

Waste Monitoring System

U.D.R.L. Vijayanthi
2020



Waste Monitoring System

**A dissertation submitted for the Degree of Master of
Information Technology**

U.D.R.L.Vijayanthi
University of Colombo School of Computing
2020



Abstract

Waste management becomes a major problem all over the whole world. Solid waste management is the proper management of the disposal of wastes by its production and recycling. Therefore, solid waste management can be defined as systematic control, collection, storage, transportation, processing, and disposal of solid waste production. Waste management is important because it helps in maintaining the cleanliness of the whole world. It needs to be practiced because the entire population's health and protection depend on it. The Importance of solid waste management is Its Protects the environment, reduces all types of waste, Saves the earth and Recycling helps you to get money

WMS is a web-based software solution with a web application integrated with a mobile solution. It is facilitated for monitoring the waste collection process for an organization.

The main problem is addressed in this project is to give an automated solution for any organization private or government who are handling waste collecting and monitoring, when having a large waste to manage and a manual process to monitoring waste. When this task is done manually, we can see people face a lot of problems without a proper solution for waste collecting and monitoring. Now a day's waste is a major problem for the society without having a proper system for collecting and monitoring the waste.

This project is introduced a sophisticated system for send requests to collect the waste, scheduling transporters with waste categories, track the transporters while working, selling recycle items to regular buyers, maintaining transporter details, maintaining buyer details. Maintain Client requests. In brief clients or users of the Waste Monitoring System can register with the system via mobile application and able to send a request as a single or schedule for the collection of the waste by using a smartphone with a "Smart Bin" mobile application. The administrator can register the transporters with a truck and able to map registered transporters with waste categories and available routes of the region. Transporters can start a job via a mobile application, and they can update the status of the request after done it. Administrators can track the transporters and sell the collected recycle waste to the registered buyers.

This project is used HTML, CSS, Angular, and Bootstrap, and C # as a programming language to develop web applications and used Android Studio to develop the mobile application.

Declaration

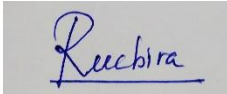
The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge, it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: Mrs. U.D.R.L. Vijayanthi

Registration Number: 2016/MIT/080

Index Number: 16550809



Signature:

Date: 21st June 2020


This is to certify that this thesis is based on the work of

Mr./Ms. U.D.R.L. Vijayanthi

Under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Dr.Ajantha Atukorale



Signature:

Date: 21st June 2020

Acknowledgments

I would like to extend my appreciation to my supervisor, Dr. Ajantha Atukorale, Senior lecturer of UCSC, for supervising me and providing useful guidance throughout the entire project.

I would like to thank our project coordinator Mr. Viraj Welgama for giving his assistance at any time throughout the entire project course.

Furthermore, my sincere gratitude to the staff of Abans Environmental services (Pvt) Ltd and Gampaha Municipal Council for sharing their experience and necessary information when gathering the requirements.

Finally, I would like to express my gratitude to my parents, sister and beloved husband, and all batch mates for the support and guidance you have given throughout the entire project course to make a success of the project.

Table of Contents

Abstract.....	ii
Declaration.....	iii
Acknowledgements	iv
Table of Contents	v
List of Figures.....	ix
List of Tables	xii
List of Abbreviations	xiii
Chapter 1	1
Introduction to Waste Monitoring System	1
1.1. Introduction.....	1
1.2. Problem Specification	1
1.3 Motivation for the project	2
1.4 Objectives of the project.....	2
1.4 Scope of the project.....	3
1.4.1 Functions within of the system.....	3
1.4.2 Mobile Application.....	4
1.4.3 Web Application.....	4
1.5 Overview of the Report.....	5
Chapter 2	6
Background	6
2.1 Introduction.....	6
2.2 Background of the Business	7
2.3 Overview of Current Process and Drawbacks	7
2.3.1 Analyzing the current system.....	7
2.3.2 Drawbacks of the current system	8
2.4 Requirements of the proposed project	8
2.4.1 Functional Requirements of the project	8
2.4.2 Non-Functional Requirements of the Project.....	10
2.5 Review of similar Waste Management Systems	10
2.5.1 Related Works.....	10
2.5.2 Research Articles.....	12
Chapter 3	13
Methodology	13

3.1. Introduction	13
3.2. Requirement Analysis	13
3.3. Fact Finding Techniques	13
3.3.1 Observe similar systems	13
3.3.2 Interviews	13
3.3.3 Review of available documents and manuals	14
3.4 Requirements for the new system	14
3.4.1. Functional Requirements	14
3.4.2. Non-Functional Requirements	15
3.5. Design of the system	15
3.6. Software Architecture	16
3.6.1. Proposed Architecture for the Mobile Application	17
3.7. System Software & Hardware Requirement	18
3.7.1. Web application - Hardware requirements	18
3.7.2. Web application - Software requirements	18
3.7.3. Mobile application - Hardware requirements	18
3.8. Database Design	18
3.8.1. ER Diagram	19
3.8.2. Use case Diagram	20
3.8.3. Descriptive Use Cases	23
3.8.4 Class Diagram	27
3.8.5 Sequence Diagram	28
3.9 Testing	30
3.9.1 V Model	30
3.9.2 Integration Testing	30
3.9.3 System Testing	30
3.9.4 Acceptance Testing	31
3.10 Test Plan of the WMS	31
3.11 Implementation Details	32
3.11.1 Tools and Technologies	33
3.12 Main Interfaces of the System	34
3.12.1 Login Page of the Web Application	34
3.12.2 Dashboard of the Web Application	35
3.12.3 Transporter Registration Page of the Web Application	35
3.12.4 Transporter waste category - web application	36

3.12.5 Recycle Details Page - Web Application	36
3.12.6 Welcome and Login Page - Mobile Application	37
3.12.7 Client Registration - Mobile Application	38
3.12.8 Client request page - mobile application	39
Chapter 4	40
Evaluation	40
4.1 Introduction	40
4.1.1 Validation	40
4.1.2 Verification	40
4.2 Testing Types	41
4.3 Testing Types carried on the Application	42
4.3.1 Static testing within WMS	42
4.3.2 Dynamic Testing within WMS	42
4.4 Test cases of the Mobile Application	43
4.4.1 Login functions and Client Registration	43
4.4.2 Client Request	45
4.4.3 Automation Scripts	48
4.4.3 User Evaluation Questionnaire	49
4.5 Test Deliverables	50
4.6 Lesson Learnt	51
4.7 Achievement of Objectives	51
Chapter 5	52
Conclusion	52
5.1 Introduction	52
5.2 Future Enhancements	52
References	54
Appendix A	56
A.1 Web Application Screenshots	56
A.1.1 Login	56
A.1.2 Reset Password	56
A.1.3 Buyer Registration	58
A.1.4 Waste Category	59
A.1.5 Buyer Emailing	59
A.1.6 Buyer Purchase	60
A.1.7 Track Transporter	60

A.1.8 Rating Transporter	61
A.1.9 Client request Report	61
A.1.10 Buyer Purchase Report	62
A.2 Mobile Application Screenshots	63
A.2.1 Login	63
A.2.2 Client Menu	64
A.2.3 New Request - Schedule	65
A.2.4 My Request	67
A.2.5 Rating	68
A.2.6 Request Cancel	69
A.2.7 Transporter Menu	70
A.2.8 Job Request	71
A.2.9 Job Cancel	72
A.2.10 FAQ	73
A.2.11 About Us	74
Appendix B	75
B.1 Web Application Development Coding List	75
B.2 Mobile Application Development Coding List	87
Appendix C	93
C.1 User Guide – Mobile Application	93
C.2 User Guide – Web Application	111

List of Figures

Figure 1:	Basic Structure of the Application	16
Figure 2:	Paper Prototype-Mobile Application	17
Figure 3:	ER Diagram	20
Figure 4:	Use Case Diagram – Mobile Application	21
Figure 5:	Use Case Diagram – Web Application	22
Figure 6:	Class Diagram	27
Figure 7:	Client Registration	28
Figure 8:	Transporter Registration	28
Figure 9:	Send New Request	29
Figure 10:	Cancel Client request	29
Figure 11:	V model	30
Figure 12:	Login page - web application	34
Figure 13:	Dashboard - web application	35
Figure 14:	Transporter Registration Page	35
Figure 15:	Transporter waste category Page	36
Figure 16:	Recycle details page	36
Figure 17:	Welcome Screen and Login page	37
Figure 18:	Client Registration page	38
Figure 19:	Client request -Single page	39
Figure 20:	Test model	41
Figure 21:	Types of Software testing	41
Figure A.1	Login	56
Figure A.2.a	Reset password	56

Figure A.2.b	Reset password	57
Figure A.2.c	Reset password	57
Figure A.3	Buyer registration	58
Figure A.4	Waste Categories	59
Figure A.5	Buyer emailing	59
Figure A.6	Buyer purchase	60
Figure A.7	Track transporter	60
Figure A.8	Rating transporter	61
Figure A.9	Client request report	61
Figure A.10	Buyer purchase report	62
Figure A.11.a	Login page	63
Figure A.11.b	Login page with validation	63
Figure A.12.	Client Menu	64
Figure A.13.a	New Schedule request	65
Figure A.13.b	Pickup location	65
Figure A.13.c	Pickup Invalid location	66
Figure A.14.a	My request	67
Figure A.14.b	My request Detail	67
Figure A.15	Rating	68
Figure A.16	Request cancels by client	69
Figure A.17	Transporter Menu	70
Figure A.18.a	Job request	71
Figure A.18.b	Job request Detail	71
Figure A.19	Job cancel by transporter	72

Figure A.20	FAQ page	73
Figure A.21	About Us	74
Figure B.1	Login Controller.cs	79
Figure B.2	WasteManagementContext.cs	80
Figure B.3	Login Entity. cs	81
Figure B.4	Login Html	83
Figure B.5	Login Component.ts	84
Figure B.6	Login Service.ts	86
Figure B.7	Login Model.ts	86
Figure B.8	LoginScreen.java	89
Figure B.9	LoginScreen.xml	92

List of Tables

Table 1:	Use case 01- Transporter Registration	23
Table 2:	Use case 02- Client Registration	24
Table 3:	Use case 03- Client request to collect waste	24
Table 4:	Use case 04- Cancel Client request to collect waste	25
Table 5:	Use case 05- Functions of the Administrator	25
Table 6:	Use case 06- View Client requests	26
Table 7:	Test Plan	32
Table 8:	Test Case - Login Functions-Mobile and Web Application	44
Table 9:	Test Case - User Registration-Mobile and Web Application	45
Table 10:	Test Case - Client request	46
Table 11:	Integration Test cases	47
Table 12:	Questionnaire	49
Tables 13:	Test Deliverables	51

List of Abbreviations

CSS	Cascading Style Sheet
ER Diagram	Entity Relationship diagram
GPS	Global Positioning system
HTML	HyperText Markup Language
IIS	Internet Information Service
IoT	Internet of Things
MVC	Model–View–Controller
UAT	User Acceptance Test
UCSC	University of Colombo School of computing
UML	Unified Modeling Language
WMS	Waste Monitoring System

Chapter 1

Introduction to Waste Monitoring System

1.1. Introduction

Waste collection and monitoring are a complex process and nowadays it becomes very cost without a systematic process. The long-established method of monitoring and controlling the wastes is not systematic and utilizes more human effort, time, and cost. It is not suitable for present-day technologies. The Improper management of waste typically domestic waste, industrial waste, and environmental waste is a fundamental cause for many human problems such as pollution, diseases, and has adverse effects on the hygiene of living beings. To overcome all these issues, we are proposing the idea of a smart waste monitoring system that helps manage waste with less human effort to maintain a clean environment.

The opinion of smart waste monitoring is implemented as a systematic process for the collection and monitor the waste collection process in a metropolis where waste production is domestically very high but the effort for control it is relatively very low. Smart waste management mainly avoids the overfull collection of waste generated domestically which creates difficulty to manage its disposal.

1.2. Problem Specification

Sri Lanka produces 7000MT of waste per day in the Western Province. Each individual generates an average of 1-0.4kg of waste per day. According to the Waste Management Authority and the Central Environmental Authority, only half of the waste generated is collected [6].

Waste collection and disposal responsibilities are distributed with the local authorities of the Divisional Secretariat, either a municipal council (as per the Municipal Councils Ordinance - 1947), the urban council (Urban Councils Ordinance – 1939), or local council (Pradeshiya Sabha Act – 1987). Provisions related to waste management and disposal, are made under the National Environmental Act No.47 of 1981 and Public Nuisance Ordinance.

There are some organizations concerned with waste management at different stages, including the Ministry of Local Government and Provincial Councils, Ministry of Mahaweli Development and Environment, Ministry of Megapolis and Western Province Development, Central Environmental Authority, Urban Development Authority, National Solid Waste

Management Support Centre, Western Province Waste Management Authority, Local Authorities.

In the last 20 years, government organizations have attempted to implement the best waste monitoring approach for the country. Some policies and actions supported sanitary landfills, some initiatives were driven towards waste to energy projects. In 2008, Central Environmental Authority commences a 10-year Waste Management Program called “Pilisaru Programme” for “Waste Free Sri Lanka by 2018”.

As a result of this process finally we are created Karadiyana, Bluemendhal, Meethotamulla, Kolonnawa, and coastline, rivers, and other streams which become dumping sites for plastic and polythene waste, and other waste.

1.3 Motivation for the project

As a result of the non-systematic approach the waste collection, we can see waste on the roads, side of the roads, in the water bodies in most cities. The various governments have implemented several projects to bring a solution to this problem but no government has able to bring an everlasting solution for the waste management problem in Sri Lanka.

1.4 Objectives of the project

- The prime objective of the Smart Waste Monitoring System, “To establish a proper solution for waste collection and monitoring to enhance the quality of the environment ”
- Enhance effectiveness waste collection operation
- Improve the waste disposal process
- Introduce the waste collection classification
- Advance support to monitor the waste collection operation
- Eliminating waste on the environment to keep up improving the better condition of life”
- Supervising the waste collection operation to reducing waste in the environment
- On the elementary condition to transform a smart technology for waste collection system
- Make a minimum effort and time of people to eliminate waste in the environment
- Proceed in a healthy and clean environment.

1.4 Scope of the project

Smart Waste Monitoring System is a web-based solution with Smart-Bin mobile application to manage waste in geographical areas like city areas. It can be used for organizations in the government sector or the private sector.

According to the discussion with the Municipal council, Gampaha following components and generic requirements are identified as the scope of the project.

The system will be addressed four location categories to collect waste. Following are the location categories system will be maintained.

- Commercial (Market waste)
- Domestic
- Industrial (schools, public offices, hotels, etc.)
- Hospital

There are main target waste categories manage by the proposed system.

- Glass
- Plastic
- Paper & Board
- Food & Agricultural waste
- Metal
- Hazardous waste
- E-waste
- Construction Debris
- Garden waste (Tree cuttings and grass cutting wastes)

1.4.1 Functions within the system

- Automate the waste collection process.
- Automate recycle selling process.
- Mobile interface for the clients to send requests for the collected waste.
- Mobile interface for the transporters to view and update the status of the job.
- Web interface for the administrator of the organization to manage the waste collection process.
- Supports for the forecasting, decision making through report generation.

1.4.2 Mobile Application

There are two logins of the “Smart Bin” mobile application which are login for the User and login for the garbage tracker. Users can register by entering a username and password. Garbage tracker can register by entering the username, password, and vehicle number. The garbage tracker’s registration will be validated with the web application from the administrator.

Client:

- Client registration
- Post a request to collect waste as a single request or a schedule request.
- Clients can post an image of the waste if need in a request. (This feature will be supported only for a single request)
- Additional Feature: Users can select a waste pickup location from google Maps when send a request.
- The client can cancel the requests.

Transporter (Garbage Truck Driver):

- The transporter can start the job and when he accepts the client request system sends the notifications for the relevant locations.
- The transporter can view client requests.
- The transporter can update the request with status (Complete/Partially Completed) when he has done his duty.
- The transporter can cancel the request (vehicle breakdown/road breakdowns/natural disasters) to the system via mobile application.

1.4.3 Web Application

Administrator:

- Transporter (Garbage Truck Driver) registration.
- Waste classification
- Track transporter

The system will be provided a separate screen for the track transporter including updated details of transporter job status. The administrator can track the transporter by using this screen.

- Dashboard

The dashboard will be displayed in the following items.

Today Total Client requests

Today Pending Requests

Today In Progress Requests

Today Partial Completed Requests

Today Completed Requests

Today Cancelled Requests

- Compose emails to buyers of the recycled items
- Enter current stock of the recycled items.
- View transporter ratings

Assumptions of the Smart Waste Monitoring System

- There should be an android device for each waste truck driver.
- There should be an android device for the client.
- There should be an internet connection

1.5 Overview of the Report

Chapter 01 includes a brief introduction to the project and objectives and scope. Problems encounter and being faced in the current system and how to overcome those problems and features of the proposed system will be included in this chapter. Chapter 02 discusses similar systems, related work, or background study of the proposed system. Chapter 03 explains the methodological approach used in the system for designing. Chapter 04 discuss the implementation details of the system. Finally, Chapter 05 discusses the achievements and future enhancements of the system.

Chapter 2

Background

2.1 Introduction

This chapter demonstrates an overview of the current system and drawbacks of the current system and describes the functional and non-functional requirements of the proposed system. Also, a review of existing systems and research papers are published related to waste management all over the world.

Germany has come to first place all over the world for having the best waste management and recycling system in the world. Austria is in second place and Belgium is in third place in recycling and managing the waste. In Europe, these three countries are the best waste management and recycling performers [1].

Germany also implemented many waste management systems and policies. The entrepreneurs in Germany have identified waste management and recycling as a source of business in an innovative way. As a result of their advanced waste management policies, the 50,000 garbage dumps have been transformed into [1]:

- 70 incinerators
- 60 biological and mechanical waste processing factories
- 800 units producing compost from organic waste

The “green dot system “is one of the most innovative recycling solutions that Germany has promoted. Manufacturers and retailers have to pay for a green dot on the packaging of their products. The more packaging, the higher the fee creating an incentive for businesses to reduce packaging and facilitate recycling. This system has led to less paper, thinner glass, and less metal been used therefore reducing the amount of waste produced [1].

Austria has successfully implemented traditional methods to separate recyclable waste and reduce landfill using taxes and incentives. Austrian Biotech Company has developed a new high-tech method of waste management which uses fungal enzymes to recycle PET [1].

Belgium is another country with the best waste management and recycling solution. Belgium has introduced two waste management techniques: The Ecolizer and the green event and

assessment guide. The Ecolizer is tracked the waste problem at the source. The green event and assessment guide are digital tools used by Belgium to gain fight against waste generation [1].

The purpose of this system is to design an effective waste management system for Sri Lanka based on the current waste collection process.

2.2 Background of the Business

In Sri Lanka currently, most of the government and private organizations of waste management are followed a manual process for the collected waste from the divisions and recycling process. Gampaha municipal councils also followed a manual process for the waste collection. They have divided the area into divisions and assign transporters and supervisor for every division. The Supervisor has monitored the waste collection process of his area and take decisions appropriately. The waste recycling process also doing as a manual process, they have some buyers to purchase waste like glass, paper: metal, and cardboard.

2.3 Overview of Current Process and Drawbacks

2.3.1 Analyzing the current system

Currently, Gampaha municipal council has managed this waste collection process manually. They have garbage trucks and allocated transporters to divisions. Currently, they have waste categories as followed and they schedule days into categories to collect waste. Glass, plastic, and polyethylene collect twice a week.it is the schedule as for every Wednesday twice a week. It has schedules one day per week to collect metal, paper, and cardboard.

- Glass
- Plastic
- Cardboard & Paper
- Food waste
- Metal
- Hazardous waste & Construction Debris

Trucks are not allocated waste category wise. Currently, they have allocated one truck to collect food waste. Another truck is allocated to collect glass, plastic, and polyethylene. Another truck is allocated to metal, paper, and cardboard. After finish collects waste domestically that they

will collect waste in hospitals and collect waste in construction sites and factories if there any special request.

There is a separate compost unit and after collect food waste, they transfer it into that unit. Other categories such as glass. Plastic, polyethylene, metal, paper, and cardboard they are selling to internal and external parties. External parties are some buyers and internal parties are garbage transporters.

2.3.2 Drawbacks of the current system

Following drawbacks are identified after analyzed the current system.

- As per the current system unable to collect waste on time and people need to wait until the day is coming to collect waste.
- Waste is kept on the roads for a very long time.
- High Environmental pollution
- Re-use ability of the waste reduced because they are kept a very long time on the roads and out of the houses.

2.4 Requirements of the proposed project

The proposed system is addressed solution for the drawbacks of the current system is above and the following functional requirements are described system behaviors of the proposed solution. The functional and non-functional requirements of the proposed system are as followed.

2.4.1 Functional Requirements of the project

Mobile Application: Client

- Client registration
The clients should be registered for the system.
- Post request to collect waste as a single request or a schedule request.
The client can post requests to collect waste as a single request or schedule request. They can schedule for a week to collect waste. In both requests' client can add multiple waste categories.
- The client can attach an image of the waste if need in a request.

This feature will be supported only for a single request

- The client can select a waste pickup location from google Maps when send a request.
- The client can cancel the requests.

Mobile Application: Transporter (Garbage Truck Driver)

- The transporter can start the job and when he accepts the client request system display the notifications for the relevant locations.
- The transporter can view client requests.
- The transporter can update the request with status (Complete/Partially Completed) when he has done his duty.
- The transporter can cancel the request (vehicle breakdown/road breakdowns/natural disasters) to the system via mobile application.
- Notifications

Notifications will be displayed transporter's end when clients send requests to collect waste.

Web Application: Administrator

- Transporter registration.

This is for the registered transporters to the system.

- Waste categorization
- Track the transporter

The system will be provided a separate screen for the track transporter including updated details of transporter job status. The administrator can track the transporter by using this screen.

- Dashboard

The dashboard will be displayed in the following items.

Transporter progress.

Transporter request canceled list

Recycle quantity details

Status widgets (Pending/ In Progress/ Partially Completed/ Completed/ Canceled)

Today Total Client requests

Today Pending Requests

Today in Progress Requests

Today Partial Completed Requests

Today Completed Requests

Today Cancelled Requests

- Compose emails to buyers of the recycled items
- Enter the current storage of recycled items.
- View transporter ratings
- Reports generation

The system should support the analysis of the information and support for making decisions.

2.4.2 Non-Functional Requirements of the Project

Non-functional requirements define as system attributes. They are affected by how the system should work. The non-functional requirements of the proposed system as followed.

- It should be user-friendly and should provide easy interfaces.
- Accurate and consistent.
- Flexible and portable
- It should be maintained the necessary security measures.
- Reusable and maintainable.

2.5 Review of similar Waste Management Systems

Waste Management systems are getting popular in the world these days. In the term waste management, there can be many waste management systems such as waste management systems and waste monitoring systems. This Project is mostly aimed at waste management because it is critical to managing a large amount of waste in the cities and it will be easy if there is a systematic process to manage the waste. There are many waste management systems all over the world. Followings are some of them.

2.5.1 Related Works

1) Abans City Clean – Mobile Application

Abans City clean is one of the innovative mobile solutions to protect the environment and keep the city clean Sri-Lanka. By using this mobile application people can report any incident and immediate response is given to supervisors.

2) CleanCityNetworks, the waste analytics platform by ECUBElabs.

CleanCityNetworks, the waste analytics platform by ECUBElabs- CleanCityNetworks, or CCN, is the leading waste management platform and the glue that binds all our solutions together. CCN sends and receives real-time data from sensor devices on the ground. The network transforms the data into actionable information usable by the waste collection manager [7].

3) Solid Waste Collection and Monitoring System

Pune Municipal Corporation is leading for the solid waste management system with complete end-to-end waste collection and management. They are using cutting edge technologies like GPS, RFID, GSM, IOT sensors with innovative mobile and web application to improve and smoothen mechanism for waste collection and management process [8].

4) Sensors Smart City Waste Management System

Smart bin sensors provide support for organizations and municipal corporations to manage the clearance of waste bins timely. Sensor-based waste collection bins are used to identify the status of the waste bins. The real-time waste management system can use smart dustbins to check the status of the dustbins and information of all smart dustbins can be accessed. The main objective of this system is cities with a decent quality of life for citizens and a clean, sustainable environment [9].

5) Cognito Smart Solid Waste Management System

The Municipal corporations or any other organizations that are handling waste management are adopted to different systems for handling and managing their waste based on the type, its geography or vehicles being employed by them, and the availability of landfill sites. Cognito's waste management system aims to address the problems faced by the municipal councils or any other organizations in their entire solid waste management process being adopted by them. The system uses the various technologies that have evolved to provide data from the various processes and aim to improve the efficiency of those processes. It utilizes tools like Cloud Computing, IoT Sensors, GSM/ GPRS, RFID, and other low power wireless technologies to manage the system efficiently [10].

2.5.2 Research Articles

- Smart Waste Management System - Collage: MANGALORE INSTITUTE OF TECHNOLOGY AND ENGINEERING, MANGALURU BRANCH: DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
- Smart Waste Management System-Bangalore Institute of Technology & Engineering- Project Reference Number 39s_BIN_0373.
- Smart Waste Management System-KIs Gogte Institute of Technology, Udyambag- Project Reference Number 39s_BIN_0321
- Public Participation in Solid Waste Management: Challenges and Prospects (MSc. Development Management Thesis) By Mukisa Philemon Kirunda
- Design an Effective Solid Waste Management Model for Sri Lankan Context.

Methodology

3.1. Introduction

This chapter demonstrates the blueprint for the design stage of the proposed system and describes design methodologies and techniques which have been used to develop the system.

The methodology is aimed at requirement gathering, fact-finding, and identifying the functional and non-functional requirements of the system. This chapter includes the design of the system from many contexts using the UML diagrams such as class, use case, and sequence diagrams. This chapter gives an overview of the implementation details of the system such as implementation environment, hardware, and software requirements, development methodologies, tools, and technologies are used for developing the application, and software architecture of the proposed system are described in this section.

3.2. Requirement Analysis

Requirement analysis is a vital component of any software development project and also it is a most challenging piece of work. Requirement analysis is an essential part of any software project and the foundation on which all projects should be built. These are some techniques used by the WMS for requirement gathering.

3.3. Fact-Finding Techniques

3.3.1 Observe similar systems

Observe and review the available waste management solutions around the world for this project and cross reviewed functions of existing applications and reading some research articles on waste management.

3.3.2 Interviews

Interviews are the preliminary approach to gathering requirements any software project has followed. This can be doing face-to-face interaction with relevant stakeholders. Formally in software projects business analysts interview system users and system owners during the early

stages of the project life cycle to gather and get clear the requirements. The interview is one of the best sources to find out the facts for the real requirements.

- Unstructured Interviews

These involve a conversation by the interviewee asking general questions.

- Structured Interviews

The interviewer will be the one making specific questions to obtain the required information from the interviewee. This type of interview is efficient.

3.3.3 Review of available documents and manuals

Review the available documentation, recruitment analysis documents regarding waste management around the world.

3.4 Requirements for the new system

3.4.1. Functional Requirements

- Enable signup for transporter and client.
 - transporter's profile
 - client's profile
- Build post a client request functionality to collect waste.
- Enable cancel client request functionality.
- Allow Transporter ratings
- Start the job and view client requests
- Enable cancel accepted client requests from the transporter end.
- Accommodate for the administrator in the organization to handle day to day duties.
- Provide support to manage the waste of the organization.
- Supports for the sale recycle for buyers.
- Enable email functionality for buyers when the order level has reached.
- Provide support for decision making through reports.

3.4.2. Non-Functional Requirements

- **Performance**

Response time is an important part of any website's user experience.

- **Reliability**

Information that contains within the system needs to be accurate and consistent. Acknowledgments or emails should be real-time to overcome any gaps in the system downtime and all errors in the system or process need to be well advised dynamically to the user.

- **Usability**

It should be easy for the user, and easy to learn, operate, and prepare inputs and outputs through interaction with a system. Transporter and Clients should be quickly needed to get the hang of the system flow.

- **Availability**

Web site and mobile application should be available whenever the user needs.

- **Security and safety**

The system should be secure enough to handled transporter's and client's profiles and their personal information. As well as have to provide secure access to the admin users because they can control all activities so admin users must have authorized by the company or who is responsible for the web application.

