PROJECT REPORT

SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITAN CITIES

Team ID: PNT2022TMID28660
Batch: B8-2A4E

TEAM LEADER:

Name: JAGADISH D

Register Number: 410719104033

TEAM MEMBERS:

Name: DEEPAN S

Register Number: 410719104019

Name: DINESHKUMAR M

Register Number: 410719104026

Name: YOGESHKUMAR N

Register Number: 410719104125

TABLE OF CONTENTS

1.INTRODUCTION

1.1 PROJECT OVERVIEW

Smart waste management is characterized by the usage of technology in order to be more efficient when it comes to managing waste. This makes it possible to plan more efficient routes for the trash collectors who

empty the bins, but also lowers the chance of any bin being full for over a week!

1.2 PURPOSE

Smart waste management is about using technology and data to create a more efficient waste industry. Based on IoT (Internet of Things) technology, smart waste management aims to optimize resource allocation, reduce running costs, and increase the sustainability of waste services

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

Misunderstanding of the operations of smart sensors:

Because this is a new and emerging technology, there is a general misunderstanding of its operations. A lot of people believe that it is a complicated and expensive method to dispose of waste, which is not. They are actually very affordable, easy to use, durable and save costs.

Setting up the smart sensor:

While smart sensors are easy to use, you cannot just buy one and install it on your waste bin. There are other steps that need to be taken after purchase to ensure its effectiveness like ensuring there is a communication technology in place for your sensor. This is responsible for collecting information about your garbage bin and sending alerts to appropriate ports for attention.

Non-optimized truck routes:

Truck routes that are not optimized lead to the use of excessive fuel. In addition, some bins may end up being overfilled and others under-filled as a result. Overfilled bins pollute the environment and have poor aesthetics quality.

Recycling:

Having smart technology is only half the solution. The other half is ensuring that wastes are disposed of responsibly and that recyclable wastes are properly sorted. The method of trash disposal from their creation points is generic, that is, all the waste is in one bin so the job of sorting falls on the waste management company. The method of sorting is predominantly manual, so the process is very slow. However, an

automated system that manages the life cycle of products could make the process of recycling used products easier. Technical information about the product, including the materials from which it was gotten, the appropriate method of recycling, among other things, would greatly improve the recycling process and make it more cost effective.

Non-uniform waste distribution of waste in bins:

Most smart sensors use just distance measurement to determine the fill levels of the trash bins, so if trash has been deposited unevenly in one part of the bin than the other, the sensors may read that the bin is full, when in reality it is only half full.

2.2 REFERENCES

PAPER 1:

TITLE: IoT Based Waste Management for Smart City

AUTHOR NAME: Parkash Tambare, Prabu Venkatachalam

PUBLICATION YEAR: 2016

DESCRIPTION: In the current situation, we frequently observe that the trash cans or dust cans that are located in public spaces in cities are overflowing due to an increase in the amount of waste produced each day. We are planning to construct “IoT Based Waste Management for Smart Cities” to prevent this from happening because it makes living conditions for people unsanitary and causes unpleasant odours in the surrounding area. There are numerous trash cans scattered throughout the city or on the campus that are part of the proposed system. Each trash can is equipped with a low-cost embedded device that tracks the level of the trash cans and an individual ID that will enable it to be tracked and identified.

PAPER 2:

AUTHOR NAME: Mohammad Aazam, Marc St-Hilaire, Chung-Horng Lung, Ioannis Lambadaris

PUBLICATION YEAR: 2016

DESCRIPTION: Each bin in the Cloud SWAM system that Mohammad Aazam et al suggested has sensors that can detect the amount of waste inside. There are

separate bins for organic, plastic/paper/bottle/glass, and metal waste. This way, each form of waste is already divided, and it is known how much and what kind of waste is collected thanks to the status. Different entities and stakeholders may benefit from the accessibility of cloud-stored data in different ways. Analysis and planning can begin as soon as garbage is collected and continue through recycling and import/export-related activities. Timely garbage collection is provided via the Cloud SWAM system. A timely and effective method of waste collection improves health, hygiene, and disposal.

PAPER 3:

TITLE: Arduino Microcontroller Based Smart Dustbins for Smart Cities

AUTHOR NAME: K. Suresh, S. Bhuvanesh and B. Krishna Devan

PUBLICATION YEAR: 2019

DESCRIPTION: In this paper, a technique for cleaning up our surroundings and environment is described. The Indian government just began work on a smart city initiative, and in order for these towns to be smarter than they already are, the garbage collection and disposal system must be improved upon. SelfMonitoring Automated Route Trash (SMART) dustbins are intended for use in smart buildings such as colleges, hospitals, and bus stops, among other places. In this study, we have employed the PIR and Ultrasonic sensors to detect human presence, the Servomotor to open the dustbin lid, and the Ultrasonic sensor to detect the level of rubbish. Signals between two trash cans are transmitted using a communication module, and the GSM module sends the message to the operator.

PAPER 4:

AUTHOR NAME: Mohd Helmy Abd Wahab, Aeslina Abdul Kadir, Mohd Razali Tomari and Mohamad Hairol Jabbar

PUBLICATION YEAR: 2014

DESCRIPTION: Proposed a smart recycle bin that can handle the recycling of plastic, glass, paper, and aluminium cans. It generates a 3R card after automatically determining the value of the trash thrown away. The recycle system makes it possible to accumulate points for placing waste into designated recycle bins. By allowing the points to be redeemed for goods or services, such a system promotes recycling activities. The system keeps track of information on disposal procedures, materials disposed of, user identification,

and points accrued by the user. To use the recycle bin, the user must tap his card to the designated RFID reader. Doors to recycling bins are opened, and rubbish is placed one by one.

PAPER 5:

TITLE: Waste Management Initiatives in India For Human Wellbeing

AUTHOR NAME: Dr. Raveesh Agarwal, Mona Chaudhary and Jayveer Singh
PUBLICATION YEAR: 2015

DESCRIPTION: The objective of this paper is to examine the present methods used in India for the welfare of its people in different waste management efforts. The other goal is to offer advice on how to make Indian municipalities' trash disposal procedures better. On secondary research, this essay is founded. The system is improved by looking at the reports that have already been written about waste management and the suggestions made for improvement by planners, NGOs, consultants, government accountability organisations, and important business leaders. It provides in-depth understanding of the various waste management programmes in India and identifies areas where waste management might be improved for societal benefit. The essay makes an effort to comprehend the crucial part that our nation's official waste management sector plays in the waste management process.

2.3 PROBLEM STATEMENT DEFINITION

Problem Statement:

PROJECT TITLE: SMART WASTE MANAGEMENT SYSTEM	**PROJECT DESIGN PHASE-1 -SOLUTION FIT TEMPLATE**	**TEAM ID:PNT2022TMID28660**

1. CUSTOMER SEGMENT(S) Waste holders, such as private individuals, property owners of companies are our customers.	**4.CUSTOMER** • As it is technology based it needs internet access to work properly. • Customers need to buy some IOT Devices to access. • They may use solar energy instead of electrical power.	**7.AVAILABLE SOLUTIONS** • Shop eco-friendly with reusable bags. • Join buy -and-sell groups. • Digital trash bins are alternative to dustbins, because digital bins can detect the trash level and send notifications to the customers.
2. JOBS-TO-BE-DONE / PROBLEMS • Separate your waste. • Create a composite site. • Growing pressure in outdated waste management infrastructure, with declining level of capital investments and maintenance.	**5.PROBLEM ROOT CAUSE** i • Lack of industry expertise. • Emission of greenhouse gases. • Poor recycling quality due to lack of education.	**8. BEHAVIOUR** • If the sensors are not working properly contact the customer care of drop a message.

