SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITAN CITIES

TEAM ID:PNT2022TMID04740

SUBMITTED BY

SILVIYA X	-737819ECR177
SRINITHI V	-737819ECR187
VIMAL PRASAD M	-737819ECR216
SELCIYADEVI S	-737819ECR170

TABLE OF CONTENT

CHAPTER NO.	TOPIC
1	**INTRODUCTION**
	1.1 PROJECT OVERVIEW
	1.2 PURPOSE
2	**LITERATURE SURVEY**
	2.1 EXISTING PROBLEM
	2.2 REFERENCES
	2.3 PROBLEM STATEMENT DEFINITION
3	**IDEATION AND PROPOSED SOLUTION**
	3.1 EMPATHY MAP CANVAS
	3.2 IDEATION AND BRAINSTORMING
	3.3 PROPOSED SOLUTION
	3.4 PROBLEM SOLUTION FIT
4	**REQUIREMENT ANALYSIS**
	4.1 FUNCTIONAL REQUIREMENTS
	4.2 NON- FUNCTIONAL REQUIREMENTS
5	**PROJECT DESIGN**
	5.1 DATA FLOW DIAGRAMS
	5.2 SOLUTION AND TECHNICAL ARCHITECTURE
	5.3 USER STORIES
6	**PROJECT PLANNING AND SCHEDULING**
	6.1 SPRINT PLANNING AND ESTIMATION
	6.2 SPRINT DELIVERY SCHEDULE
7	**CODING AND SOLUTIONING**
	7.1 FEATURE 1
	7.2 FEATURE 2

7.3 DATABASE SCHEMA

8 TESTING

8.1 TEST CASES

9 RESULTS

9.1 PERFORMANCE METRICS

10 ADVANTAGES AND DISADVANTAGES

11 CONCLUSION

12 FUTURE SCOPE

13 APPENDIX

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

This project aims to design and implement a combination of IoT and Application Development based Waste Management Systems. The combination of IoT and Application Development has plenty of applications such as home security systems, payment technologies, intruder recognition systems, etc. This research utilizes the application for Waste Management. The kit consists of hardware and software parts. The hardware part comprises a sensor unit, which detects the volume of waste present in the bin, a weight-detecting garbage system, a GPS locator, and a GSM module to communicate with a mobile device. The software part uses Python codes and C codes.

1.2 PURPOSE:

The purpose of this project is a small step to Reduce Air, Water, and Soil Pollution. The world faces major environmental challenges associated with waste generation and inadequate waste collection, transport, treatment, and disposal. It is a matter of health safety. Tuberculosis, pneumonia, diarrhoea, tetanus, whooping cough, etc. are other common diseases spread due to improper waste management. The toxic wastes can lead to different kinds of pollution - air, water, and soil. Our current systems cannot cope with the volumes of waste generated by an increasingly urban population and this has a huge impact on the environment and public health. It reduces manual labour, increases sustainable development, and reduces common health issues related to improper waste management techniques.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

Waste management plays a crucial role these days. As environmental concerns grow, wastes are to be properly managed and recycled. Improper management will lead to air pollution, and soil erosion may even affect human health. Lisa Safer, et al. enhance the point about the health impacts of incineration, landfill, composting, land spreading sewage sludge, and sewage discharges. A step to reduce the risks is the proposed work of waste management using IoT. Gopal Krishna Shyam, et al. submitted a work that utilizes sensors and uses an IoT algorithm that can read, collect, and transmit a huge volume of data over the Internet. These data, when put into a Spatio-temporal context and processed by intelligent and optimized algorithms, can be dynamically handled by waste collection processes. The published work by Tran Anh Khoa et al put forth a low-cost IoT architecture that efficiently achieves waste management by predicting the probability of the waste level in trash bins, using machine learning and graph theory, and determining the shortest path of waste collection. It also examines the data transfer on the LoRa module and demonstrates the advantages of then system, which is implemented through a simple circuit designed with low cost, ease of use, and repeatability. "Challenges and Opportunities of Waste Management in IoT-Enabled Smart Cities: A Survey" by Theodoros Anagnostopoulos, et al. gives detailed information on various aspects of IoT in waste management. With the above references, this project proposes a Smart Waste Management System For Metropolitan Cities that detects the level of Garbage in bins, and the weight of the garbage in the bin and alerts the authorized person to empty the bin whenever the bins are full. With further advancements, the Garbage level of the bins can be monitored through a Web App through which we can view the location of every bin by sending GPS location from the device.

2.2 REFERENCES

[1].Parkash Tambare, Prabu Venkatachalam, "IoT Based Waste Management for Smart City", International Journal of Innovative Research in Computer and Communication Engineering, 2016.

[2].K. Suresh, S. Bhuvanesh and B. Krishna Devan, "Arduino Micro-controller Based Smart Dustbins for Smart Cities ",International Journal of Recent Trends in Engineering & Research, 2019.

[3].Dr. Raveesh Agarwal, Mona Chaudhary and Jayveer Singh,"Waste Management Initiatives in India For Human Wellbeing " ,International Journal of Scientific & Technology Research, 2015.

[4].Na Jong Shen, Azham Hussain and Yuhanis Yusof ,"Design and Development of Smart Waste Management System: A Mobile App for Connecting and Monitoring Dustbin Using IoT", 1st International Conference on Innovations in Information and Communication Technology (ICIICT), 2020.

[5]. Rahul Kumar Borah, Sahana Shetty, Rahul Patidar, Anisha Raniwala and Kratee Jain , "IoT based smart garbage collection system ", International Journal of Civil Engineering and Technology, 2018.

[6].S.Vishnu, S.R.Jino Ramson, Samson Senith, Adnan M. Abu-Mahfouz, S.Srinivasan, Theodoros Anagnostopoulos, Xiaozhe Fan and A. Alfred Kirubaraj ,"IoT-Enabled Solid Waste Management in Smart Cities", IEEE Explore, 2021.

[7].Aderemi A. Atayero, Segun I. Popoola, Rotimi Williams, Joke A. Badejo and Sanjay Misra ,"Smart City Waste Management System using IoT and Cloud Computing", IEEE Students Conference on Engineering and Systems (SCES), 2021.

