Smart Waste Management System For Metropolitan Cities

Team ID: PNT2022TMID03073

Team Members:

19euec030 Deepak S

19euec031   Deepak S

19euec032 Deepikarani K

19euec033 Deva Dharshini S

Project Guide:

Industry mentor: Mr. Dinesh

Faculty Mentor: N.Kalaivani

ABSTRACT

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

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

INDEX

CHAPTER 1: INTRODUCTION

1.1 Project Overview

Smart waste management is an innovative approach to handling and collecting waste. Based on IoT (Internet of Things) technology, smart waste management provides data on waste generation patterns and behaviour.

Our Smart waste management solution uses sensors placed in garbage bins to measure fill levels and notifies city collection services when bins are ready to be emptied. There are load and ultrasonic sensors placed to continuously monitor the bins. This data is sent to the cloud (via a microcontroller that is connected to Wi-Fi) where it is stored after which it is processed further. When the levels exceed a certain limit, a notification is sent to the garbage collector via a web application.

Over time, historical data collected by sensors can be used to identify fill patterns, optimize driver routes and schedules, and reduce operational costs. The cost of these sensors is steadily decreasing, making IoT waste bins more feasible to implement and more attractive.

1.2 Purpose

Around 2.1 billion tonnes of municipal solid waste is generated annually around the globe. Population growth and rapid urbanization lead to a huge increase in waste generation, so the traditional methods of waste collection have become inefficient and costly. This system cannot measure the fullness levels of containers, and as a result, half-full containers can be emptied, and in contrast, pre-filled ones need to wait until the next collection period comes. Moreover, since drivers collect empty bins, predefined collection routes of the system cause waste of time, an increase in fuel consumption, and excessive use of resources.

In today’s ever-technological world, an innovative and data-driven approach is the only way forward, the waste sector needs a solution that empowers event-driven waste collection. The most efficient way this extraordinary amount of waste can be solved is through smart waste management without obsolete methods of waste collection. This empowers municipalities, cities, and waste collectors to optimize their waste operations, become more sustainable, and make more intelligent business decisions.

CHAPTER 2: LITERATURE SURVEY

2.1 Existing Problem

Around 80% of waste collections happen at the wrong time. Late waste collections lead to overflowing bins, unsanitary environments, citizen complaints, illegal dumping, and increased cleaning and collection costs. Early waste collections mean unnecessary carbon emissions, more traffic congestion, and higher running costs. The old way of doing waste management is highly inefficient. And in today's ever-technological world, an innovative and data-driven approach is the only way forward. Traditionally, municipalities and waste management companies would operate on a fixed collection route and schedule. This means that waste collection trucks would drive the same collection route and empty every single waste container – even if the waste container did not need emptying. This means high labour and fuel costs – which residents ultimately foot the bill for.

2.2 References

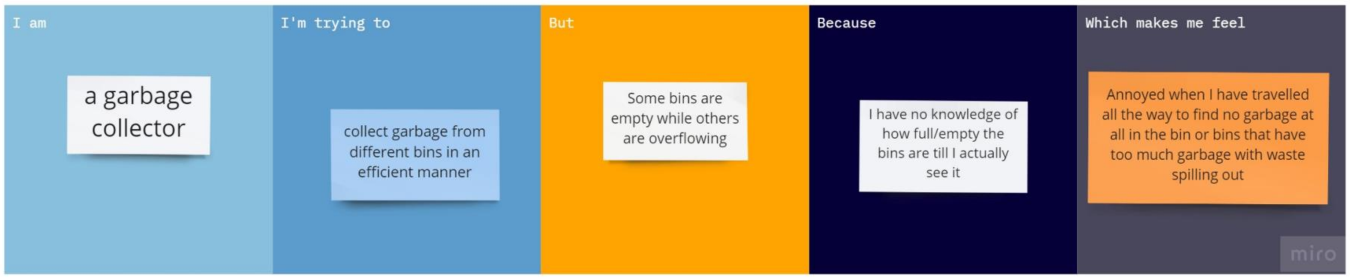
Paper Title	Author	Outcome
IOT based Smart Garbage System	1) T.Sinha 2) R.M. Sahuother	IoT Based Smart Garbage System which indicates directly that the dustbin is filled to a certain level by the garbage and cleaning or emptying them is a matter of immediate concern. This prevents lumping of garbage in the roadside dustbin which ends up giving foul smell and illness to people. The design of the smart dustbin includes a single by ultrasonic sensor which configured with Arduino Uno with this research, it is sending SMS to the Municipal

		Council that dustbin is to overflow.
Raspberry pi-based smart waste management system using Internet of Things.	1)Shaik Vaseem Akram 2)Rajesh Singh	Nowadays it is becoming a difficult task to distinguish wet and dry waste. The new waste management system covers several levels of enormous workforce. Every time, laborers must visit the garbage bins in the city area to check whether they are filled or not. The data communicates to the cloud server for real-time monitoring of the system. With the real-time fill level information collected via the monitoring platform, the system reduces garbage overflow by informing about such instances before they arrive

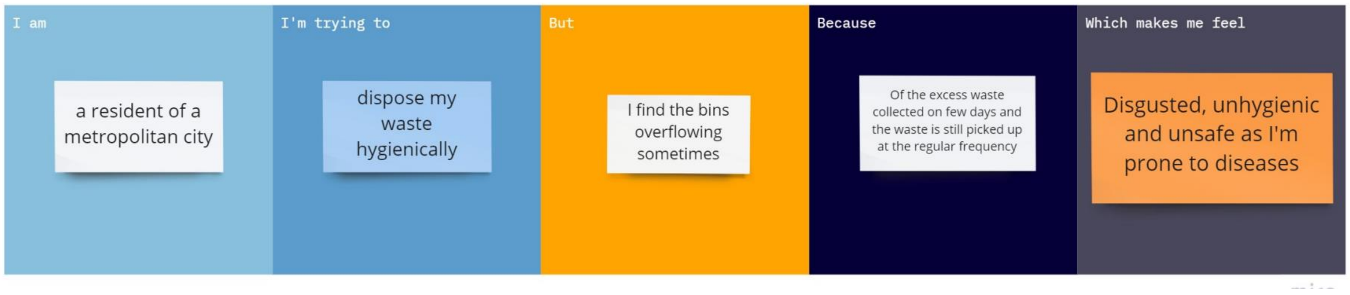
Smart Waste Management System.	1) Sanjiban Charkraborty	This Waste management is one of the serious challenges of the cities, the system now used in cities, we continue to use an old and outmoded paradigm that no longer serves the entail of municipalities, Still find over spilled waste containers giving off irritating smells causing serious health issues and atmosphere impairment.
Smart Solid Waste Management.	1) Mohd Helmy Abd Wahab	At the time of trash disposal, the material to be recycled could be identified using RFID technology.
Analysis of Load cell.	1) Ranjeet Kumar 2) Sandeep Chhabra	Load Cells 4.1 General Load Cell related information A load cell is meant to measure the size of a mass but actually is a force sensor which transforms force into an electrical signal. The load cell needs the earth gravity to work. Every mass is attracted by the earth gravimetric field, that force is named “load”.
Smart Waste Management using Wireless Sensor Network	1) Tarandeep Singh 2) Rita Mahajan 3) Deepak Bagai	In most of the places, garbage bins are not cleaned at periodic intervals, giving a hygienic issue. Thus, a system to manage bins, by using intelligent bins, gateway and remote base station is created. But this system is prone to attacks

		from hackers and complexity to build it is very high.
Smart Waste Management for Green Environment	1) T. P. Fei	The system is based on Bootstrap platform. This system works on the waterfall methodology which has 4 crucial phases: planning and analysis, system design, system implementation and system testing. Using this system, operators can get the information regarding collection from trash bins. The limitations of this approach are that the resultant product has a short life and uniformity is lost after a certain period.