3. TRIGGERS	6. YOUR SOLUTION	9.CHANNELS OF BEHCHANAVIOUR
Seeing how neighbours are having a clean environment after using it people will get admire my seeing others. **BEFORE /AFTER** • Before using this technology, society is suffered by health issues because the waste products produce are pollution. • After using this technology, they feel at easy as it provides a clean society.	• Our solutions is to manage the waste efficiently by indicating the garbage level to the users as well as authenticating persons to collect it and proceed to further process with the garbage. • The purpose is of making clean Environment.	ONLINE: If it is in online mode, the bin is full it sends the notification to the authorized persons OFFLINE: • If it is offline every day the waste collecting trucks will collect garbage from home.

3.IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

Empathy Map:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

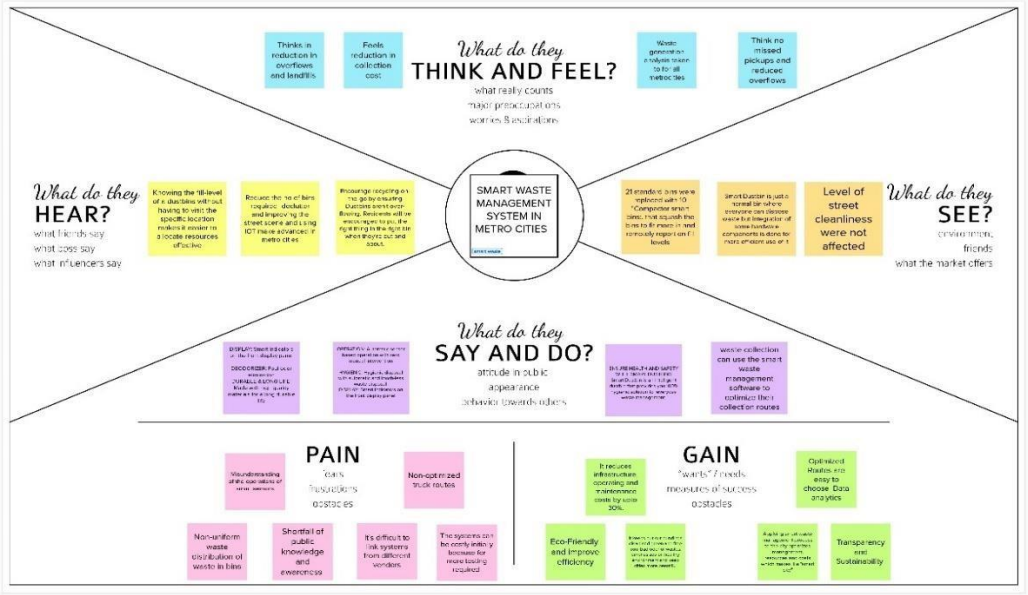
It is a useful tool to helps teams better understand their users.

Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

Empathy Map

smart waste management system for metro cities using IOT

Team ID: PNT2022TMID28660



3.2 IDEATON & BRAINSTORMING

Step-1: Team Gathering, Collaboration and Select the Problem Statement

Step-2: Brainstorm, Idea Listing and Grouping

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

JAGADISH

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).

Placing Ultrasonic sensor to detect level of bins

DEEPAN

Enable GPS function to locate bins easier

Waste generation analysis to understand cities usages

DEEPAN

Load cell on bottom of bins

Place Arduion board at left side of bins

YOGESH KUMAR

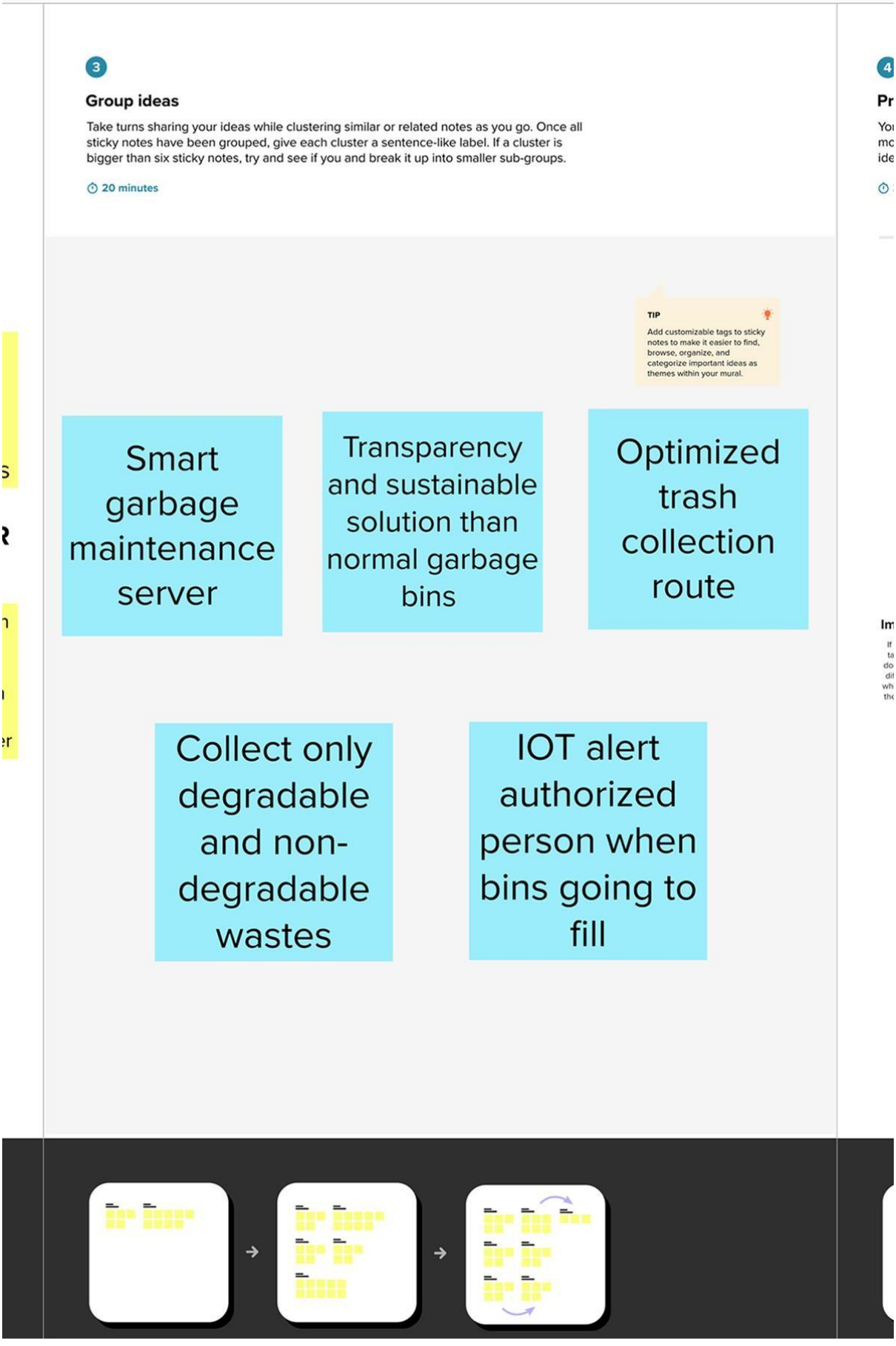
Visual fill status indicators on top of bins

using by GSM in bins achieve wireless communication with bins and managing center

