SMART SYSTEM FOR WASTE MANAGEMENT

P.D. Gunasekara

2021

SMART SYSTEM FOR WASTE MANAGEMENT

A dissertation submitted for the Degree of Master of Information Technology

P.D. Gunasekara

University of Colombo School of Computing
2021

UCSC

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: P.D. Gunasekara

Registration Number: 2018/MIT/023

Index Number: 18550239

P. Gunasekara

Signature: Date: 12/09/2021

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

Ms. P.D. Gunasekara

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

Certified by:

Supervisor Name: Dr. B.H.R. Pushpananda

Signature: Date:

ABSTRACT

Sri Lanka as a country with developing economy, experience an exhausted waste collection service and inadequately managed solid waste collection process. Waste management is an ongoing process and it is a challenge with rapid urbanization. furthermore problem is worsening with improper and inefficient systems.

The traditional daily door-step collection system has many downsides. Physically going to the households and checking waste levels unnecessarily consume labor power, fuel and time as some bins may not need emptying. Therefore, this project aims at implementing a web application to maintain garbage bins, monitor garbage collection and to effectively communicate with customers and truck drivers. Current stage of the system is focused on collecting solid garbage types like glass, plastic, rubber, e-waste, paper, etc. It incorporates a Raspberry Pi module which automatically updates the garbage bin status in the system. The incremental development model was used during the lifecycle of development and the system has been implemented using the technologies such as PHP and MYSQL.

The system had been implemented to achieve the functional requirements identified. The system was tested using black box testing approach as mentioned in the initial test plan. All high priority test cases had been executed and bugs are fixed.

The implemented system provides a platform for communication, customer management, garbage management, schedule management for staff, drivers and customers.

ACKNOWLEDGMENT

I would like to extend my gratitude to many people who helped to bring this project to fruition. First and foremost I wish to thank my supervisor, Dr. B.H.R. Pushpananda. I am so deeply grateful for his help, professionalism and valuable guidance throughout this project.

My thanks also goes to Thilan Weerasinghe, Chanduni Wickremasinghe for been good friends and helping me with various aspects of implementing this system.

Finally, I must express my profound gratitude to my sister Ms. N.D. Gunaseakara and family for providing me with unfailing support and continuous encouragement throughout my years of study. This accomplishment would not have been possible without them. Thank you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

- 1NF - First Normal Form
- 2NF - Second Normal Form
- 3NF - Third Normal Form
- AJAX - Asynchronous JavaScript and XML
- CSS - Cascading Style Sheet
- GPS - Global Positioning System
- HTML - Hypertext Markup Language
- IoT – Internet of Things
- MVC - Model View Controller
- OOP - Object-oriented programming
- PHP - PHP: Hypertext Preprocessor
- ROI – Return of Investment
- TCP/IP - Transmission Control Protocol/Internet Protocol
- UML – Unified Modeling Language
- URL - Uniform Resource Locator

INTRODUCTION

Waste will exist on earth till human beings live on earth. Since the early days various waste disposal methods were practiced by the individuals but with the urbanization it has become a responsibility of the government or private authorities. In urban areas collecting waste by government or private authority is more common than in rural areas. Improper waste disposal methods cause many environmental and health hazards. Thus, implementing proper waste management systems using emerging technologies reflects the development of a country.

1.1 STATEMENT OF THE PROBLEM

Sri Lanka as a country with developing economy finds it challenging to impose a financially, environmentally and socially acceptable waste management solution. Sri Lanka generates 7000 Metric Ton of solid waste per day but only half of that waste generated is collected (Admin, 2017).

Sri Lankan citizens practice many waste disposal methods such as dispose to community bins, burn, burying in the garden, dump by the road side and hand over to public or private organizations. Dumping waste by the road sides forced the dwellers to live in an unhygienic and unhealthy environment. Moreover it contaminates soil and attracts disease carrying pets. Burning the waste containing plastic and heavy metals pollutes the air and causes extreme climate changes in the long run.

Because of the downsides in the traditional waste management system, a systematic process should be implemented for efficient waste collection.

1.2 MOTIVATION FOR THE PROJECT

The traditional door-step waste collection system has many downsides. The collectors collect garbage according to a pre-schedule so they visit each and every household to check the waste levels of the bins. That wastes fuel, time and consume labor power to visit bins which do not need emptying. On the other hand, if the bins are overloaded they attract animals and create unhygienic environment which helps to spread diseases.

The citizens have to offer an incentive to garbage collectors or in some occasions solid waste traders collect waste by giving so little money to citizens. Either way the citizens are dissatisfied and will not readily collect and handover inorganic waste. That increases the solid waste generated in cities that are inaccessible.

Ineffective waste management swallow considerable amount of city's budget. The current waste collection system includes routine garbage trucks doing rounds daily or weekly which waste both time and fuel as they don't have prior idea of the places that need visits. Currently about Rs. 75,000.00 is paid per a lorry that can transport 10 tons of garbage (Anon., 2019). Without route optimization sometimes, they may not cover every zone of the city.

Therefore, as a solution a proper waste management system is needed for the benefit of the citizens as well as for the authority.

1.3 OBJECTIVE OF THE SYSTEM

The proposed system is intended to meet the following objectives:

1. Establishing a centrally administered system to monitor garbage collection
2. Enhance the communication with citizens and garbage truck drivers
3. Allow easy updating of garbage bin details
4. Encourage citizens to properly dispose waste
5. Decrease cost and time of garbage collection
6. Obtain feedback from citizens to enhance the service
7. Provide efficient standard reporting and ad hoc reporting capabilities

1.4 BACKGROUND OF THE STUDY

Different countries follow different strategies for waste management. Most countries follow door to door waste collection mechanism in their own way. In Italy door to door waste collection was initiated in 2012. Waste collection truck visited households during less traffic times of the day. They have used various methods for public awareness such as a free computing application, newspaper and radio plus TV advertising. Singapore is practicing innovative concepts such as 'cash for trash', 'school recycling corner programs' to promote waste management among citizens (Ong, 2015). With Singapore's 'zero waste nation' vision they expect to harness ICT, such as data analytics, to inform policy formulation and infrastructure planning in waste management and resource recycling (MSE, 2021). In Netherland citizens dump residual mixed waste to collection points and segregated waste materials are collected door to door. Netherlands waste management company Mic-O-Data and Vodafone and has a 30% share of the Netherlands' waste management market and they have fitted bins with a Vodafone M2M SIM that sends a daily status signal to the refuse collectors, alerting them when a bin is getting full or has not been closed properly (Jordan, 2012). Current systems in India cannot cope with the volumes of waste generated by the increasing urban population thus urban India is the world's 3rd largest garbage generator and by 2050 waste is expected to rise to 436 million tons up (Bhatia, 2016). So They are using different waste management mechanisms and also taking the leads in experimenting on IoT based smart waste collection systems.

Sri Lanka is improving on waste management by introducing acts, distributing the responsibility to different authorities, building sanitary landfills and establishing municipal solid waste composting facilities. Sri Lankan government has launched 'Pilisaru' project (Dassanayake, 2011) for effective solid waste management. In Sri Lankan context although research have been done on smart systems for solid waste management, according to author's knowledge none have been implemented yet. ICT is used as a mode of public awareness thus municipal council web sites and social media is used to raise awareness on proper waste management.

1.5 SCOPE OF THE PROJECT

The scope of the project is to implement a web application for waste management. It handles segregation and collection process.

Main modules of the system:

- Garbage management module

Citizens can sign up and login to the system then request bins based on the types and sizes he needs. The request goes to the staff and he allocates bins to the customer. The garbage bins are delivered to the customers and they will login to the system and notify bins received.

When the garbage bin is 80% filled an automatic message is sent to the system through the IoT unit. The garbage collector weighs the garbage and updates the system. Based on that green points will be calculated based on a formula and the points will be added to the particular customer. The system provides reports such as garbage weigh reports according to customer/type/route, green points reports, bin allocation reports. The functionalities of the module are listed below.

  - Citizen registration
  - Garbage bin allocation
  - Trash collection request
  - Green points calculation
  - Garbage weight reports
  - Segregation reports

- Notification module

As the garbage truck does not arrive as soon as the bin is full, a fixed threshold level of 80% had been set for which an automatic notification is sent to staff through the IoT unit. After a schedule is created driver and the customer receive notifications. As soon as the driver starts trip both the staff and the customer receive notifications. Thus the main notification types handled by the system are,

- Bin full notifications
- Trash collection schedule notifications

- Trash collection module

When the system receives adequate requests, the staff schedules a collection trip. The trips will be scheduled based on date or the garbage capacity whichever comes first. Drivers can register with the system. Therefore, for a trip on a particular date the route, driver and the vehicle is assigned by the staff. A list is generated with households the Trash should be collected. When the staff assigns a scheduled trip the customer and the driver receives notifications and the driver can accept or reject that. If he accepts the job he receives the list. The driver updates the system when he starts the trip. He weighs the trash and he is responsible for feeding those data to the system. The customers receive green points accordingly. Following is a summary of the tasks implemented in this module

- Driver registration
- Driver assignment
- Collection trip scheduling
- Driver task list report
- Route management

- e-Commerce module

The shopping cart contains eco-friendly home items listed with an image, description and required green points to buy item. Customers can add products to the shopping cart if they are available. They can check out the products then the total number of green points consumed will be displayed and they can add the shipping address. If they don’t have sufficient green points an error message will be displayed. On successful completion they receive a mail with order information.

- Complaint module

The customers can use this module to complain about drivers, bins, schedules or any other relevant matter. The staff can view all complaints and resolve them. The system generates reports based on complaint category, month and customer.

The system does not intend to handle waste disposal and waste recycling process. Furthermore, GPS based garbage truck tracking is not implemented in the current system.

1.6 FEASIBILITY STUDY

The main objective of a feasibility study is to determine whether a certain project plan is feasible. That is, whether it will work and whether it is worth economically.

1.6.1. TECHNICAL FEASIBILITY

Currently waste management process is handled manually by the government authorities. Although they use computers for some of the tasks they do not use ICT effectively. But in the new system 90% of process is going to be automated so it will be necessary to install more computers with network connection. Nowadays most people own smart phones so they can access the system without a hassle.

For the staff of the relevant authorities who are using the system a training session is needed, and the authorities should have a support team to handle issues of the smart waste bins.

1.6.2 ECONOMIC FEASIBILITY

The authorities can use their existing computers to use the system thus there will be less expenses to purchase hardware. There will be no cost for software as the system is implemented on a free platform.

By using the new web solution, the authorities can cut off the cost of unnecessary trips of garbage trucks to collect partially filled bins. As a result it will minimize the fuel cost and labor cost.

This system needs some more initial investment than the existing system as the IOT devices should be purchased. The cost of the Raspberry Pi module is considerably higher compared to other units used such as sensors and wires. So there is a possibility of using Arduino board instead of Raspberry Pi module to reduce the cost. Although an initial cost exists it can be justifiable as it will improve the quality of the service.

1.6.3 OPERATIONAL FEASIBILITY

In operational feasibility an assessment is done on the political and managerial environment the system is implemented and concerns on how the system will work when it is installed in the real environment. People are inherently resistant to change but as the proposed system facilitates day to day tasks of the authorities they may get accustomed to it. The new proposed system is very much useful to the citizens and there for it will be accepted by a broad audience from the country.

1.7 PROJECT SUMMERY

The requirements of the proposed system were identified by observing the traditional waste collection process. domain analysis was performed prior to finalizing the requirements. After studying several alternative solutions, it was decided that a web based application is the best possible solution. At first the functionalities were elaborated with UML diagrams then system was implemented mainly using PHP and other related technologies such as CSS, AJAX , jQuery with MYSQL as the Database Management Software. Finally, system was tested against the test plan using black box testing strategy and the user feedback was gathered through a questionnaire.

1.8 STRUCTURE OF THE DISSERTATION

Information on the project carried out is given in a comprehensive manner in the following chapters.

Chapter 2 - Discusses the requirements and brings out a requirement analysis with the details of currently available systems.

Chapter 3 - ‘Design Chapter’ is the core part of the project. It includes architectural design, UML design diagrams , presenting overall structure of the system .

Chapter 4 – ‘Implementation Chapter ‘ includes important code segments and module structures.

Chapter 5 - This chapter includes test plans which ensure the system would work properly. Also it contains a critical evaluation of the system.

Chapter 6 - Conclusion -Discusses results of the project. And also describes future work based on this specific project.

BACKGROUND

2.1 INTRODUCTION

The software requirements are description of features and functionalities of the target system. Requirement analysis helps to study the nature of the present situation and it gives a clear picture of the existing process and the weaknesses associated with the current practices. After identifying the functionalities to be implemented, by reviewing existing similar systems give an opportunity to improve the proposed system further. Thereafter among different design strategies, the best suited one can be selected.

2.2 ANALYSIS OF THE CURRENT PROCESS

Waste collection responsibility is vested with the local authorities in Sri Lanka. In highly populated areas like Colombo, the task is outsourced to private companies. In Sri Lanka, a door-to-door waste collection system is followed where the citizens are required to leave the waste bins outside their houses for the collectors who visit according to the garbage pickup schedule set by the municipality. For degradable waste multiple visits per week are scheduled whereas for recyclable waste weekly visits are scheduled. In some cities there are waste collection points with waste sorting bins to dump recyclable waste. Occasionally solid waste traders visit households to collect recyclable waste.

After domain analysis major activities of the waste collection process has been identified as Garbage Management, Garbage Collection and Communication. A detailed study had been carried out to get a deep understanding of each process.

2.2.1 GARBAGE MANAGEMENT

Citizens use their own bins for garbage collection. As they cannot get rid of garbage whenever they require usually they keep organic waste on bags till the garbage truck arrives. Mostly bins are not used for recyclable waste, they are kept outside the house till the trucks arrive or some may dump them on bins located at garbage collection points.

2.2.2 GARBAGE COLLECTION

Garbage trucks of the urban councils or municipal councils travel through main roads once a week or more depending on the population density of the area. Citizens keep garbage outside the houses in their bins on those particular days. The citizens who are not living along the roads where the truck travels, have to keep garbage on collection points. The citizens are required to separate organic waste , plastic and glass. To collect organic waste the trucks visit more occasionally than to collect recyclable waste. Two or three garbage collectors are assigned for each truck for door to door visits.

2.2.3 COMMUNICATION

Garbage collection days are informed through poster stands displayed by the road. Some may get to know about that through word of mouth. Few municipal councils such as CMC (Colombo Municipal Council) inform the schedule through their web site. There are no proper mechanisms to handle complaints so citizens complaint garbage collectors informally.

