PROJECT REPORT

Smart Waste Management System For Metropolitan Cities

Team id: PNT2022TMID34215

Team Leader	VINIKADEVI S P
Team Member 1	PRATHISHA P
Team Member 2	RESHMI G M
Team Member 3	SUNMATHI K

1. INTRODUCTION

 Smart waste management system in metropolitan towns is approximately the usage of generation and facts to create a extra efficient waste enterprise primarily based totally on IOT generation.

1.1Project Overview:

 The venture is primarily based totally on a real-time clever rubbish bin mechanism for stable waste control in clever towns.

1.2 Purpose:

 A waste control machine is the approach an company makes use of to dispose reduce, reuse and prevent waste.

2. LITERATURE SURVEY

2.1 EXISTING SYSTEM

Heavy metals and other toxic compounds from landfills, pollution.

2.2 REFERENCES

1.Smart Waste Management: Garbage Monitoring Using Iot 1Mrs Sarmila SS , 2 Siva Kumar V, V3asanth Kumaur P K 1Assistant Professor .Department of Computer Science and Engineering K.L.N. College of Engineering Madurai, India ISSN: 2348 – 8387 (APRIL ,2018).

2.Review Paper on Implementation of Automatic Waste Management System Using IOT & Android for Smart Cities Pulkit Bindal1 , Utkarsh Srivastava2 , Chirag Agarwal3 , Himanshu Gupta4 , Chhaya Sharma5 1,2,3,4

Department of Computer Science and Engineering, Raj Kumar Goel Institute of Technology, GhaziaD ISSN: 2349-6002 (MAY 2022) .

3.Location Based Garbage Management System for Smart City Harini P K S1 , Ramya S1 , Yamini R2 1 Student, Dept. of Computer Science and Engineering, Adhiyamaan College of Engineering, Hosur, India (november-2020).

4.IoT Enabled Smart Waste Bin with Real Time Monitoring for efficient waste management in Metropolitan Cities Manju Mohan1 , RM. Kuppan Chetty1 , Vijayram Sriram2 , Mohd. Azeem2 , P. Vishal2 and G. Pranav2 1Centre for Automation and Robotics (ANRO), School of Mechanical Sciences, Hindustan Institute of Technology and Science, Padur, Chennai –603103 ISSN : 2619- 8150 Volume 1, Number 3, (September 2019).

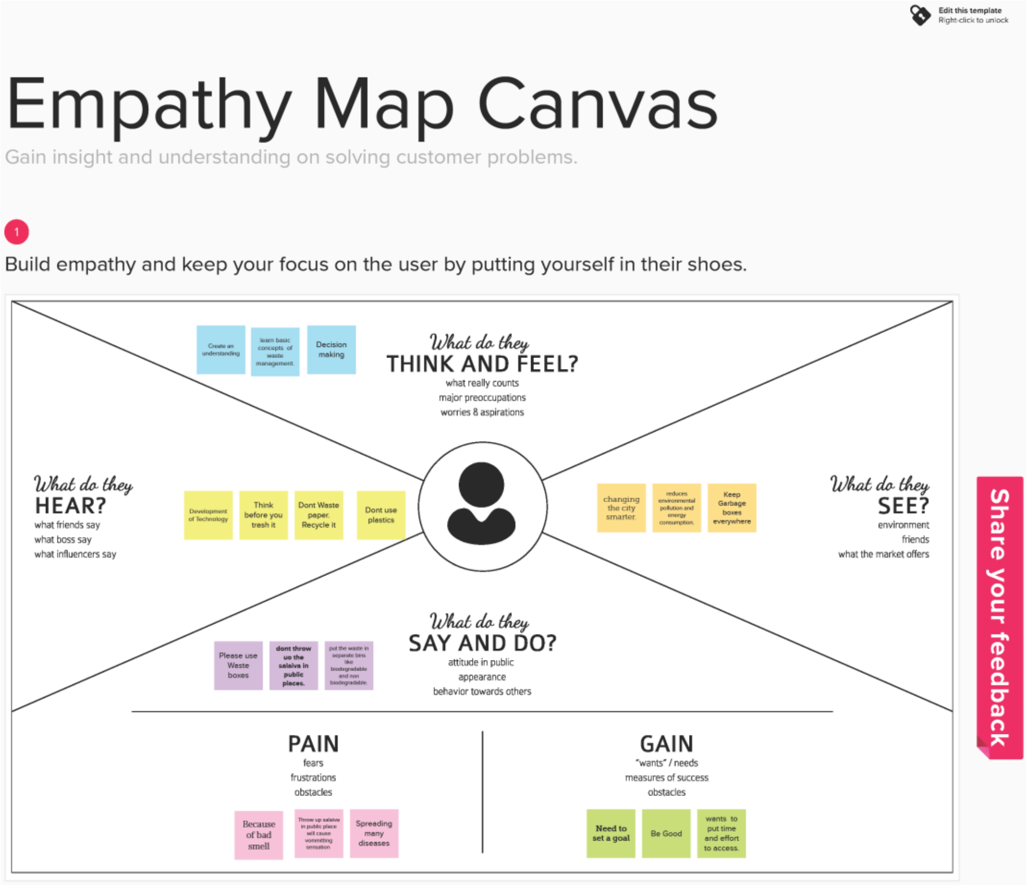
5.Smart Waste Management System using IOT Tejashree Kadus1 , Pawankumar Nirmal2 , Kartikee Kulkarni3 Department of Mechanical Engineering MIT Academy of Engineering, Pune Savitribai Phule University (April 2020).

2.3 PROBLEM STATEMENT

Indiscriminate disposal of waste is a major issue in most developing countries' urban centers and poses a serious threat to the healthy living of the citizens. The fill level of waste in each of the containers, which are strategically situated across the communities, is detected using sensors.

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



Edit this template
Right-click to unlock

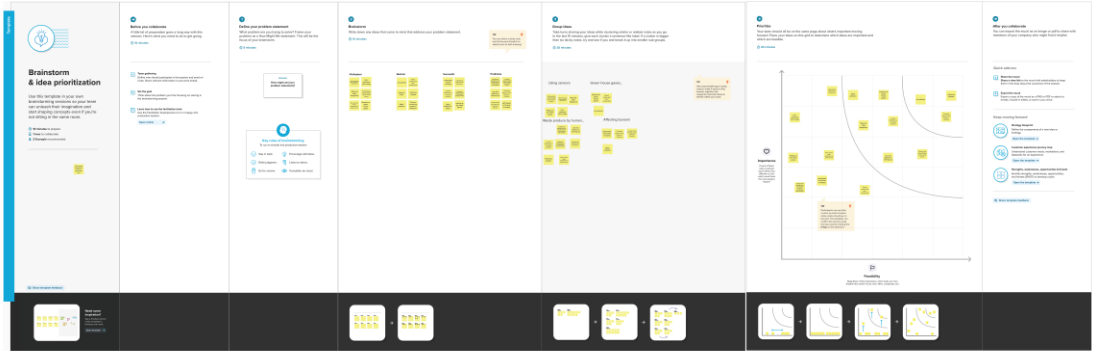
Empathy Map Canvas

Gain insight and understanding on solving customer problems.

