Department of Computer Science

University of Gujrat

WASTE CONTROL USING SMART BIN

Submitted By:

Muhammad Atif Bashir	**16201519-009**
Daud Rauf	**16201519-023**
Hamza Khalid	**16201519-108**

Supervised By:

Dr. Abdul Rehman

Chapter 1: Project Feasibility Report

Table of Contents

1.1. Introduction

Main reason behind excessive pollution is absence of a waste management system to manage garbage. People dump garbage in open, empty spaces available in the vicinity. It creates unhygienic condition for the people and creates bad smell around the surroundings, which leads spreading diseases to the people living in such environment. A waste management system works by placing bins in different areas and collecting waste from them after regular intervals. In some areas, bins may overflow due to unavailability or frequency of garbage collection vehicles.

To avoid such situations, we are going to implement a project called IOT Based waste management using smart dustbin. In this system, multiple dustbins are located throughout the city or the Campus. These dustbins are provided with a sensor which detects the level of the waste and a GPS device for getting the location and to identify which bin is full. Data is transmitted by using 2G/3G network. When the level of the bin exceeds, the device will transmit the reading along with the unique ID provided. It will inform the status of each bin in real time so that concerned authority can send the garbage collection vehicle only when the bin is full. By implementing this system resource optimization, cost reduction, effective usage of smart bins can be done.

1.2. Project/Product Feasibility Report

1.2.1. Technical Feasibility

In Technical point of view tools that are used to develop our project are available, so it is possible to develop our application and our team members have skills to develop such system. It is measure of the practicality of specific technical solution and the availability of technical resources and expertise. The development of 'Waste Control Using Smart Bin' is technical feasible because of the following reason:

Hardware Components are easily available.

Technicality of our project is to detect Waste Level, for which Ultrasonic Sensors are available.

Communication of bin and system can easily be done with GPS and Wi-Fi Sensors.

To check whether the module technically feasible or not we have to give the following two question answer.

Q-1 Is the proposed system practical?

Ans. The proposed system is practical as we have all the resources available. Also building up of this application requires Node-ESP32 or Arduino, Ultrasonic Sensors, PS and Wi-Fi as a hardware and software modules are easily available. So, the proposed system is extremely efficient and practical.

Q-2 Do we currently possess the necessary technology?

Ans. Looking into the system requirement, we can see that we possess all the hardware and software requirements. Also, the technology and the hardware used is easily available and deployed all around the world.

1.2.2. Operational Feasibility

In operational feasibility we will discuss the skill set of our team if we have the capability of doing this project or not?

The question arises in operational feasibility is that is problem is worth solving or not? The project is very helpful especially in countries like Pakistan where there is no such system available which can efficiently solve the Waste System. As mostly waste is managed by

T.M.A of tehsil or in big city managed by private company like OZ-PAK. Which failed in providing Waste free environment. So, it is required that to solve these issues as everybody wants to live in clean and green environment. Our proposed system will provide users with the facility of Online Complaints and intimation of bins.

Moreover, if we talk about the skill set of our team members required for such project then our team has such level of domain knowledge to develop this project.

1.2.3. Economic Feasibility

Economically our Project has many benefits. Economic Feasibility has two types:

Cost Estimates

In Cost Estimates Total Cost of a Project can be calculated

Acquisition Cost = 85k with one bin

Maintenance and Operation Cost = 10k Per Year of Application Services and 10k for every year testing and updating.

Benefit Estimates

Benefit estimates enclose tangible benefits and intangible benefits.

Tangible Benefits

First year we provide our services for free and make our system easier to use then our services are paid. But the price of resources like smart bins is not free.

Intangible Benefits

We try to provide best services to the users of our system and provide correct and useful information and help.

1.2.4. Schedule Feasibility

Scheduling and Planning of Project is very important to complete Project on Time. In this regard we divide the project into small incremental with proper dependencies and durations and applying the agile development methods to complete it in the given timeframe.

1.2.5. Specification Feasibility

The requirements specification and analyses is the core part of project building we must satisfy the requirements and constraints to make our project feasible for the customer according to its requirements for this purpose we have managed some interviews with some of the people to know the requirements of customer for the proposed system so, by doing this we are able to say the proposed system is feasible with the aspects of customer requirements.

1.2.6. Information Feasibility

To complete our project, we divide into the modules and assigned each module to each member according to its skill set so that everyone would manage its module properly to make this project done in the time. We would use advance technologies to make our application more reliable and useful for the end user.

.

1.2.7. Motivational Feasibility

In Pakistan there is no trend of controlling the waste in efficient way and using the old methods still to manage the waste. As technology is overcoming everything day by day and the world moves towards the smart system to manage the waste that’s the main motivation behind our project that it would provide solution to issues in Pakistan and would revolutionize the waste management system in a smarter way. That's why we decided this to develop such application and system which provides a smart way to control the waste.

1.2.8. Legal & Ethical Feasibility

Legally and ethically we have all legal rights to use the tools which are used to develop our project. As we are using cross platform technologies to develop our desktop application, so these technologies and sensors that is used are available to everybody for use that's why we are legally and ethically able to use these technologies.

1.3. Project/Product Scope

Today there is trend of doing things in smarter way rather than putting extra effort to do the work. In Pakistan, still waste is managed by T.M.A and still overload problem is not resolved. There's need of proper awareness and proper system to overcome the problem. So, we come up with the solution of this issue with a smarter way in the form of web application and mobile application. Web application will help to manage the driver's data and reports whereas mobile application help drivers to check the level of waste in bins. It makes the entire collection system smart and controllable by eliminating the overflow problems by reducing time on waste collection, vehicle numbers and fuel consumption. We can reduce the cost and protect the environment using such techniques.

Moreover, there is also a trend of suggestions and prediction in software-based application so our proposed system has a module that if bins are filled with waste but not respond by sensor then an authenticated person can intimate through application.

1.4. Project/Product Costing

1.4.1. Project Cost Estimation by Function Point Analysis

Following are five information domain characteristics are determined and counts are provided in here as:

	Simple	**Average**	**Complex**
Number of User Inputs			2
Number of User Outputs			9
Number of User Inquiries			6
Number of Files			3
Number of External Interfaces			3

Table 1: Cost Estimation

Complexity Adjustment Factors

Data communications	3
Distributed data processing	2
Performance	4
Heavily used configuration	3
Transaction rate	0

On-Line data entry	4
End-user efficiency	3
On-Line update	5
Complex processing	5
Reusability	4
Installation ease	3
Operational ease	3
Multiple sites	2
Facilitate change	4

Table 2: Complexity Adjustment Factor

FP EST = 204.6
Cost / FP = 20,000 / 4.55
Cost / FP = 4395 Rs/FP
Total Project Cost = FP est. * (cost / FP)
= 204.6 * (4395)
= 899340 Rs
Total Estimated Effort = FP est. / productivity parameter
= 204.6 / 4.55
= 44.96 pm

1.4.2. Activity Based Costing

	Activity	Resources	Cost Rate	Duration
A	Feasibility	MS Word	1k	2 Weeks
B	Requirement Specification	MS Word	3k	4 Weeks
C	Sensor Integration	Angular, Android Studio	4k	2 Weeks
D	DBMS	SQL Server, Android Studio, Angular	6k	5 Weeks
E	Interface Designing	Angular, Android Studio	5k	4 Weeks
F	Hardware Component Configuration	Angular, Android Studio	7k	8 Weeks
G	Sensor Control and connectivity to DBMS along with	Android Studio, SQL Server	5k	2 Week

	integration with Application			
H	Integration of all Modules	Angular, Android Studio	5k	4 Weeks
I	Unit Compilation	Android Studio	3k	3 Weeks
J	Testing	Android Studio, Visual Code	5k	4 Weeks

Table 3: Activity Based Costing

1.5. Task Dependency Table

	TASK	**DEPENDENCY**	**Duration (Weeks)**
A	Feasibility	None	2 Weeks
B	Requirement Specification	A	2 Weeks
C	Sensor Integration	A, B	1 Weeks
D	DBMS	A, B, C	2 Weeks
E	Interface Designing	A, B, C, D	2 Weeks
F	Hardware Component Configuration	A, B, C, D, E	2 Weeks
G	Sensor Control and connectivity to DBMS along with integration with Application	A, B, C, D, F	4 Week
H	Integration of all Modules	A, B, C, D, E, F, G	3 Weeks
I	Unit Compilation	A, B, C, D, E, F, G, H	3 Weeks
J	Testing	A, B, C, D, E, F, G, I	4 Weeks

Table 4 : Task Dependency Table

1.6. CPM - Critical Path Method

1. Specify the Individual Activities

a. Feasibility
b. Requirement Specification
c. Sensor Integration
d. DBMS
e. Interface Design
f. Hardware component configuration
g. Sensor Control and connectivity to DBMS along with integration with Application
h. Integration of all modules
i. Unit Compilation
j. Testing

2. Determine the Sequence of the Activities

	TASK	DEPENDENCY	Duration (Weeks)
A	Feasibility	None	2 Weeks
B	Software Requirement Specification	A	2 Weeks
C	Sensor Integration	A, B	1 Weeks
D	DBMS	A, B, C	2 Weeks
E	Interface Designing	A, B, C, D	2 Weeks
F	Hardware Component Configuration	A, B, C, D, E	2 Weeks
G	Sensor Control and connectivity to DBMS along with integration with Application	A, B, C, D, F	4 Week
H	Integration of all Modules	A, B, C, D, E, F, G	3 Weeks
I	Unit Compilation	A, B, C, D, E, F, G, H	3 Weeks
J	Testing	A, B, C, D, E, F, G, I	4 Weeks

Table 5 : Sequence of Activities

3. Draw the Network Diagram

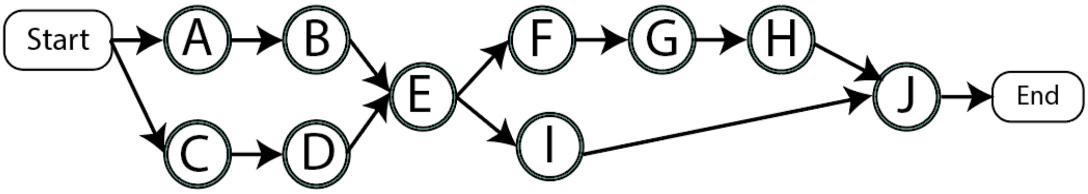


Figure 1: Network Diagram

4. Estimate Activity Completion Time

Activity	Duration	ES	EF	LS	LF	TS	FS
A	2	0	5	0	5	0	0
B	2	3	3	3	6	3	2
C	1	5	13	5	13	0	0
D	2	5	12	6	13	1	1
E	2	0	7	6	13	6	6
F	2	13	17	13	17	0	0
G	4	17	22	17	13	0	0

H	3	21	34	12	14	0	0
I	3	13	17	13	17	0	0
J	4	17	22	17	13	0	0

Table 6 : Activity Completion Time

5. Identify the Critical Path

B, D, F, E, H

1.7. Gantt chart

Task	Start	End	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr
Proposal & idea	03/09/19	23/09/19	🟧							
SRS	26/09/19	25/11/19		🟦	🟦					
Design	26/11/19	24/12/19				🟨				
Coding	25/12/19	24/02/20					🟪	🟪		
Testing and Implement	25/02/20	24/03/20							🟦	
Maintenance	25/03/20	24/04/20								⬛

Table 7 : Gantt chart

1.8. Introduction to Team member and their skill set

Members Name	**Skill Set**
Muhammad Atif Bashir	Sensor Configuration, Development
Hamza Khalid	Interface Design, Gestures handling,
Daud Rauf	Algorithms Designing, Testing

Table 8 : Team Members

1.9. Task and Member Assignment Table

Task durations and dependencies

Task	**Duration (weeks)**	**Dependencies**	**Members**
T1	2		M1, M2, M3
T2	4	T1	M1, M2
T3	2	T1, T2	M2, M3
T4	5	T1, T2, T3	M1, M3
T5	4	T1, T2, T3, T4	M1, M2, M3
T6	8	T1, T2 ,T3, T4	M2, M3
T7	2	T1, T2 ,T3, T4, T6	M1, M2
T8	4	T1, T2 ,T3, T4, T5, T6, T7	M1, M2, M3

T9	3	T1, T2 ,T3, T4, T5, T6, T7,T8	M1, M2, M3
T10	4	T1, T2 ,T3, T4, T5, T6, T7,T8,T9	M1, M2, M3

Table 9 : Task Member & Assignment Table

Allocation of People to Activities:

Task	**Member**
T1	**M Atif Bashir, Hamza Khalid, Daud Rauf**
T2	**M Atif Bashir, Hamza Khalid**
T3	**Daud Rauf, Hamza Khalid**
T4	**M Atif Bashir , Daud Rauf**
T5	**Daud Rauf, Hamza Khalid, M Atif Bashir**
T6	**Hamza Khalid, Daud Rauf**
T7	**Hamza Khalid, M Atif Bashir**
T8	**Hamza Khalid, M Atif Bashir, Daud Rauf**
T9	**Hamza Khalid, M Atif Bashir, Daud Rauf**
T10	**Hamza Khalid, M Atif Bashir, Daud Rauf**

Table 10 : Allocation of Activities

1.10. Tools and Technology with reasoning

Android Studio
For Front-End designing of an Application.