The UML diagram in Figure 2.1 depicts the functionalities of the existing system.

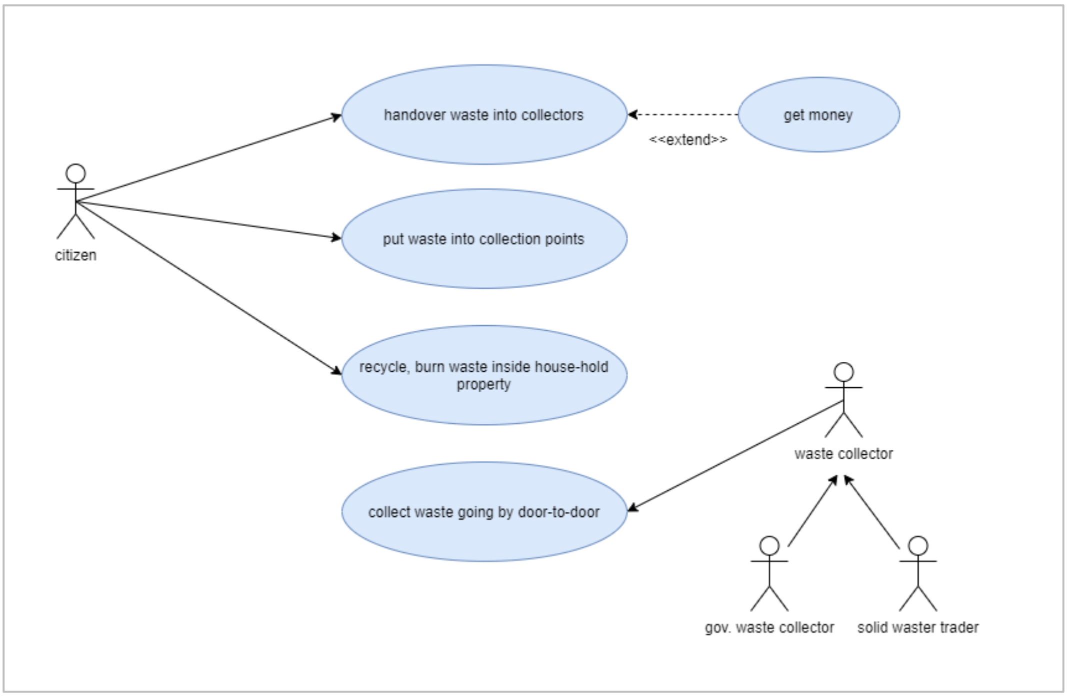

handover waste into collectors
<<extend>>
get money
citizen
put waste into collection points
recycle, burn waste inside house-hold property
waste collector
collect waste going by door-to-door
gov. waste collector
solid waster trader

Figure 2.1 - Use case diagram of the current process

2.3 REQUIRMENT ANALYSIS

2.3.1 SPECIFICATION OF USERS

In the domain analysis four user roles had been identified namely administrator, staff member, customer and driver. Each of these user types have different requirements thus they have different use of the system. Administrator is responsible for managing the whole system. Other types of users access the system to perform specific tasks which facilitates their daily activities. The staff members access the system frequently to manage the waste collection process. Drivers may access the system daily to check the jobs assigned for them. Customers tend to access the system mainly to purchase items from green points and may occasionally access to purchase bins, check green points or to use any other functionality available for them.

2.3.2 FUNCTIONAL REQUIREMENTS

Functional requirements of the system are listed in Tables 2.1-2.6 categorized by modules in the system.

Smart system for waste management	
ID	Description
1	The customers should be able to signup
2	The customers should be able to login
3	The staff should be able to login
4	The driver should be able to register
5	The driver should be able to login
6	The admin should be able to activate/deactivate users
7	The admin should be able to add users

Table 2.1 - Functional requirements of the system

Garbage management module		
ID	Description	Auto handled by the system
1	The customer should be able to request bins	
2	The customer / staff should be able to activate bins	Yes
3	The staff should be able to view all bin allocations	
4	The customer should be able to notify bin received	Yes
5	The customer should be able to view his bin allocation	
6	The system should be able to get bin full message from IoT unit	Yes
7	The driver should be able to update garbage weight	
8	The system should calculate the green points	Yes
9	The customer should be able to view green points	
10	The customer should be able to view Trash collection requests	

Table 2.2 - Functional requirements of the garbage management module

Trash collection module		
ID	Description	Auto handled by the system
1	The staff should be able to schedule a trip	
2	The staff should be able to assign a driver to the trip	
3	The staff should be able to view Trash to be collected	Yes
4	The staff should be able to view available drivers	Yes
5	The driver should be able to accept/reject job	
6	The driver should be able to view collection list	Yes
7	The driver should be able to mark trip start	

8	The driver should be able to view assigned jobs	Yes
9	The driver should be able to update garbage weight	
10	The system should calculate the green points	Yes
11	The customer should be able to view green points	Yes
12	The customer should be able to view Trash collection requests	Yes

Table 2.3 - Functional requirements of the trash collection module

Notification module		
ID	Description	Auto handled by the system
1	The system should be able to get bin full message from IoT unit	Yes
2	The customer should be able to send Trash collection request	
3	The system should be able to notify schedule to the customer	Yes
4	The system should be able to notify schedule to the driver	Yes
5	The system should be able to mail order information	Yes

Table 2.4 - Functional requirements of the notification module

e-Commerce module	
ID	Description
1	The staff should be able to add products
2	The staff should be able to update products
3	The customer should be able add products to the shopping cart
4	The customer should be able to checkout products

Table 2.5 - Functional requirements of the e-Commerce module

Complaint module	
ID	Description
1	The staff should be able to view complaints
2	The staff should be able to resolve complaints
3	The customer should be able add a complaint
4	The customer should be able to view complaints

Table 2.6 - Functional requirements of the complaint module

2.3.3 NON FUNCTIONAL REQUIREMENTS

Nonfunctional requirements are the constraints the system must work within.

2.3.3.1 SECURITY

The system facilitates to create users with specific rights for the application usage. The system must maintain the privacy of the users by maintaining their personal information inaccessible to others. Physical access to the computer storing the past pupil database shall be restricted to authorized personnel.

2.3.3.2 USER INTERFACE REQUIREMENTS

The user interface of the application must be user-friendly, intuitive and easy to use. The interface must provide ways to navigate easily from one place to another. The system must allow the user to preview queries and reports before printing.

2.3.3.3 PERFORMANCE REQUIREMENTS

The system shall function in real-time: any operation on the stored information, triggered by the member, should complete in less than 10 seconds. The system should allow simultaneous use by at least 50 users, without data corruption.

2.3.3.4 PORTABILITY REQUIREMENTS

The system runs on any platform and is accessible using any browser. For better performance a laptop or a desktop with an Intel core i3 processor or higher is recommended.

2.4 REVIEW OF THE SIMILAR SYSTEMS

Waste management is an undying topic in every country thus many research had been done on the topic and many online systems had been implemented. To be familiarize with the system to be implemented, similar waste management systems had been studied.

- Trash flow (Flow, 2020) -

It is a waste management software that automates daily tasks with tools like route management, in truck mobile solutions, and container tracking.

- Waste logics (Logics, 2020) -

It is a cloud-based waste management software which allows create bookings, track progress. It is available for different user sizes; small (<50 employees), medium (50 to 1000 employees), enterprise (>1001 employees)

- MY WM App (WM Intellectual Property Holdings, 2020) -

WM is waste management service provider in North America, offering services that range from collection and disposal to recycling and renewable energy generation. Their mobile app can be used to track garbage truck, pay bills, Request bulky item disposal and extra pickups.

- SENSONEO (Sensoneo, 2021)

Sensoneo is a smart waste management system which incorporates many modern technologies such as RFID features for bin management and container tracking, IOT for filled bin identification, smart tags and smart sensors for route planning. Sensoneo supports many functionalities including asset management, waste monitoring and route planning.

The features of the above systems can be compared with the solution which is going to be implemented. Trash flow is a software for waste handlers and it does not have any functionalities for the clients. MY WM App is for the clients only and does not support the employees to coordinate the garbage truck drivers nor generate any managerial reports for decision making. Waste Logics is a software with rich functionalities for customers, drivers and staff. But it lacks the smart bin feature. Sensoneo includes many functionalities as well as smart bin feature but it is not designed to collect waste from households. The mobile app allows citizens to check the status of nearby bins and dump waste to those. But the proposed solution is designed for doorstep waste collection. Comparison of proposed system with the existing systems is presented in Table 2.7.

Feature	**Trash Flow**	**Waste Logic**	**MY WM App**	**SENSONEO**	**Proposed System**
Users	Staff of waste management company	▪ Staff of waste management company ▪ Drivers ▪ Citizens	citizens	▪ Staff of waste management company ▪ Drivers ▪ Citizens	▪ Staff of waste management company ▪ Drivers ▪ Citizens
Use of IoT	No	No	No	Yes	Yes
Route Management	Yes	Yes	Yes	Yes	Yes
E-Commerce Module	No	No	No	No	Yes
Track collection trucks real time	No	Yes	Yes	Yes	No

Table 2.7 - Comparison of waste management systems

Many research have been done on IoT based waste management systems. This year Pardini et al. (Pardini, 2020) proposed a solution that follows an IoT-based approach where the discarded waste from the smart bin is continuously monitored by sensors that inform the filling level of each compartment, in real-time. These data are stored and processed in an IoT middleware providing information for collection with optimized routes and generating important statistical data for monitoring the waste collection accurately in terms of resource management and the provided services for the community. Citizens have easy access to information about the public waste bins through the Web or a mobile application. This study introduces a solution for public waste bins but the proposed system is implemented for bins of citizens thus the garbage will be collected based on the citizen's requirement and they get more benefits such as selling the garbage and buying products through the proposed solution.

In the system proposed by Mahajan et al. (S.A. Mahajan, 2017) in India, public dustbins will be provided with embedded device which helps in real time monitoring of level of garbage in garbage bins. The data regarding the garbage levels used to provide optimized route for garbage collecting vans, which reduce cost associated with fuel. The load sensors increase efficiency of data related to garbage level and moisture sensors provide data of waste segregation in a dust bin. As mentioned earlier, this study also discussing of public garbage bins thus lacks features which support citizens. But this system includes moisture sensors which lacks in the proposed system.

In 2017 Alaam et al. (Alam, 2017) proposed a system, in which multiple dustbins in different locations have been provided with low-cost embedded devices which sends data to internet cloud after reaching a certain threshold level. Moreover, the device transmits the level along with the unique ID via SMS to the closest authorized cleaner. The total system is clearly visible to the highest authority of waste management system and the system notifies the authority if any cleaner avoids the SMS.

In Mahajan's research an Arduino board was used as the central processing system and in the other two Raspberry pi was used. In Mahajan and Pardini's solutions ultrasonic sensors; for measuring the level of waste filling present inside the compartment, GPS module, load sensors; that measures the weight of the residues present in the compartment, temperature and humidity sensors was used. Alaam had used only an ultra-sonic sensor and GPS module.

The current system focuses on main modules of waste management so real time fleet monitoring of garbage trucks is not fully implemented as it is beyond the scope and does not fit to the given time frame. Further it is impractical to test that module during this travel restricted period. But the last visited garbage collection point of the truck visited is indicated in the map in the map which much the same as real time tracking.

There are variety of systems for waste management with numerous functionalities but only few integrated the smart bin feature. Many research-based systems are available which implements smart bins for waste management but they lack the functionalities a waste management system should have. The proposed system will fill the gap and will include smart bin feature together with many functionalities of a waste management system that is tailored to Sri Lankan context.

By reviewing the existing systems it is clear that there is a possibility of building an effective waste management system with smart bin feature by integrating features of commercial waste management systems and research based systems.

2.5 PROCESS MODEL FOR THE SYSTEM

Among many of the process models available, the Incremental development model will be used to develop the system. It involves developing the system in an incremental fashion. The most important part of the system is first delivered and the other parts of the system are then delivered according to their importance. Thus the system is implemented in four phases and the modules delivered in each phase is as below.

- Phase 1: Garbage Management module, Notification module
- Phase 2 : Trash collection module
- Phase 3: E commerce Module
- Phase 4: Complaint Module

Incremental development model was chosen focusing on following benefits of it.

- A working software is available in less time
- Errors can be identified easily
- Changing requirements

DESIGN

Findings of the requirement analysis phase was taken into consideration when designing the system. A good design leads to a successful implementation. This chapter describes the desired features and the operations of the system in detail, using screen layouts, UML diagrams and database diagrams.

3.1 DESIGN STRATEGIES

While the software is being conceptualized, a plan is chalked out to find the best possible design for implementing the intended solution (Anon., 2017). Out of the following alternative solutions, develop a web based application for a client server environment was selected as best possible solution to implement the system.

Solution 1 : Develop a web based application

After the design stage, the system could be implemented as a web solution using reusable codes and components. Reusing existing components will speed up the work and enhance the quality of the system. Web solution will allow access to the citizens from anywhere from any platform. Furthermore this approach enables customization in future releases.

Solution 2: Develop a standalone application

A standalone system could be implemented for the waste management authorities. The main issue is that then the garbage bin status cannot be informed to them. Other problems that can occur are the citizens will not have any interaction with the system so online bin requesting, communication, viewing schedule will not be possible. This approach was rejected as the information needed to be accessible from multiple computers.

3.2 SYSTEM ARCHITECTURE

The system architecture is the conceptual model that defines the structure, behavior, and more views of a system (Anon., n.d.). It helps to visualize an overall picture of the system and how system components and sub systems work together to implement the overall system.