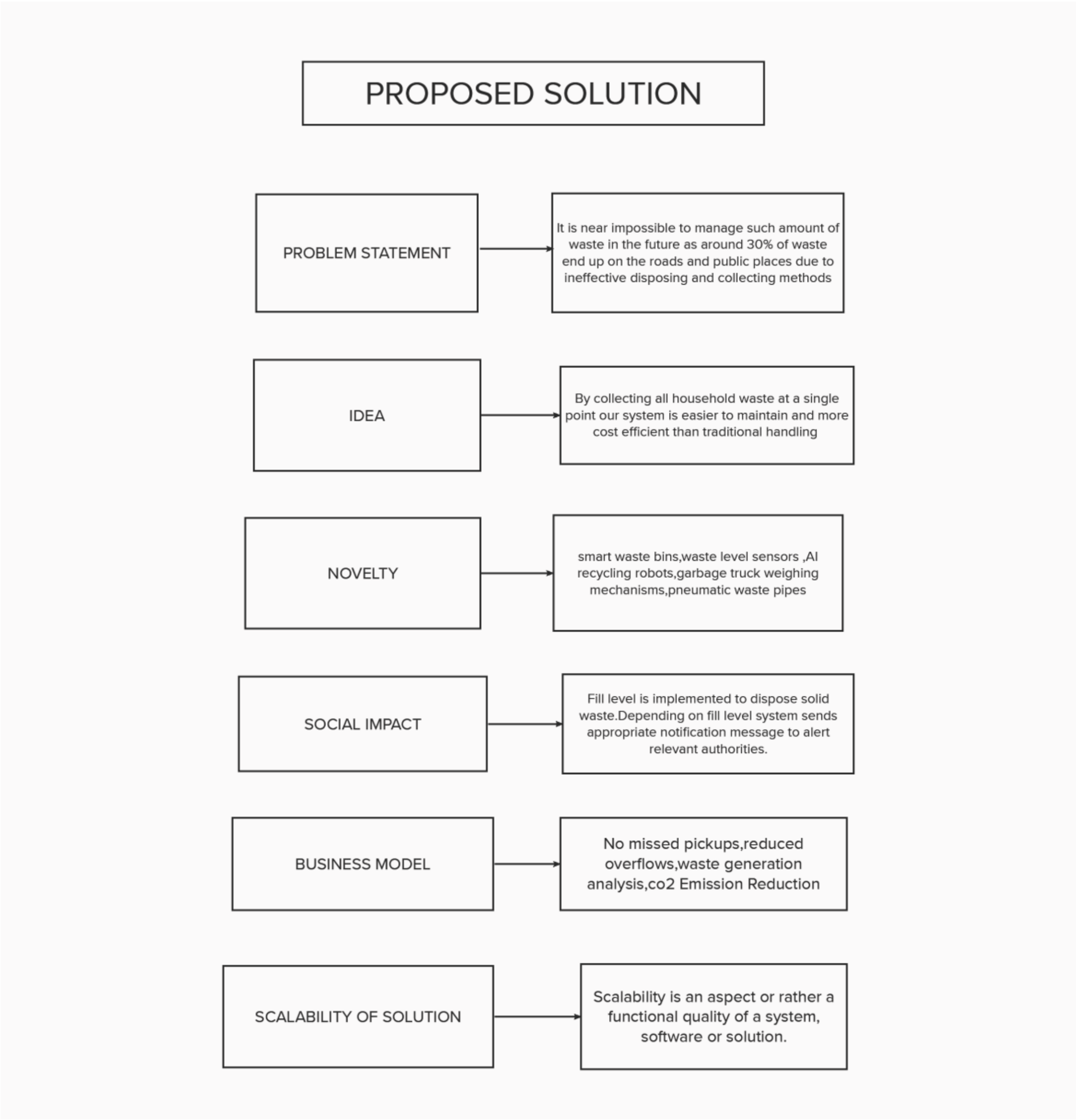
1

Build empathy and keep your focus on the user by putting yourself in their shoes.

3.2 IDEATION AND BRAINSTROMING



3.3 PROPOSED SOLUTION



PROPOSED SOLUTION

PROBLEM STATEMENT → It is near impossible to manage such amount of waste in the future as around 30% of waste end up on the roads and public places due to ineffective disposing and collecting methods

IDEA → By collecting all household waste at a single point our system is easier to maintain and more cost efficient than traditional handling

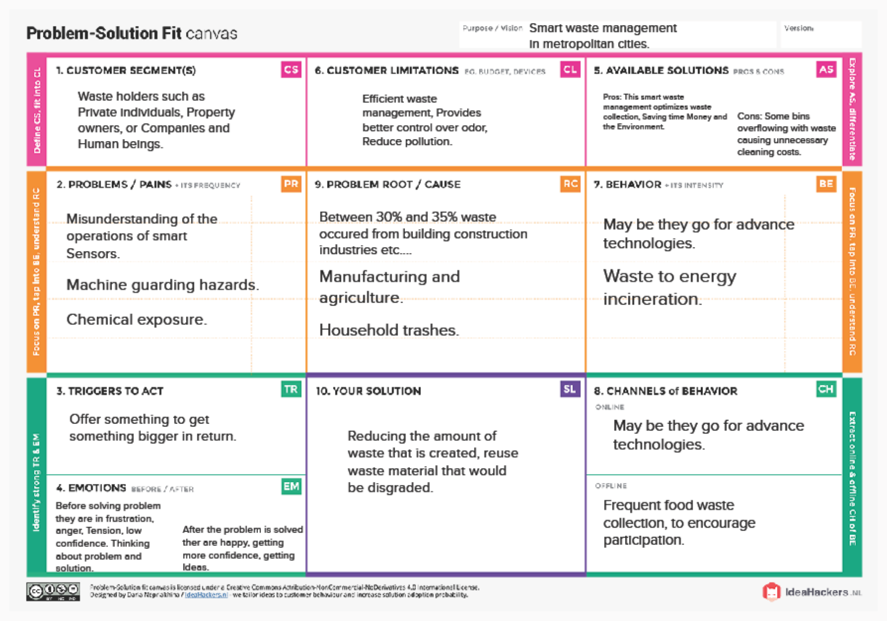
NOVELTY → smart waste bins,waste level sensors ,AI recycling robots,garbage truck weighing mechanisms,pneumatic waste pipes

SOCIAL IMPACT → Fill level is implemented to dispose solid waste.Depending on fill level system sends appropriate notification message to alert relevant authorities.

BUSINESS MODEL → No missed pickups,reduced overflows,waste generation analysis,co2 Emission Reduction

SCALABILITY OF SOLUTION → Scalability is an aspect or rather a functional quality of a system, software or solution.

3.4 Problem Solution Fit



Problem-Solution Fit canvas

Purpose / Vision: Smart waste management in metropolitan cities.

Version:

1. CUSTOMER SEGMENT(S) CS

Waste holders such as Private individuals, Property owners, or Companies and Human beings.

6. CUSTOMER LIMITATIONS EG. BUDGET, DEVICES CL

Efficient waste management, Provides better control over odor, Reduce pollution.

5. AVAILABLE SOLUTIONS PROS & CONS AS

Pros: This smart waste management optimizes waste collection, Saving time Money and the Environment.

Cons: Some bins overflowing with waste causing unnecessary cleaning costs.

2. PROBLEMS / PAINS + ITS FREQUENCY PR

Misunderstanding of the operations of smart Sensors.

Machine guarding hazards.

Chemical exposure.

9. PROBLEM ROOT / CAUSE RC

Between 30% and 35% waste occured from building construction industries etc....

Manufacturing and agriculture.

Household trashes.

7. BEHAVIOR + ITS INTENSITY BE

May be they go for advance technologies.

Waste to energy incineration.

3. TRIGGERS TO ACT TR

Offer something to get something bigger in return.

4. EMOTIONS BEFORE / AFTER EM

Before solving problem they are in frustration, anger, Tension, low confidence. Thinking about problem and solution.

After the problem is solved ther are happy, getting more confidence, getting Ideas.

10. YOUR SOLUTION SL

Reducing the amount of waste that is created, reuse waste material that would be disgraded.

8. CHANNELS of BEHAVIOR CH

ONLINE

May be they go for advance technologies.

