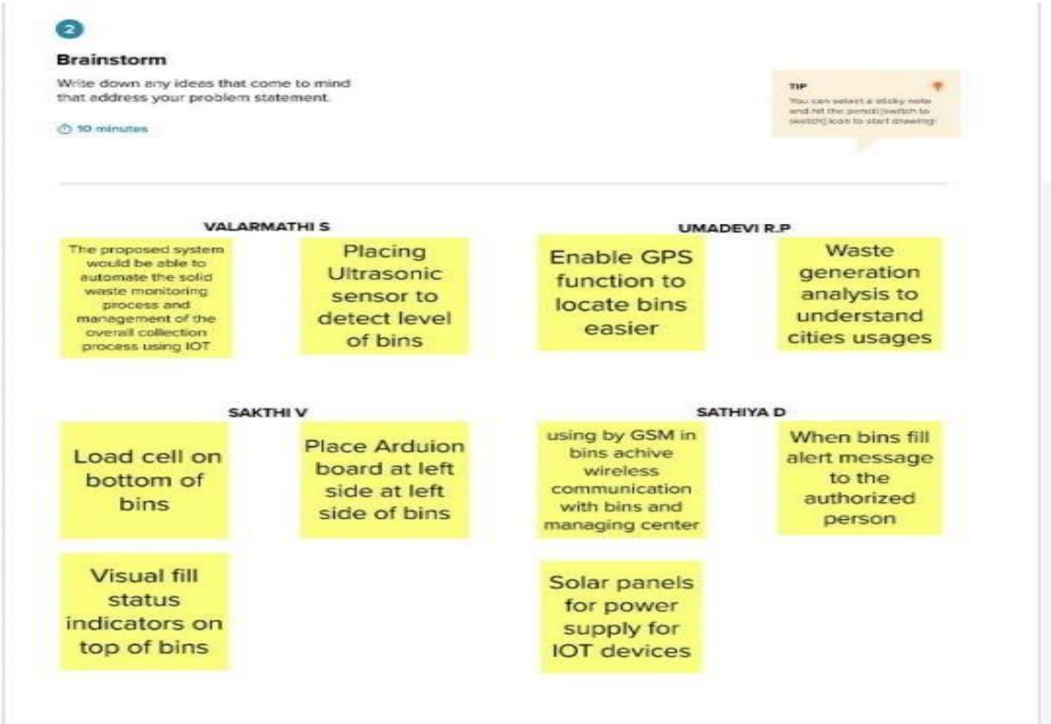
3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