Arduino/Node ESP-32
For Circuitry on-board including the power supply jack and input-output pins and everything in between. It is simply a ready to use micro-controller-based board.

Visio
For UML Diagrams and Documentation

Ultrasonic Sensor
For checking waste level in bins.

GPS Sensor
For location.

Wi-Fi
Transmit data.

Angular
For Front-End designing of an Application.

1.11. Vision Document

The Vision behind the development of this project is to control waste in efficient and smarter way. As waste cause several diseases if not dumped and collect in proper way. So our vision is to reduce the waste problem especially in Pakistan where waste is the main problem. As we can see the situation of Karachi, Pakistan where waste is dumped at roads and different residential area. Our project plant smart bins in that polluted are which will then monitor. Moreover, the constraints and requirements of all the stake holders of this project has collected, discussed and managed because the discussion with the stake holders about this has done. The main requirements of stake holders involve in this project are mentioned below in the form of a list.

a. The application should be easily controllable with the gestures.
b. It should provide a platform where users dump the waste and waste level intimate to system by bins.
c. All things regarding waste (level, overload etc.) should be controlled by the sensor to intimate.
d. It should provide flexibility to intimate by user that the bins are overload.
e. It should provide shortest path to driver to collect the garbage.
f. It should provide a friendly user interface to both owner of the system and user using it.
g. It should be reliable and fast.
h. It should enhance the user experience.

So, these are all the major requirements of the stakeholders of the user, and an interview is also conducted with the stake holders to discuss the constraints and requirement of this project and everything about this is well planned.

1.12. Risk List

There would be several risks in the proposed system as described it a web-based application with smart bins so the major risks of this mentioned below in the following list.

Can’t be afford by the local market

As it’s a web and mobile application and require the support for sensors which has a good cost so it’s not feasible for the local to afford it with respect to cost and it also require a proper setup at a specific place where no one interfere the sensors except authentic person therefore it can’t be afford by the local market.

Can’t setup for personal use

As it’s mentioned above it required an extra setup place and smart bins which is also impossible to afford for the personal use and it also seems to be useless for personal use because no one buy smart bin for home.

Natural disaster like storms etc.

Sensors may be damaged when heavy storms occurred so it will affect our system as sensor proofing cost is too much.

1.13. Product Features/ Product Decomposition

The main key features of the project are listed below.

a. Employee management.
b. Vehicle management
c. Bin Tracking
d. Waste Level Monitoring

e. Intimation to Driver
f. Manual bin level intimation

Second Deliverable for Object Oriented Approach

Chapter 2: System Requirement Specification

2.1 Introduction:

Main reason behind excessive pollution is absence of a waste management system to manage garbage. People dump garbage in open, empty spaces available in the vicinity. It creates unhygienic condition for the people and creates bad smell around the surroundings, which leads spreading diseases to the people living in such environment. A waste management system works by placing bins in different areas and collecting waste from them after regular intervals. In some areas, bins may overflow due to unavailability or frequency of garbage collection vehicles.

To avoid such situations, we are going to implement a project called IOT Based waste management using smart dustbin. In this system, multiple dustbins are located throughout the city or the Campus. These dustbins are provided with a sensor which detects the level of the waste and a GPS device for getting the location and to identify which bin is full. Data is transmitted by using 2G/3G network. When the level of the bin exceeds, the device will transmit the reading along with the unique ID provided. It will inform the status of each bin in real time so that concerned authority can send the garbage collection vehicle only when the bin is full. By implementing this system resource optimization, cost reduction, effective usage of smart bins can be done.

2.1.1 Systems Specifications

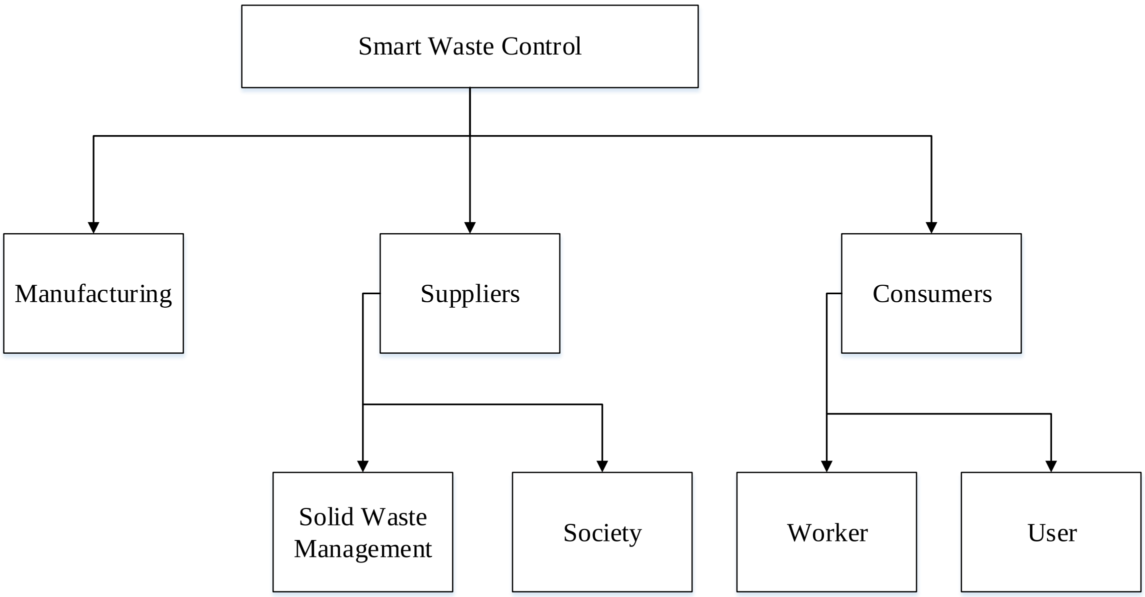
Introduction

Smart waste management is the idea from which we can smartly mange the waste and control lots of problems which disturbs the society in pollution and diseases. The waste management has to be done instantly else it leads to irregular management which will have adverse effect on nature. The Smart waste management is compatible mainly with concept of smart cities.

Existing System

The current system is not such efficient because of less management, the garbage is not collected properly. Also, there is no check and balance that the garbage is collected by the concerned person or not. People dump garbage in open, empty spaces available in the vicinity. It creates unhygienic condition for the people and creates bad smell around the surroundings, which leads spreading diseases to the people living in such environment. A waste management system keeps an eye on all parts whether it is about to collect the garbage, to track the drivers, to analyze the data, to manage the funds etc. It can also be job provider through which we can hire different workers, drivers, and IT professionals to manage efficiently. By implementing this system resource optimization, cost reduction, effective usage of smart bins can be done.

Organizational Chart



Scope of the System

Today there is trend of doing things in smarter way rather than putting extra effort to do the work. In Pakistan, still waste is managed by T.M.A and still overload problem is not resolved. There's need of proper awareness and proper system to overcome the problem. So, we come up with the solution of this issue with a smarter way in the form of web application and mobile application. Web application will help to manage the driver's data and reports whereas mobile application help drivers to check the level of waste in bins. It makes the entire collection system smart and controllable by eliminating the overflow problems by reducing time on waste collection, vehicle numbers and fuel consumption. We can reduce the cost and protect the environment using such techniques.

Moreover, there is also a trend of suggestions and prediction in software-based application so our proposed system has a module that if bins are filled with waste but not respond by sensor then an authenticated person can intimate through application.

Summary of Requirements: (Initial Requirements)

The proposed system must follow the following requirements as follows:

1. There should be a web application from which admin can manage drivers, bins, and other things (funding, vehicle tracking, and bin configurations).

2. There should be a bin locator application from which a driver can locate which bin is filled and have to be collected from particular place.
3. There should be a module for authenticated user from which he can send bin location in case of inconvenient.

4. Mobile app has an extra module from which a driver can be located through GPS.

5. There should be a module of data analysis from which driver data can be analyzed as well as bin and admin can create reports from that analytic data.

2.1.2. Identifying External Entities

The Identification of External Entities is done in two phases.

a. Over Specify Entities from Abstract:

On the basis of the Abstract, one might identify the entities from the problem.

a) Admin
b) Agent
c) Bins
d) Employees
e) System
f) Routes
g) Reports
h) Payment

b. Perform Refinement:

After refining the over specified entities with the respect of our business domain we have the following external entities.

a) Agent
b) Admin
c) System
d) Bins
e) Routes

2.1.3. Capture "shall" Statements:

Para#	Initial Requirements
1	Admin “Shall” login with registered ID.
2	Admin “Shall” change passwords.
3	Admin “Shall” select routes.
4	Admin “Shall” add routes.
5	Employee “Shall” see routes.
6	System “Shall” intimate filled bins.