2.3 Problem Statement Definition



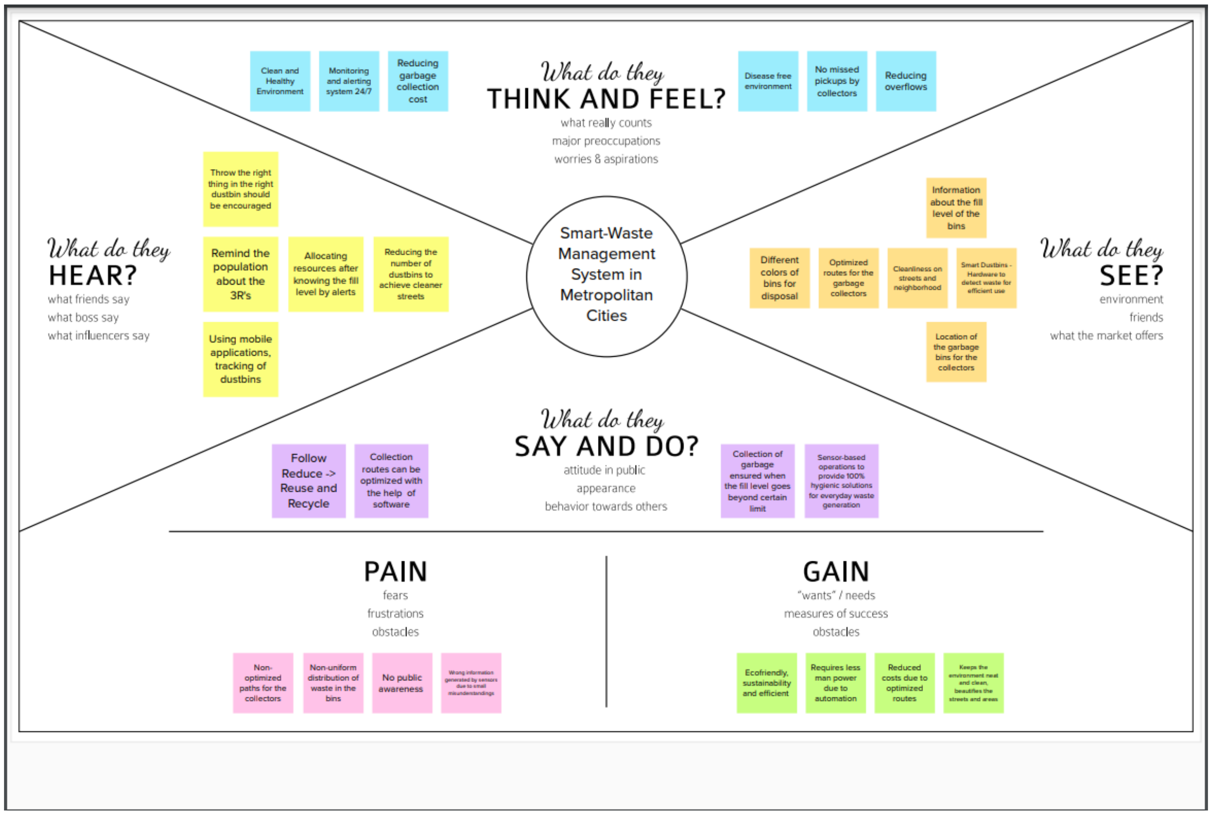
I am	I'm trying to	But	Because	Which makes me feel
a garbage collector	collect garbage from different bins in an efficient manner	Some bins are empty while others are overflowing	I have no knowledge of how full/empty the bins are till I actually see it	Annoyed when I have travelled all the way to find no garbage at all in the bin or bins that have too much garbage with waste spilling out



I am	I'm trying to	But	Because	Which makes me feel
a resident of a metropolitan city	dispose my waste hygienically	I find the bins overflowing sometimes	Of the excess waste collected on few days and the waste is still picked up at the regular frequency	Disgusted, unhygienic and unsafe as I'm prone to diseases

CHAPTER 3: IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas


Clean and Healthy Environment
Monitoring and alerting system 24/7
Reducing garbage collection cost
What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Disease free environment
No missed pickups by collectors
Reducing overflows
Throw the right thing in the right dustbin should be encouraged
What do they
HEAR?
what friends say
what boss say
what influencers say
Remind the population about the 3R's
Allocating resources after knowing the fill level by alerts
Reducing the number of dustbins to achieve cleaner streets
Using mobile applications, tracking of dustbins
Smart-Waste Management System in Metropolitan Cities
Information about the fill level of the bins
Different colors of bins for disposal
Optimized routes for the garbage collectors
Cleanliness on streets and neighborhood
Smart Dustbins - Hardware to detect waste for efficient use
Location of the garbage bins for the collectors
What do they
SEE?
environment
friends
what the market offers
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
Follow Reduce -> Reuse and Recycle
Collection routes can be optimized with the help of software
Collection of garbage ensured when the fill level goes beyond certain limit
Sensor-based operations to provide 100% hygienic solutions for everyday waste generation
PAIN
fears
frustrations
obstacles
Non-optimized paths for the collectors
Non-uniform distribution of waste in the bins
No public awareness
Wrong information generated by sensors due to small misunderstandings
GAIN
"wants" / needs
measures of success
obstacles
Ecofriendly, sustainability and efficient
Requires less man power due to automation
Reduced costs due to optimized routes
Keeps the environment neat and clean, beautifies the streets and areas

3.2 Ideation & Brainstorming

3.2.1 Brainstorm by team members

Deepak s

24/7 monitoring system is designed for monitoring bins

The bin will contain a microcontroller with an embedded Wi-fi module

Deva Dharshini S

if the bin is full then an alert message is sent

The message contains the weight of the bin and its location with the time it was recorded at

Once the message is sent, route optimization for nearby bins that have to be emptied can be done using the application

Deepak S

Use a single point load cell module to measure the weight as it can be used for small weighing systems and has stainless steel versions suitable for even organic waste

Embed a GPS module in the bin to send it's location

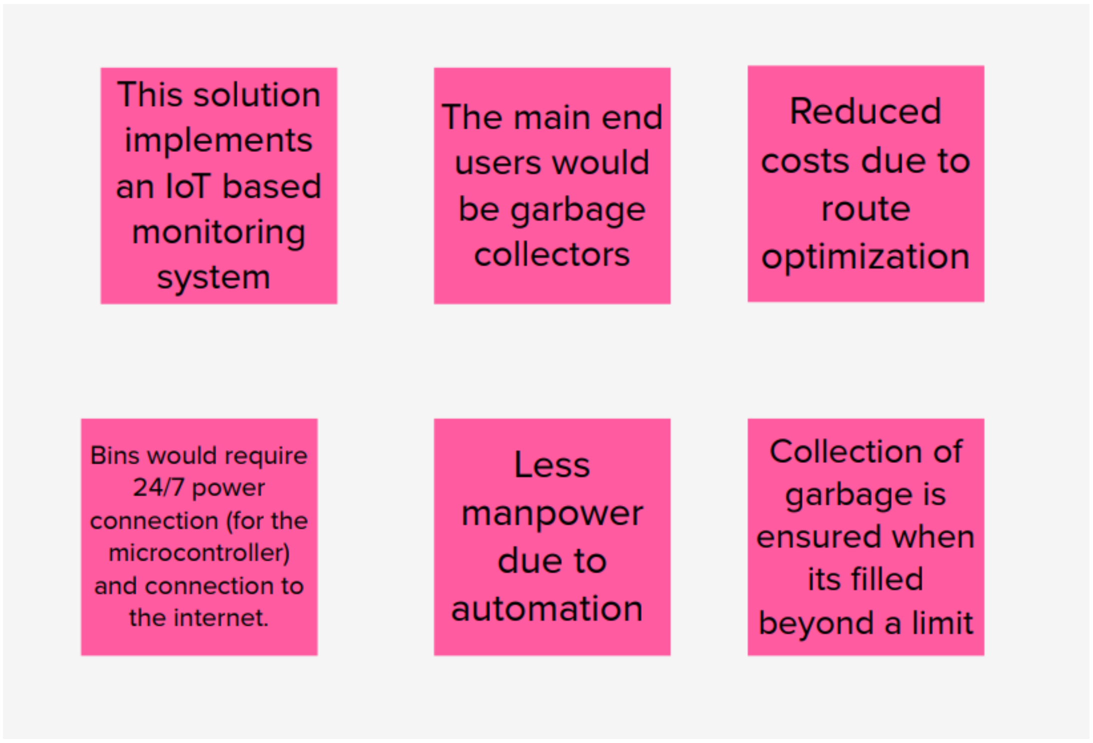
Ultrasonic sensor is used for measuring the level of waste in the bin

Deepikarani K:

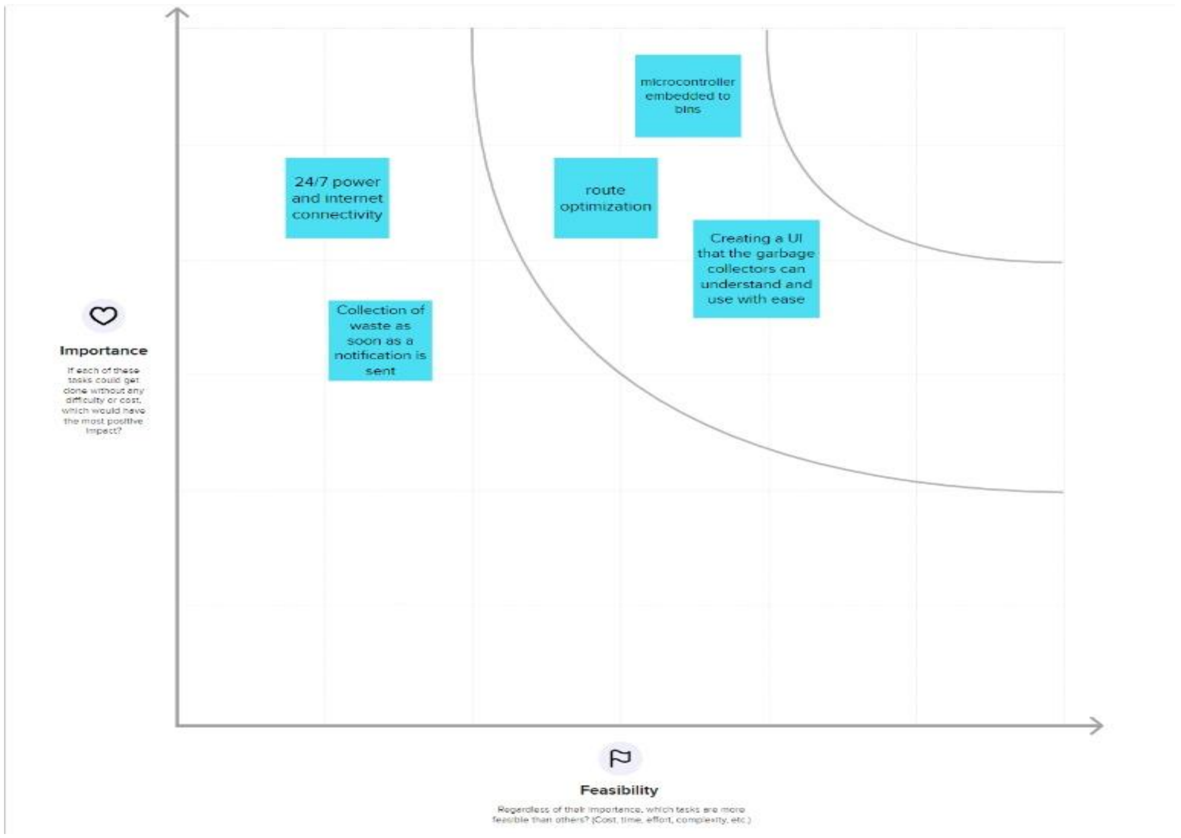
The message is sent to the cloud where it is stored

When there is a change in the data in the cloud, it is notified to the user via the web application

3.2.2 Group ideas


This solution implements an IoT based monitoring system
The main end users would be garbage collectors
Reduced costs due to route optimization
Bins would require 24/7 power connection (for the microcontroller) and connection to the internet.
Less manpower due to automation
Collection of garbage is ensured when its filled beyond a limit

3.2.3 Prioritize

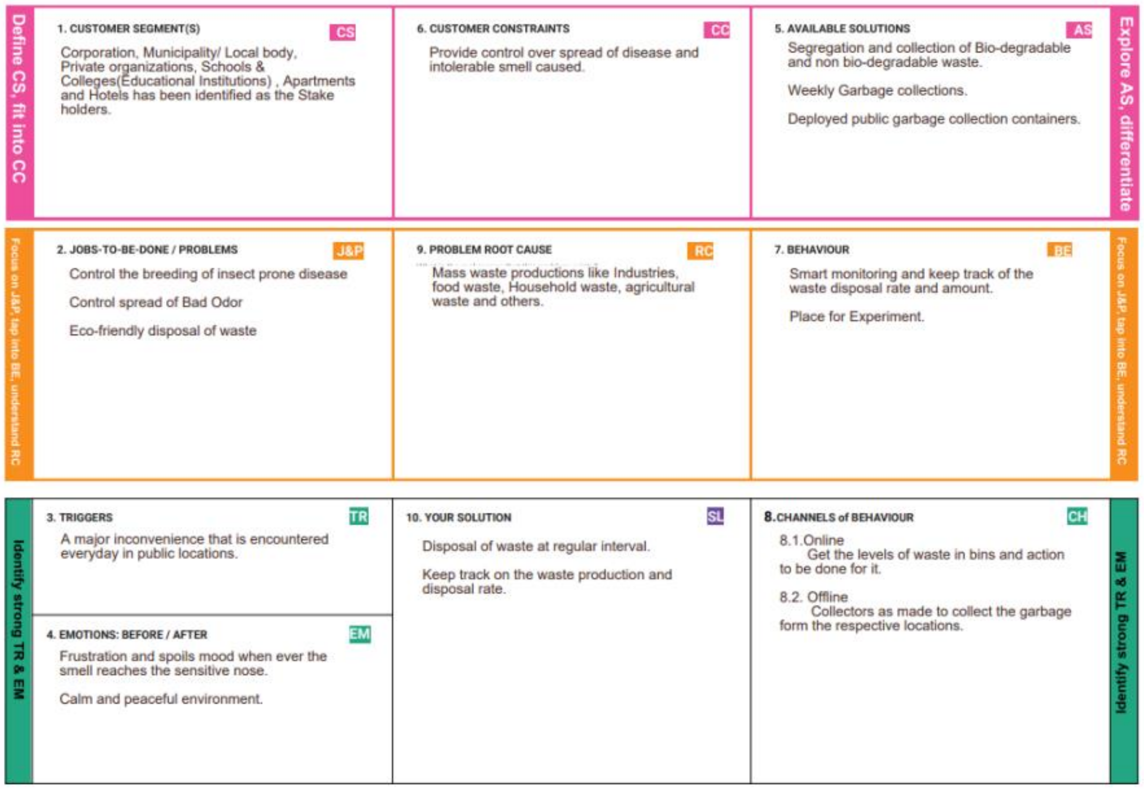


3.3 Proposed Solution

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 fire collecting recyclables like paper, glass,plastic. ❖ 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



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Corporation, Municipality/ Local body, Private organizations, Schools & Colleges(Educational Institutions) , Apartments and Hotels has been identified as the Stake holders.

6. CUSTOMER CONSTRAINTS CC

Provide control over spread of disease and intolerable smell caused.

5. AVAILABLE SOLUTIONS AS

Segregation and collection of Bio-degradable and non bio-degradable waste.

Weekly Garbage collections.

Deployed public garbage collection containers.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Control the breeding of insect prone disease

Control spread of Bad Odor

Eco-friendly disposal of waste

9. PROBLEM ROOT CAUSE RC

Mass waste productions like Industries, food waste, Household waste, agricultural waste and others.

7. BEHAVIOUR BE

Smart monitoring and keep track of the waste disposal rate and amount.

Place for Experiment.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

A major inconvenience that is encountered everyday in public locations.

4. EMOTIONS: BEFORE / AFTER EM

Frustration and spoils mood when ever the smell reaches the sensitive nose.

Calm and peaceful environment.

10. YOUR SOLUTION SL

Disposal of waste at regular interval.

Keep track on the waste production and disposal rate.

8.CHANNELS of BEHAVIOUR CH

8.1.Online
Get the levels of waste in bins and action to be done for it.

8.2. Offline
Collectors as made to collect the garbage form the respective locations.

Identify strong TR & EM

CHAPTER 4: REQUIRMENT ANALYSIS

4.1 Functional Requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
1	Detailed bin inventory.	All monitored bins can be seen on the map, such that the route can be optimized for the garbage collectors. Bins or stands are visible on the map as green, orange, or red circles. You can see bin details such as – capacity, last measurement, etc.
2	Real time bin monitoring.	The amount of fill is displayed in %, based on the garbage level and the tool predicts when the bin will become full, which is one the functionalities not included in the best waste management software. Sensors recognize picks as well; so, we 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.
3	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	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.
5	Eliminate inefficient picks.	Eliminate the collection of half-empty bins. The sensors recognize picks. By using real-time data on fill-levels and pick recognition, we can show you how full the bins you collect are.