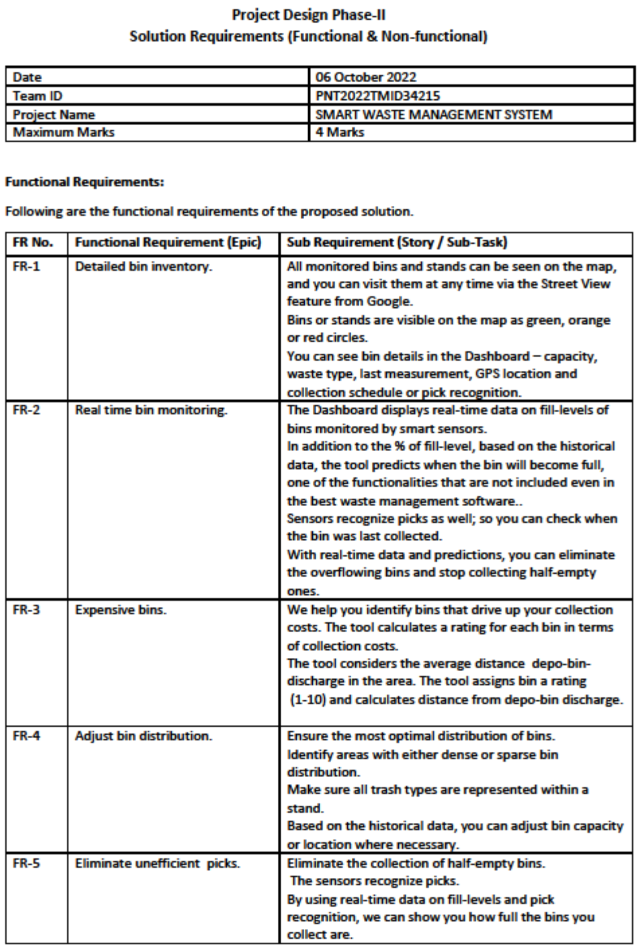
OFFLINE

Frequent food waste collection, to encourage participation.

Define CS, fit into CL
Explore AS, differentiate
Focus on PR, tap into BE, understand RC
Focus on PR, tap into BE, understand RC
Identify strong TR & EM
Extract online & offline CH of BE
IdeaHackers.NL

4.Requirement Analysis

4.1 Functional Requirements



Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Date	06 October 2022
Team ID	PNT2022TMID34215
Project Name	SMART WASTE MANAGEMENT SYSTEM
Maximum Marks	4 Marks

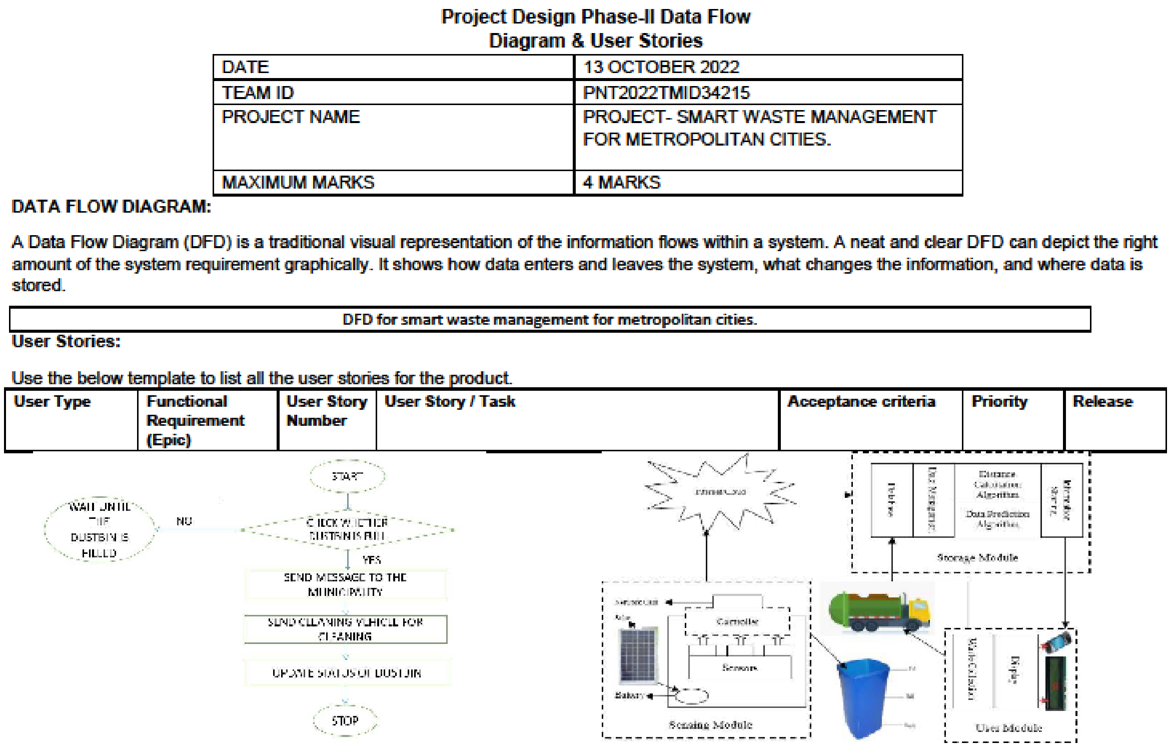
Functional Requirements:

Following are the functional requirements of the proposed solution.

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 real-time data on fill-levels of bins monitored by smart sensors. In addition to the % of fill-level, based on the historical data, the tool predicts when the bin will become full, one of the functionalities that are not included even in the best waste management software.. Sensors recognize picks as well; so you can check when the bin was last collected. With real-time data and predictions, you can eliminate the overflowing bins and stop collecting half-empty ones.
FR-3	Expensive bins.	We help you identify bins that drive up your collection costs. The tool calculates a rating for each bin in terms of collection costs. The tool considers the average distance depo-bin-discharge 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 unefficient 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.

5 Project Design

5.1 Data Flow Diagrams



Project Design Phase-II Data Flow
Diagram & User Stories

DATE	13 OCTOBER 2022
TEAM ID	PNT2022TMID34215
PROJECT NAME	PROJECT- SMART WASTE MANAGEMENT FOR METROPOLITAN CITIES.
MAXIMUM MARKS	4 MARKS

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.

DFD for smart waste management for metropolitan cities.

User Stories:

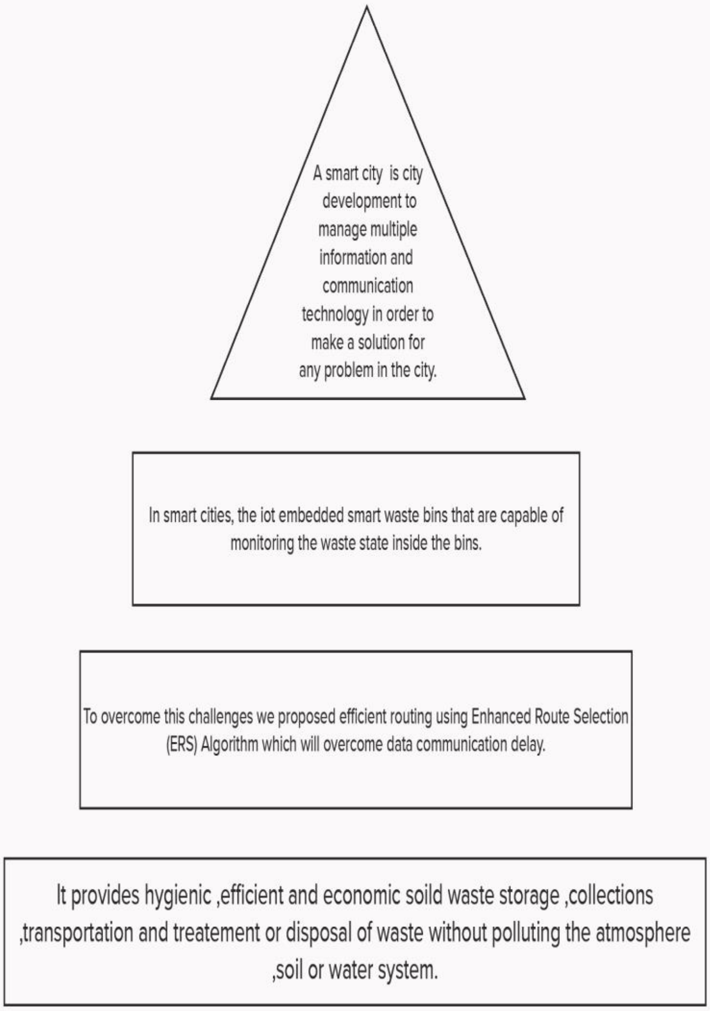
Use the below template to list all the user stories for the product.