7	Employee “Shall” collect the waste.
8	Admin “Shall” logout from system.

2.1.5. Allocate Requirements:

Para#	Initial Requirements	Use Case Name
1	Admin “Shall” login with registered ID.	UC_Login
2	Admin “Shall” change passwords.	UC_Change_Password
3	Admin “Shall” select routes.	UC_Select_Routes
4	Admin “Shall” add routes.	UC_Add_Routes
5	Employee “Shall” see routes.	UC_See_Routes
6	System “Shall” intimate filled bins.	UC_Bins_Intimation
7	Employee “Shall” collect the waste.	UC_Waste_Collect
8	Admin “Shall” logout from system.	UC_Logout

2.1.6. Prioritize Requirements:

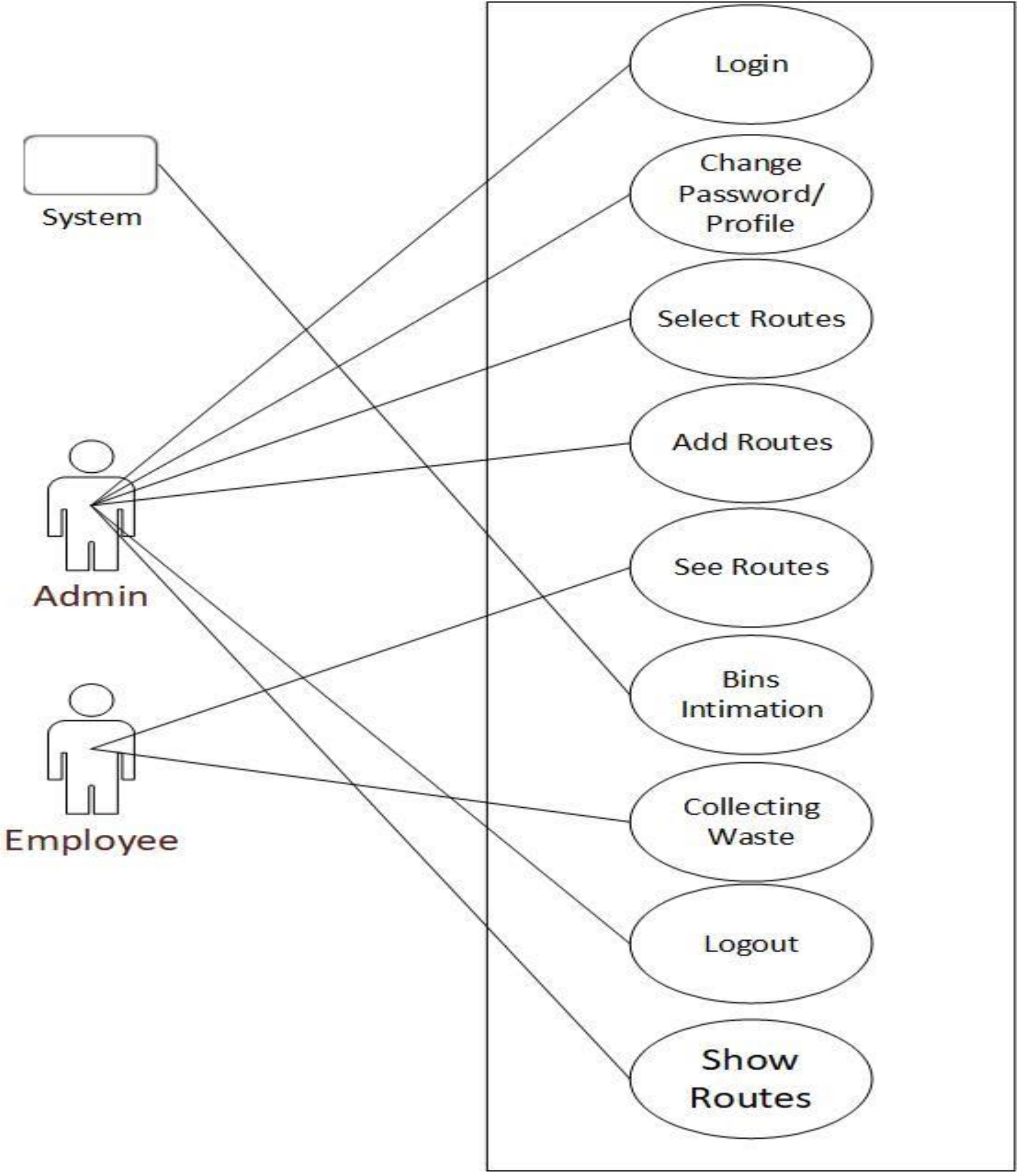
Para#	Priority	Initial Requirements	Use Case ID	Use Case Name
1	Highest	Admin “Shall” login with registered ID.	UC_01	UC_Login
2	Medium	Admin “Shall” change passwords.	UC_02	UC_Change_Password
3	Highest	Admin “Shall” select routes.	UC_03	UC_Select_Routes
4	Highest	Admin “Shall” add routes.	UC_04	UC_Add_Routes
5	Medium	Employee “Shall” see routes.	UC_05	UC_See_Routes

6	Highest	System “Shall” intimate filled bins.	UC_06	UC_Bins_Intimation
7	Highest	Employee “Shall” collect the waste.	UC_07	UC_Waste_Collect
8	Lowest	Admin “Shall” logout from system.	UC_08	UC_Logout

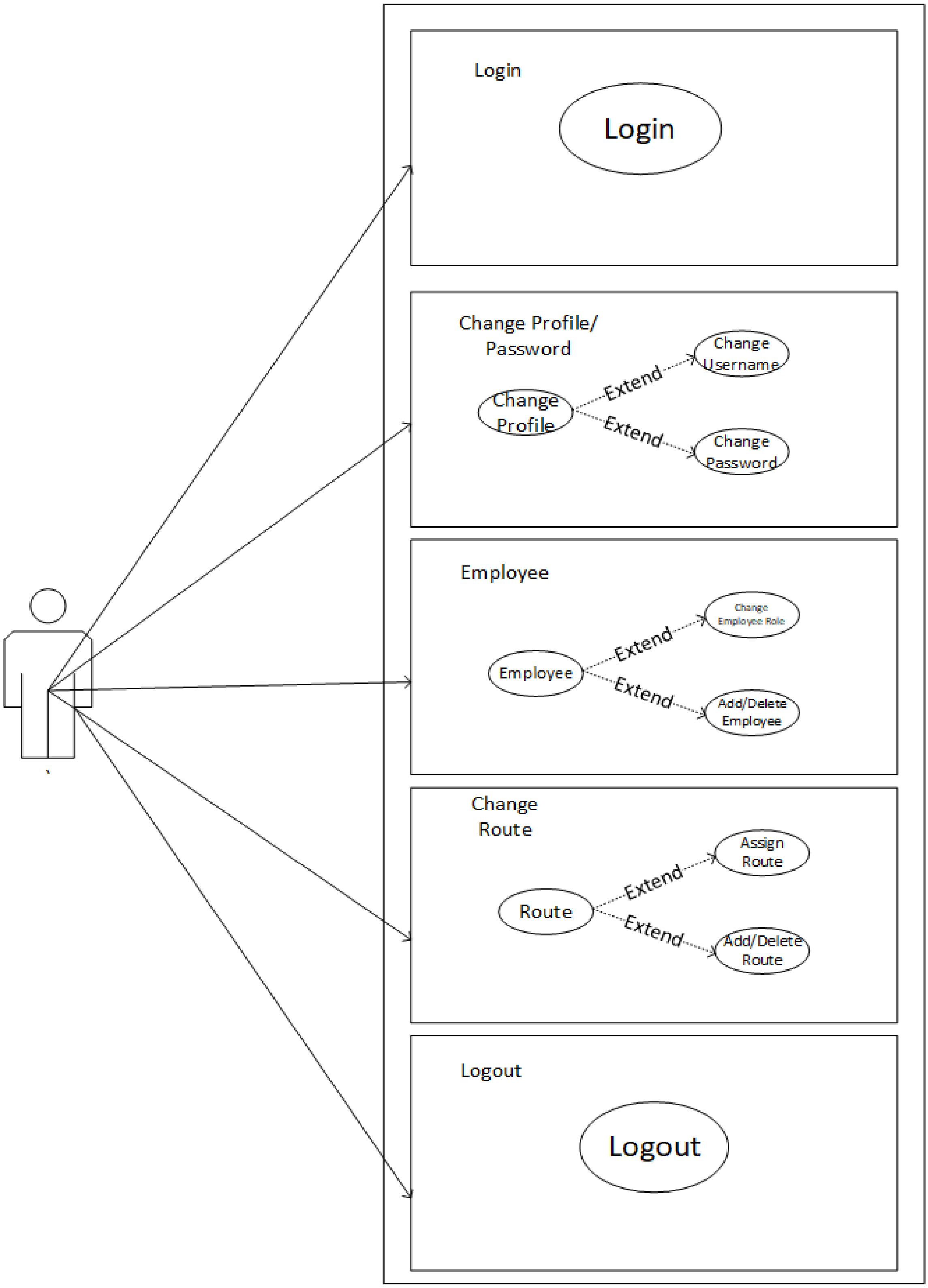
2.1.7. Requirements Trace-ability Matrix:

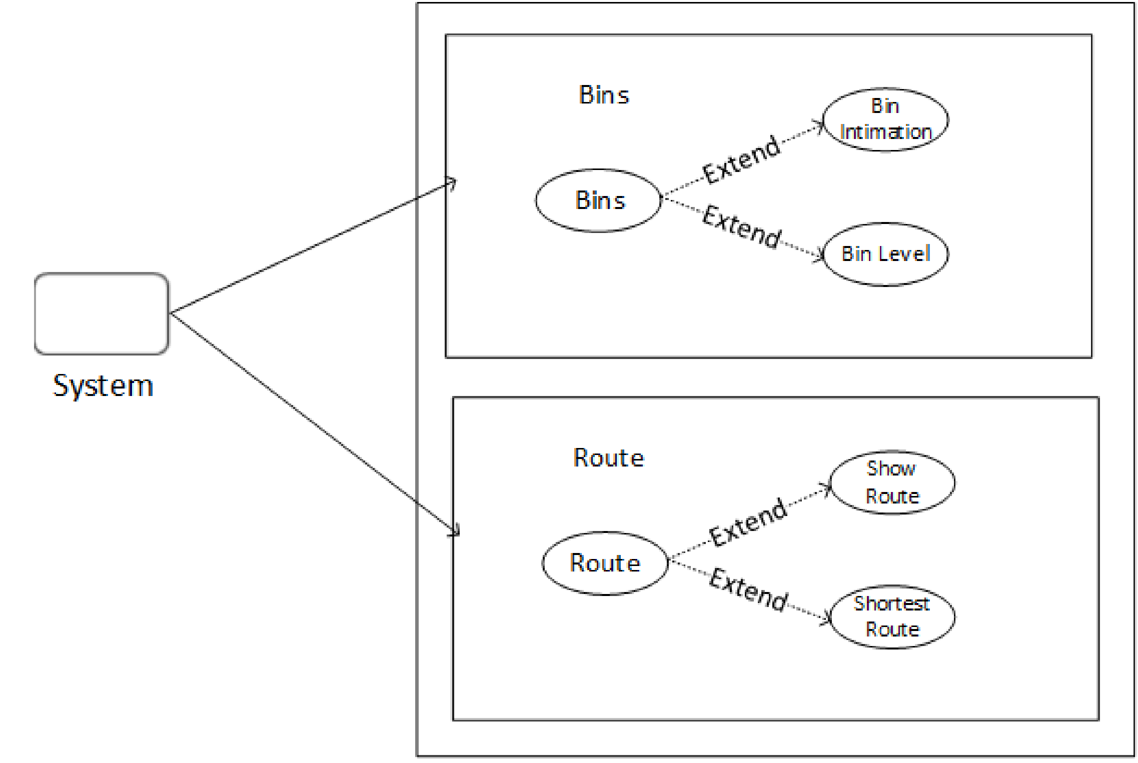
Para#	System Specification Text	Build	Use Case Name	Category
1	Admin "Shall" login with registered ID.	B1	UC_Login	Business
2	Admin "Shall" change passwords.	B1	UC_Change_Password	Business
3	Admin "Shall" select routes.	B1	UC_Select_Routes	Business
4	Admin "Shall" add routes.	B1	UC_Add_Routes	Business
5	Employee "Shall" see routes.	B1	UC_See_Routes	Business
6	System "Shall" intimate filled bins.	B1	UC_Bins_Intimation	Business
7	Employee "Shall" collect the waste.	B1	UC_Waste_Collect	Business
8	Admin "Shall" logout from system.	B1	UC_Logout	Business