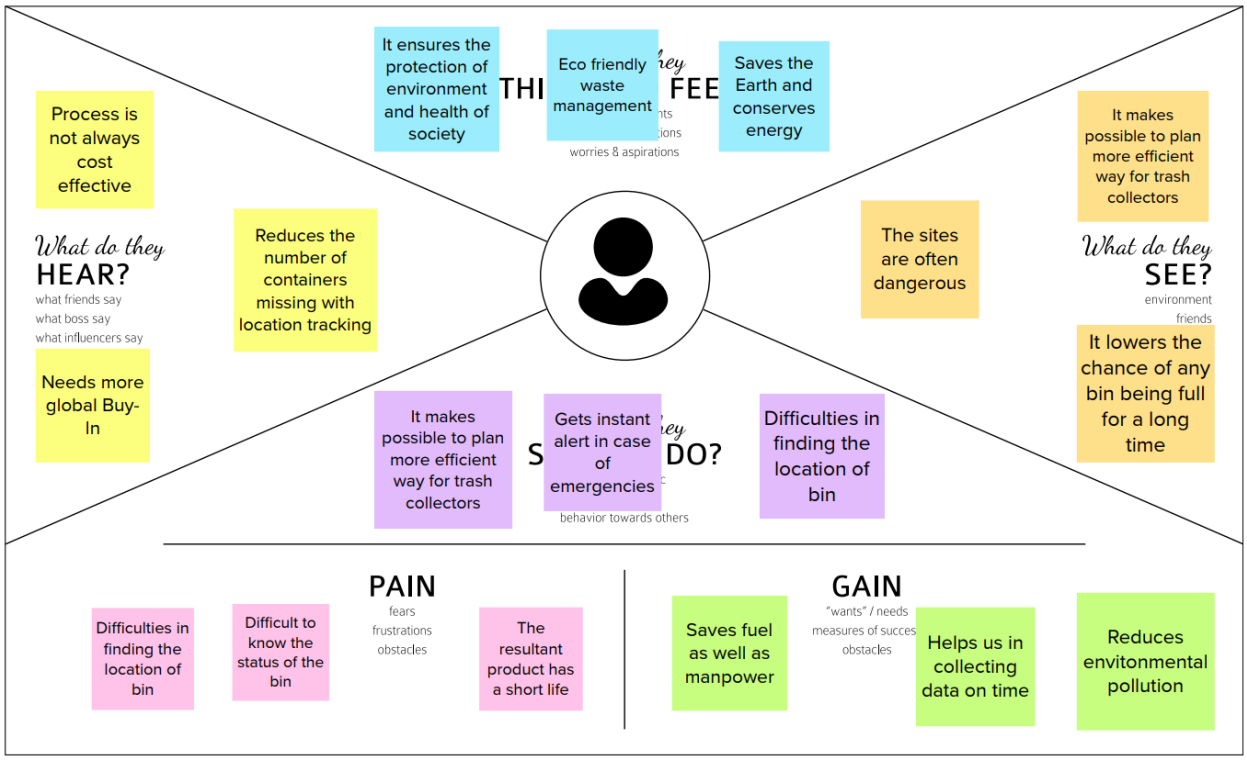
2.3 PROBLEM STATEMENT DEFINITION

Waste management in metropolitan cities faces numerous challenges. The main problem faced by the metropolitan cities are detecting the garbage level weather it filled or not and also we need to measure the weight of the garbage bin. Then alerts the authorized person to empty the bin whenever the bins are full. We need to develop a web application to monitor the status of the bins remotely at anywhere. The application should provide the location of the every bin connected in the application with the help of global positioning system (GPS). The indication of the bins and the location of the every bin should be provided by web applications simultaneously.

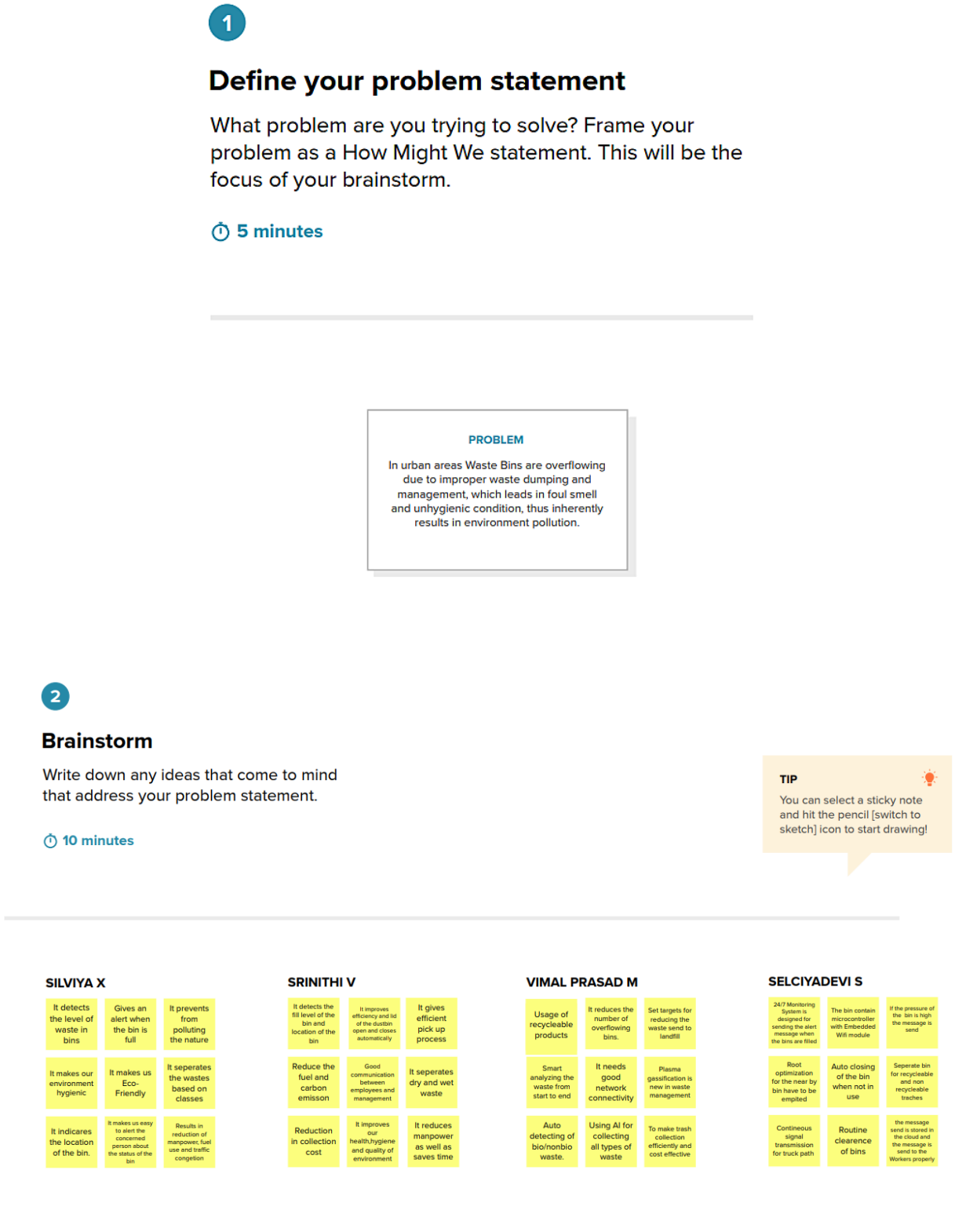
CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION AND BRAINSTORMING:



3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

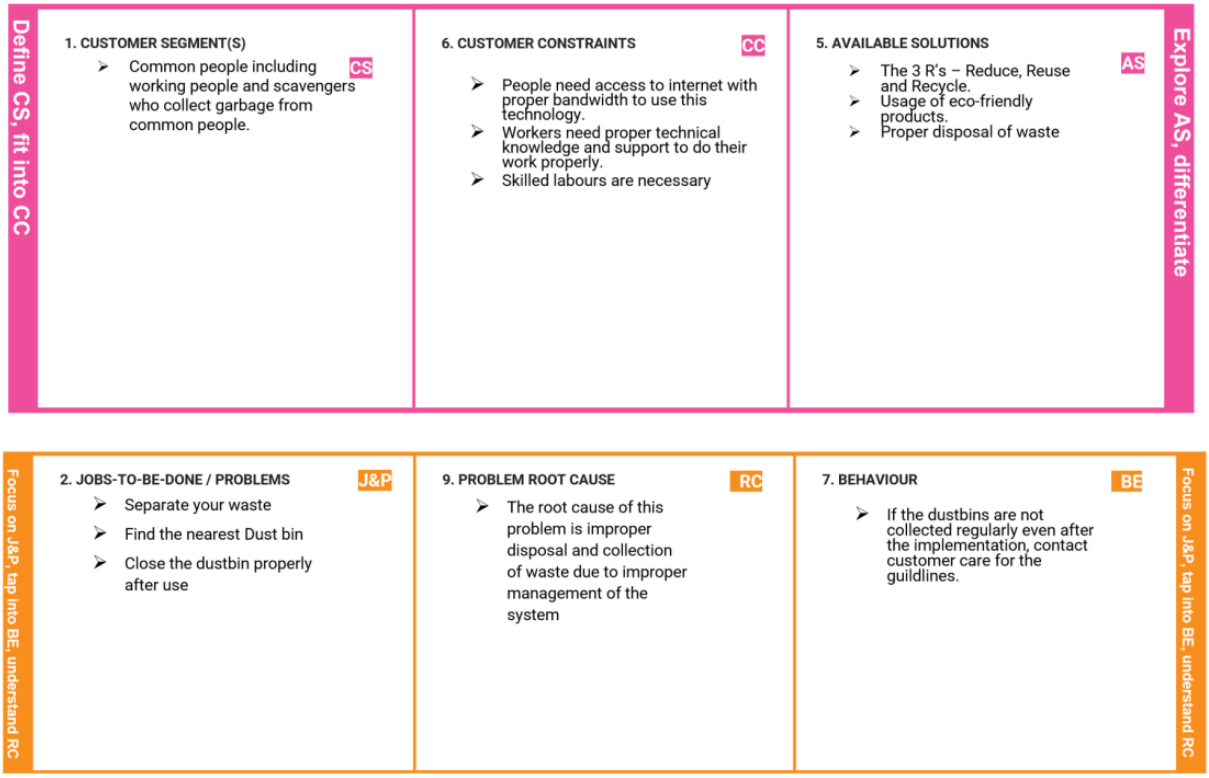
20 minutes

3.3 PROPOSED SOLUTION

S.NO.	PARAMETER	DESCRIPTION
1.	Problem Statement (Problem to be solved)	➢ Trashes from bin is collected at fixed interval of time whether the is empty or full ➢ If trash is overfilled, it may lead to several health problems to human ➢ Difficult to identify location of trash bin ➢ Excessive use of resources including manpower, vehicle and fuel
2.	Idea / Solution description	➢ Ultrasonic sensor is interfaced with Arduino to identify the level of the dustbin ➢ GPS module is used to identify the location of the bin ➢ With the help of moisture sensor, it separates dry and wet waste ➢ DC motor is used for segregating dry and wet waste ➢ The status of the bin and location is sent to drivers with the help of GSM module
3.	Novelty / Uniqueness	➢ It communicates information about fill level of bin and ensure collection only when the bin is full ➢ It sends location of the bin to garbage truck. They spend less time on road so that traffic congestion can be reduced ➢ In this project,24x7 monitoring system is used for monitoring dumpsters ➢ Compare to other system ,it is less cost

4.	Social Impact / Customer Satisfaction	➢ Reduces environmental pollution – if trash in bin is full, workers collect trash at correct time ➢ When people come nearer to bin, it opens automatically ➢ Truck drivers can easily identify the location of the bin ➢ Administrator can know the status of bin in all cities
5.	Business Model (Revenue Model)	➢ This system is designed by government to people. ➢ Government provides this system to ensure public health and environmental safety
6.	Scalability of the Solution	➢ Efficient collection management ➢ Increase efficiency ➢ Keep the environment clean and fresh ➢ Ensure physical safe

3.4 PROBLEM SOLUTION FIT



CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Details of the Bin	The bin details are displayed in the Dashboard – capacity, waste type, last measurement, GPS location and collection schedule or pick recognition.
FR-2	Bin Monitoring	We can eliminate the risk of bin overflow and stop collecting half-empty bins by using real-time data and predictions.
FR-3	Cost of bins	It is beneficial to identify bins that increase your collection costs. In terms of collection costs, the tool assigns a rating to each bin.
FR-4	Adjusting level of Garbage	Ensure the best possible bin distribution. Determine whether the bin distribution is dense or clustered. Ensure that all types of trash are represented within a stand. You can adjust bin capacity or location as needed based on historical data.
FR-5	Eliminate insufficient garbage	Remove the collection of half-empty bins. Picks are recognised by the sensors. We can show you how full the bins you collect are by using real-time data on fill-levels and pick recognition. The report indicates how full the bin was when it was picked. Any picks that are less than 80% full are immediately visible.
FR-6	Planning for waste collection	The programme plans waste collecting routes semi-automatically. You are prepared to respond and plan waste collection based on current bin fill levels and projections of approaching full capacity. To find discrepancies, compare planned and accomplished routes.