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

START
CHECK WHETHER DUSTBIN IS FULL
NO
WAIT UNTIL THE DUSTBIN IS FILLED
YES
SEND MESSAGE TO THE MUNICIPALITY
SEND CLEANING VEHICLE FOR CLEANING
UPDATE STATUS OF DUSTBIN
STOP
Database
Data Management
Distance Calculation Algorithm
Data Prediction Algorithm
Information Sharing
Storage Module
Controller
Sensors
Battery
Sensing Module
Waste Collection
Display
User Module

5.2 Solution Architecture

SOLUTION ARCHITECHTURE



A smart city is city development to manage multiple information and communication technology in order to make a solution for any problem in the city.

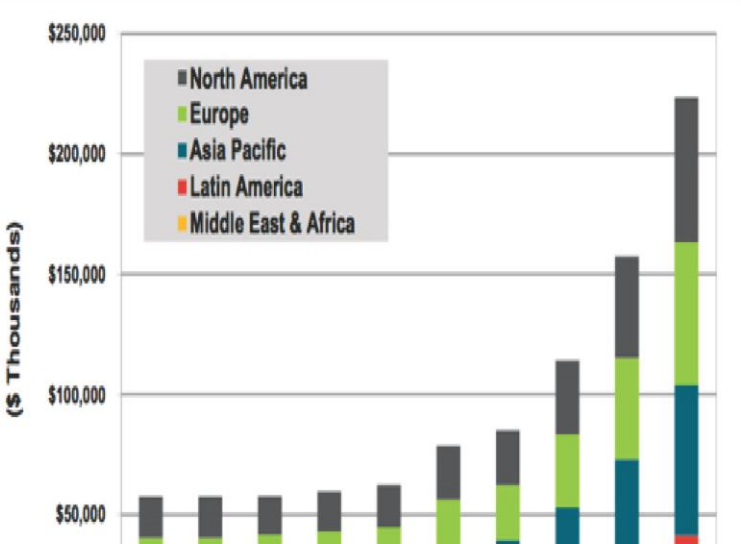
In smart cities, the iot embedded smart waste bins that are capable of monitoring the waste state inside the bins.

To overcome this challenges we proposed efficient routing using Enhanced Route Selection (ERS) Algorithm which will overcome data communication delay.

It provides hygienic ,efficient and economic soild waste storage ,collections ,transportation and treatement or disposal of waste without polluting the atmosphere ,soil or water system.

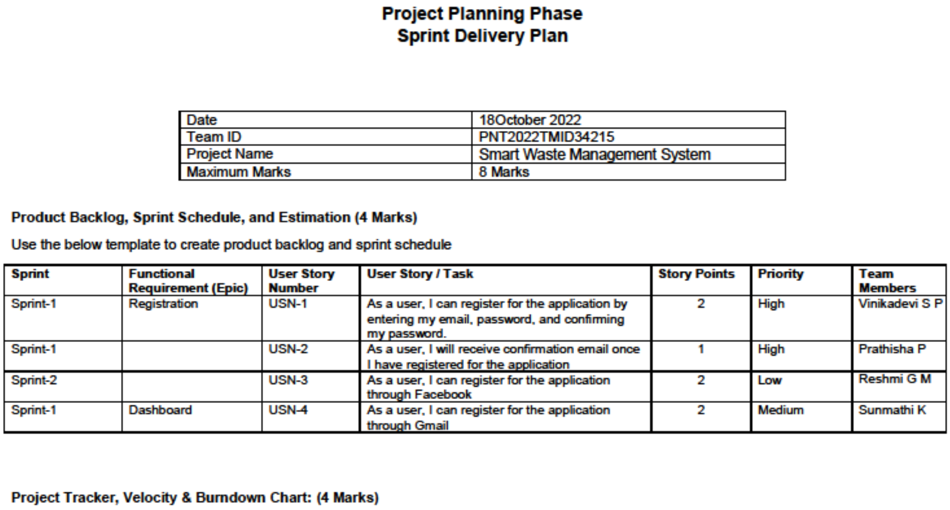
GRAPH

Chart 1 *Annual Smart Waste Collection Technology Revenue by Region, World Markets: 2016-2025*



6 Project Planning And Scheduling

6.1 Sprint Planning And Estimating



Project Planning Phase
Sprint Delivery Plan

Date	18October 2022
Team ID	PNT2022TMID34215
Project Name	Smart Waste Management System
Maximum Marks	8 Marks

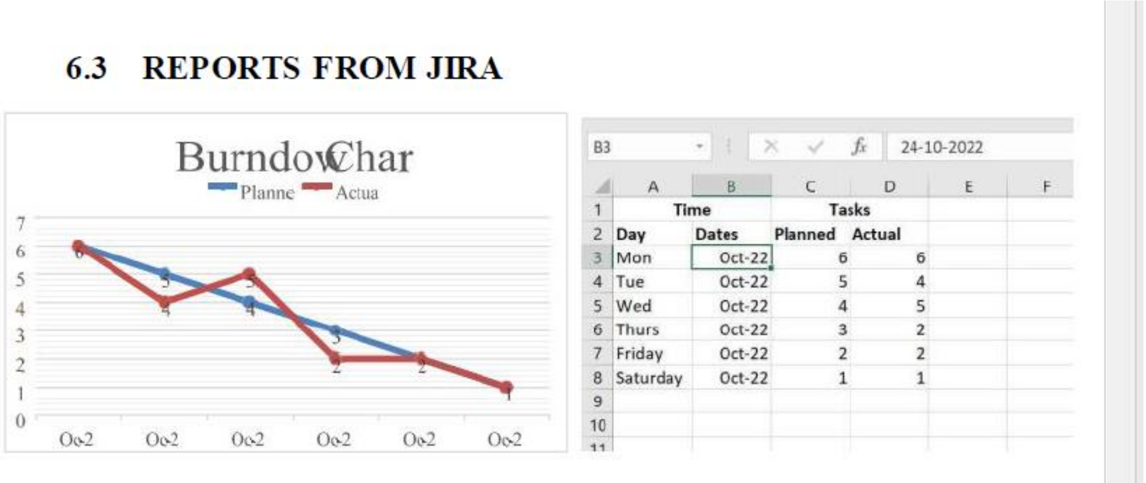
Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Vinikadevi S P
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Prathisha P
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	Reshmi G M
Sprint-1	Dashboard	USN-4	As a user, I can register for the application through Gmail	2	Medium	Sunmathi K

Project Tracker, Velocity & Burndown Chart: (4 Marks)

6.3 Reports From Jira



6.3 REPORTS FROM JIRA

	A	B	C	D	E	F
1		Time	Tasks			
2	Day	Dates	Planned	Actual		
3	Mon	Oct-22	6	6		
4	Tue	Oct-22	5	4		
5	Wed	Oct-22	4	5		
6	Thurs	Oct-22	3	2		
7	Friday	Oct-22	2	2		
8	Saturday	Oct-22	1	1		
9						
10						