3.2.1 HARDWARE ARCHITECTURE

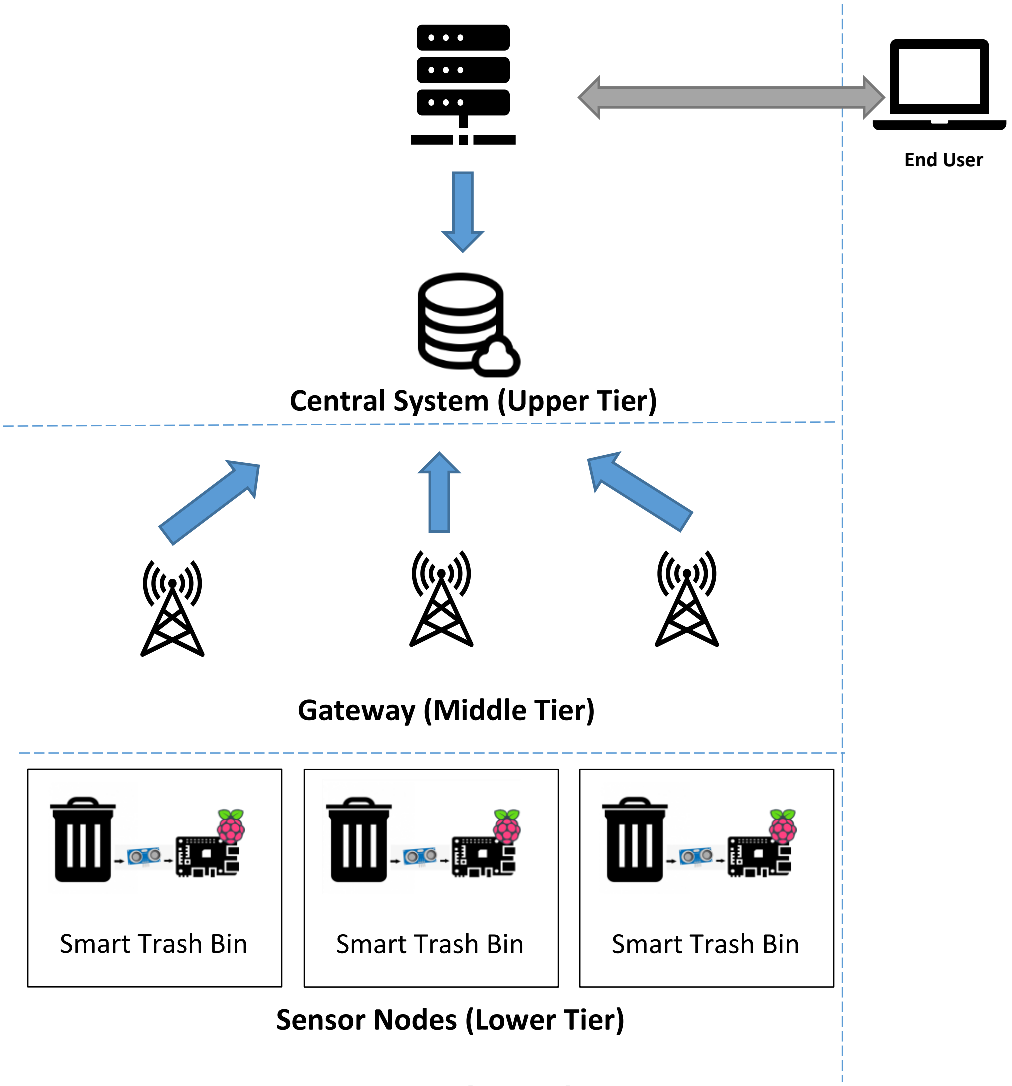


Figure 3.1 - Hardware architecture

Figure 3.1 describes the overall system hardware including components in the organization and their connectivity with components outside the organization's network. The system composed of three tier architectures: upper, middle and lower tiers. The upper tier comprises of the central database; the middle tier contains the gateway; and the lower tier consists of the sensors. The system receives and stores the waste status via the gateways from different citizen’s location after establishing connection with the server. A graphical user interface is provided with role-based restrictions for both administrator and the citizens to access the system.

3.2.2 SOFTWARE ARCHITECTURE

Client server architectural style is used for this system in which many clients (citizens, drivers) request and receive service from the centralized server of the organization.

When the trash bin is filled it will send the notification to the web server of the organization. The communication from the trash bin to the central station is done by using TCP/IP protocol using present GSM/GPRS wireless infrastructure. The officers monitor the status of the trash bins through a web page. The citizens are communicating with the web page through smart phones or PCs using wireless network connection.

3.3 SYSTEM DESIGN

Unified Modeling Language (UML) was used to design the diagrams based on Object Oriented concepts. To elaborate the functionalities of the system, diagrams such as use case diagrams, activity diagrams, sequence diagrams had been used. Such design diagrams are useful to capture requirements and to ensure that the system meets those requirements.

3.3.1 SYSTEM USE CASE DIAGRAM

A system use case diagram displays the relationships between consumers and providers of application services. Application services are consumed by actors or other application services and the application use case diagram provides added richness in describing application functionality by illustrating how and when that functionality is used. The purpose of the system use case diagram is to help to describe and validate the interaction between actors and their roles with applications (Anon., 2014).

Figure 3.2 illustrates the system use case diagram of the waste management system in which always the customer receives garbage points when they handover the garbage and may consume the points when purchasing cart items.

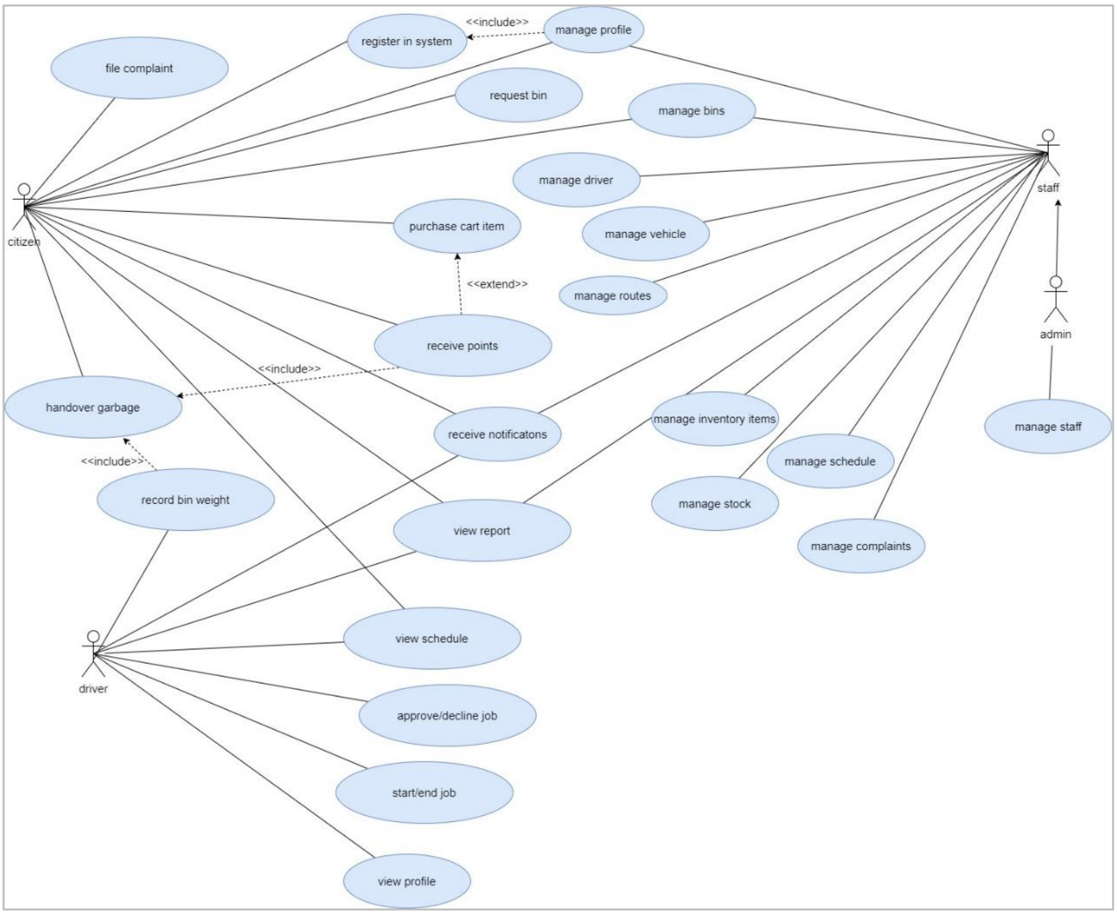


Figure 3.2 - System use case diagram

3.3.2 GARBAGE MANAGEMENT MODULE

This module is used for citizens to sign up and request bins. Further, this module handles trash collection requests.

System's functionality and requirements for each user captured using use case diagram as shown in Figure 3.3.

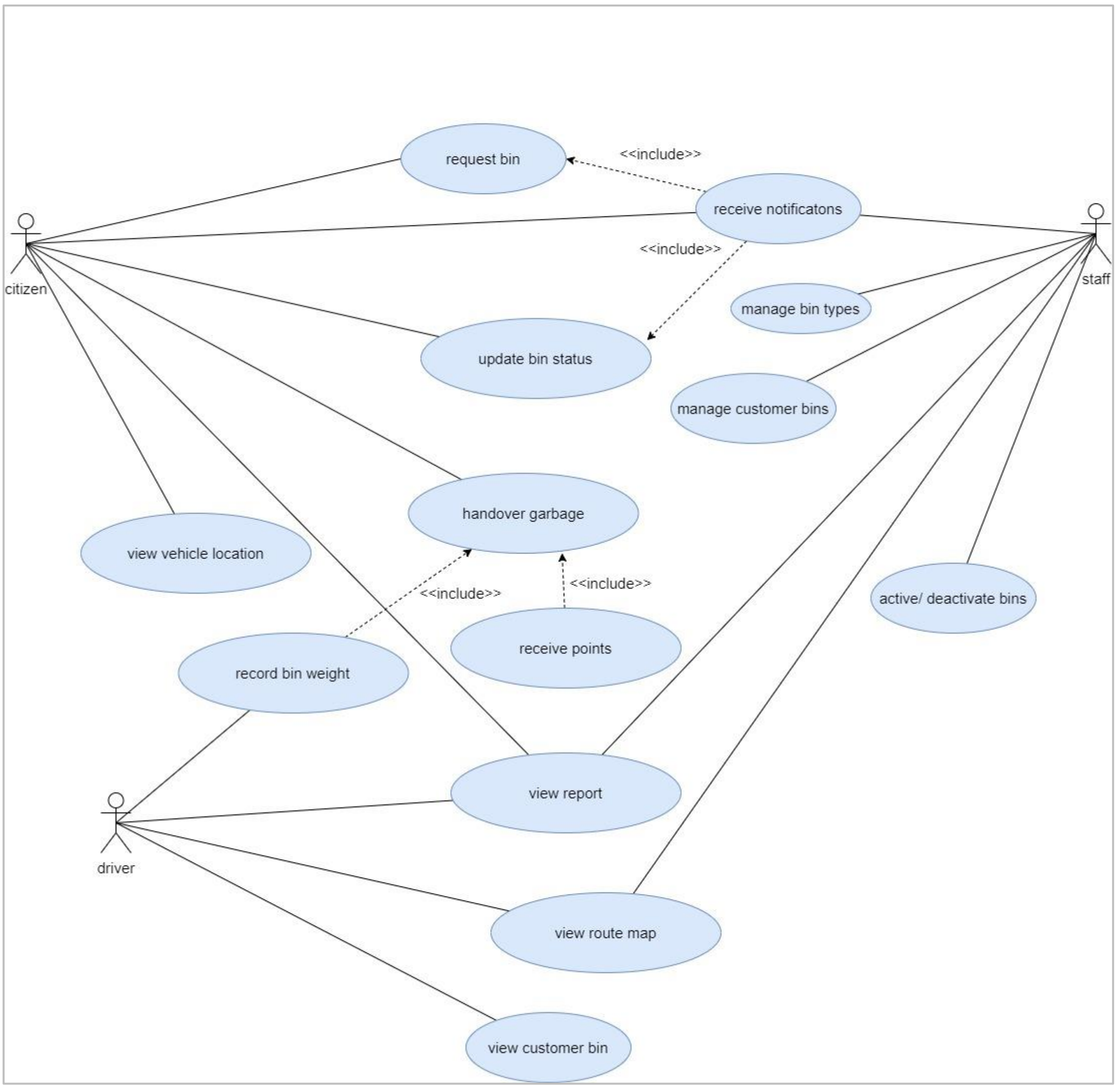


Figure 3.3 - Use case diagram; Garbage management system

User actions were grasped by the activity diagrams. The activity diagram in Figure 3.4 describes the trash collection request process. Figure 3.4 further describes hand over garbage use case, which is initiated by bin full notification sent by the trash bin.

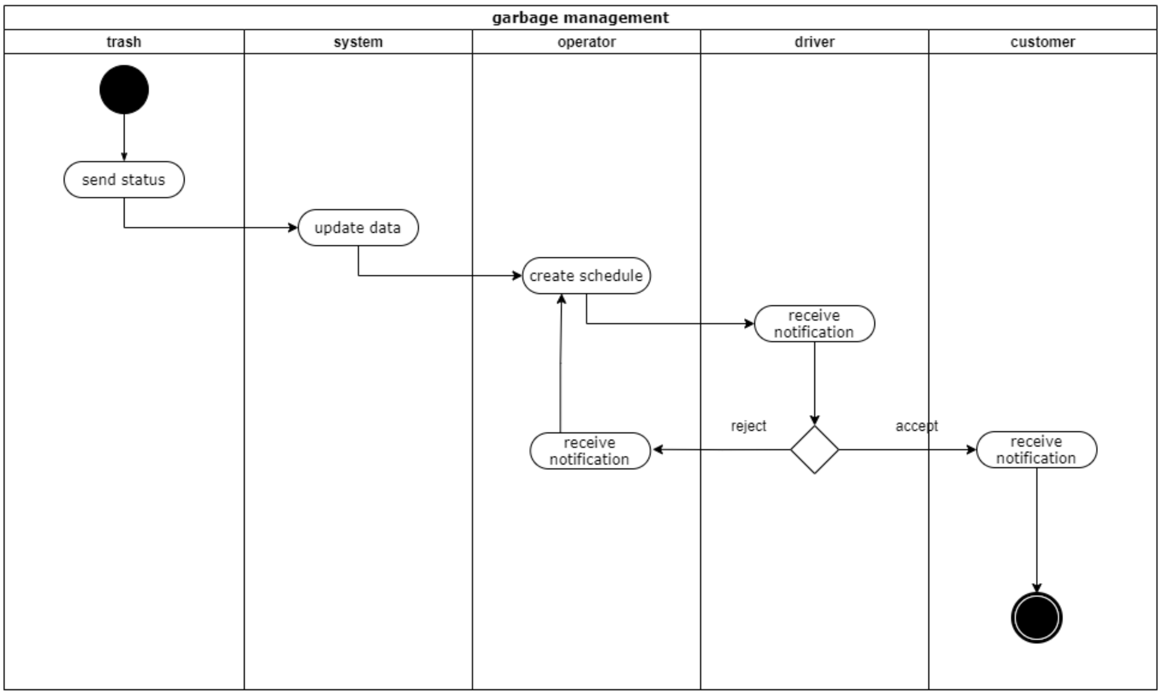


Figure 3.4 - Activity diagram; Trash collection request

3.3.3 TRASH COLLECTION MODULE

This module deals with collection trip scheduling, driver assignment and trash collection.

