GOVERNMENT COLLEGE OF ENGINEERING

CHETTIKARAI, DHARMAPURI

636704

Affiliated to Anna university, Chennai

SMART WASTE MANAGEMENT FOR METROPOLITAN CITIES – IOT BASED

IBM NALAIYATHIRAN

Document Title

TITLE	Smart waste management for metropolitan cities
DOMAIN NAME	INTERNET OF THINGS
TEAM ID	PNT2022TMID41308
TEAM LEADERNAME	G.VIHASHNI
TEAM MEMBER NAME	S.ISHWARYA T.PRIYADHARSHINI S.SNEKA
MENTOR NAME	Dr. DINESH G

7. **CODING & SOLUTIONING (Explain the features in the project along with code)**

    7.1 Feature code 1

    7.2 Feature code 2

8. **TESTING**

    8.1 Test Cases

    8.2 User Acceptance Testing

9. **RESULTS**

    9.1 Performance Metrics

10. **ADVANTAGES &DISADVANTAGES**

11. **CONCLUSION**

12. **FUTURE SCOPE**

13. **APPENDIX**

    13.1 Source Code

    13.2 GitHub &Project Demo Link

1.INTRODUCTION

1.1 Project Overview

As the population is increasing the solid waste is also increasing in metropolitan cities and waste management has become a global concern. We need to take right decision in order to manage this overflowing garbage. Mainly there are three process they are collecting, segregating and decomposing. Here, we are going to deal the problem in the collecting process due to climate change. Developing the python script for publishing the location (latitude and longitude) and weather data along with bin values to the IBM IOT platform and alerting the authority when the bin level and weight crosses its threshold level. The garbage level of the bins can be monitored through web App

1.2 Purpose

At present solid waste management is a major concern in the metropolitan cities of the developing and developed countries. As the population is growing, the garbage is also increasing. This huge unmanaged accumulation of garbage is polluting the environment, spoiling the beauty of the area and also leading to the health hazard. Due to climate changes the garbage collectors facing difficulties, by this process the authority can easily manage the situation and plan accordingly.

2. LITERATURE SURVEY

2.1 Existing problem

PAPER 1:

Authors have proposed a useful garbage collection through shortest path semi static and dynamic routing for controlling traffic that is created by the trucks which carrying the waste. Here they have used two layers, in which upper layer is semi static shortest path routing model. This layer contains the waste collection terrain for each city sector. Lower layer dynamic shortest path routing model handles the dynamic requirements of real time routing in case of emergency. Waste routing was achieved by the Ant colony system algorithm (ACS), in turn to group the garbage bins allocation in the form of clusters, they have used K-means algorithm. By using above mechanisms the authors effectively measured distance covered, time spent, fuel consumption and the quantity of solid waste collected. They have mentioned the future work will be in the area of time critical scheduling, where once the waste bins are full and need to be emptied at the earliest by available waste collecting vehicles.

PAPER 2:

Authors have illustrated the Top-k query based dynamic scheduling for smart city garbage collection. They introduced Top-k query to denote the number of filled bins in turn to begin dynamic scheduling. Authors have used adaptive large neighbourhood search algorithm to determine the cost optimal routes for the trucks to empty the bins. They used roll on-roll off routing mechanism to help several dumping services to collect large amount of garbage from the location of

shopping malls and construction sites. The demerit of this model is, in dynamic scheduling depending on the k-value, CPU overhead cost is high. The future work they mentioned is dynamic routing model depends on fuzzy demands. Here the customer acts as variables of fuzzy.

PAPER 3:

The waste collection as a potential Internet of things service which exploits robustness and cost efficiency of different types of fleets. Authors have used robust dynamic routing algorithm to find the shortest path, by this they achieved cost efficiency. They used Android app for truck navigation, GPS to track the truck location, RFID to identify the certain bins and actuators to lock the lid of the bin when bin gets full to avoid the overflow of the garbage. Here they used two types of trucks: High Capacity Trucks (HCT) to transport waste from depots to dump yard and Low Capacity Trucks (LCT) to transport waste from dump yard to depots.

PAPER 4:

The authors aim at inventory routing for dynamic waste collection. Here they mainly focus on the problems of scheduling of emptying the containers and to take quick decision on selection of nearest route for the vehicles. By this the garbage collection costs can be minimized and at the same time customer satisfaction can also be improved. Here they used heuristic approach to deal with the dynamic and stochastic nature of the problem. Here they considered two policies viz. Sequential Rigging Optimization (SKO) and Hierarchical knowledge gradient (HKG). HKG quickly identify the optimization areas within the network space and then use SKO for communication.

PAPER 5:

Authors aim at designing an integer programming model. The main goal of this model is to make the decision maker's job easy in two important aspects. Selecting the location of the dust bin and defining the capacity of the dust bin to be placed in each collection sites. Authors have proposed a two phase heuristic approach to solve the above problem. Authors have undergone the difficulty of where to place the dust bin in collection sites of the urban waste management system.

PAPER 6:

Authors have proposed a system, where multiple dustbins are located throughout the city. These dustbins are embedded with low cost devices and unique ID will be given for every dustbin in the city. This will help in tracking the level of garbage in each bin. In this system, these bins are connected to the internet to get the real time information of the smart bins. By implementing this system authors have achieved cost reduction, resource optimization, real time data transmission and effective use of smart dustbins. They have mentioned the future work as; the system can be implemented with time stamps.

PAPER 7:

Authors have developed a model that identifies the level of garbage in the bin. By using wireless sensor networks and embedded Linux board it send message for cleaning of the bin to the authorized person. This system gives a web interface to the cleaning authority so that they monitor and clean the garbage bin. Here they used Raspberry Pi as an embedded Linux board, it makes communication to be

distributed to sensor nodes located in the sensor area via ZigBee protocol and itself act as a coordinated node in the wireless sensor network.Aim of coordinated node is to gather the factors such as level of the bin and odor and transmit the wireless message. The smart waste bin display a message for emptying the waste bin when the waste bin is about to fill through the coordinator node.

PAPER 8:

Authors have proposed an integrated system combined with an incorporated system of Radio Frequency Identification (RFID), General Packet Radio Service (GPRS), Geographic Information System (GIS) and web camera. Built in RFID was used mechanically to fetch all types of client data and dustbin data from RFID tag, GPS would give the locality data of the truck available. Through GPRS communication system the entire information of the centre server will be automatically updated. To obtain real time truck tracking and monitoring information of the system authors have used an integrated system which consists of RFID, GPS, GPRS, GIS and web camera. The future work they have mentioned was analysing storage information by authority for garbage management. In order to achieve this, one need to concentrate on vehicle management, route management, dumping site selection etc. To deal with the problems of waste management and also a system to work in a real time.

PAPER 9:

Authors have used RFID and sensor model. This model mainly gives the solution to the automatic garbage recognition, weight and identification of the stolen bins. RFID waste tag read the data without really seeing it. Also, waste tags are capable to store a large amount of information easily and more rapidly when

compare to bar codes. To lower the waste tag price authors have selected a 13.56 MHZ solution. When the driver of the garbage collection truck complete his work shift, personal digital assistant then sends all the information to a SQL back end server for storing and processing the garbage information in real time. The data related to garbage is carried out via WIFI connection and the Internet.

PAPER 10:

Authors have proposed a system that alerts municipality when the bin is about to fill so that they empty the bin on time. This Smart garbage bin model separates five types of plastic resins by using Near Infrared (NIR) spectroscopy and rest of the biodegradable waste will be used to produce the biogas. The Beer-Lambert law is the inherent nature of NIR spectroscopy. The law states that when absorbance increases the thickness of the sample is also increases; it means both are directly proportional. The drawbacks of NIR spectroscopy process are: It will not work if garbage is covered in black plastic and will not identify pet bottles if they have covered by plastic caps. Using GSM technology this model automatically emit a notification when the dustbin is about to fill and NIR reflectance spectroscopy method helps to separate and take out plastic piece from the wastage to generate some renewable energy.

2.2 References

PAPER 1- Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammad Aledhari, Moussa Ayyash, “Internet of Things: A survey on enabling technologies, Protocols and applications.”IEEE communications surveys and tutorials.1553-877x © (2015).

PAPER 2- Theodoros Vasileios Anagnostopoulos and Arkady Zaslavsky, “Effective waste collection with shortest path semi-static and dynamic routing.” LNCS vol.8638, pp. 95-105, Springer (2014).

