Smart Waste Management System For Metropolitan Cities

PROJECT REPORT

Submitted by

Team ID: PNT2022TMID41127

Team member:

NETHAJI.S **612719106017**

KANAGARAJ.P **612719106007**

AJAY KUMAR.M **612719106001**

In partial fulfilment for the award of the degree

Of

BACHELOR OF ENGINEERING

In

ELECTRONIC AND COMMUNICATION ENGINEERING

THE KAVERY ENGINEERING COLLEGE,

MECHERI: SALEM-636453

ANNA UNNIVERSITY: CHENNAI 600 025

ABSTRACT

One issue that most cities and municipalities are dealing with currently, is the degradation of environmental cleanliness with reference to waste management. This is a result of improper garbage collection management. Dumping garbage onto the streets and in public areas is a common synopsis found in all developing countries and this mainly ends up affecting the environment and creating several unhygienic conditions. To avoid improper garbage management and to create a hygienic environment, the concept of automation is used in waste management system. Any city being referred to as a "smart city" is because of its orderly and tidy surroundings. But currently, many issues including those related to smart grids, smart environments, and smart living are faced. Today, cities and metropolitan areas' top priority is proper garbage management.

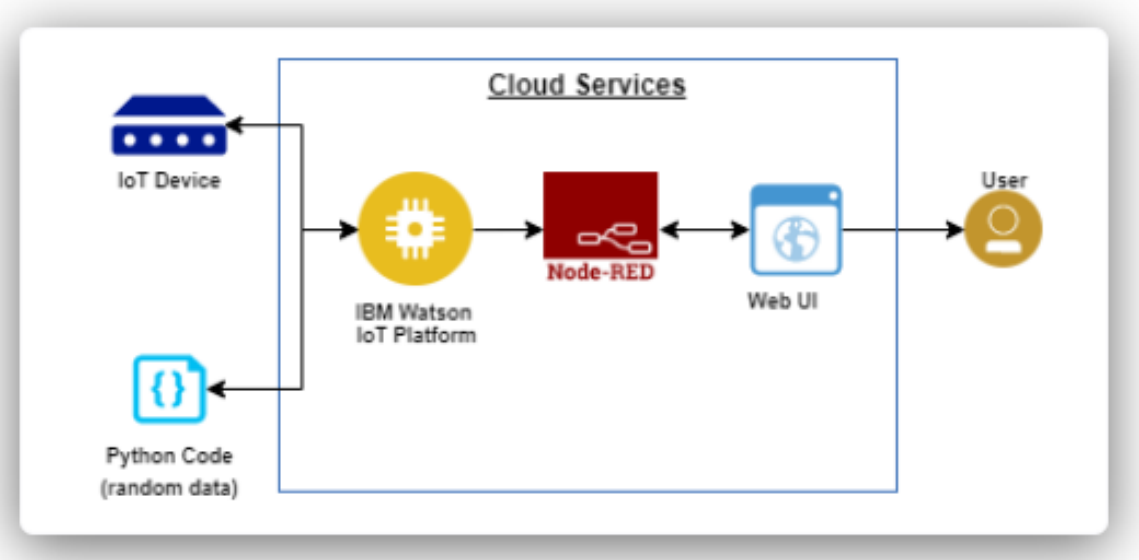
Traditional waste management techniques are too simplistic to create an effective and reliable waste management. The ideology put forward includes hardware and software technologies i.e. connecting Wi-Fi system to the normal dustbin in order to provide free internet facilities to the user for a particular period of time. The technology awards the user for keeping the surrounding clean and thus work hand in hand for the proper waste management in a locality. The smart bin uses multiple technologies - firstly the technology for measuring the amount of trash dumped and secondly the movement of the waste and lastly sending necessary signals and connecting the user to the WiFi system. The proposed system will function on client server model, a cause that will assure clean environment, good health, and pollution free society.

TABLE OF CONTENT

1. INTRODUCTION

1.1 project overview

Waste management from small town to the metropolitan city by the local government, NGO's of that city as well as peoples of that city. Metropolitan cities like Delhi, Mumbai, Chennai, Pune, Indore, Mysore, Bangalore, etc. As we are seeing facing a lot of issues for the implementation of waste management. Therefore from this paper, I want to see Smart waste management concept nowadays emerges a new phenomenon and it is mostly applied in the Metropolitan cities where the production of the waste is high and management of waste and awareness about the waste management in between the people is very low. Smart waste management helps to reduce the waste, create waste to energy source also it helps to keep the environment clean and neat. All the city's urban local bodies depending upon the available technology have to spend the money and innovate the new concept of waste management that is the main purpose of smart waste management. This paper is based on the secondary as well as primary data. Secondary data took from the newspaper, article, etc. And primary data based on the observation and survey that did in 2016 and recent in Mumbai. At the end in the discussion try to show the difference between small-town waste management and. Keywords: Healthy environment, Local Government, Metropolitan Cities, Smart city, Waste management.