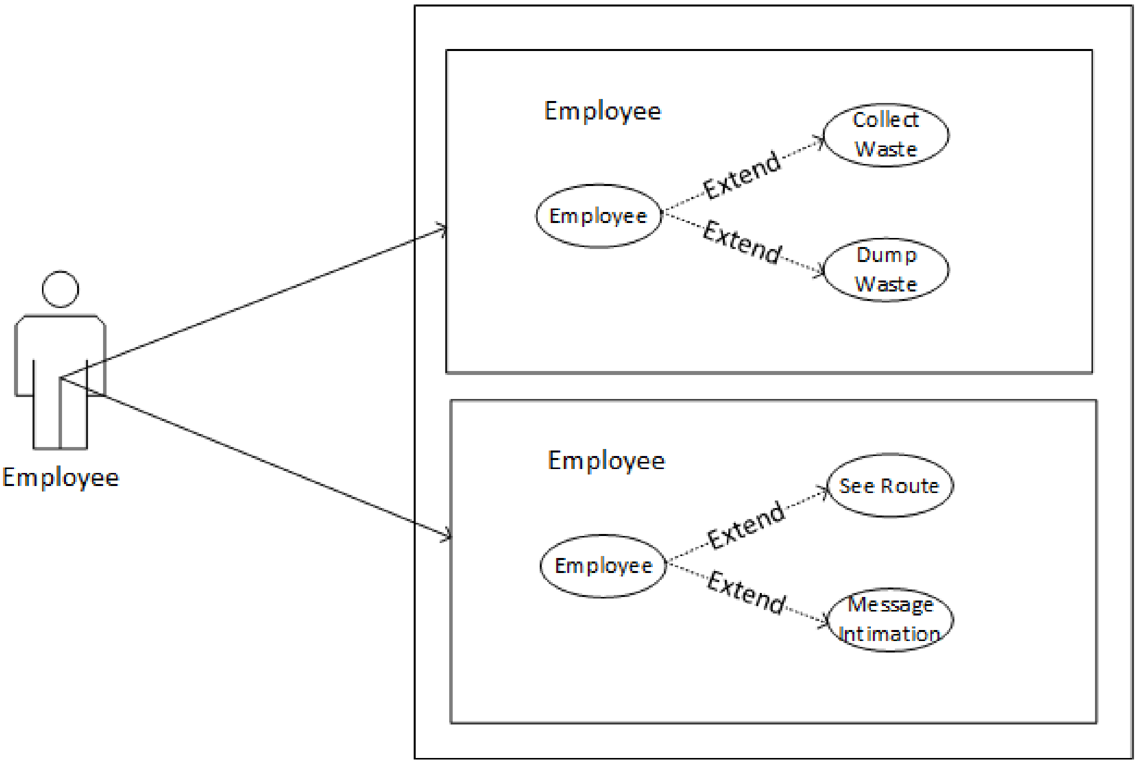
2.1.8. High Level Usecase Diagram:


System
Admin
Employee
Login
Change Password/ Profile
Select Routes
Add Routes
See Routes
Bins Intimation
Collecting Waste
Logout
Show Routes

2.1.9. Analysis Level Usecase Diagram




System
Bins
Bins
Extend
Extend
Bin Intimation
Bin Level
Route
Route
Extend
Extend
Show Route
Shortest Route


Employee
Employee
Employee
Extend
Extend
Collect Waste
Dump Waste
Employee
Employee
Extend
Extend
See Route
Message Intimation

Chapter 3: Design Document

3.1. Introduction:

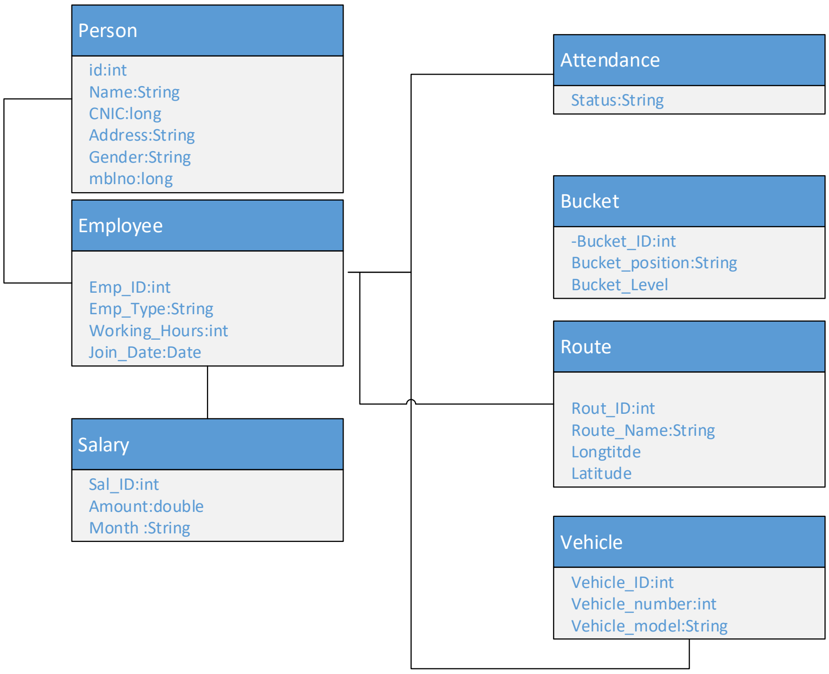
Design phase show system working state, sequence of process, structure of model and different process that include in system. Following artifacts must be included in the 3rd deliverable.

1. Domain Model
2. System Sequence Diagram
3. Sequence Diagram
4. Collaboration Diagram
5. Operation Contracts
6. Design Class Diagram
7. State Transition Diagram
8. Data Model

3.2. Domain Model

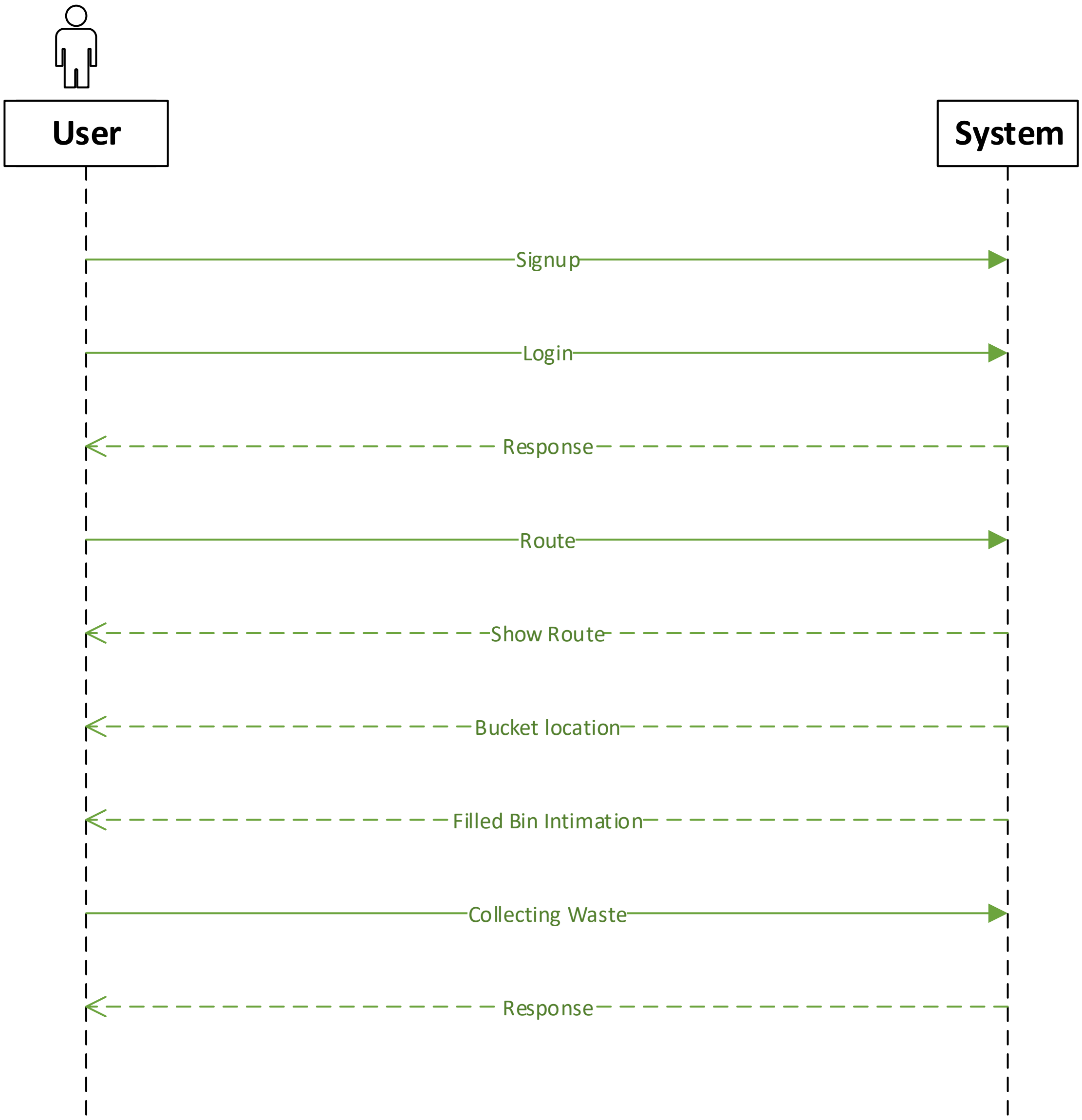
A domain model is a visual representation of conceptual classes or real - situation objects in a domain. Domain models have also been called conceptual models. Domain modeling is a technique used to understand the project problem description and to translate the requirements of that project into software components of a solution. The software components are commonly implemented in an object-oriented programming language. A domain model contains

- Identify Classes.
- Add associations necessary to record the relationships that must be retained.
- Add attributes necessary for information to be preserved.
- Apply existing Analysis Patterns.