7. CODING & SOLUTIONING

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

7.1 feature 1

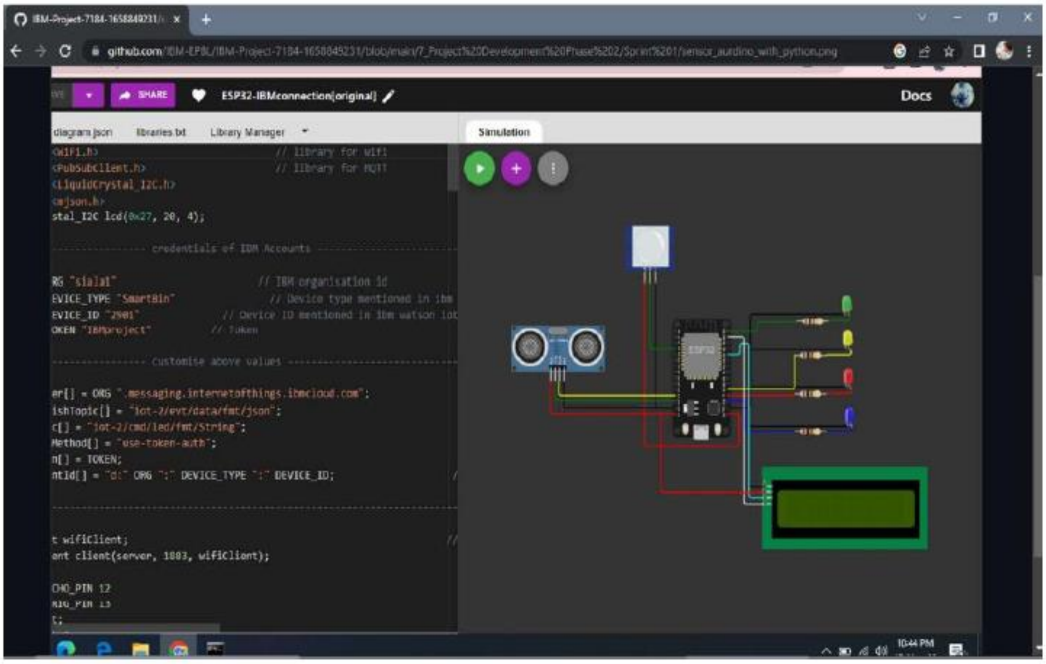
```
#include <WiFi.h> // library for wifi
```

```
#include <PubSubClient.h> // library for MQTT
#include <LiquidCrystal_I2C.h>
#include <mjson.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
//----------------------- credentials of IBM Accounts -----------------
-------------
#define ORG "siala1" // IBM organisation id
#define DEVICE_TYPE "SmartBin" // Device type
mentioned in ibm watson iot platform
#define DEVICE_ID "2901" // Device ID mentioned in
ibm watson iot platform
#define TOKEN "IBMproject" // Token
//----------------------- customise above values -----------------------
------------------------------
char server[] = ORG
".messaging.internetofthings.ibmcloud.com"; // server
name
char publishTopic[] = "iot-2/evt/data/fmt/json"; //
```

```
topic name and type of event perform and format in which data to
be send
char topic[] = "iot-2/cmd/led/fmt/String"; //
cmd Represent type and command is test format of strings
char authMethod[] = "use-token-auth"; //
authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
//Client id
//------------------------------------------------------------------------
---------------------------------------
WiFiClient wifiClient; // creating
instance for wificlient
PubSubClient client(server, 1883, wifiClient);
#define ECHO_PIN 12
#define TRIG_PIN 13
float dist;
String data3;
void setup()
{
Serial.begin(115200);
pinMode(LED_BUILTIN, OUTPUT);
pinMode(TRIG_PIN, OUTPUT);
pinMode(ECHO_PIN, INPUT);
//pir pin
pinMode(34, INPUT);
//ledpins
pinMode(23, OUTPUT);
pinMode(2, OUTPUT);
pinMode(4, OUTPUT);
pinMode(15, OUTPUT);
lcd.init();
lcd.backlight();
lcd.setCursor(1, 0);
lcd.print("");
wifiConnect();
mqttConnect();
}
float readcmCM()
{
digitalWrite(TRIG_PIN, LOW);
delayMicroseconds(2);
digitalWrite(TRIG_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW);
int duration = pulseIn(ECHO_PIN, HIGH);
return duration * 0.034 / 2;
}
void loop()
{
lcd.clear();
publishData();
delay(500);
if (!client.loop())
{
mqttConnect(); // function call to connect
to IBM
}
}
/* -----------------------------retrieving to cloud--------------------
------------------------------*/
void wifiConnect()
{
Serial.print("Connecting to ");
Serial.print("Wifi");
WiFi.begin("Wokwi-GUEST", "", 6);
while (WiFi.status() != WL_CONNECTED)
{
delay(500);
```

```
Serial.print(".");
}
Serial.print("WiFi connected, IP address: ");
Serial.println(WiFi.localIP());
}
void mqttConnect()
{
if (!client.connected())
{
Serial.print("Reconnecting MQTT client to ");
Serial.println(server);
while (!client.connect(clientId, authMethod, token))
{
Serial.print(".");
delay(500);
}
initManagedDevice();
Serial.println();
}
}
void initManagedDevice()
{
if (client.subscribe(topic))
{
Serial.println("IBM subscribe to cmd OK");
}
else
{
Serial.println("subscribe to cmd FAILED");
}
}
void publishData()
{
float cm = readcmCM();
if(digitalRead(34)) //pir motion detection
{
Serial.println("Motion Detected");
Serial.println("Lid Opened");
digitalWrite(15, HIGH);
if(digitalRead(34)== true)
{
if(cm <= 60) //Bin level detection
{
digitalWrite(2, HIGH);
Serial.println("High Alert!!!,Trash bin is about to be full");
Serial.println("Lid Closed");
lcd.print("Full! Don't use");
delay(2000);
lcd.clear();
digitalWrite(4, LOW);
digitalWrite(23, LOW);
}
else if(cm > 60 && cm < 120)
{
digitalWrite(4, HIGH);
Serial.println("Warning!!,Trash is about to cross 50% of bin
level");
digitalWrite(2, LOW);
digitalWrite(23, LOW);
}
else if(cm > 120)
{
digitalWrite(23, HIGH);
Serial.println("Bin is available");
digitalWrite(2,LOW);
digitalWrite(4, LOW);
}
delay(10000);
Serial.println("Lid Closed");
}
else
{
Serial.println("No motion detected");
digitalWrite(2, LOW);
digitalWrite(15, LOW);
digitalWrite(4, LOW);
digitalWrite(23, LOW);
}
}
else
{
digitalWrite(15, LOW);
}
if(cm <= 60)
{
digitalWrite(21,HIGH);
String payload = "{\"High_Alert\":";
payload += cm;
payload += " }";
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str())) // if
data is uploaded to cloud successfully,prints publish ok else prints
publish failed
{
Serial.println("Publish OK");
}
}
else if(cm <= 120)
```

```
{
digitalWrite(22,HIGH);
String payload = "{\"Warning\":";
payload += cm ;
payload += " }";
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload);
if(client.publish(publishTopic, (char*) payload.c_str()))
{
Serial.println("Publish OK");
}
else
{
Serial.println("Publish FAILED");
}
}
else
{
digitalWrite(23,HIGH);
String payload = "{\"Safe\":";
payload += cm;
payload += " }";
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str())) // if
data is uploaded to cloud successfully,prints publish ok else prints
publish failed
{
Serial.println("Publish OK");
}
}
float inches = (cm / 2.54); //print on lcd
lcd.setCursor(0,0);
lcd.print("Inches");
lcd.setCursor(4,0);
lcd.setCursor(12,0);
lcd.print("cm");
lcd.setCursor(1,1);
lcd.print(inches, 1);
lcd.setCursor(11,1);
lcd.print(cm, 1);
lcd.setCursor(14,1);
delay(1000);
lcd.clear();
}
//handles commands from user side
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength)
{
Serial.print("callback invoked for topic: ");
Serial.println(subscribetopic);
for (int i = 0; i < payloadLength; i++) {
data3 += (char)payload[i];
}
Serial.println("data: "+ data3);
const char *s =(char*) data3.c_str();
double pincode = 0;
const char *buf;
int len;
if (mjson_find(s, strlen(s), "$.command", &buf, &len)) //
And print it
{
String command(buf,len);
if(command=="\"Seal Bin\"")
{
Serial.println("Sealed");
}
}
data3="";
}
```