PAPER 3- Theodoros Vasileios Anagnostopoulos and Arkady Zaslavsky, Alexey Medvedev, Sergei Khoruzhniov,“Top-k Query based dynamic scheduling for IOT-enabled smart city waste collection .” International conference on mobile data management (ICMDM)(2015)

PAPER 4- Theodoros Vasileios Anagnostopoulos and Arkady Zaslavsky, Alexey Medvedev, “Robust waste collection exploiting cost efficiency of IOT potentiality in smart cities.” International conference on recent advances in internet of things (ICRIOT)(2015)

PAPER 5- Giapaolo Ghiani, Demetri Lagana, Emanuele Manni, Chefi Triki, “Capacitated location of collection sites in an urban waste management system.” Elsevier (2012)

PAPER 6- Giapaolo Ghiani, Demetri Lagana, Emanuele Manni, Chefi Triki, “Capacitated location of collection sites in an urban waste management system.” Elsevier (2012)

PAPER 7- Kusum Lata, Shri S K Singh, “IOT based smart waste management system using Wireless Sensor Network and Embedded Linux Board.” International journal of current trends in engineering and research (IJCTER)(2016)

PAPER 8- Md.Shafiqul Islam, Maher Arebey, M.A.Hannan, Hasan Basri “Overview of solid waste bin monitoring and collection system,” International conference on innovation, management and technology research (ICIMTR) (2012)

PAPER 9- Belal Chowdhury, Morshed U Chowdhury, “RFID-based real time smart waste management system.” Australian telecommunications networks and applications conference (ATNAC) (2007)

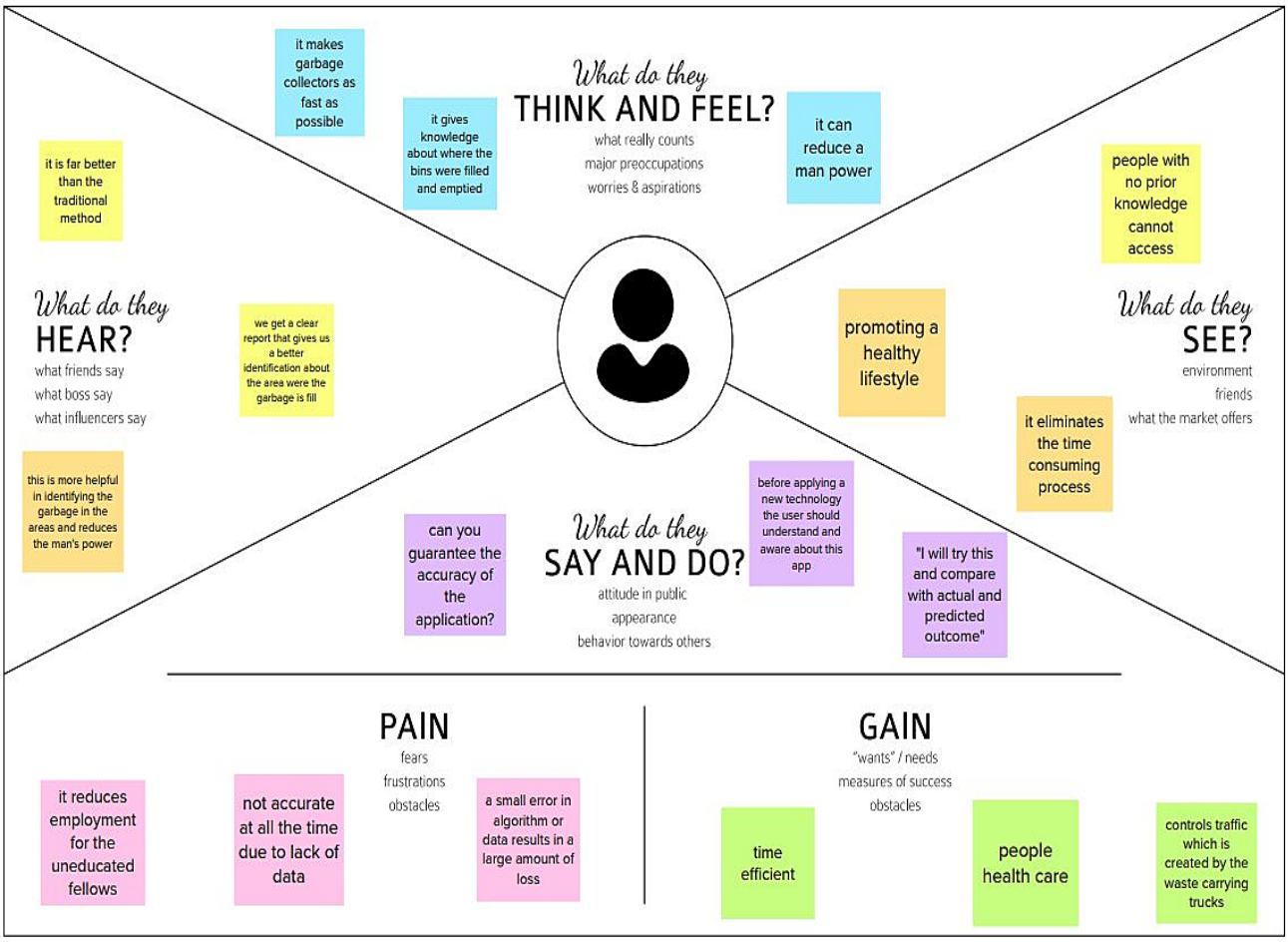
PAPER 10- Shubham Thakker, R. Narayanamoorthi, “Smart and wireless waste management.” International conference on Innovations in information embedded and communication systems (ICIIECS) (2015)

2.3 problem definition

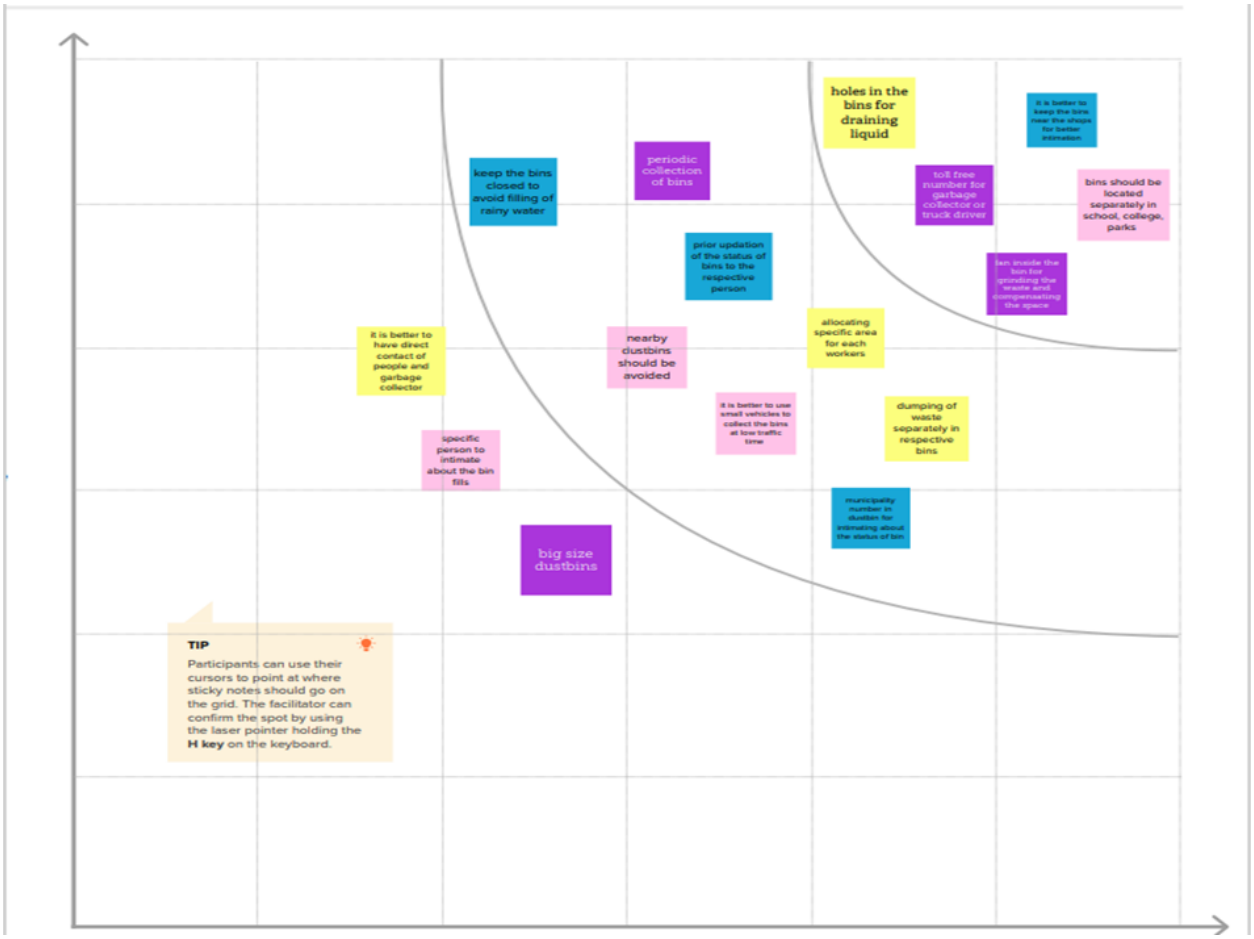
Due to climate changes collection of garbage from the bins get affected, to overcome this problem the “SMART WASTE MANAGEMENT FOR METROPOLITAN CITIES” under the technology IOT has been developed in this system the authority get alert when the bin crosses its threshold level and the location of the bin is identified using GPS location tracker so that the authority can plan the collection process accordingly along with the location the weather condition also given as a data.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



3.3 Proposed Solution

1. Problem Statement (Problem to be solved)

As the population is growing, the garbage is also increasing. This huge unmanaged accumulation of garbage is polluting the environment, spoiling the beauty of the area and also leading to the health hazard to overcome these, smart waste management for metropolitan cities are introduced.

2. Idea / Solution description

Garbage level detection in bins. Getting the weight of the garbage in the bin. Alerts the authorized person to empty the bin whenever the bins are full. Garbage level of the bins can be monitored through a web App. We can view the location of bins in the web application by sending GPS location from the device.