1.2 PURPOSE

Round Glass Foundation's Waste Management program aims to make cleaner village and cities. The program implements a decentralized segregation model for composting and recycling for efficient, cost-effective, and sustainable waste management. The Foundation's Waste Management model partners with the Government o to provide a functioning solution to manage rural waste.

2 LITERATURE SURVEY

2.1 Existing problem

In this webinar with UN-Habitat's Waste Wise Cities, we share the findings from four contrasting towns and cities from Africa and South Asia. Hear from our CEO, Sarah Roberts, as well as a panel of experts and local stakeholders.

- Sarah Roberts (CEO of Practical Action)
- Dr Lucy Stevens (Acting Director Influence and Impact at Practical Action)
- Noémie de la Brosse (Consultant at OPLM)
- Mrs Aklima Akhter (Waste Worker from Faridpur, Bangladesh)
- Mr. Shamim Ahmed (President of BPGMEA – Bangladesh Plastic Goods Manufacturers and Exporters Association)
- John Sande (Deputy Director of Environment, County Government of Kisumu, Kenya)
- Lucy Slack (Secretary-General of Commonwealth Local Government Forum)
- Yashwant Deval (Area Director for India & Bangladesh at RiverRecycle

2.2 Reference

1 IOT-BASED SMART WASTE MANAGEMENT FOR ENVIRONMENTAL15 November2019 Dinesh Goyal S.Balamurugan ,sheng-lung Peng.

Highlighting a wide range of topic such as e- commerce, security management, and web infrastructure, this book is ideal for academicians ,students, researchers, industry professional, IT consultants, engineers, and scientists. using this technology , waste management agencies can increase operational efficiency , cut cost, and enhance customer satisfaction

2 WASTE MANAGEMENT IN THE SMART CITY 23 march 2019 Burak kantarci Sema oktug

Opportunistically exploiting internet of things for wireless sensor networking routing in smart cities. this book is a printed edition of the special issue wireless sensor and actuator network for smart cities that was published in jasan.

3 SECURITY AND ORGANIZATION WITH IN IOT AND SMART CITIES 10 June 2019

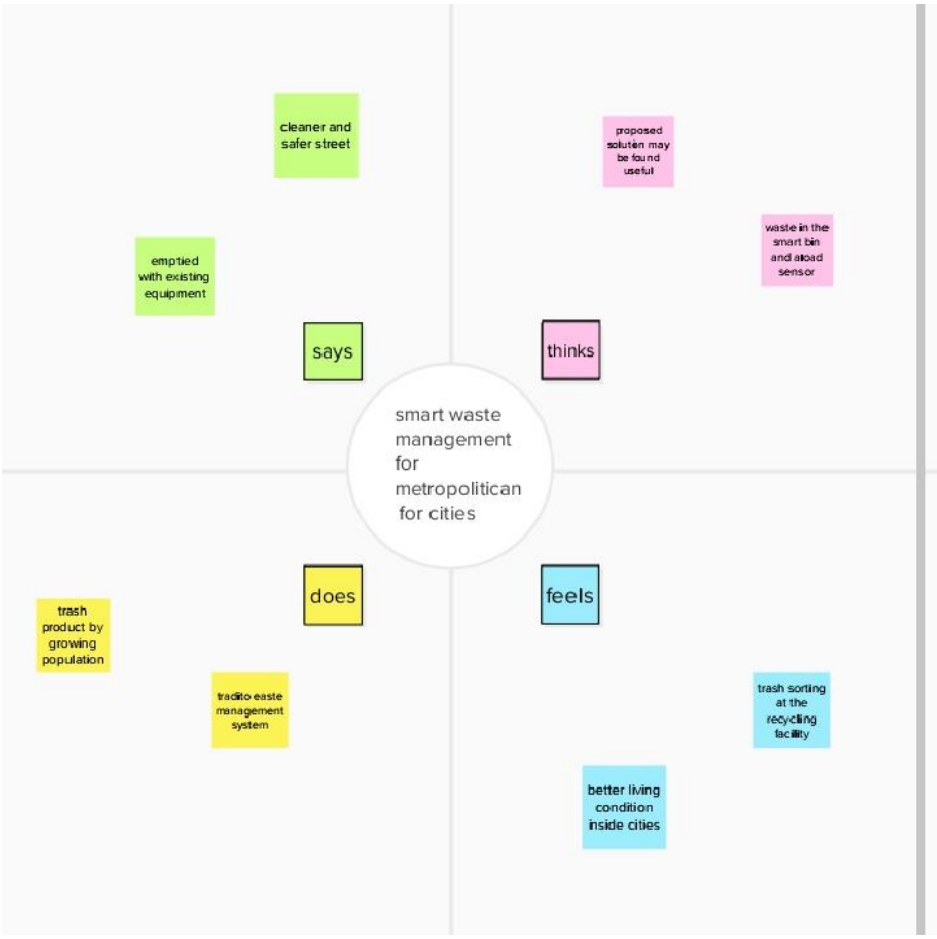
This book explores the most important IOT automated and smart applications to help the reader understand the principle of using IOT in such application. this book provides state of the art research result and discusses current issues , challenges, solutions and recent trends related to security and organization within iot and smart cities.