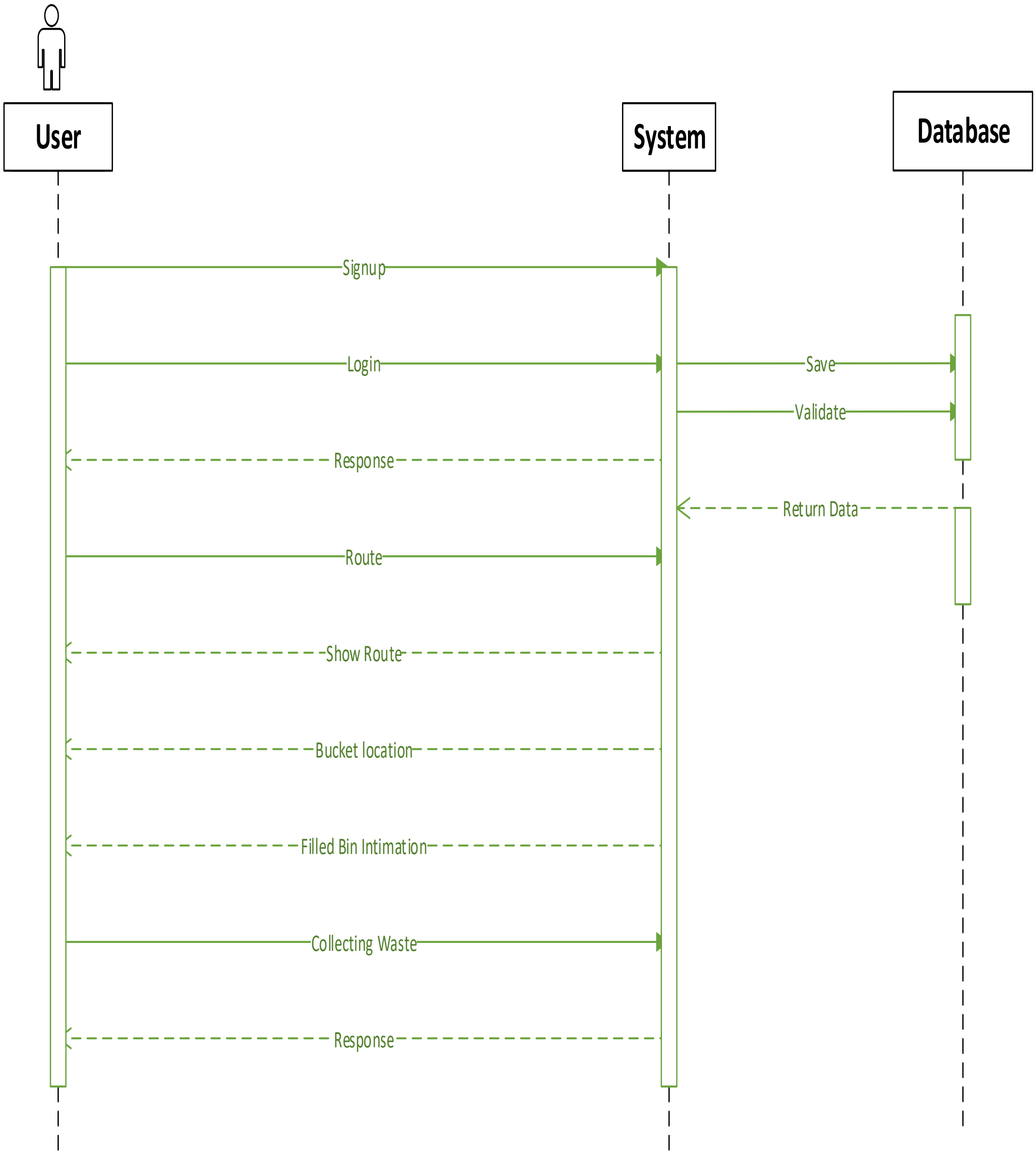
3.3. System Sequence Diagram

System sequence diagrams, also known as SSD, are actually a sub-type of sequence diagrams. SSDs are ideal for demonstrating when and how tasks are completed in a system, especially as those tasks relate to use cases. The UML System Sequence Diagram (SSD) describes events that enter the system from an external source. SSD will determine the system's events and activities. System Sequence Diagram shows interaction between system & actors. SSD is a sequence diagram that shows, for a particular scenario of a use case, the events that external actors generate their order, and possible inter-system events.



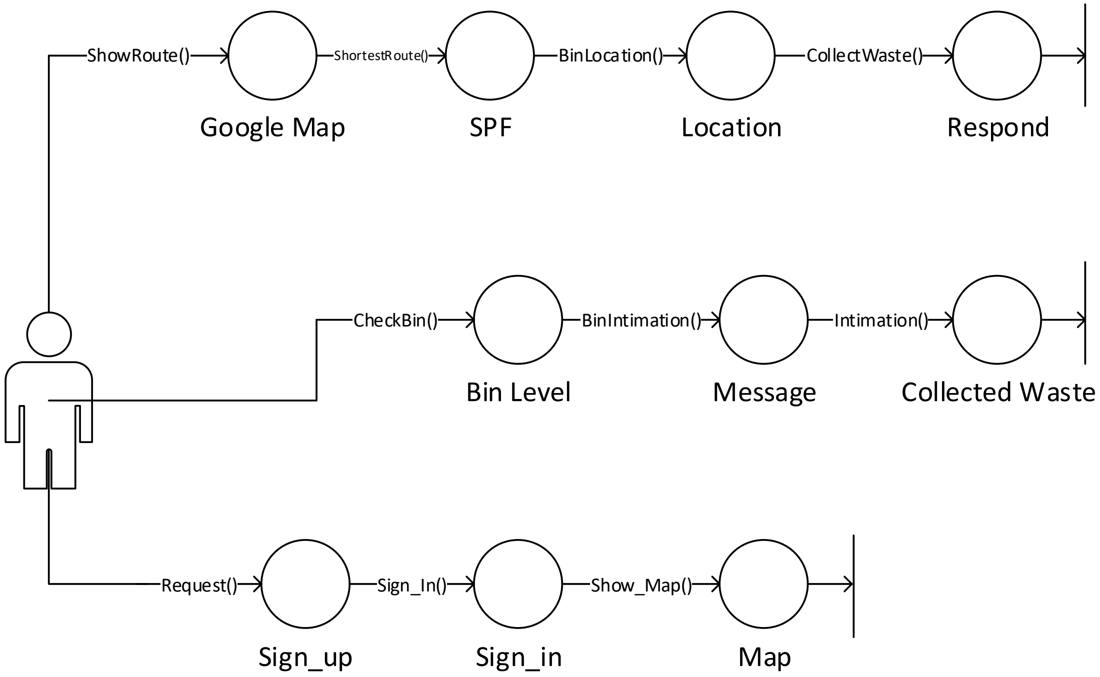
3.4. Sequence Diagram

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also diagram or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function.



3.5. Collaboration Diagram

A collaboration diagram, also called a communication diagram or interaction diagram. A collaboration diagram shows the objects and relationships involved in an interaction, and the sequence of messages exchanged among the objects during the interaction.



3.6. Operation Contracts

A **Contract C01**: Predict Price

Operation: predict()

Cross Reference: UC_Predict

Preconditions: Analysis should have been done on the data.

Postconditions: Predict prices will be displayed

Contract C02: Display Results

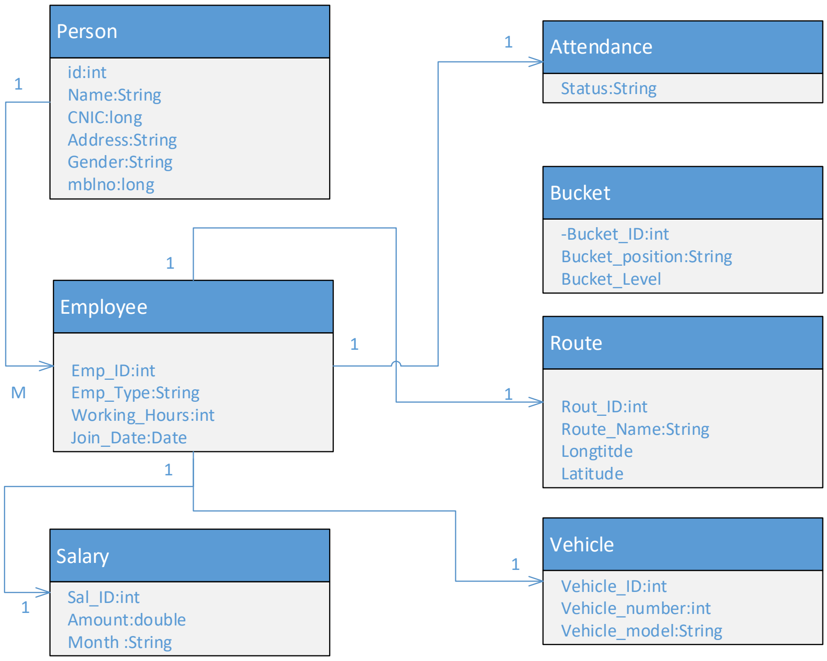
Operation: show()
Cross Reference: UC_Display
Preconditions: Analysis should have been done on the Tweets.

Postconditions: User will have a number of options to do next. User can read news, etc.

3.7. Design Class Diagram

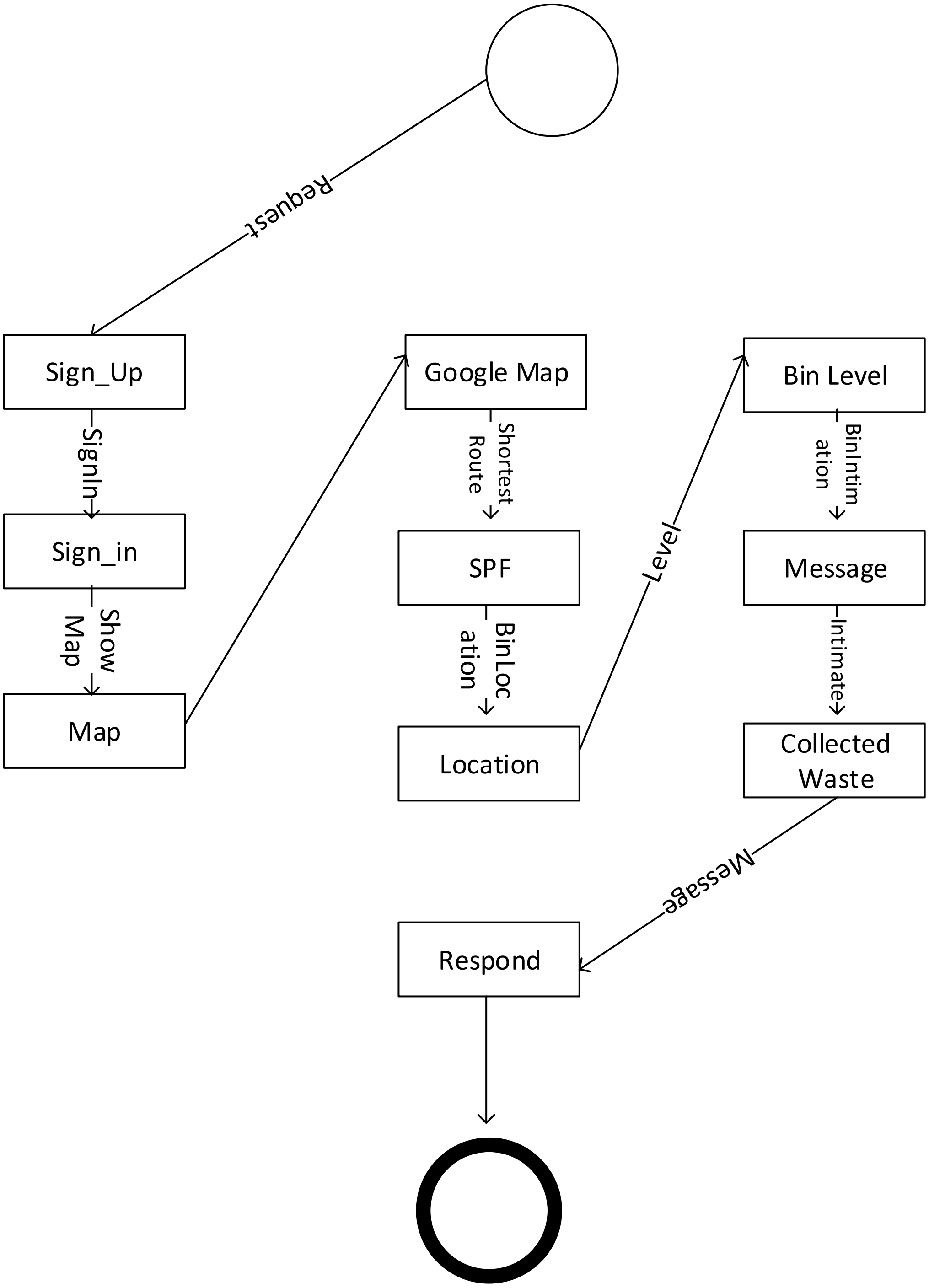
A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that illustrate the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity.

