TEAM ID - PNT2022TMID43127

SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITAN CITIES

1. INDRODUCTION

1.1 Project Description

Garbage disposal has become a global concern. Because of the authorities' the trash containers are entirely full due to a lack of care and attention. As a result, it is up to us to think about what method we may employ to overcome this. Everything in India's future will be digital and reliant on the internet. The internet has become ingrained in our daily routines. The Internet of Things (IoT) and Cloud Computing are rapidly developing technologies. This is where machine and human communication takes place. The Internet of Things connects the things we use on a daily basis (IoT).

1.2 Project Purpose

Various major concerns, such as filthy environments, air pollution, and unhealthy lifestyles, are producing significant problem. The environment is harming people’s health. One application displays the status of the garbage can, while another displays the location of the garbage bin. Garbage pickup vehicle can get to their destination faster if they use the shortest path approach.

2. LITERATURE SURVEY

2.1 Existing problem

The municipal corporation’s community garbage can for causing a slew of health, environmental, and social problem. This might be due to a variety of factors, including inadequate planning. The positioning of dustbins across the city, as well as the collection method, are both

troublesome wastes created by the municipal corporation, as well as citizens who are unaware of how to properly utilize dustbin.

2.2 References

1) Gayanthika, W.A. L, G.K.C.D. Maduranga, A.I.S Silva, S.D.H.S. Wikramarathne, and R.M.I.S. Ranasinghe. “Smart dustbin for waste management." International Journal of Environmental Science and Development 10, no. 4 (2019): 118-121.

2) Housing, Urban, and MOBILITY STUDIES OTB. "Smart cities ranking of European medium-sized cities." (2018).

3) Meghana, KC, and K. R. Nataraj, "OT Based Intelligent Bin for smart cities," International Journal on Recent and Innovation Trends in Computing and Communication 4, no. 5 (2016): 225-229

4) Gupta, Suyog, and Pradeep Kumar, "Real Time Solid Waste Monitoring and Management System: A Case Study of Kanpur City," International Journal of Science, Environment and Technology 4 no. 2 (2015).

5) Bhor, Vikrant, Pankaj Morajkar, Maheshwar Gurar, Dishant Pandya, and Amol Deshpande, "Smart garbage management system."