3.5. Design of the system

The design and implementation phase of this system development will concern the design of the suggested program using unified modeling language (UML) plus the interpretation of the design to the desired design specifications and source code.

This phase starts when all requirements from the analysis phase are gathered and subsequently mapped into an application architecture. Whilst the analysis phase focuses on doing the "right" thing, the design phase focuses on doing "things" right. Thus, the design phase determines which programming languages, application architectures, architecture layering, data structures, and many others to use the main objective of the implementation is to generate the source code

and that adheres to the specifications. Simply in this exploration C#, Angular, MSSQL used back, CSS, HTML has been used to design a user-friendly interface.

System design is mainly focused on the design of the system which includes the ER diagrams, Sequence diagram, Use case diagram, and Class diagram. Interface design etc. Designing the front end and the back end includes under this section.

3.6. Software Architecture

Software application architecture is the process of defining a structured solution that meets all of the technical and operational requirements while optimizing common quality attributes such as performance, security, and manageability. It involves a series of decisions based on a wide range of factors, and each of these decisions can have a considerable impact on the quality, performance, maintainability, and overall success of the application.

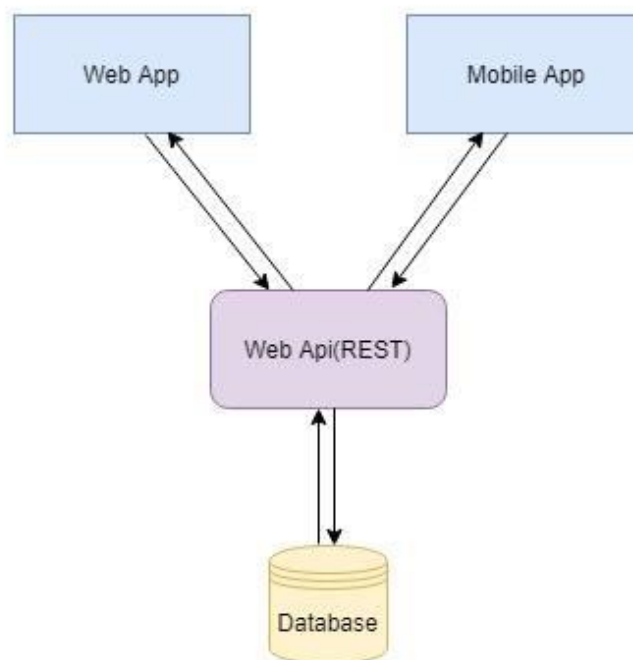


Figure 3.1 Basic Structure of the Application

3.6.1. Proposed Architecture for the Mobile Application

This phase includes user interfaces to access the system. The plan of UI Design is shown in the following figures.

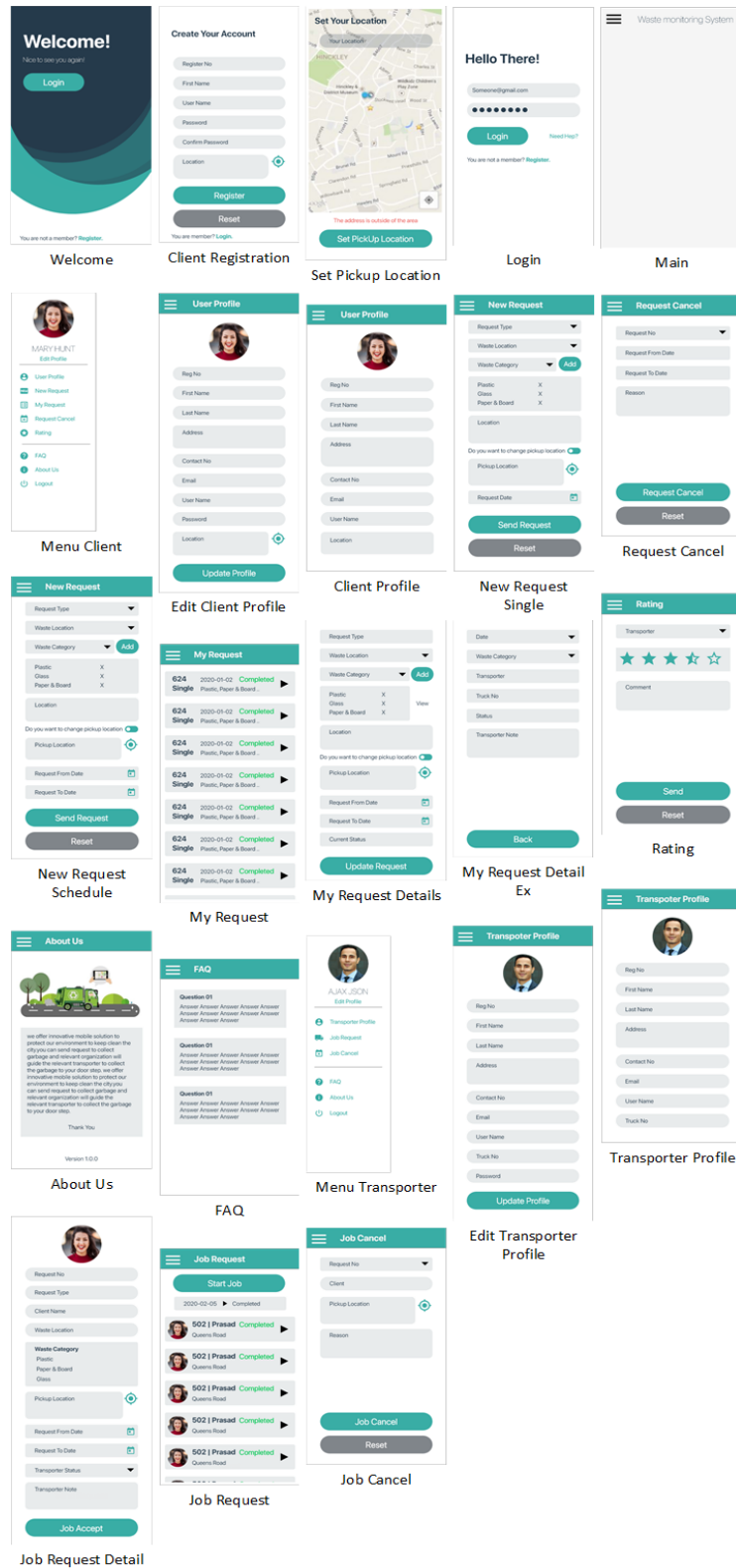


Figure 3.2 Paper Prototype-Mobile Application

3.7. System Software & Hardware Requirement

3.7.1. Web application - Hardware requirements

- Operating System: Window 8 or later (x86 or x64)
- Processor: Intel Core 2 Duo (2GHz) or later
- RAM: 4GB
- HDD: 1GB
- Network: Broadband Internet connection

3.7.2. Web application - Software requirements

- Web Browser: Chrome, Firefox, Microsoft Edge
- Web Server: Internet Information Service (IIS)
- Database server: MSSQL

3.7.3. Mobile application - Hardware requirements

- Operating System: Android devices (up to Android version 5.0 API Level 21)
- Processor: ARM
- RAM: 2GB
- Storage Space: 150MB
- Display size: Up to 4.7 inches
- Resolution: Up to-
- Android devices should be supported by GPS and Mobile Broadband connection

3.8. Database Design

Good design is very important in developing a good system. To convert the analyzed requirements into code, designing should be done properly. Unified Modeling Language (UML), which has become a standard modeling language for object-oriented modeling. Programmers can easily understand models of objects in UML and the programmers can easily write software. Few structural and behavioral UML diagrams use for design proposed systems are mentioned below

- Use-Case diagrams – shows what the system needs to do.

- Class diagrams – shows the needed objects and relationships between them.
- Sequence diagrams – shows how the objects interact over time.

Database design illustrates the table structure of the database, the relationships among tables, and how each entity joins with other entities of the database. This information has been depicted using ER Diagram.

3.8.1. ER Diagram

An Entity-Relationship (ER) Diagram is a type of UML diagram that illustrates how “entities” such as people, objects, or concepts relate to each other within a system. The Entity Relational Model is a high-level conceptual data model diagram. The Entity-Relation model is based on the real-world entities and the relationship between them. ER model helps you to analyze data requirements systematically to produce a well-designed database. So, it is considered a best practice to create an ER model before implementing a database.

ER diagrams help to explain the logical structure of databases. ER diagrams are looks very similar to the flowchart. However, ER Diagram includes many symbols to illustrate the objects and relationships among objects within the system.

The current version of the ER diagram shows the number of main attributes that are grouped with elements.

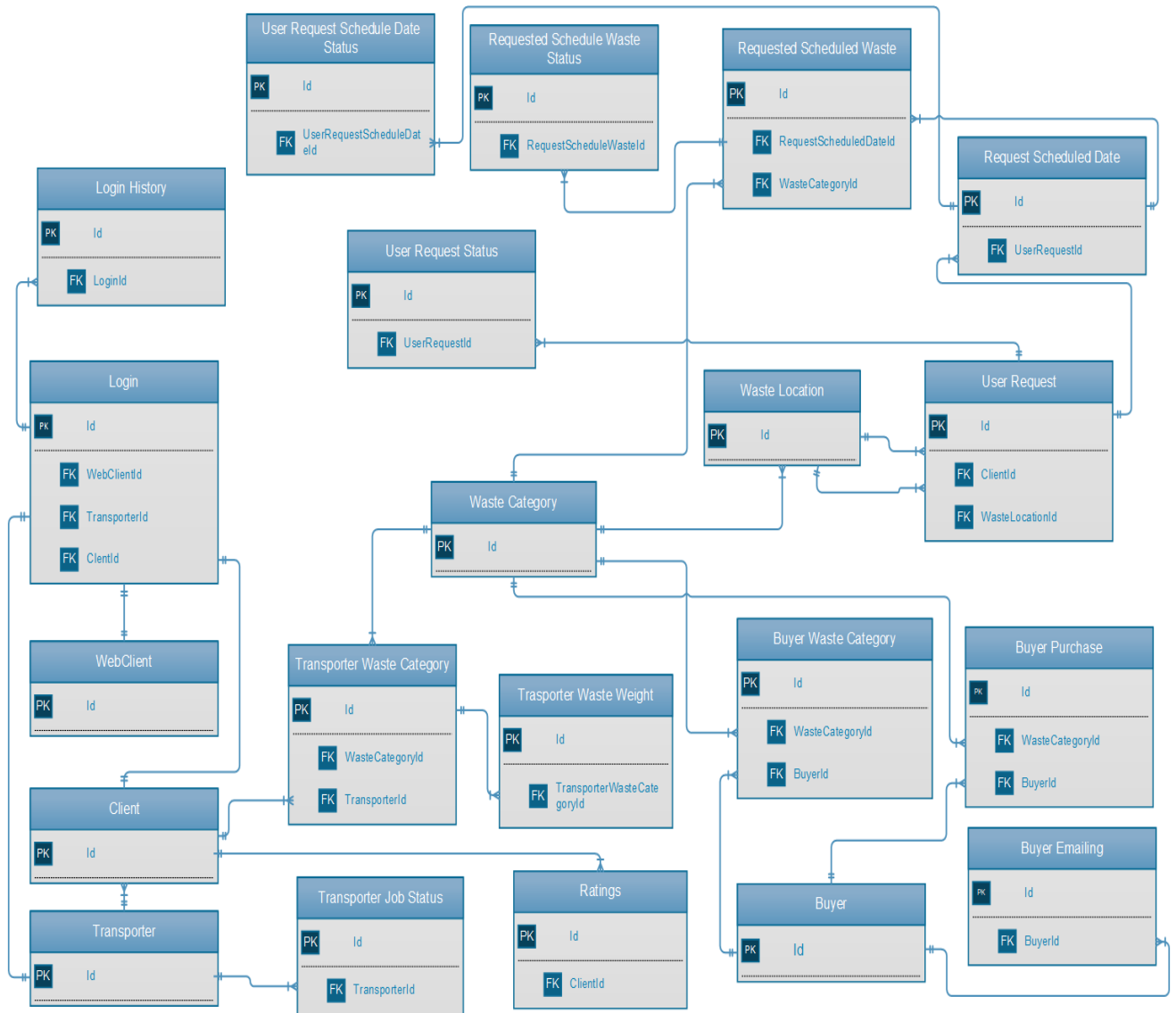


Figure 3.3 ER Diagram

3.8.2. Use case Diagram

A use case is another UML diagram used in system analysis. Use Case diagram illustrates the user's interaction with the system. The system collaborates with five active actors in one cooperate system. Following actors are available in the system. Transporter and Client are actors of the mobile application. The administrator is the actor of the web application.

- Administrator
- Transporter
- Client

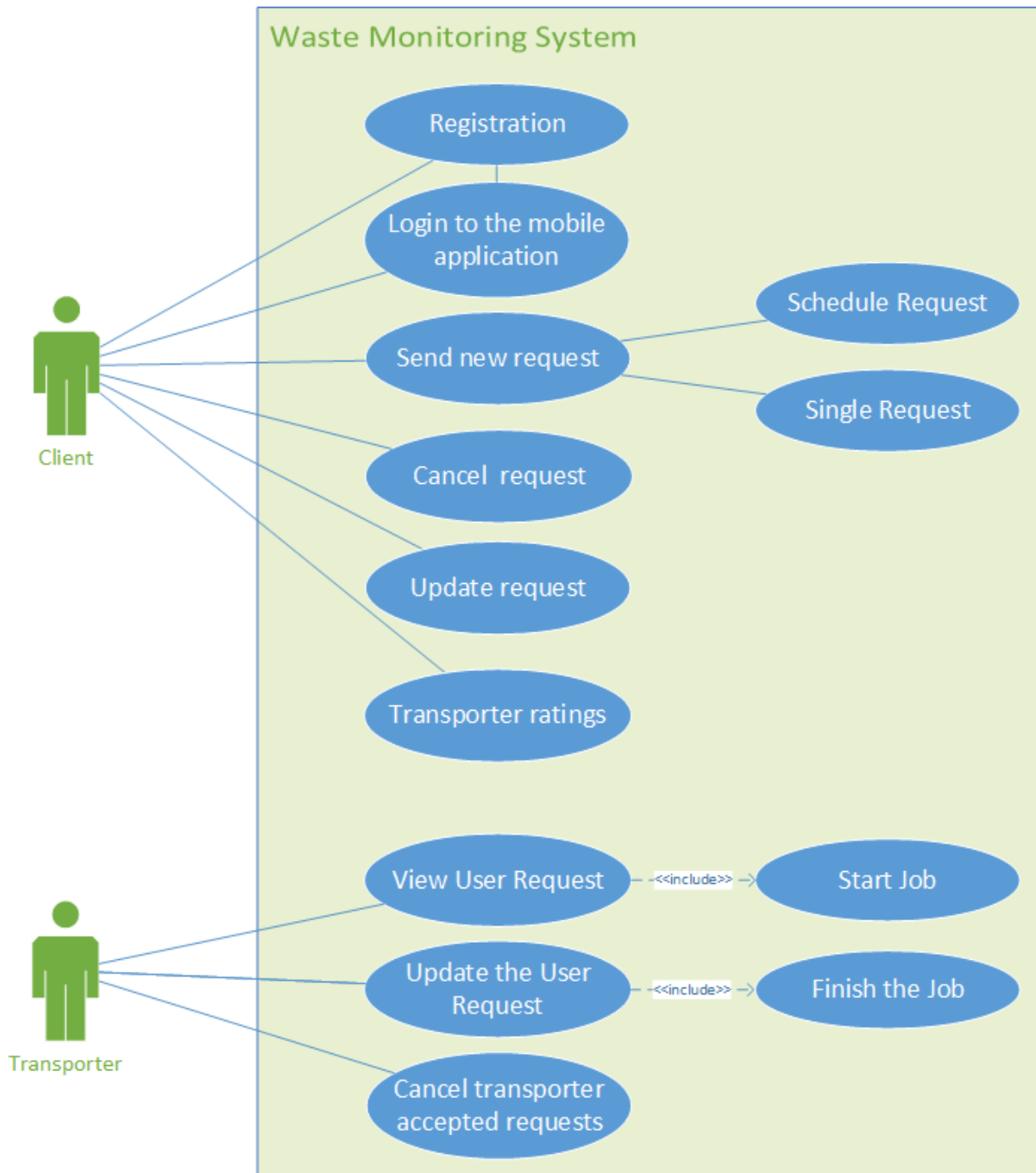


Figure 3.4 Use Case Diagram – Mobile Application

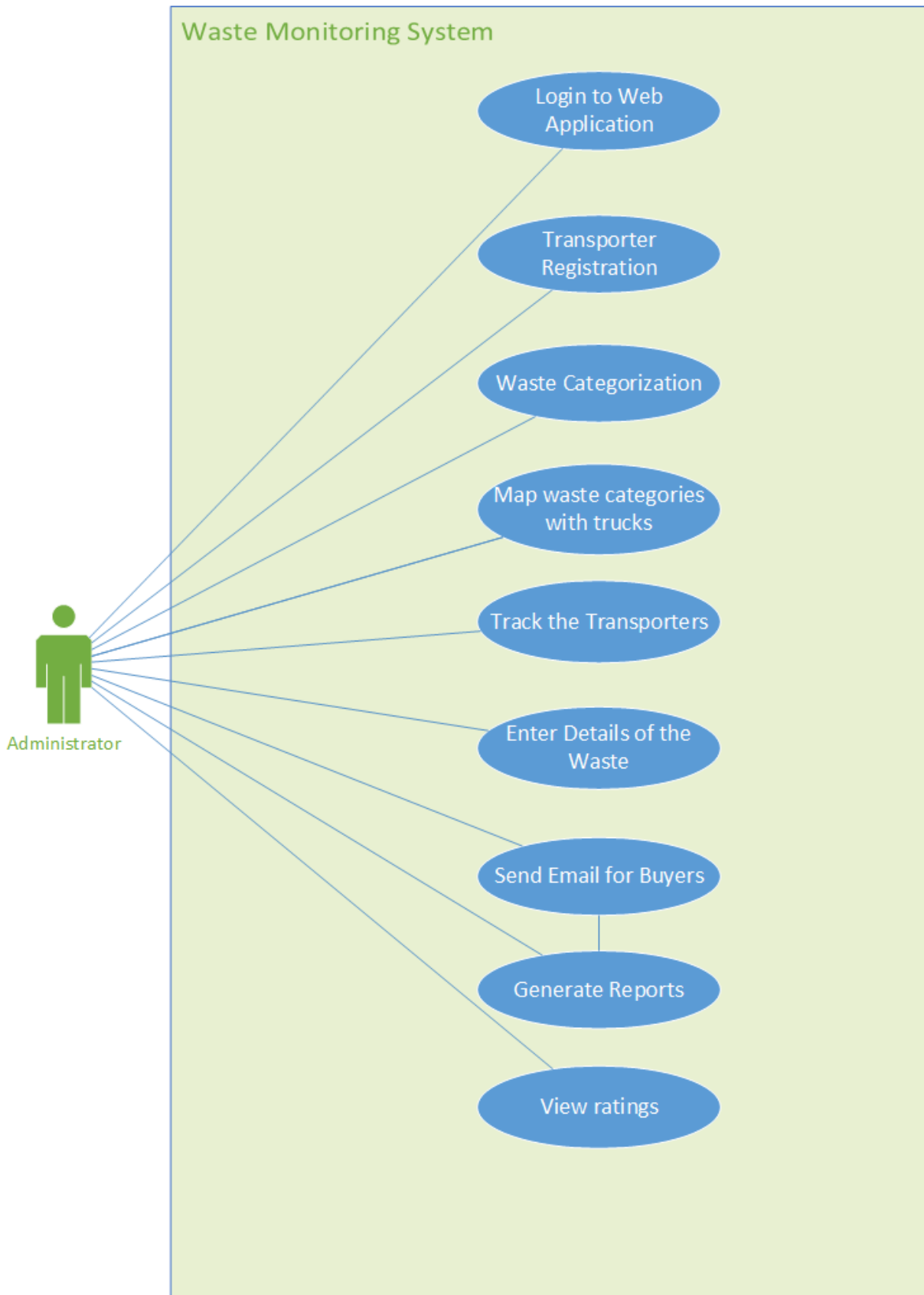


Figure 3.5 Use Case Diagram – Web Application

3.8.3. Descriptive Use Cases

Use Case	Transporter Registration
Primary Actors	Administrator
Preconditions	
<ol style="list-style-type: none"> 1. Required to have an email address. 2. Internet Connection 	
Postconditions	
Once a user registered user belongs to one category transporter he can access it via mobile application.	
Main Scenario	
<p>Transporters must provide initial details to register with the system.</p> <ol style="list-style-type: none"> 1. Transporters must provide a password and confirm the password. 2. Once the user is registered user will get an email notification to confirm his registration with the system. 	

Table 3.1 Use case 01- Transporter Registration

Use Case	Client Registration
Primary Actors	Client
Preconditions	
<p>Required to have an email address.</p> <p>Internet Connection</p> <p>Smart Phone</p>	
Postconditions	
Once the user registered user belongs to one category client he can access via mobile application.	

Main Scenario
<p>The client must provide initial details to register with the system.</p> <p>The client must provide a password and confirm the password.</p> <p>Once the user is registered user will get an email notification to confirm his registration with the system.</p>

Table 3.2 Use case 02- Client Registration

Use Case	Send new client request (single request/schedule request)
Primary Actors	Client
Preconditions	
<p>The client must register with the system.</p> <p>Internet connection</p>	
Postconditions	
Client request (Single/Schedule) for the waste collection.	
Main Scenario	
<p>The client can send requests with relevant details to collect the waste.</p> <p>Waste categories</p> <p>Pick up location</p> <p>Dates</p>	

Table 3.3 Use case 03- Client request to collect waste

Use Case	Cancel client request (single request/schedule request)
Primary Actors	Client
Preconditions	
The client must register with the system. Internet connection.	
Postconditions	
Cancellation of a client request (Single/Schedule) for the waste collection.	
Main Scenario	
The client can cancel the requests to collect the waste which are not accepted by the transporter end.	

Table 3.4 Use case 04- Cancel Client request to collect waste

Use Case	Functions related to the Administrator
Primary Actors	Administrator
Preconditions	
Login for the system Internet connection	
Postconditions	
Track the transporters Waste categorization Manage to recycle items.	
Main Scenario	
The administrator can manage and monitor all administrative tasks.	

Table 3.5 Use case 05- Functions of the Administrator

Use Case	View Client requests
Primary Actors	Transporter
Preconditions	
Login for the mobile application Internet connection	
Postconditions	
View the client requests and do the job	
Main Scenario	
The transporter can view the client requests by the route and update the client request as per the relevant.	

Table 3.6 Use case 06- View Client requests

3.8.4 Class Diagram

The class diagram also UML diagram used for the system analysis. It represents the static view of an application. It gives an overview of a software system by displaying classes, attributes, operations, and relationships. The following class diagram in Figure 4 depicts the overall class system of the system.

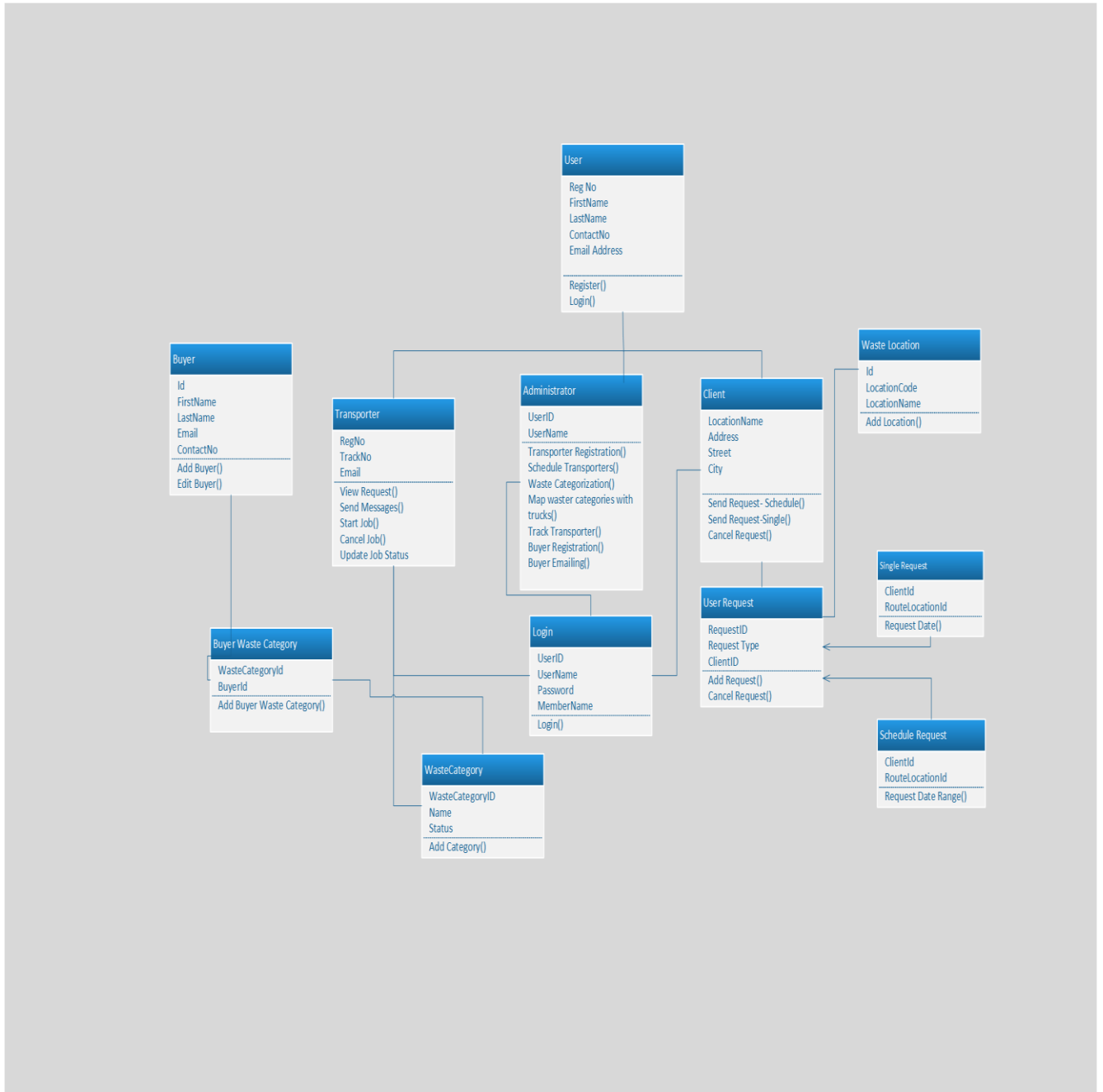


Figure 3.6 Class Diagram

3.8.5 Sequence Diagram

A sequence diagram simply illustrates the interaction between objects in sequential order. It shows object interactions arranged in time sequence.