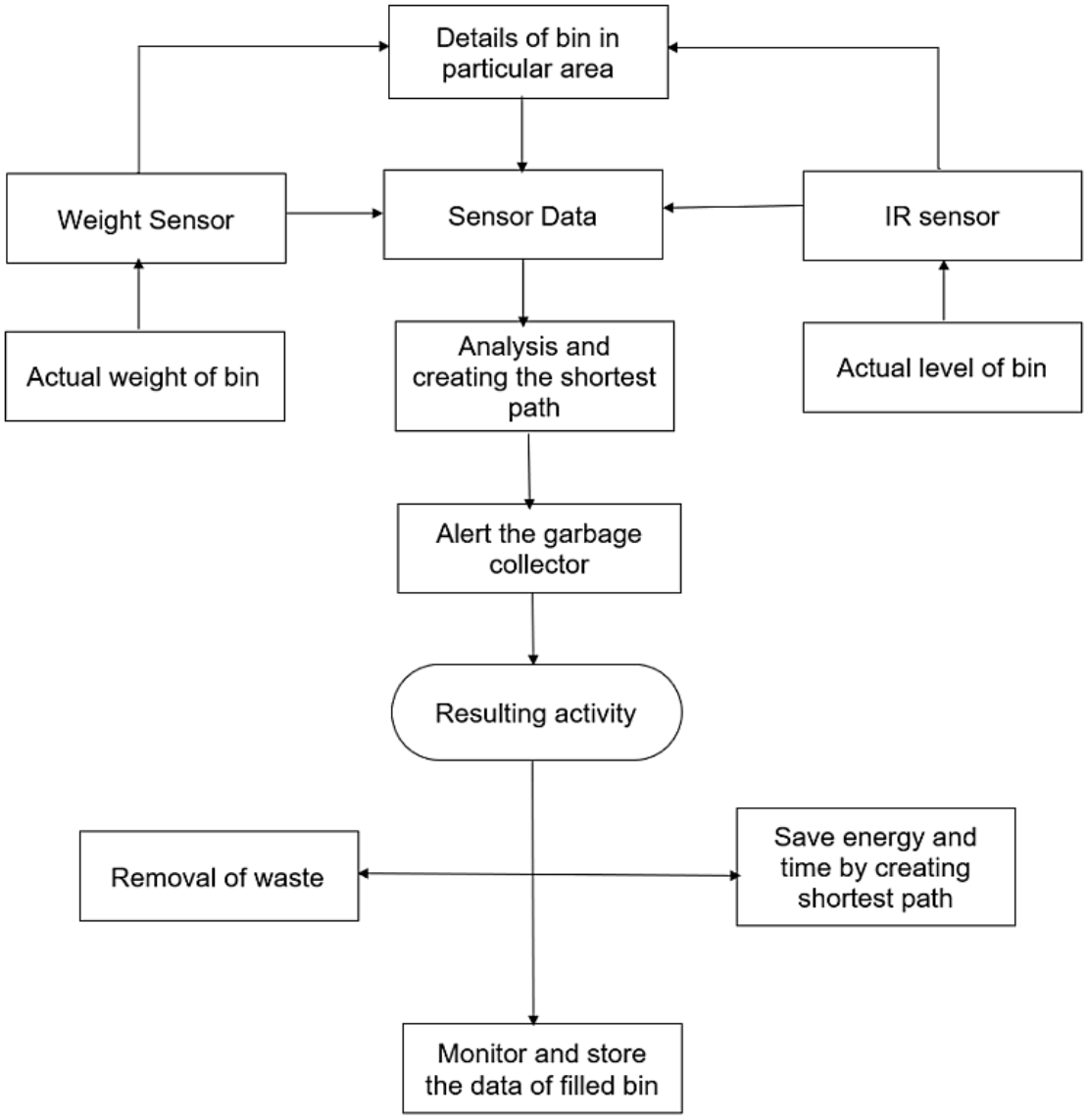
4.2 NON-FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	This Smart Waste Management system enables bins to be evacuated before they become overloaded with garbage or recycling, along with before contamination becomes an issue.
NFR-2	**Security**	This method is robust to threats since the data processed is simply about trash amount and bin placement. Trash reduction technology advancements enable us to better monitor, avoid, and manage our waste.
NFR-3	**Reliability**	Smart Bins contribute to a cleaner, safer, and more hospitable environment, as well as increased operating efficiency while lowering management costs, resources, and road-side emissions.
NFR-4	**Performance**	Rather of travelling the same collection routes and servicing empty bins, trash collectors will use their time more efficiently servicing bins that require service.
NFR-5	**Availability**	When needed, the system should be accessible at all times. To receive all data and analyze all complaints and bin data, the admin entire network should have a quickest possible connection.
NFR-6	**Scalability**	Using smart bins helps reduce the number of bins within cities although we can monitor waste 24 hours a day, seven days a week.

CHAPTER 5

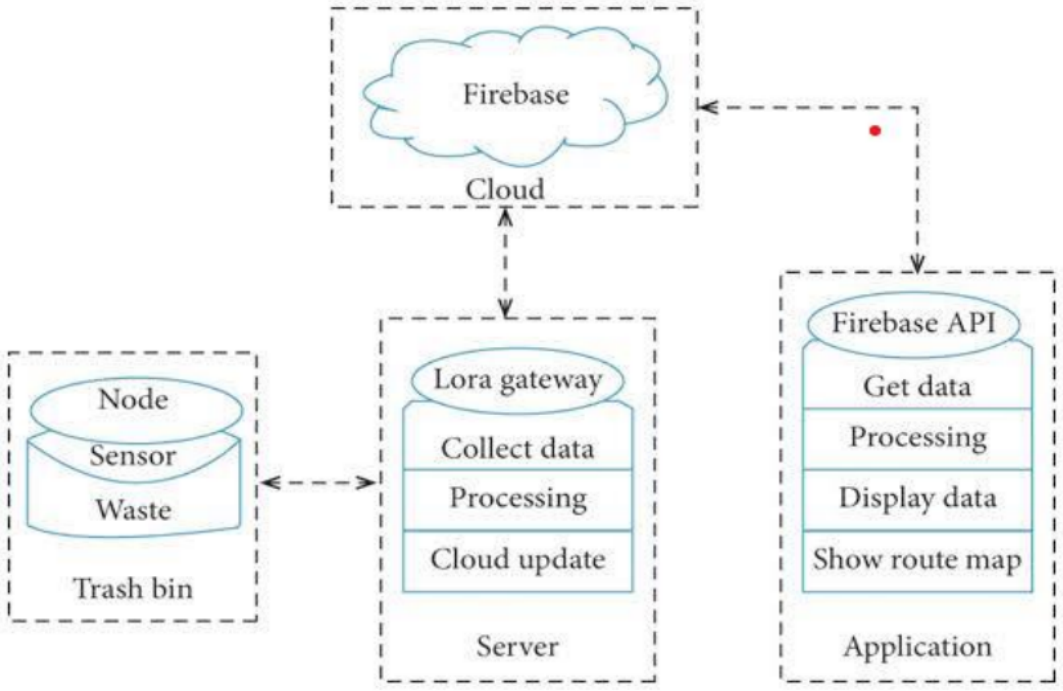
PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



- Weight sensor is used to identify weight of the bin
- IR sensor is used to indicate level of the bin
- The resulting data from sensor is analysed
- Shortest path for the bin is identified
- Alert message is sent to garbage collector to collect the garbage
- Finally waste is removed from bin before overflowing
- It saves time and energy
- It ensures safety environment

5.2 SOLUTION ARCHITECTURE



- Trash bin alert is sent to server
- Server is connected with cloud

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application byentering my email, password, and confirmingmy password.	I can access my account /dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation emailonce I have registered for the application	I can receive confirmationemail & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-4	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-5	As a user ,I can have user information ,data display, bin level		High	Sprint-2
Customer (Webuser)	Login	USN-1	As a user, I can register for the application by entering my email and password	I can access my accountor dashboard	High	Sprint-1
		USN-2	As a user ,I can confirm the captcha as I amnot a robot.	Security prioritize	High	Sprint-1
Customer Care Executive	Resolving issues	USN-1	As a customer care executive ,I can resolve all the issues regarding to the user application.	I can access user login details	High	Sprint-1
Administrator	Managing and controlling	USN-1	As an administrator ,I can manage the application and control the issues in higher level.	I can access each andevery details in the application of the particular user .	High	Sprint-1

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

TITLE	DESCRIPTION	DATE
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains. Prepare list of problem Statements that are to be solved by this project.	10 SEPTEMBER 2022
Literature Survey & Information-gathering	Literature survey on the project based on previous researchpapers published.	21 SEPTEMBER 2022
Ideation	List the ideas by organizing a brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	26 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution document, which includes novelty, feasibility of idea, revenue model, social impact, scalability of solution.	10 OCTOBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	14 OCTOBER 2022

Solution Architecture	Prepare solution architecture document.	20 OCTOBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application.	1 NOVEMBER 2022
Functional Requirements	Prepare the Functional and Non-Functional Requirements for the project	11 NOVEMBER 2022
Data Flow Diagrams and User Stories	Draw the data flow diagrams and write the user stories for it .	11 NOVEMBER 2022
Technology Architecture	Prepare the technology architecture diagram with technologies and components used in it .	11 NOVEMBER 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list that has been done so far in the project .	14 NOVEMBER 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