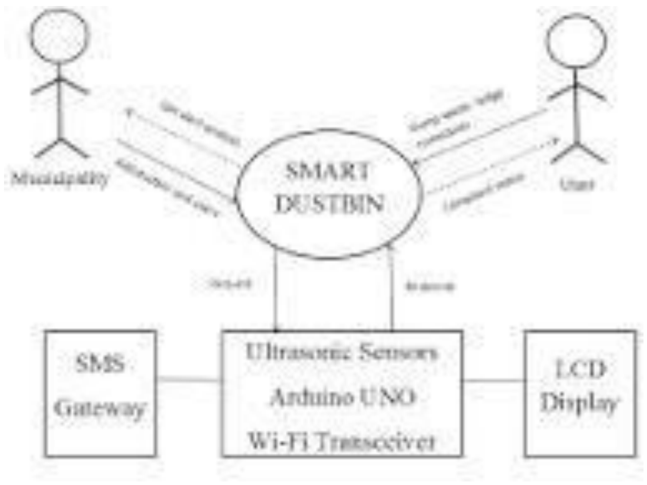
4.2 Non-functional Requirements

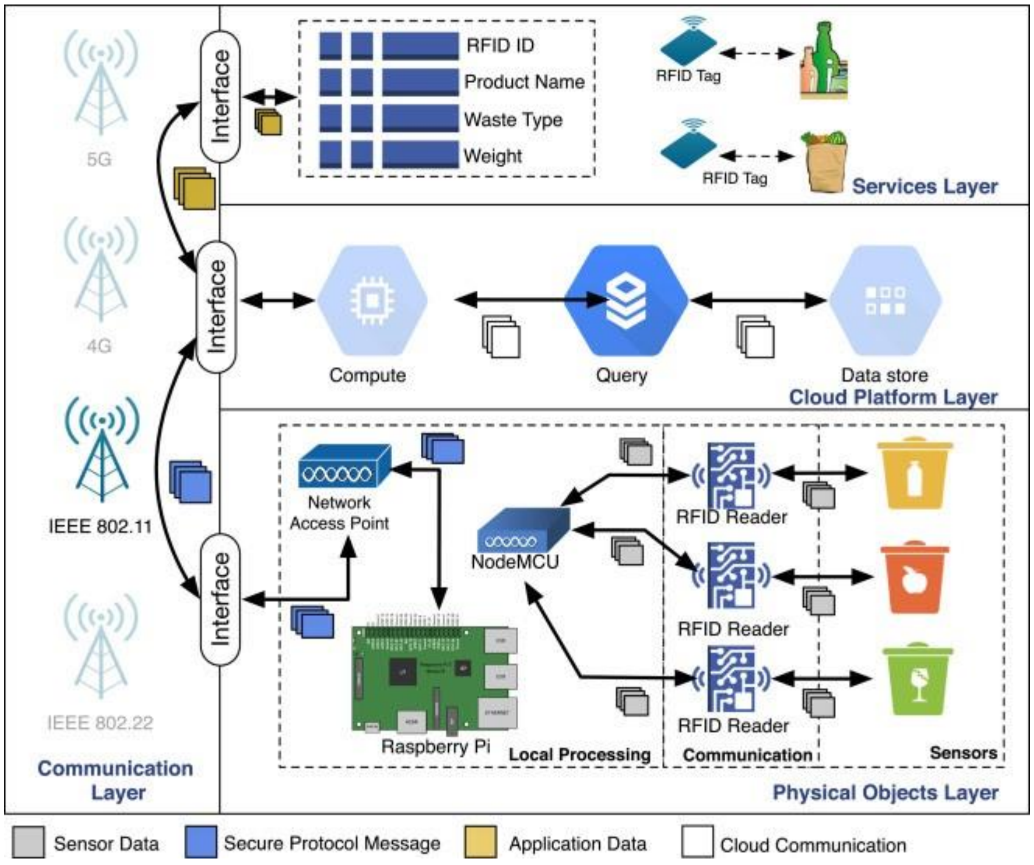
NFR No.	Non-Functional Requirement	Description
1	Usability	IoT device verifies that usability is a special and important perspective to analyze user requirements, which can further improve the design quality. In the design process with user experience as the core, the analysis of users' product usability can indeed help designers better understand users' potential needs in waste management, behavior and experience.
2	Security	• Use of reusable bottles • Use of reusable grocery bags • Purchase wisely and recycle • Avoid single use food and drink containers
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.
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 IBM Watson, that contains all the devices. 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%

CHAPTER 5: PROJECT DESIGN

5.1 Data Flow Diagram:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.





5.2 Solution and Technical Architecture

S.NO	COMPONENTS	DESCRIPTION	TECHNOLOGY
1.	User Interface	Web portal	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	To calculate the distance of dreck and show the real time level in web portal , information getting via ultra sonic sensor and the alert message activate with python script to web portal.	Java / Python
3.	Application Logic-2	To calculate the weight of the garbage and show the real time weight in web portal, this info getting via load cell and the alert message activate with python to web portal	Load cell/Python.
4.	Application Logic-3	Getting location of the Garbage.	GSM / GPS.
5.	Ultrasonic sensor	To throw alert message when garbage is getting full	Distance Recognition Model.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration:localhost Cloud ServerConfiguration:localhost,Firebase	Localhost,Web portal.

Application Characteristics:

S.NO	CHARACTERISTICS	DESCRIPTION	TECHNOLOGY
1.	Open-Source Frameworks	NodeRed,Python,IBM simulator	IOT
2.	Security Implementations	Raspberry Pi is connected to the internet and for example used to broadcast live data, further security measures are recommended	IOT
3.	Scalable Architecture	Raspberry pi:Specifications Soc: rspi ZERO W CPU: 32-bit computer with a 1 GHz ARMv6 RAM: 512MB Networking : Wi-Fi Bluetooth: Bluetooth 5.0, Bluetooth Low Energy (BLE). Storage: MicroSD GPIO: 40-pin GPIO header, populated Ports: micro HDMI 2.0, 3.5mm analogue audio-video jack, 2x USB 2.0, 2x USB	IOT

		3.0, Ethernet Dimensions: 88mm x 58mm x 19.5mm, 46g	
4.	Availability	These smart bins use sensors like ultrasonic and load cell to send alert message about the trash level recognition technology, and artificial intelligence, enabling them to automatically sort and categorize recycling litter into one of its smaller bin.	IOT
5.	Performance	Number of request :RPI manages to execute 129-139 read requests per second. Use of Cache:512mb Use of CDN's :Real time	IOT

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Web server	Login	USN-1	As a user, I give a user id and password for every workers and manage them.	I can manage web account	High	Sprint 1
Co admin	Login	USN-2	As a co admin, I monitor how wastes are filtering if garbage full, i give id to truck driver.	I monitor the garbage.	High	Sprint 1
Truck driver	Login	USN-3	As a truck driver, i follow te route send by co admin to collect the garbage.	I go to garbage filled place.	Low	Sprint 2

Local g arbage collector	Login	USN-4	As a garbage collector, i collect trash from garbage and load into truck and get them into landfill.	I collect wastes and send off to landfill.	Mediu m	Sprint 1
Municipal it y	Login	USN-5	As a municipalit y, i enquire te process are done properly.	I will manage all these process are going correctly.	High	Sprint 1

CHAPTER 6: PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation:

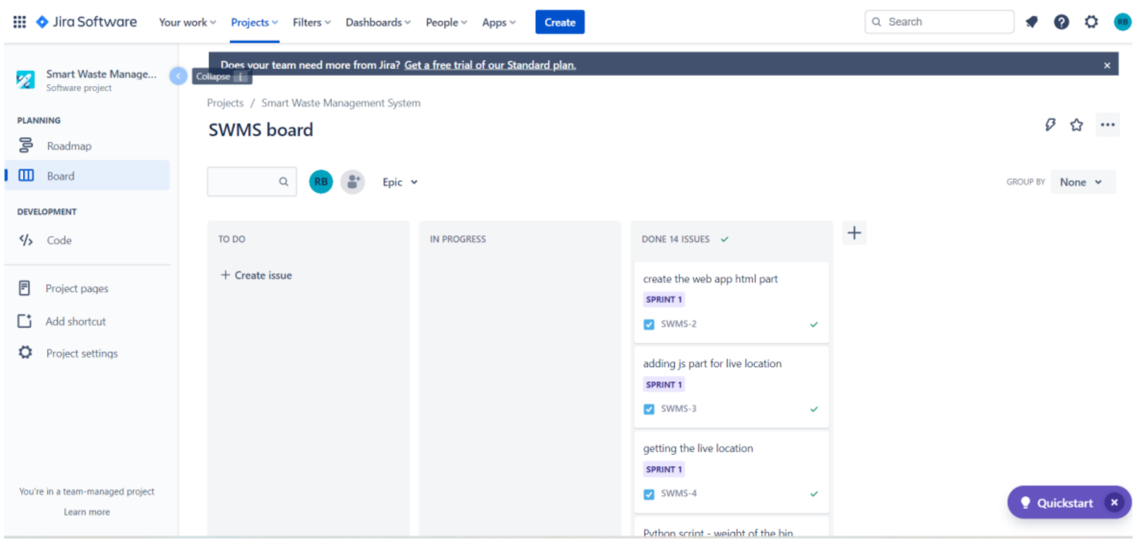
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Member
Sprint-1	Login	USN-1	As a user, I give a user id and password for every workers and manage them.	20	High	Deepak S
Sprint-1	Login	USN-2	As a co admin, I monitor how wastes are filtering if garbage full, i give id to truck driver.	10	High	Deepak S
Sprint-2	Login	USN-3	As a truck driver, i follow te route send by co admin to collect the garbage.	10	Low	Deepika Rani
Sprint-3	Login	USN-4	As a garbage collector, i collect trash from garbage and load into	20	Medium	Deva Dharshini

			truck and get them into landfill.			
Sprint -4	Login	USN-5	As a municipality , i enquire te process are done properly.	20	High	Deepak

6.2 Sprint Delivery Schedule:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	04 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	05 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	06 Nov 2022

6.3 Reports from JIRA:



Does your team need more from Jira? Get a free trial of our Standard plan.