Client Registration

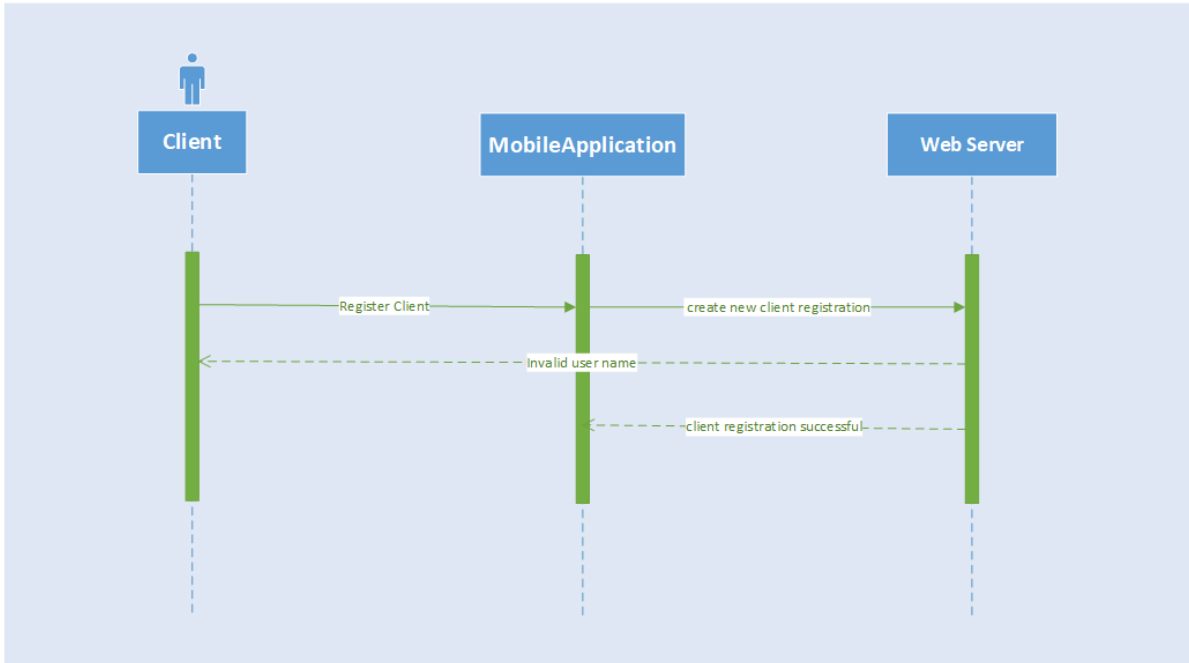


Figure 3.7 Client Registration

Transporter Registration

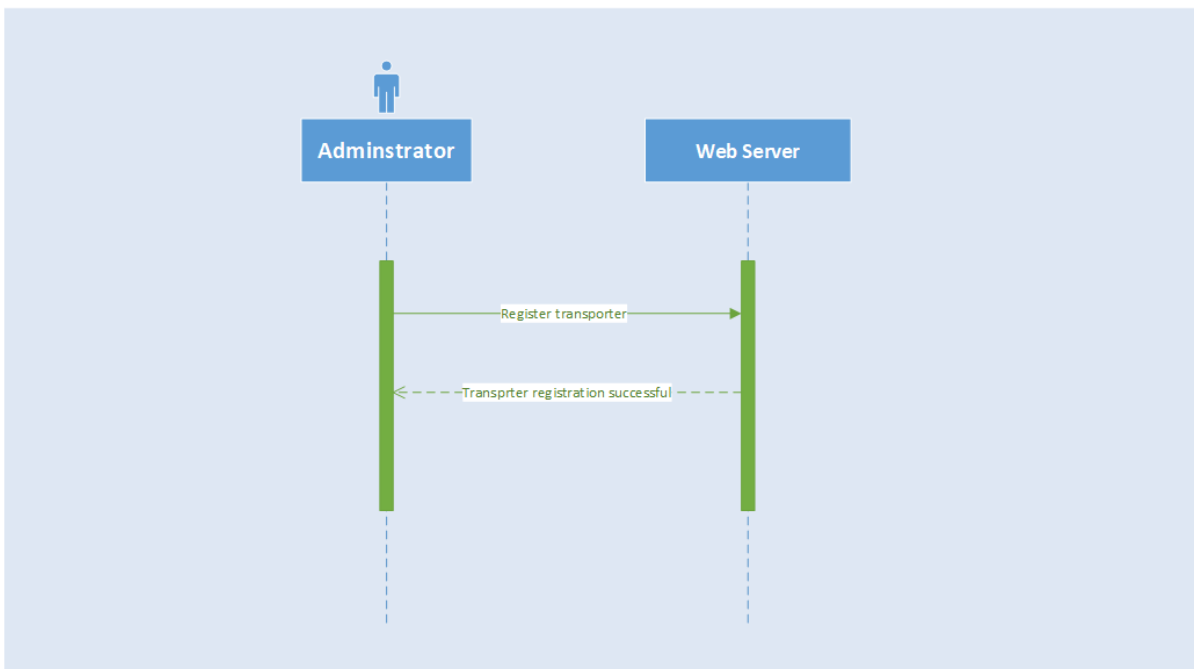


Figure 3.8 Transporter Registration

Send new request for waste collection

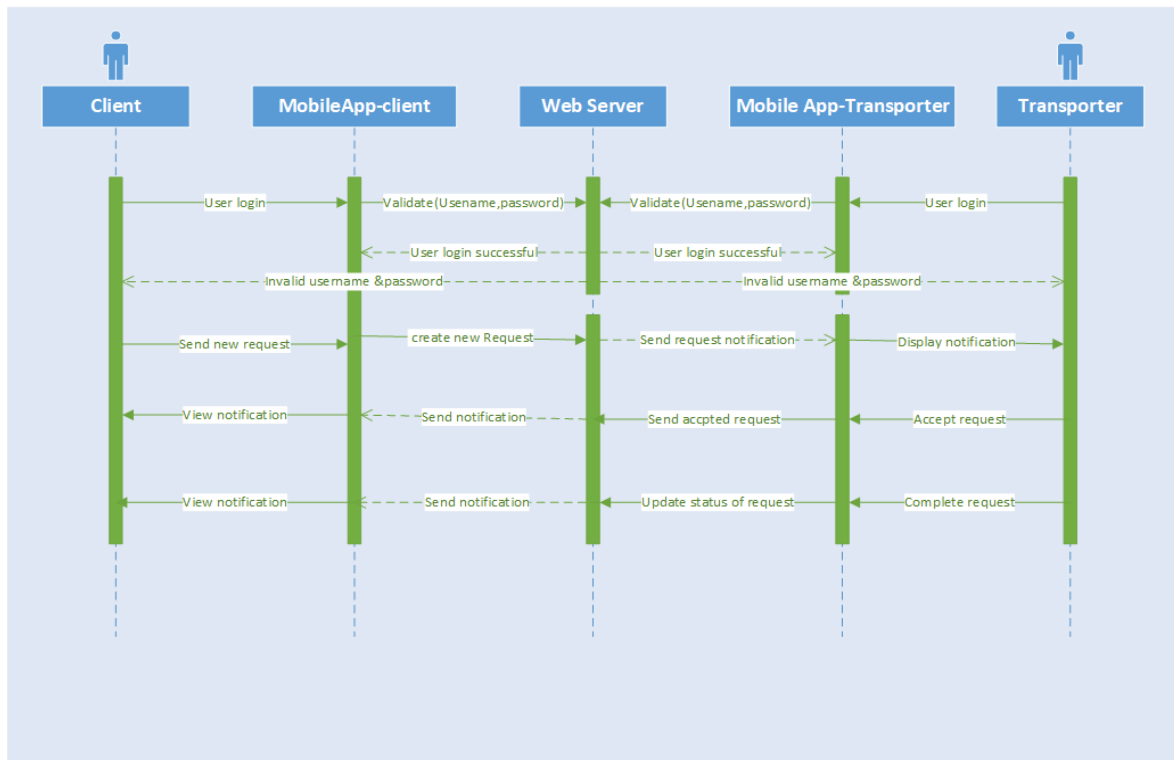


Figure 3.9 Send New Request

Cancel Client request

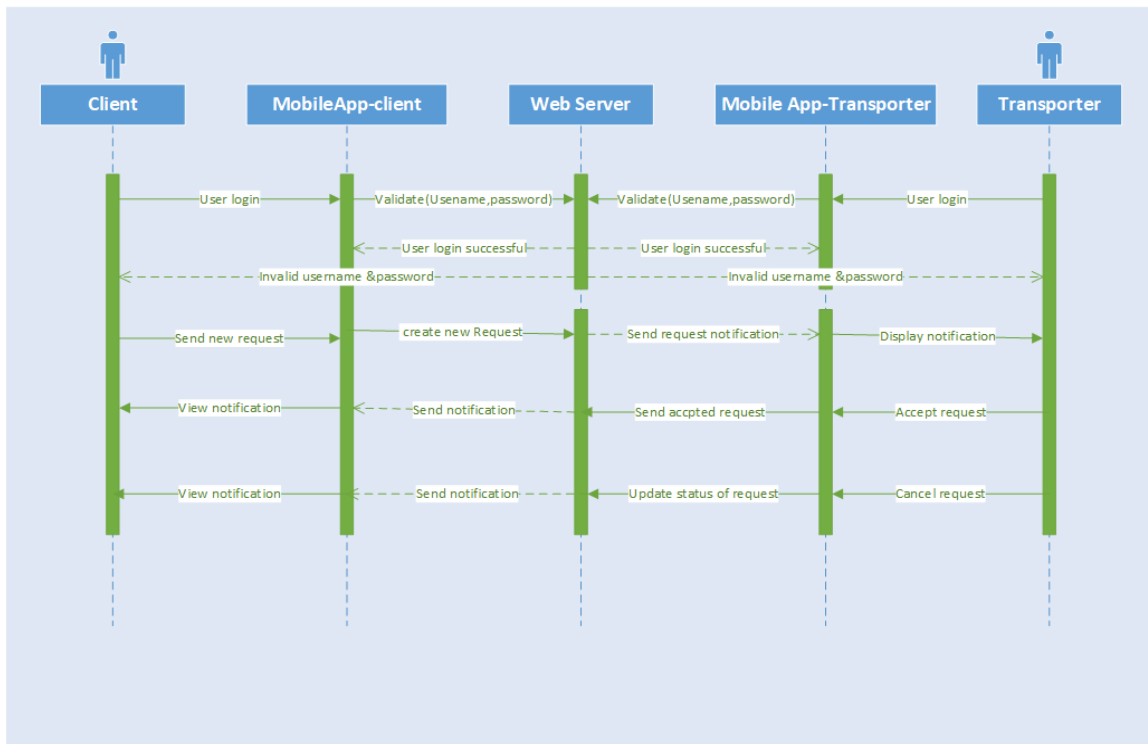


Figure 3.10 Cancel Client request

3.9 Testing

Software testing is the process of executing a program or application with the intent of finding software bugs. Also, it can be defined as the process of validating and verifying that a software program or application meets the business and technical requirements that guided its design and development.

3.9.1 V Model

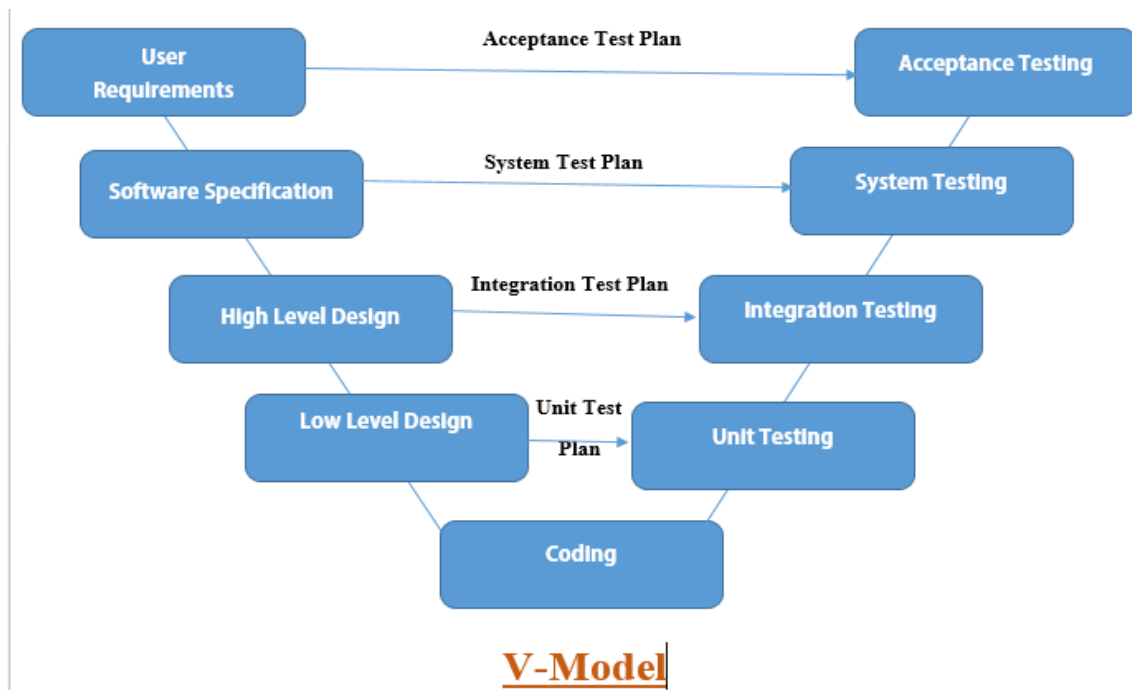


Figure 3.11 V model

3.9.2 Integration Testing

There are separate three modules in this project such as web application and mobile application-client and mobile application-transporter. These three modules are separated and tested together.

3.9.3 System Testing

After successfully finishing the unit and integration testing the system is all together tested. This testing will be done by the developers their selves to check the system if it meets the requirements. As the unit testing and integration testing were successful, the system testing also successful.

3.9.4 Acceptance Testing

Formal testing concerning user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers, or other authorized entity to determine whether or not to accept the system.

3.10 Test Plan of the WMS

A Test Plan is a document describing software testing scope and activities. It is the basis for formally testing any software/product in a project. A test plan is a document describing the scope, approach, resources, and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment, the test design techniques, and entry and exit criteria to be used, and the rationale for their choice, and any risks requiring contingency planning. It is a record of the test planning process.

Introduction The Test Plan has been created to communicate the test approach to team members. It includes the objectives, scope, schedule, risks, and approach. This document will identify what the test deliverables will be and what is deemed in and out of scope.
Objectives The objective is testing the product and ensuring it meets their needs.
Scope Create manual tests for the mobile application and web application. Start the automation test and create an automation test for the selected test scope.
Test Approach Smoke Testing Functional Testing Integration Testing Usability Testing

Automation Testing
User Acceptance Testing
Features to be tested
Functionalities-Web Application
Functionalities-Mobile Application
Features not to be tested
Mobile testing for iOS devices.
Performance of the Mobile and Web application.
Security Aspects of the application
Test Deliverables
Test Plan
Test Cases
Automation Scripts (For the selected test scope)
Test Results
Resources & Environment Needs
Windows 8 and above
Mobile Testing - Huawei GR5 2017
Risks
Approvals

Table 3.7 Test Plan

3.11 Implementation Details

Waste Monitoring System is implemented on three tier architecture. It utilizes Microsoft Visual studio for the backend development, Microsoft SQL Server Management Studio for the database, and angular and Visual Studio Code for the frontend development with CSS and HTML5. Smart Bin mobile application used Android studio for mobile development.

3.11.1 Tools and Technologies

3.11.1.1 HTML5

HTML is the markup language use sets of markup tags in the web pages and that HTML tags support interpreting the webpage over the network. So many new advanced features were introduced in HTML5 such as caching files, drawing objects and graphics, audio/video tags, etc.

3.11.1.2 CSS3

Cascade style sheet is the layout design and formatting language used to enhance web presentation in HTML web pages. CSS3 is the latest version and improved style rules and standards likewise dynamic states of elements, the position of an element, and animations and transitions effects.

3.11.1.3 Angular 9.0

Angular is a platform to support build web and mobile in any environment. Angular is led by the Angular Team at Google and the newest version support both JavaScript and typescript. The angular framework aims to make web development feel effortless, focused on developer productivity, speed, and testability

3.11.1.4 Microsoft Visual Studio 2019

Microsoft Visual Studio is an integrated development environment (IDE) developed by Microsoft. MS Visual Studio support to build an app in various environments such as Windows, Mac, and Linux and including many languages likes Typescript, C#, .NET, JavaScript, XML, HTML, and CSS

3.11.1.5 Microsoft SQL Server Management Studio 2018

SQL Server Management Studio (SSMS) is a software application. It is used to communicate with a database server and appropriate SQL statements for retrieve data, update data, create data, and delete data operations. SQL Server may run either on the same computer or another computer through the network.

3.11.1.6 Android Studio

Android Studio is the official integrated development environment (IDE) for Android application development. It is based on the IntelliJ IDEA, a Java integrated development environment for software, and incorporates its code editing and developer tools. To support application development within the Android operating system, Android Studio uses a Gradle-based build system, emulator, code templates, and GitHub integration. Every project in Android Studio has one or more modalities with source code and resource files. These modalities include Android app modules, Library modules, and Google App Engine modules.

3.11.1.7 Visual Studio Code

Visual Studio Code is a source code editor developed by Microsoft for Windows, Linux, and Mac OS. It includes support for debugging, embedded Git control and GitHub, syntax highlighting, intelligent code completion, snippets, and code refactoring.

3.12 Main Interfaces of the System

This segment illustrates a few main user interfaces of web and mobile applications. The remaining interfaces are under Appendix C.

3.12.1 Login Page of the Web Application

Figure 3.12 authorizes the user to login to the WMS.

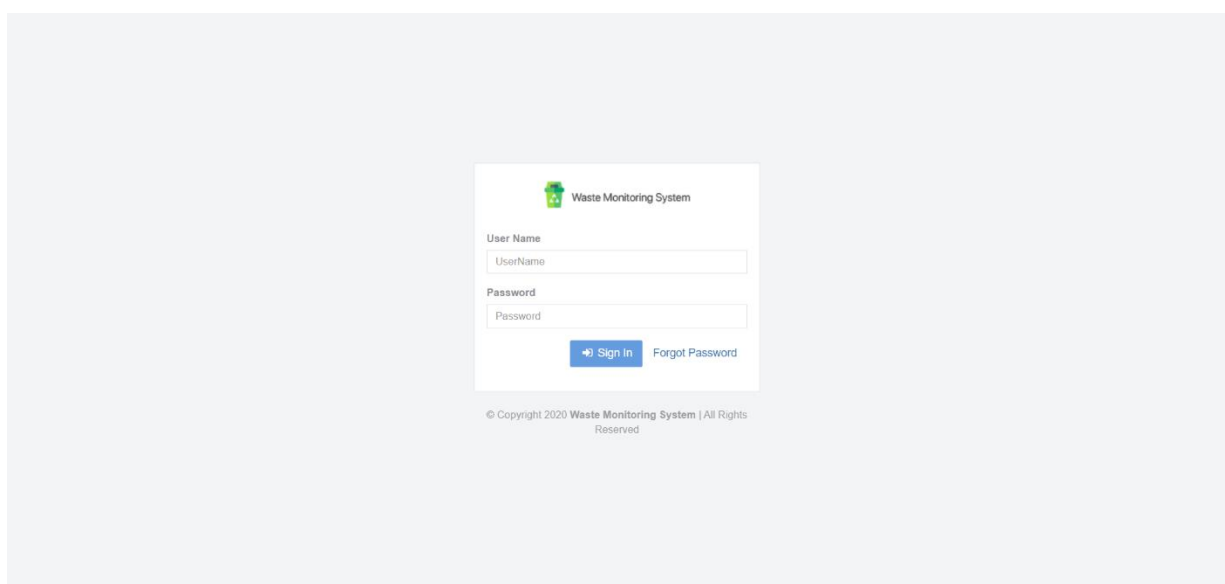


Figure 3.12 Login page of the web application

3.12.2 Dashboard of the Web Application

The dashboard allows displaying summarize information of the system. Figure 3.13 shows the dashboard of the web application.

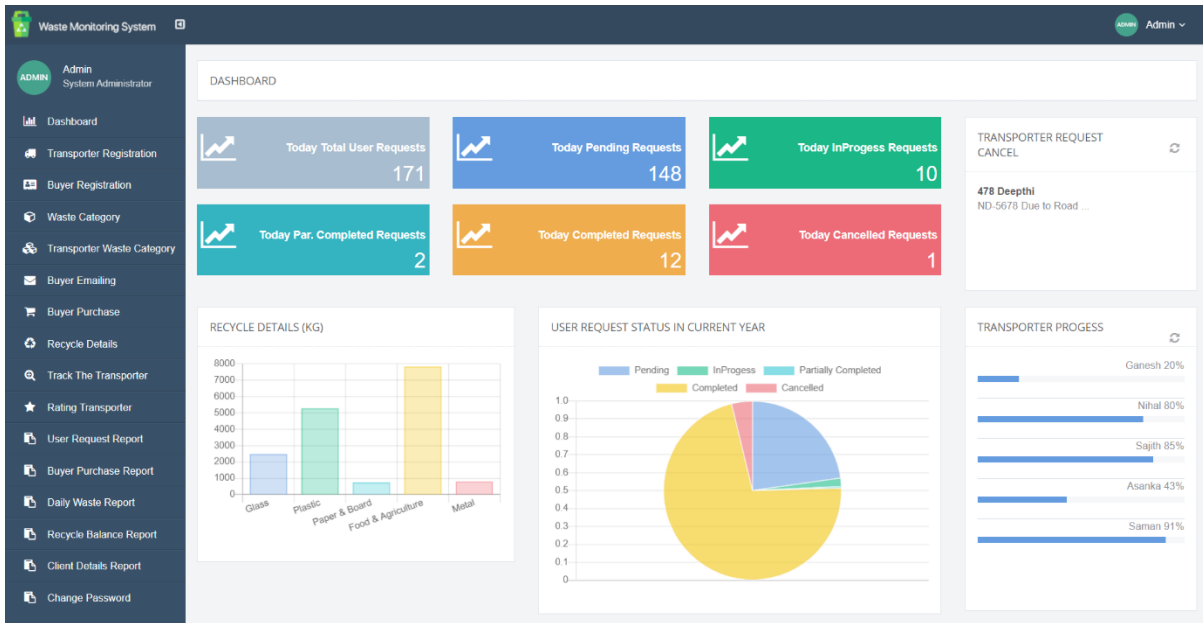


Figure 3.13 Dashboard of the web application

3.12.3 Transporter Registration Page of the Web Application

Figure 3.14 shows the transporter registration page. It allows to administrator to register transporters to the system.

The registration form includes the following fields:

- Reg No: T-005
- First Name: First Name
- Last Name: Last Name
- Contact No: Contact No
- Email: Email
- Truck No: Truck No
- User Name: User Name
- Password: Password
- Confirm Password: Confirm Password
- Active:
- Buttons: Reset, Save

Table of registered transporters:

Edit	Reg No	First Name	Last Name	Contact No	Email	Truck No	Status
✕	T-001	Nihal	Perera	774502560	nihal.perera@gmail.com	ND-5678	true
✕	T-002	Sajith	Madusanka	774502560	sajith.1980@yahoo.com	SP-5485	true
✕	T-003	Asanka	Pathum	785206320	asanka.pathum@gmail.com	WP-4520	true
✕	T-004	Saman	Kumara	771284152	samank@gmail.com	WP-4952	true

Figure 3.14 Transporter Registration Page

3.12.4 Transporter waste category - web application

This screen allows an administrator to map transporters with waste categories in the system. Figure 3.15 is shown on the transporter waste category screen.

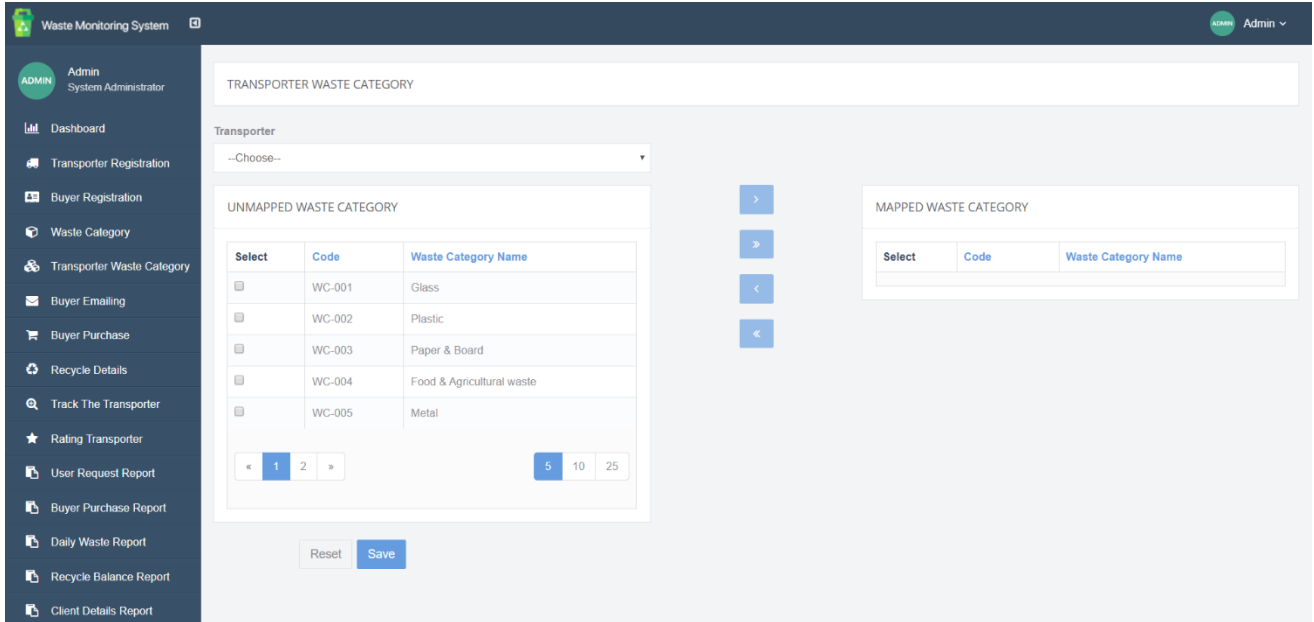


Figure 3.15 Transporter waste category page

3.12.5 Recycle Details Page - Web Application

Figure 3.16 is shown the recycle details page of the system. It allows users to enter details of the recycled items into the system.

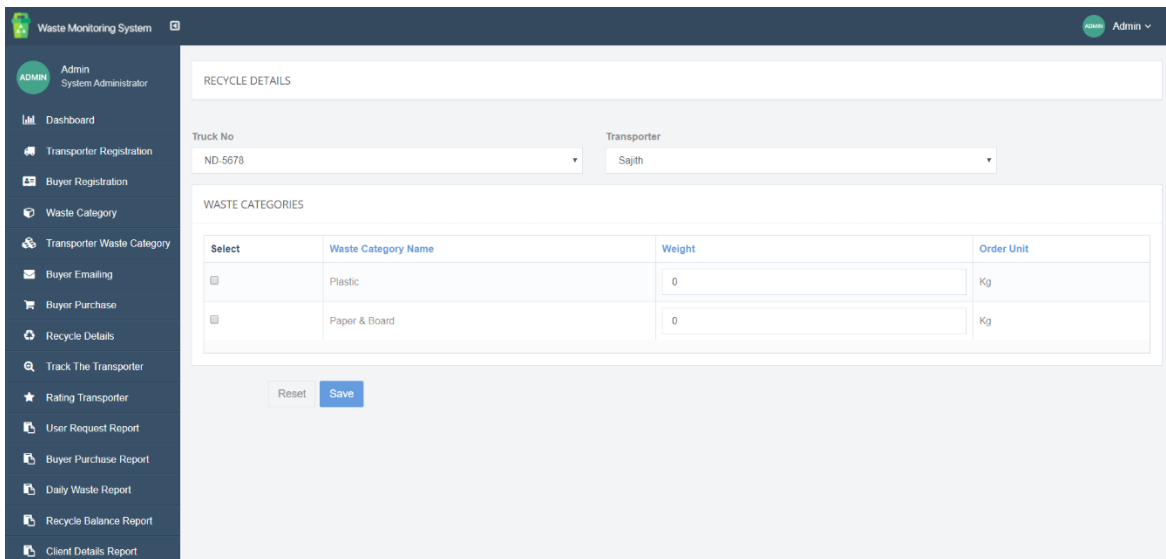


Figure 3.16 Recycle details page

3.12.6 Welcome and Login Page - Mobile Application

Figure 3.17 is shown on the login page of the mobile application. The user should enter valid credentials to access the application.