Green points scheme introduced to encourage customers to dispose garbage properly so they earn credits and buy products from e-Shop. Green points are assigned to the citizen based on the garbage type and weight of garbage as shown in Table 3.1. Following formula is used to calculate the total green points for each customer. In the formula total green points are calculated by reducing the consumed green points (C_k) from the green points collected so far (P_i) then the current earned green point amount (wx) is added to that. For example when calculating the customer's total green points for 4th time, their total consumed green points (C_k) will be reduced from the green points collected so far (3rd time) and will add the current earned green points amount (4th time) to that.

$$P_n = wx + \sum_{i=1}^{n-1} P_i - \sum_{k=1}^{n} C_k$$

P_n = Points earned on the n^{th} time

w = weight of the trash

x = Points per 1kg. for given trash type

C_k = Points consumed on the k^{th} time

Simply, each time the customer handing over the garbage, green points are added and each time he purchases items from the cart green points are reduced. for example if a customer is handing over 3 kg of glass and his green point history is as Table 3.1 then his green points are calculated as follows.

Date	Earned Points	Consumed Points
3/4/2021	5	
16/4/2021	12	
22/4/2021		7
5/5/2021	9	
8/6/2021		3
Total	**26**	**10**

Table 3.1 - Green points per Kg. for waste types

P_4 = 3 x 15 + 26 – 10

= 61

The current green point is calculated by multiplying the weight of each type of garbage with the corresponding point of that particular type then summing up the values. Point of each garbage type is assigned by analyzing the demand of waste collectors in the country.

Garbage type	Green points per 1 Kg.
Glass	15
Plastic	10
Paper	5
e-Waste	25

Fabric	5
Rubber	10
Metal	20

Table 3.2 - Green points per Kg. for waste types

The sequence diagram in Figure 3.5 describes the trash collection process based on a timeline.

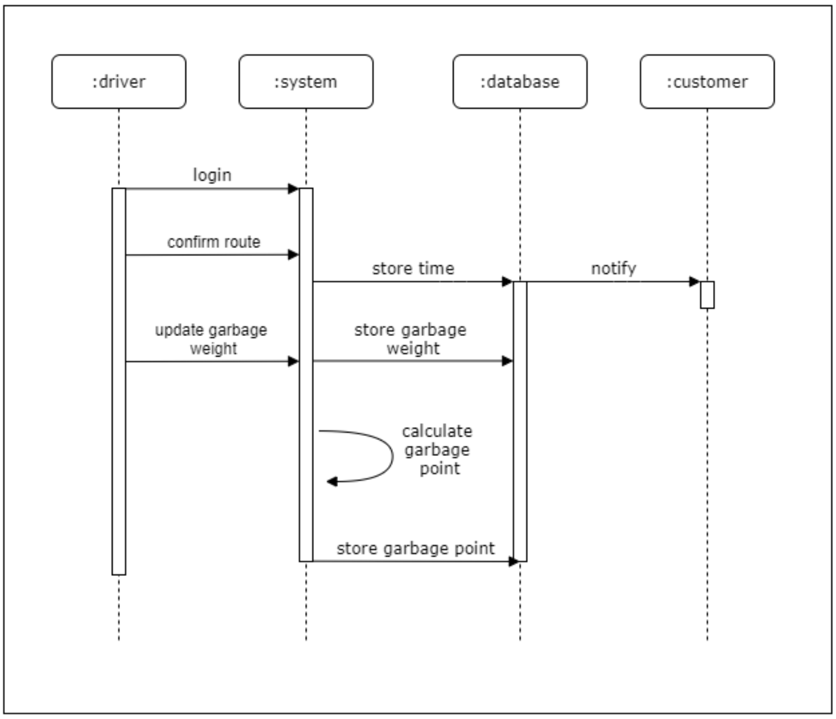


Figure 3.5 - Sequence diagram; Trash collection process

3.3.4 COMPLAINT MODULE

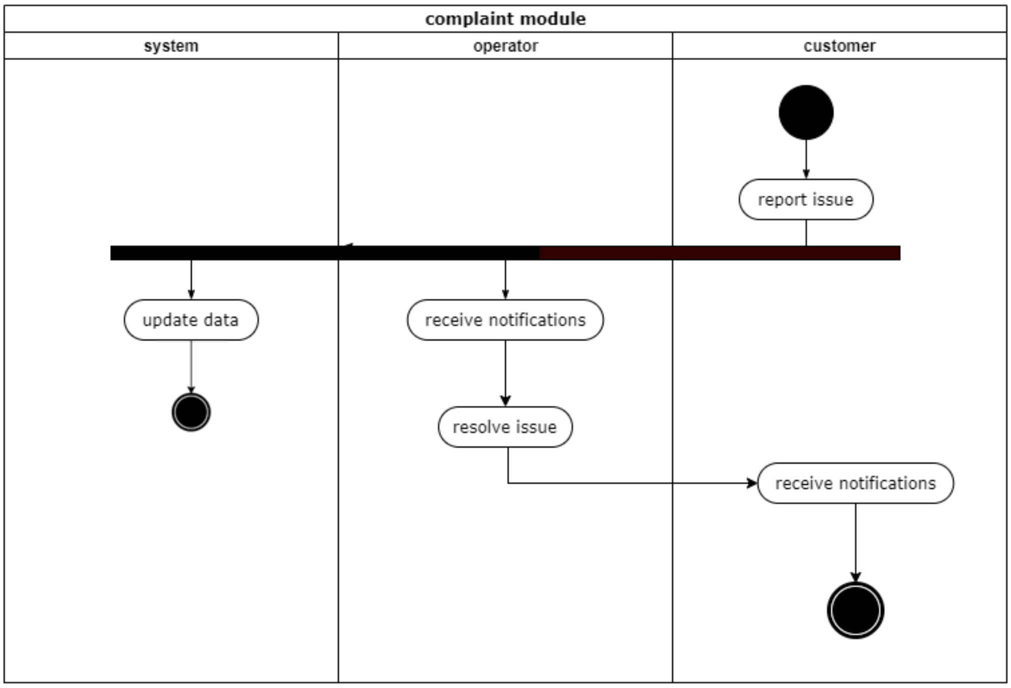


Figure 3.6 - Activity diagram; Complaint module

This module deals with recording and resolving complaints made by citizens regarding drivers, bins or schedules. issuing scholarships to the students.

In the activity diagram in Figure 3.6 swim lanes are used to show activities performed by each user.

3.3.5 CLASS DIAGRAM

Class Diagram gives an overall picture of the system in the form of classes with their attributes and behaviors.

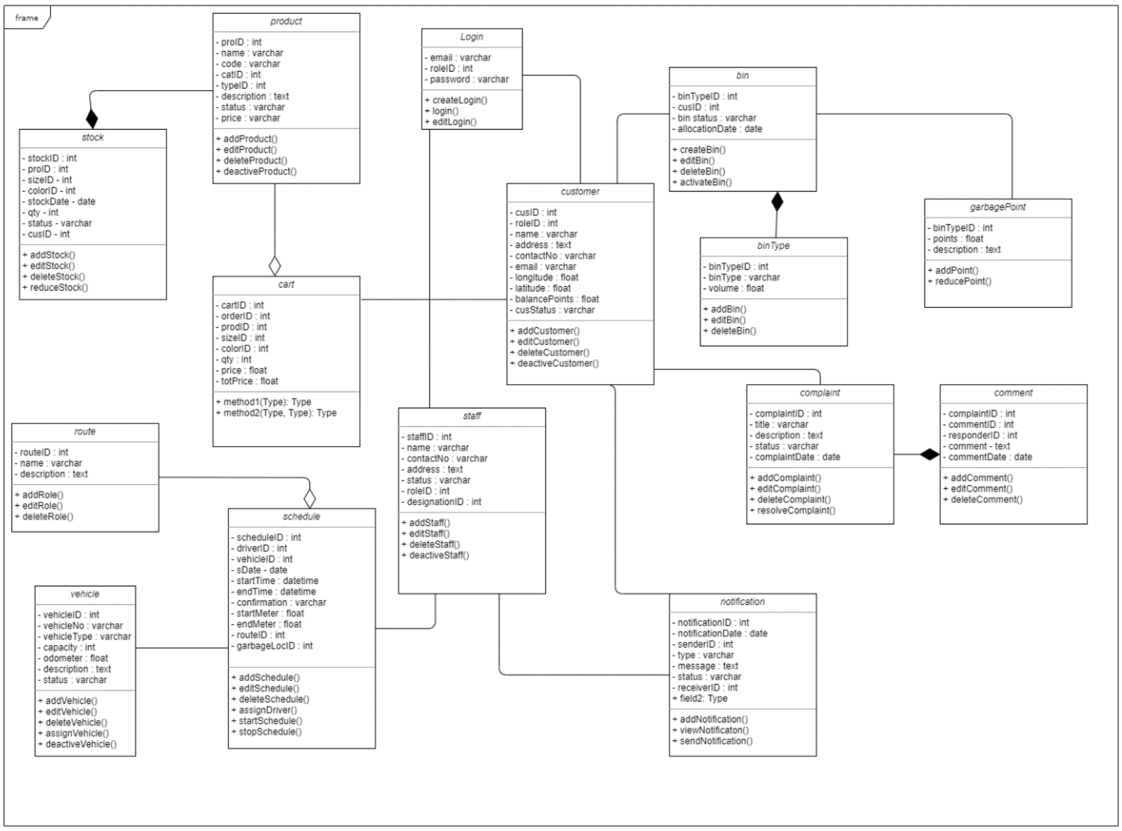


Figure 3.7 - Class diagram

3.4 INTERFACE DESIGN

Graphic User Interface (GUI) of an application is receiving greater attention, since it can be decisive in determining if the application is accepted or rejected by users (Ana Isabel Molina, 2009). Different mechanisms had been used to increase user friendliness and to keep user attracted. Familiar Terminology, Cool colors, Image icons and instant help are some of them. Combo boxes, dialog boxes are used to limit user typing and hence limit user errors and increase speed.

3.4.1 INTERFACE DESIGN CONSIDERATION

3.4.1.1 USER FRIENDLINESS

The application will be used mostly by entry level users namely the customers and the drivers who are not much experienced with using technical applications. So interfaces are designed in a user centred approach so that they are user friendly enough to be used by anyone that doesn’t have a comprehensive knowledge of the technical details behind it

3.4.1.2 ACCESSIBILITY

If the user can’t find what they want specially the customers, may reject the application. The most frequently needed functionalities are placed in the top of the page so they will be quickly available. Tabs placed to make things readily accessible and is presented in Figure 3.8.

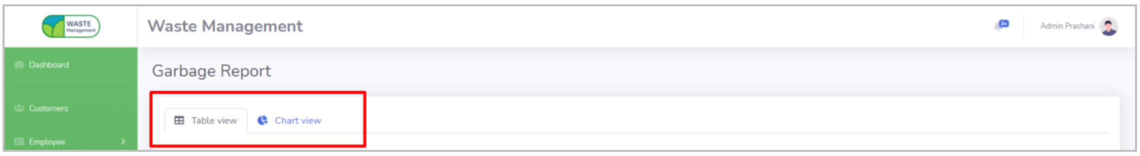


Figure 3.8 - Page tabs

3.4.1.3 CONSISTENCY

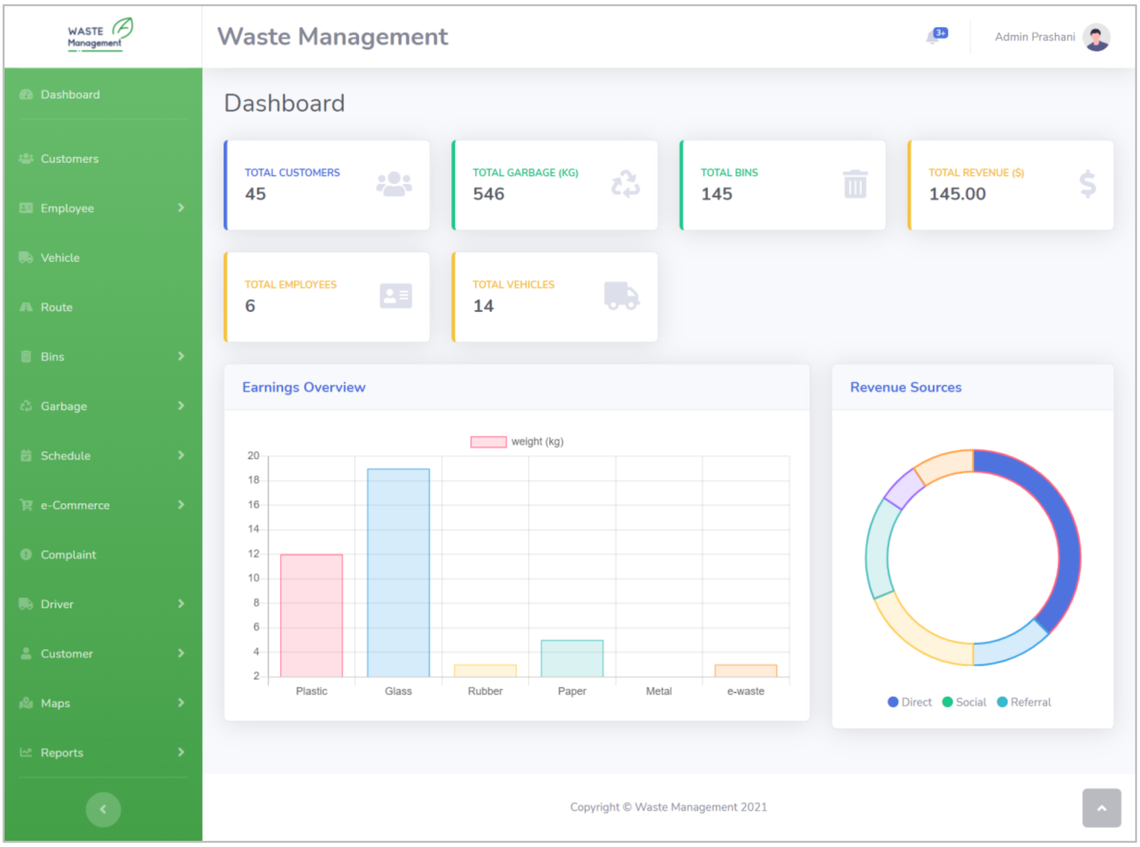
The user interface are consistent throuhout the whole application. For example the same color code is used for action buttons placed on every page and the search bar is placed in the top section of the page.