The main classes of Bio-informatics Annotator



3.8. State chart diagram

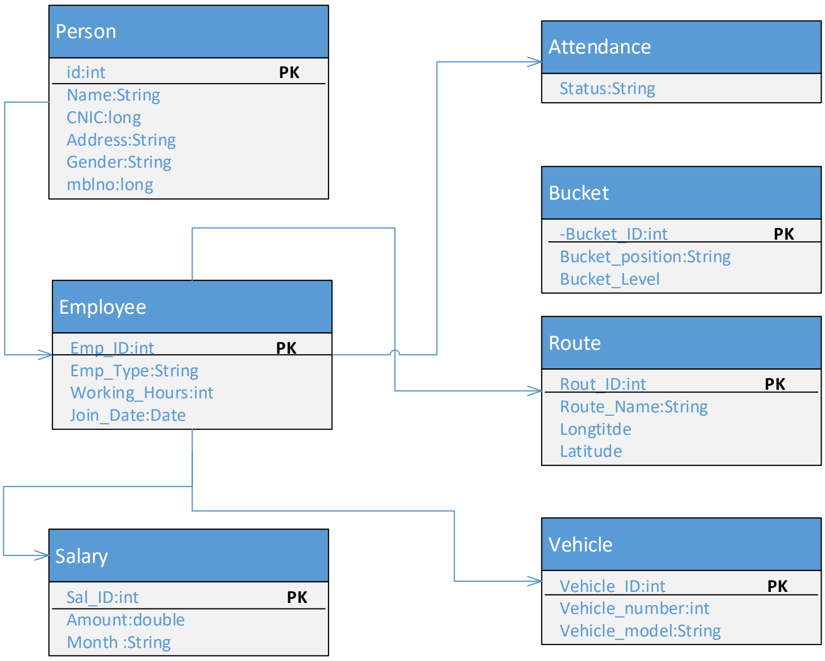
State chart diagram is one of the five UML diagrams used to model the dynamic nature of the system. They define different states of an object during its lifetime and these states are changed by events. State chart diagrams are useful to model the reactive systems. In our project, we have drawn State chart diagram using functionality provided by our app and according to the use case given in previous chapter.



3.9. Data Model

A data model refers to the logical inter-relationships and data flow between different data elements involved in the information world. ... Data models help represent what data is required and what format is to be used for different business processes.

The entities data model uses the terms event


Person
id:int PK
Name:String
CNIC:long
Address:String
Gender:String
mblno:long
Attendance
Status:String
Bucket
-Bucket_ID:int PK
Bucket_position:String
Bucket_Level
Employee
Emp_ID:int PK
Emp_Type:String
Working_Hours:int
Join_Date:Date
Route
Rout_ID:int PK
Route_Name:String
Longtitde
Latitude
Salary
Sal_ID:int PK
Amount:double
Month :String
Vehicle
Vehicle_ID:int PK
Vehicle_number:int
Vehicle_model:String

Chapter 4

User Interface Design

4.1. Introduction

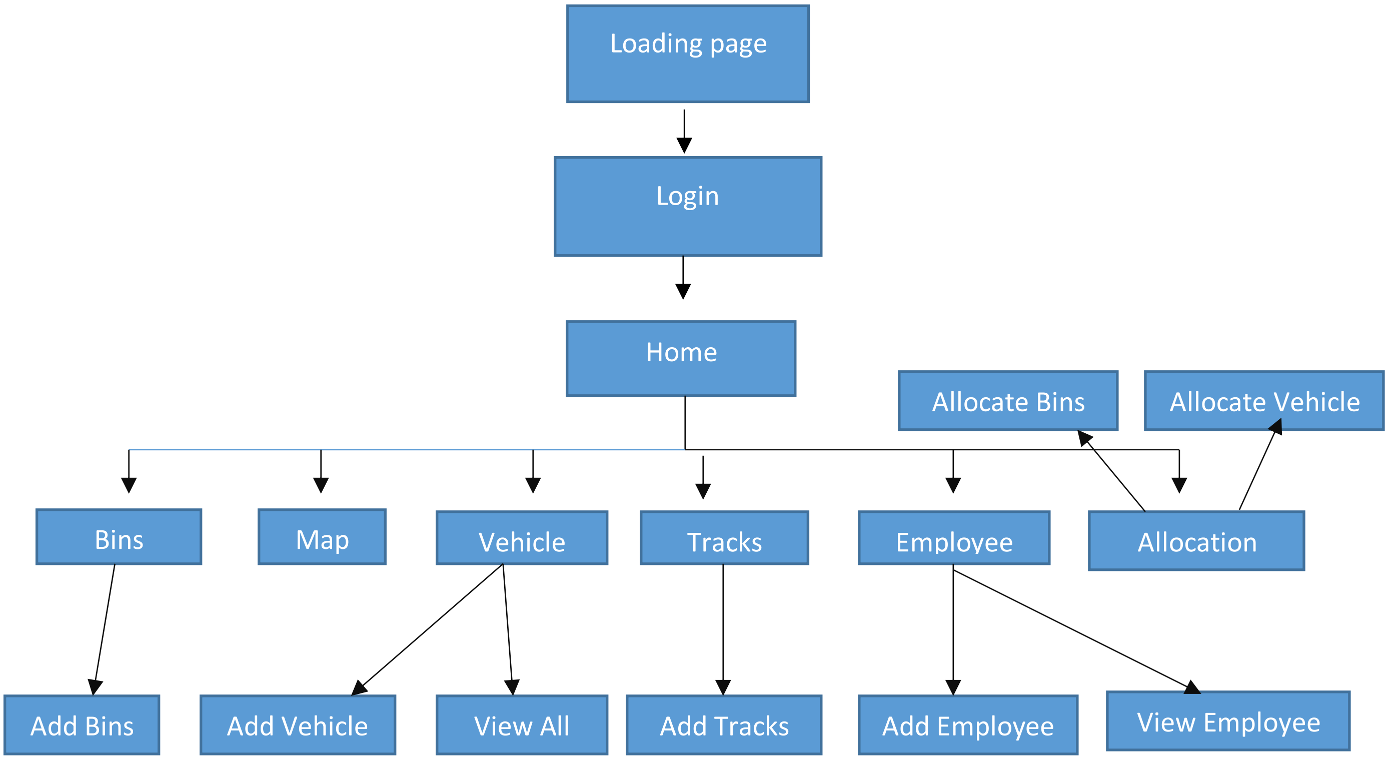
A user interface design consists of three main parts:

Page elements should be visualized on paper before building them in the computer. Just as you draw a site map to plan the site, use cartoons and storyboards to begin blocking out the site's appearance and navigational scheme.

a) Site maps
b) Storyboards

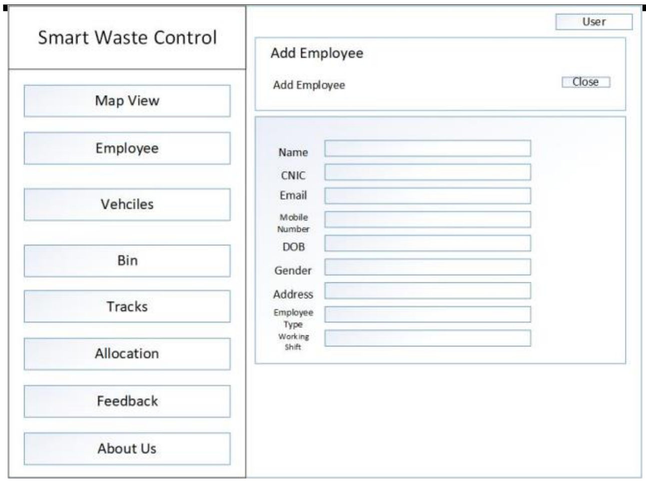
4.2. Site Maps

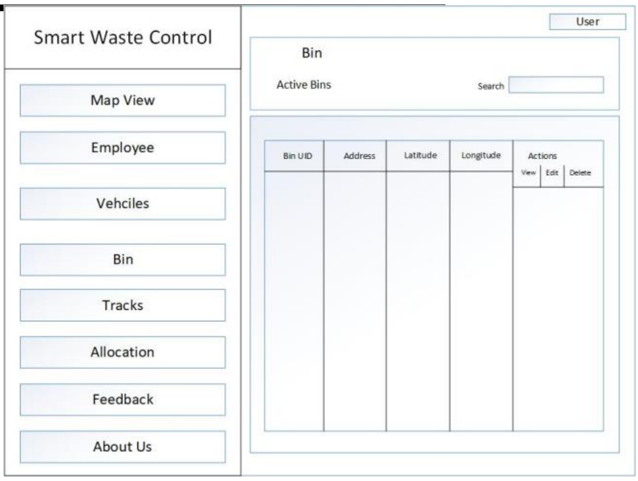
4.2.1. Smart Waste Control Sitemap



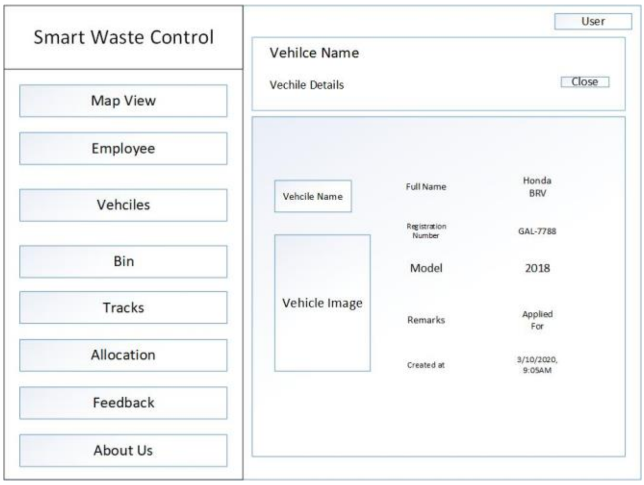
4.3 Storyboards

4.3.1 Storyboard of admin panel



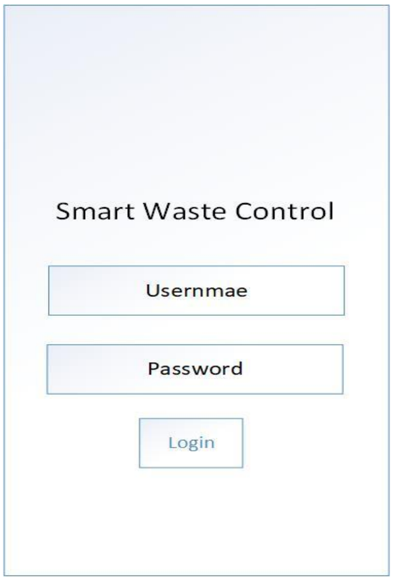


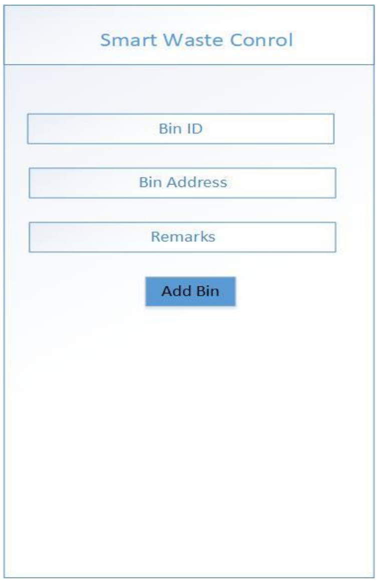
Bin UID	Address	Latitude	Longitude	Actions		
				View	Edit	Delete



Full Name	Honda BRV
Registration Number	GAL-7788
Model	2018
Remarks	Applied For
Created at	3/10/2020, 9:05AM

4.3.2 Storyboards of Mobile Application





Chapter 5

Software Testing

5.1. Introduction

From the start of software engineering different software testing schemes and testing templates has invented. These testing templates are used to test both UI and unit testing. Every credible software house developed their testing documentation for the purpose of developing efficient systems.

To provide a common set of standardized documents the IEEE developed the 829 Standard for Software Test Documentation for any type of software testing, including User Acceptance Testing.

In our testing module, we are going to test different features and modules of the system using documentation template supported by IEEE829.

5.2. Test Plan

5.2.1. Purpose

To prescribe the scope, approach, resources, and schedule of the testing activities. To identify the items being tested, the features to be tested, the testing tasks to be performed, the personnel responsible for each task, and the risks associated with this plan.