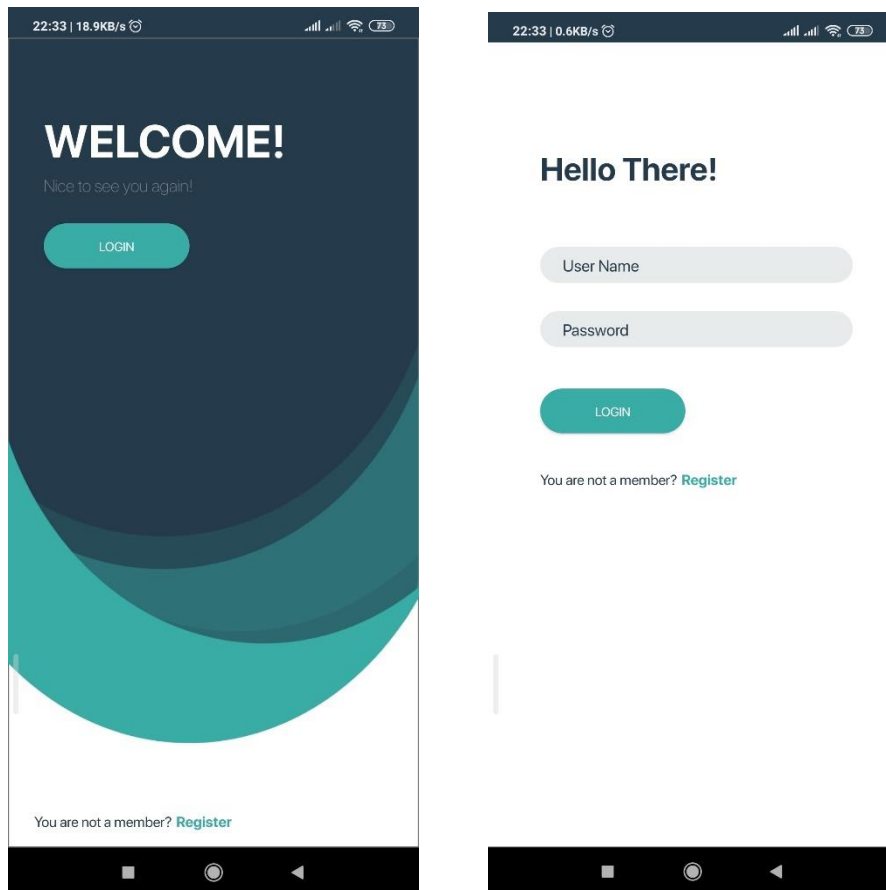


Figure 3.17 Welcome Screen and Login page

3.12.7 Client Registration - Mobile Application

Client registration of the application is shown in Figure 3.17.

22:35 | 10.3KB/s

Create Your Account

First Name

User Name

Password

Confirm Password

Location

REGISTER

RESET

You are member? [Login](#)

Figure 3.18 Client Registration page

3.12.8 Client request page - mobile application

Figure 3.18 is shown the Client request form of the mobile application. It allows users to send single requests for the collected waste.

The screenshot displays the 'New Request' form in a mobile application. At the top, there is a teal header with a hamburger menu icon on the left, the text 'New Request' in the center, and a vertical ellipsis icon on the right. Below the header, the form consists of several elements: a dropdown menu with 'Single' selected, another dropdown menu with 'Commercial' selected, a third dropdown menu with 'Glass' selected and a teal 'ADD' button to its right. Below these is a grey tag labeled 'Metal' with a black 'X' icon for removal. A text input field contains the text 'finite pvt ltd'. A toggle switch for 'Do you want to change pickup location' is currently turned off. Below that is a 'Pickup Location' field with a teal location pin icon. A date picker shows three options: '03 Apr 2024', '04 May 2025', and '05 Jun'. At the bottom, there are two large buttons: a teal 'SEND REQUEST' button and a grey 'RESET' button.

Figure 3.19 Client request -Single page

Chapter 4

Evaluation

4.1 Introduction

This chapter illustrates an overview of the testing and evaluation of the Waste Monitoring System. Evaluation is a vital part of any software development project. Evaluation can be done through the basic testing process. Mainly evaluation can be done through the verification and validation process. This chapter gives an overview of the testing techniques used for the test the WMS and Smart Bin mobile application.

4.1.1 Validation

Validation is a process of checking what we are developing is the right product. It is validated actual product with expected product. Validation belongs to dynamic testing from the software testing context. Blackbox testing, white box testing, and non-functional testing are methods used in the validation process. Under the validation process, it checks whether the software meets expectations.

4.1.2 Verification

Verification is a check whether developed software conforms to specifications or not. it includes checking codes, design documents, and other relevant documents of the project. Verification is coming under static testing from the software testing context. The methods used in verification are reviews, walkthroughs, inspections, and desk-checking.

Figure 4.1 is shown one of the testing model used in the software testing context.

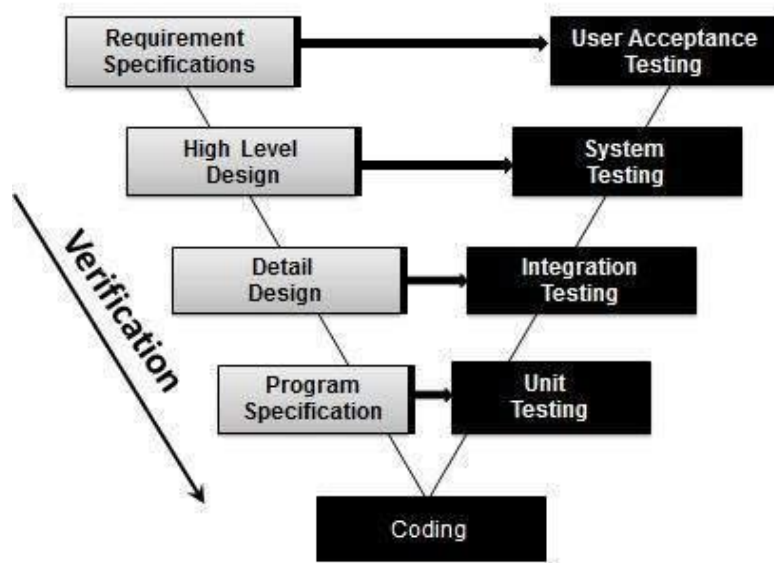


Figure 4.1 Test model

The main objective of the Test process

1. To assure the client requirements are satisfied by the software product.
2. To ensure the system is bug-free and in a stable condition.
3. To confirm that the system is ready for production.

4.2 Testing Types

Types of Software Testing:

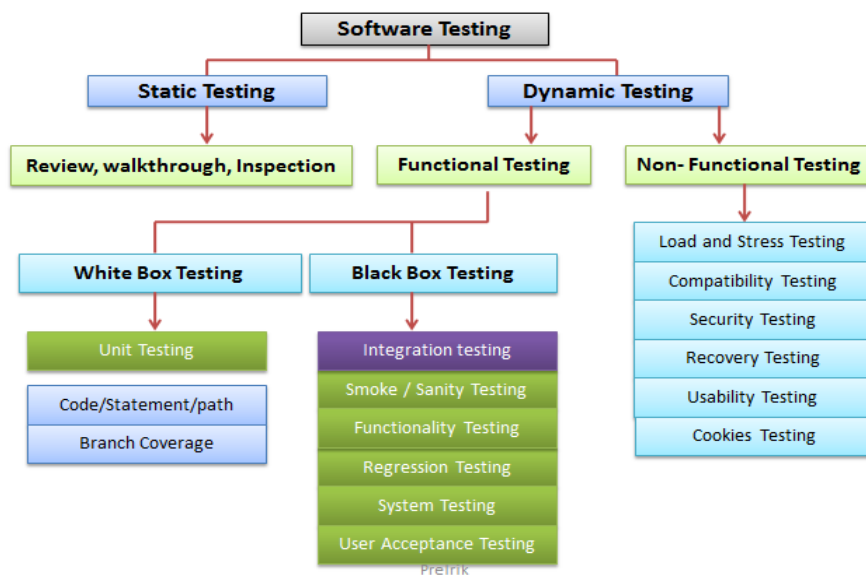


Figure 4.2 Types of Software testing

4.3 Testing Types carried on the Application

4.3.1 Static testing within WMS

Informal Reviews

The static testing technique in which the document is reviewed informally and informal comments are provided.

Inspection

Here the main purpose is to find defects. Code walkthroughs are conducted by the moderator. It is a formal type of review where a checklist is prepared to review the work documents.

Technical Reviews

This is static testing to check if the code is made according to technical specifications and standards. Generally, the test plans, test strategy, and test scripts are reviewed here.

4.3.2 Dynamic Testing within WMS

Cross-browser Testing

Cross-Browser testing was carried out for WMS through Firefox Chrome, IE 10, and Chromium Edge to verify the browser compatibility of the WMS.

Integration Testing

Integration testing is a technique of software testing carried out to verify all individual components are working fine as a group. In WMS, testing has been carried out individually for both mobile and web applications and after that integration testing is carried out to verify both applications are working fine as a system.

System Testing

System testing is also a black box testing technique performed to verify the whole system is working fine against specified requirements. In system testing, the functionalities of the entire solution are tested from an end-to-end perspective. System testing includes functional and non-functional testing to verify software has to meet client specifications.

User Acceptance Testing

User Acceptance testing is a technique performed by the end-user or client to verify software has met specifications before moving to the production environment. UAT is performed by the end-user/client to verify the application before moving to a production environment. UAT is done in the final stage of the software testing after unit, integration, and system testing is done.

4.4 Test cases of the Mobile Application

4.4.1 Login functions and Client Registration

There are two types of user registrations in the mobile application as client registration and transporter registration. All users should enter a valid username and can register with the system. There are two types of logins of the mobile application as the client and transporter. There is one login to the web application as an administrator.

Test Case - Login Functions

Test Case Id	Test Case	Expected Output	Actual Output	Pass/Fail/Not Executed/Hold
1	Check whether login functionality is working with valid inputs.	The system should successfully log in to the mobile application.	The system successfully logs in to the mobile application.	Pass
2	Check whether login functionality is not working with invalid inputs.	The system should be displayed an error as "Username or password Incorrect." and the user cannot log on to the mobile application.	The system not login into the application.	Pass
3	Check whether login functionality without mandatory fields	The system should be displayed an error message as "Username cannot be blank"	The system, not login into the application and	Pass

			display an error message	
4	Check whether login functionality without mandatory fields	The system should be displayed as an error message as “Password cannot be blank”	The system not login to the application and displays an error message.	Pass

Table 4.1 Test Case - Login Functions-Mobile and Web Application

Test Case - Client Registration Functions

Test Case Id	Test Case	Expected Output	Actual Output	Pass/Fail/Not Executed/Hold
1	Check whether the Register function with valid inputs.	The system should successfully be registered. Add client details to the database	The system displays confirmation messages as save functionality and saved client details to the database.	Pass
2	Check whether client registration with invalid inputs	The system should display error notifications in relevant fields. The system rejects to proceed	The system has not saved client registration details.	Pass

5	Check whether the email address with invalid characters.	The system should be displayed as an error message as “Invalid email address.”	The system displays an error message as “Invalid email address.”	Pass
6	Check whether Entered passwords mismatch with confirms password field.	The system should be displayed an error message as “Passwords do not match” and the user cannot register.	System displays error message as “Password do not match”	Pass
7	Check whether matched password with confirms password field.	The system should allow them to proceed and save records successfully.	The system saved the record successfully.	Pass
9	Check whether the reset button works successfully.	The system should successfully reset the details and erase the fields.	The reset button is working fine.	Pass

Table 4.2 Test Case - Client Registration -Mobile Application

4.4.2 Client Request

The client request is a major function of the Waste Monitoring System. The client can send a request as a single request or a schedule.

Test Case - Client Request

Test Case Id	Test Case	Expected Output	Actual Output	Pass/Fail/Not Executed/Hold

1	Check whether a Single request is saved.	The system should save the request	The system successfully saved the request.	Pass
2	Check whether the Schedule request is saved.	The system should be saved the request	The system successfully saved the request.	Pass
3	Google map is load successfully for the pickup location of the client	The system should be loaded with the google map for the pickup location field.	The system successfully loaded google map.	Pass
4	Check whether the client can pin location within the covered area.	The system should successfully pin and get the location	The system successfully gets location if the user pin location within a covered area.	Pass
5	Check whether the client unable to pin location, not within the covered area	The system should be displayed an error message when the location is outside “The address is outside of the area”	The system successfully displayed an error message when the location is outside of the area.	Pass

Table 4.3 Test Case - Client Request

Integration Test cases between web application and mobile application.

Test Case Id	Test Case	Expected Output	Actual Output	Pass/Fail/Not Executed/Hold
---------------------	------------------	------------------------	----------------------	------------------------------------

1	Check whether client requests are displayed as notifications in the mobile application at the transporter's end.	Client requests should be displayed in the mobile application as notifications at the relevant transporter's end.	The client's requests are successfully displayed at the relevant transporter's end in the mobile application.	Pass
2	Check whether the client request widget is updated with the client request.	The client request widget should be updated along with when the client request has been received.	The client request widget is updated successfully when the request has been received.	Pass
3	Check whether cancel client requests are displayed as notifications in a mobile application at the transporter's end	Cancel client requests should be displayed in the mobile application as notifications at the relevant transporter's end.	Cancel client's requests are successfully displayed at the relevant transporter's end in the mobile application.	Pass
4	Check whether the cancel client request list is updated with a cancel client request.	The cancel client request list should be updated along with when the client has canceled the request.	Cancel client request list is updated successfully when client has canceled the request.	Pass
5	Check whether cancel client requests by transporters are displayed as notifications in the	Cancel client requests by transporters should be displayed in the mobile application as	Cancel client's requests by transporters are successfully displayed at the relevant client's end in a mobile application.	Pass

	mobile application at the client's end	notifications at the relevant client's end.		
--	--	---	--	--

Table 4.4 Integration Test cases

4.4.3 Automation Scripts

Automation scripts are delivered for the selected scope from the system within the planned timeline of the project. Tools and technologies are selected for the test automation as followed.

- IntelliJ IDEA Community Edition
- Maven Build Tool
- Selenium WebDriver
- Java

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.Assert;
import org.testng.annotations.AfterTest;
import org.testng.annotations.Test;

public class WMS_LoginPage {
    @Test
    public void login() { // TODO Auto-generated method stub
        System.setProperty("webdriver.chrome.driver",
"E:\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("http://localhost:4200/login");
        WebElement username = driver.findElement(By.id("userName"));
        WebElement password = driver.findElement(By.id("password"));
        WebElement login = driver.findElement(By.id("SignIn"));
        username.sendKeys("Admin");
        password.sendKeys("123456");
        login.click();
        String actualUrl = "http://localhost:4200/login";
        String expectedUrl = driver.getCurrentUrl();
        Assert.assertEquals(expectedUrl, actualUrl);
    }