3. Novelty / Uniqueness

A system that detects the level of garbage in the dustbins with the help of sensor systems and send this information to the authorized control room. Weight sensor determines the weight of the garbage in the dustbin and (IR) sensor is used to detect the waste level in the dustbins.

4. Social Impact / Customer Satisfaction

Municipality authority is the customer whose work will get easy and they get satisfied if they know to use this system the garbage collector can complete their work within the time without leaving any place. The workload for the garbage collector and authority may get reduced 75% .

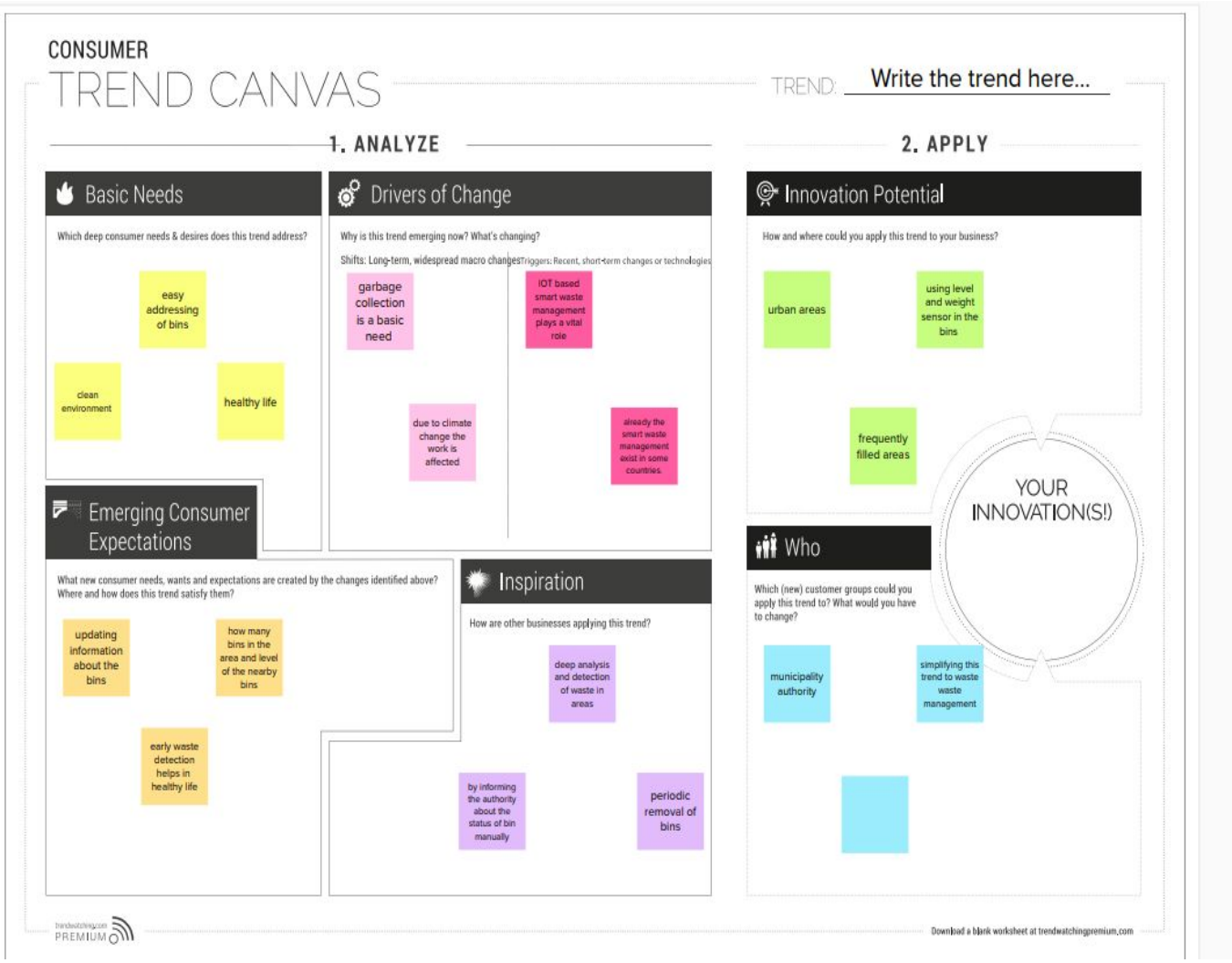
5. Business Model (Revenue Model)

The cost for the truck fuel get reduced, man power reduced so, wages for them reduced but the process is cost efficient for implementation and everyone should get training for better performance.

6. Scalability of the Solution

The process can also denote the climate change in the area, what is the average time the bin takes to fill, denoting the weight of liquid and solid separately, percentage of biodegradable and non-biodegradable waste

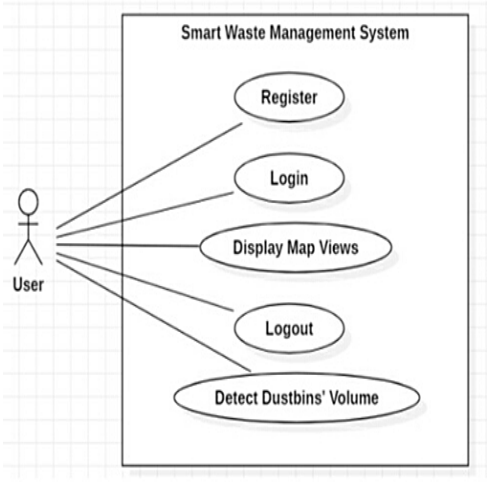
3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

1. Communicate and exchange information to provide server for user
   - To detect level of bins
   - Weight of bins
   - Alert the authorized person
2. Live asset monitoring
   - Track the bins
   - Waste collection Bins fill level monitor
3. User requirement
   - Promote health care service
   - Pollution prevention
   - Improve efficiency
4. Mandatory
   - Reduce traffic and fuels
   - Reduce running cost
   - Safe disposal of waste
   - Avoid health hazard
5. Captured in use case (image)



6. Testing (The test case status is pass if)

- Empty(T1) ⟶null
- Medium(T2) ⟶intermediate level
- Nearly full(T3) ⟶above the threshold level
- Full (T4) ⟶maximum level
- Threshold crossed(T5) ⟶spill over

4.2 Non-Functional requirements

1. Usability

High usability of user experience design for user, which is to solve the problem efficiently.

2. Security

The system can only accessed by authorized person.

3. Reliability

Identifying the problem and providing clear solution for it, by accurate real time data collection.

4. Performance

The performance should effective and efficient.

5. Availability

The existing system affects the garbage collector due to climate change and indirectly affects the people health care.