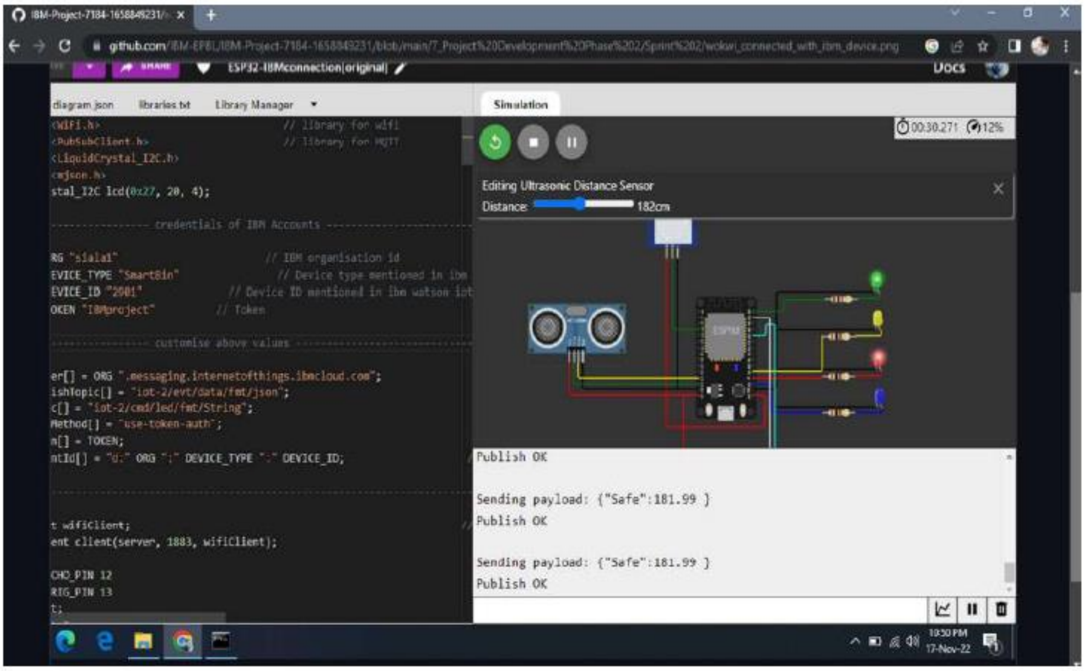


IBM-Project-7184-1658849231/
github.com/IBM-EPBL/IBM-Project-7184-1658849231/blob/main/7_Project%20Development%20Phase%202/Sprint%201/sensor_aurdino_with_python.png
SHARE
ESP32-IBMconnection(original)
Docs
diagram.json
libraries.txt
Library Manager
Simulation
ESP32
10:44 PM

```
<WiFi.h>                                 // library for wifi
<PubSubClient.h>                         // library for MQTT
<LiquidCrystal_I2C.h>
<mjson.h>
stal_I2C lcd(0x27, 20, 4);

---------------- credentials of IBM Accounts ----------------

RG "sialat"                      // IBM organisation id
EVICE_TYPE "SmartBin"               // Device type mentioned in ibm
EVICE_ID "2901"              // Device ID mentioned in ibm watson iot
OKEN "IBMproject"          // Token

---------------- Customise above values ----------------

er[] = ORG ".messaging.internetofthings.ibmcloud.com";
ishTopic[] = "iot-2/evt/data/fmt/json";
c[] = "iot-2/cmd/led/fmt/String";
Method[] = "use-token-auth";
n[] = TOKEN;
ntId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

--------------------------------------------------------

t wifiClient;
ent client(server, 1883, wifiClient);

CHO_PIN 12
RIG_PIN 13
t;
```

Sensor-ardunio connections

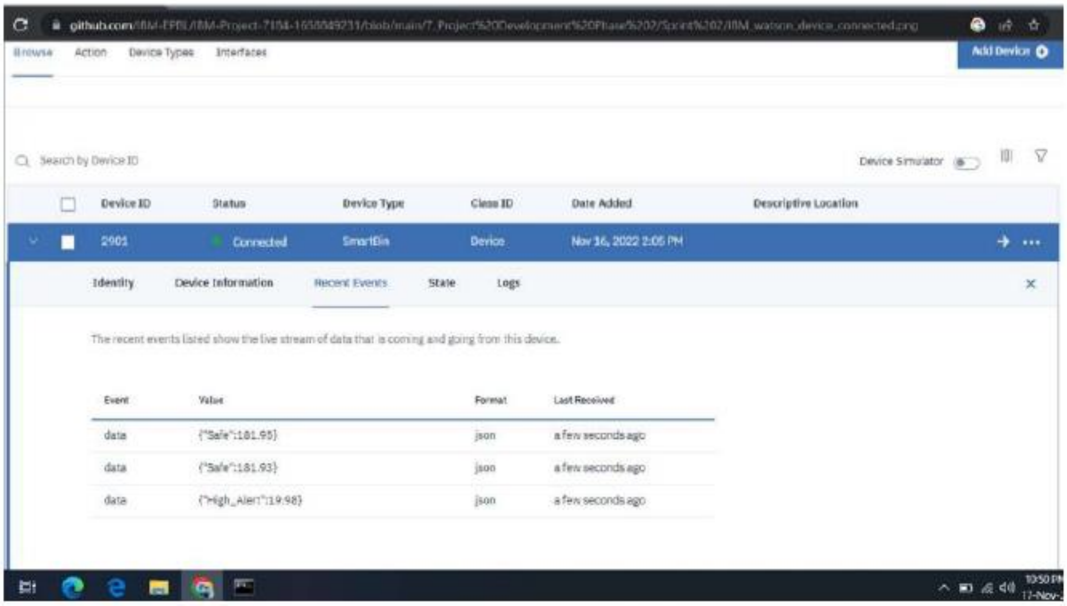
7.2 Feature 2



IBM-Project-7184-1658849231/
github.com/IBM-EPBL/IBM-Project-7184-1658849231/blob/main/7_Project%20Development%20Phase%202/Sprint%202/wokwi_connected_with_ibm_device.png
SHARE
ESP32-IBMconnection(original)
Docs
diagram.json
libraries.txt
Library Manager
Simulation
00:30.271
12%
Editing Ultrasonic Distance Sensor
Distance
182cm
ESP32
10:50 PM
17-Nov-22

```
<WiFi.h>                                   // library for wifi
<PubSubClient.h>                           // library for MQTT
<LiquidCrystal_I2C.h>
<mjson.h>
stal_I2C lcd(0x27, 20, 4);

---------------- credentials of IBM Accounts ----------------------

RG "sialal"                        // IBM organisation id
EVICE_TYPE "SmartBin"               // Device type mentioned in ibm
EVICE_ID "2001"                 // Device ID mentioned in ibm watson iot
OKEN "IBMproject"           // Token

---------------- customise above values ----------------------------

er[] = ORG ".messaging.internetofthings.ibmcloud.com";
ishTopic[] = "iot-2/evt/data/fmt/json";
c[] = "iot-2/cmd/led/fmt/String";
Method[] = "use-token-auth";
n[] = TOKEN;
ntId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

----------------------------------------------------------------------


t wifiClient;
ent client(server, 1883, wifiClient);

CHO_PIN 12
RIG_PIN 13
t;
```

```
Publish OK

Sending payload: {"Safe":181.99 }
Publish OK

Sending payload: {"Safe":181.99 }
Publish OK
```