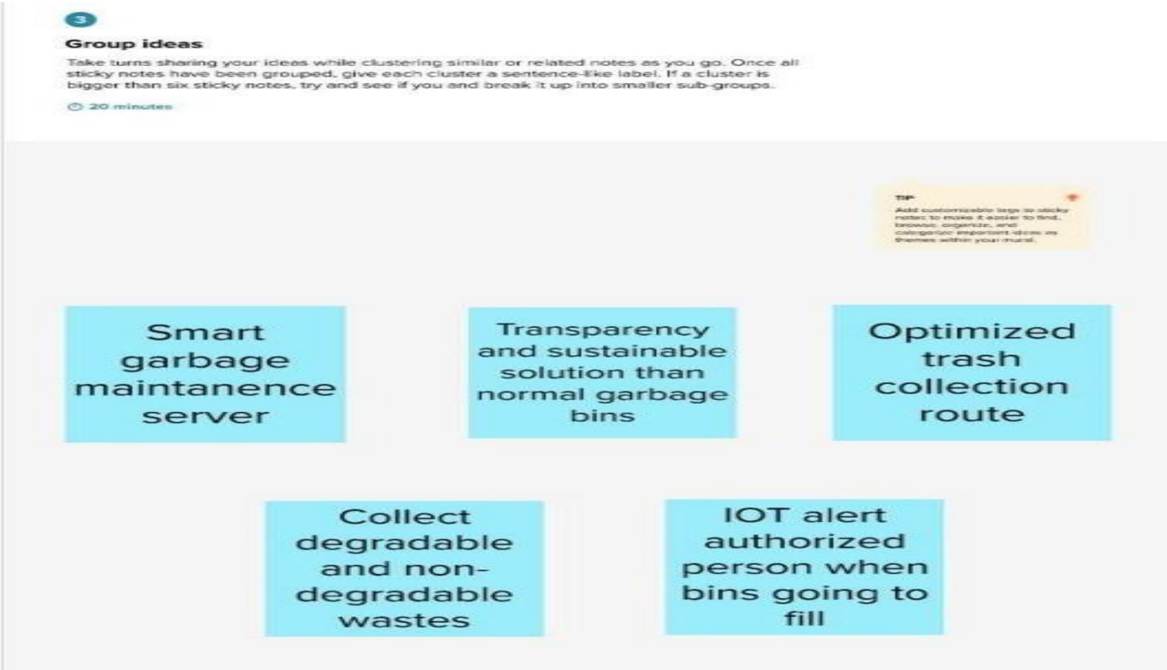


3.2 Ideation & Brainstorming

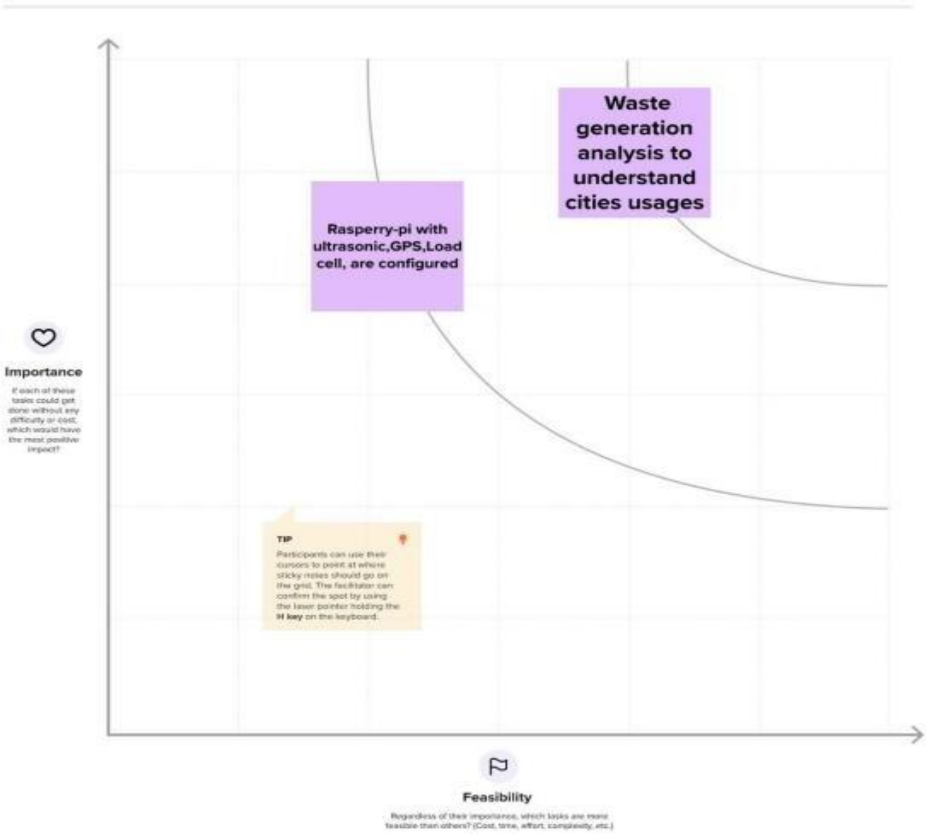
3.2.1 Brainstorm by team members



3.2.2 Group ideas



3.2.3 Prioritize



3.3 Proposed Solution

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Collection of garbage management in cities, towns and villages is a major concern and emerging problem in smart city paradigm. Lack of proper resource distribution in the process of garbage collection is great risk to sanitation, cleanliness, and health.
2.	Idea / Solution description	Collecting data from smart bins and alerting to the wastage collector. Data analysis for a smarter collection process. The data gathered in this process is analysed and it is useful insights empowering the users in their decision-making process.
3.	Novelty / Uniqueness	Identify potential waste streams. Create a waste management-focused community outreach plane.
4.	Social Impact / Customer Satisfaction	It will help us to clean the cities and gives us healthy environment. When waste is disposed or recycled in a safe, ethical, and responsible manner, it helps reduce the negative impacts of the environment.
5.	Business Model (Revenue Model)	It generates revenue through the provision of various waste management and disposal services. Recycling solutions to residential, commercial, industrial, and municipal clients.
6.	Scalability of the Solution	In this model costumer gets benefits on using smart bins by providing prediction on day-to-day analysis in waste management system.

3.4 Proposed Solution fit

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS	6. CUSTOMER CONSTRAINTS CC	5. AVAILABLE SOLUTIONS AS
People and Local authorities of Metropolitan cities are the customers.	1. Lack of infrastructure 2. Cost 3. Limitation of technology 4. Lack of participation in waste segregation	1.Central point for managing daily operations. 2.Identification, track & control the infrastructure of smart waste management.

Explore AS, differentiate

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

2. JOBS-TO-BE-DONE / PROBLEMS J&P	9. PROBLEM ROOT CAUSE RC	7. BEHAVIOUR BE
1. Avoid fixed routine for waste collection 2. Avoid overflowing of bins 3. Proper Segregation of wastes 4. Lack of information about the collecting time and area	1. Increased population growth and urbanization leads to escalation of wastes 2. Lack of notifications about the garbage to the municipality	1. Sensors are used to sense the amount of waste in the trash 2. People must wait until the next day for the garbage to be cleared by the truck drivers

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

Identify strong TR & EM

3. TRIGGERS TR	10. YOUR SOLUTION SL	8.CHANNELS of BEHAVIOUR CH
To make the environment disaster free and clean and to save the people for some toxic wastes.	1. Regular monitoring of waste disposals 2.Creating an app for monitoring the location, weight, and level of garbage cans 3. Maintenance of transport and workers.	8.1 ONLINE 1. We can monitor in live 2. People can give complaints and feedback about the work 8.2 OFFLINE Taking necessary action on collecting the garbage regularly
4. EMOTIONS: BEFORE / AFTER EM Before: People get irritated on seeing the wastes that end up on the roads After: After the cleaning of trash people feel pleasant and secure		

4. REQUIRMENT ANALYSIS

4.1 Functional Requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Bin invention	1.Our proposed model provide real time monitoring to the garbage bins placed in various locations. 2.Overflow of dustbins will be notified
FR-2	Real time monitoring	1.The garbage bins are monitored by smart sensors. 2.In addition to the percentage of fill-level, based on the historical data, the sensor predicts when the bin will become full, one of the functionalities that are not included even in the best waste management software. 3.With real-time data and predictions, you can eliminate the overflowing bins and stop collecting half- empty ones
FR-3	Processing	1.Through sensor, the percentage of garbage levels will be detected. 2.When 1the garbage level moves to critical (i.e.,80%), it gives alert notification to the security system. 3.After receiving the notification, the garbage collector collects the garbage
FR-4	User Confirmation	1.Until the notification is received from the authorised person, the garbage collector will wait for the alert message. 2.We can view the location of every bin through web app by sending GPS location from the device.