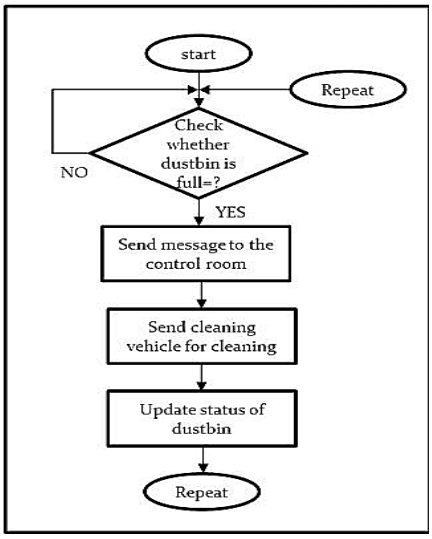
6. Scalability

The system should have upgradable feature

5. PROJECT DESIGN

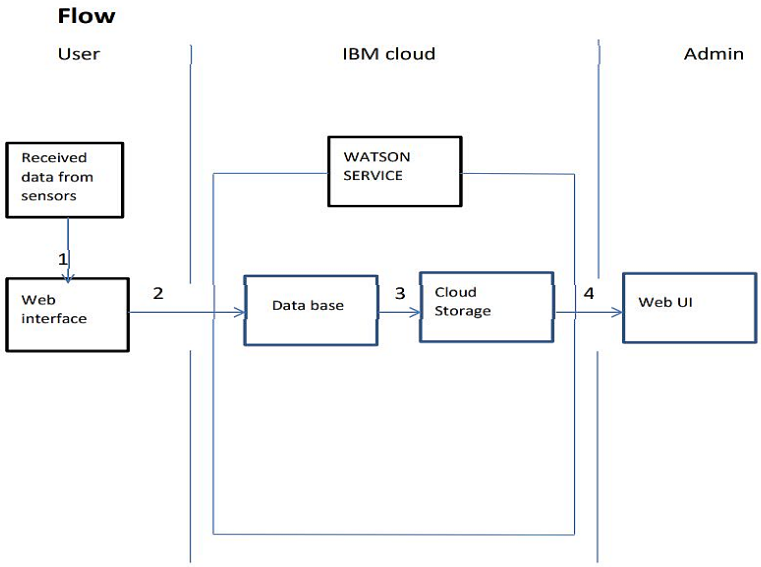
5.1 Data Flow Diagrams

Data Flow Diagram

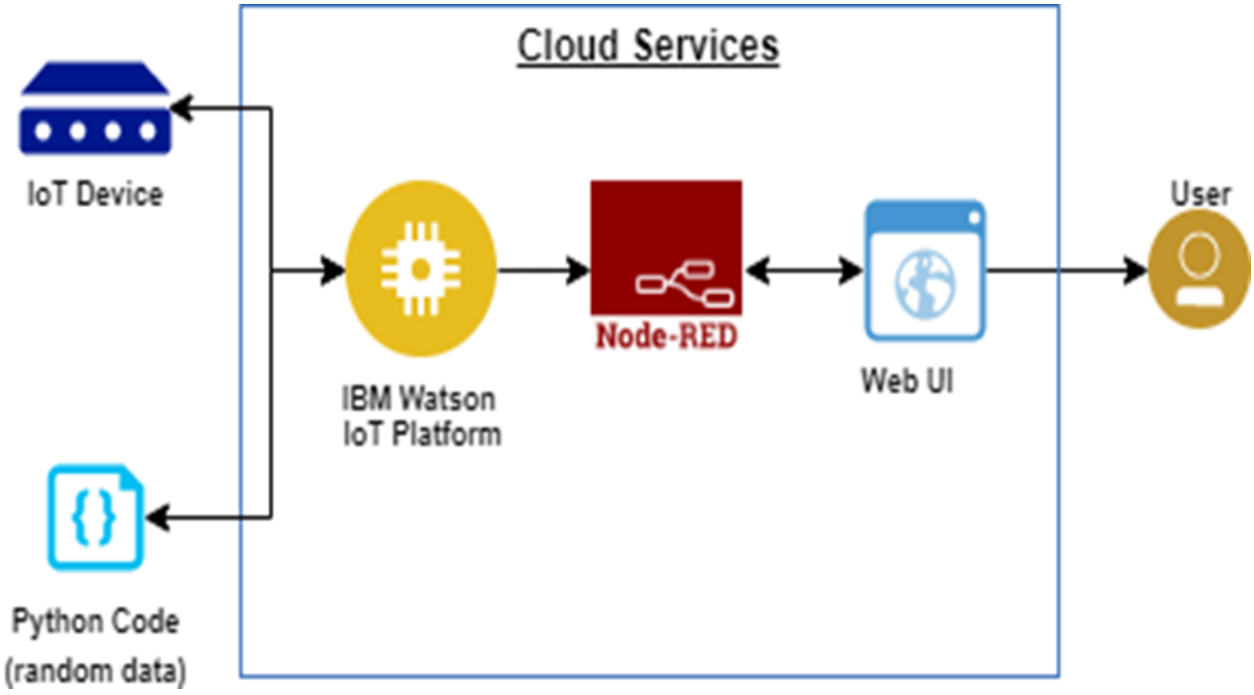


Summary

This code pattern explains how to build an IOT based smart waste management for the metropolitan cities with predefined value.



5.2 Technical architecture



1. Feed the data from the sensor placed in the bins tothe web interface.
2. The data will display in the web page of theauthority (user).
3. The collected data is sent to the data base, where the collected data and predefined data are checkedand monitored if the data exceeds the predefined data the control signalsend to the admin.
4. The data is providedto the cloud service and stored
5. The authority monitors the web page continuouslyto collect the data and send the alert to the authority.

Component and technology

S.No	Component	Description	Technology
1.	User Interface	user interacts with Web UI, Mobile App, Chatbox	Node red
2.	Application Logic-1	Random data set as threshold	Java / Python
3.	Application Logic-2	Cloudservice	IBM Watson cloud service
4.	Application Logic-3	Node red and web interface	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	Random data
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storageor Other Storage Service or Local Filesystem
8.	External API-1	Security purpose and to accessthe system	IBM Weather API,etc.
9.	External API-2	To access the system	Aadhar API,etc.
10.	Machine LearningModel	To provide the data	Object RecognitioModel, etc.
11.	Infrastructure (Server/ Cloud)	Application Deployment onLocal System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Application Characteristics

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Sensors(level, weight)	Random data in python script
2.	Security Implementations	API keys, weather APIand IBM cloud and Watsonaccount	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Upgrade	IBM cloud
4.	Availability	The app contains the data indicates the level and weight	Sensors, python script
5.	Performance	The system continuously update the data	Mobile app, web UI,Chatbox

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 by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access easily to go dashboard	High	Sprint-1
	Interface	USN-6	As a user, the interface should be user friendly manner	I can access the application easily	High	Sprint-1
Customer (Web user)	Dashboard	WUSN-1	As a Web user, I can register & access the specific information (map, level of bins, weight, etc…)	I can easily monitor the truck location	High	Sprint-1
Customer Care Executive	View manner	CCE-1	As a customer care, I can view the data in graphical representation	I can easy to view the status of the truck location, levels of bins	High	Sprint-1
	Detector	CCE-2	As a customer care, I can detect the bins (volume)	I can detect the bins (volume) easily	High	Sprint-1
	Monitor	CCE-3	As a customer care, I can monitor the truck location	I can easily monitor the location of the truck	Medium	Sprint-1
Administrator	Risk Tolerant	ADMIN-1	As an Administrator, who can handle the system update & take care of the application	Admin should monitor the system properly	High	Sprint-2

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the applicationby entering my email, password, and confirming my password.	2	High	Vihashni. G
Sprint-1		USN-2	As a user, I will receive confirmation emailonce I have registered for the application	1	High	Ishwarya. S
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	Priyadharshini. T
Sprint-3		USN-4	As a user, I can register for the application through Gmail	2	Medium	Sneka. S
Sprint-4	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Vihashni. G

6.2 Sprint Delivery Schedule

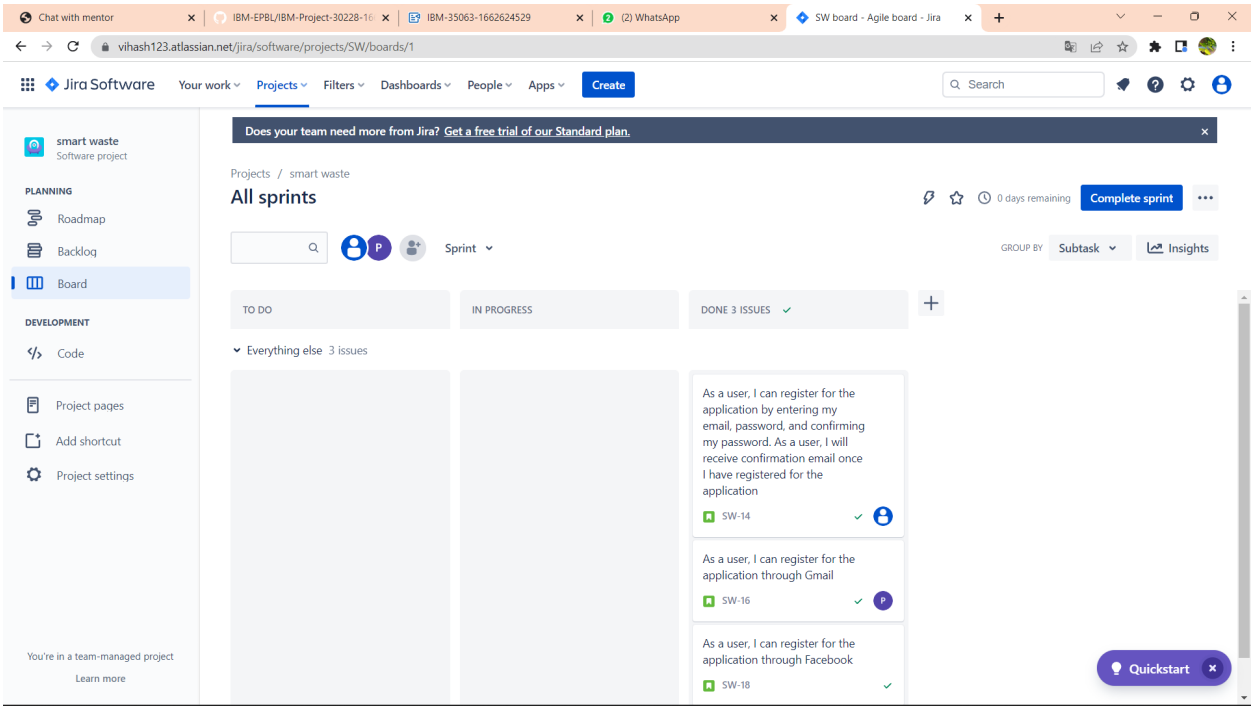
Sprint	Total StoryPoints	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	30	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	40	11 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	50	16 Nov 2022