2.3 Problem statement definition

Recyclable quality depends largely on geographic location. For example, Seattle and the rest of the Pacific Northwest is ecologically conscious and the quality of their recyclable materials is generally very high. The quality elsewhere in the country can vary greatly. These days you'll see just about everything moving down the line, including the kitchen sink. Some of these items (especially garden hoses) can actually get wrapped around equipment and cause real damage. Having knowledgeable onsite operations managers and well trained employees will help you maintain productivity even when the quality of recyclables is low.

3 Ideation and proposed solution

3.1 empathy map canvas



3.2 Ideation brainstorm

The waste management services take care of a healthy environment allowing optimization of the utilities and prevent overloading the carrier for waste disposal. Smart waste management also contributes to the overall waste recycling efficiency and provides the route optimization opportunity for utilities to reduce traffic and fuel use.

An example of a modern smart waste management system would include; a sensor attached to the trash bin that measures fill level; and a communication system that transfers this data to Cloud. Data is processed in the Cloud, thus, the route of collection trucks is optimized.

Smart waste management companies have recently developed solutions based on ultrasonic distance measurement. Some companies prefer to approach the problem with an alternative solution using image processing and camera as a passive sensor. However, the majority of these solutions use ultrasonic sensor for measurement of the distance.

3.3 proposed solution

s.no	parameter	description
1	Problem Statement (Problem to be solved)	This project enables the organizations to meet their needs of smart garbage management systems. This system allows the authorized person to know the fill level of each garbage bin in a locality or city at all times, to give a costeffective 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.
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 neighborhood of landfills to communities, breeding of pests and in property values.
5	Business Model (Revenue Model)	Waste Management organizes 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 function.
6	Scalability of the Solution	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

4 REQUIREMENT ANALYSIS

4.1 Functional requirement

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. 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 bin discharge in the area. The tool assigns bin a rating (1-10) and calculates distance from 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. Raspberry Pi camera with 12 MP and high resolution of upto 1080p is used. By using real-time data on fill-levels and pick recognition, we can show you how full the bins you collect are.