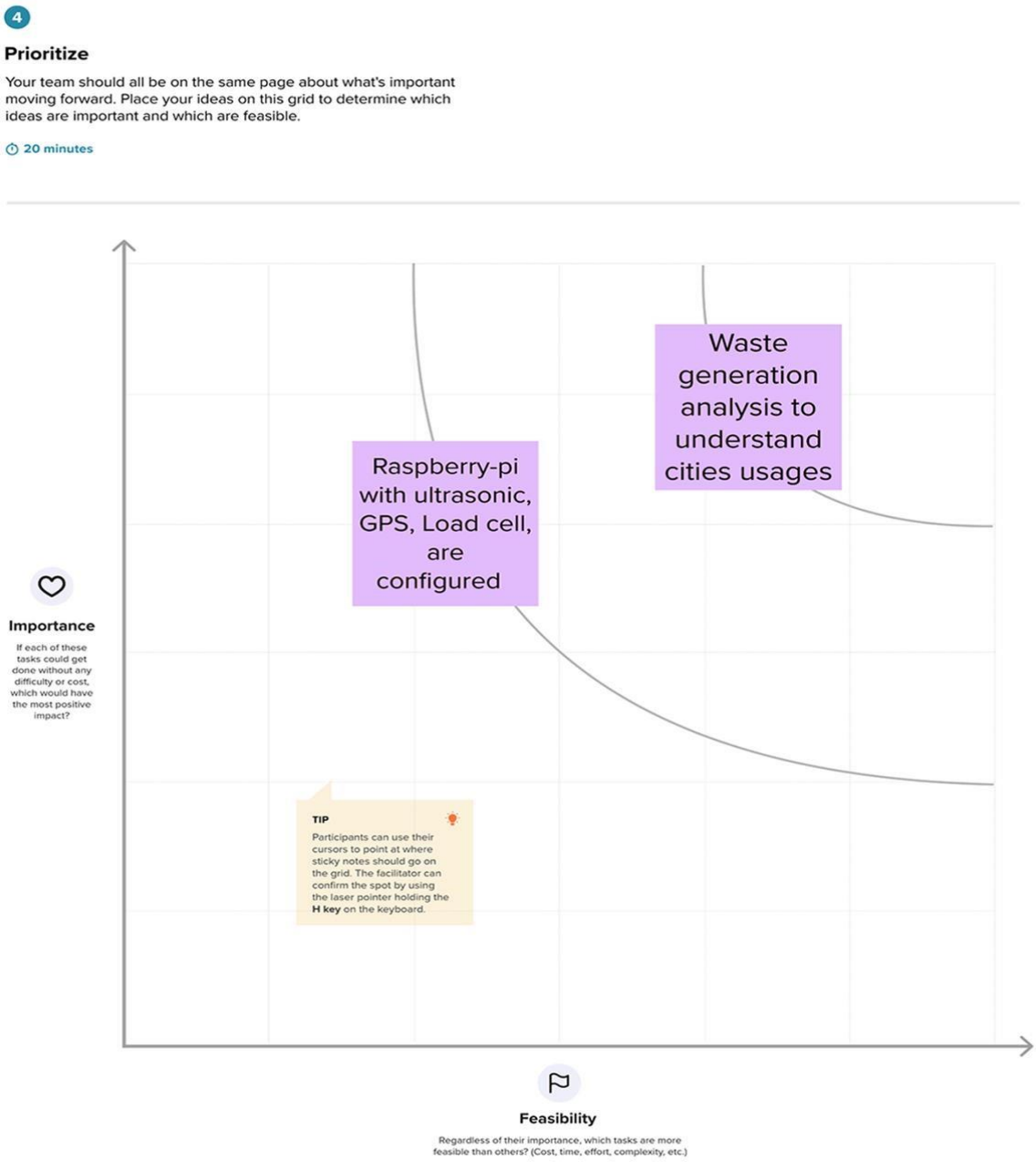
DINESH KUMAR

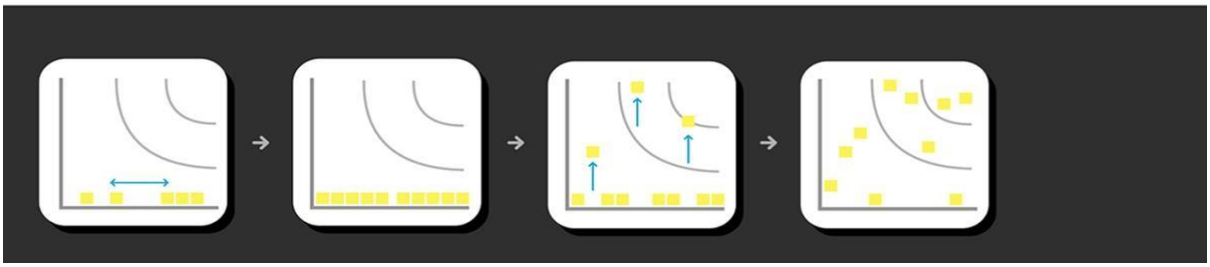
when bins fill alert message to the authorized person

solar panels for power supply for IOT devices



Step-3: Idea Prioritization





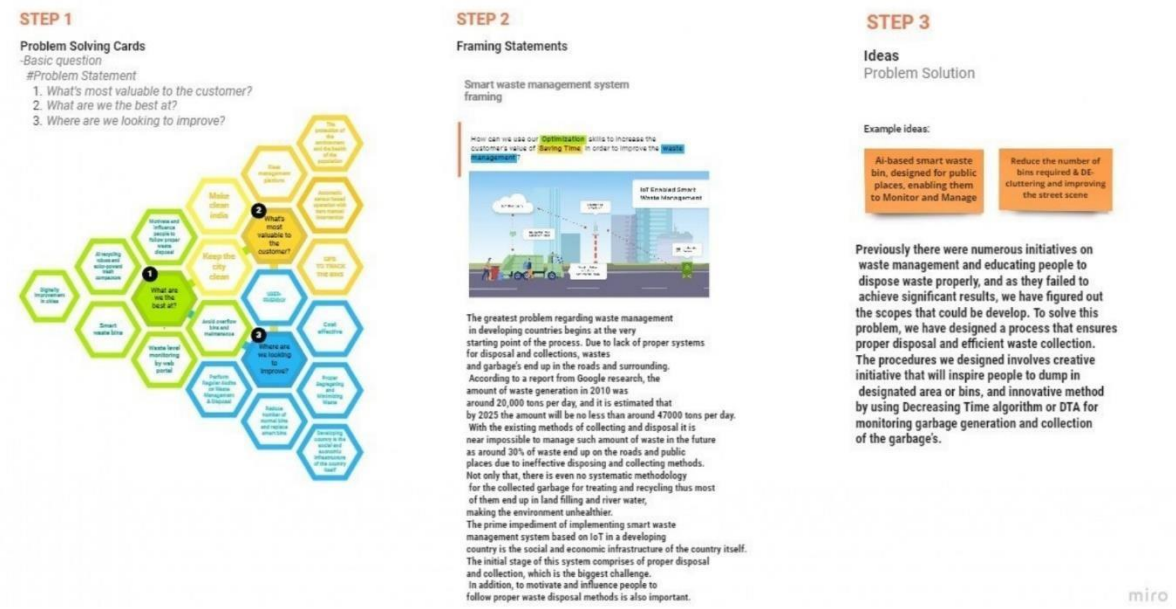
3.3 PROPOSED SOLUTION

Proposed Solution Template:

S.No.	Parameter	Description

1.	Problem Statement (Problem to be solved)	• Rubbish and waste can cause air and water pollution. • Rotting garbage is also known to produce harmful gases mix with the air and cause breathing problem in people. • Due to improper waste disposal, we may face several problems like unpleasant odour and health problems.
2.	Idea / Solution description	• To solve this problem of waste management for disposal using a smart refuse bin built with technologies like Sensors, Arduino Yun. • Garbage truck Weighing Mechanisms. • AI Recycling Robots.
3.	Novelty / Uniqueness	• Identify potential waste streams. • Create a waste management-focused community outreach plane.
4.	Social Impact / Customer Satisfaction	• Neighbourhood of landfills to communities, breeding of pests and loss in property values. • The IOT solution uses the data and selects optimum routes for waste collection trucks
5.	Business Model (Revenue Model)	• It generates revenue through the provision of various waste management and disposal services. • Recycling solutions to residential, commercial , industrial, and municipal clients.
6.	Scalability of the Solution	• Installing more bins fir collecting recyclables like paper, glass,plastice. • Recycling not only save energy but also prevent the material from going to landfills & Incineration and provides raw materials for new products.

3.4 PROPOSED SOLUTION FIT



4.REQUIREMENT ANALYSIS

4.1 FUNCTONAL REQUIREMENTS

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. • Bins or stands are visible on the map as green, orange or red circles. • 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 realtime 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 depo-bindischarge in the area. The tool assigns bin a rating (1-10) and calculates distance from depo-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 un-efficient picks	• Eliminate the collection of halfempty bins. • The sensors recognize picks. • By using real-time data on filllevels and pick recognition, we can show you how full the bins you collect are. • The report shows how full the bin was when picked. You immediately see any inefficient picks below 80% full.

FR-6	Plan waste collection routes.	• The tool semi-automates waste collection route planning. • Based on current bin fill-levels and predictions of reaching full capacity, you are ready to respond and schedule waste collection. ● You can compare planned vs. executed routes to identify any inconsistencies

4.2 NON-FUNCTIONAL REQUIREMENTS

FR 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 Purchase wisely and recycle Avoid single use 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 data driven 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 , cities coz we able to monitor the garbage 24/7 more cost efficient and scalability when we move to smarter.

5.PROJECT DESIGN

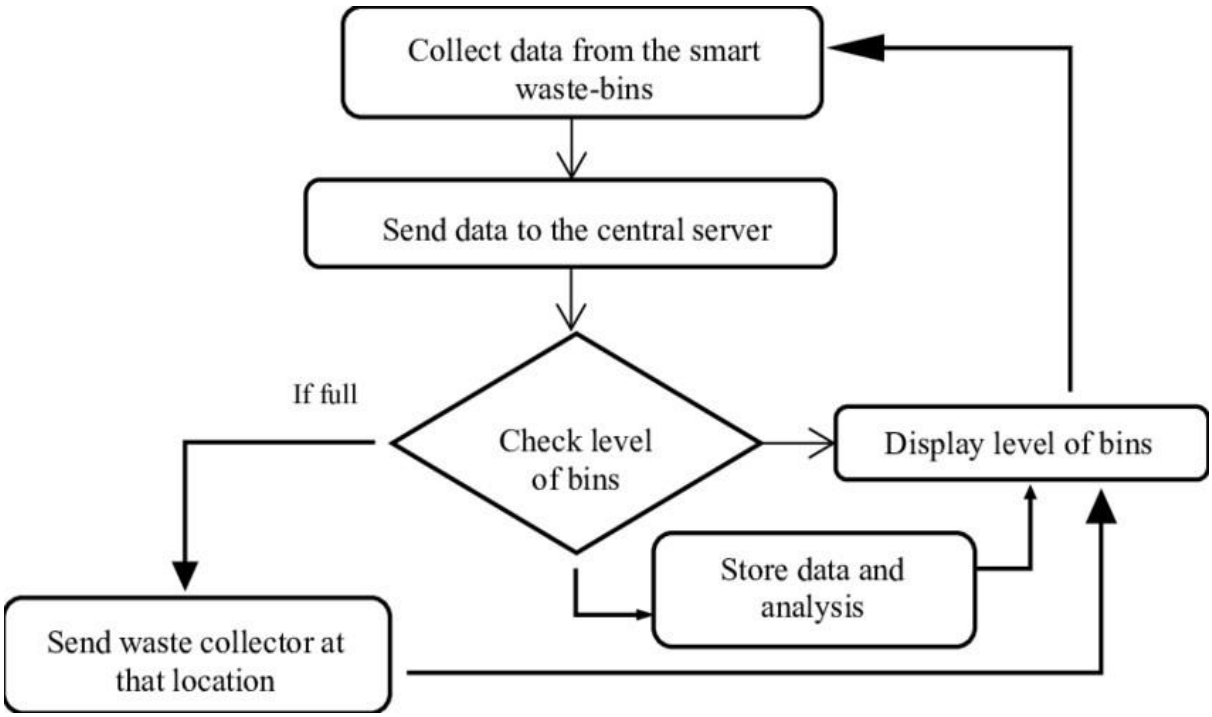
5.1 DATA FLOW DIAGRAMS

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically.