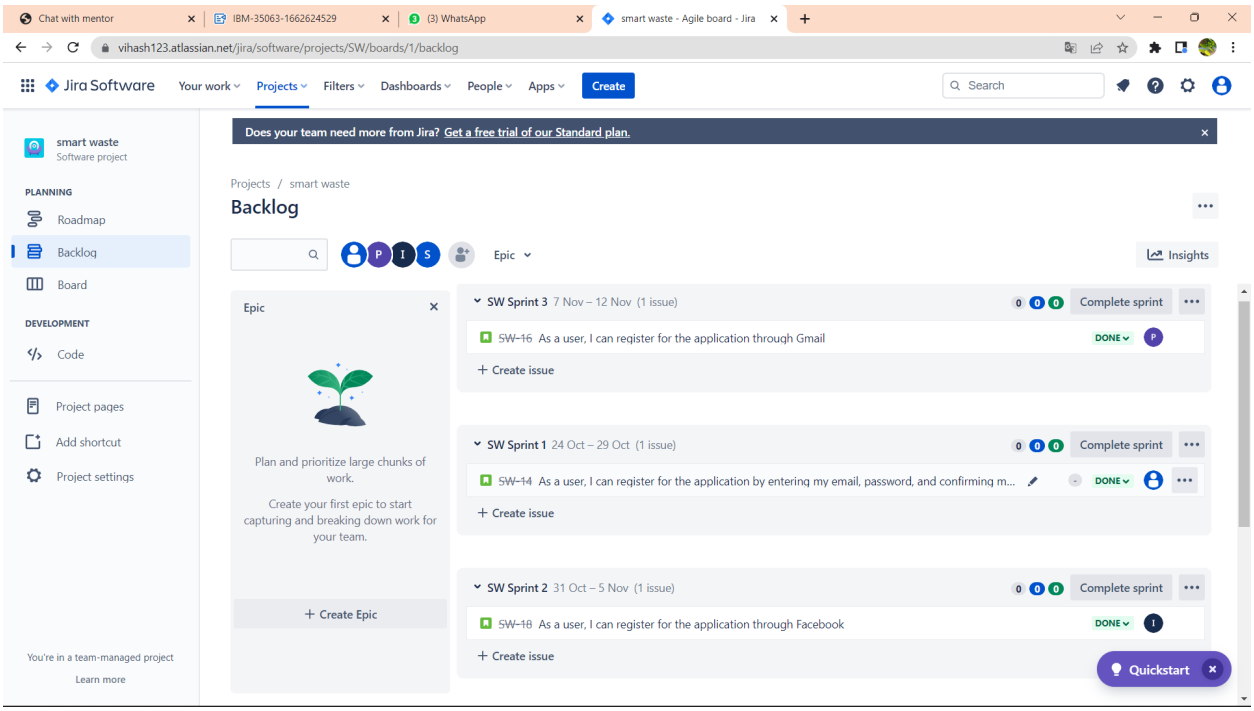
Velocity:

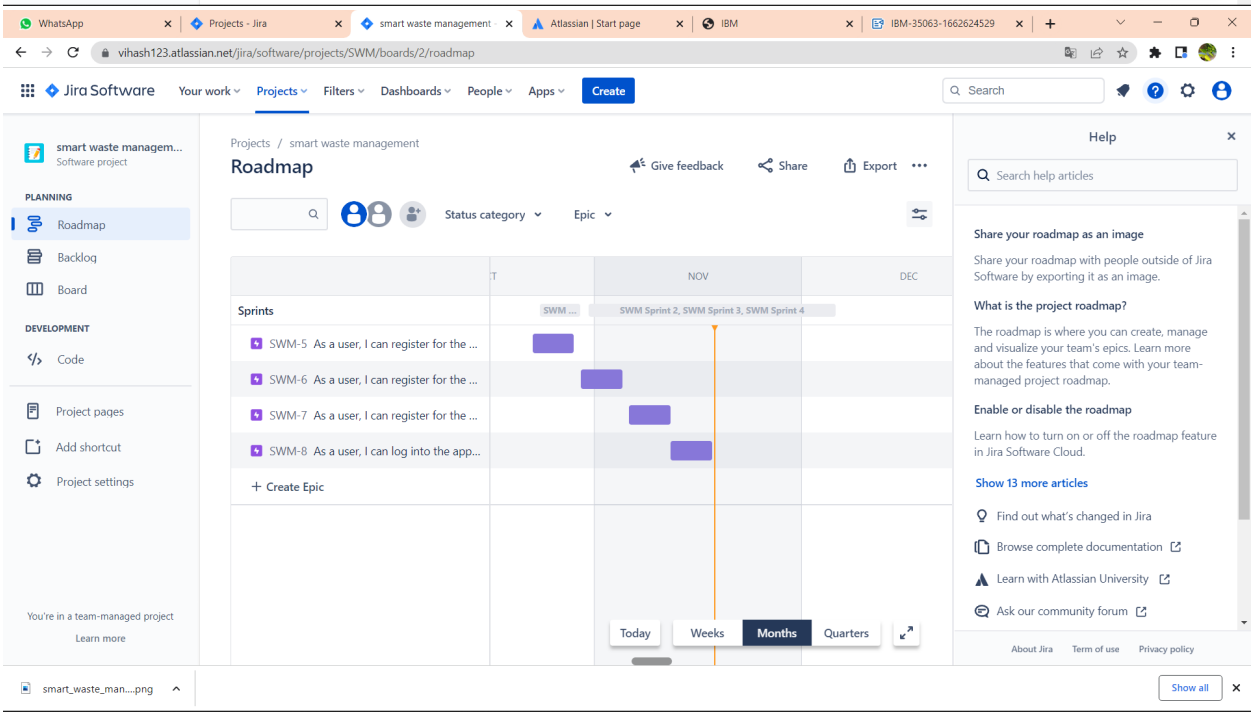
Imagine we have a10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let’s calculate the team’s average velocity (AV) per iteration unit (story points per day)

BURNDOWN

6.3 REPORTS FROM JIRA




Chat with mentor
IBM-35063-1662624529
(3) WhatsApp
smart waste - Agile board - Jira
vihash123.atlassian.net/jira/software/projects/SW/boards/1/backlog
Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
smart waste
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
Does your team need more from Jira? Get a free trial of our Standard plan.
Projects / smart waste
Backlog
Epic
Insights
Epic
Plan and prioritize large chunks of work.
Create your first epic to start capturing and breaking down work for your team.
+ Create Epic
SW Sprint 3 7 Nov – 12 Nov (1 issue)
Complete sprint
SW-16 As a user, I can register for the application through Gmail
DONE
+ Create issue
SW Sprint 1 24 Oct – 29 Oct (1 issue)
Complete sprint
SW-14 As a user, I can register for the application by entering my email, password, and confirming m...
DONE
+ Create issue
SW Sprint 2 31 Oct – 5 Nov (1 issue)
Complete sprint
SW-18 As a user, I can register for the application through Facebook
DONE
+ Create issue
Quickstart


WhatsApp
Projects - Jira
smart waste management
Atlassian | Start page
IBM
IBM-35063-1662624529
vihash123.atlassian.net/jira/software/projects/SWM/boards/2/roadmap
Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
smart waste managem...
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
Projects / smart waste management
Roadmap
Give feedback
Share
Export
Status category
Epic
NOV
DEC
Sprints
SWM ...
SWM Sprint 2, SWM Sprint 3, SWM Sprint 4
SWM-5 As a user, I can register for the ...
SWM-6 As a user, I can register for the ...
SWM-7 As a user, I can register for the ...
SWM-8 As a user, I can log into the app...
+ Create Epic
Today
Weeks
Months
Quarters
Help
Search help articles
Share your roadmap as an image
Share your roadmap with people outside of Jira Software by exporting it as an image.
What is the project roadmap?
The roadmap is where you can create, manage and visualize your team's epics. Learn more about the features that come with your team-managed project roadmap.
Enable or disable the roadmap
Learn how to turn on or off the roadmap feature in Jira Software Cloud.
Show 13 more articles
Find out what's changed in Jira
Browse complete documentation
Learn with Atlassian University
Ask our community forum
About Jira
Term of use
Privacy policy
smart_waste_man....png
Show all

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature code:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
from geopy.geocoders import Nominatim
geolocator=Nominatim(user_agent="geoapiExercises")

#Provide your IBM Watson Device Credentials
organization = "pb6xw8"
deviceType = "efgh"
deviceId = "1234"
authMethod = "token"
authToken = "12345678"
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:
        latitude=random.uniform(12.867342,13.043514)
        longitude=random.uniform(77.477635,77.695109)
        Latitude=str(latitude)
        Longitude=str(longitude)
        location=geolocator.reverse(Latitude+","+Longitude)
        address=location.raw['address']
        city=address.get('city','')
        print(city)
        data = { 'lon':longitude,'lat':latitude,'city':str (city)}
        def myOnpublishCallback():
           print ( "latitude=%s %%" % latitude,"longitude=%s %% "%
longitude,"city= %s "%city, "to IBM watson" )
        success=deviceCli.publishEvent("project", "json", data, qos=0,
on_publish=myOnpublishCallback)
        if not success:
                print("not connection last from sensor to IBM IOT")


        time.sleep(10)
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

7.2 Feature code:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "pb6xw8"
```

```
deviceType = "efgh"
deviceId = "1234"
authMethod = "token"
authToken = "12345678"
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:
    level=random.randint(1,100)
    weight=random.randint(1,100)
    if(level<30):
        level_status="low level"
        print("level_status=low level garbage")
    elif(level>30)and(level<80):
        level_status="medium level garbage"
        print("level_status=low level garbage")
    else:
        level_status="high level garbage"
        print("level_status=high level garbage")
    if (weight<30):
        weight_status="low level"
        print("weight_status=low level garbage")
```

```
        elif(weight>30)and(weight<80):
                weight_status="medium level garbage"
                print("weight_status=low level garbage")

        else:
                weight_status="high level garbage"
                print("weight_status=high level garbage")

   data =
{'level':level,'level_status':level_status,'weight':weight,'weight_status':weight_stat
us }
        def myOnpublishCallback():
           print ("level=%s m"%level,"weight=%s kg"%weight , "to IBM watson" )
        success=deviceCli.publishEvent("project", "json", data, qos=0,
on_publish=myOnpublishCallback)
        if not success:
                  print("not connection last from sensor to IBM IOT")

        time.sleep(10)
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

In feature code 1 the location (latitude and longitude) is send to the IBM Watson IOT platform and in feature code 2 the level and weight of the bin these data are send to the IBM Watson IOT platform. Using Node red service the data are notified and displayed in the web Application.