Projects / Smart Waste Management System

SWMS board

TO DO

+ Create issue

IN PROGRESS

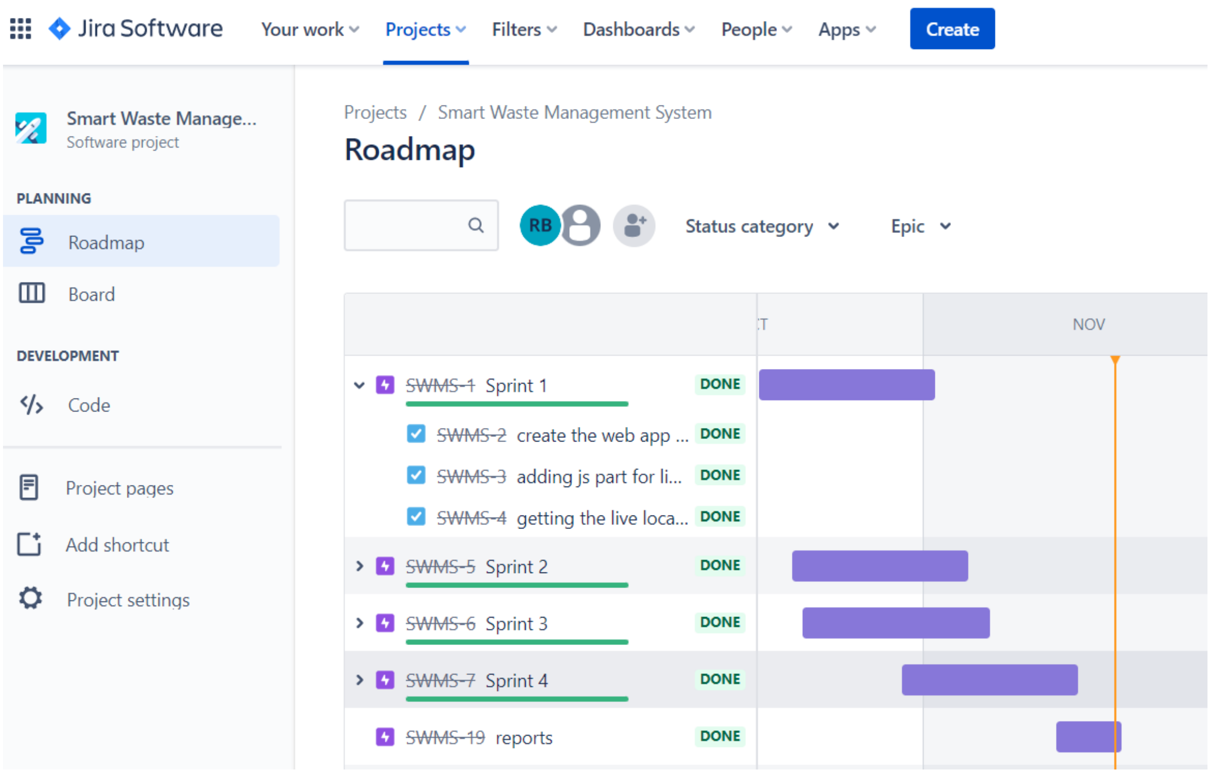
DONE 14 ISSUES

create the web app html part
SPRINT 1
SWMS-2

adding js part for live location
SPRINT 1
SWMS-3

getting the live location
SPRINT 1
SWMS-4

Python script - weight of the bin



Jira Software | Your work | Projects | Filters | Dashboards | People | Apps | Create

Smart Waste Manage...
Software project

PLANNING
- Roadmap
- Board

DEVELOPMENT
- Code

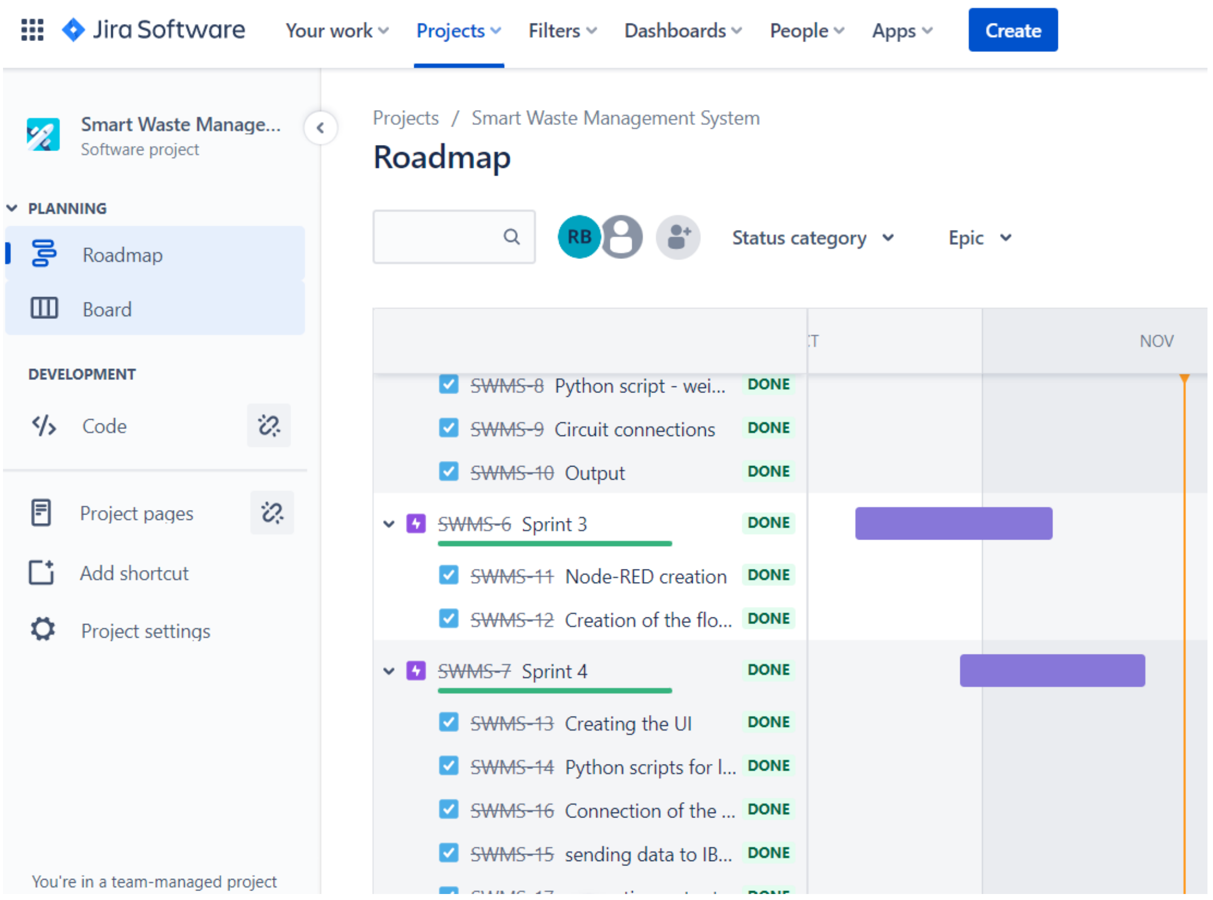
- Project pages
- Add shortcut
- Project settings

Projects / Smart Waste Management System

Roadmap

RB | Status category | Epic

		T	NOV
~~SWMS-1~~ Sprint 1	DONE		
~~SWMS-2~~ create the web app ...	DONE		
~~SWMS-3~~ adding js part for li...	DONE		
~~SWMS-4~~ getting the live loca...	DONE		
~~SWMS-5~~ Sprint 2	DONE		
~~SWMS-6~~ Sprint 3	DONE		
~~SWMS-7~~ Sprint 4	DONE		
~~SWMS-19~~ reports	DONE		



Jira Software | Your work | Projects | Filters | Dashboards | People | Apps | Create

Smart Waste Manage...
Software project

PLANNING
- Roadmap
- Board

DEVELOPMENT
- Code

- Project pages
- Add shortcut
- Project settings

You're in a team-managed project

Projects / Smart Waste Management System

Roadmap

RB | Status category | Epic

		T	NOV
~~SWMS-8~~ Python script - wei...	DONE		
~~SWMS-9~~ Circuit connections	DONE		
~~SWMS-10~~ Output	DONE		
~~SWMS-6~~ Sprint 3	DONE		
~~SWMS-11~~ Node-RED creation	DONE		
~~SWMS-12~~ Creation of the flo...	DONE		
~~SWMS-7~~ Sprint 4	DONE		
~~SWMS-13~~ Creating the UI	DONE		
~~SWMS-14~~ Python scripts for l...	DONE		
~~SWMS-16~~ Connection of the ...	DONE		
~~SWMS-15~~ sending data to IB...	DONE		

CHAPTER 7: CODING AND SOLUTIONING

7.1 Feature 1:

The main and first feature of the smart waste management is to get the live location of anyone who access the website for putting out a request for garbage collection in their locality. The live location is obtained as a result of the below code.