2.3 Problem Statement Definition

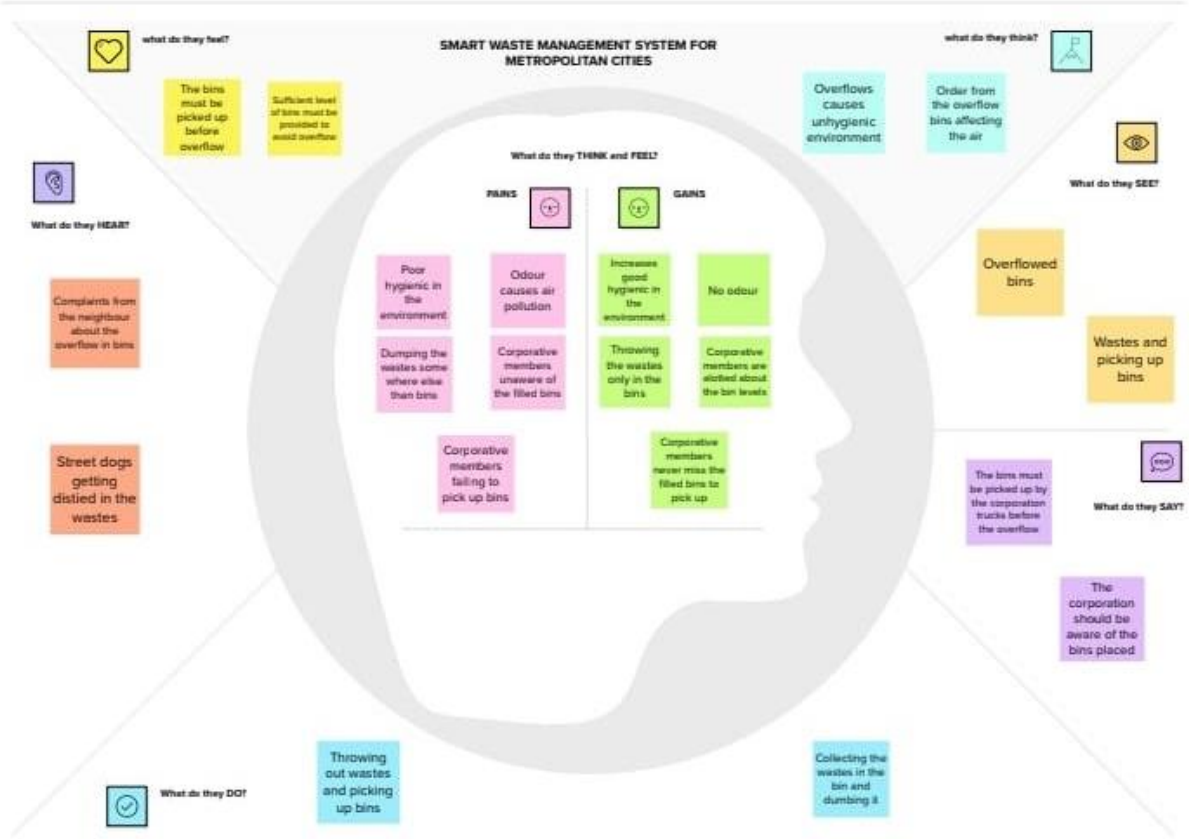
Internet of Things (IOT) expands in scope. To evaluate waste levels, the proposed garbage collection system uses data from trash cans in a Metropolitan area. Sensor data is sent to a server through the internet, where it is stored and processed.

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

Develop shared understanding and empathy

Summarize the data you have gathered related to the people that are impacted by your work. It will help you generate ideas, prioritize features, or discuss decisions.