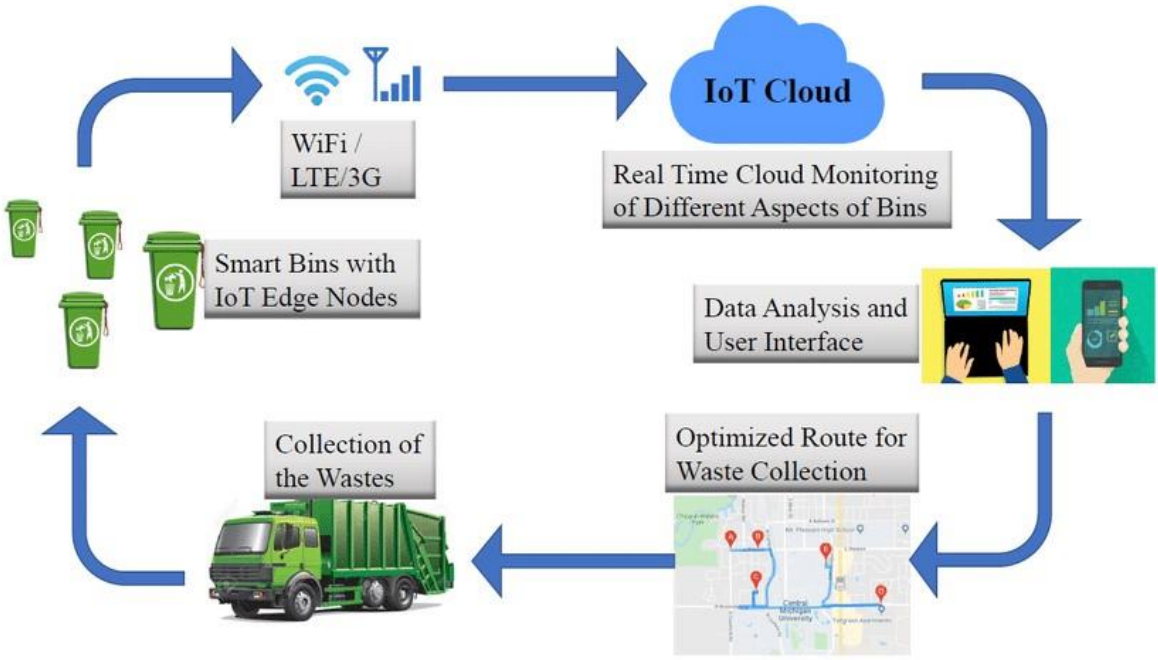
It shows how data enters and leaves the system, what changes the information, and where data is stored. A smart waste management platform uses analytics to translate the data gather in your bins into actionable insights to help you improve your waste services.

You can receive data on metric such as:

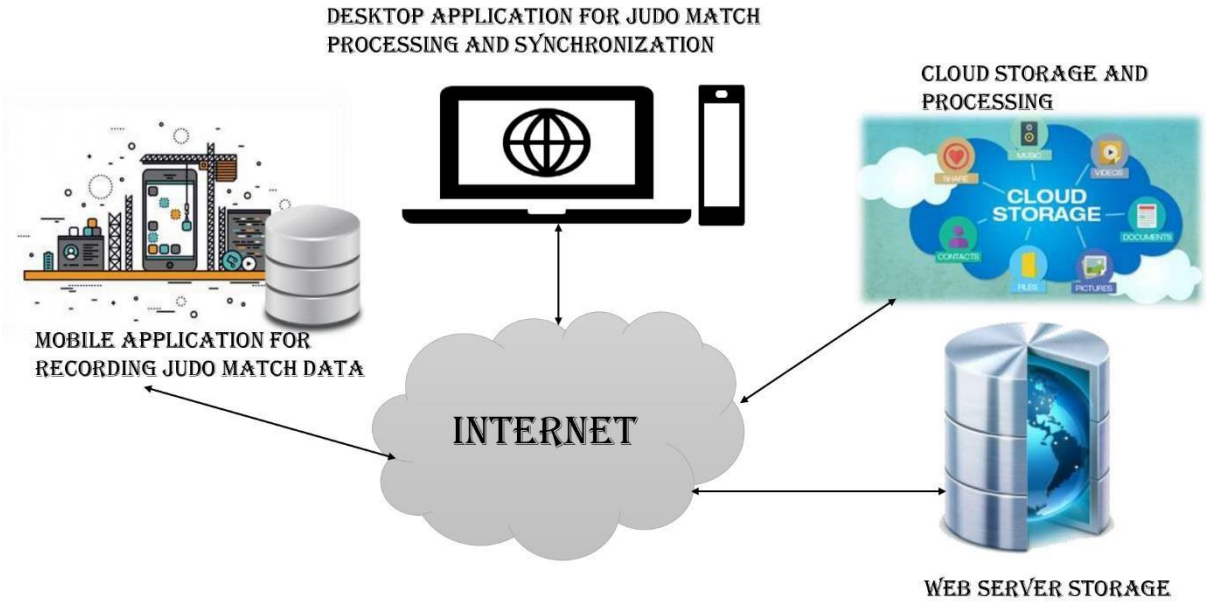
- The first test conducted is the situation where the garbage bin is empty or its garbage level is very low

- Then, the bin is filled with more garbage until its level has surpassed the first threshold value, which is set to 80% then the first warning SMS is being sent, as depicted The first notification SMS sent by the system, once the waste reaches the level of 85% full
- The second notification SMS sent by the system, indicating that bin is at least 95% full and the garbage needs to be collected immediately
- Locations prone to overflow
- The number of bins needed to avoid overflowing waste
- The number of collection services that could be saved
- The amount of fuel that could be saved
- The driving distance that could be saved
- 

5.2 SOLUTION AND TECHNICAL ARCHITECTURE



Technical Architecture:



5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release

Admin(who manages server)	Web server login	USN-1	As a admin, I can able to track the truck driver name, id, contact number, location, and also the location of the dustbin	I can Manage and direct workers through web server	High	Sprint-1
Co-Admin	Login	USN-2	As a coadmin I'll monitor the workers, whether the work has been done properly, checking the availability of workers and also monitor the waste	I can monitor the garbage bin activity	High	Sprint-1
			collected by the truck driver within the scheduled time			
Customer (Web user)	User	USN-3	As a user , I can able to raise queries to higher authorities about the maintenance and disposal of waste	I can raise queries	Medium	Sprint-2
Customer Care Executive	Worker	USN-4	As a customer care executive I will try to rectify the queries from customers by contacting coadmin. In case of emergency situation query can be reported to Admin.	I can attend calls and respond people and solve their problems	High	Sprint-1

Truck driver	Worker	USN-5	The truck driver is a worker who has been assigned to collect the garbage and he have to report to admin about when and where and also the timings , the garbage has been picked up according the daily schedule.	I will do the work properly and report the data at the scheduled time	High	Sprint-1

6.PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

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

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications 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 Requirement	Prepare the functional requirement document.	8 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	9 OCTOBER 2022
Technology Architecture	Prepare the technology architecture diagram.	10 OCTOBER 2022
Prepare Milestone & Activity List	Prepare the milestones& activity list of the project.	22 OCTOBER 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it.	IN PROGRESS..