Source Code

```
#set GPIO direction (IN / OUT)
GPIO.setup(GPIO_TRIGGER,
GPIO.OUT) GPIO.setup(GPIO_ECHO,
GPIO.IN)
def distance():
# set Trigger to HIGH
GPIO.output(GPIO_TRIGGER,
True) # set Trigger after 0.01ms
to LOW time.sleep(0.00001)
GPIO.output(GPIO_TRIGGER,
False) StartTime = time.time()
StopTime =
time.time() # save
StartTime
while GPIO.input(GPIO_ECHO) == 0:
StartTime =
time.time() # save
time of arrival
while GPIO.input(GPIO_ECHO) == 1:
StopTime = time.time()
# time difference between start and
arrival TimeElapsed = StopTime –
StartTime
# multiply with the sonic speed (34300
cm/s) # and divide by 2, because there
and back distance = (TimeElapsed *
34300) / 2
retun distance
23
if__name__== '.main_':
try:
```

```
while True:
dist = distance()
print ("Measured Distance = %.1f cm" % dist)
percent = (100.0 - (dist * 100/40.0))
url =
"http://localhost:80/demoaddbin.php?bin_id=1&percent_filled="+st
r(percent) x= urllib.urlopen(url)
print(x.read
)
time.sleep(5
)
# Reset by pressing CTRL+ C except
KeyboardInterrupt:
print("Measurement stopped by
User") GPIO.cleanup()
```

PROGRAM CODE FOR ACCESS DATABASE:

```
package com.bin;
import android.app.NotificationManager;
import android.app.PendingIntent; import android.app.Service;
import
android.content.Context;
import
android.content.Intent;
import
android.content.SharedPreferences;
import
android.media.RingtoneManager;
import android.net.Uri;
import
android.os.AsyncTask;
import android.os.Handler;
import android.os.IBinder;
import android.support.annotation.Nullable;
import
android.support.v4.app.NotificationCompat;
import android.util.Log;
import android.widget.Toast;
import org.json.JSONArray;
import
org.json.JSONException;
import org.json.JSONObject;
import
java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import
java.io.InputStreamReader;
import
java.net.HttpURLConnection;
import java.net.URL;

public class GetData extends Service {
String BASE_URL = "http://dustbin.000webhostapp.com/";
String POPMOVIES_BASE_URL = BASE_URL + "getresponsefrombin.php";
SharedPreferences preferences;
SharedPreferences.Editor
editor;




@Override
public void onCreate() {
Toast.makeText(this, "Service Called",
Toast.LENGTH_SHORT).show(); Log.d("Create:","called");
//addNotification(); super.onCreate();
}
@Override
public int onStartCommand(Intent intent, int flags, int startId) {
Log.d("onStartCommand:","called");
preferences =
getSharedPreferences("DustBin",MODE_PRIVATE); final
Handler handler = new Handler();
Runnable runnable = new Runnable() {
```

```
@Override
public void
run() {
Log.d("handler","run();");
new
DustbinTask().execute();
handler.postDelayed(this, 5000);
}
};
//Start
handler.postDelayed(runnable,
1000); return START_STICKY;
}
@Override
  public void
  onDestroy() {
  Log.d("Destroy:","cal
  led"); 28
super.onDestroy();
  }
@Nullable @Override
public IBinder onBind(Intent intent) {
Log.d("Bind:","called");
return null;
}
```

```
public class DustbinTask extends AsyncTask<Void,
Void,Void>{ @Override
protected                       void
onPreExecute()                  {
Log.d("onPreExecute","initi
ate"); try {
if (!new
Network(GetData.this).isConnected()) {
Log.d("onPreExecute","No Internet
Available!!"); cancel(true);
}
}
catch (InterruptedException | IOException e)
{ e.printStackTrace();
}
@Override
protected Void doInBackground(Void...
params) { HttpURLConnection urlConnection
= null; BufferedReader reader = null;
URL url;
String
MoviesJsonStr; try
{
url = new URL(POPMOVIES_BASE_URL);
urlConnection = (HttpURLConnection)
url.openConnection();
urlConnection.setRequestMethod("GET");
urlConnection.connect();
InputStream inputStream =
urlConnection.getInputStream(); 29
StringBuilder buffer = new StringBuilder()
reader = new BufferedReader(new
InputStreamReader(inputStream)); String line;
while ((line = reader.readLine()) !=
null) {
buffer.append(line).append("\n");
}
MoviesJsonStr = buffer.toString();
getMovieNames(MoviesJsonStr);
} catch (IOException | JSONException
e1) { e1.printStackTrace();
} finally {
if (urlConnection !=
null) {
urlConnection.disconnec
t();
}
if (reader !=
null) { try {
reader.close();
} catch (final IOException ignored) {}
}
}
return null;
}
}
private void getMovieNames(String MovieJsonStr) throws
JSONException { JSONObject MovieJson = new
JSONObject(MovieJsonStr);
JSONArray movieLists =
MovieJson.getJSONArray("bin_info"); for (int i = 0; i <
movieLists.length(); i++) {
```

```
JSONObject jMovieDetails =
movieLists.getJSONObject(i); String name =
jMovieDetails.getString("bin_id");
int id =
jMovieDetails.getInt("percent_filled");
Log.d("DATA", name + " " + id);
MainActivity.percent = id;
if(id>=80
){ 30
addNotification(id);
}
}
/Log.v("Length: ", String.valueOf(movieLists.length()));
//Show a notification
private void addNotification(int
id) { int min, max;
int percentage =
preferences.getInt("last_percent",0); min =
percentage - 5;
max = percentage +
5; if (min > id || id >
max) {
Intent intent = new Intent(this,
MainActivity.class
intent.setFlags(Intent.FLAG_ACTIVITY_CLEA
R_TOP); editor = preferences.edit();
editor.putInt("last_percent",id);
editor.apply();
PendingIntent pendingIntent = PendingIntent.getActivity(this, 0/*Request
code*/, intent, PendingIntent.FLAG_ONE_SHOT);
//Set sound of
notification Uri
notificationSound =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
NotificationCompat.Builder notifiBuilder = new
NotificationCompat.Builder(this)
.setSmallIcon(R.mipmap.ic_launcher
.setContentTitle(id + "% Dustbin Full")
```

```
.setContentText("Please clear your trash")
.setAutoCancel(true)
.setSound(notificationSound)
.setContentIntent(pendingIntent);
NotificationManager notificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
notificationManager.notify(999 /*ID of notification*/,
notifiBuilder.build());
//stopSelf()
```

PROGRAM FOR CONNECTING APPLICATION TO INTERNET:

```
package com.bin;
import android.content.Context;
import
android.net.ConnectivityManager;
import android.util.Log;
import java.io.IOException;
class Network {
private Context
mContext;
Network(Context
mContext) {
this.mContext =
mContext;
private boolean isNetworkAvailable() {
final ConnectivityManager connectivityManager =
((ConnectivityManager)
mContext.getSystemService(Context.CONNECTIVITY_SERVI
CE)); return connectivityManager.getActiveNetworkInfo() !=
null &&
connectivityManager.getActiveNetworkInfo().isConnected();
}
boolean isConnected() throws InterruptedException, IOException
{
if (isNetworkAvailable()) {
String command = "ping -c 1 google.com";
return (Runtime.getRuntime().exec (command).waitFor() == 0);
}
```