    @AfterTest
    public void tearDown() {
        WebDriver driver = new ChromeDriver();
        driver.close();
    }
}
```

4.4.3 User Evaluation Questionnaire

Questionnaire

1) Do you recommend this system to an organization?

- Yes
- No
- Highly Recommend

2) How satisfied are you with the reliability of this system?

- Satisfied
- Not satisfied

3) How satisfied are you with this system's ease of use?

- Satisfied
- Not satisfied

4) Is it easy to learn to use this system?

- Yes
- No

5) How satisfied are you with the look and feel of this system?

- Satisfied
- Not satisfied

6) How easy to understand the contents of the applications?

- Very Good
- Good
- Need Improvements

7) Any suggestions to improve the product? State below.

Table 4.5 Questionnaire

4.5 Test Deliverables

Test Deliverables are the test artifacts or things that are produced by the people who are involved in the testing process in the organization. Some test artifacts are documented early stages of the software life cycle means before the testing phase. Some test artifacts are documented during the testing phase and some are end of the testing phase. Below are some of the test deliverables.

Test Strategy	A test strategy is a high-level document and it defines the approach for the testing process for the organization.
Test Plan	A test plan is a document describing the plan for the testing phase including the scope of the test, approach, resources, environmental details, and risks are identified.
Test Scenarios and Test Cases Document	A Test case document is a set of test cases consist of conditions, scenarios, actual output and expected output under test satisfy requirements works correctly.
Automated Test Cases	Automated test cases are instead of manual test cases.
Test Incident Report	Test incident report set of incidents are observed and contains data like Summary, Steps Used, Priority, Severity, and Impact for the system.
Test Status Report	The test status report is about test results. This involves analyzing metrics available to support decisions making.
Test Summary Report	Test Summary Report is an important deliverable consists of various details and

	activities about the Testing performed for the Project, to the respective stakeholders
--	--

Table 4.6 Test Deliverables

4.6 Lesson Learnt

In the process of implementing WMS, it was required to gain knowledge in several aspects. Such as requirement gathering, requirement analysis, planning, web application's frontend, and backend development, mobile application development, software testing, infrastructure setup, and documentation. It has roles as System Analysis, Developer, Tester, Network Administrator, Content Writer, and Project Manager. The self-confidence to complete the project successfully on time has been a great challenge for individual and it has been a self-satisfactory goal for myself. It was mandated to learn the programming C#, HTML, Android, MySQL, and many other important new technologies to lead the way and it was a very dynamic learning curve.

4.7 Achievement of Objectives

- Web application for the organizations to manage their operations support for decision making.
- Mobile application for the Clients to log the requests
- Mobile application for the Transporters to manage the operations.

Conclusion

5.1 Introduction

This describes the conclusion of the work indicating and summary of the results of the project. It identifies any deficiencies in the final product and highlights how improvements could be made through future enhancements. Web-based system for the Administrator in the office (private or government organization) to assist daily workflows.

- Mobile application for the transporters
- Mobile application for the clients to send requests to collect waste.
- Users of the “Smart Bin” mobile application can be raising the request to collect garbage.
- An administrator can register the transporters.
- Transporters can visible the notifications of the client requests to collect garbage.
- An administrator can generate reports (Daily Client request Report/Daily Waste Report)
- An administrator can track the transporter.
- An Administrator can map the transporters with waste categories.
- An Administrator can maintain the weight records to sell to buyers
- An Administrator can send e-mails to buyers who are re-order level is reached according to the waste category.
- The transporter can view locations (house/hotels/Factories) that are Waste available within an area relevant to him.
- The transporter can start his job and finish it.
- Clients can rate transporters using a mobile application.

5.2 Future Enhancements

Some functions need to be introduced to the system to enrich the mobile application and web application. They can add more value to users.

- Introducing IoT for the system.

- Introducing sensors for the client locations to measure the weight of the waste.
- When the order level is reached system automatically, send relevant transporter notification
- Automate the Email system to buyers when the order level has reached.
- GPS Integration for the track of the transporter.

References

- [1] "Countries with the Most Sophisticated Waste Management", *www.greentumble.com/*, 2019. [Online]. Available: <https://greentumble.com/countries-with-the-most-sophisticated-waste-management/> [Accessed: 11 - Jul – 2019]
- [2] "Smart City Solutions: Successfully tackling urban challenges and problems ", *www.hub.beesmart.city/en/the-global-smart-city-knowledge-center-bee-smart-city*, 2019. [Online]. Available: <https://hub.beesmart.city/solutions/en/smart-environment/smart-waste-management-solutions-in-smart-cities> [Accessed: 04 - Jul – 2019]
- [3] "Global waste collection industry and inventor of the pneumatic waste collection system", *www.envacgroup.com*, 2019. [Online]. Available: <https://www.envacgroup.com/> [Accessed: 08 - Jul – 2019]
- [4] "Project of MANGALORE INSTITUTE OF TECHNOLOGY AND ENGINEERING, MANGALURU Project Reference NO.:39S_BE_0373", *www.pdf.semanticscholar.org/*, 2019. [Online]. Available: <https://pdf.semanticscholar.org/f835/7990df9283b6ed61286b4146059d38297984.pdf> [Accessed: 08 - Jul – 2019]
- [5] "Sri Lanka as BOI approved project on year 2010", *www.ewaste.lk*, 2018. [Online]. Available: <https://ewaste.lk/recycling-products/> [Accessed: 05 - Jul – 2019]
- [6] "Eco-friendly waste management system", *www.ecubelabs.com*, 2019. [Online]. Available: [https://www. https://www.ecubelabs.com/waste-collectors/](https://www.https://www.ecubelabs.com/waste-collectors/) [Accessed: 04 - Jul – 2019]
- [7] "Basic MVC Architecture", *www.tutorialspoint.com*, 2019. [Online]. Available: https://www.tutorialspoint.com/struts_2/basic_mvc_architecture.htm. [Accessed: 04 - Jul – 2019]

- [8] "Solid Waste Collection and Monitoring System", *www.gpsintegrated.com*, 2019.
[Online]. Available: <https://www.gpsintegrated.com/solutions/solid-waste-collection-and-monitoring-system>. [Accessed: 08 - Jul – 2019]
- [9] "Smart Bins for Smart City", *https://www.gpsintegrated.com*, 2019. [Online].
Available: <https://www.gpsintegrated.com/solutions/smart-bins-for-smart-city>.
[Accessed: 08 - Jul – 2019]
- [10] "Cognito smart solid waste management ", *www.cognitotec.com*, 2019. [Online].
Available: <http://cognitotec.com/smartwastemgmt>. [Accessed: 08 - Jul – 2019]
- [11] "Software Testing Fundamental", *www.softwaretestingfundamentals.com*, 2019.
[Online]. Available: <https://www.softwaretestingfundamentals.com>. [Accessed: 09 - Jul – 2019]

Appendix A

A.1 Web Application Screenshots

A.1.1 Login

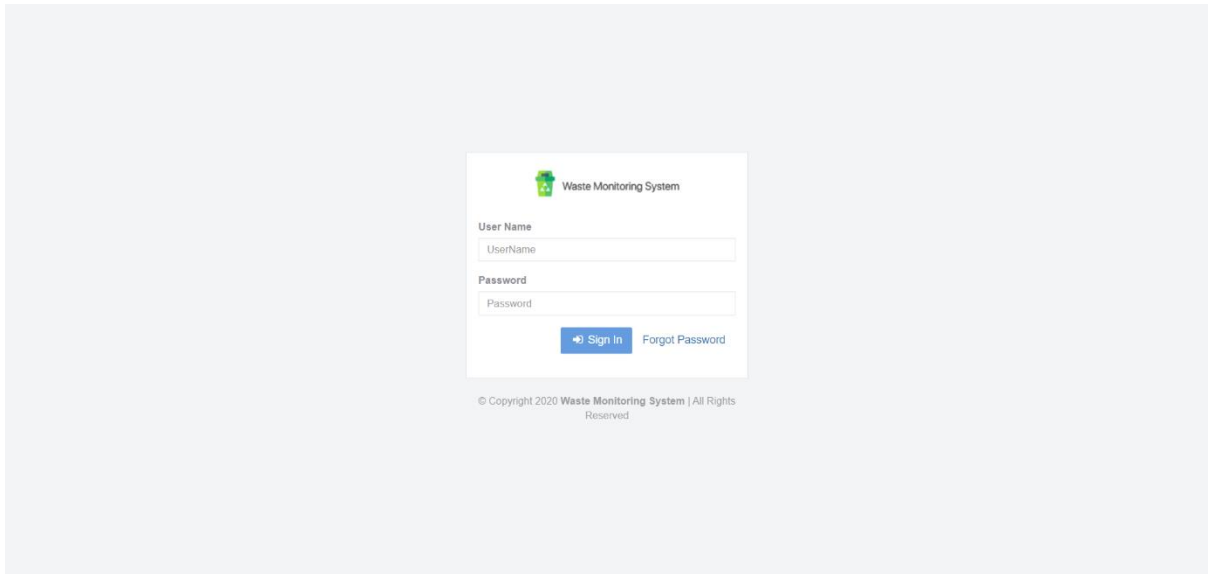


Figure A.1 Login

A.1.2 Reset Password

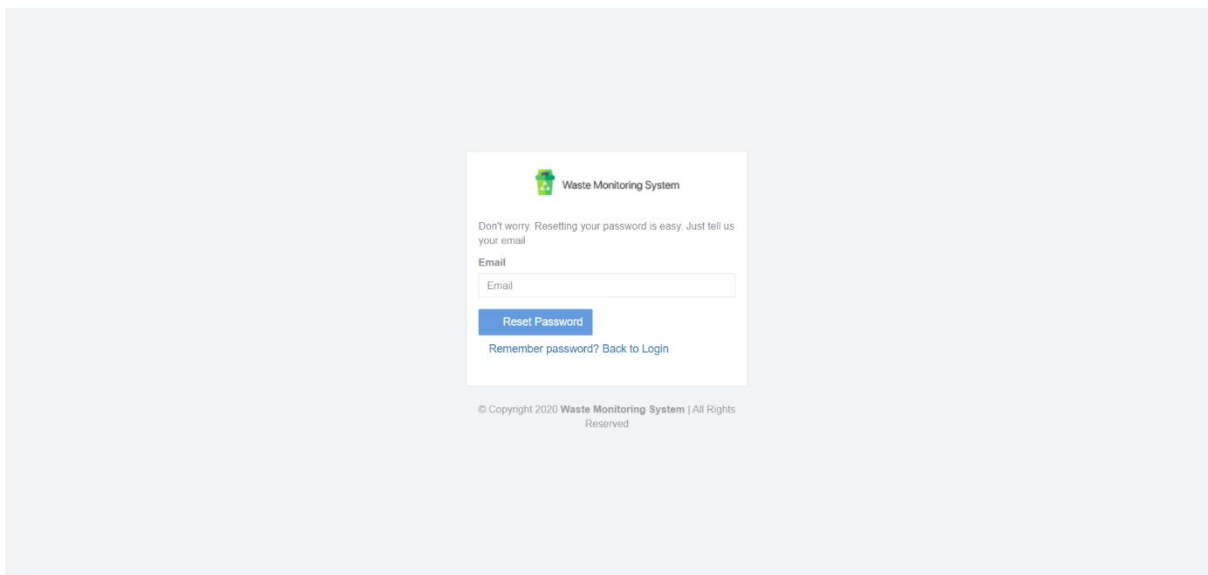


Figure A.2.a Reset password

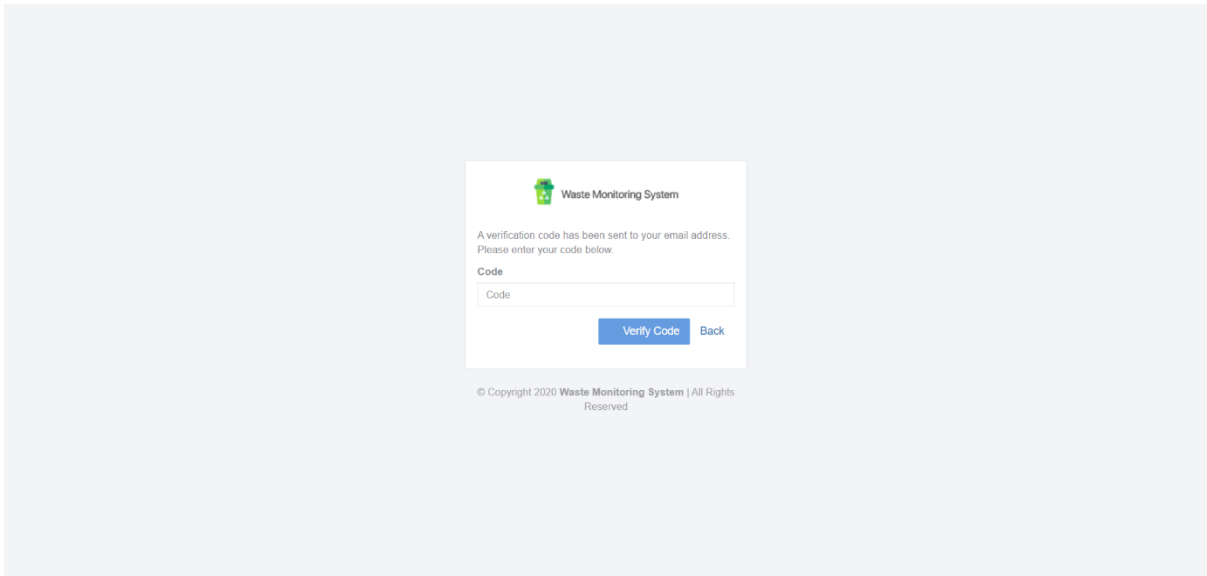


Figure A.2.b Reset password

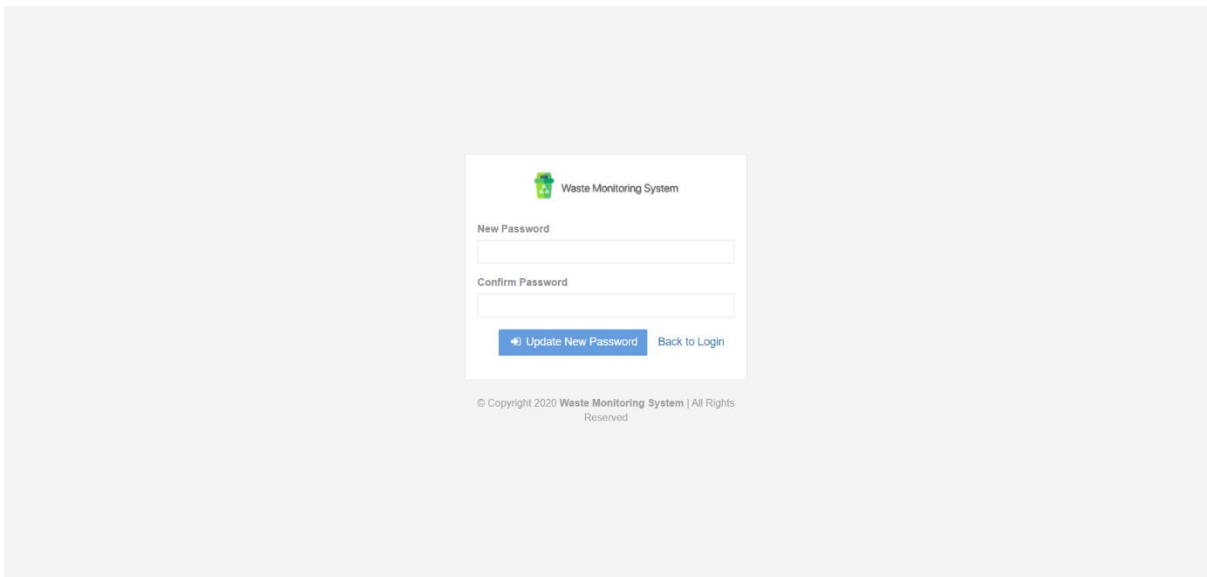


Figure A.2.c Reset password

A.1.3 Buyer Registration

The screenshot displays the 'Waste Monitoring System' interface. On the left is a dark sidebar with navigation options: Dashboard, Transporter Registration, Buyer Registration, Waste Category, Transporter Waste Category, Buyer Emailing, Buyer Purchase, Recycle Details, Track The Transporter, Rating Transporter, User Request Report, Buyer Purchase Report, Daily Waste Report, Recycle Balance Report, Client Details Report, and Change Password. The main content area is titled 'BUYER REGISTRATION' and contains a form with the following fields: Reg No (B-008), First Name, Last Name, Contact No, Email, Representative Name, and Representative ContactNo. Below the form is a 'WASTE CATEGORIES' table with columns for Select, Waste Category Name, Order Level, and Order Unit. The table lists five categories: Glass, Plastic, Paper & Board, Food & Agricultural waste, and Metal, each with an 'Order Level' of 0 and an 'Order Unit' of Kg. At the bottom of the form are 'Reset' and 'Save' buttons. Below the registration form is a 'BUYERS' table listing five registered buyers with their details and status.

BUYER REGISTRATION

Reg No: B-008

First Name: First Name

Last Name: Last Name

Contact No: Contact No

Email: Email

Representative Name: Representative Name

Representative ContactNo: Representative ContactNo

WASTE CATEGORIES

Select	Waste Category Name	Order Level	Order Unit
<input type="checkbox"/>	Glass	0	Kg
<input type="checkbox"/>	Plastic	0	Kg
<input type="checkbox"/>	Paper & Board	0	Kg
<input type="checkbox"/>	Food & Agricultural waste	0	Kg
<input type="checkbox"/>	Metal	0	Kg

Active

Reset Save

BUYERS

Edit	Reg No	First Name	Last Name	Contact No	Email	Representative Name	Representative Contact No	Status
<input type="checkbox"/>	B-001	Nihal	Silva	778702530	ruchiralakmali@gmail.com	Emil Perera	779506025	true
<input type="checkbox"/>	B-002	Nishan	Perera	782156085	nishannnb@gmail.com	Sandun Perera	762156052	true
<input type="checkbox"/>	B-003	Vinushi	Wickramasinghe	785651025	vinushiranshila@gmail.com	Kalana Perera	782456325	true
<input type="checkbox"/>	B-004	Kamal	Vithanage	778704520	kamak.s@yahoo.com	Gayani Perera	764502056	true
<input type="checkbox"/>	B-005	Saman	Perera	778954152	saman@gmail.com		4555	true

Figure A.3 Buyer registration

A.1.4 Waste Category

The screenshot displays the 'WASTE CATEGORY' management interface. On the left is a sidebar with navigation options including Dashboard, Transporter Registration, Buyer Registration, Waste Category, Transporter Waste Category, Buyer Emailing, Buyer Purchase, Recycle Details, Track The Transporter, Rating Transporter, User Request Report, Buyer Purchase Report, Daily Waste Report, Recycle Balance Report, Client Details Report, and Change Password. The main content area features a form for adding a new waste category with fields for 'Waste Category Code' (pre-filled with 'WC-007'), 'Waste Category Name', and 'Description'. There is also an 'Active' checkbox and 'Reset' and 'Save' buttons. Below the form is a table listing existing waste categories.

Edit	Waste Category Code	Waste Category Name	Description	Status
<input type="checkbox"/>	WC-001	Glass	Glass	true
<input type="checkbox"/>	WC-002	Plastic	Plastic	true
<input type="checkbox"/>	WC-003	Paper & Board	Paper & Board	true
<input type="checkbox"/>	WC-004	Food & Agricultural waste	Food & Agricultural waste	true
<input type="checkbox"/>	WC-005	Metal	Metal	true

Figure A.4 Waste Categories

A.1.5 Buyer Emailing

The screenshot displays the 'BUYER EMAILING' interface. The sidebar is identical to the previous figure. The main content area features a 'BUYER EMAILING' header, a 'Waste Category' dropdown menu (set to 'Paper & Board'), and a 'Waste Category Amount: 85 Kg' label. Below this is a table for selecting buyers to email.

Select	Buyer	Order Level
<input type="checkbox"/>	Vinushi Wickramasinghe	40 Kg
<input type="checkbox"/>	Kamal Vithanage	75 Kg

At the bottom of the table, there are 'Reset' and 'Send' buttons.

Figure A.5 Buyer emailing

A.1.6 Buyer Purchase

Waste Monitoring System

ADMIN Admin System Administrator

BUYER PURCHASE

Buyer: Vinushi

Waste Category	Available Amount	Request Amount	Order Unit
Glass	60	12	Kg
Plastic	180	5	Kg
Paper & Board	85	0	Kg

Reset Purchase

Figure A.6 Buyer purchase

A.1.7 Track Transporter

Waste Monitoring System

ADMIN Admin System Administrator

TRACK TRANSPORTER

Transporter Name	Truck No	Current Location	Status
Nihal	ND-5678	None	JobStarted
Sajith	SP-5485	None	JobEnd
Asanka	WP-4520	No125, Perera Rd, Gampaha	Ongoing
Saman	WP-4952	None	JobCanceled

Figure A.7 Track transporter

A.1.8 Rating Transporter

Waste Monitoring System Admin System Administrator

RATING TRANSPORTER

Transporter
Nihal

RATING

Client Name	Rating	Comments
Inoka peris	5	Very good

Figure A.8 Rating transporter

A.1.9 Client request Report

Waste Monitoring System Admin System Administrator

USER REQUEST REPORT

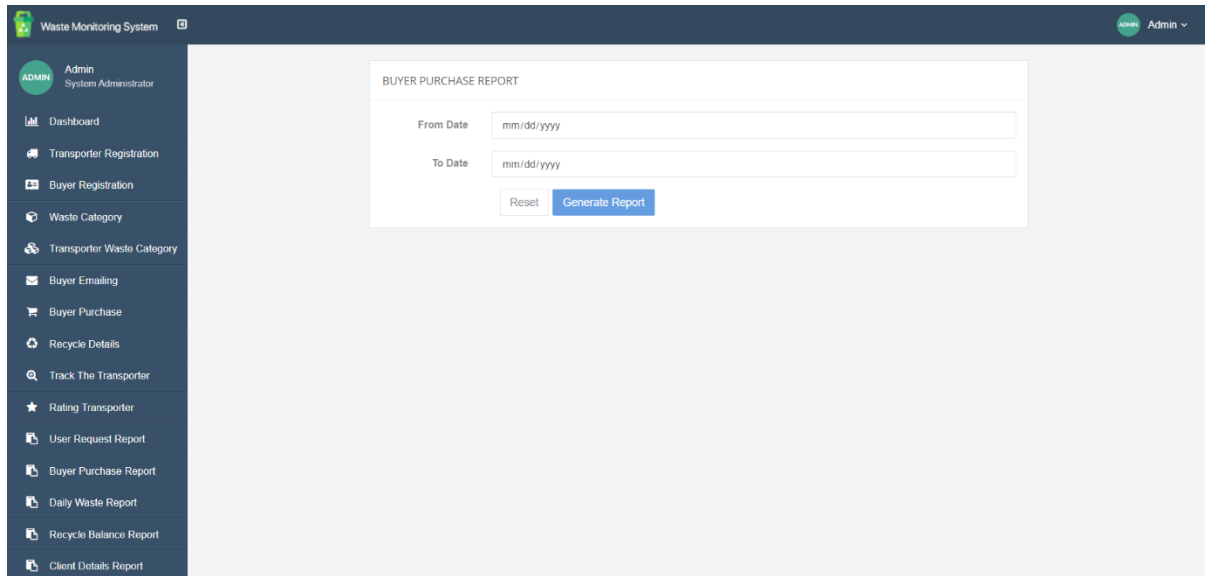
From Date mm/dd/yyyy

To Date mm/dd/yyyy

Reset Generate Report

Figure A.9 Client request report

A.1.10 Buyer Purchase Report



The screenshot displays the 'Waste Monitoring System' interface. On the left is a dark blue sidebar with a menu. The top right corner shows the user 'Admin' with a dropdown arrow. The main content area is titled 'BUYER PURCHASE REPORT' and contains a form with two date input fields labeled 'From Date' and 'To Date', both with a placeholder 'mm/dd/yyyy'. Below the fields are two buttons: 'Reset' and 'Generate Report'.

Waste Monitoring System

Admin System Administrator

ADMIN

- Dashboard
- Transporter Registration
- Buyer Registration
- Waste Category
- Transporter Waste Category
- Buyer Emailing
- Buyer Purchase
- Recycle Details
- Track The Transporter
- Rating Transporter
- User Request Report
- Buyer Purchase Report
- Daily Waste Report
- Recycle Balance Report
- Client Details Report

BUYER PURCHASE REPORT

From Date mm/dd/yyyy

To Date mm/dd/yyyy

Reset Generate Report

Figure A.10 Buyer purchase report

A.2 Mobile Application Screenshots

A.2.1 Login

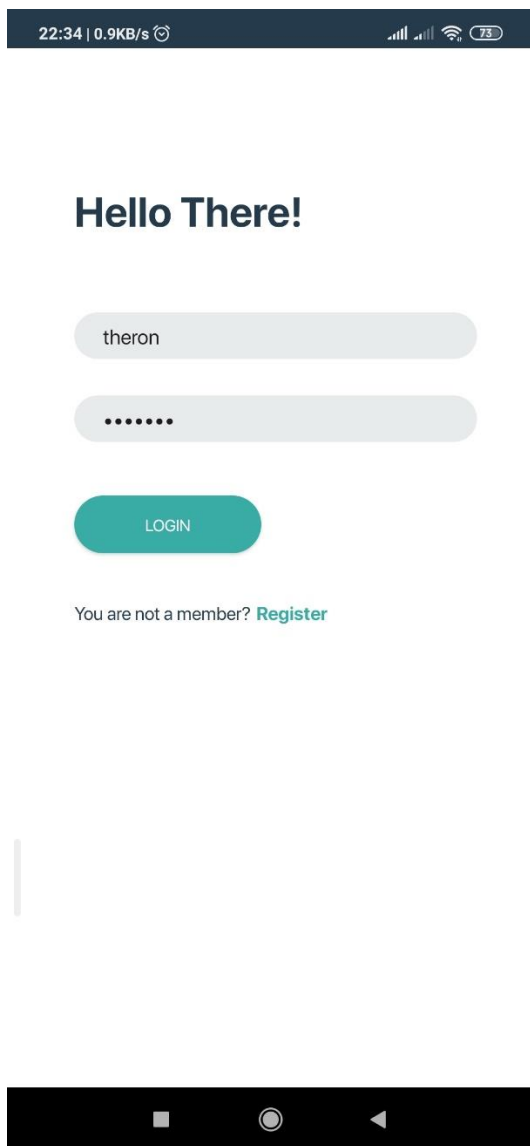


Figure A.11.a Login page

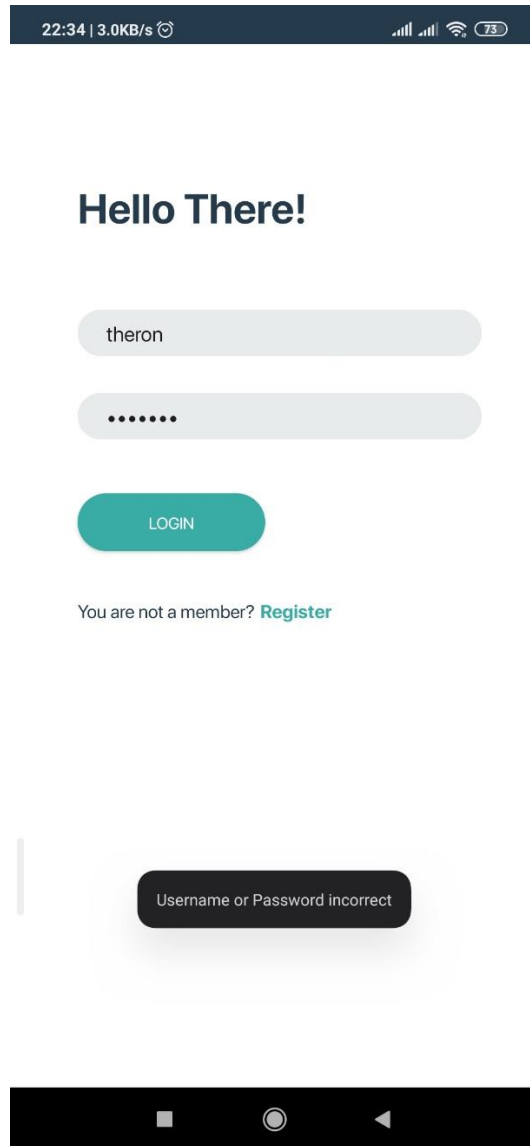


Figure A.11.b Login page with validation

A.2.2 Client Menu

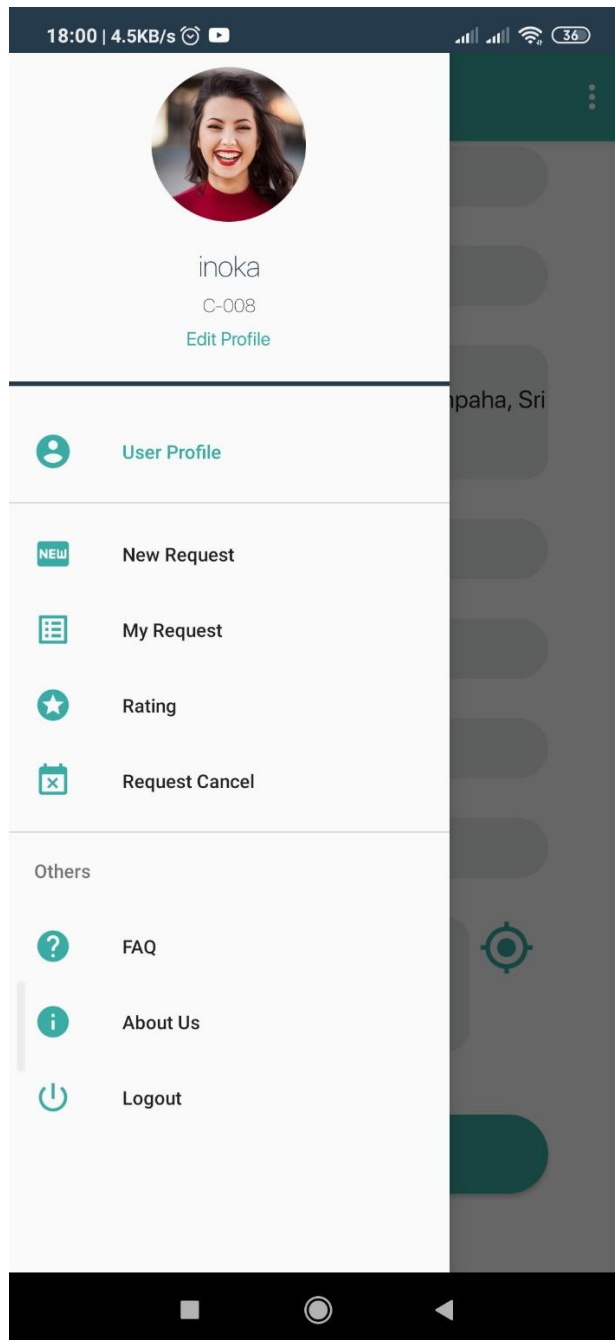


Figure A.12 Client Menu

A.2.3 New Request - Schedule

Schedule

Residential

Plastic ADD

Glass X

Plastic X

no.5, kanthi road, gampaha

Do you want to change pickup location

vikasitha asiri uyana, gampaha

31 Apr
01 May 2020
02 Jun 2021

06 Apr
07 May 2020
08 Jun 2021

SEND REQUEST

RESET

Figure A.13.a New Schedule request

22:39 | 1.4KB/s

Sri Wijaya

al Bank Branch

Nawaloka Medicare - Gampaha

District General Hospital Gampaha...

Samadhi Inst

SET PICKUP LOCATION

Figure A.13.b Pickup location

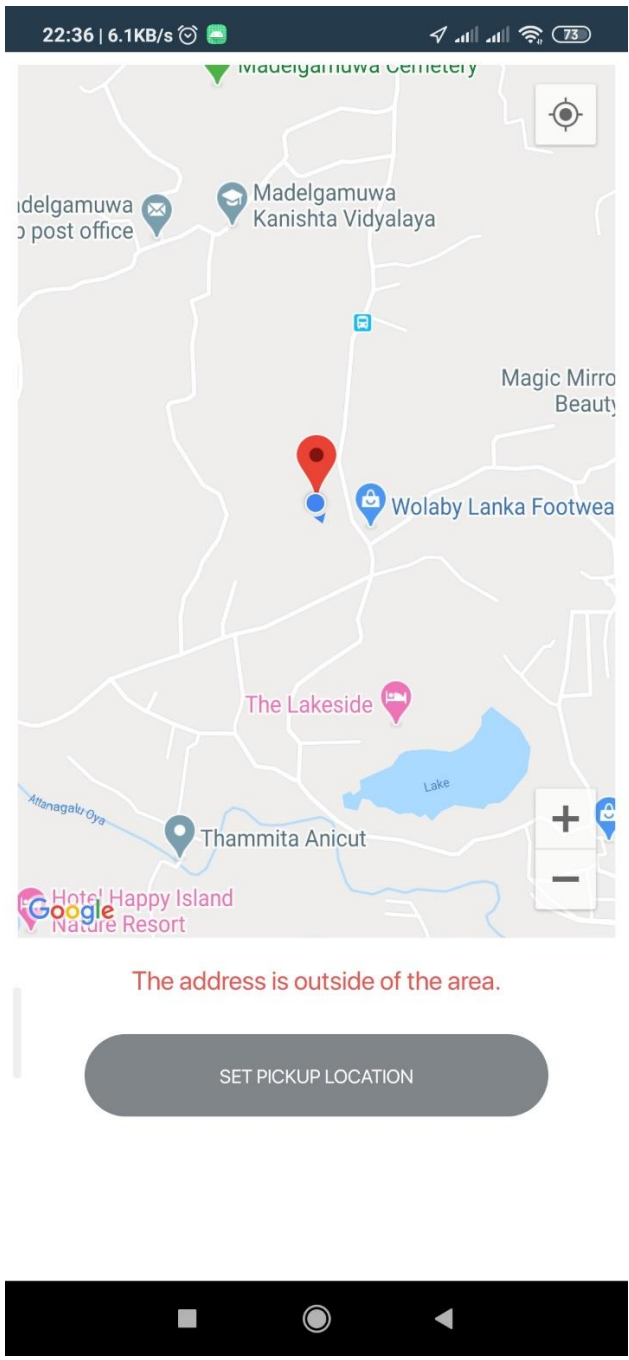


Figure A.13.c Pickup Invalid location

A.2.4 My Request

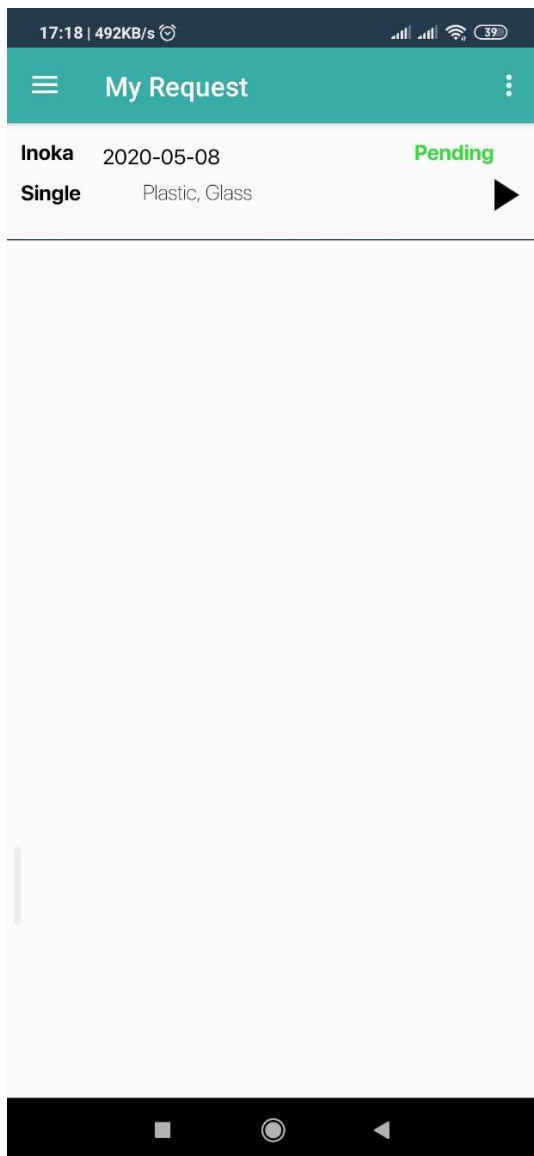


Figure A.14.a My request

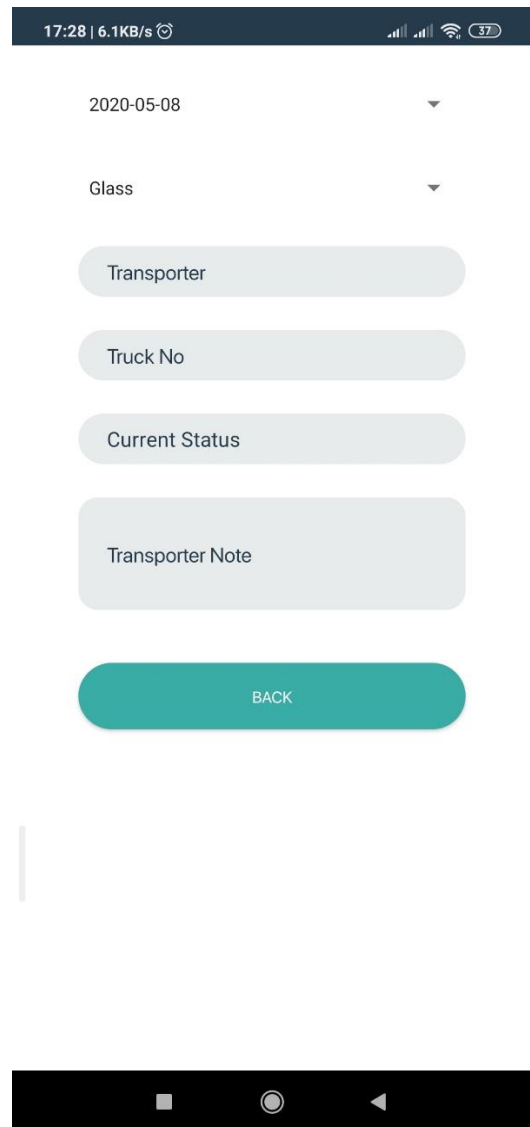


Figure A.14.b My request Detail

A.2.5 Rating

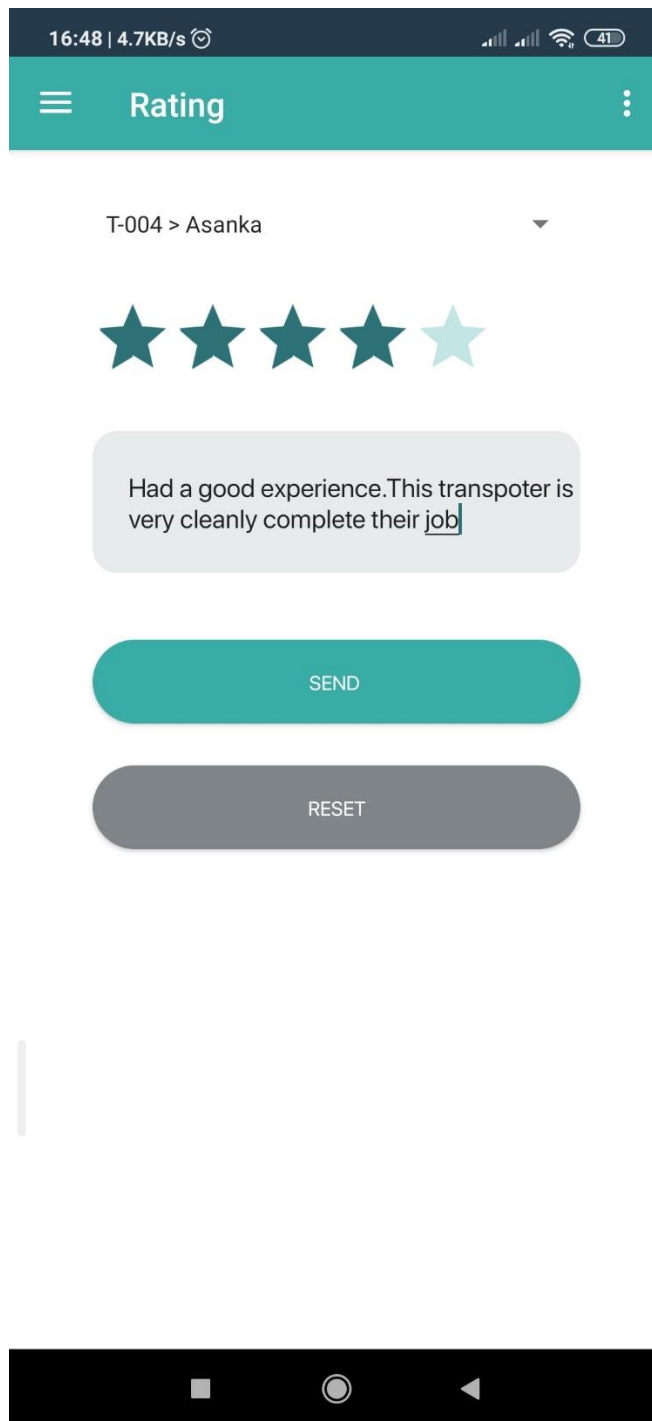


Figure A.15 Rating

A.2.6 Request Cancel

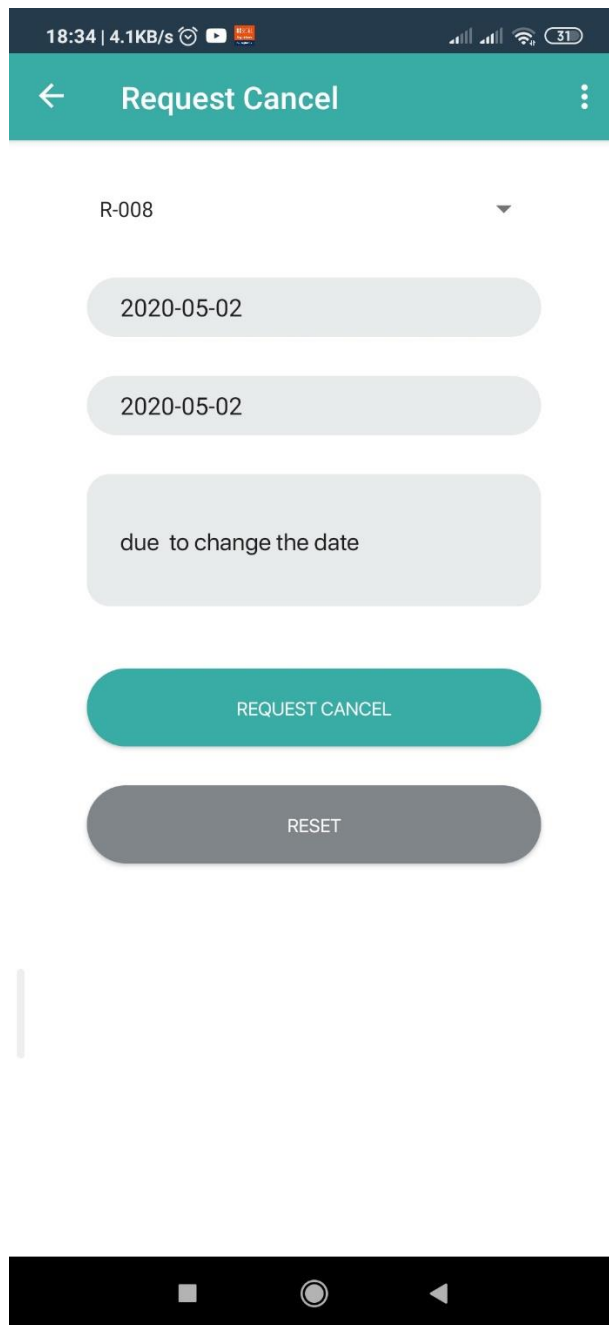


Figure A.16 Request cancels by client

A.2.7 Transporter Menu

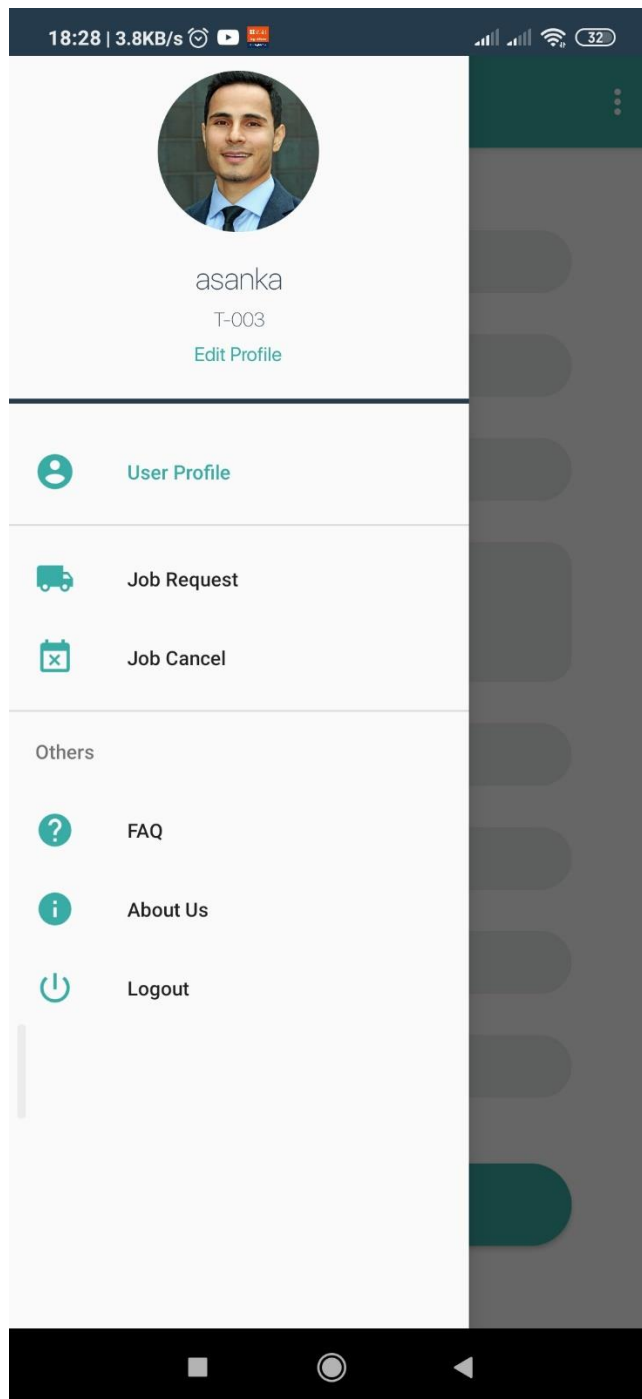


Figure A.17 Transporter Menu

A.2.8 Job Request

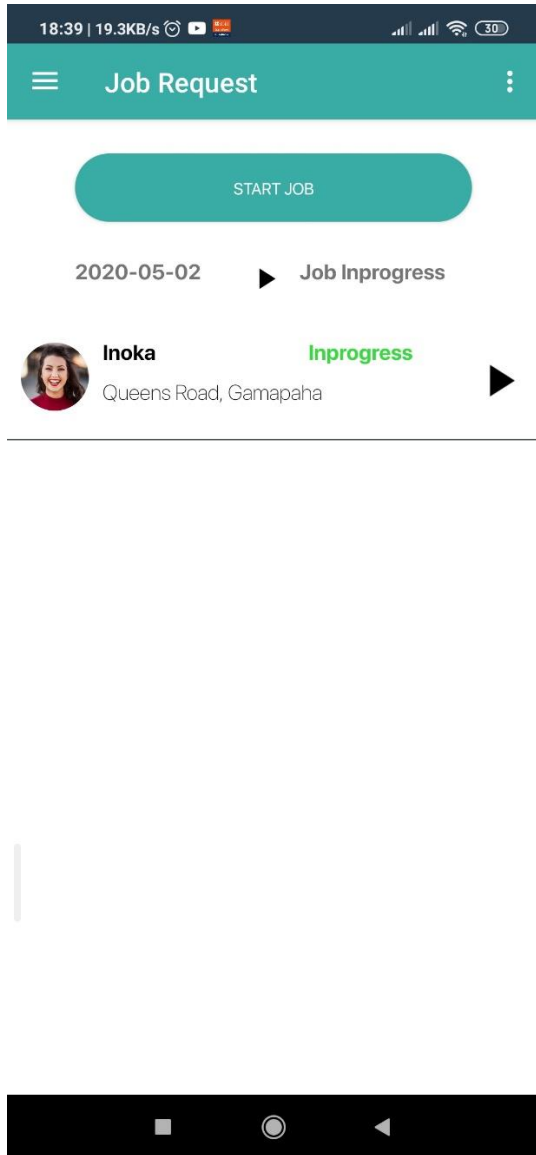


Figure A.18.a Job request

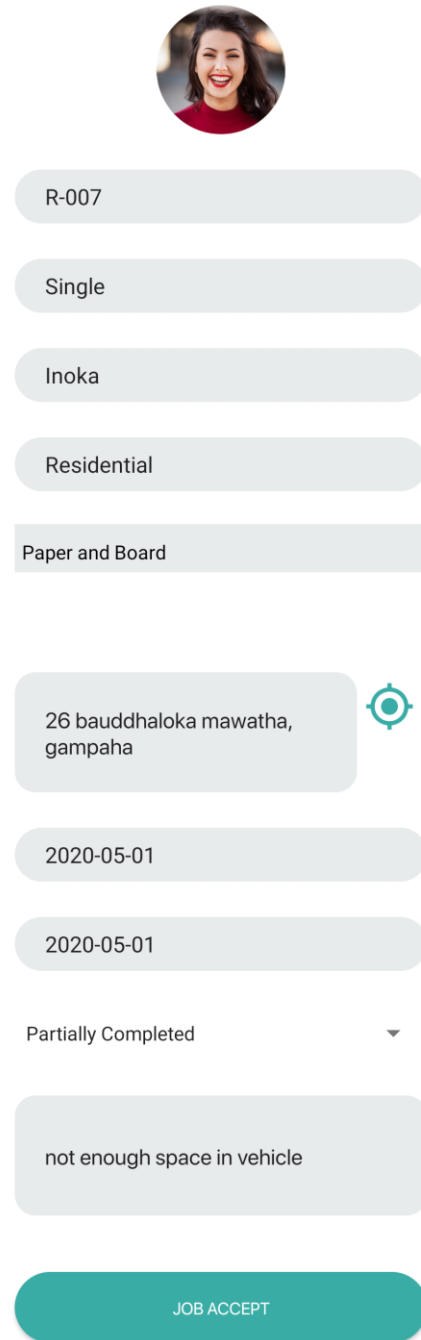


Figure A.18.b Job request Detail

A.2.9 Job Cancel

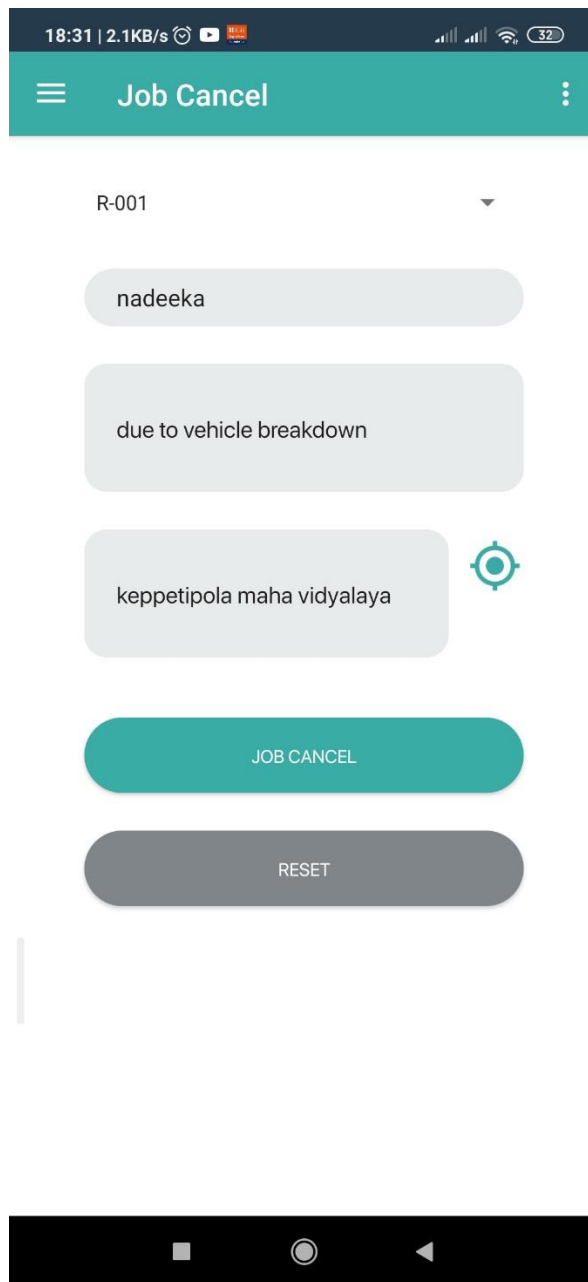


Figure A.19 Job cancel by transporter

A.2.10 FAQ



What is the request limit of the send request?

No limit. You can send any request per day as a single request or schedule request.

What if I have forgotten my password?

If forgot password should be contact service team.

Can I Cancel Request ?

Transporter can be cancel accepted request due to valid reason. Also Client can be cancel request in the pending status.

Figure A.20 FAQ page

A.2.11 About Us

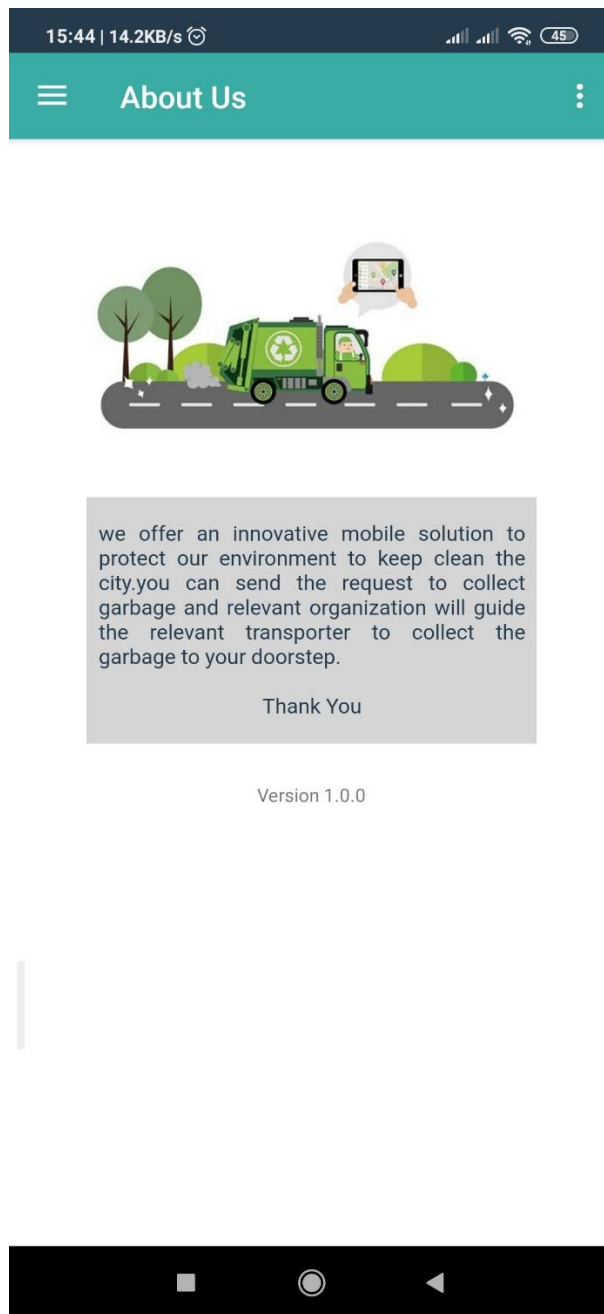


Figure A.21 About Us

Appendix B

B.1 Web Application Development Coding List

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Mail;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using WasteManagementAPI.Entities;
using WasteManagementAPI.Response;

namespace WasteManagementAPI.Controllers
{
    [ApiController]
    public class LoginsController : ControllerBase
    {
        private readonly WasteManagementContext _context;
        ApiResponse apiResponse;

        public LoginsController(WasteManagementContext context)
        {
            _context = context;
        }

        [Route("api/Logins/MobileLogin")]
        [HttpPost]
        public ApiResponse Login([FromBody] MobileLogin list)
        {
            apiResponse = new ApiResponse();
            try
            {
                var loginUser = _context.Login.Where(c => c.UserName == list.UserName
&& c.Password == list.Password).FirstOrDefault();

                if (loginUser != null)
                {
                    if (loginUser.TransporterId != null)
                    {
                        apiResponse.Code = ApiResponseCode.Success;
                        apiResponse.Message = "isTransporter";
                        apiResponse.Result = loginUser;
                    }
                    else if (loginUser.ClientId != null)
                    {
                        apiResponse.Code = ApiResponseCode.Success;
                        apiResponse.Message = "isClient";
                        apiResponse.Result = loginUser;
                    }
                    else
                    {
                        apiResponse.Code = ApiResponseCode.Error;
                        apiResponse.Message = "Login User Not Found";
                        apiResponse.Result = false;
                    }
                }
            }
        }
    }
}
```

```

        else
        {
            apiResponse.Code = ApiResponseCode.Error;
            apiResponse.Message = "Login User Not Found";
            apiResponse.Result = false;
        }
    }
    catch (Exception ex)
    {
        apiResponse.Code = ApiResponseCode.Error;
        apiResponse.Message = ex.Message;
        apiResponse.Result = false;
    }
    return apiResponse;
}

[Route("api/Logins/Login")]
[HttpPost]
public ApiResponse Login([FromBody] Login login)
{
    apiResponse = new ApiResponse();
    try
    {
        var loginUser = _context.Login.Where(c => c.UserName == login.UserName
&& c.Password == login.Password).FirstOrDefault();

        if (loginUser != null)
        {
            apiResponse.Code = ApiResponseCode.Success;
            apiResponse.Message = "";
            apiResponse.Result = loginUser;
        }
        else
        {
            apiResponse.Code = ApiResponseCode.Error;
            apiResponse.Message = "Login User Not Found";
            apiResponse.Result = false;
        }
    }
    catch (Exception ex)
    {
        apiResponse.Code = ApiResponseCode.Error;
        apiResponse.Message = ex.Message;
        apiResponse.Result = false;
    }
    return apiResponse;
}

[Route("api/Logins/ForgotPassword")]
[HttpPost]
public ApiResponse ForgotPassword([FromBody] ForgotPassword forgotPassword)
{
    apiResponse = new ApiResponse();
    try
    {
        var webClient = _context.WebClient.Where(c => c.Email ==
forgotPassword.Email).SingleOrDefault();

        if (webClient != null)
        {
            string subject = "reset your password";