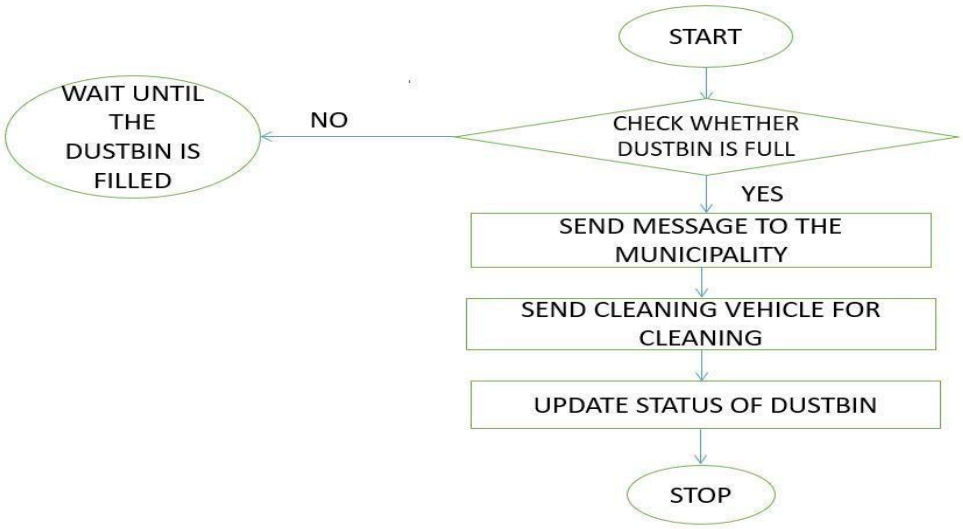
Non functional requirments

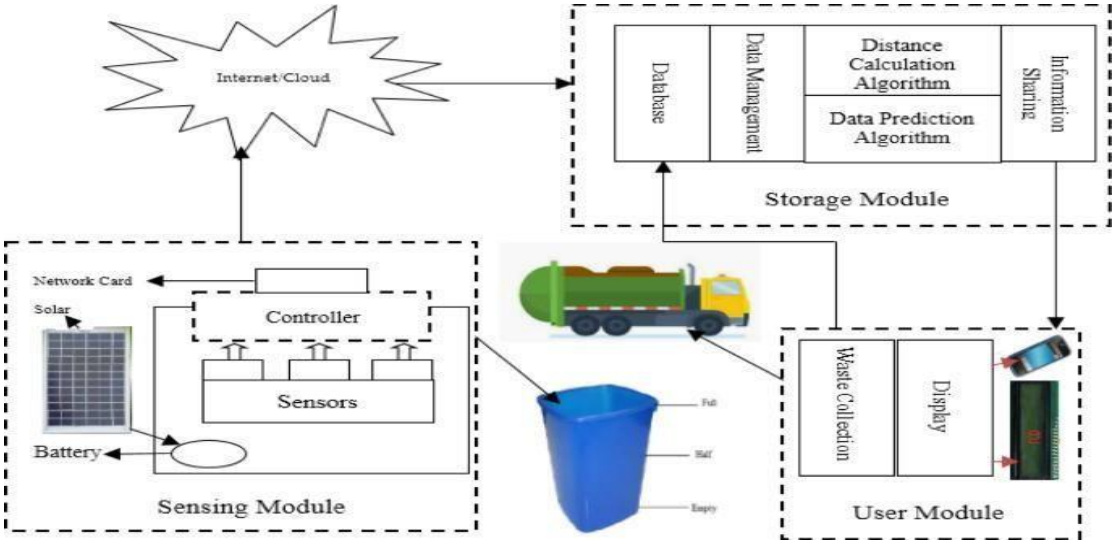
NFR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	IoT device verifies that usability is a special and important perspective to analyze user requirements, which can further improve the design quality. In the design process with user experience as the core, the analysis of users' product 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 Compost it Purchase wisely and recycle Avoid using use and throw 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 datadriven 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 and cities because we are able to monitor the garbage 24/7 more cost effectively and scalability is high

5 PROJECT DESIGN

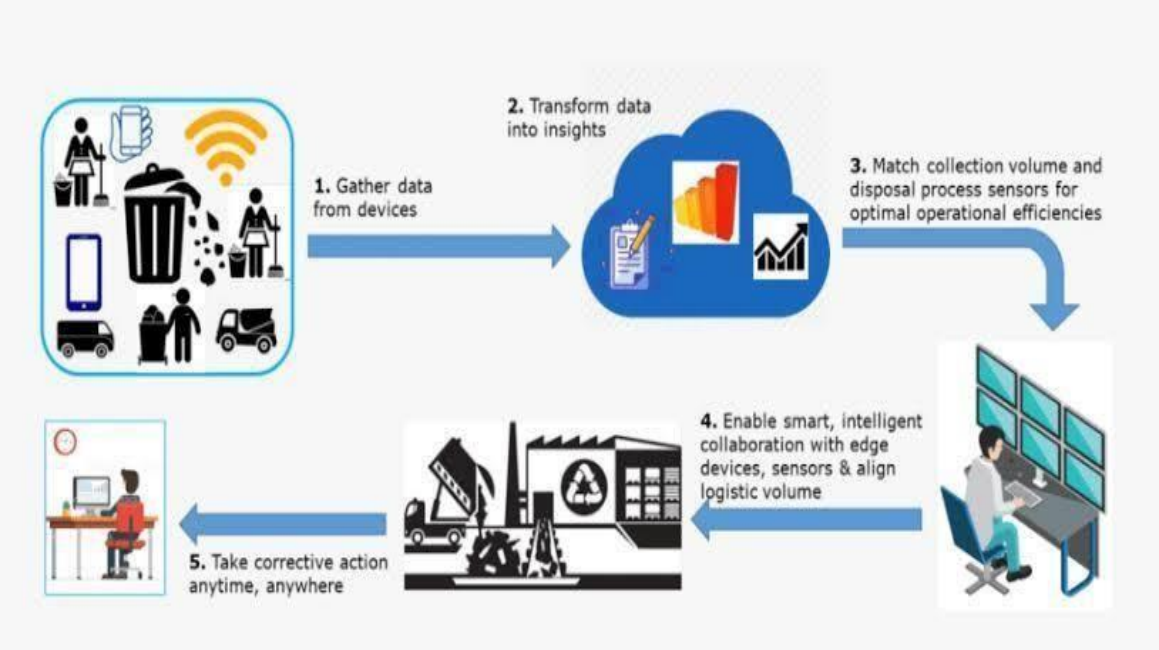
5.1 Data flow diagram

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 information and where data is stored.





5.2 solution & technical architecture



Guidelines:

1. Our proposed model provide real time monitoring to the garbage bins placed in various locations.
2. The garbage bins are build with a sensor module(Ultrasonicc sensor) which continuosly monitors the garbage bin. Any moment the garbage level passes over the critical level(i.e80%),thesystem generates a notification to the monitoring panel (admin panel
   3 garbage cleaning team) and so the cleaning team collects the garbage from the identified garbage bin.

Table-1 : Components & Technologies

1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.