3.4.1.4 CLARITY

Users will not be confused of the purpose of any page as there will be no clutter. One way of achieving clarity is to move from one step to another on different pages. In the checkout process the customers have to navigate from product page to a shopping cart page then to a checkout page, rather than scrolling down a single page for the whole process.

3.4.2 MAIN USER INTERFACES

3.4.2.1 DASHBOARD



Dashboard

TOTAL CUSTOMERS 45

TOTAL GARBAGE (KG) 546

TOTAL BINS 145

TOTAL REVENUE ($) 145.00

TOTAL EMPLOYEES 6

TOTAL VEHICLES 14

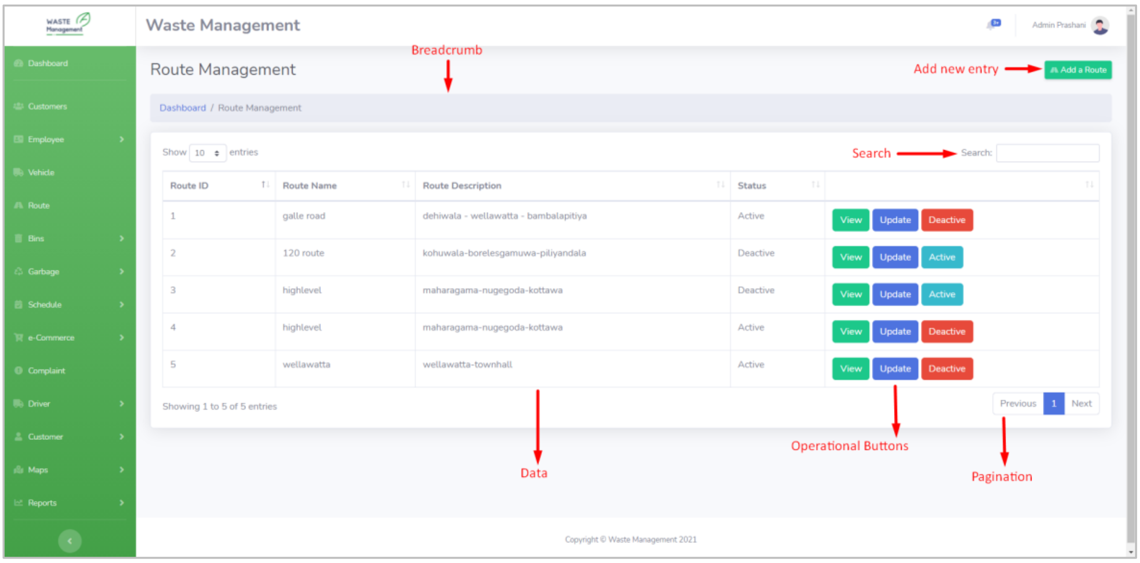
Copyright © Waste Management 2021

Figure 3.9 - Dashboard; Waste management system

Figure 3.9 shows the main screen a user would get that has provision for access into the system. The available features may differ based on the privileges of the user.

3.4.2.2 MAIN VIEW PAGE

In almost every module there is a data grid with important data showed as a main page. On the right side of each record there are operation buttons. In Figure 3.10 customer management page given as an example to show the basic structure of the data table view (management) page.



Route Management

Dashboard / Route Management

Show 10 entries

Search:

Route ID	Route Name	Route Description	Status	
1	galle road	dehiwala - wellawatta - bambalapitiya	Active	View Update Deactive
2	120 route	kohuwala-borelesgamuwa-piliyandala	Deactive	View Update Active
3	highlevel	maharagama-nugegoda-kottawa	Deactive	View Update Active
4	highlevel	maharagama-nugegoda-kottawa	Active	View Update Deactive
5	wellawatta	wellawatta-townhall	Active	View Update Deactive

Showing 1 to 5 of 5 entries

Previous 1 Next

Copyright © Waste Management 2021

Figure 3.10 - Main data table view page

BREADCRUMBS

Shows current page location and links to previous pages.

SEARCH TABLE

Data in the table can be search using search field. Searching is possible with this data or set of data.

ADD NEW ENTRY

New entry can be added to the table using the given button.

PAGINATION

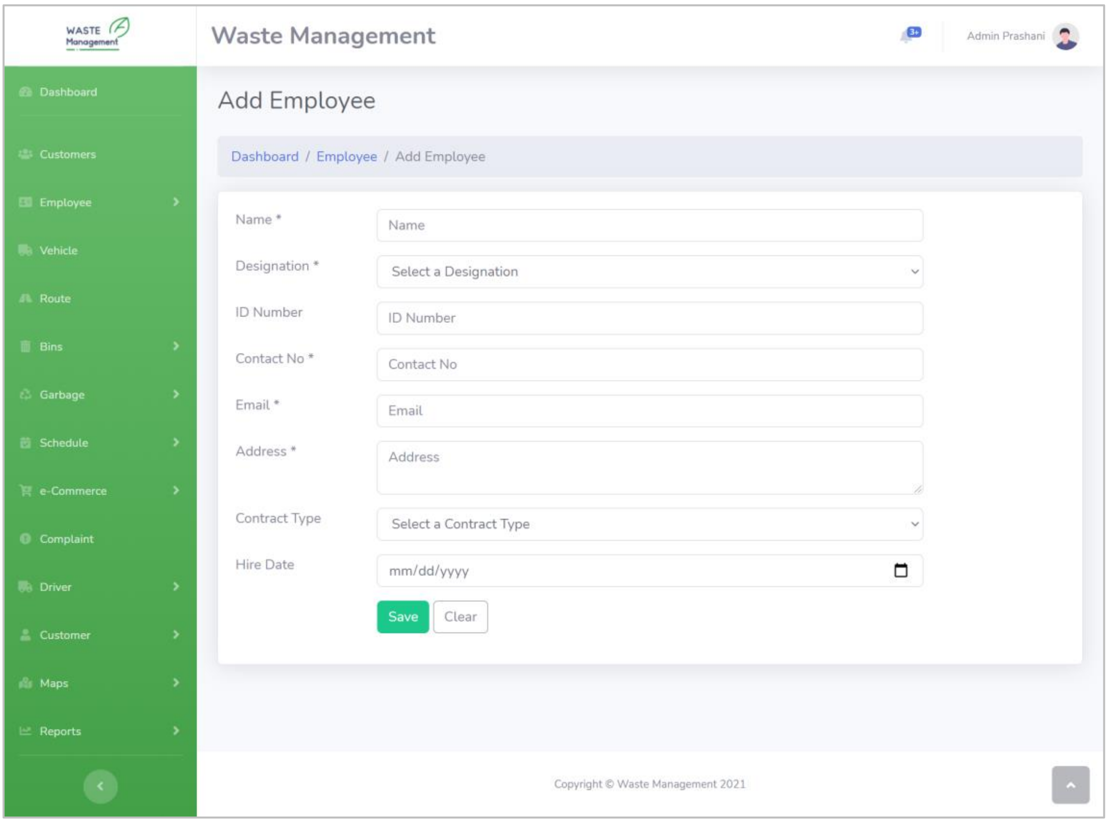
Use to go through the pages of the data table

OPERATIONAL BUTTONS

- View – Can get the full view of the relevant entry
- Update- Update the entry
- Active/Deactivate – change the status of the entry

3.4.2.3 DATA ENTRY FORM

The system has input forms to gather necessary data and to pass them to the database. The forms are designed in a user friendly manner and data validation has been implemented to minimize user errors. Data retrieved from the database are filled in to input forms so same form is used for updating data. The system consists of many forms similar to the one shown in Figure 3.11.



Add Employee

Dashboard / Employee / Add Employee

Name *	Name
Designation *	Select a Designation
ID Number	ID Number
Contact No *	Contact No
Email *	Email
Address *	Address
Contract Type	Select a Contract Type
Hire Date	mm/dd/yyyy

Copyright © Waste Management 2021

Figure 3.11 - Add employee form

3.4.2.4 BIN MAP VIEW

Using Bin map view staff can see the status of the bins through the map and arrange schedules according to it. It gives graphical view of the bin status geographically so can easily identify which bins needs to be collected together.

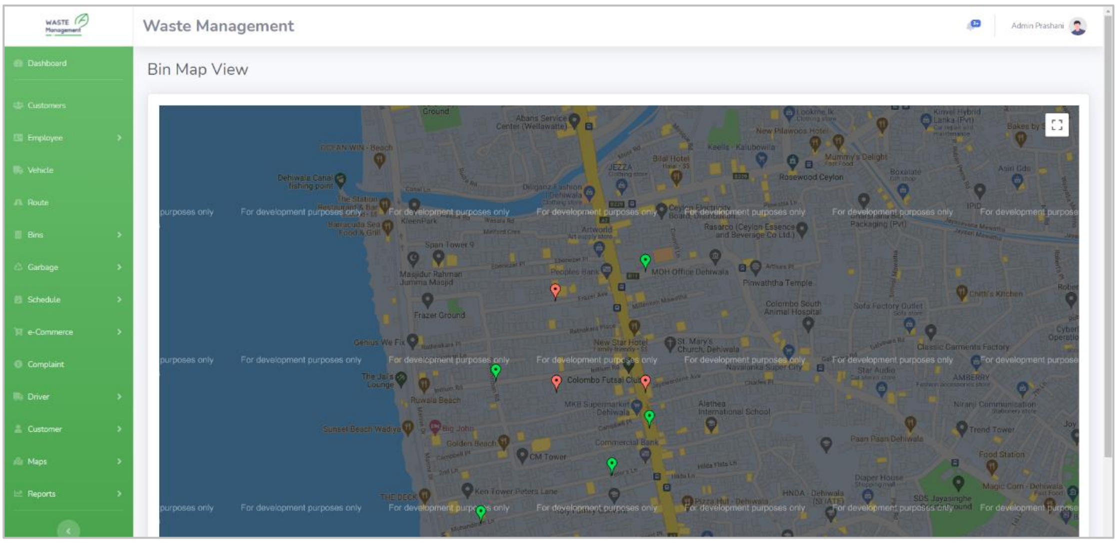
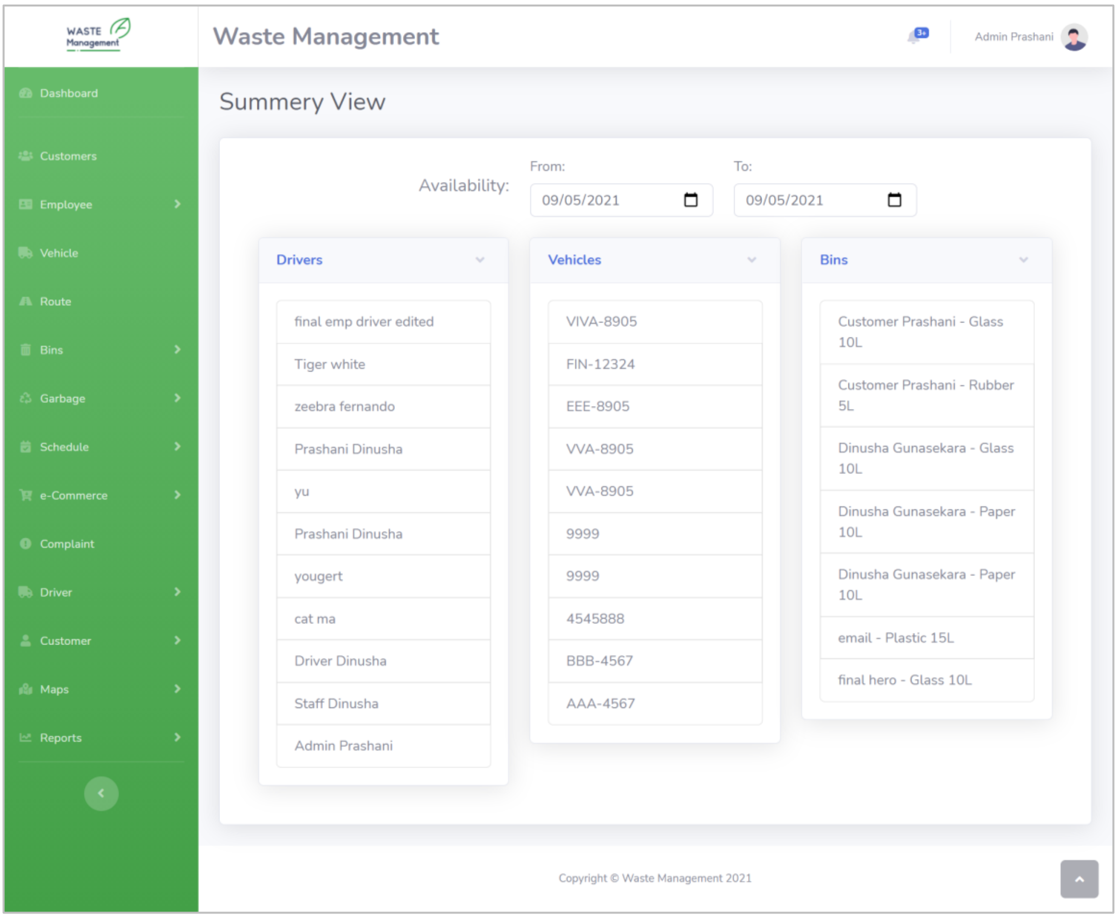


Figure 3.12 - Bin map view

3.4.2.5 SUMMERY VIEW

Using summery view staff can have an overall idea about what are the available drivers, vehicles and bins that need to be collected for given date which will help them to create schedules effectively. It acts as centralized information hub.