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 passcode for ever workers over there in municipality	1	High	Silviya.X Srinithi.V
Sprint-1	Login	USN-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	1	High	Vimal Prasad.M Selciyadevi.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	2	Low	Silviya.X Srinithi.V Vimal Prasad.M Selciyadevi.S
Sprint-3	Dashboard	USN-4	As a Local Garbage Collector, I'll gather all the waste from the garbage, load it onto a garbage truck, and deliver it to Landfills	2	Medium	Silviya.X Srinithi.V Vimal Prasad.M Selciyadevi.S
Sprint-4	Dashboard	USN-5	As a Municipality officer, I'll make sure everything is proceeding as planned and without any problems	2	High	Silviya.X Srinithi.V Vimal Prasad.M Selciyadevi.S

PROJECT TRACKER , VELOCITY AND BURNDOWN CHART:

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

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)

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

CHAPTER 7

CODING AND SOLUTION

7.1 FEATURE 1

- IOT DEVICE
- WOKWI SOFTWARE
- IOT WATSON PLATFORM
- NODE RED
- WEB UI
- CLOUDANT DB

Code:

```
import requests
import json
import ibmiotf.application
import ibmiotf.device
import time
import random
import sys
# Watson device details
organization = "j5bxb7"
devicType = "IOT123edevicetype"
deviceId = "IOTece4"
authMethod= "token"
authToken= "e2)-17xkqIFMvm3@ll"
#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,"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 data point "temp" with value integer value into the cloud as a type of event
for every 10 seconds
deviceCli.connect()
while True:
    distance= random.randint(10,70)
    loadcell= random.randint(5,15)
    data= {'dist':distance,'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 distance < 15:
        dist = 'Risk warning:' 'Dumpster poundage getting high, Time to collect :) 90 %'
```

```
    elif distance < 40 and distance >16:
        dist = 'Risk warning:' 'dumpster is above 60%'
    elif distance < 60 and distance > 41:
        dist = 'Risk warning:' '40 %'
    else:
        dist = 'Risk warning:' '17 %'
    if load == "90 %" or distance == "90 %":
        warn = 'alert :' ' Dumpster poundage getting high, Time to collect :)'
    elif load == "60 %" or distance == "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("Gandigramam, Karur")
    print("published distance = %s " %distance,"loadcell:%s " %loadcell,"lon = %s "%long,"lat = %s"
%lat)
    print(load)
    print(dist)
    print(warn)
    time.sleep(10)
    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(30)
        deviceCli.commandCallback=myCommandCallback
#disconnect the device
deviceCli.disconnect
```

7.2 FEATURE 2

- REGISTRATION
- LOGIN
- VERIFICATION
- SELECT THE CITY
- DISPLAY THE STATUS OF BIN
- ADD QUERY

Code:

Ibm.html:

```
<!DOCTYPE html>
<html>
<head>
<title>Registration system PHP and MySQL</title>
 <link rel="stylesheet" href="style.css">
</head>
<body>
<div class="header">
<h2>Register</h2>
</div>
<form method="post" action="register.php">
<div class="input-group">
<label>Username</label>
<input type="text" name="username" value="">
```

```
</div>
<div class="input-group">
<label>Email</label>
<input type="email" name="email" value="">
</div>
<div class="input-group">
<label>Password</label>
<input type="password" name="password_1">
</div>
<div class="input-group">
<label>Confirm password</label>
<input type="password" name="password_2">
</div>
<div class="input-group">
<button type="submit" class="btn" name="register_btn">Register</button>
</div>
<p>
Already a member? <a href="login.php">Sign in</a>
</p>
</form>
</body>
</html>
```

7.3 DATABASE SCHEMA

labl_0 = Label(base, text="Registration form",width=20,font=("bold",20))
labl_0.place(x=90,y=53)
lb1= Label(base, text="Enter Name", width=10, font=("arial",12))
lb1.place(x=20, y=120)
en1= Entry(base) en1.place(x=200, y=120)
lb3= Label(base, text="Enter Email", width=10, font=("arial",12))
lb3.place(x=19, y=160)
en3= Entry(base) en3.place(x=200, y=160)
lb4= Label(base, text="Contact Number", width=13,font=("arial",12))
lb4.place(x=19, y=200)
en4= Entry(base) en4.place(x=200, y=200)
lb5= Label(base, text="Select Gender", width=15, font=("arial",12))
lb5.place(x=5, y=240)
var = IntVar()
Radiobutton(base, text="Male", padx=5,variable=var, value=1).place(x=180, y=240)
Radiobutton(base, text="Female", padx =10,variable=var, value=2).
place(x=240,y=240)
Radiobutton(base, text="others", padx=15, variable=var,
value=3).place(x=310,y=240)
list_of_cntry = ("United States", "India", "Nepal",
"Germany") cv =StringVar()
drplist= OptionMenu(base, cv, *list_of_cntry) drplist.config(width=15)
cv.set("United States")
lb2= Label(base, text="Select Country", width=13,font=("arial",12))
lb2.place(x=14,y=280)
drplist.place(x=200, y=275)
lb6= Label(base, text="Enter Password", width=13,font=("arial",12))
lb6.place(x=19, y=320)
en6= Entry(base, show='*') en6.place(x=200, y=320)
lb7= Label(base, text="Re-Enter Password", width=15,font=("arial",12))
lb7.place(x=21, y=360)

```
en7 =Entry(base, show=&#39;*&#39;) en7.place(x=200, y=360)Button(base,
text=&quot;Register&quot;, width=10).place(x=200,y=400) base.mainloop()
def generateOTP() :
# Declare a digits variable # which stores all digits digits = &quot;0123456789&quot;
OTP = &quot;&quot;
# length of password can be changed # by changing value in range
for i in range(4) :
OTP += digits[math.floor(random.random() * 10)] return OTP
# Driver code
if name == &quot; main &quot; :
print(&quot;OTP of 4 digits:&quot;, generateOTP()) digits=&quot;0123456789&quot;
OTP=&quot;&quot;
for i in range(6): OTP+=digits[math.floor(random.random()*10)]
otp = OTP + &quot; is your OTP&quot; msg= otp
s = smtplib.SMTP(&#39;smtp.gmail.com&#39;, 587) s.starttls()
s.login(&quot;Your Gmail Account&quot;, &quot;You app password&quot;) emailid =
input(&quot;Enter your email: &quot;)
s.sendmail(&#39;&amp;&amp;&amp;&amp;&amp;&amp;&amp;&amp;&amp;&amp;&amp;&#39;,e
mailid,msg) a = input(&quot;Enter Your OTP &gt;&gt;: &quot;)
if a == OTP: print(&quot;Verified&quot;)
else:
print(&quot;Please Check your OTP again&quot;)
```