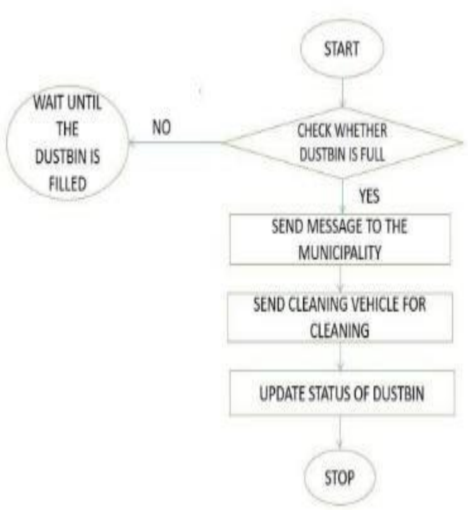
4.2 Non-Functional Requirements

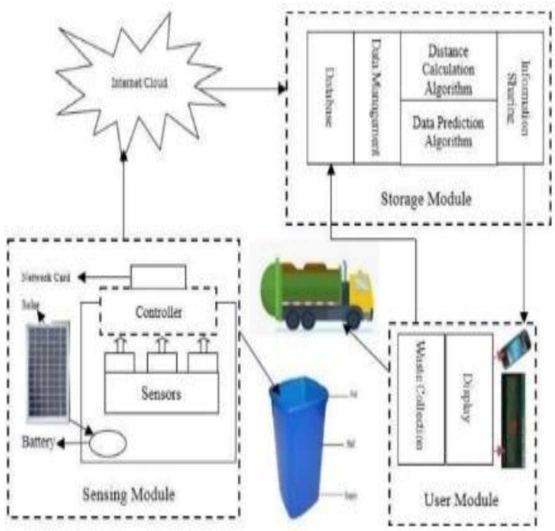
NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	A Smart city waste management technology allows crews to empty bins before they become overflowing with trash or recycling, and before infestation becomes an issue
NFR-2	Security	Innovations in waste reduction technologies allow us to better monitor, prevent, and manage our waste. This includes appliances that deal with waste sustainably, smartphone apps to track waste and help us develop eco-friendly habits, and sensors to accurately measure what we have and what we are tossing.
NFR-3	Reliability	Smart Bins help to create a cleaner, safer, more hygienic environment and enhanced operational efficiency while reducing management costs, resources, and road-side emissions.
NFR-4	Performance	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-5	Availability	The system should be available all the time when required.
NFR-6	Scalability	Using smart bin reduces the number of bins inside cities because we able to monitor the garbage 24/7 more efficient and scalability when we move smarter.

5. PROJECT DESIGN

5.1 Data Flow Diagrams:

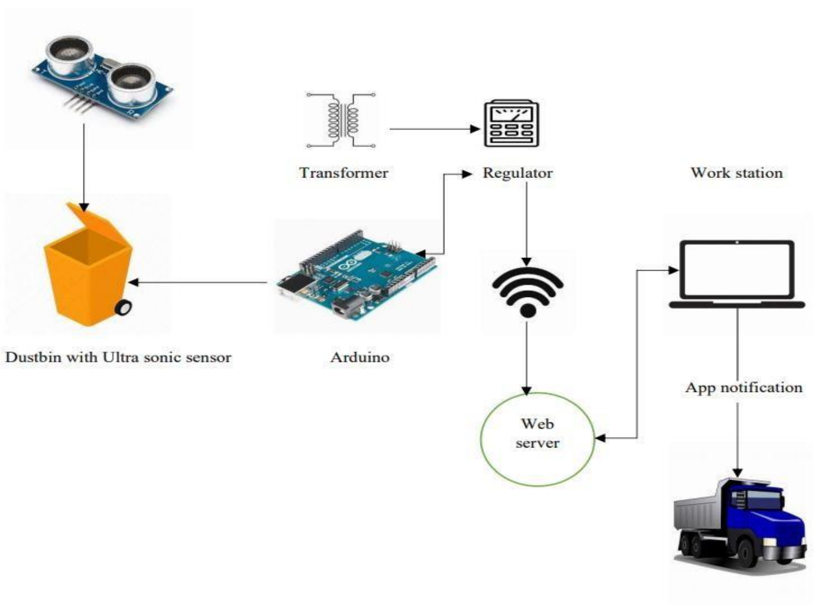
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



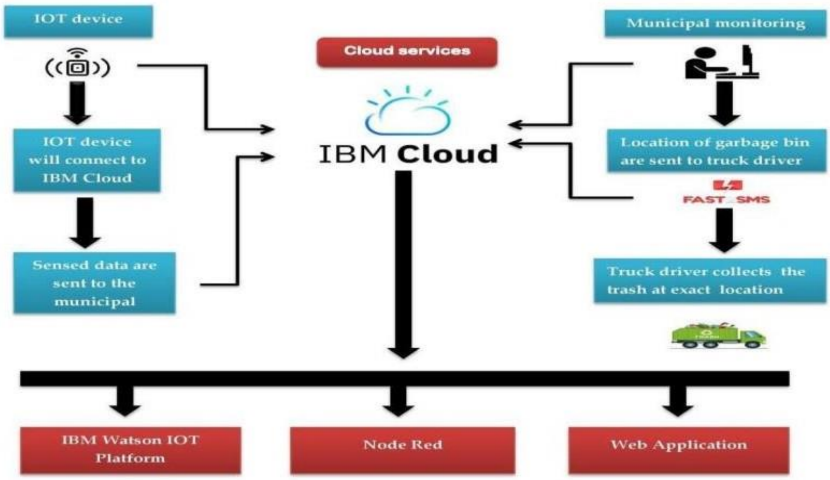


5.2Solution and Technical Architecture

Solution Architecture:



Technical Architecture:



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story/ Task	Acceptance criteria	Priority	Release
Admin	Login	USN-1	As an administrator,I assigned user names and passwords to each employee and managed them	I can control my online account and dashboard.	High	Sprint-1
Co-Admin	Login	USN-2	As a Co-Admin, I'll control the waste level monitor. If a garbage filling alert occurs, I will notify the trash truck of the location	I can handle the waste collection.	High	Sprint-1

			and rubbish ID.			
Truck Driver	Login	USN-3	As a Truck Driver, I'll follow Co Admin's instruction to reach the filled garbage.	I can take the shortest path to reach the waste filled route specified.	Low	Sprint-2
Local Garbage Collector	Login	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	I can collect the trach, pull it to the truck, and send it out.	Medium	Sprint-4
Municipality officer	Login	USN-5	As a Municipality officer, I'll make sure everything is proceeding as planned and without any problems.	All of these processes are under my control.	High	Sprint-5

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Login	USN-1	1.As a Administrator, I need to give user id and passcode for ever workers over there in municipality. 2.As a Co-Admin, I'll control the waste level by monitoring them vai real time web portal. Once the filling happens, I'll notify trash truck with location of bin with bin ID	20	High	S.Valarmathi R.P.Umadevi V.Sakthi D.Sathiya
Sprint-2	Dashboard	USN-2	As a Truck Driver, I'll follow Co-Admin's Instruction to reach the filling bin in short roots and save time	20	Low	S.Valarmathi R.P.Umadevi V.Sakthi D.Sathiya
Sprint-3	Dashboard	USN-3	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	S.Valarmathi R.P.Umadevi V.Sakthi D.Sathiya
Sprint-4	Dashboard	USN-4	As a Municipality officer, I'll make sure everything is proceeding as planned and without any problems	20	High	S.Valarmathi R.P.Umadevi V.Sakthi D.Sathiya

6.2 Sprint Delivery Schedule:

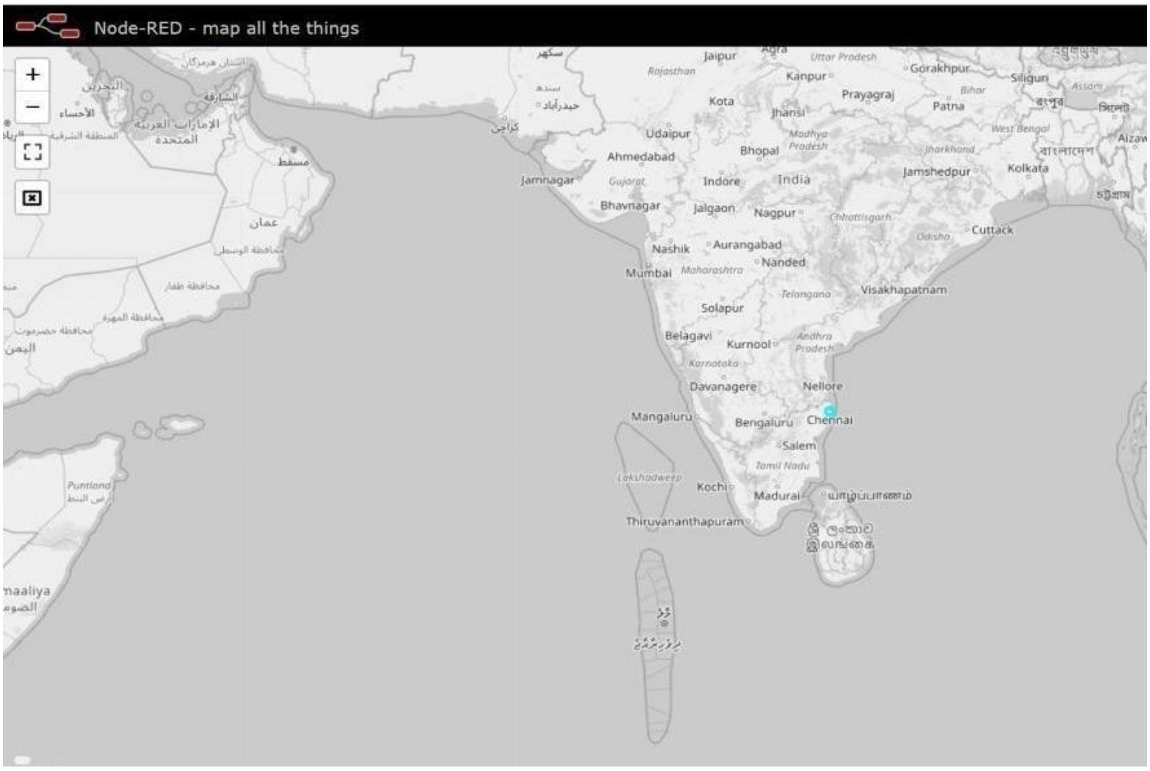
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	10 Days	22 Oct 2022	31 Oct 2022	35	31 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

7. CODING & SOLUTIONING

7.1 Feature 1:

The main and first feature of the smart waste management is to get the live location of anyone who access the website for putting out a request for garbage collection in their locality. The live location is obtained as a result of the below code.