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Functional Requirement (Epic)	Task	Story Point	Priority	Team Member
Sprint-1	Registration	As a team lead, I can enrolled for the project by entering my email, password and within that I can enter my team members name and their email.	**2**	High	**Jagadish**
Sprint-1		As a team lead, I will receive confirmation email once, I have enrolled for the project with team id and along with team members name.	**2**	High	**Jagadish**
Sprint-2		As a team member, I can login to the IBM portal by entering email & password	**1**	Medium	**Deepan**

Sprint-2		As a team member, I can login to the IBM portal by entering email & password	**1**	Medium	**Deepan**
Sprint-2		As a team member, I can login to the IBM portal by entering email & password Sprint-	**1**	Medium	**Yogesh**
Sprint-2		As a team member, I login to the IBM portal by entering email & password	**1**	Medium	**Dinesh**

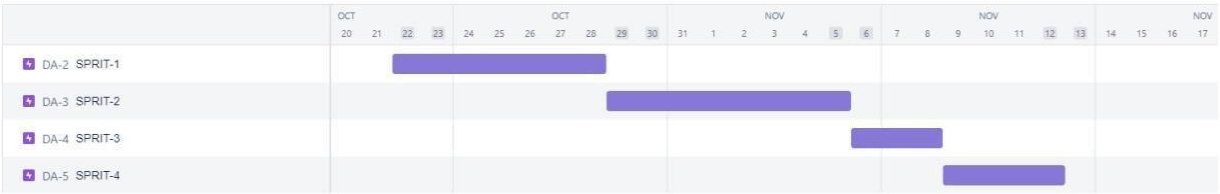
Project Tracker, Velocity & Burndown Chart:

Sprint	Total Story Points	Duration	n Sprint StartDate	Sprint End Date (Planned)	Story Points Completed (Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	22 Oct 2022	27 Oct 2022	**20**	06 Nov 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	**30**	07 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	**49**	08 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	**50**	09 Nov 2022

6.3 REPORTS FROM JIRA

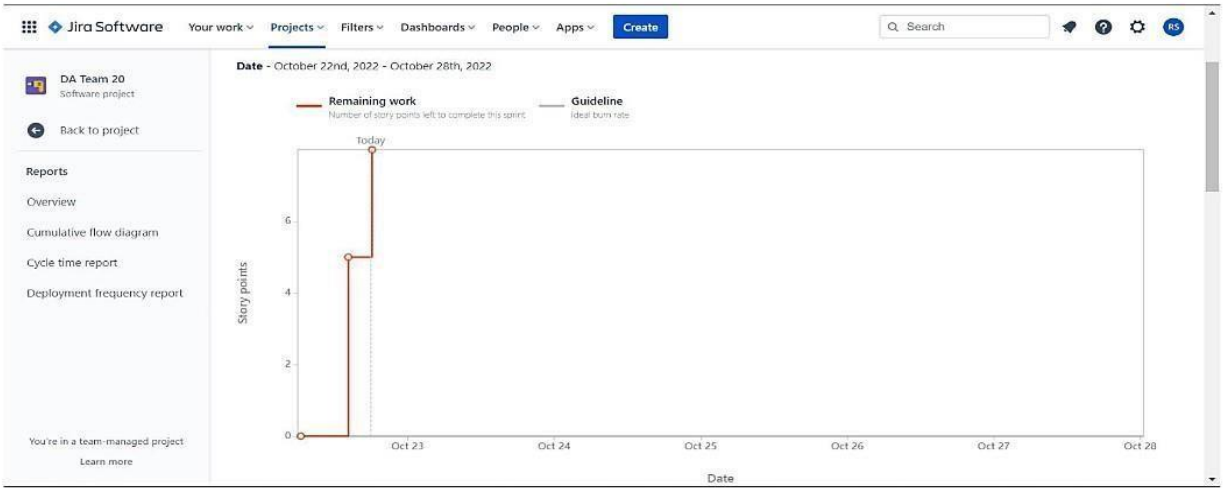
Road Map:

A roadmap is a strategic plan that defines a goal or desired outcome and includes the major steps or milestones needed to reach it. It also serves as a communication tool, a high-level document that helps articulate strategic thinking the why behind both the goal and the plan for getting there.



Kanban Board:

A kanban board is an agile project management tool designed to help visualize work, limit work-in-progress, and maximize efficiency (or flow). It can help both agile and DevOps teams establish order in their daily work.



7.CODING & SOLUTIONING

7.1 FEATURE 1

1. HTML for web page making
2. CSS, JS, vendor for static design content
3. Python Flask for the web server integration

7.2 FEATURE 2

1. Creating IBM cloud account
2. Develop a web application using node-red service 3. Cloud Object Storage
4. Accessing IBM Watson assistance for the chatbot 5. Cloudant DB.

8.TESTING

8.1 TEST CASES

8.1 Test Cases

- To predict the weight of the garbage bin

- To locate the location of the bin
- To send alerts to the local authorities about the garbage level

TEST CASE ID	FEATUR E TYPE	COM PO NENT	TEST SCENARIO	STEPS TO EXEC UT E	TEST DATA	EXPECTED RES	ACTUAL RESULT	STAT U S	COMMENTS	EXECUTED BY
LOGIN PAGE_TC_ 001	FUNCTI ONA L	HOM E PAGE	VERIFY THE USER IS ABLE TO SEE THE LOGIN/SI G N UP WEN USER CLICK ON MY ACCOUNT BUTTON	1.ENT ER URL AND CLICK GO 2.VER IFY LOGI N/SI GN UP	https:// 1 69.51.2 0 4.219.3 0 106	L0gin page is visible	Working as expected	PASS	Successful	Jagadish
LOGIN PAGE_TC_ 002	UI	HOM E PAGE	VERIFY THE USER IS ABLE TO SEE THE LOGIN/SI G N UP WEN USER CLICK ON MY ACCOUNT BUTTON	1.ENT ER URL AND CLICK GO 2.VER IFY LOGI N/SI GN UP Elements a.ID text b o x B . password text box c login	https:// 1 69.51.2 0 4.219.3 0 106	Application should show below UI element	Working as expected	PASS	Successful	Deepan
LOGIN PAGE_TC_ 003	FUNCTI ONA L	LOGI N PAGE	VERIFY THE USER IS ABLE TO SEE THE LOGIN/SI G N UP WEN USER CLICK ON MY ACCOUNT BUTTON	1.ent er url and click go 2.click on my account 3.Ent er valid ID 4.Ent er valid password 5.click on login	Id:1111 password d:5678	User should navigate your home page.	Working as expected	PASS	Successful	Deepan
LOGIN	FUNCTI ONA L	LOGI N	VERIFY	THE USER IS ABLE TO SEE THE LOGIN/SIG N UP WEN USER CLICK ON MY ACCOUNT BUTTON	Id:1 111 pass worked:56 78	Confirmation message sent	Working as expected	PASS	Successful	Yogesh

LOGIN PAGE_TC_ 004	FUNCTI ONA	LOGIN PAGE	VERIFY THE USER IS ABLE TO SEE THE LOGIN/SIG N UP WEN USER CLICK ON MY ACCOUNT BUTTON	1.enter url and click go 2.click on my account 3.Enter valid ID 4.Enter valid password 5.click on login button	Id:1 111 password d:56 78	Confirmation message sent	Working as expected	PASS	Successful	Dinesh
LOGIN PAGE_TC_ 005	FUNCTI ONA	LOGI N	VERIFY	1.ent er url	Id:1111	Confirm	Working as expected	PASS	Successful	Dinesh
LOGIN PAGE_TC_ 006	FUNCTIONA L	LOGIN PAGE FOR ADMIN	VERIFY THE USER IS ABLE TO SEE THE LOGIN/SIG N UP WEN USER CLICK ON MY ACCOUNT BUTTON	1.enter url and click go 2.click on my account 3.Enter valid ID 4.Enter valid password 5.click on login button	Id:1 111 password d:56 78	Customer database is visible	Working as expected	PASS	Successful	Yogesh