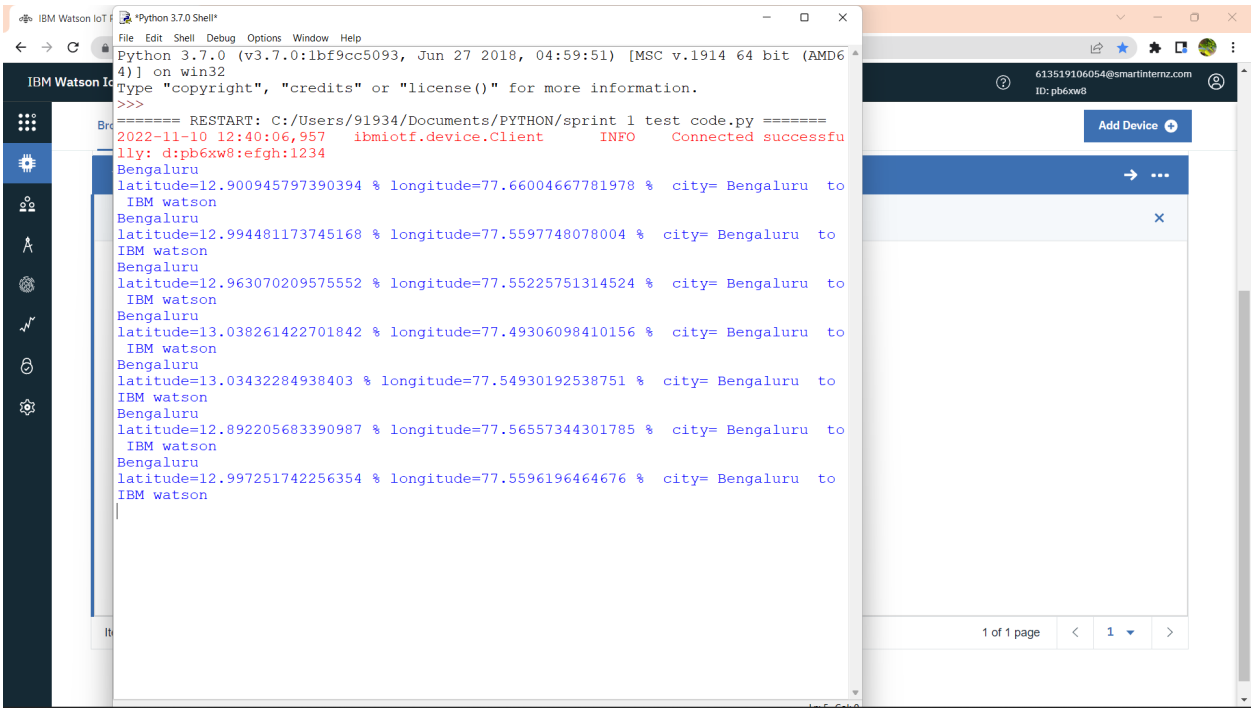
8. TESTING

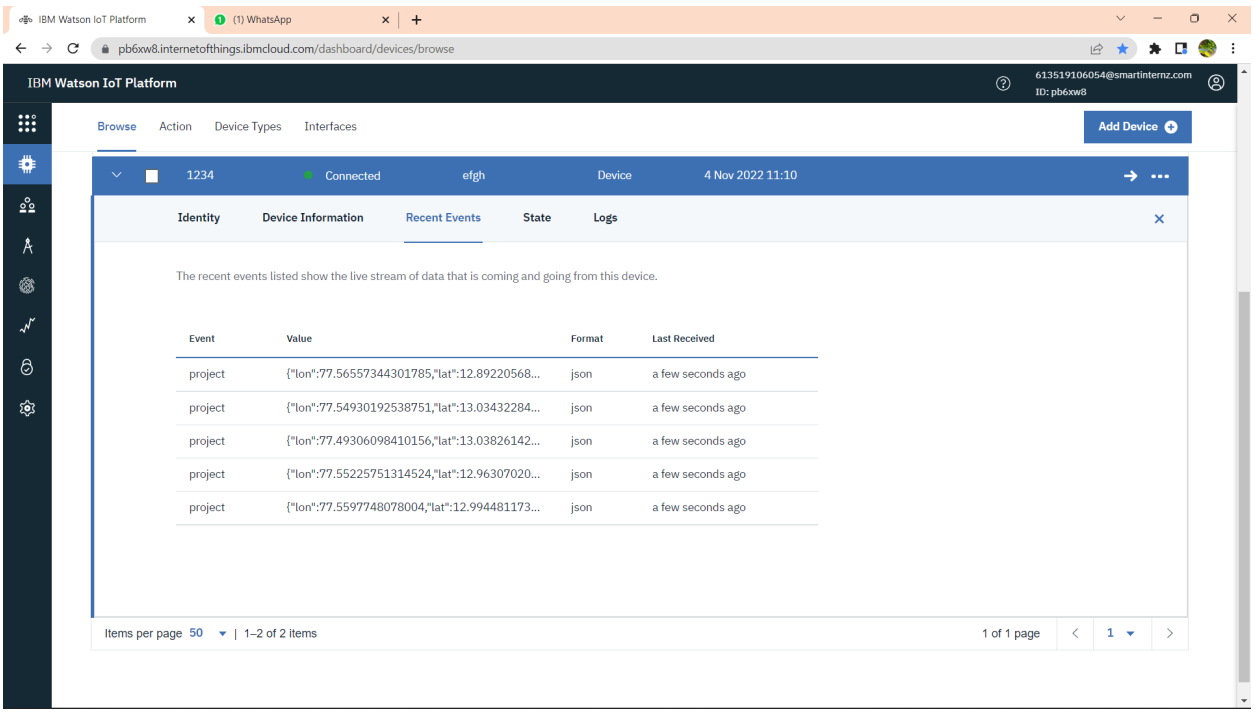
8.1 Test cases

SI.NO	INPUT	OUTPUT	RESULT
01.	Latitude, longitude, level, weight of bin	Level status: high Weight status: low	Passed
02.	Latitude, longitude, level, weight of bin	Level status: high Weight status: high	Passed
03.	Latitude, longitude, level, weight of bin	Level status: medium Weight status: low	Passed
04.	Latitude, longitude, level, weight of bin	Level status: low Weight status: low	Passed
05.	Latitude, longitude, level, weight of bin	Level status: high Weight status: high	Passed
06.	Latitude, longitude, level, weight of bin	Level status: high Weight status: low	Passed
07.	Latitude, longitude, level, weight of bin	Level status: high Weight status: medium	Passed
08.	Latitude, longitude, level, weight of bin	Level status: medium Weight status: low	Passed
09.	Latitude, longitude, level, weight of bin	Level status: low Weight status: low	Passed
10.	Latitude, longitude, level, weight of bin	Level status: high Weight status: high	Passed
11.	Latitude, longitude, level, weight of bin	Level status: high Weight status: low	Passed
12.	Latitude, longitude, level, weight of bin	Level status: high Weight status: low	Passed
13.	Latitude, longitude, level, weight of bin	Level status: high Weight status: medium	Passed
14.	Latitude, longitude, level, weight of bin	Level status: high Weight status: low	Passed

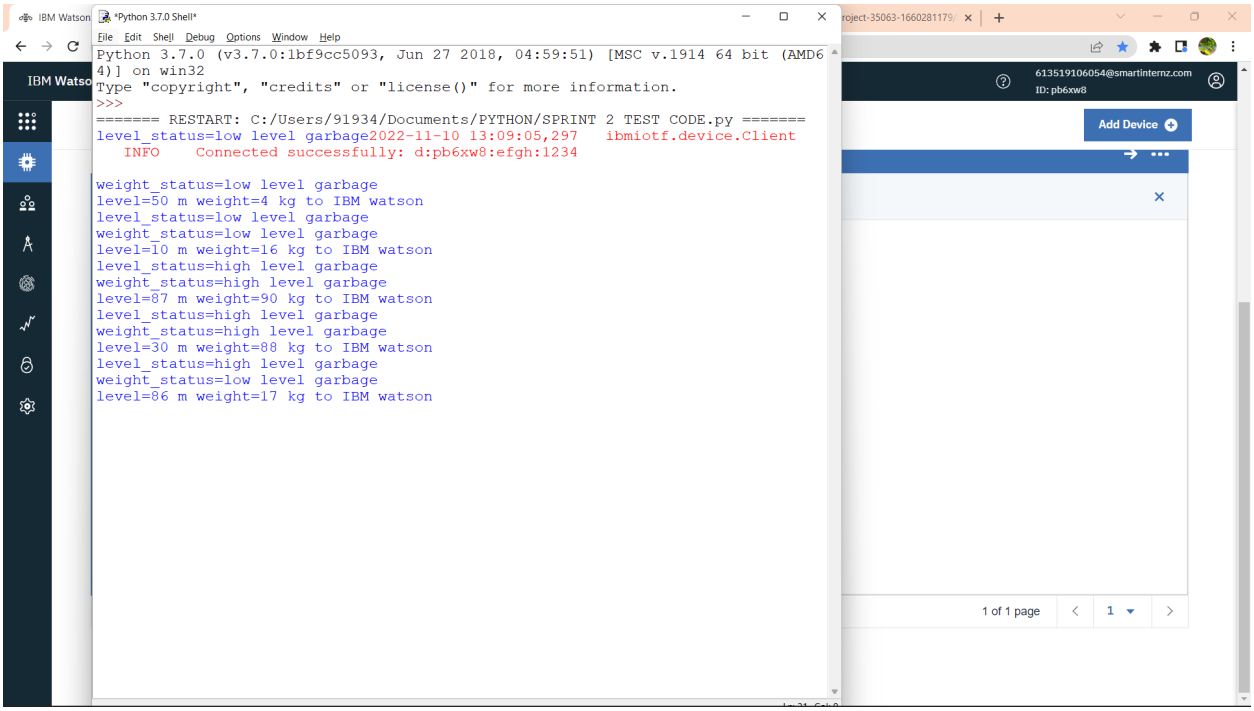
SI.NO	INPUT	OUTPUT	RESULT
15.	Latitude, longitude, level, weight of bin	Level status: high Weight status: medium	Passed
16.	Latitude, longitude, level, weight of bin	Level status: medium Weight status: low	Passed
17.	Latitude, longitude, level, weight of bin	Level status: low Weight status: low	Passed
18.	Latitude, longitude, level, weight of bin	Level status: high Weight status: high	Passed
19.	Latitude, longitude, level, weight of bin	Level status: medium Weight status: medium	Passed
20.	Latitude, longitude, level, weight of bin	Level status: low Weight status: low	Passed
21.	Latitude, longitude, level, weight of bin	Level status: high Weight status: high	Passed
22.	Latitude, longitude, level, weight of bin	Level status: high Weight status: medium	Passed
23.	Latitude, longitude, level, weight of bin	Level status: medium Weight status: low	Passed
24.	Latitude, longitude, level, weight of bin	Level status: high Weight status: low	Passed
25.	Latitude, longitude, level, weight of bin	Level status: high Weight status: low	Passed
26.	Latitude, longitude, level, weight of bin	Level status: low Weight status: low	Passed
27.	Latitude, longitude, level, weight of bin	Level status: medium Weight status: low	Passed