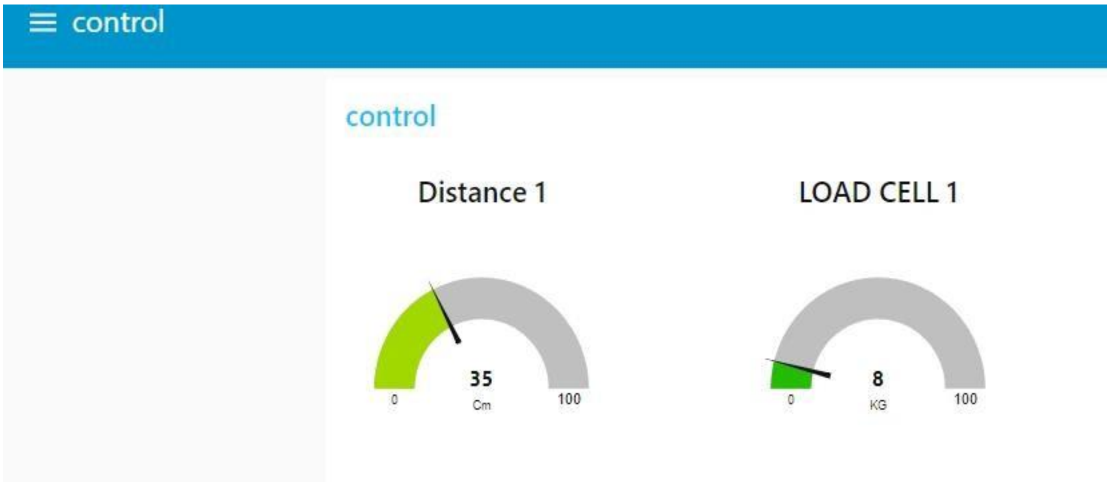
Location Tracker:

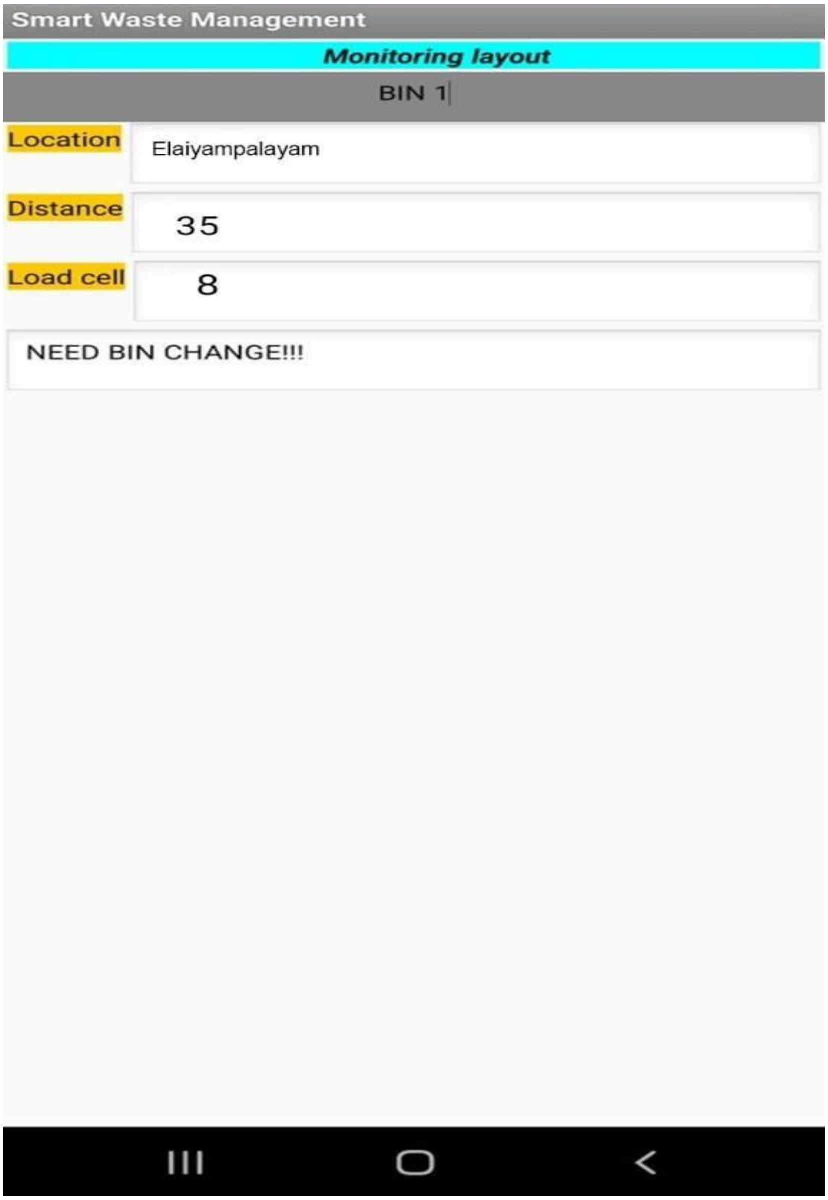


7.2 Feature 2:

In this part, the filled level of the bin is measured with the help of IBM IOT Watson platform devices, IBM Cloud interface and Node-RED is used for creating the dashboard nodes that helps us create a UI to display the distance, that is, the fill level of the bin. It also intimates the location of the bin with the fill level and alerts the collection authority if the fill level goes beyond a threshold value.