7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

5.3 User stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Authorised Person(Manages web app)	Login	USN-1	As an authorised person, I gave user id and password for every workers and manage them.	I can access web page /dashboard	Medium	Sprint-2
Admin	Login	USN-2	As a admin, I will manage garbage level monitor. When garbage gets filling alert, I will post location and garbage Id to trash truck.	I can manage garbage monitoring.	High	Sprint-1
Truck Driver	Login	USN-3	As a driver, I'll follow the route sent by user to reach the filled garbage location.	I can drive to reach the garbage filled route in dynamic route given.	Medium	Sprint-2
Garbage Collector	Login	USN-4	As a garbage collector, I'll collect all the garbage from garbage bin and load it to the truck and send them to landfill.	I can collect garbage and pulled to truck.	Medium	Sprint-2
	Login	USN-5	As a municipality, I'll check the process are happening in discipline manner without .	I can manage all the process going good.	High	Sprint-1

6 PROJECT PLANNING &SCHEDULING

6.1 Sprint planning & estimation

Title	Description	Details
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers,research publication etc .	28 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem Statements.	24 SEPTEMBER 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	25 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	23 SEPTEMBER 2022
Problem Solution Fit	Prepare problem - solution fit Document.	30 SEPTEMBER 2022
Solution Architecture	Prepare solution architecture Document.	28 SEPTEMBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	20 OCTOBER 2022
Functional Requirements	Prepare the functional requirement document.	08 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	09 OCTOBER 2022
Technology Architecture	Prepare the technology architecture diagram.	10 OCTOBER 2022

6.2 Sprint delivery schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application enteringmy email,password, and confirming my password.	2	High	Nethaji
Sprint-2	Registration	USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Ajay kumar
Sprint-3	Registration	USN-3	As a user, I can register for the application through Facebook	2	Low	Kanagaraj
sprint-4	Registration	USN-4	As a user, I can register for the application through Gmail	2	Medium	Kanagaraj
Sprint-1	Login	USN-5	As a user, I can log into the application by Entering email & password	1	High	Nethaji

7 CODING AND 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.

Web Application to get the Live location:

index.html:

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width">
    <title>Smart Waste Management System</title>
    <link rel="icon" type="image/x-icon" href="/imgs/DUMPSTER.png">
    <link href="style.css" rel="stylesheet" type="text/css" />
    <script src="https://www.gstatic.com/firebasejs/9.14.0/firebase-app.js"></script>
    <script src="https://www.gstatic.com/firebasejs/9.14.0/firebase-analytics.js"></script>
    <script src="https://www.gstatic.com/firebasejs/9.14.0/firebase-database.js"></script>

    <script>
      var firebaseConfig =
      {
        apiKey: "AIzaSyCcZk7b1CLOGviwUpthRDLotrmFX0MFuTs",
        authDomain: "swms-3840.firebaseapp.com",
        projectId: "swms-3840",
```

```
            storageBucket: "swms-3840.appspot.com",
            messagingSenderId: "479902726304",
            appId: "1:479902726304:web:3d822880d1275ee57a71c5",
            measurementId: "G-MHP4N77MTP"
        };
        firebase.initializeApp(firebaseConfig)
    </script>
    <script defer src="db.js"></script>
</head>

<body style="background-color:#1F1B24;">
    <script src="maps.js"></script>
    <div id="map_container">
        <h1 id="live_location_heading" >LIVE LOCATION</h1>
        <div id="map"></div>
        <div id="alert_msg">ALERT MESSAGE!</div>
    </div>
    </div>
    <center>
        <a href="https://goo.gl/maps/G9XET5mzSw1ynHQ18" type="button" class="btn
btn-dark">
            DUMPSTER
        </a>
    </center>
    <script
        src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBBLyWj-
3FWtCbCXGW3ysEiI2fDfrv2v0Q&callback=myMap"></script></div>
</body>
</html>
```

db.js:

```
const cap_status = document.getElementById("cap_status");
const alert_msg = document.getElementById("alert_msg");

var ref = firebase.database().ref();

ref.on(
  "value",
  function (snapshot) {
    snapshot.forEach(function (childSnapshot) {
      var value = childSnapshot.val();

      const alert_msg_val = value.alert;
      const cap_status_val = value.distance_status;

      alert_msg.innerHTML = `${alert_msg_val}`;
    });
  },
  function (error) {
    console.log("Error: " + error.code);
  }
);
```

maps.js:

```
const database = firebase.database();

function myMap() {
  var ref1 = firebase.database().ref();

  ref1.on(
    "value",
    function (snapshot) {
```

```
      snapshot.forEach(function (childSnapshot) {
        var value = childSnapshot.val();
        const latitude = value.latitude;
        const longitude = value.longitude;

        var latlong = { lat: latitude, lng: longitude };
        var mapProp = {
          center: new google.maps.LatLng(latlong),
          zoom: 10,
        };
        var map = new google.maps.Map(document.getElementById("map"), mapProp);
        var marker = new google.maps.Marker({ position: latlong });
        marker.setMap(map);
      });
    },
    function (error) {
      console.log("Error: " + error.code);
    }
  );
}
```

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

Code to evaluate the level of the garbage in bin:

bin1.py:

```
import requests
import json
```

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

# watson device details
organization = "73ffyv"
devicType =  "BIN1"
deviceId = "BIN1ID"
authMethod= "token"
authToken= "123456789"

#generate random values for randomo variables (temperature&humidity)
def myCommandCallback(cmd):
    global a
    print("command recieved is:%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("Exception while connecting device %s" %str(e))
    sys.exit()
#connect and send a datapoint "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:' 'Garbage level is high, collection time :) 90 %'
    elif distance < 40 and distance >16:
            dist = 'Risk warning:' 'garbage 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 :' ' Garbage level is high, collection time :)'
    elif load == "60 %" or distance == "60 %":
            warn = 'alert :' 'garbage is above 60%'
    else :
            warn = 'alert :' 'Levels are low, collection not needed '


    def myOnPublishCallback(lat=11.035081,long=77.014616):
        print("Peelamedu, Coimbatore")
        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()
```

bin2.py:

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

# watson device details
organization = "73ffyv"
devicType =  "BIN2"
deviceId = "BIN2ID"
authMethod= "token"
```

```
authToken= "123456789"


#generate random values for randomo variables (temperature&humidity)
def myCommandCallback(cmd):
    global a
    print("command recieved is:%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("Exception while connecting device %s" %str(e))
    sys.exit()


#connect and send a datapoint "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:' 'Garbage level is high, collection time :) 90 %'
    elif distance < 40 and distance >16:
        dist = 'Risk warning:' 'garbage 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 :' ' Garbage level is high, collection time :)'
    elif load == "60 %" or distance == "60 %":
        warn = 'alert :' 'garbage is above 60%'
    else :
        warn = 'alert :' 'Levels are low, collection not needed '


    def myOnPublishCallback(lat=11.068774,long=77.092978):
        print("PSG iTech, Coimbatore")
        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()
```

bin3.py:

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

# watson device details
organization = "73ffyv"
devicType =  "BIN3"
deviceId = "BIN3ID"
authMethod= "token"
authToken= "123456789"

#generate random values for randomo variables (temperature&humidity)
def myCommandCallback(cmd):
    global a
    print("command recieved is:%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("Exception while connecting device %s" %str(e))
    sys.exit()


#connect and send a datapoint "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:' 'Garbage level is high, collection time :) 90 %'
    elif distance < 40 and distance >16:
        dist = 'Risk warning:' 'garbage 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 :' ' Garbage level is high, collection time :)'
    elif load == "60 %" or distance == "60 %":
        warn = 'alert :' 'garbage is above 60%'
    else :
        warn = 'alert :' 'Levels are low, collection not needed '


    def myOnPublishCallback(lat=11.007403,long=76.963439):
        print("Kattoor, Coimbatore")
        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()
```

bin4.py:

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

# watson device details
organization = "73ffyv"
devicType =  "BIN4"
deviceId = "BIN4ID"
authMethod= "token"
authToken= "123456789"

#generate random values for randomo variables (temperature&humidity)
def myCommandCallback(cmd):
    global a
    print("command recieved is:%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("Exception while connecting device %s" %str(e))
```

```
    sys.exit()


#connect and send a datapoint "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:' 'Garbage level is high, collection time :) 90 %'
    elif distance < 40 and distance >16:
        dist = 'Risk warning:' 'garbage 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 :' ' Garbage level is high, collection time :)'
    elif load == "60 %" or distance == "60 %":
        warn = 'alert :' 'garbage is above 60%'
    else :
        warn = 'alert :' 'Levels are low, collection not needed '

    def myOnPublishCallback(lat=11.453306,long=77.426024):
        print("Seethammal Colony, Gobichittipalayam")
        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()
```

7.3 Feature 3:

An additional feature added to the smart waste management system is to measure the weight of the bin using hx711 load cell. The weight of the bin is the output of the below code.

Measuring the weight of the garbage bin:

main.py:

```
from hx711 import HX711
hx = HX711(5,4,64)
print(1)
while True:
    hx.tare()
    read = hx.read()
    value=hx.read_average()
    print(value,"#")
```

hx711.py:

```
from machine import Pin, enable_irq, disable_irq, idle


class HX711:
    def __init__(self, dout, pd_sck, gain=128):
        self.pSCK = Pin(pd_sck , mode=Pin.OUT)
        self.pOUT = Pin(dout, mode=Pin.IN, pull=Pin.PULL_DOWN)
        self.pSCK.value(False)


        self.GAIN = 0
        self.OFFSET = 0
        self.SCALE = 1
        self.time_constant = 0.1
        self.filtered = 0
        self.set_gain(gain);
```

```
def set_gain(self, gain):
    if gain is 128:
        self.GAIN = 1
    elif gain is 64:
        self.GAIN = 3
    elif gain is 32:
        self.GAIN = 2
    self.read()
    self.filtered = self.read()
    print('Gain & initial value set')

def is_ready(self):
    return self.pOUT() == 0

def read(self):
    # wait for the device being ready
    while self.pOUT() == 1:
        idle()