8.2 ACCEPTANCE TESTING

1.Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2.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	10	4	2	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	7

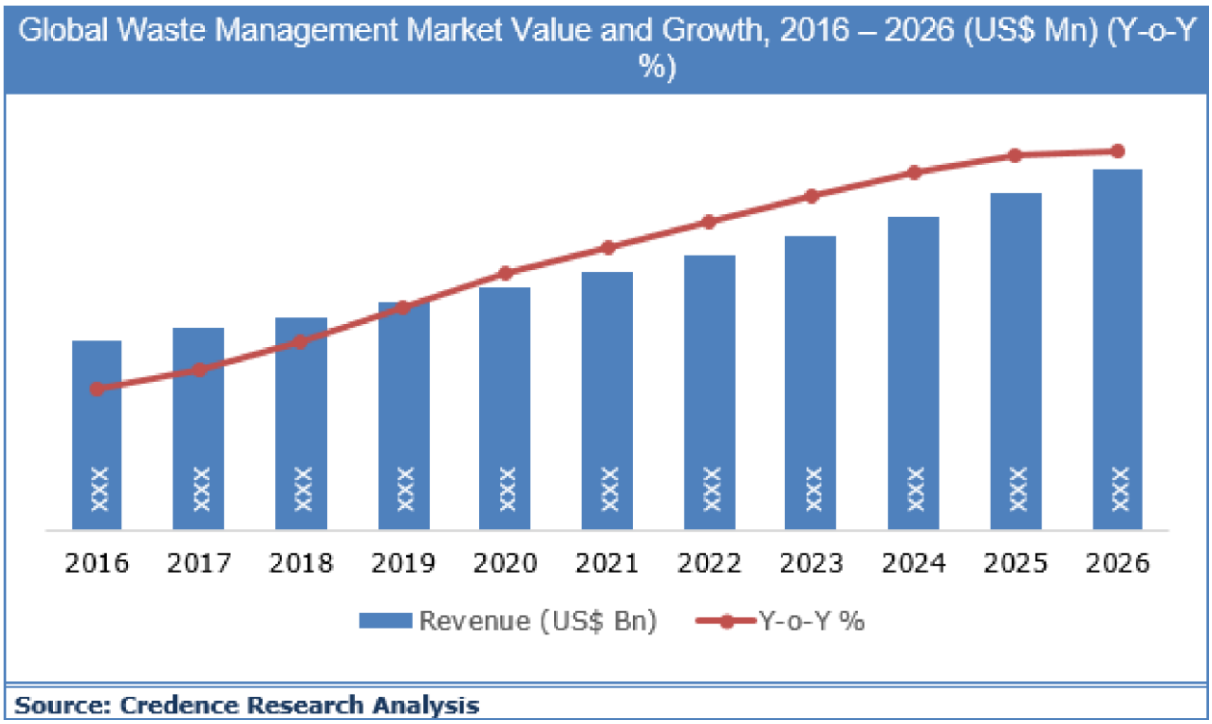
3.Test Case Analysis

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

Section	**Total Cases**	**Not Tested**	**Fai l**	**Pas s**
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

9.RESULTS

9.1 PERFORMANCE METRICS



10.ADVANTAGES & DISADVANTAGES

Advantages of Smart Waste Management System

- Overflowing of the solid waste is prevented due to regular monitoring of the parameters
- Location of the garbage bins are sent to the local authorities so that it is easy to monitor the bins of that locality
- Can be easily accessible from any part of the world
- Easy to debug errors
- Parameters are more accurate

Disadvantages of Smart Waste Management System

Misunderstanding of the operations of smart sensors:
Because this is a new and emerging technology, there is a general misunderstanding of its operations. A lot of people believe that it is a complicated and expensive method to dispose of waste, which is not. They are actually very affordable, easy to use, durable and save costs.

Setting up the smart sensor:
While smart sensors are easy to use, you cannot just buy one and install it on your waste bin. There are other steps that need to be taken after purchase to ensure its effectiveness like ensuring there is a communication technology in place for your sensor. This is responsible for collecting information about your garbage bin and sending alerts to appropriate ports for attention.

Non-optimized truck routes:

Truck routes that are not optimized lead to the use of excessive fuel. In addition, some bins may end up being overfilled and others under-filled as a result. Overfilled bins pollute the environment and have poor aesthetics quality.

11. CONCLUSION

A Smart Waste Management system that is more effective than the one in use now is achievable by using sensors to monitor the filling of bins. Our conception of a "smart waste management system" focuses on monitoring waste management, offering intelligent technology for waste systems, eliminating human intervention, minimizing human time and effort, and producing a healthy and trash free environment. The suggested approach can be implemented in smart cities where residents have busy schedules that provide little time for garbage management. If desired, the bins might be put into place in a metropolis where a sizable container would be able to hold enough solid trash for a single unit. The price might be high.

12.FUTURE SCOPE

There are several future works and improvements for the proposed system, including the following:

1. Change the system of user authentication and atomic lock of bins, which would aid in protecting the bin from damage or theft.
2. The concept of green points would encourage the involvement of residents or end users, making the idea successful and aiding in the achievement of collaborative waste management efforts, thus fulfilling the idea of Swachh Bharath.
3. Having case study or data analytics on the type and times waste is collected on different days or seasons, making bin filling predictable and removing the reliance on electronic components, and fixing the coordinates.
4. Improving the Server's and Android's graphical interfaces