CHAPTER 8

TESTING

8.1 TEST CASES

SNO	TEST CASE	FEATURE	STEPS TO EXECUTE	EXPECTEDRESULT	ACTUAL RESULT	EXECUTED BY
1	FUNCTIONAL	LOGIN	LOGIN TO EXECUTE BY FILLING THE DETAILS	CORRECT LOGIN CREDENTIALS	WORKING AS EXPECTED	Selciyadevi S
2	FUNCTIONAL	REGISTRATION	REGISTRATION THROUGH FORMS	REGISTRATION FORMTO BE FILLED AND DISPLAYED	WORKING AS EXPECTED	Vimal prasad M
3	FUNCTIONAL	WOKWI	TO DEVELOP THE IOT DEVICE AND CODE THE IOT DEVICE	SENSE THE DATA	WORKING AS EXPECTED	Silviya X
4	FUNCTIONAL	IBMWATSON	PUSHTHE SENSED DATA FROM WOKWI	SENSEDDATA IN IBMWATSON	WORKING AS EXPECTED	Srinithi V

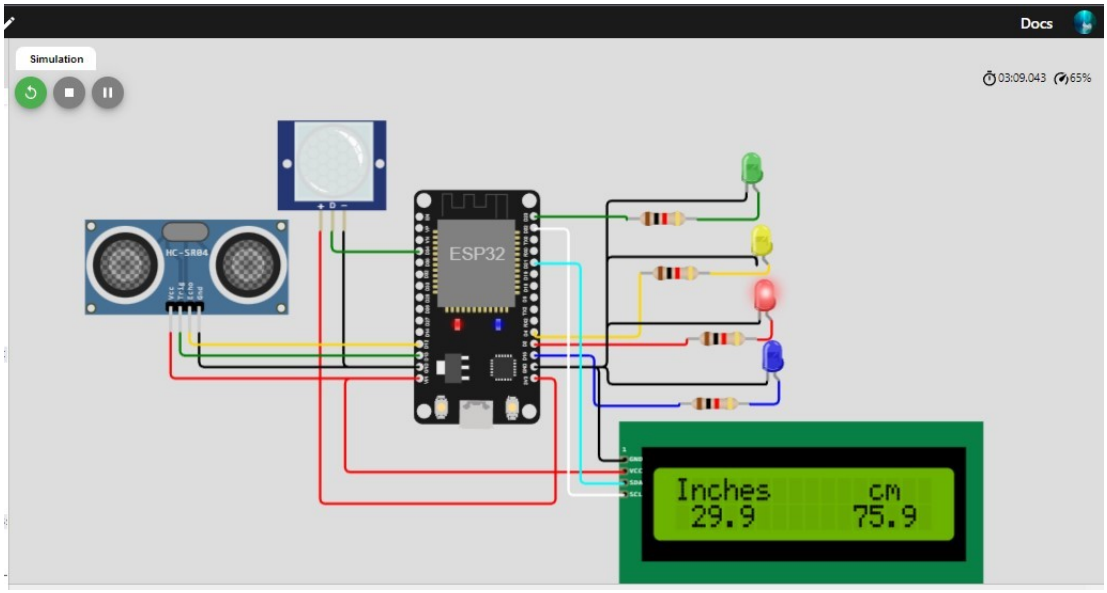
5	FUNCTIONAL	NODERED	TO CONNECT WITH THE IBM WATSON AND THEN COLLECT THE SENSEDDATA AND DISPLAY IN NODE RED DASHBOARD	VISUAL REPRESENTATION OFSENSED DATA IN NODE RED DASHBOARD	WORKING AS EXPECTED	Srinithi V, Selciyadevi S
6	TESTING	TESTTHE ENTIRE WORK	TO CHECK ALL THE MENTIONED TESTCASE ARE WORKINGPROPERLY	TEST CASE ARE WOKING PROPERLY	WORKING AS EXPECTED	Silviya X, Vimal prasad M