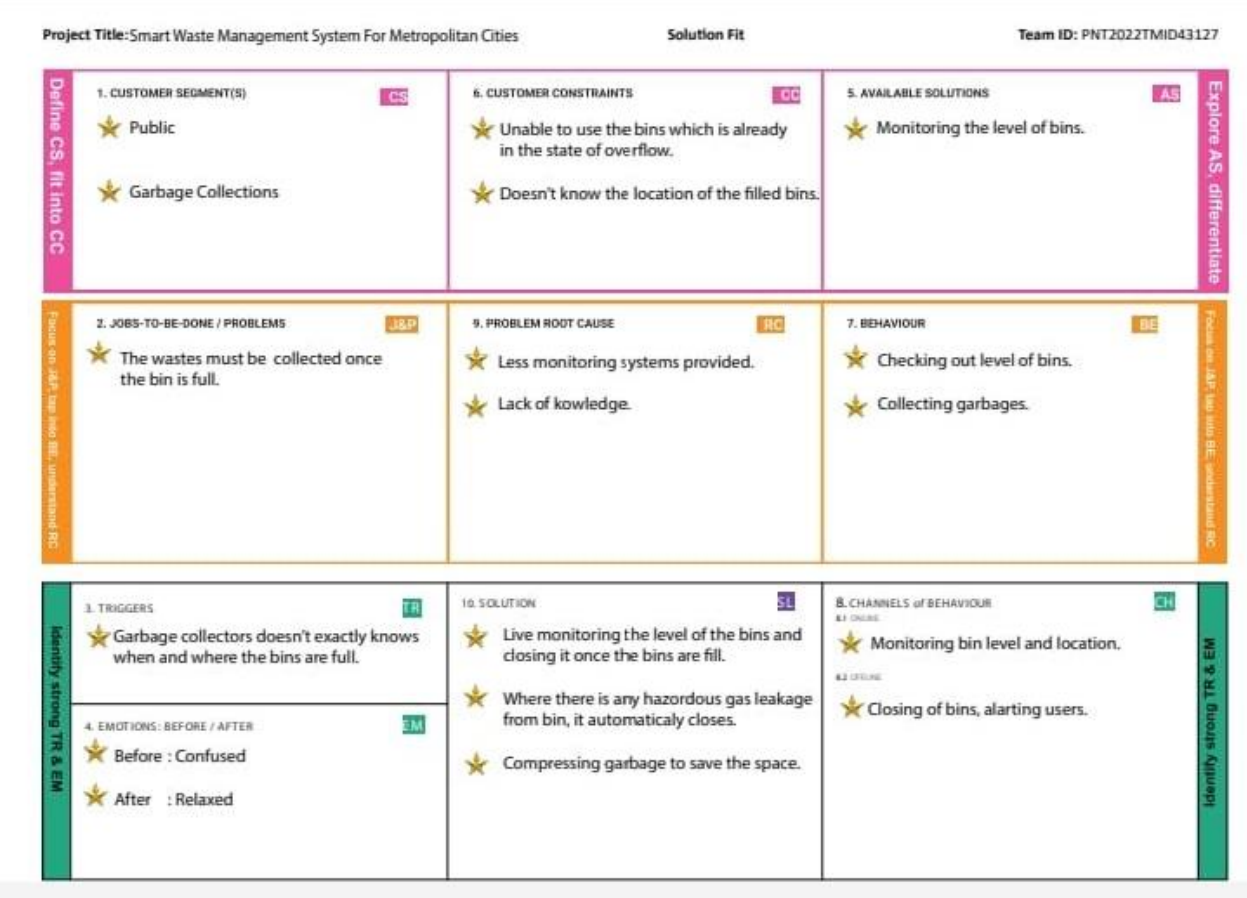
3.2 Ideation & Brainstorming

3.3 Proposed Solution

Proposed Solution:

S.NO	Parameter	Description
1.	Problem Statement (Problem to be solved)	• Smart Waste Management System for Metropolitan Cities
2.	Idea/Solution Description	• Monitoring the bin levels and notifying user once it is filled along with its location
3.	Novelty/Uniqueness	• Live monitoring the level of the bins and closing it once the bins are fill • When there is any hazardous gas leakage from bin, it automatically closes • Compressing garbage to save the place
4.	Social Impact/Customer Satisfaction	• Picking up bins after its filled • No overflow in bins
5.	Business Model (Revenue Model)	• The smart bin with gas sensor, magnetic sensor, ultrasonic sensor, compressor motor and GPS
6.	Scalability of the Solution	• The UI can notify the concern garbage collector • Closing of bins whether there is overflow or any sense of hazardous gas

3.4 Problem Solution Fit



4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR NO.	Functional Requirement (Epic)	Sub Requirement (Story/Sub- Task)
FR – 1	Visiting Website	Visiting website to check bin levels Visiting website to check air purity rate Visiting website to check overflowing bin locations
FR – 2	Searching	Searching different bins in different areas Searching for overflowing bins
FR – 3	Notifying	Notifying user whenever the bin is full

4.2 Non – Functional Requirement

Non – functional Requirements:

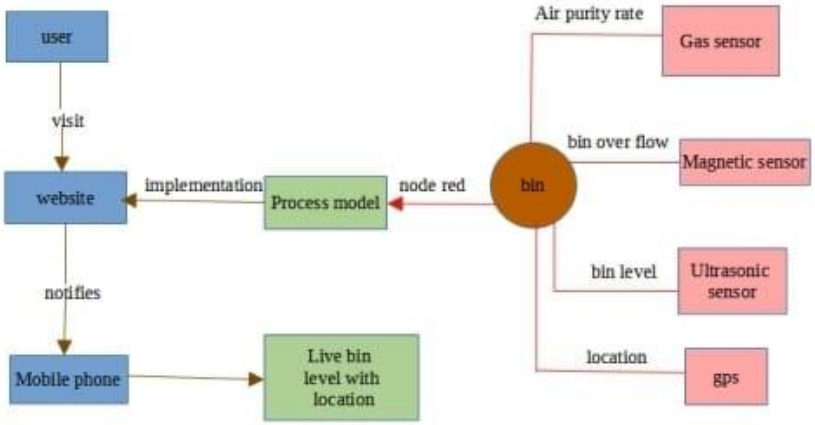
Following are the non – functional requirements of the proposed solution.