Summery View

Availability: From: 09/05/2021 To: 09/05/2021

Drivers
- final emp driver edited
- Tiger white
- zeebra fernando
- Prashani Dinusha
- yu
- Prashani Dinusha
- yougert
- cat ma
- Driver Dinusha
- Staff Dinusha
- Admin Prashani

Vehicles
- VIVA-8905
- FIN-12324
- EEE-8905
- VVA-8905
- VVA-8905
- 9999
- 9999
- 4545888
- BBB-4567
- AAA-4567

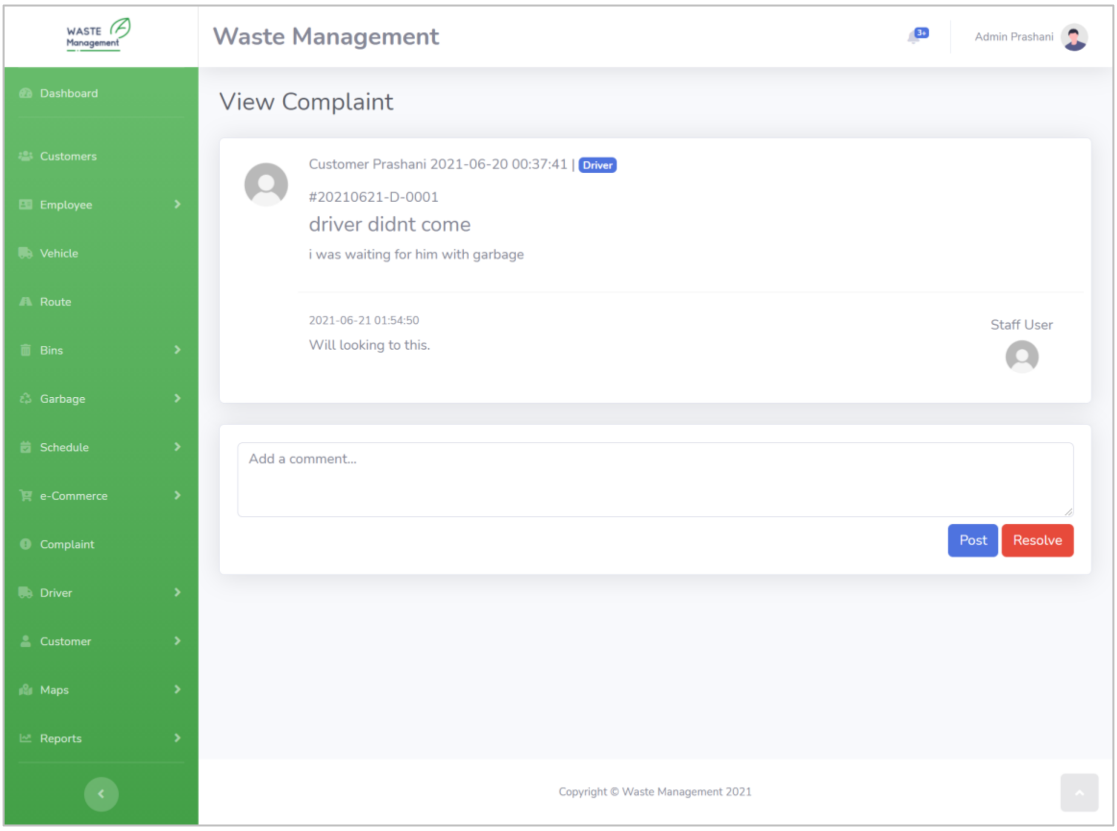
Bins
- Customer Prashani - Glass 10L
- Customer Prashani - Rubber 5L
- Dinusha Gunasekara - Glass 10L
- Dinusha Gunasekara - Paper 10L
- Dinusha Gunasekara - Paper 10L
- email - Plastic 15L
- final hero - Glass 10L

Copyright © Waste Management 2021

Figure 3.13 - Summery view

3.4.2.6 VIEW COMPLAINT PAGE

After a customer created a complaint, staff can update the information related to complaint such as status, what are the actions taken and other comments in the view complaint page. Customer can exchange information with staff as a thread and follow up the process of the complaint. View complain page shows all details related to complaint. When complain resolved staff can close the complaint using ‘Resolve’ button.



Customer Prashani 2021-06-20 00:37:41 | Driver

#20210621-D-0001

driver didnt come

i was waiting for him with garbage

2021-06-21 01:54:50

Will looking to this.

Staff User

Figure 3.14 - View complaint page

3.5 DATABASE DESIGN

Data are a valuable asset of any organization. Database design is a process of modeling an enterprise in the real world. In fact, a database itself is a model of the real world that contains selected information needed by the enterprise (Jewett, n.d.). A good database design results in proper management, easy retrieval of data in the system.

3.5.1 DATA NORMALIZATION

Normalization is a systematic approach of decomposing tables to eliminate data redundancy and undesirable characteristics like Insertion, Update and Deletion anomalies (Anon., n.d.). It follows multiple steps to remove duplicate data from relational tables. The purpose of normalization is to eliminate data redundancy and ensuring data consistency.

For example Table 3.2 show the initial form of bin type table.

binTypeID	binType	binVolume
1	Rubber	10, 25
2	Glass	10, 25, 50
3	Paper	10

Table 3.3 - Bin type table – before 1NF

As per first normal form each column must have a unique value so the above table is converted in to 1NF as follows.

binTypeID	binType	binVolume
1	Rubber	10
2	Rubber	25
3	Glass	10
4	Glass	25
5	Glass	50
6	Paper	10

Table 3.4 - Bin type table - 1NF

The bin table in 1NF shown in Table 3.4

empID	empName	Role ID	emp Status	Leave To	leave From	reason
1	Admin Prashani	1	active	Null	Null	Null
2	Staff Dinusha	2	active	2021-3-10	2021-3-12	fever
2	Staff Dinusha	2	active	2021-5-20	2021-5-21	personal
3	Driver Dinusha	3	active	2021-6-20	2021-6-21	personal

Table 3.5 - Employee table; 1NF

As per the Second Normal Form there must not be any partial dependency of any column on primary key (Anon., n.d.). So to achieve the 2NF the above table was split in to two moving the leave details in to an independent table.

New employee table 2NF would be shown in Table 3.5;

empID	empName	roleID	empStatus
1	Admin Prashani	1	active
2	Staff Dinusha	2	active
3	Driver Dinusha	3	active

Table 3.6 - Employee table; 2NF

New leave table introduced for 2NF would be,

empLeaveID	empID	Leave To	leave From	reason
1	2	2021-3-10	2021-3-12	fever
2	2	2021-5-20	2021-5-21	personal
3	3	2021-6-20	2021-6-21	personal

Table 3.7 - Leave table; 2NF

3NF applies that every non-prime attribute of table must be dependent on primary key. So this transitive functional dependency should be removed from the table and also the table must be in Second Normal form (Anon., n.d.).

IMPLEMENTATION

Implementation converts the concept presented in the design into an outcome. Latest available technologies had been used to implement the system. This chapter presents the implementation environment, implementation approach and the coding details.

4.1 IMPLEMENTATION ENVIRONMENT

A suitable environment has to be established to ensure that the implementation process runs smoothly. Hardware was used to implement the system efficiently and to get best performance of software. When choosing software compatibility, features were considered and more attention was given to choose free and open source software.

4.1.1 SOFTWARE ENVIRONMENT

Following languages and technologies had been used for system development

- PHP
- Bootstrap
- HTML
- CSS
- JavaScript
- jQuery
- AJAX

PHP had been chosen mainly because it is freely available and feature rich scripting language widely used for the website and web application development.

Following software and development tools had been used with above languages and technologies.

- Windows 10 Pro operating system

- XAMPP version 7.2.34 including
    - Apache server 2.4.46
    - PHP 7.2.34
    - phpMyAdmin 5.0.4
- PhpStorm IDE 2021.1.3 x64
- GitHub for source code management

4.1.2 HARDWARE ENVIRONMENT

The System was implemented in a computer having following hardware configurarions.

- Intel(R) Core(TM) i7-10510U CPU @ 2.30 GHz
- 16 GB RAM
- 1 TB Hard disk

The system incorporates an IoT module with following related accessories.

- Raspberry Pi3 model B
- Breadboard
- Female to male and male to female jumper wires
- Ultrasonic sensor
- Resistors
- Power bank

4.2 REUSED MODULES

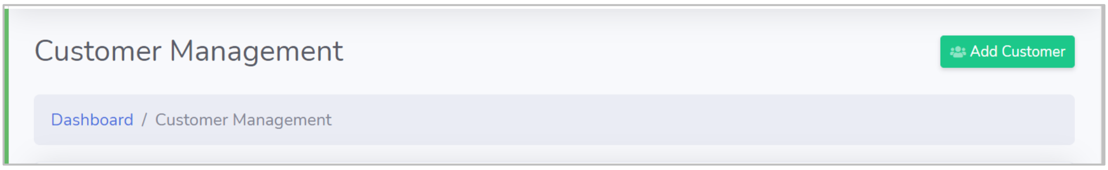
- Data Tables - DataTables is a free and open source plug-in for the jQuery JavaScript library. It is a highly flexible tool, and add advanced interaction controls to any HTML table. It supports Pagination, instant search and multi-column ordering (Anon., n.d.).

- PHPMailer - PHPMailer is a code library to send emails safely and easily via PHP code from a web server (Anon., 2021).
- Chart.js – Simple, clean and engaging HTML5 based JavaScript charts. Chart.js is an easy way to include animated, interactive graphs on website for free (Anon., 2021).
- Google Map API - The Maps JavaScript API lets customize maps with own content and imagery for display on web pages and mobile devices. The Maps JavaScript API features four basic map types (roadmap, satellite, hybrid, and terrain) which can modify using layers and styles, controls and events, and various services and libraries (Anon., 2021).
- Ultrasonic Distance Measurement Using Python – The module works by sending an ultrasonic pulse into the air and measuring the time it takes to bounce back. This value can then be used to calculate the distance the pulse travelled (Matt, 2012) .

4.3 WEB FRAMEWORKS USED

4.3.1 BOOTSTRAP

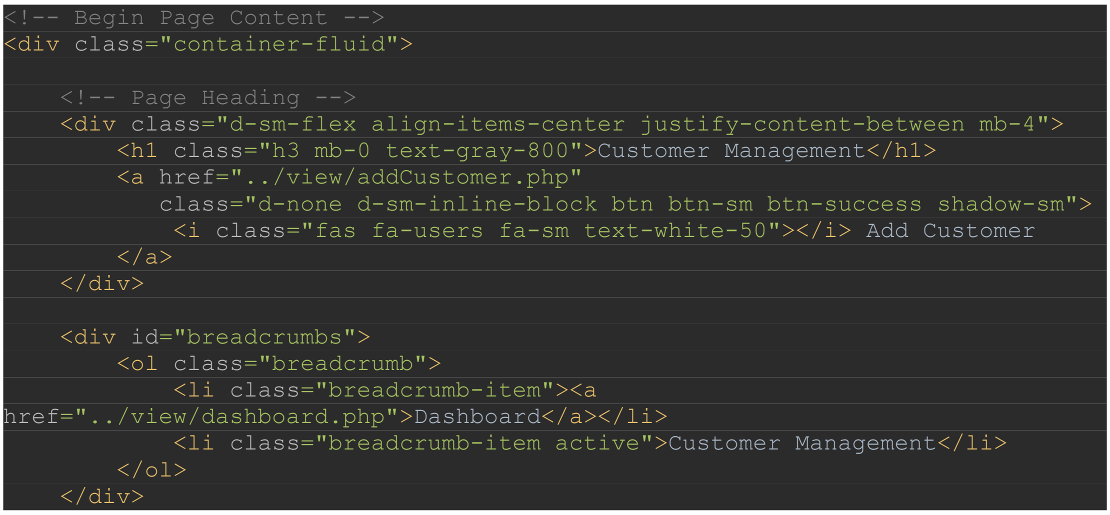
Bootstrap is a free and open source client side web UI framework. It helps to overcome the overhead associated with common activities performed in designing the front end of the web pages. Bootstrap contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components as well as optional JavaScript extensions (Anon., n.d.).



Customer Management

Figure 4.1 - Bootstrap code output

The below code creates the output shown in Figure 4.1



```
<!-- Begin Page Content -->
<div class="container-fluid">

    <!-- Page Heading -->
    <div class="d-sm-flex align-items-center justify-content-between mb-4">
        <h1 class="h3 mb-0 text-gray-800">Customer Management</h1>
        <a href="../view/addCustomer.php"
           class="d-none d-sm-inline-block btn btn-sm btn-success shadow-sm">
            <i class="fas fa-users fa-sm text-white-50"></i> Add Customer
        </a>
    </div>

    <div id="breadcrumbs">
        <ol class="breadcrumb">
            <li class="breadcrumb-item"><a
href="../view/dashboard.php">Dashboard</a></li>
            <li class="breadcrumb-item active">Customer Management</li>
        </ol>
    </div>
```

Figure 4.2 - Bootstrap code snippet

4.3.2 MVC FRAMEWORK

MVC is a well known architectural pattern popular among professionals in IT industry. It divides a given application into three interconnected parts. That is, it seperates data modal with the business logic as well as user interface. The benefit of using the MVC pattern speeds up the development process and also the modifications does not affect the entire model.

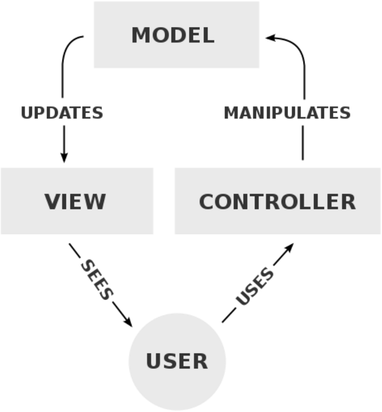


Figure 4.3 - MVC pattern

Source: https://en.wikipedia.org

4.4 MODULE INTERACTONS

The whole system consists of five modules and the interaction among them is depicted in Figure 4.4. It is clear from the Figure 4.4 that this is a system with high cohesion and low coupling which leads to easy maintenance.