Live update on collected data:


control
control
Distance 1
LOAD CELL 1
35
0
Cm
100
8
0
KG
100


Smart Waste Management
Monitoring layout
BIN 1
Location
Elaiyampalayam
Distance
35
Load cell
8
NEED BIN CHANGE!!!

8. TESTING

8.1 Test cases:

Unit testing

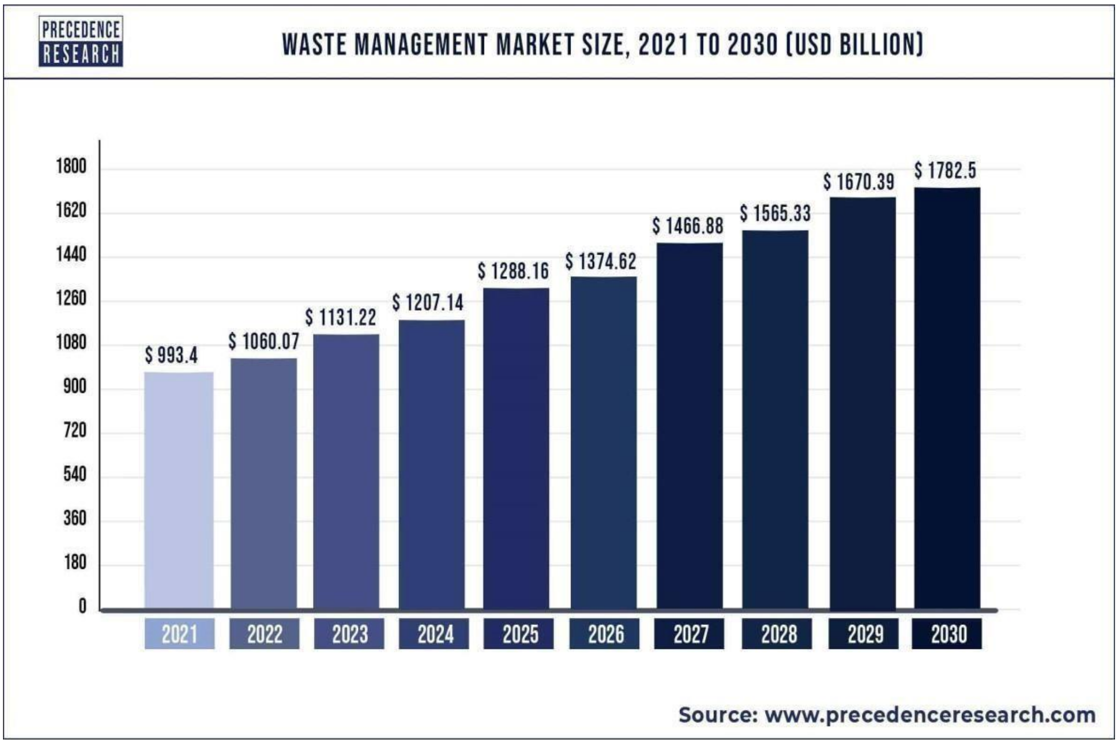
Test case no.	Sensor/Stage	Input	Expected output	Obtained output	Status
1.	Ultrasonic	Garbage level in bin i)Null ii)Full iii)Range in %	Correct level or distance	As expected,	Pass
2.	ESP – 32	Microcontroller to process the input data	To collect the data from sensor	As expected,	Pass
3.	Load cell	To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

8.2 User Acceptance Testing

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

9. RESULTS

9.1 Performance Metrics



10. ADVANTAGES & DISADVANTAGES

10.1 Advantages:

- Reduction in Collection Cost
- Waste Generation Analysis
- CO_2 Emission Reduction
- It keeps our surroundings clean and keeps free from bad odour.
- No Missed Pickups
- Reduces manpower requirement to handle the garbage collection
- Reduced Overflows
- Increases recycling rate of waste.

10.2 Disadvantages:

- It requires a well-structured hardware.
- The onetime cost of installation will be higher than the present technique.
- 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.
- Wireless technologies used in the system such as zigbee and wifi have shorter range and lower data speed
- It reduces man power requirements which results into increase in unemployments for unskilled people.
- The trainining must be provided to the people involved in the smart waste management system.

11. CONCLUSION

Improper disposal and improper maintenance of domestic waste create issues in public health and environment pollution thus this paper attempts to provide practical solution towards managing the waste collaborating it with the use of IOT. 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. The price might be high.

12. FUTURE SCOPE

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

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

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

13. APPENDIX

13.1 Source Code:

<u>Code to evaluate the level of the garbage in bin and intimate the collection authority with the location of the bin:</u>