CHAPTER 8: TESTING

8.1 Test cases:

Unit testing

TEST CASE ID	FEATURE TYPE	COMPONENT	TEST SCENARIO	STEPS TO EXECUTE	TEST DATA	EXPECTED RESUL T	ACTUAL RESUL T	STATUS	COMMENTS	EXECUTED BY
LOGIN PAGE_ TC_001	FUNCTIONAL	HOME PAGE	VERIFY THE USER IS ABLE TO SEE THE LOGIN/SIGN UP WEN USER CLICK ON MY ACCOUNT BUTTON	1.ENTER URL AND CLICK GO 2.VERIFY LOGIN/SIGN UP	https://169.51.204.219.30106	Login page is visible	Working as expected	PASS	Successful	Deepak S
LOGIN PAGE_ TC_002	UI	HOME PAGE	VERIFY THE USER IS ABLE TO SEE THE LOGIN/SIGN UP WEN USER CLICK ON MY ACCOUNT BUTTON	1.ENTER URL AND CLICK GO 2.VERIFY LOGIN/SIGN UP Elements a.ID text box B. password	https://169.51.204.219.30106	Application should show below UI element	Working as expected	PASS	Successful	Deepak S

				text box c..login button D.new user E.already have an account						
LOGIN PAGE_ TC_003	FUNCTIONAL	LOGIN PAGE	VERIFY THE USER IS ABLE TO SEE THE LOGIN/SIGN 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:1111 password:5678	User should navigate your home page.	Working as expected	PASS	Successful	Deva Dharshini S
LOGIN PAGE_	FUNCTIONAL	LOGIN	VERIFY	and click	Id:1111	Confirmati	working as	PASS	Successful	Deepak S

TC_004		PAGE	THE USER IS ABLE TO SEE THE LOGIN/SIG N UP WEN USER CLICK ON MY ACCOUNT BUTTON	go 2.click on my account 3.Enter valid ID 4.Enter valid password 5.click on login button	password :5678	on of message sent	expected			

LOGIN PAGE_TC _005	UI	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:1111 passwod: 5678	Confirmation message sent	Working as expected	PASS	Successful	Deva Dharshini S

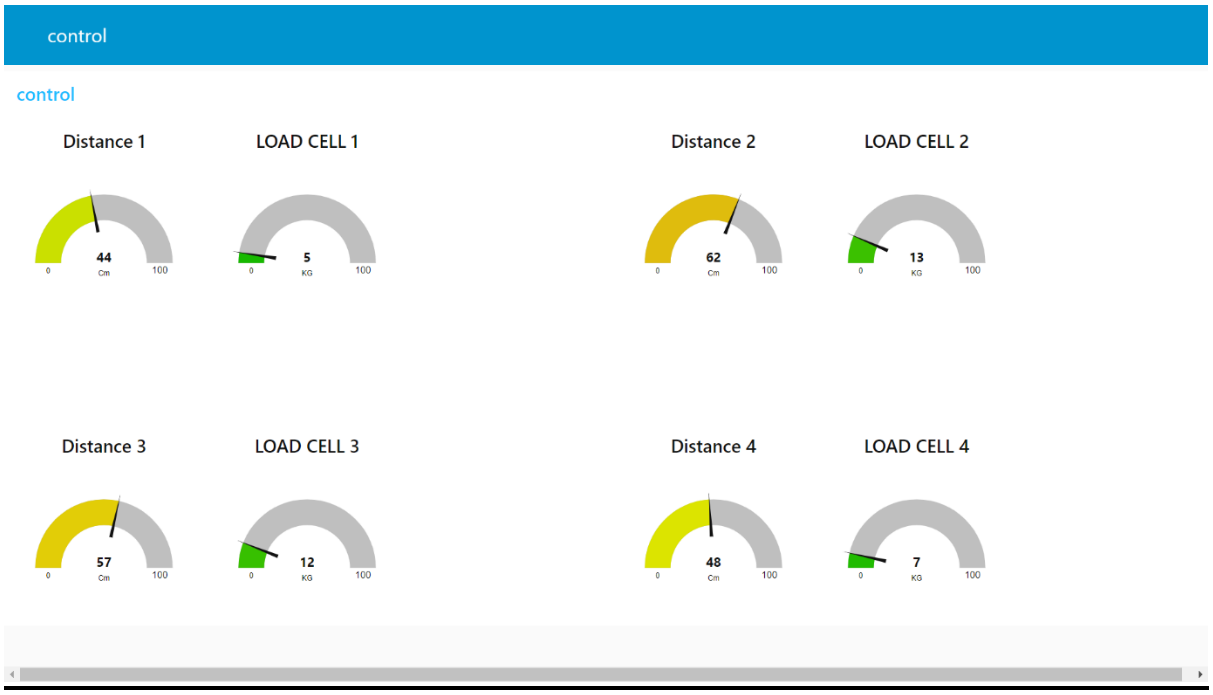
LOGIN PAGE_TC _006	FUNCTION A L	LOGIN PAGE FOR ADMIN	VERIFY THE USER IS ABLE TO SEE THE LOGIN/SIG N UP WEN USER CLICK ON MY ACCOUNT BUTTON	1.enter url and click go 2.click on my account 3.Enter valid ID 4.Enter valid password 5.click on login button	Id:1111 password :5678	Customer database is visible	Working as expected	PASS	Successful	Deepika Rani K

8.2 User Acceptance testing

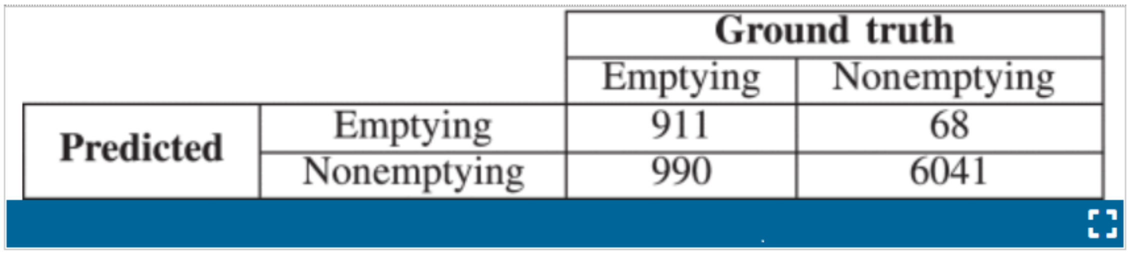
Acceptance testing - is the final phase of product testing prior to public launch. A level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system’s compliance with the business requirements and assess whether it is acceptable for delivery.

CHAPTER 9: RESULTS

Sample output:

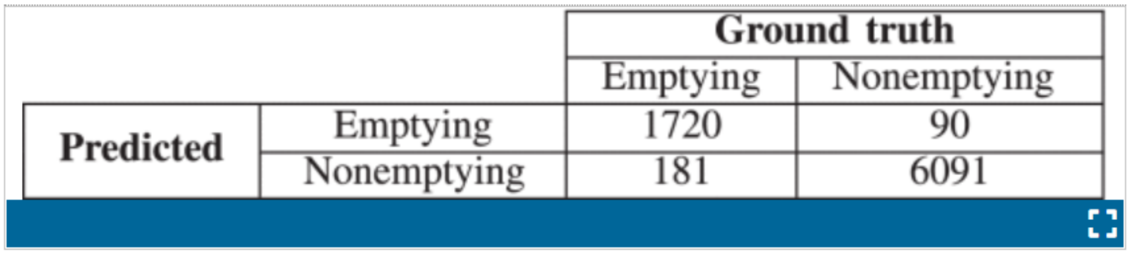


Before implementation of Smart waste management system



		Ground truth	
		Emptying	Nonemptying
Predicted	Emptying	911	68
	Nonemptying	990	6041

After implementation of smart waste management system



		Ground truth	
		Emptying	Nonemptying
Predicted	Emptying	1720	90
	Nonemptying	181	6091

CHAPTER 10: ADVANTAGES AND DISADVANTAGES

10.1 Advantages:

- Intelligent compaction of waste by monitoring fill level in real-time using sensors.
- It keeps our surroundings clean and keeps free from bad odour.
- Reduces manpower requirement to handle the garbage collection
- Emphasizes of healthy environment and keep the cities cleaner and more beautiful.
- It reduces infrastructure, operating and maintenance costs by upto 30%.
- Increases recycling rate of waste.

10.2 Disadvantages:

- Initial large-scale implementation takes cost.
- System requires more number waste bins for separate waste collection.
- Wireless technologies used should have proper connections as they have shorter range and lower data speed
- Training programs should be provided to people involving in the ecosystem of smart waste management.
- Sensors may encounter damage so it should be kept under protective ambience to prevent the damage.
- Replacement of sensors require knowledgeable people and thus acknowledgement of malfunction of sensor.

CHAPTER 11: CONCLUSION

Improper disposal and improper maintenance of domestic waste create issues in public health and environment pollution thus this paper attempts to provide practical solution towards managing the waste collaborating it with the use of IOT. by using the smart waste management system, we can manage waste properly we are also able to sort the Bio-degradable and non-Biodegradable waste properly which reduces the pollution in the environment. Various waste management initiatives taken for human well-being and to improve the TWM practices were broadly discussed in this chapter. The parameters that influence the technology and economic aspects of waste management were also discussed clearly. Different types of barriers in TWM, such as economic hitches, political issues, legislative disputes, informative and managerial as well as solutions and success factors for implementing an effective management of toxic organic waste within a globular context, were also discussed giving some real examples. The effect of urbanization on the environmental degradation and economic growth was also discussed. The proposed system will help to overcome all the serious issues related to waste and keep the environment clean.