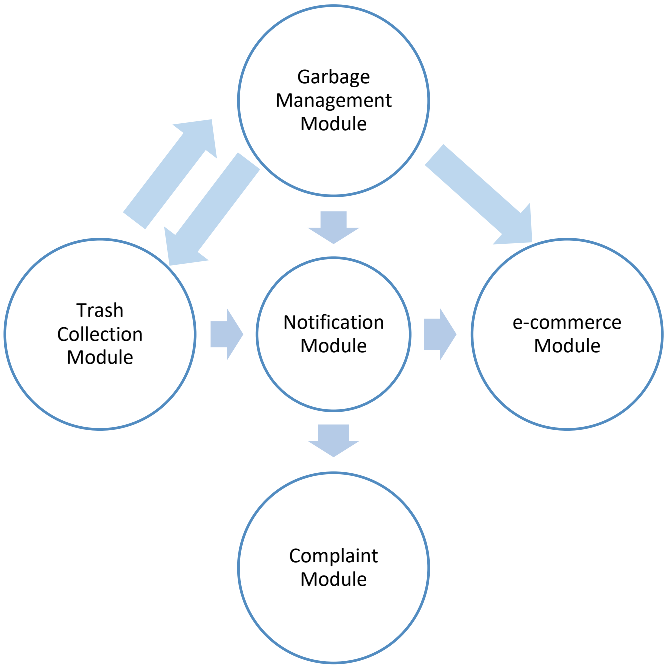


Figure 4.4 - Interaction among system modules

4.5 MAJOR CODE SEGMENTS

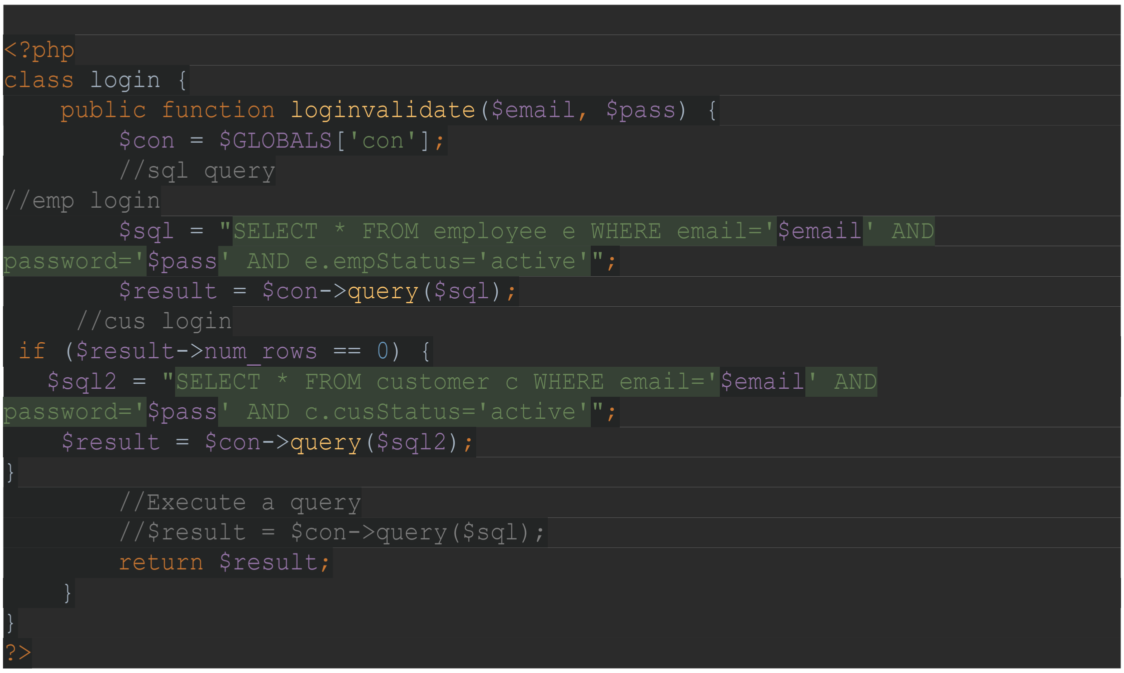
The code was written in using OOP concepts, which is industry standard that helps to achieve reusability, modularity, readability and understandable format to ease future enhancements. The readability of the code was maintained by comments, meaningful variable names and code indentation.

MVC design pattern had been followed when writing the code. According to MVC pattern, code related to a specific function comprises of three parts; Model, View and a Controller. Implementation of each component is described below with appropriate code segments using login module as an example.

4.5.1 LOGIN COMPONENT

4.5.1.1 MODEL

The username and password passed for the login model is obtained by the controller and if the values are not null, the value is matched with the usernames in the database. Since both employees and customers are using same login page, given username and password will be queried in the employee table first and if there are no records exist will check in customer table next. A user session is created if relevant record is found in either table and the status of the user is 'active'. The user is redirected to the dashboard.

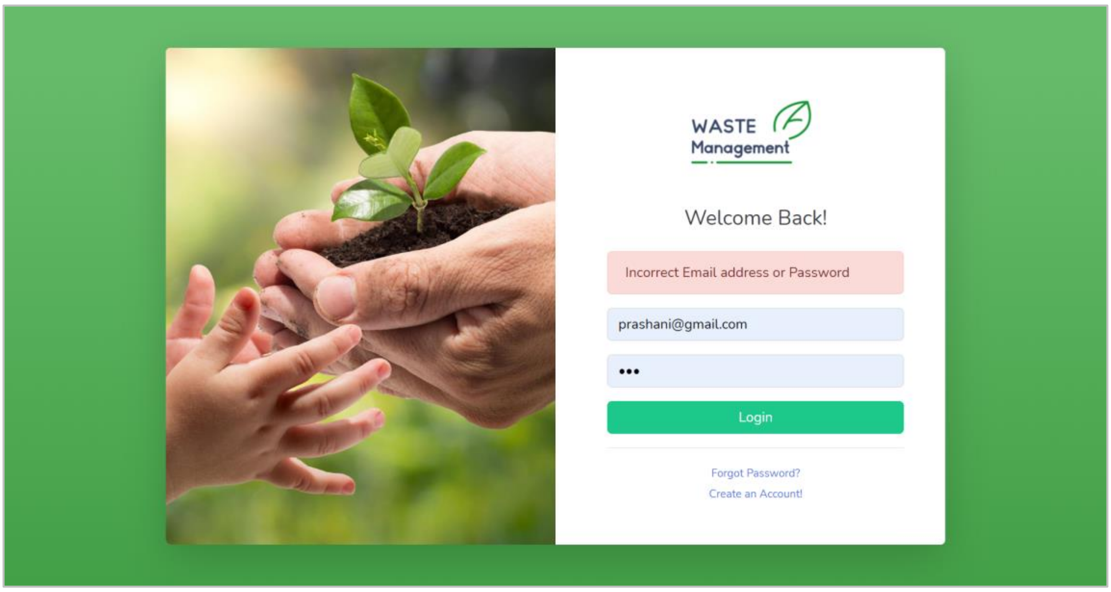


```
<?php
class login {
    public function loginvalidate($email, $pass) {
        $con = $GLOBALS['con'];
        //sql query
//emp login
        $sql = "SELECT * FROM employee e WHERE email='$email' AND
password='$pass' AND e.empStatus='active'";
        $result = $con->query($sql);
     //cus login
 if ($result->num_rows == 0) {
   $sql2 = "SELECT * FROM customer c WHERE email='$email' AND
password='$pass' AND c.cusStatus='active'";
    $result = $con->query($sql2);
}
        //Execute a query
        //$result = $con->query($sql);
        return $result;
    }
}
?>
```

Figure 4.5 - Login model code snippet

4.5.1.2 VIEW

The users are interacting with the View component which is presented to the users as login form. For client side validation form displays error messages if data are empty and server side validation if data are invalid.



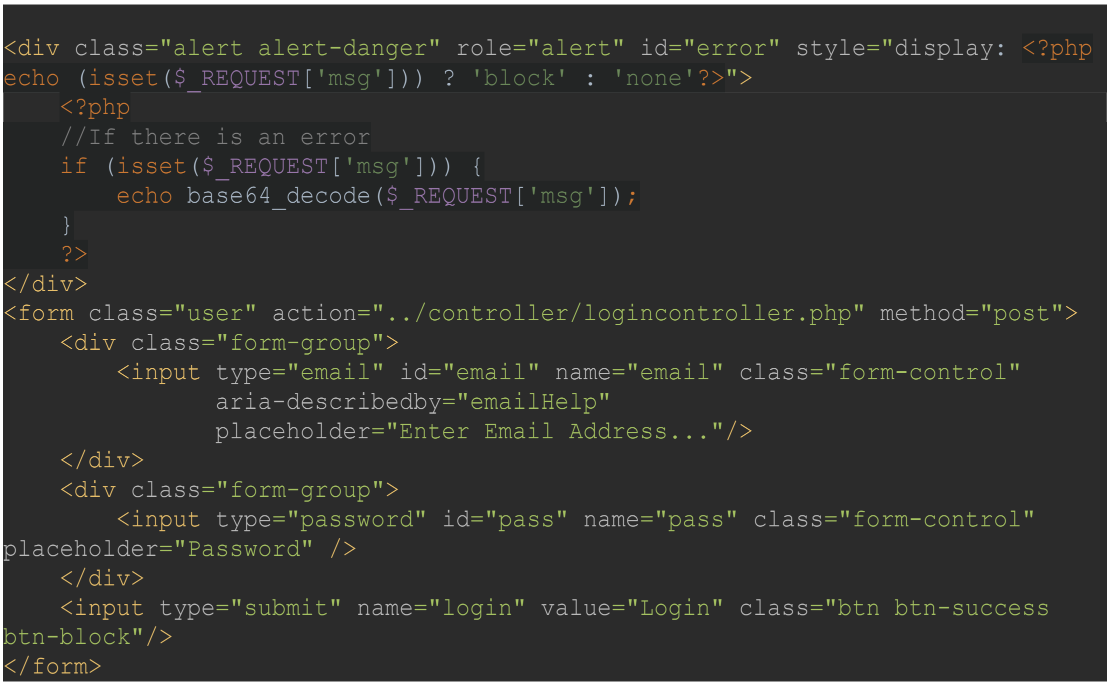
Welcome Back!

Incorrect Email address or Password

prashani@gmail.com

•••

Figure 4.6 - Login page

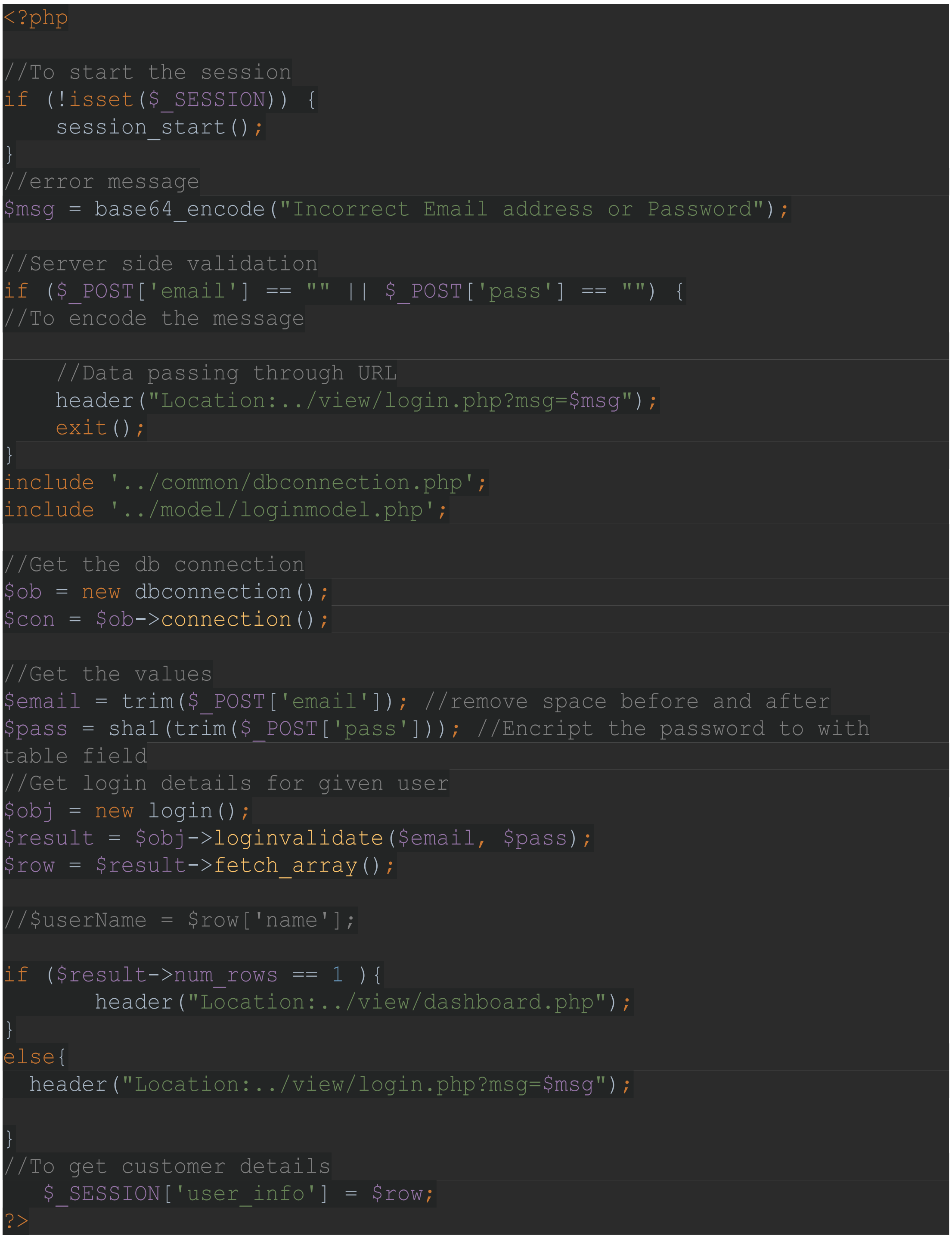


```
<div class="alert alert-danger" role="alert" id="error" style="display: <?php
echo (isset($_REQUEST['msg'])) ? 'block' : 'none'?>">
    <?php
    //If there is an error
    if (isset($_REQUEST['msg'])) {
        echo base64_decode($_REQUEST['msg']);
    }
    ?>
</div>
<form class="user" action="../controller/logincontroller.php" method="post">
    <div class="form-group">
        <input type="email" id="email" name="email" class="form-control"
               aria-describedby="emailHelp"
               placeholder="Enter Email Address..."/>
    </div>
    <div class="form-group">
        <input type="password" id="pass" name="pass" class="form-control"
placeholder="Password" />
    </div>
    <input type="submit" name="login" value="Login" class="btn btn-success
btn-block"/>
</form>
```

Figure 4.7 - Login view code snippet

4.5.1.3 CONTROLLER

The controller gets the username and password through the $_POST method while removing blank spaces in front and back of the email & password. After that it encrypt the password using SHA1 algorithm because passwords are saved in database encrypted using SHA1. Then it checks the results gets from login model. For the given username & password will be queried in the employee table first and if there are no records exist will check in customer table next. If matching record found in either of the table it adds user info to a session and let user to redirect to dashboard. If user name and password not matching with database table it shows a validation message "Invalid Email address or Password".