5.2.2. Outline

A test plan shall have the following structure:

a. Test plan identifier
b. Introduction
c. Test items
d. Features to be tested
e. Features not to be tested
f. Approach
g. Item pass/fail criteria
h. Suspension criteria and resumption requirements
i. Test deliverables
j. Testing tasks
k. Environmental needs
l. Responsibilities
m. Staffing and training needs
n. Schedule
o. Risks and contingencies
p. Approvals

5.2.2.1. Test Plan Identifier

The identifier for the test plan is **SWC AFTP.01.1**

The abbreviation for the identifier is as:

SWC	Smart Waste Control
AFTP	All features Test Plan
01.1	Version 1 and Revision 1

5.2.2.2. Introduction

Software test plan is the most core part of the testing phases on which all the other phases depends in one or another way.

Test planning phases defines and elaborate following points:

a) What tester must perform in testing
b) Standards of the quality to use in testing
c) Resources to be employed for testing
d) Schedule and time scale for the testing phase
e) Describes all the risks and contingencies which involved in testing and how they will be overcome.

5.2.2.3. Test Items

Following are the testing items which we are going to test in this plan:

a) External Database connection
b) Sign in test
c) Hardware Connection
d) Bins Location
e) Movement of bins location when moving bins
f) Driver path
g) Waste collect intimation
h) Category Panel
i) Dress Panel
j) All sensors working

All these are the generalized items which will in fact contain different tests to remove anomalies and defects from the Project.

5.2.2.4. Features to be Tested

Following is the list of all the features which will be tested:

a) Every driver must assign the area.
b) All sensors are working properly
c) Intimation when bin is full
d) Bins location displayed correctly
e) Waste level show correctly
f) Animations working correctly

5.2.2.5. Features Not to be Tested

As all the testing is going to be check on every feature, so it will make sure that nothing left without testing.

5.2.2.6. Approach

In android App and web app following two strategies are used

- **Unit Wise Testing**

  Unit testing is a testing software where we test individual methods, function and classes. Unit test covers almost 70 percent of test coverage.

In unit wise testing we test all features of mobile application, tasks are divided each member test the application and submit results to group leader.

5.2.2.7. Item Pass/Fail Criteria

All core functionality of the application and its features should function as expected result. All testing tasks will be performed within the ratio of 100% in which passing criteria is 75%. If the test results are less than 75%, it will be considered fail.

- If system don't work properly it will be considered as fail case.
- If expected page of application won't appear then it will be considered as fail case.

Each member of a group will test each feature and mark each case as Pass/ Fail. Each member will note the actual result and all relevant data or details. Once all tests will complete, the test leader will review the test report and check it is failing or pass.

If all the written tests are passed successfully then they are termed to be a successful test.

Pass/Fail Criteria for Unit Testing

a) A unit test will be considered as completed if it evaluates that the functionality which is tested is also there in the project

b) A unit test will be considered as fail if the time out happens during testing.

Pass/Fail Criteria for Instrumentation Testing

a) In UI testing everything will be tested by manual performing operations on the UI of application and a test will be considered pass if it will perform the desired operation on UI and the result in the UI is displayed as expected.

Now all the test which will not be able to follow the above criteria will be deemed as flaky test and considered as fail test.

5.2.2.8. Suspension Criteria and Resumption Requirements

The criteria for the suspension and resumption of testing shall be as follow

a) If the number or types of defects reach a point where the follow-on testing has no value, it will make no sense to continue the testing. In this case I will be sending tests for further reviews and development.

b) After completion of development and reviews and refactoring, testing shall resume.

5.2.2.9. Test Deliverables

After completion the result will be saved, and the tester will circulate the result to the group members and high authorities. The following are the test deliverable

Test plan document

Test design specifications document

Test case specifications document

Test procedure specifications document

Defective system Non-defective system

5.2.2.10. Testing Tasks

Task ID	**Testing task**

Task 1	Server connection
Task 2	Login page
Task 3	Registration
Task 4	Analysis
Task 5	Intimation
Task 6	Driver path
Task 7	Sensor

5.2.2.11. Environmental Needs

We need following environments for the execution of unit and instrumented tests

- Access of the internet.
- Android Mobile and App installed (Android lollypop 4.2 OS or higher).
- Android Studio
- Latest web page supporting browser.
- Latest operating system should be installed.

5.2.2.12. Responsibilities

The main person which is responsible for the testing execution and management is called test manager. In our documentation and project, the testing is performed by **Mr. Hamza Khalid.**

The other two members are also part of testing team in one way or another.

Following are the responsibilities table depicting jobs of different team member in planning, designing and executing tasks.

Testing Module	Performed by
Test Planning	Daud Rauf
Test Specification	Hamza Khalid
Test Case Development	Muhammad Atif Bashir
Test Writing	Hamza Khalid
Test Execution	Hamza Khalid

Table 5.1: Test Case Responsibilities

5.2.2.13. Staffing and Training Needs

There’s no need of any staff member for the testing of this project. All the required testing will be performed by the group members. They all have domain knowledge of their respective field.

Also, we have Supervisor for any the project related help.

5.2.2.14. Schedule

Following is the testing schedule which will be used:

Activities	**Responsible**	**Time Required**

Test Planning	Hamza Khalid	2 days
Test Specification	Muhammad Atif Bashir	4 days
Use case Testing	Muhammad Atif Bashir	4 days
User Interface Testing	Hamza Khalid	2 days

5.2.2.15. Risks and Contingencies

Some of the potential actions to be taken as a part of resolution of planning risks and contingencies are listed below.

a) The system would be rejected by the end users if they found it less interactive and difficult to use.
b) A proper documentation of requirement specification subsequently system specification is crucial. In the absence of these documents test failure can be the resultant of our effort.
c) All requirements and feature should be provided, and cross checked before the application is delivered to the client.
d) If defects remain at the end of designated development period, that do not fall into above categories then I will conduct a review of the defects to determine that which action should be taken.

5.2.2.16. Approvals

Test managers and other team members need to approve the overall test strategy.

a) Mr. Muhammad Atif Bashir ____________

b) Mr. Daud Rauf ____________

c) Mr. Hamza Khalid ____________

5.3. Test Design Specification

5.3.1. Purpose

Test design specification deals with the designing and predicting about the features which we are going to test. In this phase we also prioritize the test cases in their order of importance.

This is quite important document and is missed by some organization. This is the biggest problem that people try to execute and develop test without first designing them.

5.3.2. Outline

Following is the content which we are going to describe in test design specification.

a) Test design specification identifier
b) Feature to be tested
c) Approach Refinement

d) Testing identification
e) Feature pass fail criteria

5.3.2.1. Test design Specification Identifier

The identifier for the test design specification is **VRD DSI**

5.3.2.2. Features to be Tested

Following is the list of all the features which will be tested:

a) Validation on login test
b) Bins location
c) Sensor values are accurate
d) Sensor connection with esp32 are proper
e) Empty bins on waste collect
f) Change location when change the bins location
g) Intimations
h) Analysis test

5.3.2.3. Approach Refinement

We will be using the same strategy and approach as mentioned in the test plan phase.

5.3.2.4. Testing Identification

In this phase we make a list of testing features which can be tested and can be taken as a test case in the future step of testing.

In our testing strategies we are taking all the test items as test cases.

5.3.2.4. Feature Pass/Fail Criteria

Testing tasks will be performed within the ratio of 100% in which passing criteria is 75%. If the test results are less than 75%, it will be considered fail. If all the written tests are passed successfully then they are termed to be a successful test. Each member of a group will test each feature and mark each case as Pass/ Fail. Each member will note the actual result and all relevant data or details. Once all tests will complete, the test manager or leader will review the test report and check it is failed or pass.

5.4. Test Case Specification

5.4.1. Purpose

The purpose of test case specification is to guide the end user about the system usage and its relevant Application. It identifies the required inputs and expected results, provides step-by-step procedures for executing the test

5.4.2. Outline

A test case specification shall have the following structure:

a. Test case specification identifier
b. Test items
c. Input specifications
d. Output specifications

e. Environmental needs
f. Special procedural requirements
g. Inter case dependencies

5.4.2.1. Test Case Specification Identifier

Following are test cases identifier that will be used in our test:

Serial No	Test Feature	Test Case Identifier
1	Validation on login test	TC1
2	Bins Location	TC2
3	Data Retrieval is proper form external server	TC3
4	Sensor values accuracy	TC4
5	Sensors connection with esp32	TC5
6	Empty bins on waste collect	TC6
7	Change location of bins when move bins	TC7
8	Intimation	TC8
9	Analysis test	TC9

5.4.2.2. Test Items

Following is the list of all the features which will be tested:

a) Validation on login test
b) Bins location
c) Sensor values are accurate
d) Sensor connection with esp32 are proper
e) Empty bins on waste collect
f) Change location when change the bins location
g) Intimations

5.4.2.3. Input Specifications

Here we will specify the input for every test case:

Serial Number	Test Case Input	Test case identifier
1	Sensor values	TC1
2	Bins installment	TC2
3	Employee data	TC3
4	Track Data	TC4

5.4.2.4. Output Specifications

Here we will specify the output for every case:

Serial Number	Test Case Output	Test Case Identifier
1	Display Route to driver	TC1
2	Specify whether bin is empty or not	TC2
3	Display assigned trucks to driver	TC3

4	Display assigned tracks to driver	TC4
5	Display bins location on google maps	TC5
6	Specify data is retrieved or not	TC6
7	Display Bins waste level	TC7
8	Display total vehicles, bins and employee on home page	TC8
9	Previous records	TC9

5.4.2.5. Environment Needs

5.4.2.5.1. Hardware

Following test cases need hardware for their successful execution:

Serial Number	Hardware Required	Test Case Identifier
1	Esp32	TC1
2	Arduino	TC2
3	Battery	TC3
4	Ultrasonic sensors	TC4
5	GPS	TC5

5.4.2.5.2. Software

Following are the software required by some use cases for their successful execution.

a) Visual code
b) Arduino IDE
c) Postman
d) Android Studio

Following is the list of test cases which need specific software for their successful execution.

Serial Number	Software Required	Test Case Identifier
1	Arduino IDE	TC1
2	Arduino IDE	TC2
3	Visual Code, Arduino IDE, Postman	TC3

5.4.2.6. Special Procedural Requirements

There are no other special procedural requirements for the execution of these tests cases.

5.4.2.7. Inter Case Dependencies

Serial Number	Test Case Input	Test case identifier	Test Case Dependency
1	Sensors values	TC1	-
2	Employee Data	TC2	TC6
3	Bins Location	TC3	TC6, TC1

4	Bins Level	TC4	TC1
5	QR Code generated for picture	TC5	TC1,TC3
6	GPS	TC6	TC1
7	Ultrasonic Sensors	TC7	TC1
8	Waste Level	TC8	TC1,TC7
9	Tracks record	TC9	-

5.5. Test Procedure Specification

5.5.1. Purpose

To specify the steps for executing a set of test cases or, more generally, the steps used to analyze a software item in order to evaluate a set of features.

5.5.2. Outline

Following are the contents of test procedure specification:

a) Test procedure specification identifier
b) Purpose
c) Special requirements
d) Procedure steps

5.5.2.1. Test Procedure Specification Identifier

The identifier for the test procedure specification is **SWC TPSI**

5.5.2.2. Purpose

Describes the purpose of this procedure. If this procedure executes any test cases, provide a reference for each item. In addition, provide references to relevant sections of the test item documentation (e.g. references to usage procedure)

5.5.2.3. Special Requirements

To proceed with the testing in desktop application one should have knowledge of following techniques:

a) Unit testing
b) UI testing with manual testing
c) Arduino programming
d) Application should be working properly
e) Keep the esp32 connected with internet

5.5.2.4. Procedural Steps

Following are the procedural steps for the successful execution of tests.