<u>bin1.py:</u>

```
import requests
import json
import ibmiotf.application
import ibmiotf.device
import time
import random
import sys

# watson device details

organization = "46x7xk"
devicType = "Garbage"
deviceId = "Garbage1"
authMethod= "token"
authToken= "123456789"

#generate random values for randomo variables (temperature&humidity)



def myCommandCallback(cmd):
    global a
    print("command recieved:%s" %cmd.data['command'])
    control=cmd.data['command']
    print(control)

try:
        deviceOptions={"org": organization, "type": devicType,"id": deviceId,"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()


deviceCli.connect()

while True:

    ultrasonic= random.randint(10,70)
    loadcell= random.randint(5,15)
    data= {'dist':ultrasonic,'load':loadcell}


    if loadcell < 13 and loadcell > 15:
          load = "90 %"

    elif loadcell < 8 and loadcell > 12:
          load = "60 %"

    elif loadcell < 4 and loadcell > 7:
          load = "40 %"
    else:
          load = "0 %"

    if ultrasonic < 10:
          dist = ' 90 %'


    elif ultrasonic < 20 and ultrasonic >11:
          dist = '60%'

    elif ultrasonic < 60 and ultrasonic > 41:
          dist =  '40 %'
    elif ultrasonic < 80 and ultrasonic > 61:
          dist =  '20 %'




    if load == "90 %" or ultrasonic == "90 %":
          warn = 'alert :' ' Dumpster poundage getting high, Time to collect :)'

    elif load == "60 %" or ultrasonic == "60 %":
```

```
        warn = 'alert :' 'dumpster is above 60%'
    else :
        warn = 'alert :' 'No need to collect right now '
    def myOnPublishCallback(lat=10.678991,long=78.177731):
        print("")
        print("published distance = %s " %ultrasonic,"loadcell:%s " %loadcell,"lon =
%s " %long,"lat = %s" %lat)
        print(load)
        print(dist)
        print(warn)

    time.sleep(5)


    success=deviceCli.publishEvent      ("IoTSensor","json",warn,qos=0,on_publish=
myOnPublishCallback)

    success=deviceCli.publishEvent       ("IoTSensor","json",data,qos=0,on_publish=
myOnPublishCallback)



    if not success:
        print("not connected to ibmiot")
    time.sleep(5)




    deviceCli.commandCallback=myCommandCallback
#disconnect the device
deviceCli.disconnect()
```

bin2.py:

```
import requests
import json
import ibmiotf.application
import ibmiotf.device
import time
import random
import sys
```

```
# watson device details

organization = "46x7xk"
devicType = "Garbage"
deviceId = "Garbage2"
authMethod= "token"
authToken= "123456789"

#generate random values for randomo variables (temperature&humidity)



def myCommandCallback(cmd):
    global a
    print("command recieved:%s" %cmd.data['command'])
    control=cmd.data['command']
    print(control)

try:
        deviceOptions={"org": organization, "type": devicType,"id": deviceId,"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()


deviceCli.connect()

while True:

    ultrasonic= random.randint(10,70)
    loadcell= random.randint(5,15)
    data= {'dist':ultrasonic,'load':loadcell}


    if loadcell < 13 and loadcell > 15:
          load = "90 %"

    elif loadcell < 8 and loadcell > 12:
          load = "60 %"

    elif loadcell < 4 and loadcell > 7:
```

```
        load = "40 %"
    else:
        load = "0 %"

    if ultrasonic < 10:
        dist = ' 90 %'


    elif ultrasonic < 20 and ultrasonic >11:
        dist = '60%'

    elif ultrasonic < 60 and ultrasonic > 41:
        dist =  '40 %'
    elif ultrasonic < 80 and ultrasonic > 61:
        dist =  '20 %'




    if load == "90 %" or ultrasonic == "90 %":
        warn = 'alert :' ' Dumpster poundage getting high, Time to collect :)'

    elif load == "60 %" or ultrasonic == "60 %":

        warn = 'alert :' 'dumpster is above 60%'
    else :
        warn = 'alert :' 'No need to collect right now '
    def myOnPublishCallback(lat=10.678991,long=78.177731):
       print("")
       print("published distance = %s " %ultrasonic,"loadcell:%s " %loadcell,"lon = 
%s " %long,"lat = %s" %lat)
       print(load)
       print(dist)
       print(warn)

    time.sleep(5)


    success=deviceCli.publishEvent      ("IoTSensor","json",warn,qos=0,on_publish=
myOnPublishCallback)

    success=deviceCli.publishEvent       ("IoTSensor","json",data,qos=0,on_publish=
myOnPublishCallback)
```

```
    if not success:
        print("not connected to ibmiot")
    time.sleep(5)




    deviceCli.commandCallback=myCommandCallback
#disconnect the device
deviceCli.disconnect()
```

bin3.py:

```
import requests
import json
import ibmiotf.application
import ibmiotf.device
import time
import random
import sys

# watson device details

organization = "46x7xk"
devicType = "Garbage"
deviceId = "Garbage3"
authMethod= "token"
authToken= "123456789"

#generate random values for randomo variables (temperature&humidity)



def myCommandCallback(cmd):
    global a
    print("command recieved:%s" %cmd.data['command'])
    control=cmd.data['command']
    print(control)

try:
```

```
        deviceOptions={"org": organization, "type": devicType,"id": deviceId,"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()


deviceCli.connect()

while True:

    ultrasonic= random.randint(10,70)
    loadcell= random.randint(5,15)
    data= {'dist':ultrasonic,'load':loadcell}


    if loadcell < 13 and loadcell > 15:
          load = "90 %"

    elif loadcell < 8 and loadcell > 12:
          load = "60 %"

    elif loadcell < 4 and loadcell > 7:
          load = "40 %"
    else:
          load = "0 %"

    if ultrasonic < 10:
          dist = ' 90 %'


    elif ultrasonic < 20 and ultrasonic >11:
          dist = '60%'

    elif ultrasonic < 60 and ultrasonic > 41:
          dist =  '40 %'
    elif ultrasonic < 80 and ultrasonic > 61:
          dist =  '20 %'
```

```
    if load == "90 %" or ultrasonic == "90 %":
        warn = 'alert :' ' Dumpster poundage getting high, Time to collect :)'

    elif load == "60 %" or ultrasonic == "60 %":

        warn = 'alert :' 'dumpster is above 60%'
    else :
        warn = 'alert :' 'No need to collect right now '
    def myOnPublishCallback(lat=10.678991,long=78.177731):
        print("")
        print("published distance = %s " %ultrasonic,"loadcell:%s " %loadcell,"lon =
%s " %long,"lat = %s" %lat)
        print(load)
        print(dist)
        print(warn)

    time.sleep(5)


    success=deviceCli.publishEvent      ("IoTSensor","json",warn,qos=0,on_publish=
myOnPublishCallback)

    success=deviceCli.publishEvent      ("IoTSensor","json",data,qos=0,on_publish=
myOnPublishCallback)



    if not success:
        print("not connected to ibmiot")
    time.sleep(5)




    deviceCli.commandCallback=myCommandCallback
#disconnect the device
deviceCli.disconnect()
```