CHAPTER 9

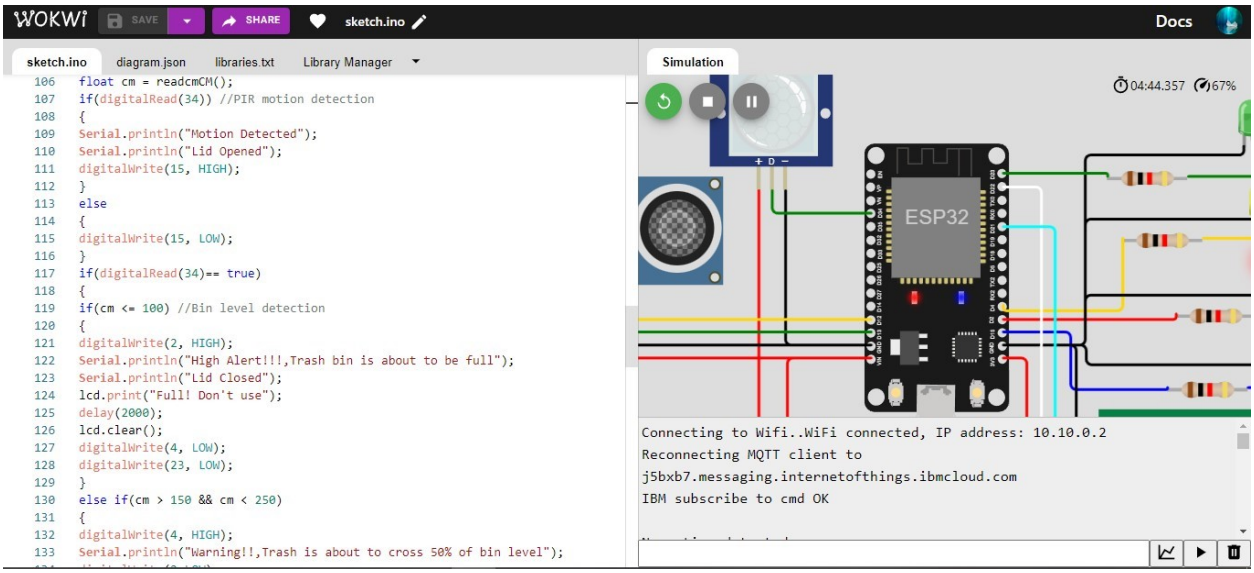
RESULTS

9.1 PERFORMANCE METRICS

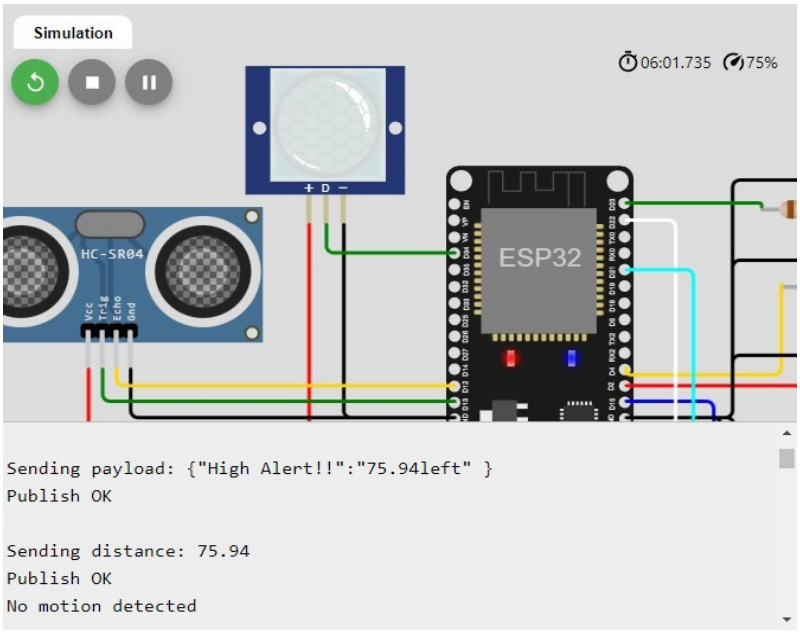
CIRCUIT DIAGRAM



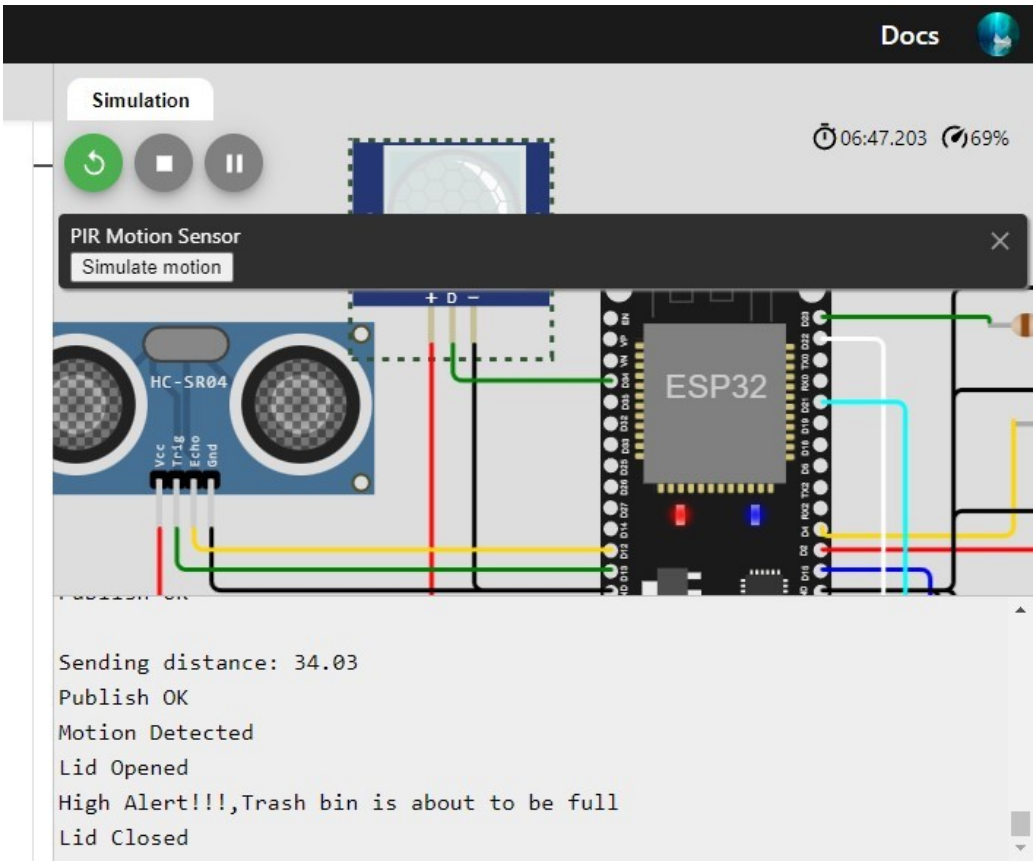
WOKWI SIMULATION



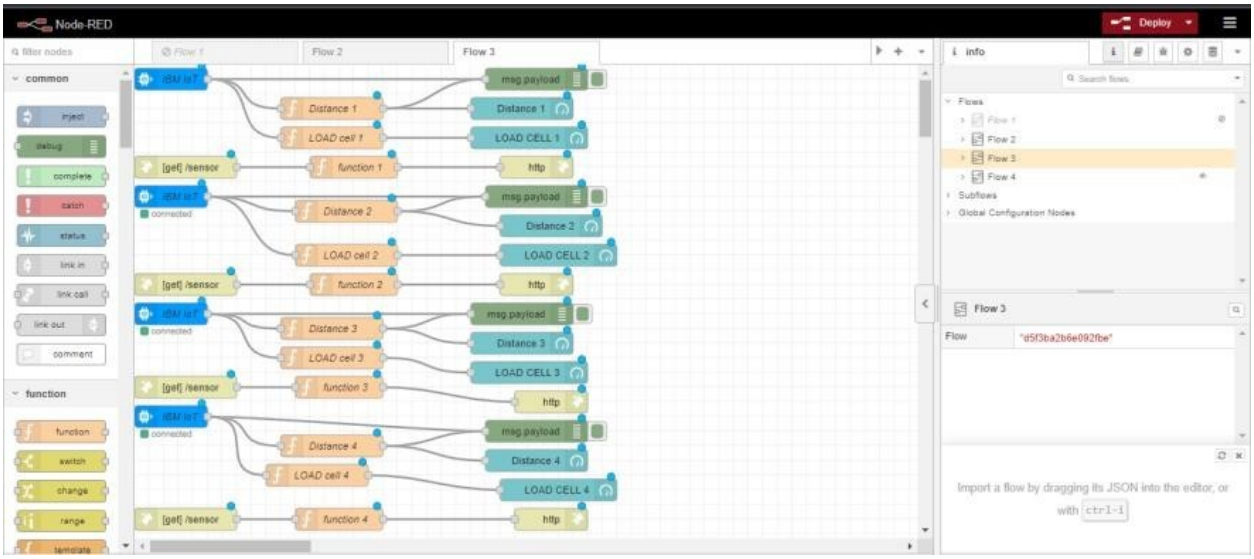
WHEN NO MOTION DETECTED



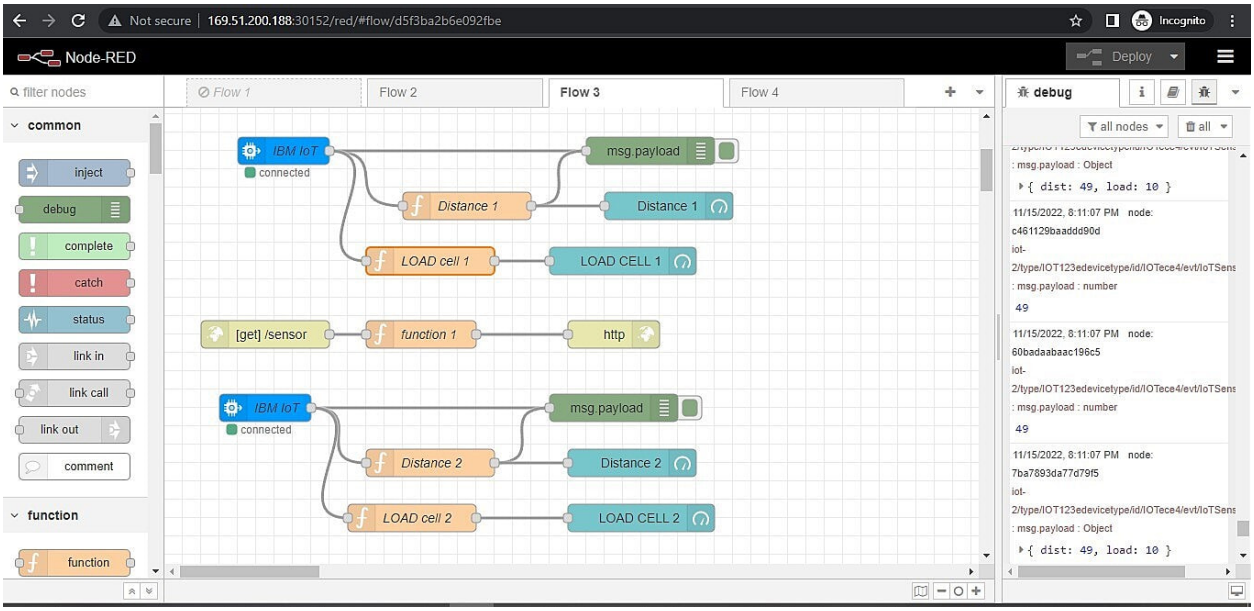
WHEN MOTION DETECTED



Node-RED Connection setup for data transmission from IBM Watson IOT platform to Node-REDdashboard:



Data transfer from IBM WatsonIOT platform and python sctrip to Node-RED:

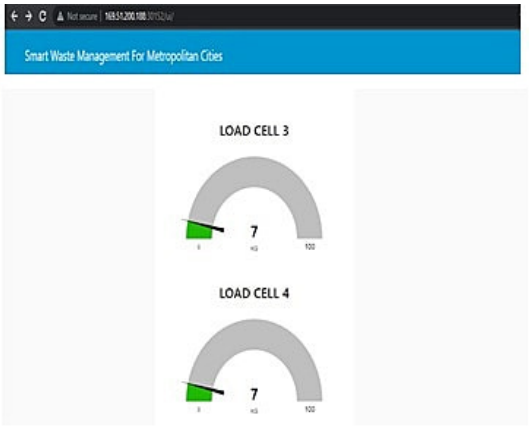


WEB UI:


Smart Waste Management For Metropolitan Cities
Smart Waste
Distance 1
30
LOAD CELL 1
6
Distance 2


Smart Waste Management For Metropolitan Cities
Distance 2
29
LOAD CELL 2
14


Smart Waste Management For Metropolitan Cities
Distance 3
50
Distance 4
50


Smart Waste Management For Metropolitan Cities
LOAD CELL 3
7
LOAD CELL 4
7

CHAPTER 10

ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- 24×7 monitoring system is designed for monitoring dumpsters
- Smart and organized system is designed for selective clearing
- The ultrasonic sensor is used for measuring the level of waste in the dumpsters
- Segregating wet and dry waste
- Separating wet and dry waste
- If either of the containers is full then an alert message is sent from the dumpster
- In turn, employees can clear the corresponding dumpster
- Saves the Earth and conserves energy
- Reduces environmental pollution

DISADVANTAGES

- It reduces man power requirements which results into increase in unemployments for unskilled people.
- The training has to be provided to the people involved in the smart waste management system.
- System requires more 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.

CHAPTER 11

CONCLUSION

An effective waste management system is necessary for long-term development. This would be a tiny step toward overcoming the challenges of waste management in a developing country. This step protects us from the health and the environmental hazards caused by a slow poison are linked. Technology in the twenty-first century refines us and our surroundings into a better identity and a safe place to live. It is time to bring about change and show respect, love, and care for the beings who have assisted us in our survival. There is an approach. And so this initiative is only a very small part of our progress.

CHAPTER 12

FUTURE SCOPE

The proposed system will undergo several future directions and improvements, including the following:

1.Modify the security and molecular lock of bins to aid in the protection of the bin from damage or robbery.

2.The aim of sustainable points should encourage residents or end consumers to interact, making the quality plans and contributing in the realization of communal wastewater treatment efforts, thus completing the Swatch bharat Bharath concept.

3.Having case stancy or business intelligence on the variety and periods waste are produced on alternating schedules or seasons, attempting to make bin refilling dependable and reducing reliance on computer devices, and fixing the positions.

4.Improving the graphical user interfaces of the Server and Android.

CHAPTER 13

APPENDIX

Source Code:

PYTHON CODE:

```
import wiotp.sdk.device
import time
import random
myConfig = {
   "identity":{
      "orgId":"j5bxb7",
      "typeId":"IOT123edevicetype",
      "deviceId":"IOTece4"
    },
    "auth": {
      "token":"e2)-17xkqIFMvm3@II"
   }
}
def myCommandCallback(cmd):
   print("Message received from IBM IoT Platform:%s"%cmd.data['command'])
   m=cmd.data['command']
client=wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=None)
client.connect()
def pub(data):

client.publishEvent(eventId="binstatus",msgFormat="json",data="myData",qos=0,onPublish=Non
e)
   print("Published data Successfully:%s",myData)
```

```
while True:
    myData={'name':'Bin1','lat':13.092677,'lon':80.188314}
    pub(myData)
    time.sleep(3)
    client.commandCallback=myCommandCallback
client.disconnect()
```

CODE FOR DATA TRANSFER FROM SENSORS:

```
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
// credentials of IBM Accounts -
#define ORG "j5bxb7" //IBM organisation id
#define DEVICE_TYPE "IOT123edevicetype" // Device type mentioned in ibm watson iot platform
#define DEVICE_ID "IOTece4" // Device ID mentioned in ibm watson iot platform
#define TOKEN "e2)-17xkqIFMvm3@ll" // Token
// customise above values - char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // server name  char publishTopic[] = "iot-2/evt/data/fmt/json";
char topic[] = "iot-2/cmd/led/fmt/String"; // cmd Represent type and command is test format of strings  char authMethod[] = "use-token-auth"; // authentication method  char token[] = TOKEN; char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //Client id //
WiFiClient wifiClient; // creating instance for wificlient
PubSubClient client(server, 1883, wifiClient);
#define ECHO_PIN 12
#define TRIG_PIN 13
float dist;
void setup()
{
Serial.begin(115200);
```

```
pinMode(LED_BUILTIN, OUTPUT); pinMode(TRIG_PIN, OUTPUT); pinMode(ECHO_PIN, INPUT);

//pir pin

pinMode(4, 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);
}
else
{
digitalWrite(15, LOW);
}
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 > 150 && cm < 250)
{
digitalWrite(4, HIGH);
Serial.println("Warning!!,Trash is about to cross 50% of bin level");  digitalWrite(2,LOW);
digitalWrite(23, LOW);
}
else if(cm > 250 && cm <=400)
```

```
{
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");
}
if(cm <= 100)
{
digitalWrite(21,HIGH);
String payload = "{\"High Alert!!\":\""; payload += cm;  payload += "left\" }";
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 or prints publish failed
{
Serial.println("Publish OK");
}
}
if(cm <= 250)
{
digitalWrite(22,HIGH);
String payload = "{\"Warning!!\":\""; payload += dist;  payload += "left\" }"; Serial.print("\n");
Serial.print("Sending distance: ");  Serial.println(cm);  if(client.publish(publishTopic,(char*)
payload.c_str()))
{
Serial.println("Publish OK");
```

```
}

else

{

Serial.println("Publish FAILED");

}

}

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();

}
```

GITHUB LINK :

https://github.com/IBM-EPBL/IBM-Project-22105-1659804669