CHAPTER 12: FUTURE WORK

Based on the real-time and historical data collected and stored in the cloud waste collection schedules and routes can be optimized. Predictive analytics could be used to make decisions ahead of time and offers insight into waste bin locations. Graph theory optimization algorithms can be used to manage waste collection strategies dynamically and efficiently. Every day, the workers can receive the newly calculated routes in their navigation devices. The system can be designed to learn from experience and to make decisions not only on the daily waste level status but also on future state forecast, traffic congestion, balanced cost-efficiency functions, and other affecting factors that a priori humans cannot foresee.

Garbage collectors could access the application on their mobile phone/tablets using the internet. Real-time GPS assistance can be used to direct them to the pre-decided route. As they go collecting the garbage from the containers, the management is also aware of the progress as the vehicle, as well as the garbage containers, are traced in real-time. The management staff gets their own personalized administration panel over a computer/tablet which gives them a bird eye view over the entire operations.

An alternative solution using image processing and camera as a passive sensor could be used. But, the cost of those image processing cameras is higher as compared to the ultrasonic sensors, which leads to high solution implementation cost.

CHAPTER 13: APPENDIX

13.1 Source Code:

```
#set GPIO direction (IN / OUT)
GPIO.setup(GPIO_TRIGGER,
GPIO.OUT) GPIO.setup(GPIO_ECHO,
GPIO.IN)
def distance():
# set Trigger to HIGH
GPIO.output(GPIO_TRIGGER,
True) # set Trigger after 0.01ms
to LOW time.sleep(0.00001)
GPIO.output(GPIO_TRIGGER,
False) StartTime = time.time()
StopTime =
time.time() # save
StartTime
while GPIO.input(GPIO_ECHO) == 0:
StartTime =
time.time() # save
time of arrival
while GPIO.input(GPIO_ECHO) == 1:
StopTime = time.time()
# time difference between start and
arrival TimeElapsed = StopTime –
StartTime
# multiply with the sonic speed (34300
cm/s) # and divide by 2, because there
and back distance = (TimeElapsed *
34300) / 2
retun distance
23
if__name__== '_main_':
try:
while True:
dist = distance()
print ("Measured Distance = %.1f cm" % dist)
percent = (100.0 - (dist * 100/40.0))
```

```
url =
"http://localhost:80/demoaddbin.php?bin_id=1&percent_filled="+st
r(percent) x= urllib.urlopen(url)
print(x.read
)
time.sleep(5
)
# Reset by pressing CTRL+ C except
KeyboardInterrupt:
print("Measurement stopped by
User") GPIO.cleanup()
```

PROGRAM CODE FOR ACCESS DATABASE:

```
package com.bin;
import android.app.NotificationManager;
import android.app.PendingIntent; import android.app.Service;
import
android.content.Context;
import
android.content.Intent;
import
android.content.SharedPreferences;
import
android.media.RingtoneManager;
import android.net.Uri;
import
android.os.AsyncTask;
import android.os.Handler;
import android.os.IBinder;
import android.support.annotation.Nullable;
import
android.support.v4.app.NotificationCompat;
import android.util.Log;
import android.widget.Toast;
import org.json.JSONArray;
import
org.json.JSONException;
import org.json.JSONObject;
```

```
import
java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import
java.io.InputStreamReader;
import
java.net.HttpURLConnection;
import java.net.URL;

public class GetData extends Service {
String BASE_URL = "http://dustbin.000webhostapp.com/";
String POPMOVIES_BASE_URL = BASE_URL + "getresponsefrombin.php";
SharedPreferences preferences;
SharedPreferences.Editor
editor;
@Override
public void onCreate() {
Toast.makeText(this, "Service Called",
Toast.LENGTH_SHORT).show(); Log.d("Create:","called");
//addNotification(); super.onCreate();
}



@Override
public int onStartCommand(Intent intent, int flags, int startId) {
Log.d("onStartCommand:","called");
preferences =
getSharedPreferences("DustBin",MODE_PRIVATE); final
Handler handler = new Handler();
Runnable runnable = new Runnable() {


@Override
public void
run() {
```

```
Log.d("handler","run();");
new
DustbinTask().execute();
handler.postDelayed(this, 5000);
}
};
//Start
handler.postDelayed(runnable,
1000); return START_STICKY;
}
@Override
  public void
  onDestroy() {
  Log.d("Destroy:","cal
  led"); 28
super.onDestroy();
  }
@Nullable @Override
public IBinder onBind(Intent intent) {
Log.d("Bind:","called");
return null;
}
```

```
public class DustbinTask extends AsyncTask<Void,
Void,Void>{ @Override
protected                    void
onPreExecute()               {
Log.d("onPreExecute","initi
ate"); try {
if (!new
Network(GetData.this).isConnected()) {
Log.d("onPreExecute","No Internet
Available!!"); cancel(true);
}
}
catch (InterruptedException | IOException e)
{ e.printStackTrace();
}
@Override
protected Void doInBackground(Void...
params) { HttpURLConnection urlConnection
= null; BufferedReader reader = null;
URL url;
String
MoviesJsonStr; try
{
url = new URL(POPMOVIES_BASE_URL);
urlConnection = (HttpURLConnection)
url.openConnection();
urlConnection.setRequestMethod("GET");
urlConnection.connect();
InputStream inputStream =
urlConnection.getInputStream();
StringBuilder buffer = new StringBuilder()
reader = new BufferedReader(new
InputStreamReader(inputStream)); String line;
while ((line = reader.readLine()) !=
null) {
buffer.append(line).append("\n");
}
MoviesJsonStr = buffer.toString();
getMovieNames(MoviesJsonStr);
} catch (IOException |
JSONException e1) {
e1.printStackTrace();
} finally {
if (urlConnection !=
null) {
urlConnection.disconn
ect();
}
if (reader !=
null) { try {
reader.close();
} catch (final IOException ignored) {}
}
}
return null;
}
}
private void getMovieNames(String MovieJsonStr) throws
JSONException { JSONObject MovieJson = new
JSONObject(MovieJsonStr);
JSONArray movieLists =
MovieJson.getJSONArray("bin_info"); for (int i = 0; i <
movieLists.length(); i++) {
JSONObject jMovieDetails =
movieLists.getJSONObject(i); String name =
jMovieDetails.getString("bin_id");
int id =
jMovieDetails.getInt("percent_filled")
; Log.d("DATA", name + " " + id);
MainActivity.percent = id;
if(id>=8
0){ 30
addNotification(id);
}
}
/Log.v("Length: ", String.valueOf(movieLists.length()));
//Show a notification
private void
addNotification(int id) { int
min, max;
int percentage =
preferences.getInt("last_percent",0); min =
percentage - 5;
max = percentage +
5; if (min > id || id
> max) {
Intent intent = new Intent(this,
MainActivity.class
intent.setFlags(Intent.FLAG_ACTIVITY_CLE
AR_TOP); editor = preferences.edit();
editor.putInt("last_percent",id);
editor.apply();
PendingIntent pendingIntent = PendingIntent.getActivity(this, 0/*Request
code*/, intent, PendingIntent.FLAG_ONE_SHOT);
//Set sound of
notification Uri
notificationSound =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
NotificationCompat.Builder notifiBuilder = new
NotificationCompat.Builder(this)
.setSmallIcon(R.mipmap.ic_launcher
.setContentTitle(id + "% Dustbin Full")
.setContentText("Please clear your trash")
.setAutoCancel(true)
.setSound(notificationSound)
.setContentIntent(pendingIntent);
NotificationManager notificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
notificationManager.notify(999 /*ID of notification*/,
notifiBuilder.build());
//stopSelf()
```

PROGRAM FOR CONNECTING APPLICATION TO INTERNET:

```
package com.bin;
import android.content.Context;
import
android.net.ConnectivityManager;
import android.util.Log;
import java.io.IOException;
class Network {
private Context
mContext;
Network(Context
mContext) {
this.mContext =
mContext;
private boolean isNetworkAvailable() {
final ConnectivityManager connectivityManager =
((ConnectivityManager)
mContext.getSystemService(Context.CONNECTIVITY_SER
VICE)); return connectivityManager.getActiveNetworkInfo()
!= null &&
connectivityManager.getActiveNetworkInfo().isConnected();
}
boolean isConnected() throws InterruptedException, IOException
{
if (isNetworkAvailable()) {
String command = "ping -c 1 google.com";
return (Runtime.getRuntime().exec (command).waitFor() == 0);
}
```

13.2 GitHub & Project Demo Link:

GitHub Link:

https://github.com/IBM-EPBL/IBM-Project-24924-1659950959

Project Demo Link:

https://drive.google.com/file/d/1w7ylAcORZpDv5lUvACoKfpmNkwT1l9ve/view?usp=sharing