FR NO.	Non – functional Requirement	Description
NFR - 1	Usability	The UI should be simple enough for everyone to understand
NFR - 2	Security	The website must be secure enough to trust by the users
NFR - 3	Reliability	The UI should be able to withstand any errors in the data
NFR - 4	Performance	The live bin level is shown in the specific bin section
NFR - 5	Availability	The UI should respond to the users within 2 seconds
NFR - 6	Scalability	Different bins according to location can be viewed

5.PROJECT DESIGN

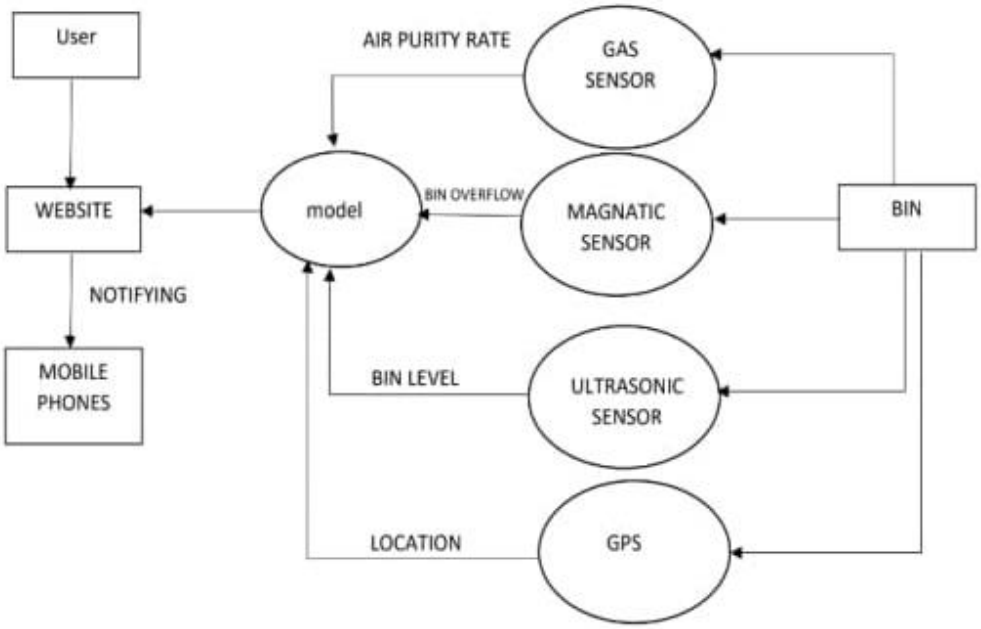
5.1 Data Flow Diagram

Data Flow Diagrams:



5.2 Solution &Technical Architecture

Solution Architecture:



5.3 User Stories

User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story/ Task	Acceptance Criteria	Priority	Release
Garbage collector	Visit website	USN – 1	As a user, I can visit website	I can visit the website	high	Sprint - 1
		USN - 2	As a user, I can monitor the bin level of different bins	I can get notification of overflowing bins	high	Sprint – 1
		USN – 3	AS a user, I can monitor the bin's location	I can get location of bins	high	Sprint – 1
		USN – 4	As a user, I can monitor air purity rate	I can get level of air purity in bins	low	Sprint - 2
Municipal officer	Visit website	USN – 5	As a user, I can monitor bin level and location of different bins	I can intimidate garbage collectors about bin levels	high	Sprint - 1