bin4.py:

```
import requests
import json
import ibmiotf.application
```

```
import ibmiotf.device
import time
import random
import sys

# watson device details

organization = "46x7xk"
devicType = "Garbage"
deviceId = "Garbage4"
authMethod= "token"
authToken= "123456789"

#generate random values for randomo variables (temperature&humidity)



def myCommandCallback(cmd):
    global a
    print("command recieved:%s" %cmd.data['command'])
    control=cmd.data['command']
    print(control)

try:
        deviceOptions={"org": organization, "type": devicType,"id": deviceId,"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()


deviceCli.connect()

while True:

    ultrasonic= random.randint(10,70)
    loadcell= random.randint(5,15)
    data= {'dist':ultrasonic,'load':loadcell}


    if loadcell < 13 and loadcell > 15:
          load = "90 %"
```

```
    elif loadcell < 8 and loadcell > 12:
          load = "60 %"

    elif loadcell < 4 and loadcell > 7:
          load = "40 %"
    else:
          load = "0 %"

    if ultrasonic < 10:
          dist = ' 90 %'


    elif ultrasonic < 20 and ultrasonic >11:
          dist = '60%'

    elif ultrasonic < 60 and ultrasonic > 41:
          dist =  '40 %'
    elif ultrasonic < 80 and ultrasonic > 61:
          dist =  '20 %'




    if load == "90 %" or ultrasonic == "90 %":
          warn = 'alert :' ' Dumpster poundage getting high, Time to collect :)'

    elif load == "60 %" or ultrasonic == "60 %":

          warn = 'alert :' 'dumpster is above 60%'
    else :
          warn = 'alert :' 'No need to collect right now '
    def myOnPublishCallback(lat=10.678991,long=78.177731):
        print("")
        print("published distance = %s " %ultrasonic,"loadcell:%s " %loadcell,"lon = 
%s " %long,"lat = %s" %lat)
        print(load)
        print(dist)
        print(warn)

    time.sleep(5)
```

```
    success=deviceCli.publishEvent      ("IoTSensor","json",warn,qos=0,on_publish=
myOnPublishCallback)

    success=deviceCli.publishEvent       ("IoTSensor","json",data,qos=0,on_publish=
myOnPublishCallback)



    if not success:
        print("not connected to ibmiot")
    time.sleep(5)




    deviceCli.commandCallback=myCommandCallback
#disconnect the device
deviceCli.disconnect()
```

bin5.py:

```
import requests
import json
import ibmiotf.application
import ibmiotf.device
import time
import random
import sys

# watson device details

organization = "46x7xk"
devicType = "Garbage"
deviceId = "Garbage5"
authMethod= "token"
authToken= "123456789"

#generate random values for randomo variables (temperature&humidity)



def myCommandCallback(cmd):
    global a
```

```
    print("command recieved:%s" %cmd.data['command'])
    control=cmd.data['command']
    print(control)

try:
        deviceOptions={"org":  organization,  "type":  devicType,"id":  deviceId,"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()


deviceCli.connect()

while True:

    ultrasonic= random.randint(10,70)
    loadcell= random.randint(5,15)
    data= {'dist':ultrasonic,'load':loadcell}


    if loadcell < 13 and loadcell > 15:
          load = "90 %"

    elif loadcell < 8 and loadcell > 12:
          load = "60 %"

    elif loadcell < 4 and loadcell > 7:
          load = "40 %"
    else:
          load = "0 %"

    if ultrasonic < 10:
          dist = ' 90 %'


    elif ultrasonic < 20 and ultrasonic >11:
          dist = '60%'

    elif ultrasonic < 60 and ultrasonic > 41:
          dist =  '40 %'
    elif ultrasonic < 80 and ultrasonic > 61:
```

```
        dist =  '20 %'



    if load == "90 %" or ultrasonic == "90 %":
        warn = 'alert :' ' Dumpster poundage getting high, Time to collect :)'

    elif load == "60 %" or ultrasonic == "60 %":

        warn = 'alert :' 'dumpster is above 60%'
    else :
        warn = 'alert :' 'No need to collect right now '
    def myOnPublishCallback(lat=10.678991,long=78.177731):
        print("")
        print("published distance = %s " %ultrasonic,"loadcell:%s " %loadcell,"lon = 
%s " %long,"lat = %s" %lat)
        print(load)
        print(dist)
        print(warn)

    time.sleep(5)


    success=deviceCli.publishEvent      ("IoTSensor","json",warn,qos=0,on_publish=
myOnPublishCallback)

    success=deviceCli.publishEvent       ("IoTSensor","json",data,qos=0,on_publish=
myOnPublishCallback)



    if not success:
        print("not connected to ibmiot")
    time.sleep(5)




    deviceCli.commandCallback=myCommandCallback
#disconnect the device
deviceCli.disconnect()
```

13.2 GitHub & Project Demo Links:

https://github.com/IBM-EPBL/IBM-Project-20995-1659768984

https://youtu.be/eaVOmeTPoeg