5.5.2.4.1. Log

Logging can be performed using following techniques:

a) Manual Logging
b) Debug Logging

Table 5.7: Dependency table

We will be using both kind of logging depending upon the scenario.

5.5.2.4.2. Set Up

a) Application should be working properly
b) Keep the esp32 connected with Wi-Fi.
c) Keep Sensors connected with esp32

5.5.2.4.3. Start

We need to keep esp32 connected with internet(Wi-Fi). Also keep sensors connected to esp32.

5.5.2.4.4. Proceed

We must execute test cases in the order of their dependency. Test case with no dependencies or solved dependency first.

5.5.2.4.5. Measure

The measurement whether a test is correct or not is solely based on the human observation.

5.5.2.4.6. Shut down

Put the server offline and shut down the application.

5.5.2.4.7. Restart

Simple connect again esp32 with Wi-Fi to getting results

5.5.2.4.8. Stop

To stop the test orderly, first we will have to wait for the competition of execution. Then disconnect esp32 with system.

5.5.2.4.9. Wrap Up

Now to test the code again we will again change the execution mode from .exe file development to test execution.

5.5.2.4.10. Contingencies

We need to submit bug reports. Test manager will use that information to create incident reports.

5.6. Test Item Transmittal Report

5.6.1. Purpose

To identify the test items being transmitted for testing. It includes the person responsible for each item, its physical location, and its status. Any variations form the current item requirements and designs are noted in this report.

5.6.2. Outline

Following are the contents of test case specification:

a) Transmittal report identifier
b) Transmittal items
c) Location

d) Status
e) Approvals

5.6.2.1. Transmittal Report Identifier

The identifier for the transmittal report is **SWC TRL**

5.6.2.2. Transmitted Items

All the items which are mentioned in the Test Plan are part of the test transmittal report. We should also provide all those tests which are failed and again revised and developed for refactoring for further testing.

As we are a small team of three members working in this project so, **Mr. Muhammad Atif Bashir** and **Mr. Hamza Khalid** are responsible for test planning and specification development as indicated in Testing Responsibilities. Whereas these test case specifications then will be transmitted to **Mr. Daud Rauf** for successful running of the test.

5.6.2.3. Location

All the test documents including most important test case specification will be placed in a PDF file and will be transmitted to test manager and test executor by email.
After receiving email tester will be able to convert test case specifications into actual written test cases.

5.6.2.4. Status

At this phase all the test cases which are provided are not tested prior so their status in unresolved or so we can say false. After the testing will be done then if there will be any problem in code, a request will be made to the developers to again correct and refactor the code.

5.6.2.5. Approvals

a) Mr. Muhammad Atif Bashir ___________

b) Mr. Daud Rauf ___________

c) Mr. Hamza Khalid ___________

5.7. Test Log

5.7.1. Purpose

The purpose of the test log is to provide a sequential record of relevant details about the execution of tests. This log represents the results of testing of whole system.

5.7.2. Outline

Following will be the content of test log.

a) Test log identifier
b) Description
c) Activity and event entries

5.7.2.1. Test Log Identifier

The identifier for test log is **SWC TLI**

5.7.2.2. Description

Here are all the test items which are planned to be tested and are transmitted to the test manager.

Serial No	Test Feature	Test Case Identifier
1	Validation on login test	TC1
2	Bins Location	TC2
3	Data Retrieval is proper form external server	TC3
4	Sensor values accuracy	TC4
5	Sensors connection with esp32	TC5
6	Empty bins on waste collect	TC6
7	Change location of bins when move bins	TC7
8	Intimation	TC8
9	Analysis test	TC9

Software Requirements

Software requirements which are included are as:

a) Arduino IDE
b) Android Studio
c) Visual Code

Hardware Requirements

Hardware requirements which are included as:

a) Corei3 CPU
b) 256 GB HDD
c) ESP32
d) Ultrasonic
e) GPS

5.7.2.3. Activity and Event Entries

For each event, including the beginning and end of activities, record the occurrence date and time along with the identity of the author.

5.7.2.3.1. Execution Description

While executing the tests whole team was present so that all the team get through understanding of testing process.

The tests were executed on April 20, 20120 in Android Studio, Arduino IDE and Visual Code with all hardware and software specification already provided.

5.7.2.3.2. Procedure Results

Following is the complete result of our test execution based on the pass/fail criteria already mentioned in planning phase.

Serial Number	Test Case Identifier	Test Revision	Test Case Result	Test Status
1	TC1	1st	Display Route to driver	Pass
2	TC2	1st	Specify whether bin is empty or not	Pass
3	TC3	1st	Display assigned trucks to driver	Pass
4	TC4	1st	Display assigned tracks to driver	Pass
5	TC5	1st	Display bins location on google maps	Pass
6	TC6	1st	Specify data is retrieved or not	Pass
7	TC7	1st	Display Bins waste level	Pass
8	TC8	1st	Display total vehicles, bins and employee on home page	Pass
9	TC9	1st	Previous records	Pass

5.7.2.3.3. Environment Information

- Windows 7,8,10
- Ram 4GB (minimum) and 8GB (recommended)

5.7.2.3.4. Anomalous Events

Our tests are successful passed and no anomaly occurred except in one case which will be later investigated. But once developer start writing more code, problems can occur at any time. So, it is necessary to use test driven approach.

5.7.2.3.5. Incident Report Identifiers

Record the identifier of each test incident report, whenever one is generated.

5.8. Test Incident Report

5.8.1. Purpose

To document any event that occurs during the testing process that requires investigation.

5.8.2. Outline

A test incident report shall have the following structure.

a. Test incident report identifier
b. Summary
c. Incident description
d. Impact

5.8.2.1. Test Incident Report Identifier

The identifier for incident report is **SWC IRI.**

5.8.2.2. Summary

During the execution of test, we have faced an anomaly when one of the tests get failed. TC1 which get failed. In this section we will analyze and investigate the reason behind failure of test case.

5.8.2.3. Incident Description

Provide a description of the incident. This description should include the following items:

a. Inputs
b. Expected results
c. Actual results
d. Anomalies
e. Date and time;
f. Procedure step;
g. Environment;
h. Attempts to repeat;
i. Testers;
j. Observers.

5.8.2.4. Impact

This is the important function provided by Application After the test incident description it was a major impact on test case.

5.9. Test Summary Report

5.9.1. Purpose

To summarize the result of the designated testing activities and to provide evaluations based on these results.

5.9.2. Outline

A test summary report shall have the following structure:

a) Test summary report identifier
b) Summary
c) Variance
d) Comprehensive assessment
e) Summary of results
f) Evaluation
g) Summary of activities
h) Approvals

5.9.2.1. Test Summary Report Identifier

Test summary report identifier is as **SWC SRI.**

5.9.2.2. Summary

All the functionalities of application are tested as per the test plan. Each test is performed according to its specified plan and test log against each test is prepared accordingly. All components were tested, there were some bugs as expected but later they were fixed. And now, each component is working properly.

5.9.2.3. Variances

Every test is executed according to the test plan, test specification and test case specification. We follow the testing approach and strategy stated in the test planning phase. Test were executed according to test procedure specification and they are deemed passed and failed according to testing criteria mentioned.

5.9.2.4. Comprehensiveness Assessment

All the features described in the test planning phase are tested and no test case is left out of testing due to any sort of reason. Our test coverage report is more than 80 percent.

5.9.2.5. Summary of Results

Summary Assessment	Total Number of Test Cases
Test Cases Planned	9
Test Cases Run	9
Test Cases Reviewed	2
Test Cases Passed	7
Test Cases Failed	0

5.9.2.6. Evaluation

Testing process was simple and enough for this phase. Active and comprehensive evaluation of problems found at last phase led us to conclusion, that user able to perform sentiment analysis.

5.9.2.7. Summary of Activities

There are three development phases in testing. The first one was planning followed by test specification and designing and test execution. All these test activities and phases are planned and executed according to Gantt chart. Here is the breakdown of all the activities with their time and staff who was responsible.

Testing activities	Person Responsible	Time Elapsed
Test Planning	Mr. Daud Rauf	5 Days
Test Designing	Mr. Muhammad Atif Bashir	7 Days

Test Execution	Mr. Hamza Khalid	2 Days

5.9.2.8. Approvals

We need approval of test planner, test designer and test manager.

a) Mr. Muhammad Atif Bashir ____________

b) Mr. Daud Rauf ____________

c) Mr. Hamza Khalid ____________

Chapter 6

User Help Manual

6.1. General Information

Main reason behind excessive pollution is absence of a waste management system to manage garbage. People dump garbage in open, empty spaces available in the vicinity. It creates unhygienic condition for the people and creates bad smell around the surroundings, which leads spreading diseases to the people living in such environment. A waste management system works by placing bins in different areas and collecting waste from them after regular intervals. In some areas, bins may overflow due to unavailability or frequency of garbage collection vehicles.

To avoid such situations, we are going to implement a project called IOT Based waste management using smart dustbin. In this system, multiple dustbins are located throughout the city or the Campus. These dustbins are provided with a sensor which detects the level of the waste and a GPS device for getting the location and to identify which bin is full. When the level of the bin exceeds, the device will transmit the reading along with the unique ID provided. It will inform the status of each bin in real time so that concerned authority can send the garbage collection vehicle only when the bin is full. By implementing this system resource optimization, cost reduction, effective usage of smart bins can be done

6.2. System Overview

Smart waste control is a desktop application and mobile application which allow user to demand bins and other services like sanitary and manual sweeping. Admin panel can allow to add trucks and employees and bins to the systems. User can locate bins from their respective application. Also, they see the levels of bins.

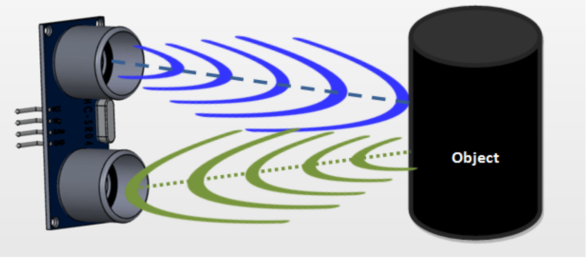
6.3. System Components

6.3.1. ESP32

A feature-rich MCU with integrated Wi-Fi and Bluetooth connectivity for a wide-range of applications. ESP32 can perform as a complete standalone system or as a slave device to a host MCU, reducing communication stack overhead on the main application processor. ESP32 can interface with other systems to provide Wi-Fi and Bluetooth functionality through its SPI / SDIO or I2C / UART interfaces.

a) Ultrasonic Sensor

HC-SR04 Ultrasonic (US) sensor is a 4-pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver.



b) GPS

GPS sensors are receivers with antennas that use a satellite-based navigation system with a network of 24 satellites in orbit around the earth to provide position, velocity, and timing information.

6.4. Installation Guide

a) System Requirement

OS: Windows 10 64-bit
Processor: Intel core i3 or higher
Memory: Minimum 2GB
HDD Space: 50GB for better performance
Android: Version 5.0 Lollipop or higher

b) Software Installation

1) **Arduino IDE**
2) **Node.js environment**
3) **Android Studio**

c) Hardware Installation

Connect esp32 with 5V battery and connect GPS and Ultrasonic sensors with it. Embed code connect esp32 with nearby Wi-Fi which help to get values of garbage level and position of the bins

6.5. How to Use the System?

a) Smart Waste Control Admin Panel

It is very simple to use open login to it; Dashboard will be visible in the front of you. Where you can add/delete/update Employees, Trucks, bins and other data.

b) Smart Waste Control

Go to website request us to any services that we provide to user.

c) How the actual User will use?

Actual user means the person who wants the use the “Smart Waste Control”. Can use user ended website where all the details are deployed. User may request bins or any other services to us.