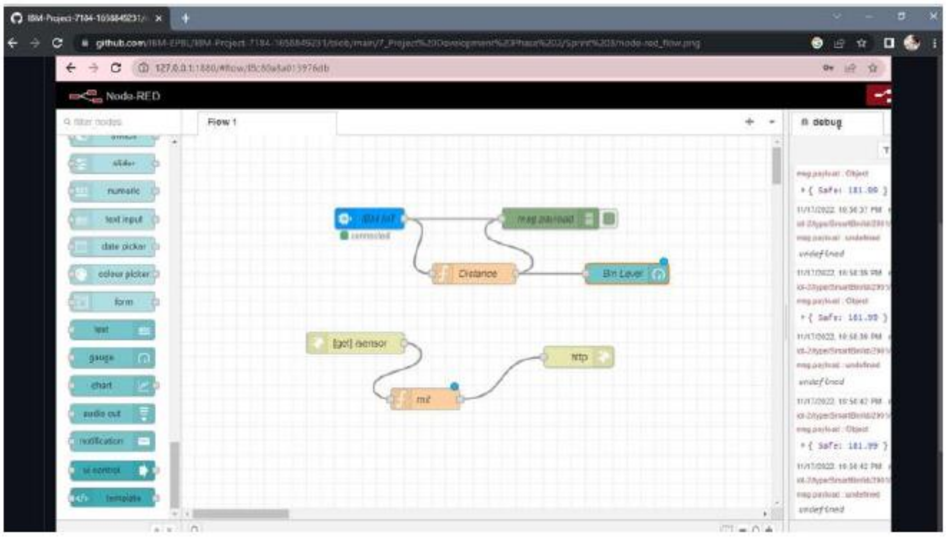


	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
■	2901	Connected	SmartBin	Device	Nov 16, 2022 2:05 PM	

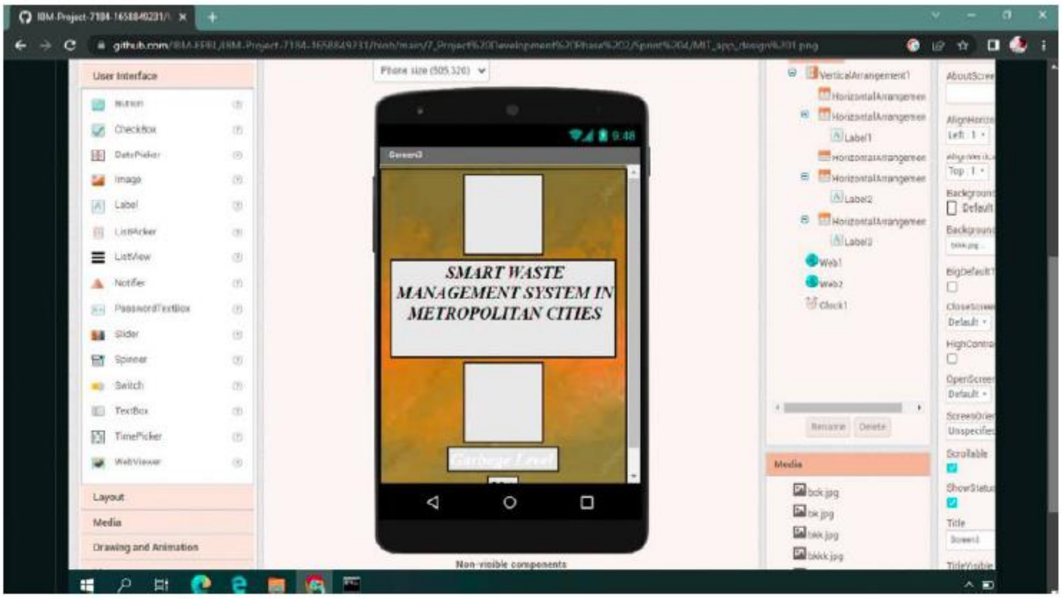
The recent events listed show the live stream of data that is coming and going from this device.

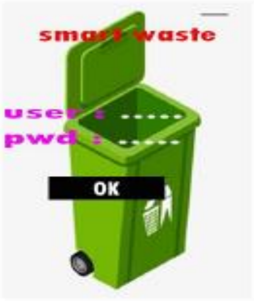
Event	Value	Format	Last Received
data	{"Safe":181.95}	json	a few seconds ago
data	{"Safe":181.93}	json	a few seconds ago
data	{"High_Alert":19.98}	json	a few seconds ago

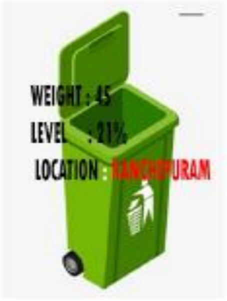
IBM Watson Device connected



Nod red connections

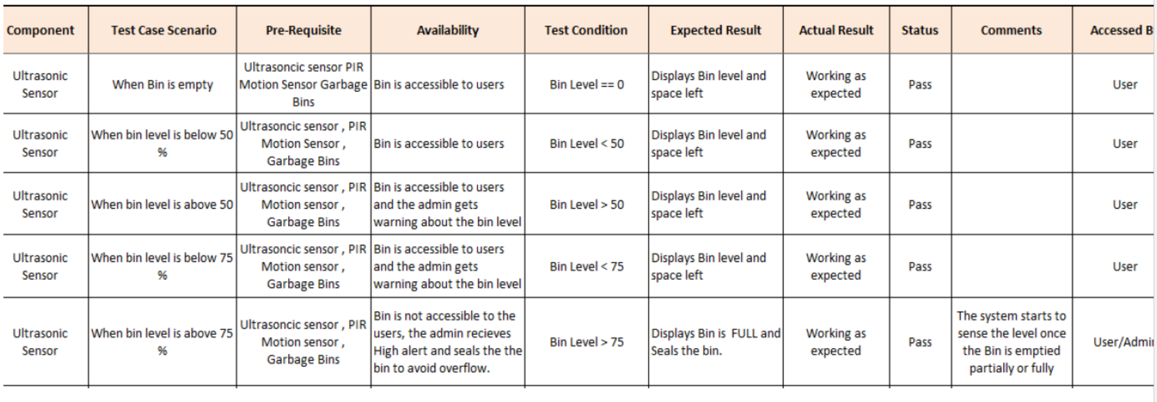





WEIGHT : 45
LEVEL : 21%
LOCATION : KANCHIPURAM

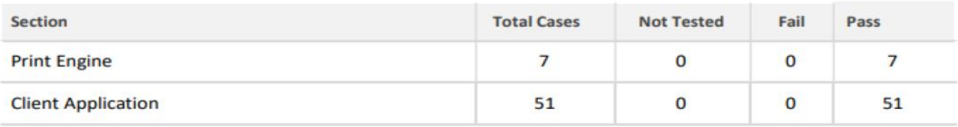
8. Testing

8.1 Testing Cases

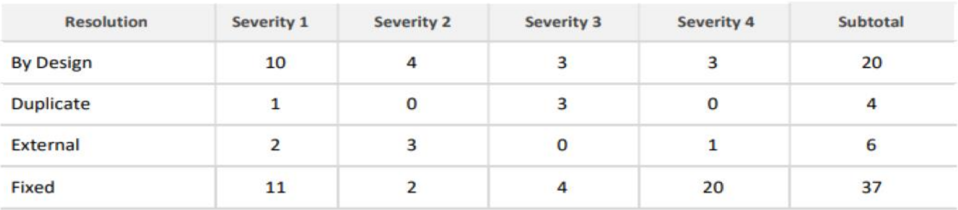


Component	Test Case Scenario	Pre-Requisite	Availability	Test Condition	Expected Result	Actual Result	Status	Comments	Accessed B
Ultrasonic Sensor	When Bin is empty	Ultrasoncic sensor PIR Motion Sensor Garbage Bins	Bin is accessible to users	Bin Level == 0	Displays Bin level and space left	Working as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

8.2 User Acceptance Testing



Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51



Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	3	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37

9. RESULTS

9.1 Performance results

The smart waste management solution to make the waste collection process more efficient is sensors. Sensors can measure the fill level of the containers and provide updated information at any time and notify waste management services to empty them when they are full or almost full. These devices help optimize the best possible route containing fully filled containers and create smart schedules for drivers.