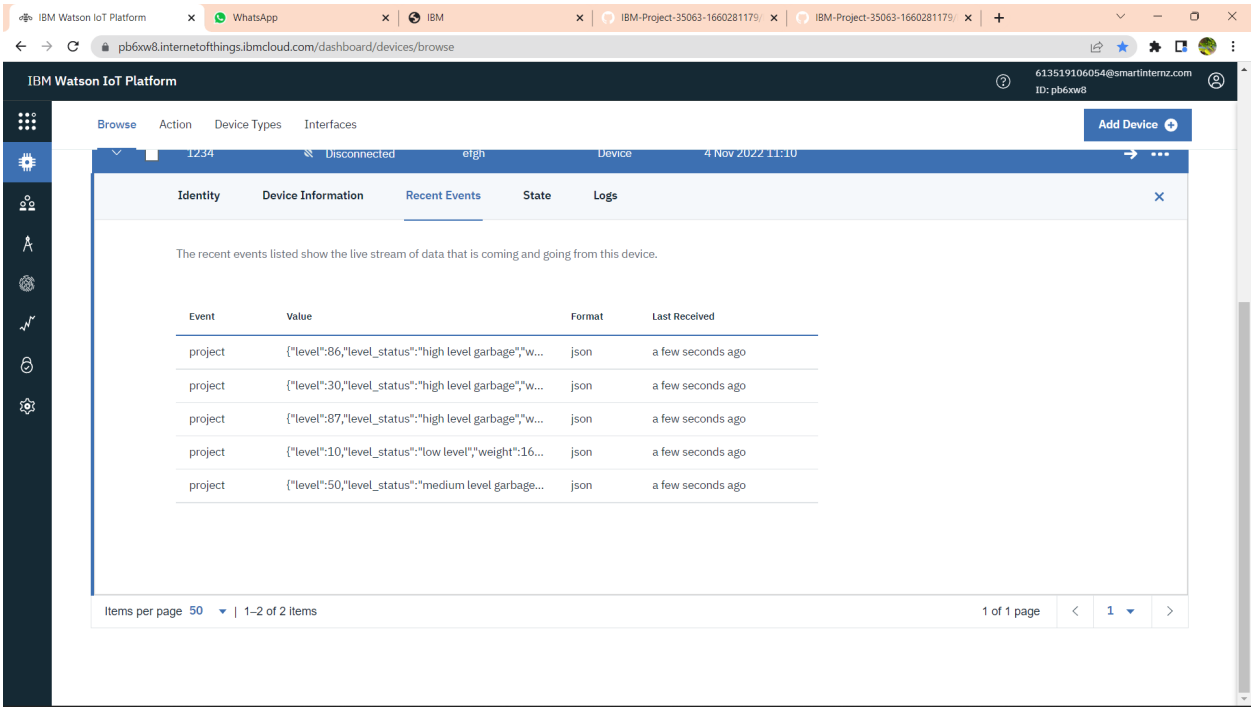
FEATURE CODE 1





FEATURE CODE 2





8.2 User Acceptance Testing

Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the smart waste management for metropolitan cities project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis

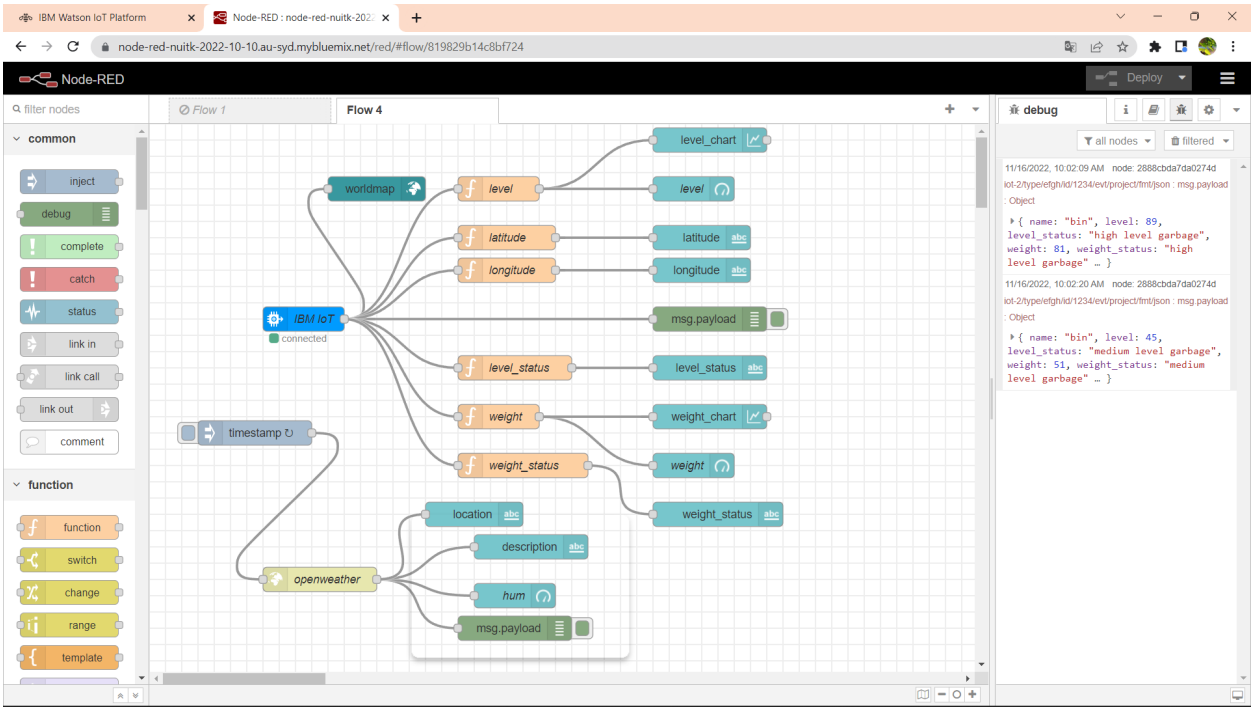
This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