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Dashboard	USN-1	As a User , I can view the location of the bin	2	High	Deenadhayalan.S
Sprint-1	Dashboard	USN-2	As a user I can monitor the garbage unit and level	1	High	Deenadhayalan.S
Sprint-2	Dashboard	USN-3	As a User , I can view the location of the bin	2	High	Jayaselvi.M
Sprint-2	Dashboard	USN-4	As a user I can monitor the garbage unit and level	1	High	Jayaselvi.M
Sprint-3	Dashboard	USN-5	As a User , I can view the location of the bin	2	High	Muppidathi.S
Sprint-3	Dashboard	USN-6	As a user I can monitor the garbage unit and level	1	High	Muppidathi.S
Sprint-4	Dashboard	USN-7	User can view all the bin's data in single dashboard map	1	High	Senthamilselvi.R

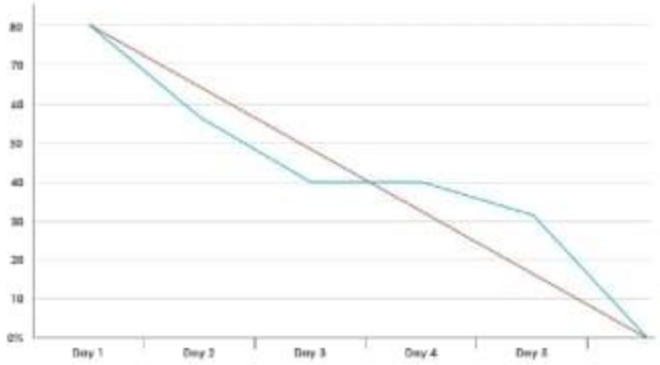
6.2 Sprint Delivery Schedule

Project Tracker, Velocity & Burn down Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	28 Oct 2022	20	18 Nov 2022
Sprint-2	20	6 Days	30 Oct 2022	04 Nov 2022	20	18 Nov 2022
Sprint-3	20	6 Days	06 Nov 2022	12 Nov 2022	20	18 Nov 2022
Sprint-4	20	6 Days	13 Nov 2022	18 Nov 2022	20	18 Nov 2022

6.3 Report from Jira

Burndown Chart:



7. CODING & SOLUTIONING

(Explain the features added in the project along with code)

Bin Level:

BIN 1:

```
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "x3lifo",
        "typeId": "Bin_1",
        "deviceId":"1234"
    },
```

```
    "auth": {
        "token":"12345678"
    }
}

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()

while True:
    level=random.randint(0,10)
    weight=random.randint(0,10)

myData={'name':'Bin_1','lat':13.092677,'lon':80.188314,'Level':level,'Weight':weig
ht  }
    client.publishEvent(eventId="status", msgFormat="json", data=myData,
qos=0,onPublish=None)
    print ("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

BIN 2:

```
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "x3lifo",
        "typeId": "Bin_2",
        "deviceId":"1234"
    },
    "auth": {
        "token":"12345678"
    }
}

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()

while True:
    level=random.randint(0,10)
    weight=random.randint(0,10)
    myData={'name':
'Bin_2','lat':14.092677,'lon':81.188314,'Level':level,'Weight':weight  }
```

```
    if weight == 10:
     print ('ALERT !! Weight is HIGH')
    if level == 10:
     print ('ALERT !! Level is HIGH')
    client.publishEvent(eventId="status", msgFormat="json", data=myData,
qos=0,onPublish=None)
    print ("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

BIN 3:

```
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "x3lifo",
        "typeId": "Bin_3",
        "deviceId":"1234"
    },
    "auth": {
        "token": "12345678"
    }
}
```

```
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()


while True:
    level=random.randint(0,10)
    weight=random.randint(0,10)
    myData={  'name': 'Bin_3', 'lat': 15.092677, 'lon': 79.188314 ,'Level':level, 'Weight':weight  }
    if weight == 10:
     print ('ALERT !! Weight is HIGH')
    if level == 10:
     print ('ALERT !! Level is HIGH')
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,onPublish=None)
    print ("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

ULTRASONIC:

```
import wiotp.sdk.device
import time
```

```
import random
myConfig = {
    "identity": {
        "orgId": "x3lifo",
        "typeId": "Ultrasonic",
        "deviceId":"1234"
    },
    "auth": {
        "token": "12345678"
    }
}

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()

while True:
    level=random.randint(0,400)

    if (level<=100):
     myData={'Level':level,"Alert":"High Alert!!!,Trash bin is about to be full"}
     client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
```

```
    print("Published data Successfully: %s", myData)


   if ((level>150) and (level<250)):
    myData={'Level':level,"Alert":"Warning!!,Trash is about to cross 50% of bin level"}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)


   if ((level>250) and (level<400)):
    myData={'Level':level,"Alert":"Bin is available"}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)


   client.commandCallback = myCommandCallback
   time.sleep(2)
client.disconnect()
```

8. TESTING

8.1 Test Cases :

Test case ID	Feature Type-Bin Level	Component	Test Case Scenario	Pre-Requisite	Availability	Test Condition	Expected Result	Actual Result	Status	Comments	Accessed By
Test case 1	Empty	Ultrasonic Sensor	When Bin is empty	Ultrasoncic sensor , Garbage Bins	Bin is accessible to users	Bin Level == 0	Displays Bin level and space left	Working as expected	Pass		User
Test case 1	Accessible	Ultrasonic Sensor	When bin level is below 50 %	Ultrasoncic sensor , Garbage Bins	Bin is accessible to users	Bin Level < 50	Displays Bin level and space left	Working as expected	Pass		User
Test case 3	Accessible	Ultrasonic Sensor	When bin level is above 50	Ultrasoncic sensor , Garbage Bins	Bin is accessible to users and the admin gets warning about the bin level	Bin Level > 50	Displays Bin level and space left	Working as expected	Pass		User
Test case 4	Accessible	Ultrasonic Sensor	When bin level is below 75 %	Ultrasoncic sensor , Garbage Bins	Bin is accessible to users and the admin gets warning about the bin level	Bin Level < 75	Displays Bin level and space left	Working as expected	Pass		User
Test case 5	Limit exceeded	Ultrasonic Sensor	When bin level is above 75 %	Ultrasoncic sensor , Garbage Bins	Bin is not accessible to the users, the admin recieves High alert and seals the the bin to avoid overflow.	Bin Level > 75	Displays Bin is FULL and Seals the bin.	Working as expected	Pass	The system starts to sense the level once the Bin is emptied partially or fully	User/Admin

Testcases Testscearnios

8.2 User Acceptance Testing:

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	3	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	78

9. RESULTS

9.1 Performance metrics:

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

10. ADVANTAGES AND DISADVANTAGES

10.1 Advantages:

- Poor hygienic in the environment
- Odour causes air pollution
- Dumping the wastes from somewhere else then bins
- Corporative members unaware of the filled bins
- Corporative members failing to pick up the bins

10.2 Disadvantages:

- Increases good hygienic in the environment
- No odour
- Throwing the wastes only in the bins
- Corporative members are allotted about the bin levels
- Corporative members never miss the filled bins to pick up

11. CONCLUSION

The population increases day by day and generate million tons of wastes per year. City administration, municipalities and wastes management organization in different countries phases the challenge to provide efficient and effective system to collect dispose off properly, and recycle the waste, keeping health standards and environment friendliness. The smart waste management system collects the wastes in proper time, disposes and recycles in the proper way.

12. FUTURE SCOPE

Servers can be maintained at the WMC‟s to collect the information from the dustbin instead of a phone number. Each level of the dustbin can be recorded instead of just recording the filled level. Whenever a dustbin is filled, a GPS locator can be used to locate another dustbin which is nearer to the user to make the waste disposal convenient to prevent the waste lying on the open streets.

13. APPENDIX

13.1 GitHub Account

https://github.com/IBM-EPBL/IBM-Project-37985-1660366782.git

13.2 Project Demo Link

https://drive.google.com/file/d/1PjQZx9_CuzUZA6hqlgj6ypH49G4RM9dq/view?usp=sharing