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES :

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

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

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

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

DISADVANTAGES:

➨System requires more number of waste bins for separate waste collection as per population in the city. This results into high initial cost due to expensive smart dustbins compare to other methods.

➨Sensor nodes used in the dustbins have limited memory size.

➨Wireless technologies used in the system such as zigbee and Wi-Fi have shorter range and lower data speed. In RFID based systems, RFID tags are affected by surrounding metal objects (if any).

➨It reduces man power requirements which results into increase in unemployment for unskilled people.

➨The training has to be provided to the people involved in the smart waste management system.

11. CONCLUSION

The behavior of generating garbage is too dangerous not only for today's generation, but also for future generations. It is critical to educate people and encourage them to practice recycle,reuse and reduse instead of producing waste. Waste disposal should be a priority for municipalities and governments.

12. FUTURE SCOPE

In this report, smart bin is built on a microcontroller based platform ArdunioUno board, which is interfaced with ultrasonic sensor. It will stop overflowing of dustbins along roadsides and localities as smart Dustbins are managed.

13. APPENDIX

SOURCE CODE:

```
#include <WiFi.h> // library for wifi
#include <PubSubClient.h> // library for MQTT
#include <LiquidCrystal_I2C.h>
#include <mjson.h>
```

```
LiquidCrystal_I2C lcd(0x27, 20, 4);
//----------------------- credentials of IBM Accounts --------------
----------------
#define ORG "siala1" // IBM organisation id
#define DEVICE_TYPE "SmartBin" // Device type
mentioned in ibm watson iot platform
#define DEVICE_ID "2901" // Device ID mentioned in
ibm watson iot platform
#define TOKEN "IBMproject" // Token
//----------------------- customise above values ---------------------
--------------------------------
char server[] = ORG
".messaging.internetofthings.ibmcloud.com"; // server
```

```
name
char publishTopic[] = "iot-2/evt/data/fmt/json";
// topic name and type of event perform and format in which
data to be send
char topic[] = "iot-2/cmd/led/fmt/String"; //
cmd Represent type and command is test format of strings
char authMethod[] = "use-token-auth"; //
authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":"
DEVICE_ID; //Client id
//----------------------------------------------------------------------
------------------------------------------
WiFiClient wifiClient; //
creating instance for wificlient
PubSubClient client(server, 1883, wifiClient);
#define ECHO_PIN 12
#define TRIG_PIN 13
float dist;
String data3;
void setup()
{
Serial.begin(115200);
pinMode(LED_BUILTIN, OUTPUT);
pinMode(TRIG_PIN, OUTPUT);
pinMode(ECHO_PIN, INPUT);
//pir pin
pinMode(34, INPUT);
//ledpins
pinMode(23, OUTPUT);
pinMode(2, OUTPUT);
pinMode(4, OUTPUT);
pinMode(15, OUTPUT);
lcd.init();
lcd.backlight();
lcd.setCursor(1, 0);
lcd.print("");
wifiConnect();
mqttConnect();
}
float readcmCM()
{
digitalWrite(TRIG_PIN, LOW);
delayMicroseconds(2);
digitalWrite(TRIG_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW);
int duration = pulseIn(ECHO_PIN, HIGH);
return duration * 0.034 / 2;
```

```
}
void loop()
{
lcd.clear();
publishData();
delay(500);
if (!client.loop())
{
mqttConnect(); // function call to
connect to IBM
}
}
/* --------------------------------retrieving to cloud-----------------
------------------------------------*/
void wifiConnect()
{
Serial.print("Connecting to ");
Serial.print("Wifi");
WiFi.begin("Wokwi-GUEST", "", 6);
while (WiFi.status() != WL_CONNECTED)
{
delay(500);
Serial.print(".");
}
Serial.print("WiFi connected, IP address: ");
Serial.println(WiFi.localIP());
}
void mqttConnect()
{
if (!client.connected())
{
Serial.print("Reconnecting MQTT client to ");
Serial.println(server);
while (!client.connect(clientId, authMethod, token))
{
Serial.print(".");
delay(500);
}
initManagedDevice();
Serial.println();
}
}
void initManagedDevice()
{
if (client.subscribe(topic))
{
Serial.println("IBM subscribe to cmd OK");
}
else
{
```

```
Serial.println("subscribe to cmd FAILED");
}
}
void publishData()
{
float cm = readcmCM();
if(digitalRead(34)) //pir motion detection
{
Serial.println("Motion Detected");
Serial.println("Lid Opened");
digitalWrite(15, HIGH);
if(digitalRead(34)== true)
{
if(cm <= 60) //Bin level detection
{
digitalWrite(2, HIGH);
Serial.println("High Alert!!!,Trash bin is about to be full");
Serial.println("Lid Closed");
lcd.print("Full! Don't use");
delay(2000);
lcd.clear();
digitalWrite(4, LOW);
digitalWrite(23, LOW);
}
else if(cm > 60 && cm < 120)
{
digitalWrite(4, HIGH);
Serial.println("Warning!!,Trash is about to cross 50% of bin level");
digitalWrite(2, LOW);
digitalWrite(23, LOW);
}
else if(cm > 120)
{
digitalWrite(23, HIGH);
Serial.println("Bin is available");
digitalWrite(2,LOW);
digitalWrite(4, LOW);
}
delay(10000);
Serial.println("Lid Closed");
}
else
{
Serial.println("No motion detected");
```

```
digitalWrite(2, LOW);
digitalWrite(15, LOW);
digitalWrite(4, LOW);
digitalWrite(23, LOW);
}
}
else
{
digitalWrite(15, LOW);
}
if(cm <= 60)
{
digitalWrite(21,HIGH);
String payload = "{\"High_Alert\":";
payload += cm;
payload += " }";
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str())) //
if data is uploaded to cloud successfully,prints publish ok else
prints publish failed
{
```

```
Serial.println("Publish OK");
}
}
else if(cm <= 120)
{
digitalWrite(22,HIGH);
String payload = "{\"Warning\":";
payload += cm ;
payload += " }";
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload);
if(client.publish(publishTopic, (char*) payload.c_str()))
{
Serial.println("Publish OK");
}
else
{
Serial.println("Publish FAILED");
}
}
else
{
digitalWrite(23,HIGH);
String payload = "{\"Safe\":";
```

```
payload += cm;
payload += " }";
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str())) //
if data is uploaded to cloud successfully,prints publish ok else
prints publish failed
{
Serial.println("Publish OK");
}
}
float inches = (cm / 2.54); //print on
lcd
lcd.setCursor(0,0);
lcd.print("Inches");
lcd.setCursor(4,0);
lcd.setCursor(12,0);
lcd.print("cm");
lcd.setCursor(1,1);
lcd.print(inches, 1);
lcd.setCursor(11,1);
lcd.print(cm, 1);
lcd.setCursor(14,1);
delay(1000);
```

```
lcd.clear();
}
//handles commands from user side
void callback(char* subscribetopic, byte* payload, unsigned
int payloadLength)
{
Serial.print("callback invoked for topic: ");
Serial.println(subscribetopic);
for (int i = 0; i < payloadLength; i++) {
data3 += (char)payload[i];
}
Serial.println("data: "+ data3);
const char *s =(char*) data3.c_str();
double pincode = 0;
const char *buf;
int len;
if (mjson_find(s, strlen(s), "$.command", &buf, &len)) //
And print it
{
```

```
String command(buf,len);
if(command=="\"Seal Bin\"")
{
Serial.println("Sealed");
}
}
data3="";
}
```

GITHUB LINK- https://github.com/IBM-EPBL/IBM-Project-40308-1667561919