    # shift in data, and gain & channel info
    result = 0
    for j in range(24 + self.GAIN):
        state = disable_irq()
        self.pSCK(True)
        self.pSCK(False)
        enable_irq(state)
        result = (result << 1) | self.pOUT()

    # shift back the extra bits
    result >>= self.GAIN
```

```
    # check sign
    if result > 0x7fffff:
        result -= 0x1000000

    return result

def read_average(self, times=3):
    s = 0
    for i in range(times):
        s += self.read()
    ss=(s/times)/210
    return '%.1f' %(ss)

def read_lowpass(self):
    self.filtered += self.time_constant * (self.read() - self.filtered)
    return self.filtered

def get_value(self, times=3):
    return self.read_average(times) - self.OFFSET

def get_units(self, times=3):
    return self.get_value(times) / self.SCALE

def tare(self, times=15):
    s = self.read_average(times)
    self.set_offset(s)

def set_scale(self, scale):
    self.SCALE = scale
```

```
def set_offset(self, offset):
    self.OFFSET = offset

def set_time_constant(self, time_constant = None):
    if time_constant is None:
        return self.time_constant
    elif 0 < time_constant < 1.0:
        self.time_constant = time_constant

def power_down(self):
    self.pSCK.value(False)
    self.pSCK.value(True)