13.APPENDIX SOURCE CODE

```
from
flask_sqlalchemy
import SQLAlchemy


    from flask_migrate import Migrate    from datetime
import datetime


                   db = SQLAlchemy()


                  def setup_db(app):        db.app = app          db.init_app(app)
        migrate = Migrate(app, db)


                                                                          def commit():
                        db.session.commit()



                   collect = db.Table('collect',
                                      db.Column('plate_number', db.Integer,
                   db.ForeignKey('vehicles.plate_number'), primary_key=True),
                                      db.Column('basket_id', db.Integer,
    db.ForeignKey('baskets.id'), primary_key=True),
db.Column('DOC', db.DateTime, primary_key=True)
                                      )


                   complaint = db.Table('complaint',
                                        db.Column('user_name', db.String,
                   db.ForeignKey('users.user_name'), primary_key=True),
```

```
                                    db.Column('basket_id', db.Integer,
                    db.ForeignKey('baskets.id'), primary_key=True),
                    db.Column('date_of_compliant', db.DateTime, primary_key=True),
                    db.Column('compliant_message', db.String),                      )



                    class Basket(db.Model):
        __tablename__ = 'baskets'           id = db.Column(db.Integer,
primary_key=True)            longitude = db.Column(db.Float,
nullable=False)       latitude = db.Column(db.Float, nullable=False)
                        software_version = db.Column(db.String, nullable=False,
                    default="0.0.0")
                        wastes_height = db.Column(db.Integer, nullable=False, default=0)
                        wastes = db.relationship('Waste', lazy=True,
                    backref=db.backref('basket', lazy=True))
                        software_versions = db.relationship('SoftwareVersion', lazy=True,
                    backref=db.backref('basket', lazy=True))
                        type = db.Column(db.Integer, db.ForeignKey('basketsTypes.id'),
                    nullable=False)
                        area_code = db.Column(db.Integer, db.ForeignKey('areas.code'),
                    nullable=False)
          def save(self):           if self.id is None:               db.session.add(self)
db.session.commit()          return self
                                                                  def delete(self):
            db.session.delete(self)             db.session.commit()
                                                def format(self):                return {
                                "id": self.id,
                                "longitude": self.longitude,
                                "latitude": self.latitude,
                                "software_version": self.software_version,
                                "micro_controller": self.basketType.micro_controller,
                                "level": "{}%".format(self.get_basket_level())
                            }


                        def set_wastes_height(self, waste_height):
                            if waste_height <= (self.basketType.height -
                    self.wastes_height):
                                self.wastes_height +=
                    waste_height             return False
                    return True


                        def get_waste_volume(self, height):
                            return (self.length * self.width * float(height)) / 1000000
                                                        def get_basket_level(self):
                            return int((self.wastes_height / self.basketType.height) *
                    100)


                  class Area(db.Model):             __tablename__ = "areas"       code =
db.Column(db.Integer, primary_key=True)             name = db.Column(db.String,
unique=True, nullable=False)         size = db.Column(db.Float, nullable=False)
longitude = db.Column(db.String, nullable=False)           latitude =
db.Column(db.String, nullable=False)        city = db.Column(db.String, nullable=False)
                        baskets = db.relationship('Basket', lazy=False,
                    backref=db.backref('area'))
                        users = db.relationship('User', backref=db.backref('area'))
```

```
                    def save(self, has_key_by_default=False):              if
self.code is None or has_key_by_default:
               db.session.add(self)            db.session.commit()
                                               def format(self):                return {
                            "area_code": self.code,
                            "area_name": self.name,
                            "area_size": self.size,
                            "longitude": self.longitude,
                            "latitude": self.latitude,
                            "city": self.city
                        }


                                                                    class User(db.Model):
                    __tablename__ = 'users'     user_name =
                db.Column(db.String, primary_key=True)     first_name
                = db.Column(db.String, nullable=False)     last_name
                = db.Column(db.String, nullable=False)     email =
                db.Column(db.String, nullable=False, unique=True)
                password = db.Column(db.String, nullable=False)
                gender = db.Column(db.String, nullable=False)
                    DOB = db.Column(db.DateTime)
                    phone = db.Column(db.String)     area_code =
                db.Column(db.Integer, db.ForeignKey('areas.code'),
                nullable=False)
                    baskets = db.relationship('Basket', secondary=complaint,
                lazy=True, backref=db.backref('complainants'))
                   def save(self, has_key_by_default=False):              if
self.user_name is None or has_key_by_default:
               db.session.add(self)
db.session.commit()              return self
                                               def format(self):                return {
                            "user_name": self.user_name,
                            "first_name": self.first_name,
                            "last_name": self.last_name,
                            "email": self.email,
                            "gender": self.gender,
                            "Date_of_birth": self.DOB
                        }


                          class Employee(db.Model):        __tablename__ = 'employees'
       SSN = db.Column(db.BigInteger, primary_key=True)          full_name =
db.Column(db.String, nullable=False)        user_name = db.Column(db.String,
nullable=False, unique=False)       password = db.Column(db.String, nullable=False)
       DOB = db.Column(db.DateTime, nullable=False)       phone =
db.Column(db.String)
                    vehicle = db.relationship('Vehicle', uselist=False,
   lazy="select", backref=db.backref('driver'))           supervise =
db.relationship("Employee")
                    supervise_SSN = db.Column(db.BigInteger,
                db.ForeignKey('employees.SSN'), nullable=True)

                    def save(self, has_key_by_default=False):
                if self.SSN is None or has_key_by_default:
                db.session.add(self)         db.session.commit()


                    def delete(self):
                        db.session.delete(self)
```

```
                        db.session.commit()
                                    def format(self):                return {
                            "SSN": self.SSN,
                            "full_name": self.full_name,
                            "user_name": self.user_name,
                            "date_of_birth": self.DOB,
                            "phone": self.phone
                        }


    class Vehicle(db.Model):         __tablename__ = "vehicles"         plate_number =
db.Column(db.Integer,  primary_key=True)           container_size  =  db.Column(db.Float)
tank_level = db.Column(db.Float)      tank_size = db.Column(db.Float)
                     employee_SSN = db.Column(db.BigInteger,
                db.ForeignKey('employees.SSN'))
                    baskets = db.relationship('Basket', secondary=collect, lazy=True,
                backref=db.backref('baskets'))
                   def save(self, has_key_by_default=False):              if
self.plate_number is None or has_key_by_default:
               db.session.add(self)            db.session.commit()
                                    def format(self):                return {
                            "plate_number": self.plate_number,
                            "container_size": self.container_size,
                            "tank_level": self.tank_level,
                            "tank_size": self.tank_size,
                            "driver": self.driver.format() if self.driver else {}
                        }



                class Waste(db.Model):
                __tablename__ = "wastes"     id =
                db.Column(db.Integer,
                primary_key=True)     size =
                db.Column(db.Float)
                    type = db.Column(db.String)
                    DOC = db.Column(db.DateTime, nullable=False)
                    basket_id = db.Column(db.Integer, db.ForeignKey('baskets.id'),
                nullable=True)
                   def save(self, has_key_by_default=False):              if self.id
is None or has_key_by_default:
               db.session.add(self)
db.session.add(self.basket)
db.session.commit()              return self
                                    def format(self):                return {
                            "basket_id": self.basket_id,
                            "size": self.size,
                            "type": self.type,
                            "date_of_creation": self.DOC
                        }
                                 def delete(self):               db.session.commit()


                class SoftwareVersion(db.Model):         __tablename__ =
"software_versions"          version = db.Column(db.String(), primary_key=True)
file = db.Column(db.LargeBinary())          date = db.Column(db.DateTime,
server_default=db.func.now())
                     basket_id = db.Column(db.Integer, db.ForeignKey('baskets.id'),
                nullable=True, primary_key=True)
                   def save(self, has_key_by_default=False):              if
self.version is None or has_key_by_default:
```

```
                db.session.add(self)
db.session.commit()                 return self
                                            def format(self, status):                 return {
                                    "version": self.version,
                                    "date": self.date,
                                    "status": status
                                }
                                            def delete(self):                 db.session.commit()


                        class BasketType(db.Model):         __tablename__ = "basketsTypes"
id = db.Column(db.Integer, primary_key=True)         length = db.Column(db.Integer,
nullable=False)         width = db.Column(db.Integer, nullable=False)                height =
db.Column(db.Integer, nullable=False)         micro_controller = db.Column(db.String,
nullable=False)
                         basket = db.relationship('Basket', lazy=True,
                        backref=db.backref('basketType', lazy=True))
            def save(self):             if self.id is None:                 db.session.add(self)
db.session.commit()             return self
                                                                                  def format(self):
                                return {
                                    "name": "{}*{}*{}/{}".format(self.length, self.width,
    self.height, self.micro_controller),
"value": self.id
                                }
```

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-50265-1660901330

PROJECT DEMO LINK:

https://drive.google.com/drive/folders/1hjj_M7Y0S_B8YrJEnrvx5E9YELQncOPQ