```

```

        string body = "Hi " + webClient.Name + ",<br/><br/>You recently
requested to reset your password for your Waste Monitoring account.Please use the
below code to reset.<br/><p><b>abc123</b></p><br/>If you did not request a password
reset, you can safely ignore this email.<br/><p>Thanks,</p>Team Waste Monitoring
system";

        if (Email(forgotPassword.Email, subject, body, null))
        {
            apiResponse.Code = ApiResponseCode.Success;
            apiResponse.Message = "";
            apiResponse.Result = "abc123";
        }
    }
    else
    {
        apiResponse.Code = ApiResponseCode.Error;
        apiResponse.Message = "Email is invalid";
        apiResponse.Result = false;
    }
}
catch (Exception ex)
{
    apiResponse.Code = ApiResponseCode.Error;
    apiResponse.Message = ex.Message;
    apiResponse.Result = false;
}
return apiResponse;
}

public static bool Email(string toEmail, string subject, string body,
List<string> toBcc)
{
    try
    {
        MailMessage mail = new MailMessage();
        SmtplibClient SmtplibServer = new SmtplibClient("smtp.gmail.com", 587);

        mail.From = new
MailAddress("wastemonitorsystem.2016MIT080@gmail.com");
        mail.To.Add(toEmail);
        if (toBcc != null)
        {
            foreach (var val in toBcc)
            {
                mail.Bcc.Add(val);
            }
        }
        mail.Subject = subject;
        mail.Body = body;
        mail.IsBodyHtml = true;
        SmtplibServer.UseDefaultCredentials = false;
        SmtplibServer.Credentials = new
NetworkCredential("wastemonitorsystem.2016MIT080@gmail.com", "waste@123");
        SmtplibServer.EnableSsl = true;
        SmtplibServer.DeliveryMethod = SmtplibDeliveryMethod.Network;
        SmtplibServer.Send(mail);
        return true;
    }
    catch (Exception ex)
    {
        return false;
    }
}
}

```

```

[Route("api/Logins/CreateChangePassword")]
[HttpPost]
public ApiResponse CreateChangePassword([FromBody] ChangePassword
changePassword)
{
    ApiResponse = new ApiResponse();
    try
    {
        var login = _context.Login.Where(c => c.UserName ==
changePassword.UserName).SingleOrDefault();

        if (login != null)
        {
            if (login.Password == changePassword.OldPassword)
            {
                login.Password = changePassword.Password;
                _context.Login.Update(login);
                _context.SaveChanges();
                ApiResponse.Code = ApiResponseCode.Success;
                ApiResponse.Message = "Password changed successfully";
                ApiResponse.Result = true;
            }
            else
            {
                ApiResponse.Code = ApiResponseCode.Error;
                ApiResponse.Message = "Password is invalid";
                ApiResponse.Result = false;
            }
        }
        else
        {
            ApiResponse.Code = ApiResponseCode.Error;
            ApiResponse.Message = "UserName is invalid";
            ApiResponse.Result = false;
        }
    }
    catch (Exception ex)
    {
        ApiResponse.Code = ApiResponseCode.Error;
        ApiResponse.Message = ex.Message;
        ApiResponse.Result = false;
    }
    return ApiResponse;
}

[Route("api/Logins/ResetPassword")]
[HttpPost]
public ApiResponse ResetPassword([FromBody] ResetPassword resetPassword)
{
    ApiResponse = new ApiResponse();
    try
    {
        var webClient = _context.WebClient.Where(c => c.Email ==
resetPassword.Email).SingleOrDefault();

        if (webClient != null)
        {
            var login = _context.Login.Where(c => c.WebClientId ==
webClient.Id).SingleOrDefault();

            if (login != null)

```

```

        {
            login.Password = resetPassword.Password;
            _context.Login.Update(login);
            _context.SaveChanges();
            apiResponse.Code = ApiResponseCode.Success;
            apiResponse.Message = "Password changed successfully";
            apiResponse.Result = true;
        }
        else
        {
            apiResponse.Code = ApiResponseCode.Error;
            apiResponse.Message = "Login is invalid";
            apiResponse.Result = false;
        }
    }
    else
    {
        apiResponse.Code = ApiResponseCode.Error;
        apiResponse.Message = "UserName is invalid";
        apiResponse.Result = false;
    }
}
catch (Exception ex)
{
    apiResponse.Code = ApiResponseCode.Error;
    apiResponse.Message = ex.Message;
    apiResponse.Result = false;
}
return apiResponse;
}
}
}

```

Figure B.1 Login Controller.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.EntityFrameworkCore;
using WasteManagementAPI.Entities;

public class WasteManagementContext : DbContext
{
    public WasteManagementContext(DbContextOptions<WasteManagementContext> options)
        : base(options)
    {
    }

    public DbSet<Transporter> Transporter { get; set; }

    public DbSet<Login> Login { get; set; }

    public DbSet<LoginHistory> LoginHistory { get; set; }

    public DbSet<Client> Client { get; set; }

    public DbSet<WasteCategory> WasteCategory { get; set; }

    public DbSet<WasteLocation> WasteLocation { get; set; }

    public DbSet<Buyer> Buyer { get; set; }

    public DbSet<BuyerWasteCategory> BuyerWasteCategory { get; set; }

    public DbSet<WebClient> WebClient { get; set; }

    public DbSet<TransporterWasteCategory> TransporterWasteCategory { get; set; }

    public DbSet<UserRequest> UserRequest { get; set; }

    public DbSet<UserRequestStatus> UserRequestStatus { get; set; }

    public DbSet<BuyerPurchase> BuyerPurchase { get; set; }

    public DbSet<TransporterJobStatus> TransporterJobStatus { get; set; }

    public DbSet<RequestedScheduledWaste> RequestedScheduledWaste { get; set; }

    public DbSet<RequestedScheduledWasteStatus> RequestedScheduledWasteStatus { get;
set; }

    public DbSet<RequestScheduledDate> RequestScheduledDate { get; set; }

    public DbSet<RequestScheduledDateStatus> RequestScheduledDateStatus { get; set; }

    public DbSet<BuyerEmailing> BuyerEmailing { get; set; }

    public DbSet<TransporterWasteWeight> TransporterWasteWeight { get; set; }

    public DbSet<TransporterRatings> TransporterRatings { get; set; }

}

```

Figure B.2 WasteManagementContext.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations.Schema;
using System.Linq;
using System.Threading.Tasks;

namespace WasteManagementAPI.Entities
{
    public class MobileLogin
    {
        public string UserName { get; set; }

        public string Password { get; set; }
    }

    public class Login
    {
        public int Id { get; set; }

        public int? WebClientId { get; set; }

        public int? TransporterId { get; set; }

        public int? ClientId { get; set; }

        public string UserName { get; set; }

        public string Password { get; set; }

        public int? CreateUser { get; set; }

        public int? ModifyUser { get; set; }

        public DateTime? CreatedDate { get; set; }

        public DateTime? ModifiedDate { get; set; }

        //Navigation Properties
        [ForeignKey("TransporterId")]
        public virtual Transporter Transporter { get; set; }

        [ForeignKey("ClientId")]
        public virtual Client Client { get; set; }

        [ForeignKey("WebClientId")]
        public virtual WebClient WebClient { get; set; }

        public ICollection<LoginHistory> LoginHistories { get; set; }
    }
}

```

Figure B.3 Login Entity.cs

```

<!-- MAIN CONTAINER -->
<main class="main-container no-margin no-padding">

  <!-- FULLSCREEN -->
  <div class="fullscreen">

    <!-- VERTICAL MIDDLE -->
    <div class="vertical-middle">

      <!-- CONTENT AREA -->
      <div class="content container">

        <div class="row">
          <div class="col-xs-12 col-sm-6 col-md-4 col-sm-offset-3 col-
md-offset-4">

            <!-- PANEL: Authorization -->
            <div class="panel">

              <!-- Panel Body -->
              <div class="panel-body">

                <div class="image mb text-center">
                  
                </div>

                <form name="form" #f="ngForm" novalidate>

                  <input type="hidden" name="action" value="lo
gin">

                  <div class="form-group">
                    <label for="userName">User Name</label>
                    <input type="text" id="userName" name="u
serName" class="form-control input-
sm" placeholder="UserName" [(ngModel)]="model.userName" #userName="ngModel" [n
gClass]="{ 'invalid-data': userName.invalid && userName.touched, 'valid-
data': userName.valid }" required>
                      <span class="invalid-
text" *ngIf="userName.touched && userName.invalid && userName.errors.required"
>User
                        Name is required</span>
                    </div>

                    <div class="form-group">
                      <label for="password">Password</label>

```



```

        <input type="password" id="password" cla
ss="form-control input-
sm" placeholder="Password" [(ngModel)]="model.password" #password="ngModel" [n
gClass]="{ 'invalid-data': password.invalid && password.touched, 'valid-
data': password.valid }" name="password" required>
        <span class="invalid-
text" *ngIf="password.touched && password.invalid && password.errors.required"
>Password is required</span>
    </div>

    <div class="form-group pull-right">
        <button type="submit" class="btn btn-
primary" (click)="login()" ><i class="fa fa-fw fa-sign-
in"></i> Sign In</button>
        <button type="button" class="btn btn-
link" routerLink="/login/forgot-password">Forgot Password</button>
    </div>

</form>

</div>
<!-- /Panel Body -->

</div>
<!-- /PANEL: Authorization -->

<!-- Copyright -->
<p class="text-muted text-center">
    &copy; Copyright 2020 <strong>Waste Monitoring System</strong> | All Rights Reserved
</p>
<!-- /Copyright -->

</div>
</div>

</div>
<!-- /CONTENT AREA -->

</div>
<!-- /VERTICAL MIDDLE -->

</div>
<!-- /FULLSCREEN -->

</main>
<!-- /MAIN CONTAINER -->

```

Figure B.4 Login Html

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { LoginService } from '../../service/login.service';
import { Login } from '../../model/login';
import { ViewChild } from '@angular/core';
import { Form, NgForm } from '@angular/forms';
import { ToastrService } from 'ngx-toastr';

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css']
})
export class LoginComponent implements OnInit {

  @ViewChild('f') form: NgForm;
  constructor(private router: Router, private loginService: LoginService, private toastr: ToastrService) { }

  model: Login = new Login();

  ngOnInit() {
    this.loginService.checkToken();
  }

  login() {
    if (this.form.valid) {
      this.loginService.login(this.model);
    }
  }
}

```

Figure B.5 Login Component.ts

```

import { Injectable } from '@angular/core';
import { Router } from '@angular/router';
import { HttpClient, HttpParams, HttpHeaders } from '@angular/common/http';
import { Login } from '../model/login';
import { ForgotPassword } from '../model/forgot-password';
import { ResetPassword } from '../model/reset-password';
import { environment } from '../../environments/environment';
import { ApiResponse } from '../model/api-response';
import { Subscription } from 'rxjs';
import { ToastrService } from 'ngx-toastr';

@Injectable({
  providedIn: 'root'
})
export class LoginService {

  constructor(private http: HttpClient, private router: Router, private toastr:
  ToastrService) { }
  sub: Subscription;
  isLoggedIn = false;

  login(login: Login) {
    const headers = { headers: new HttpHeaders({ 'Content-
Type': 'application/json' }) };
    return this.http.post<ApiResponse>(environment.apiUrl + '/Logins/Login', l
ogin, headers).subscribe(response => {
      debugger;
      if (response.result) {
        localStorage.setItem('currentUser', login.userName);
        this.isLoggedIn = true;
        this.router.navigate(['/dashboard']);
      }
    }, error => {
      console.log(error.message);
      this.toastr.error(error.message, 'error');
    });
  }

  logOut() {
    this.isLoggedIn = false;
    localStorage.removeItem('currentUser');
    this.router.navigate(['/login']);
  }

  checkToken() {
    if (localStorage.getItem('currentUser') == null || localStorage.getItem('c
urrentUser') === 'undefined') {
      this.router.navigate(['/login']);
    }
  }
}

```