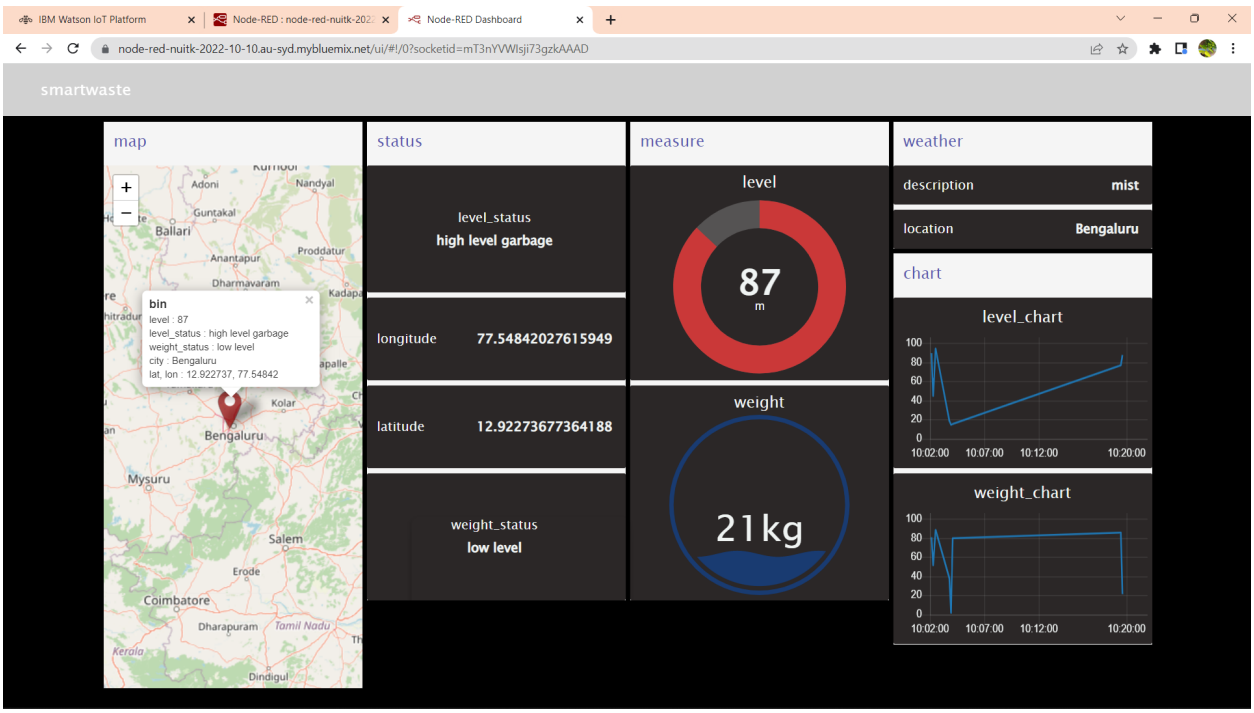
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	4	3	2	3	12
Duplicate	0	0	0	0	0
External	1	2	0	4	7
Fixed	5	5	2	6	18
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	0	0	0	0
Totals	10	10	6	14	40

Test Case Analysis

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

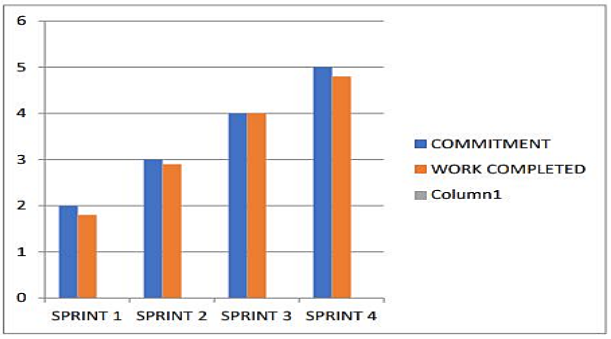
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	4	0	0	4
Client Application	1	0	0	1
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2


IBM Watson IoT Platform
Node-RED : node-red-nuitk-202
node-red-nuitk-2022-10-10.au-syd.mybluemix.net/red/#flow/819829b14c8bf724
Node-RED
Deploy
filter nodes
Flow 1
Flow 4
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
function
switch
change
range
template
worldmap
level
latitude
longitude
IBM IoT
connected
timestamp
openweather
location
description
hum
msg.payload
level_status
weight
weight_status
level_chart
level
latitude
longitude
msg.payload
level_status
weight_chart
weight
weight_status
debug
all nodes
filtered
11/16/2022, 10:02:09 AM node: 2888cbda7da0274d
iot-2/type/efgh/id/1234/evt/project/fmt/json : msg.payload : Object
{ name: "bin", level: 89, level_status: "high level garbage", weight: 81, weight_status: "high level garbage" … }
11/16/2022, 10:02:20 AM node: 2888cbda7da0274d
iot-2/type/efgh/id/1234/evt/project/fmt/json : msg.payload : Object
{ name: "bin", level: 45, level_status: "medium level garbage", weight: 51, weight_status: "medium level garbage" … }


IBM Watson IoT Platform
Node-RED : node-red-nuitk-202
Node-RED Dashboard
node-red-nuitk-2022-10-10.au-syd.mybluemix.net/ui/#!/0?socketid=mT3nYVWlsji73gzkAAAD
smartwaste
map
bin
level : 87
level_status : high level garbage
weight_status : low level
city : Bengaluru
lat, lon : 12.922737, 77.54842
Bengaluru
Mysuru
Salem
Coimbatore
status
level_status
high level garbage
longitude
77.54842027615949
latitude
12.92273677364188
weight_status
low level
measure
level
87
m
weight
21kg
weather
description
mist
location
Bengaluru
chart
level_chart
100
80
60
40
20
0
10:02:00
10:07:00
10:12:00
10:20:00
weight_chart
100
80
60
40
20
0
10:02:00
10:07:00
10:12:00
10:20:00

9. RESULTS

9.1 Performance metrics



NFT - Risk Assessment

S.No	Project Name	Ecosystem	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Volume Changes	Risk Score	Justification
1	SMART WASTE MANAGEMENT FOR METROPOLITAN CITIES	Existing	Moderate	Low	Moderate	High initial cost	+5 to 10%	GRAND	this system of waste management is already existing
						sensor has limited memory			sensor has limited memory
									the risk rate is moderate

NFT - Detailed Test Plan

S.No	Project Overview	NFT Test approach	Assumptions/Dependencies/Risks	Approvals/SignOff
1	smart waste management for metropolitan cities	to determine how frequently process in particular area	Assumptions	

End Of Test Report

S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Recommendations	Identified Defects (Detected/Closed/Open)	Approvals/SignOff
	smart waste management for metropolitan cities	issues occurred		got location lave and weight of the bin		to determine the frequently filled areas	detected and closed	
		while running		got some error while connecting with the device				
		while connecting with the device						

10. ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

- ✓ Improved cleanliness
- ✓ Dynamic routing
- ✓ Cost reduction
- ✓ System provides greater accessibility to the dustbin
- ✓ Save fuel and time using appropriate route planning.

DISADVANTAGES:

- ✓ The process is not always cost-effective
- ✓ The resultant product has a short life
- ✓ The sites are often dangerous
- ✓ The practices are not done uniformly

11. CONCLUSION:

This project will be very effective in managing waste in any big city. Rather than using conventional periodic collection methods here priority system is used to the city, is clean all the time without any overflowing dumpsters. It has been tested and verified properly to make sure all the different parts work together for a smooth function of the whole system.

This project presented a smart waste management for metropolitan cities which is based on the use of IOT technology. This system involves measuring the level and weight of the bin then send alert to the authority. The system send the location, level of the bin and weight of the bin, by developing the python script the data send to the IBM Watson IOT platform and the Node red is the web app the dashboard of the node red is the user interface.

12. FUTURE SCOPE:

This model is developed with the aim to keep environment clean and green. It can be enhanced further in many ways. Following are its future scope:-

- The scope for the future work is this system can be implemented with time stamp in which real-time clock shown to the concern person at what time dust bin is full and at what time the waste is collected from the smart dustbins.
- If this system is used to monitor dustbins in larger areas, Android app with dustbin locator can be developed so that person can track nearest bin and its status.
- A small grinder can be used along with a wet waste bin to make pieces of organic waste substances so that it will be decomposed rapidly.
- Employing camera sensor for image processing of the cleanliness of the roads and penalizing persons not throwing the garbage properly in the bin.

13. APPENDIX:

13.1 Source code:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
from geopy.geocoders import Nominatim
# initialize Nominatim API
geolocator = Nominatim(user_agent="geoapiExercises")
#Provide your IBM Watson Device Credentials
organization = "pb6xw8"
deviceType = "efgh"
deviceId = "1234"
authMethod = "token"
authToken = "12345678"
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
print("checking connection to waston iot...")
deviceCli.connect()
time.sleep(1)
while True:
       name='bin'
       level=random.randint(1,100)
       weight=random.randint(1,100)
       latitude=random.uniform(12.867342,13.043514)
       longitude=random.uniform(77.477635,77.695109)
       Latitude=str(latitude)
```

```
        Longitude=str(longitude)
        location = geolocator.reverse(Latitude+","+Longitude)
        address = location.raw['address']
        city = address.get('city', '')
        print('City : ', str(city))
        #STATUS OF GARBAGE CAN
        if(level<30):
                level_status="low level"
                print("level_status=low level garbage")
        elif(level>30)and(level<80):
                level_status="medium level garbage"
                print("level_status=low level garbage")
        else:
                level_status="high level garbage"
                print("level_status=high level garbage")
        if (weight<30):
                weight_status="low level"
                print("weight_status=low level garbage")
        elif(weight>30)and(weight<80):
                weight_status="medium level garbage"
                print("weight_status=low level garbage")
        else:
                weight_status="high level garbage"
                print("weight_status=high level garbage")
        data = { 'name' : name,'level' : level,'level_status':level_status,'weight':
weight,'weight_status':weight_status, 'lat': Latitude,'lon':Longitude,'city':str(city)}
        #print data
        def myOnPublishCallback():
           print ("Published weight = %s kg" % weight,"level=%s m" %level, "latitude = %s %%"
% latitude, "longitude = %s %%" % longitude,"city=%s" %city,"to IBM Watson")
        success = deviceCli.publishEvent("project", "json", data, qos=0,
on_publish=myOnPublishCallback)
        if not success:
           print("Not connection lost from sensor to ibm iot")
        time.sleep(10)
# Disconnect the device and application from the cloud
deviceCli.disconnect(
```

13.2 GITREPO & DEMO LINK:

GITHUB LINK

https://github.com/IBM-EPBL/IBM-Project-35063-1660281179

DEMO LINK

https://drive.google.com/file/d/1hWFRqtPmkCZo5PHA_Hu9LlqL3hGnRFyX/view?usp=share_link