def power_up(self):
    self.pSCK.value(False)
```

8 TESTING

8.1 Test cases:

Unit testing

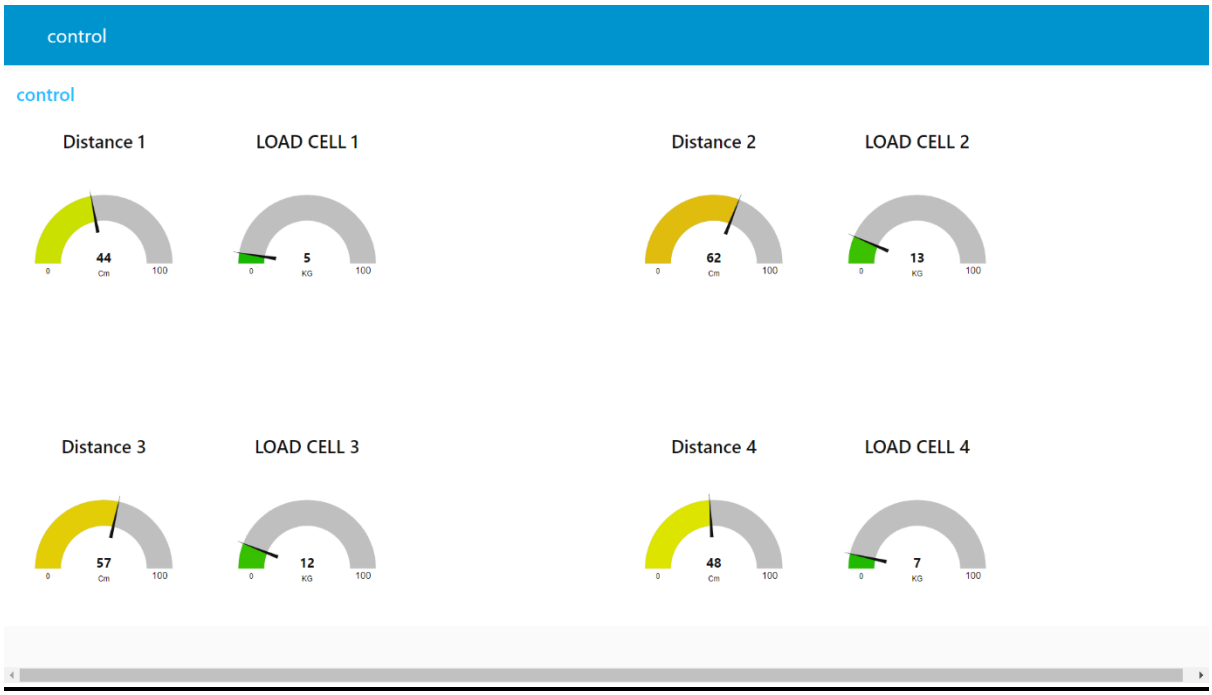
Test case no.	Sensor/Stage	Input	Expected output	Obtained output	Status
1.	Ultrasonic	Garbage level in bin i)Null ii)Full iii)Range in %	Correct level or distance	As expected	Pass
2.	ESP – 32	Microcontroller to process the input data	To collect the data from sensor	As expected	Pass
3.	Load cell	To measure 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.	HX710	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

Sample output:



Before implementation of Smart waste management system

		Ground truth	
		Emptying	Nonemptying
Predicted	Emptying	911	68
	Nonemptying	990	6041

After implementation of smart waste management system

		Ground truth	
		Emptying	Nonemptying
Predicted	Emptying	1720	90
	Nonemptying	181	6091

10 ADVANTAGES &DISAVANTAGES

Advantages:

- It saves time and money by using smart waste collection bins and systems equipped with fill level sensors. As smart transport vehicles go only to the filled containers or bins. It reduces infrastructure, operating and maintenance costs by upto 30%.

- It decreases traffic flow and consecutively noise due to less air pollution as result of less waste collection vehicles on the roads. This has become possible due to two way communication between smart dustbins and service operators.

- It keeps our surroundings clean and green and free from bad odour of wastes, emphasizes on healthy environment and keep cities more beautiful.

- It further reduces manpower requirements to handle the garbage collection process.

- Applying smart waste management process to the city optimizes management, resources and costs which makes it a "smart city".

- It helps administration to generate extra revenue by advertisements on smart devices.

Disadvantages:

- 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.
- 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. In RFID based systems, RFID tags are affected by surrounding metal objects (if any).
- It reduces man power requirements which results into increase in unemployments for unskilled people.
- The trainining has to be provided to the people involved in the smart waste management system.

11 CONCLUSION

Waste management is a major problem in sub-Saharan Africa , principally because of the lack of a recycling infrastructure, trained workforce, and other related factors. In 2010, it was estimated that 4.4 million tonnes of solid wastes were “mismanaged” on the continent . At the same time, it is important to acknowledge that waste management according to Nnaji , involves diverse stages, which include ‘‘generation and storage, collection and transfer, sorting, treatment, -material recovery and disposal.’’ In Nigeria, the infrastructure, trained labor, and willpower for these stages are poor, insufficient, or lacking. There are not enough functional waste collection and recycling facilities for plastic and nonplastic waste in Nigeria

According to Kofoworola ‘‘more than half of the waste in Lagos is left uncollected from the streets and the various locations due to the inadequacy and inefficiency of the waste management system’’. This is in line with what Ogwueleka reported; he stated that between 30 and 60% of solid waste generated in the country is uncollected. As such, the possibility that macroplastic waste (as part of such uncollected refuse) will disintegrate into microplastics is increased. In Abuja, the DFID estimated that the waste collection agency serves about 56% of individuals living in the Federal Capital Territory, but naji noted that 80% of people living in other Nigerian cities, towns, and villages do not receive the services of waste collectors, and thus dispose of their waste themselves. The dysfunctional state of most waste management agencies and infrastructure results in their inability to collect waste promptly and effectively, which in turn influences the quantity of plastic pollution.

12 FUTURE SCOPE

Although a lot of work has been done on the waste management topic, the concept of IoT-enabled waste management is quite new and the number of publications in this field is growing. The studies that have addressed IoT enabled waste management systems can be classified into the following four categories:

• Development of communication technologies and data transmission infrastructure

• Test the capabilities of IoT systems in field experiments

• Truck routing and scheduling for waste collection operations

Several studies have discussed the overall system architecture of IoT enabled waste management systems in which a number of bins are equipped with RFID tags for identification purpose, capacity sensors for waste level detection, actuators to lock the bin lids once they are filled, and wireless antennas to transmit sensor data to the network for waste collection operations

This section discusses the necessity for addressing the design, development, and implementation of an infrastructure for the collection of product lifecycle data that takes into account the synergistic nature of the above three elements. Particularly, the proposed framework views ‘waste’ as a ‘resource’, puts emphasis on waste reduction ‘upstream’, focuses on resource management (separating waste at the source to increase value recovery rather than treatment), and aims at increasing efficiency by adopting the concept of circular economy and economy of sharing.

13 APPENDIX

Source code:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "8wd932"
deviceType = "Node_Mcu"
deviceId = "123456789"
authMethod = "token"
authToken = "123456789"
# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status =="lighton":
        print("led in on")
    else :
        print ("led is off")
  try:
 deviceOptions = {"org": organization, "type": deviceType, "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()
#Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
deviceCli.connect()
```

```
while True:
        #Get Sensor Data from DHT11
       time.sleep(5)
       ult_son=random.randint(0,80)
       weight=random.randint(0,100)
        lat = round(random.uniform(11.03, 11.50), 6)
        long = round(random.uniform(76.80, 76.90), 6)
        gps = str(lat) + str(',') + str(long)
        data = {'Ultrasonic' : ult_son, 'Weight' : weight , 'GPS' : gps}
        #print data
        def myOnPublishCallback():
           print ("Published Ultrasonic = %s Cm" %ult_son, "Weight:%s kg" %weight, "GPS: %s" %gps)
        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
        if not success:
           print("Not connected to IoTF")
        time.sleep(1)
        deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

GIT HUB LINK:

https://github.com/IBM-EPBL/IBM-Project-49437-1660818924

DEMOLINK:
https://drive.google.com/file/d/1uOSMj2ZCgi5S5IeN2UnpvB2DOpkhIT7L/view?usp=drivesdk