```

    } else {
      if (this.router.url === '/login') {
        this.router.navigate(['/dashboard']);
      }
    }
  }

  forgotPassword(forgotPassword: ForgotPassword) {
    const headers = { headers: new HttpHeaders({ 'Content-Type': 'application/json' }) };
    return this.http.post<ApiResponse>(environment.apiUrl + '/Logins/ForgotPassword', forgotPassword, headers);
  }

  updateNewPassword(resetPassword: ResetPassword) {
    const headers = { headers: new HttpHeaders({ 'Content-Type': 'application/json' }) };
    return this.http.post<ApiResponse>(environment.apiUrl + '/Logins/ResetPassword', resetPassword, headers);
  }
}

```

Figure B.6 Login Service.ts

```

export class Login {
  id: string;
  userName: string;
  password: string;
}

```

Figure B.7 Login Model.ts

B.2 Mobile Application Development Coding List

```
package com.example.wastemonitoringapp;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Base64;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.IOException;

import cz.msebera.android.httpclient.HttpResponse;
import cz.msebera.android.httpclient.client.HttpClient;
import cz.msebera.android.httpclient.client.methods.HttpGet;
import cz.msebera.android.httpclient.client.methods.HttpPost;
import cz.msebera.android.httpclient.entity.StringEntity;
import cz.msebera.android.httpclient.impl.client.DefaultHttpClient;
import cz.msebera.android.httpclient.util.EntityUtils;

public class Login_Screen extends AppCompatActivity {

    String login_url = "http://192.168.1.4:5053/api/Logins/MobileLogin";

    TextView registerTxt;
    Button loginBtn;
    EditText userNameTxt;
    EditText passwordTxt;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.Login_screen);

        userNameTxt = findViewById(R.id.L_userNameTxt);
        passwordTxt = findViewById(R.id.L_passwordTxt);
        registerTxt = findViewById(R.id.L_registerTxt);
        loginBtn = findViewById(R.id.L_LoginBtn);

        registerTxt.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(Login_Screen.this,
SignUp_Screen.class);
                startActivity(intent);
            }
        })
    }
}
```

```

});

loginBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (!userNameTxt.getText().toString().isEmpty()) {
            if (!passwordTxt.getText().toString().isEmpty()) {
                new Login().execute();
            } else {
                Toast.makeText(getApplicationContext(), "Password can't
blank", Toast.LENGTH_LONG).show();
            }
        } else {
            Toast.makeText(getApplicationContext(), "User name can't
blank", Toast.LENGTH_LONG).show();
            // userNameTxt.setError("enter a valid User name address");
        }
    }
});
}

// private methods

private class Login extends AsyncTask<String, Void, String> {
    String uName = userNameTxt.getText().toString();
    String pwd = passwordTxt.getText().toString();
    HttpResponse response = null;

    protected void onPreExecute() {
        Toast.makeText(getApplicationContext(), "Redirecting...",
Toast.LENGTH_SHORT).show();
    }

    @Override
    protected String doInBackground(String... paramsObj) {
        HttpClient httpclient = new DefaultHttpClient();
        HttpPost httppost = new HttpPost(login_url);

        try {
            JSONObject jsonParam = new JSONObject();
            jsonParam.put("UserName", uName);
            jsonParam.put("Password", pwd);

            StringEntity entity = new StringEntity(jsonParam.toString());
            httppost.setEntity(entity);
            // httppost.setHeader("Accept", "application/json");
            httppost.setHeader("Content-type", "application/json");

            response = httpclient.execute(httppost);

            if (response.getStatusLine().getStatusCode() == 200) {
                String server_response =
EntityUtils.toString(response.getEntity(), "UTF-8");
                JSONObject resObj = new JSONObject(server_response);
                if (resObj.get("code").equals("Success")) {

```

```

        SharedPreferences.Editor editor =
getSharedPreferences("shared_file", MODE_PRIVATE).edit();
        JSONObject dataArr = new
JSONObject(resObj.get("result").toString());
        if (resObj.get("message").equals("isTransporter")) {
            editor.clear();
            editor.putString("is_client", "false");
            editor.putString("client_id",
String.valueOf(dataArr.get("transporterId")));
            editor.commit();
            return "Success";

        } else if (resObj.get("message").equals("isClient")) {
            editor.clear();
            editor.putString("is_client", "true");
            editor.putString("client_id",
String.valueOf(dataArr.get("clientId")));
            editor.commit();
            return "Success";
        }
        return "Failed";
    } else {
        return "Failed";
    }
} else {
    Log.i("Server response", "Failed to get server response");
}
} catch (IllegalStateException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (JSONException e) {
    e.printStackTrace();
}
}
return null;
}

protected void onPostExecute(String result) {
    super.onPostExecute(result);

    if (result == "Success") {
        Toast.makeText(getApplicationContext(), "Login Successful",
Toast.LENGTH_SHORT).show();
        Intent intent = new Intent(Login_Screen.this, MainScreen.class);
        startActivity(intent);
    } else {
        Toast.makeText(getApplicationContext(), "Username or Password
incorrect", Toast.LENGTH_LONG).show();
    }
}
}
}
}
}
}

```

Figure B.8 LoginScreen.java

```

<?xml version="1.0" encoding="utf-8"?>

<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    style="@android:style/Widget.ScrollView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorWhite"
    android:scrollbarStyle="insideInset">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".Login_Screen">

        <TextView
            android:id="@+id/l_helloThereTxt"
            android:layout_width="201dp"
            android:layout_height="50dp"
            android:layout_marginStart="50dp"
            android:layout_marginTop="100dp"
            android:fontFamily="@font/sf_pro_display_bold"
            android:text="Hello There!"
            android:textColor="@color/colorGrey"
            android:textSize="36sp"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.0"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintVertical_bias="0.0" />

        <EditText
            android:id="@+id/l_userNameTxt"
            android:layout_width="300dp"
            android:layout_height="40sp"
            android:layout_marginStart="50dp"
            android:layout_marginTop="47dp"
            android:layout_marginEnd="50dp"
            android:background="@drawable/input_field"
            android:ems="10"
            android:fontFamily="@font/sf_pro_display_regular"
            android:hint="User Name"
            android:inputType="text"
            android:textColorHint="@color/colorGrey"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.0"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/l_helloThereTxt"
            app:layout_constraintVertical_bias="0.0" />

        <EditText
            android:id="@+id/l_passwordTxt"
            android:layout_width="300dp"
            android:layout_height="40sp"
            android:layout_marginStart="50dp"
            android:layout_marginTop="25dp"

```



```

android:layout_marginEnd="50dp"
android:background="@drawable/input_field"
android:ems="10"
android:fontFamily="@font/sf_pro_display_regular"
android:hint="Password"
android:inputType="textPassword"
android:textColorHint="@color/colorGrey"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/l_userNameTxt"
app:layout_constraintVertical_bias="0.011" />

```

<Button

```

android:id="@+id/l_loginBtn"
android:layout_width="162sp"
android:layout_height="50sp"
android:layout_marginStart="50dp"
android:layout_marginTop="40dp"
android:background="@drawable/button_green"
android:fontFamily="@font/sf_pro_display_regular"
android:text="@string/login"
android:textColor="@color/colorWhite"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/l_passwordTxt"
app:layout_constraintVertical_bias="0.0" />

```

<TextView

```

android:id="@+id/l_youAreNotTxt"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="50dp"
android:layout_marginLeft="50dp"
android:layout_marginTop="35dp"
android:layout_marginBottom="50dp"
android:fontFamily="@font/sf_pro_display_regular"
android:text="@string/you_are_not_a_member"
android:textColor="@color/colorGrey"
android:textSize="16sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/l_loginBtn"
app:layout_constraintVertical_bias="0.0" />

```

<TextView

```

android:id="@+id/l_registerTxt"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="4dp"
android:layout_marginTop="35dp"
android:layout_marginBottom="50dp"
android:fontFamily="@font/sf_pro_display_bold"
android:text="@string/register"

```

```
android:textColor="@color/colorGreen"
android:textSize="16sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toEndOf="@+id/l_youAreNotTxt"
app:layout_constraintTop_toBottomOf="@+id/l_loginBtn" />

</androidx.constraintlayout.widget.ConstraintLayout>
</ScrollView>
```

Figure B.9 LoginScreen.xml

Appendix C

C.1 User Guide – Mobile Application

1.0 General Information

The General Information section explains in general terms the Smart-Bin mobile application overview and the sections of the user manual.

1.1 Application Overview

Smart-Bin application is for the automated waste collection process to improve the effectiveness of the waste collection process. The application will be provided with an innovative mobile solution to keep the environment clean. Clients can send requests to collect waste and relevant transporters come to your doorstep.

1.2 Organization of the Manual

The user manual consists of the following four sections

1. General Information
2. System summary
3. Client User
4. Transport User

2.0 System Summary

System Summary section explains the hardware and software requirements for accessing Smart-bin application and user access levels.

2.1. Hardware & Software Requirements

Requires a smartphone with Android operating system (OS) The minimum android version should be 4.0.3 and up to avail all the features in the application.

To download and use the functionalities of the Smart-bin mobile app, you require an Internet connection on your mobile.

2.2. User Access Levels

1. Client User
2. Transport User

2.2.1. Client User Privileges

Following features can be availed by the client user.

- Sign Up for the application
- Client Profile
- Send New Request
- My Requests
- Cancel Request
- Rating Transporter
- About Us
- FAQ

2.2.2. Transport User Privileges

Following features can be availed by the transporter user.

- Transporter Profile
- Job Request
- Job Cancel
- About Us
- FAQ

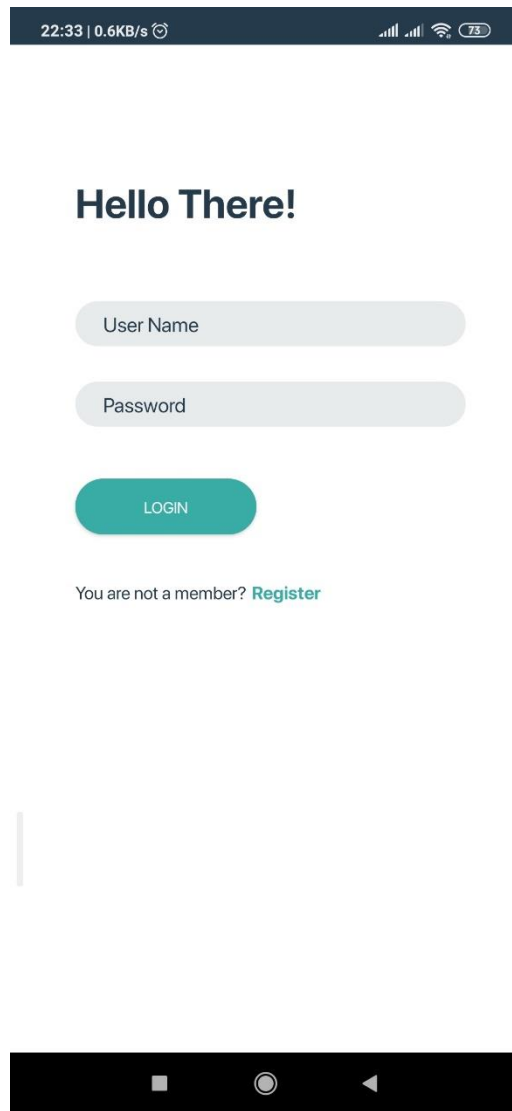
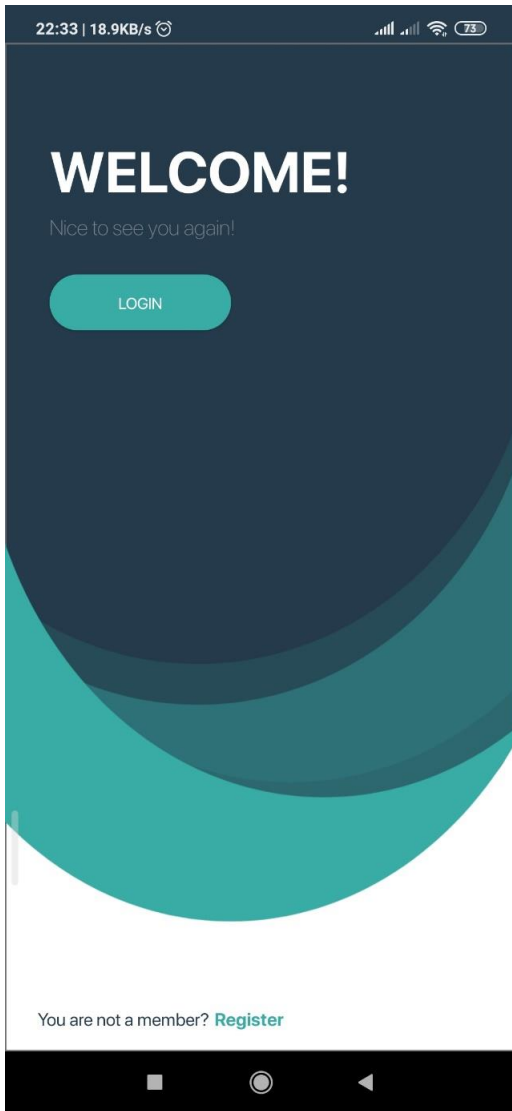
3.0. Sign In

There are two logins of the “Smart Bin” mobile application.

- Transporter login
- Client login

The user is expected to enter the following details to be allowed access to the Smart Bin mobile application System functions. All these fields are mandatory inputs and will be validated by the system.

- Username
- Password



- Username – Enter username
- Password – Enter password

Enter username and password and click on the login button. The application will route the transporter or client menu according to the login.

4.0. Client Registration

This screen helps enter sign up details of clients. The following fields should be completed to enter data into the system.

- First Name
- Last Name
- Username
- Password
- Confirm Password
- Location

22:39 | 5.0KB/s

Create Your Account

kamal

kamal100

.....

.....

7/, Vijayarama Rd, Gampaha, Sri Lanka

REGISTER

RESET

You are member? [Login](#)

- First Name - Enter First Name
- Last Name – Enter the Last Name
- Username -Enter Username
- Password - Enter Password
- Location – Click on the location button and pick up the location from google map

The client can register to the system after entering these details and click on the “Register” button.

4.1. How to Change Client Profile

Menu>Profile

The following fields are on the page.

- Reg No
- First Name
- Last Name
- Address
- Contact No
- Email
- Username
- Password
- Location



C-008

Inoka

peris

26 Bauddhaloka Mawatha, Gampaha, Sri Lanka

773899457

saman45@gmail.com

inoka

.

26 Bauddhaloka Mawatha, Gampaha, Sri Lanka 

UPDATE PROFILE

- Reg No - Non-editable field
- First Name – Editable field
- Last Name - Editable field
- Address - Editable field and the client can add address details
- Contact No - Editable field and the client can add contact no details
- Email - Editable field and the client can add email details
- Username - Editable field
- Password - Editable field
- Location - Editable field

4.2. New Request

Menu>New Request

This screen helps the client to send a request for the collect waste. The following fields should be completed to enter data into the system.

- Request Type
- Waste location
- Waste category
- Location
- Do you want to change pickup location
- Pickup Location
- Request Date

The screenshot shows the 'New Request' form with the following fields and values:

- Request Type:** Single
- Waste location:** Commercial
- Waste category:** Glass (with an 'ADD' button next to it)
- Location:** Metal (with an 'X' icon to remove it)
- Do you want to change pickup location:** No (toggle is off)
- Pickup Location:** finite pvt ltd (with a location pin icon)
- Request Date:** 04 May 2025

Buttons at the bottom: SEND REQUEST (teal), RESET (grey).

The screenshot shows the 'New Request' form with the following fields and values:

- Request Type:** Schedule
- Waste location:** Residential
- Waste category:** Plastic (with an 'ADD' button next to it)
- Location:** Glass and Plastic (each with an 'X' icon to remove it)
- Do you want to change pickup location:** Yes (toggle is on)
- Pickup Location:** no.5, kanthi road, gampaha (with a location pin icon)
- Request Date:** 01 May 2020

Buttons at the bottom: SEND REQUEST (teal), RESET (grey).

- Request Type – Select request type
- Waste location – Select waste location
- Waste category – Select waste category or
- Location – By default set resisted location
- Do you want to change pickup location – Click if need to change pickup location
- Pickup Location – Click on the button to select pick up location from google map.
- Request Date- Enter Request date

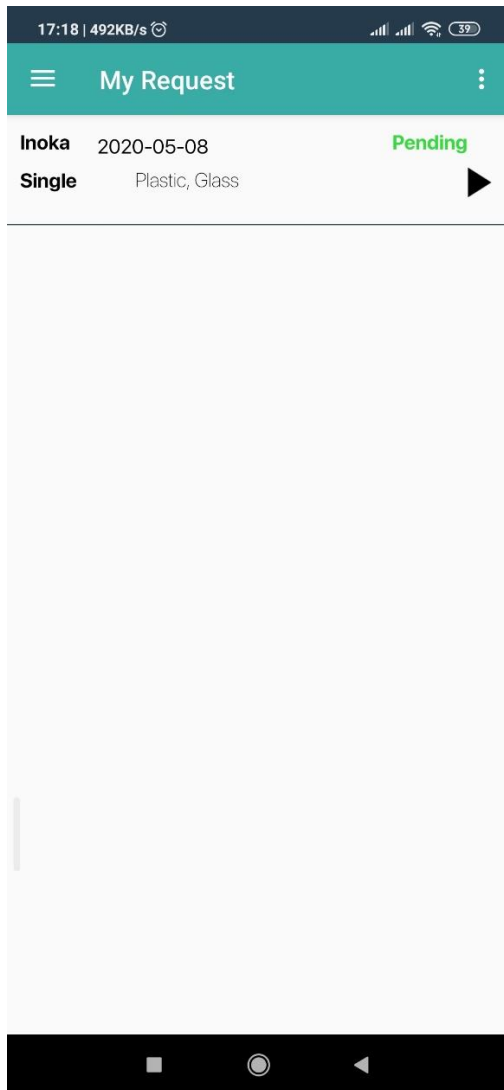
If the user selects the request type as “Scheduled” Request Date from and Request Date To fields are visible.

The client can add these details and click on the “Send Request” button.

4.3. My Request

Menu>My Request

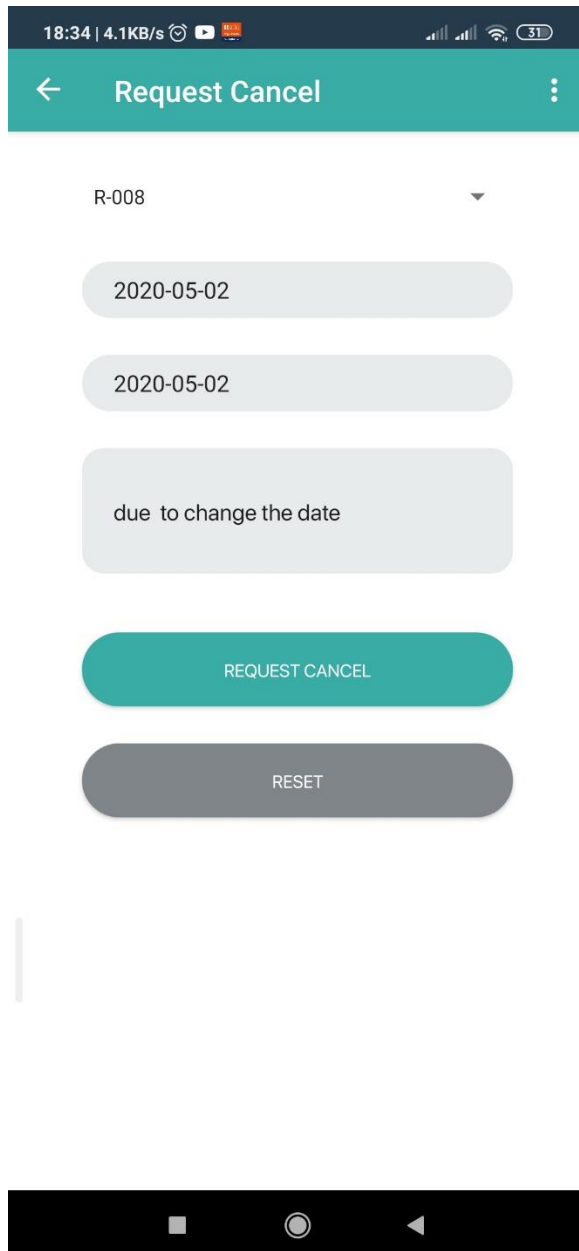
This page helps to user to view requests which are sent to collect waste. To view details of the request, click on the request.



5.0. Request Cancel

This screen provides support for the cancel requests. The following fields should be completed to enter data into the system.

- Request No
- Request From Date
- Request To Date
- Reason



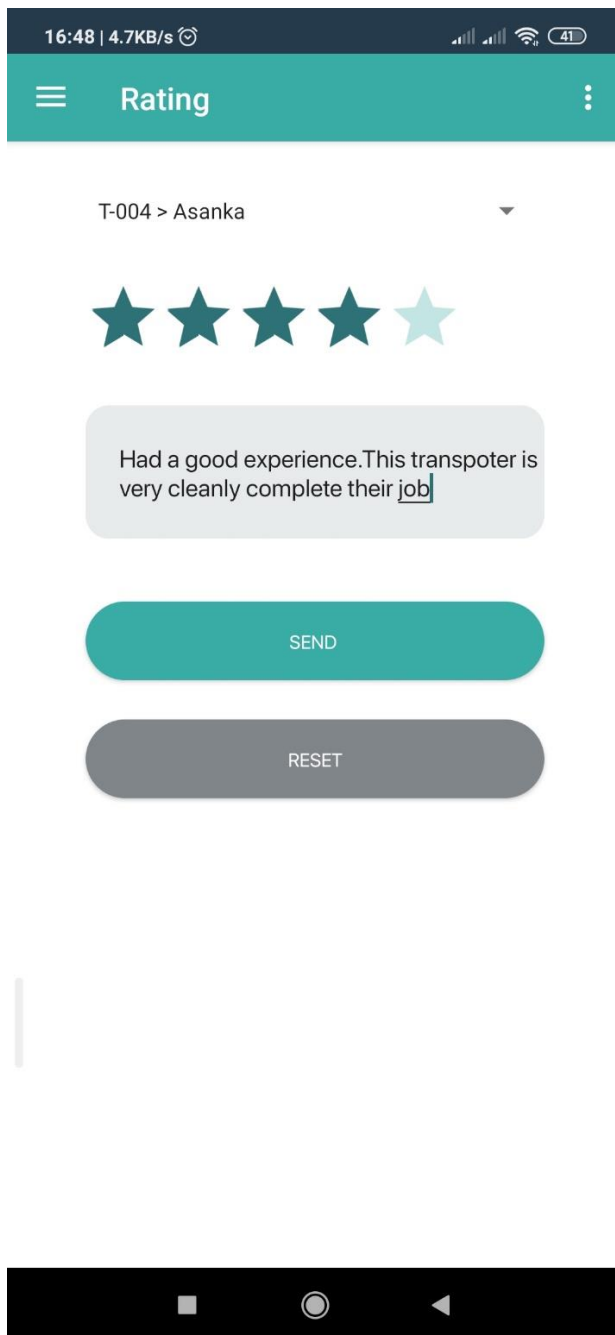
- Request No – Select request no
- Request from Date – By default set according to the request no
- Request to Date - By default set according to the request no
- Reason – Enter a reason for the cancel

5.0. Rate Transporter

This screen provides support for the rate transporter against client requests. The following fields should be completed to enter data into the system.

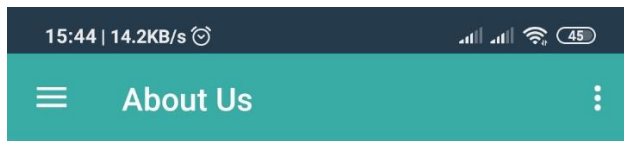
- Transporter
 - Comment
-
- Transporter - Select transporter
 - Comment – Enter comment

The client can add rating details as after adding the above details click on the “Send “button.



5.0. About Us

This page describes the application.



we offer an innovative mobile solution to protect our environment to keep clean the city,you can send the request to collect garbage and relevant organization will guide the relevant transporter to collect the garbage to your doorstep.

Thank You

Version 1.0.0



6.0. FAQ

This page provides some sample questions relates to the application.



What is the request limit of the send request?

No limit. You can send any request per day as a single request or schedule request.

What if I have forgotten my password?

If forgot password sholud be contact service team.

Can I Cancel Request ?

Transporter can be cancel accepted request due to valid reason. Also Client can be cancel request in the pending status.

2.0. Transporter Application

2.1. Transporter Profile

This page provides the facility to edit details of the Profile of the transporter. The following fields are on the page.

- Reg No
- First Name

- Last Name
- Address
- Contact No
- Email
- Username
- Truck No
- Password

- Reg No – Non-editable field
- First Name – User can update the field.
- Last Name – User can update the field.
- Address - The user can update the field.
- Contact No - The user can update the field.
- Email - The user can update the field.
- Username - The user can update the field.
- Truck No - The user can update the field.
- Password - The user can update the field.



T-003

Asanka

Pathum

Address

785206320

asanka.pethm@gmail.com

asanka

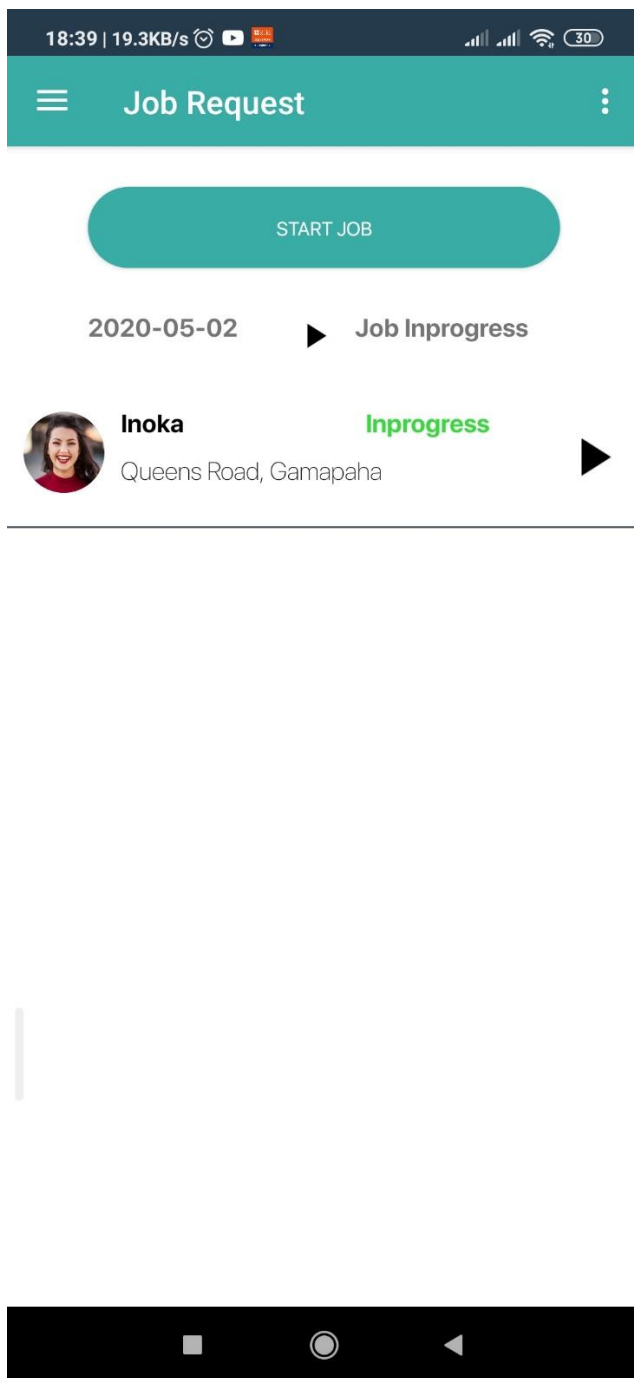
.

UPDATE PROFILE

2.2. Job Request

This page illustrates all job requests that users are sent. The following details are on the page.

- Start Job button – Transporter can start his job by click on this button.
- End Job button - Transporter can end his job by click on this button. This button gets display after the transporter has started his job.
- List of Job requests – Transporter can get details to the view of the request after select request

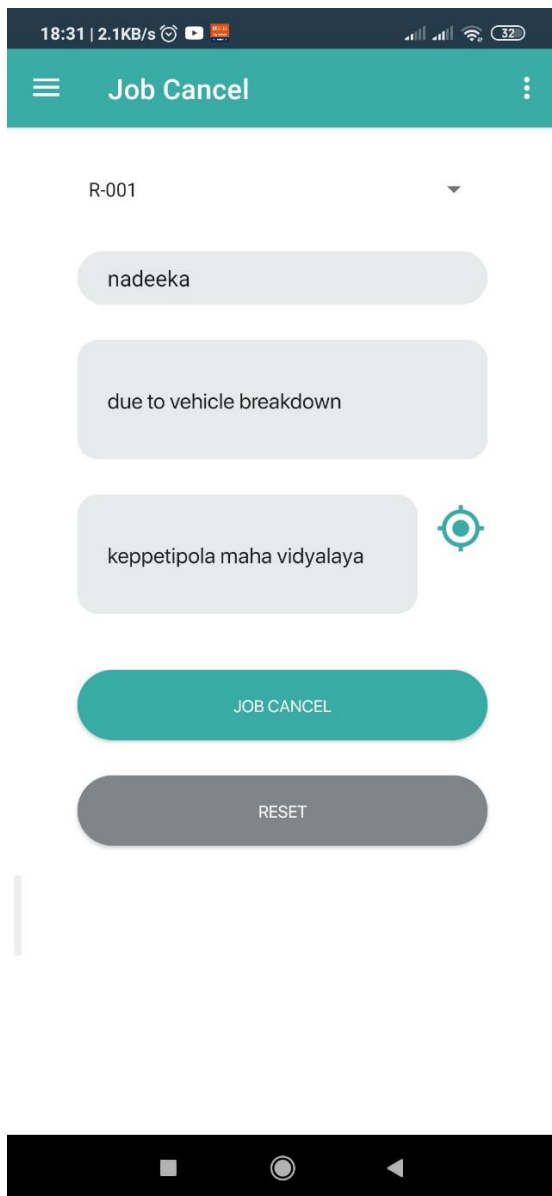


2.3. Cancel Job Request

This page provides a facility for the transporters to cancel requests. the following details are on the page.

- Request No
- Client

- Pick up location
- Reason



- Request No – Select request no
- Client – By default get selected after user select request no
- Pick up location – By default get selected after user select request no
- Reason – Enter a reason

3.0. Notifications

The notifications feature can be accessed through the Navigation Drawer. On click of the Notifications option, the recent notifications received to the mobile will be listed. On click of the notification, the notification details will be shown.

C.2 User Guide – Web Application

1.0. General Information

General Information section explains in general terms of the WMS overview and the sections of the user manual.

1.1. Application Overview

Waste Monitoring System is for the automated waste collection process to improve the effectiveness of the waste collection process. The application will be provided for innovative web solutions for the organizations to manage their waste collection process.

1.2. Organization of the Manual

The user manual consists of the following four sections

1. General Information
2. System summary
3. Administrator user

2.0 System Summary

System Summary section explains the hardware and software requirements for accessing the WMS web application and user access levels.

2.1. Hardware & Software Requirements

Requires a smartphone with Android operating system (OS) The minimum Android version should be 4.0.3 and up to avail all the features in the application.

To download and use the functionalities of the Smart-bin mobile app, you require an Internet connection on your mobile.

2.1.1. Web application - Hardware Requirements

- Operating System: Window 8 or later (x86 or x64)
- Processor: Intel Core 2 Duo (2GHz) or later
- RAM: 4GB
- HDD: 1GB

- Network: Broadband Internet connection

2.1.2. Web application - Software Requirements

- Web Browser: Chrome, Firefox, Microsoft Edge
- Web Server: Internet Information Service (IIS)
- Database server: MSSQL

2.2. User Access Levels

1. Administrator

2.2.1. Administrator User Privileges

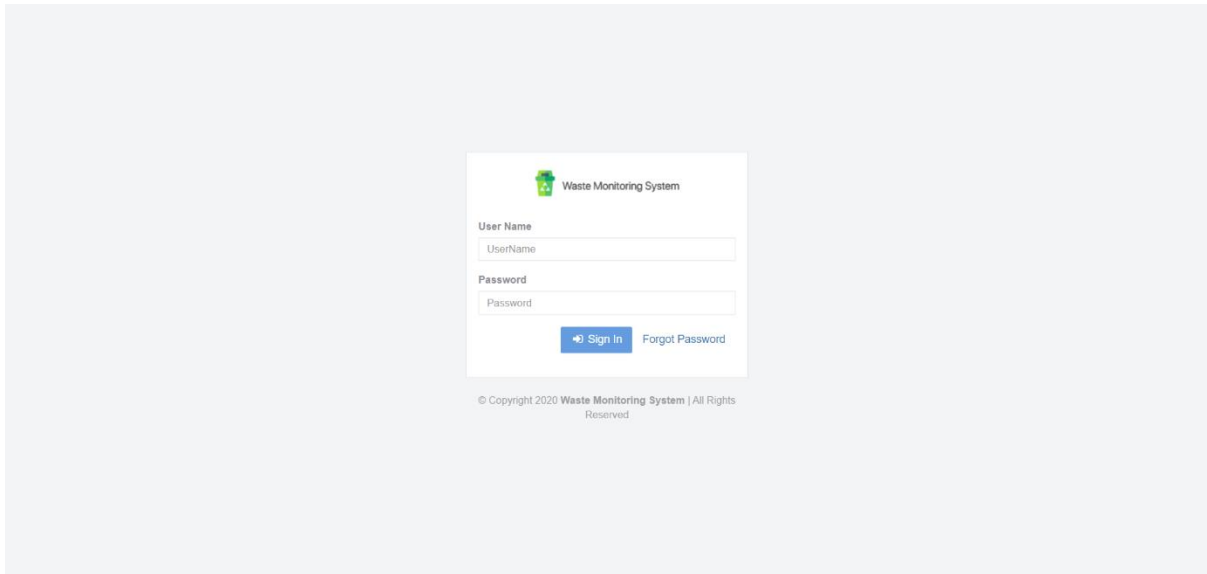
Following features can be availed by the administrator

- Transporter registration
- Buyer registration
- Define waste categories
- Map waste categories with transporters.
- Maintain recycle details
- Manage buyer purchase
- Send emails to buyers
- Track transporter
- Rating transporter
- Reporting
- Dashboard

3.1. Sign In

The user is expected to enter the following details to be allowed access to WMS system functions. All these fields are mandatory inputs and will be validated by the system.

- Username
- Password



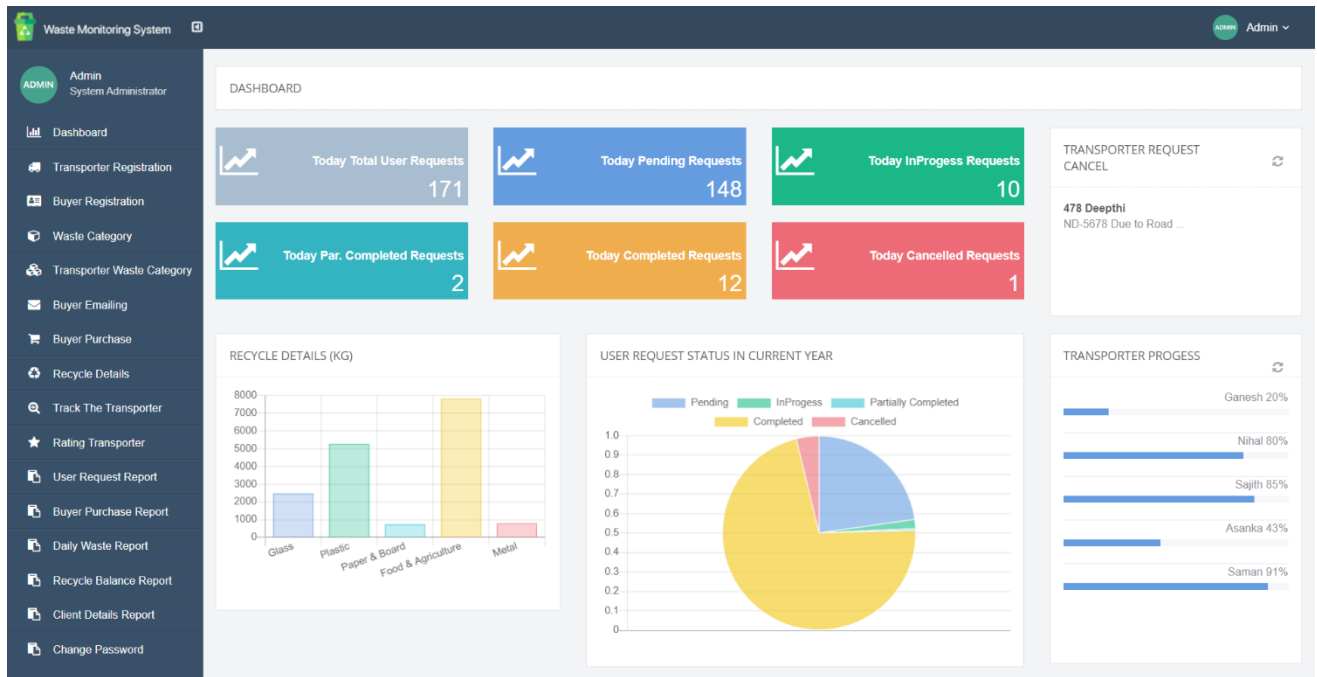
- Username – Enter username
- Password – Enter password

Enter username and password and click on the login button. The application will route to the dashboard of the system.

3.2. Dashboard

The dashboard visualizes the overall picture of the waste management process. The dashboard consists of the following items.

- All Client requests for the current date
- Pending requests for the current date
- In Progress requests for the current date
- Partial Completed requests for the current date
- Completed requests for the current date
- Canceled requests for the current date
- Transporter canceled requests
- Recycle details
- Client requests status in the current year
- Transporter progress



3.3. Transporter Registration

This screen facilitates for the register transporters to the system. The following details are on the page.

- Reg No
- First Name
- Last Name
- Contact No
- Email
- Truck No
- Username
- Password
- Confirm Password
- Active Status

The screenshot displays the 'Waste Monitoring System' Admin interface. The left sidebar contains navigation options such as Dashboard, Transporter Registration, Buyer Registration, Waste Category, and various reports. The main content area is titled 'TRANSPORTER REGISTRATION' and features a form with the following fields: Reg No (pre-filled with T-005), First Name, Last Name, Contact No, Email, Truck No, User Name, Password, and Confirm Password. There is also an 'Active' checkbox and 'Reset' and 'Save' buttons. Below the form is a table titled 'TRANSPORTERS' with columns for Edit, Reg No, First Name, Last Name, Contact No, Email, Truck No, and Status. The table contains four entries for transporters T-001 through T-004.

Edit	Reg No	First Name	Last Name	Contact No	Email	Truck No	Status
<input checked="" type="checkbox"/>	T-001	Nihal	Perera	774502560	nihal.perera@gmail.com	ND-5678	true
<input checked="" type="checkbox"/>	T-002	Sajith	Madusanka	774502560	sajith.1990@yahoo.com	SP-5485	true
<input checked="" type="checkbox"/>	T-003	Asanka	Pathum	785206320	asanka.pethm@gmail.com	WP-4520	true
<input checked="" type="checkbox"/>	T-004	Saman	Kumara	771284152	samank@gmail.com	WP-4952	true

- Reg No – System generate a number
- First Name – Enter the first name
- Last Name – Enter the last name
- Contact No – Enter contact no
- Email - Enter email address
- Truck No – Enter truck no
- Username – Enter username
- Password – Enter password
- Confirm Password – Re-enter password
- Active Status- Mark status

3.4. Buyer Registration

This screen provides support for the registered buyers of the system. The following details are on the page.

- Reg No

- First Name
- Last Name
- Contact No
- Email
- Representative name
- Representative contact no
- Waste categories
- Order Level
- Active status

The screenshot displays the 'Waste Monitoring System' Admin interface. The left sidebar contains navigation options such as Dashboard, Transporter Registration, Buyer Registration, Waste Category, and others. The main content area is divided into two sections: 'BUYER REGISTRATION' and 'BUYERS'.

BUYER REGISTRATION Form:

- Reg No:** B-008
- First Name:** First Name
- Last Name:** Last Name
- Contact No:** Contact No
- Email:** Email
- Representative Name:** Representative Name
- Representative ContactNo:** Representative ContactNo

WASTE CATEGORIES Table:

Select	Waste Category Name	Order Level	Order Unit
<input type="checkbox"/>	Glass	0	Kg
<input type="checkbox"/>	Plastic	0	Kg
<input type="checkbox"/>	Paper & Board	0	Kg
<input type="checkbox"/>	Food & Agricultural waste	0	Kg
<input type="checkbox"/>	Metal	0	Kg

Active

BUYERS Table:

Edit	Reg No	First Name	Last Name	Contact No	Email	Representative Name	Representative Contact No	Status
<input checked="" type="checkbox"/>	B-001	Nihal	Silva	778702530	ruchirakmal@gmail.com	Emil Perera	778506025	true
<input checked="" type="checkbox"/>	B-002	Nishan	Perera	782188085	nishannsb@gmail.com	Sandun Perera	782186052	true
<input checked="" type="checkbox"/>	B-003	Vinushi	Wickramasinghe	785651025	vinushiranshila@gmail.com	Kalana Perera	782456325	true
<input checked="" type="checkbox"/>	B-004	Kamal	Vithanage	778704520	kamak.s@yahoo.com	Gayani Perera	784502056	true
<input checked="" type="checkbox"/>	B-005	Saman	Perera	778654152	saman@gmail.com		4555	true

- Reg No - System generate a number
- First Name- Enter the first name
- Last Name - Enter the last name
- Contact No - Enter contact no
- Email - Enter email address
- Representative name - Enter a representative name
- Representative contact no - Enter representative contact no
- Waste categories – Select waste categories
- Order Level- Enter order level
- Active status – Select status

3.5. Waste Category

This screen provides support for the defined waste categories. The following details are on the page.

- Waste category code
- Waste category name
- Description
- Status

The screenshot displays the 'Waste Monitoring System' interface. The top navigation bar shows 'Admin' and 'System Administrator'. The sidebar menu includes options like 'Dashboard', 'Transporter Registration', 'Buyer Registration', 'Waste Category', 'Transporter Waste Category', 'Buyer Emailing', 'Buyer Purchase', 'Recycle Details', 'Track The Transporter', 'Rating Transporter', 'User Request Report', 'Buyer Purchase Report', 'Daily Waste Report', 'Recycle Balance Report', 'Client Details Report', and 'Change Password'.

The main content area features a 'WASTE CATEGORY' form with the following fields:

- Waste Category Code: WC-007
- Waste Category Name: Waste Category Name
- Description: Description
- Active:

Buttons for 'Reset' and 'Save' are located at the bottom of the form.

Below the form is a table listing existing waste categories:

Edit	Waste Category Code	Waste Category Name	Description	Status
	WC-001	Glass	Glass	true
	WC-002	Plastic	Plastic	true
	WC-003	Paper & Board	Paper & Board	true
	WC-004	Food & Agricultural waste	Food & Agricultural waste	true
	WC-005	Metal	Metal	true

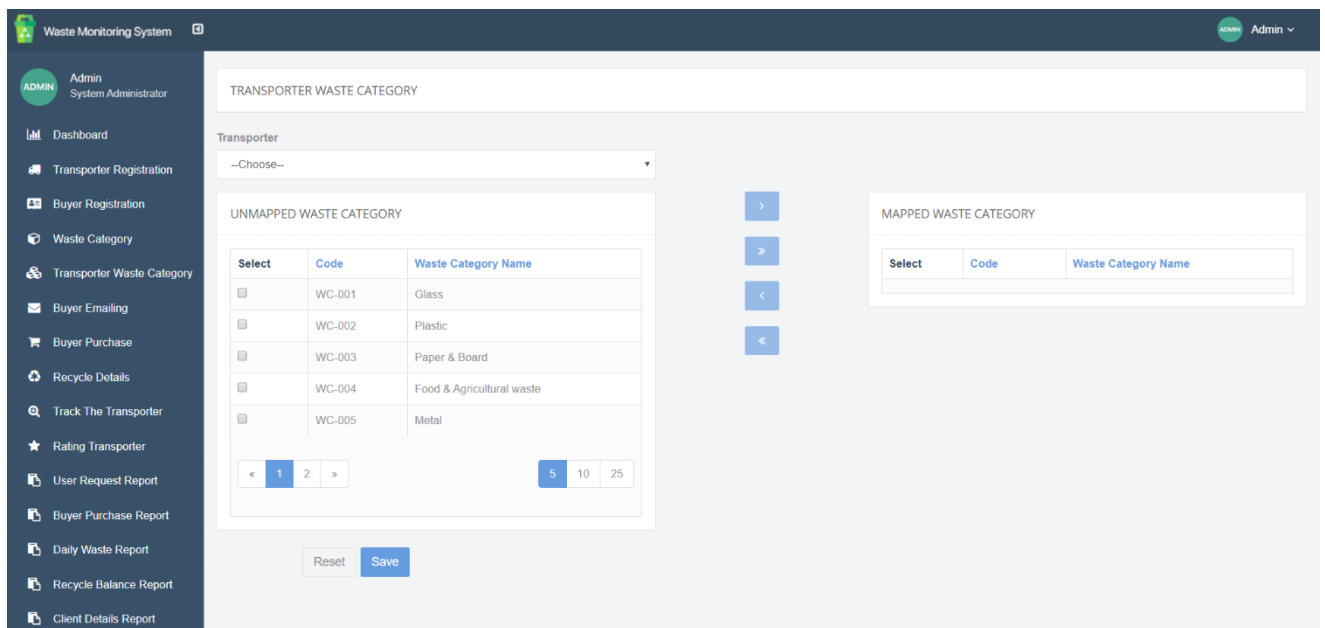
At the bottom of the table, there are pagination controls showing page 1 of 2, and a status filter set to 5.

- Waste category code – System generates code.
- Waste category name – Enter a name for the waste category
- Description – Enter description
- Status – Select status

3.6. Transporter Waste Category

This screen facilitates map waste categories with transporters. The following details are on the page.

- Transporter
- Left grid - Unmapped waste categories
- Right grid – Mapped waste categories

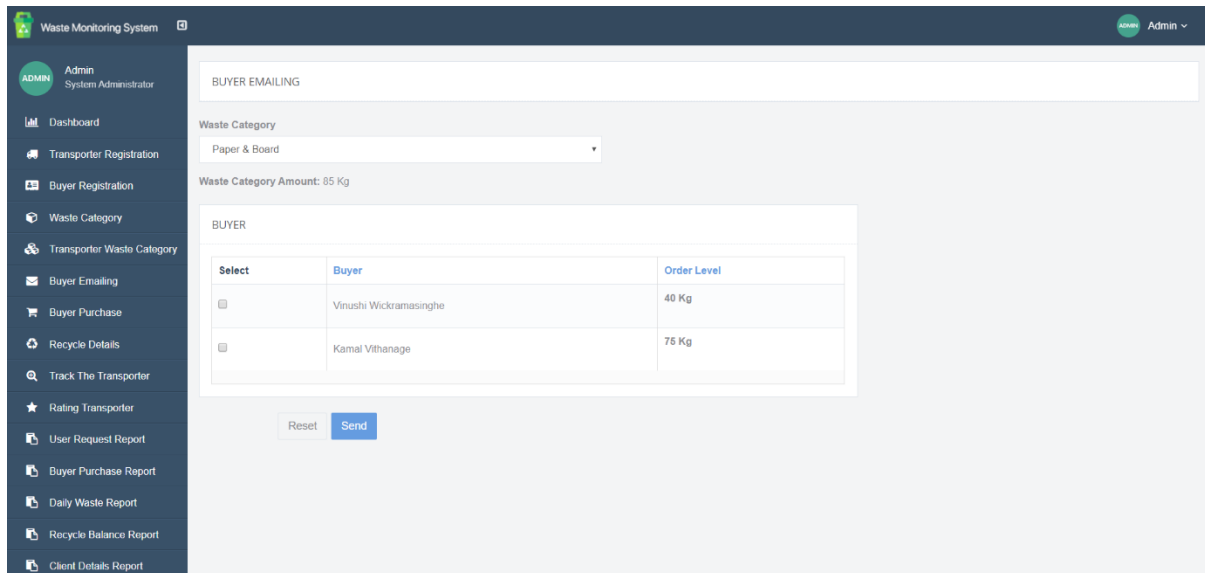


- Transporter – Select transporter
- Left grid - Unmapped waste categories – Select waste categories
- Right grid – Mapped waste categories – Mapped waste categories
- > Arrow – Move waste category from left grid to right grid
- < Arrow – Move waste category from right grid to left grid
- >> Arrow - Move waste categories from left grid to right grid
- <<Arrow - Move waste category from right grid to left grid

3.7. Buyer Emailing

This screen facilitates to send emails to buyers when their order level has reached. The following details are on the page.

- Waste Category
- Buyers list with order level

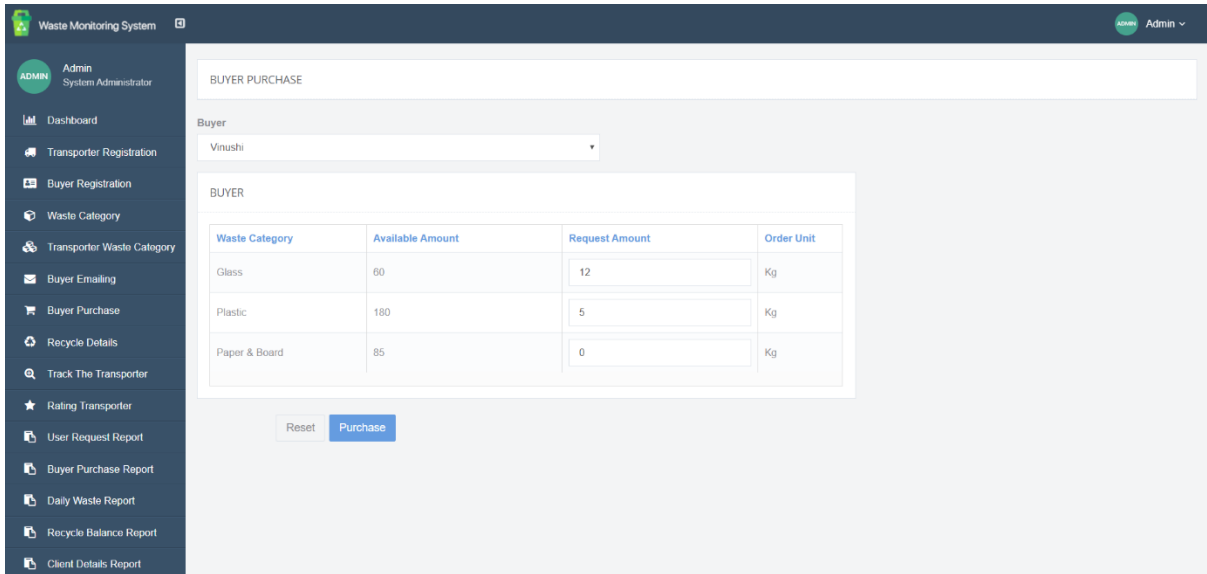


- Waste Category – Select waste category from the dropdown list
- Buyers list with order level – Select buyers to send emails

3.8. Buyer Purchasing

This screen provides support for manage details of the purchase of the buyers. The following details are on the page.

- Buyer name
- Waste categories with details (Available amount/Order unit)



- Buyer name – Select buyer name
- Request amount – Enter request amount

3.9. Recycle Details

This screen facilitates to manage to recycle details. An administrator can enter the number of recycling items into the system by using this screen. The following details are on the page.

- Truck no
- Transporter
- Waste category details (waste category, weight, order unit)

The screenshot shows the 'RECYCLE DETAILS' page in the Waste Monitoring System. The sidebar on the left contains the following menu items: Admin (System Administrator), Dashboard, Transporter Registration, Buyer Registration, Waste Category, Transporter Waste Category, Buyer Emailing, Buyer Purchase, Recycle Details, Track The Transporter, Rating Transporter, User Request Report, Buyer Purchase Report, Daily Waste Report, Recycle Balance Report, and Client Details Report. The main content area includes a 'RECYCLE DETAILS' header, a 'Truck No' dropdown menu with 'ND-5678' selected, and a 'Transporter' dropdown menu with 'Sajith' selected. Below this is a 'WASTE CATEGORIES' table with the following data:

Select	Waste Category Name	Weight	Order Unit
<input type="checkbox"/>	Plastic	0	Kg
<input type="checkbox"/>	Paper & Board	0	Kg

At the bottom of the table, there are 'Reset' and 'Save' buttons.

- Truck no – Select truck no from the dropdown list
- Transporter – Automatically select according to the truck no
- Weight – Enter the weight of the waste category

3.10. Track Transporter

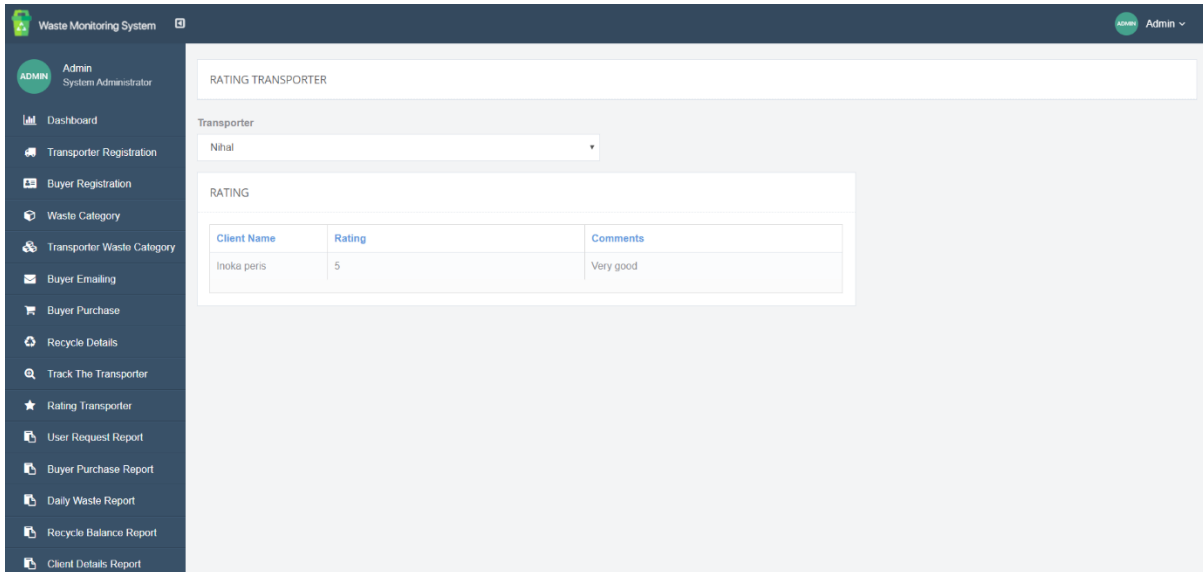
This screen facilitates the track transporter. It provides a view of the current location and its status of the transporter. The following details are on the page.

- Transporter Name
- Truck No
- Current Location
- Status

3.11. Rating Transporter

This screen supports the view ratings of the transporter send by the clients. The following details are on the page.

- Transporter Name
- Client Name
- Rating
- Comments



- Transporter Name – Select name from the dropdown list
- Client Name – Display according to the selected transporter
- Rating – Rating details send by the clients. It Displays according to the selected transporter
- Comments - Comment details send by the clients. It Displays according to the selected transporter

3.12. Reports

This section facilitates the generate reports. Following reports are provided.

- Client request report
- Buyer purchase report
- Daily waste report
- Recycle balance report
- Client details report

Waste Monitoring System Admin

ADMIN
System Administrator

- Dashboard
- Transporter Registration
- Buyer Registration
- Waste Category
- Transporter Waste Category
- Buyer Emailing
- Buyer Purchase
- Recycle Details
- Track The Transporter
- Rating Transporter
- User Request Report
- Buyer Purchase Report
- Daily Waste Report
- Recycle Balance Report
- Client Details Report

BUYER PURCHASE REPORT

From Date

To Date