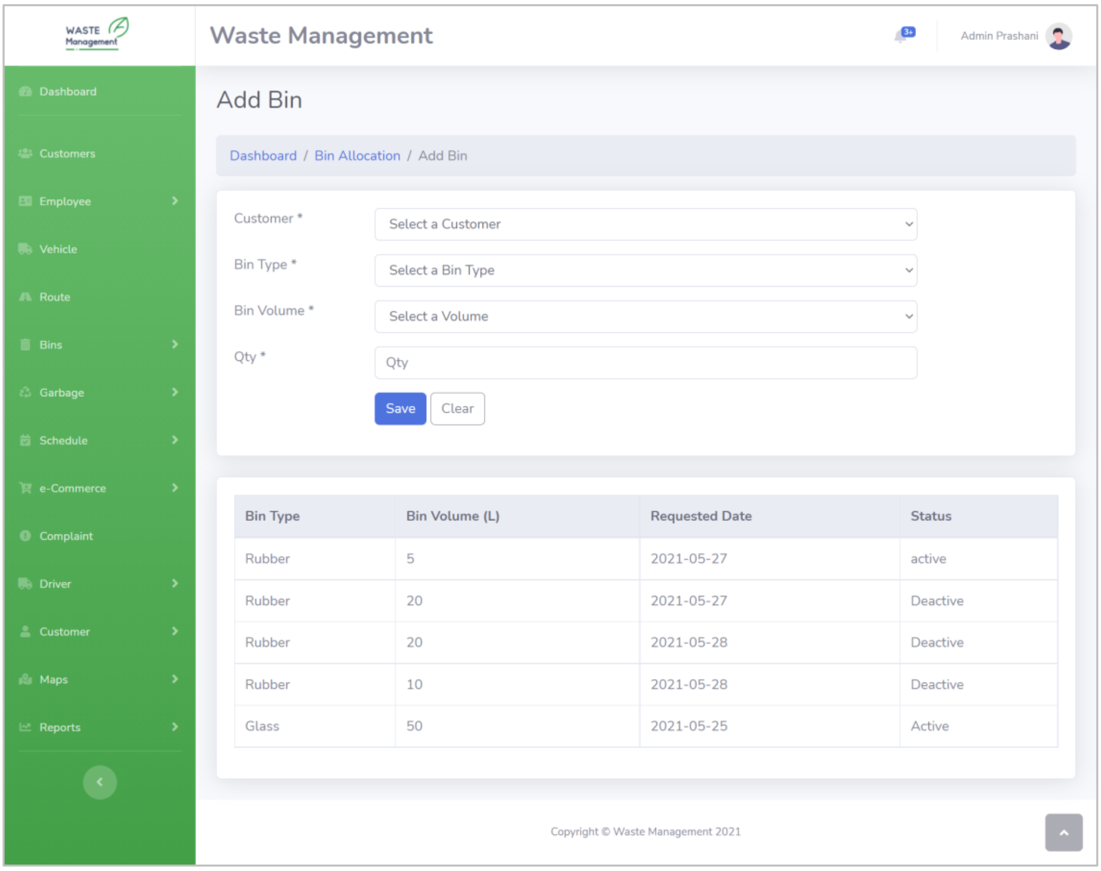


```
<?php

//To start the session
if (!isset($_SESSION)) {
    session_start();
}
//error message
$msg = base64_encode("Incorrect Email address or Password");

//Server side validation
if ($_POST['email'] == "" || $_POST['pass'] == "") {
//To encode the message

    //Data passing through URL
    header("Location:../view/login.php?msg=$msg");
    exit();
}
include '../common/dbconnection.php';
include '../model/loginmodel.php';

//Get the db connection
$ob = new dbconnection();
$con = $ob->connection();

//Get the values
$email = trim($_POST['email']); //remove space before and after
$pass = sha1(trim($_POST['pass'])); //Encript the password to with
table field
//Get login details for given user
$obj = new login();
$result = $obj->loginvalidate($email, $pass);
$row = $result->fetch_array();

//$userName = $row['name'];

if ($result->num_rows == 1 ){
       header("Location:../view/dashboard.php");
}
else{
  header("Location:../view/login.php?msg=$msg");

}
//To get customer details
   $_SESSION['user_info'] = $row;
?>
```

Figure 4.8 - Login controller code snippet

4.5.2 ALLOCATE BINS

Add bin is used to allocate bins to customers. When open the add bins page it shows all the alloacted bins for patricular customer and using form in top part of the page can assign a new bins to the customer. Figure 4.9 shows the staff view of the add bin page.



Add Bin

Dashboard / Bin Allocation / Add Bin

Customer *	Select a Customer
Bin Type *	Select a Bin Type
Bin Volume *	Select a Volume
Qty *	Qty

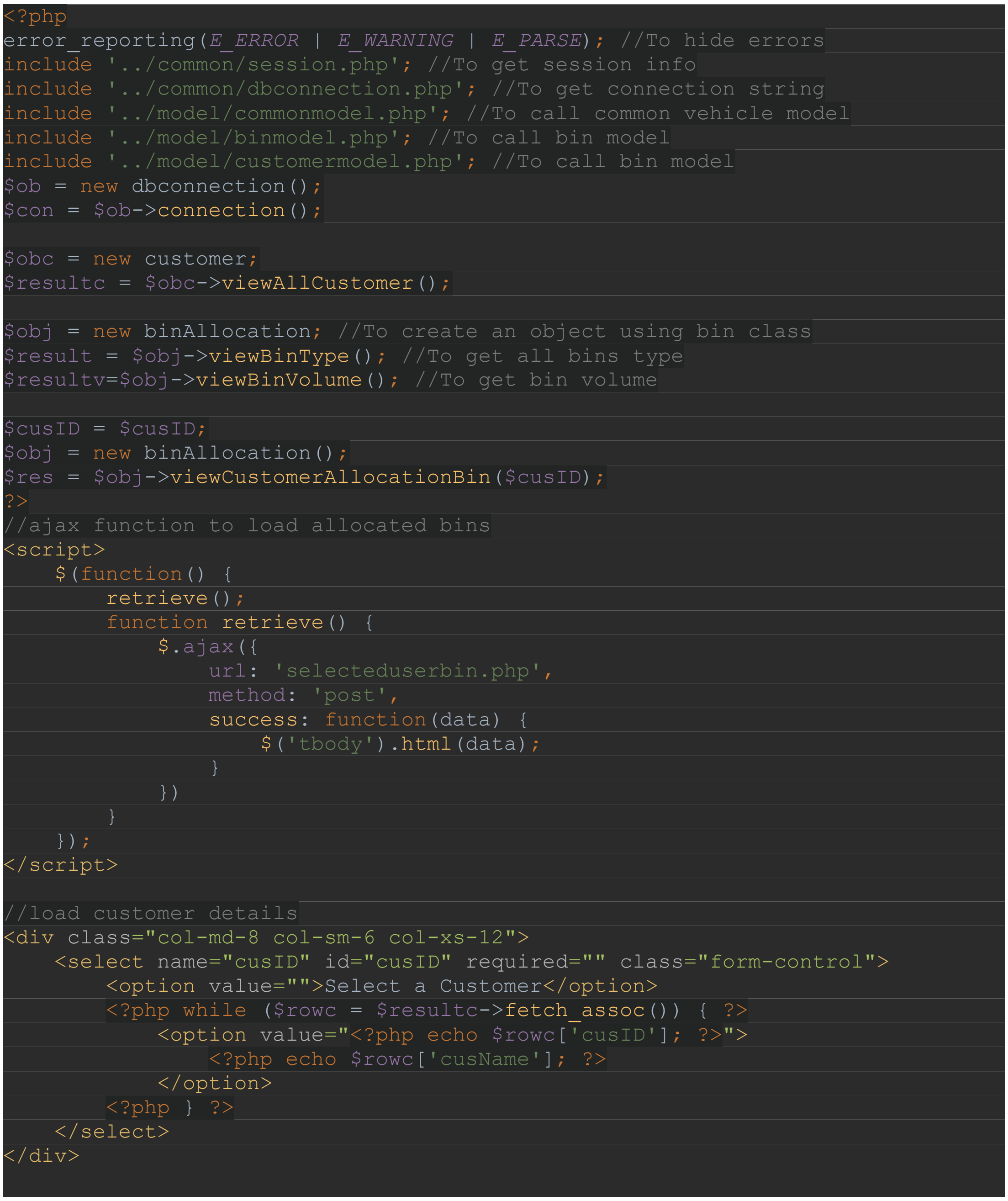
Save Clear

Bin Type	Bin Volume (L)	Requested Date	Status
Rubber	5	2021-05-27	active
Rubber	20	2021-05-27	Deactive
Rubber	20	2021-05-28	Deactive
Rubber	10	2021-05-28	Deactive
Glass	50	2021-05-25	Active

Copyright © Waste Management 2021

Figure 4.9 - Add bin page

Add bin page code snippet shows customers, bin types, volumes data loaded into page as objects and fetch into form's each dropdowns. When select a particular customer in dropdown, all allocated bins for that customer will be loaded into page using an Ajax function without refreshing the page.

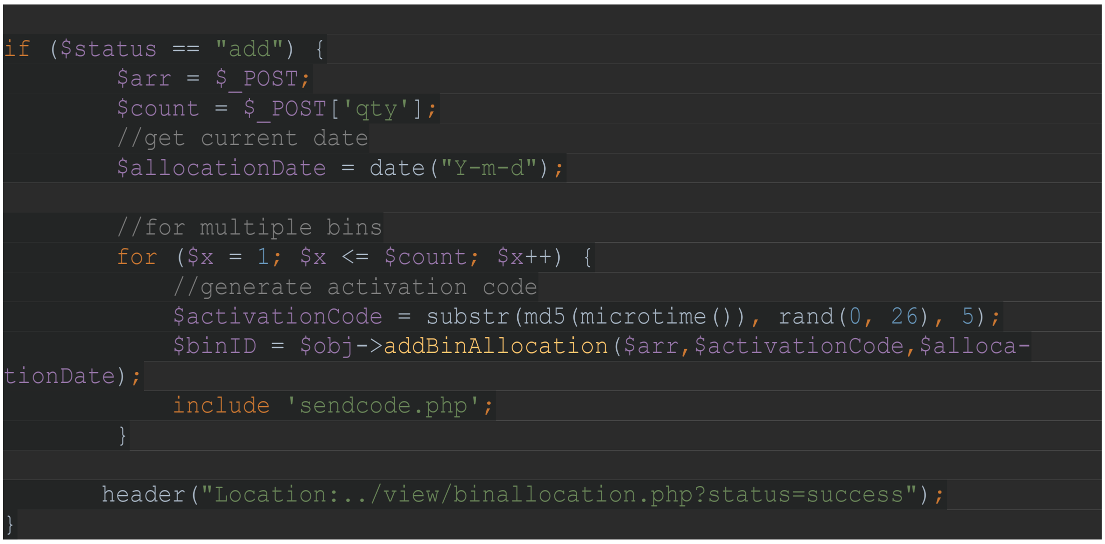


```
<?php
error_reporting(E_ERROR | E_WARNING | E_PARSE); //To hide errors
include '../common/session.php'; //To get session info
include '../common/dbconnection.php'; //To get connection string
include '../model/commonmodel.php'; //To call common vehicle model
include '../model/binmodel.php'; //To call bin model
include '../model/customermodel.php'; //To call bin model
$ob = new dbconnection();
$con = $ob->connection();

$obc = new customer;
$resultc = $obc->viewAllCustomer();

$obj = new binAllocation; //To create an object using bin class
$result = $obj->viewBinType(); //To get all bins type
$resultv=$obj->viewBinVolume(); //To get bin volume

$cusID = $cusID;
$obj = new binAllocation();
$res = $obj->viewCustomerAllocationBin($cusID);
?>
//ajax function to load allocated bins
<script>
    $(function() {
        retrieve();
        function retrieve() {
            $.ajax({
                url: 'selecteduserbin.php',
                method: 'post',
                success: function(data) {
                    $('tbody').html(data);
                }
            })
        }
    });
</script>

//load customer details
<div class="col-md-8 col-sm-6 col-xs-12">
    <select name="cusID" id="cusID" required="" class="form-control">
        <option value="">Select a Customer</option>
        <?php while ($rowc = $resultc->fetch_assoc()) { ?>
            <option value="<?php echo $rowc['cusID']; ?>">
                <?php echo $rowc['cusName']; ?>
            </option>
        <?php } ?>
    </select>
</div>
```

Figure 4.10 - Add bins code snippet

Bin controller page gets form data from add bin page using an array. It will generate an activation code by extracting a substring from MD5 hash of a current timestamp. Because of that it will generate a unique activation code at each time. According to the selected number of bins, unique activation codes will be generated and saved on database table. Email will be send to customer with other relevant details.

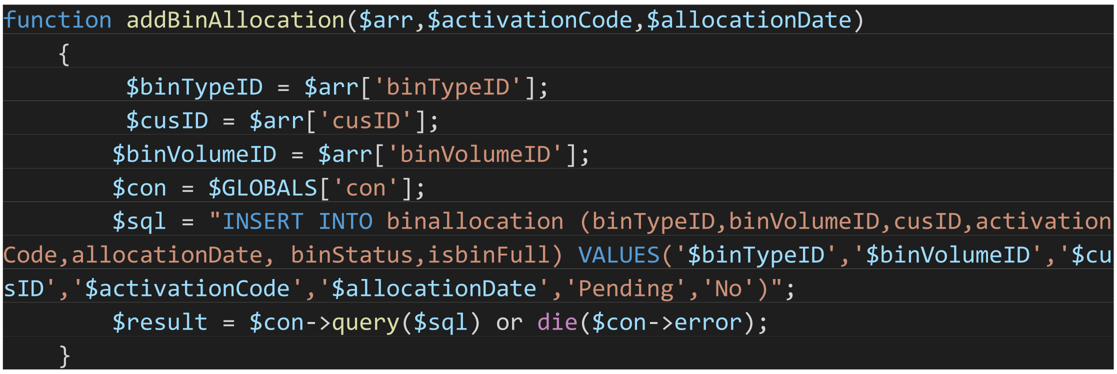
When customer add activation code to activate the bin, it will be compared against database saved code and if codes matches will activate the bin.



```
if ($status == "add") {
        $arr = $_POST;
        $count = $_POST['qty'];
        //get current date
        $allocationDate = date("Y-m-d");

        //for multiple bins
        for ($x = 1; $x <= $count; $x++) {
            //generate activation code
            $activationCode = substr(md5(microtime()), rand(0, 26), 5);
            $binID = $obj->addBinAllocation($arr,$activationCode,$alloca-
tionDate);
            include 'sendcode.php';
        }

       header("Location:../view/binallocation.php?status=success");
}
```

Figure 4.11 - Code snippet for bin controller – Add bin function

Bin model gets the data passed from controller and save the data into binallocation table. It saves bin status as pending and save activation code so when later customer can activate the bin by received code.

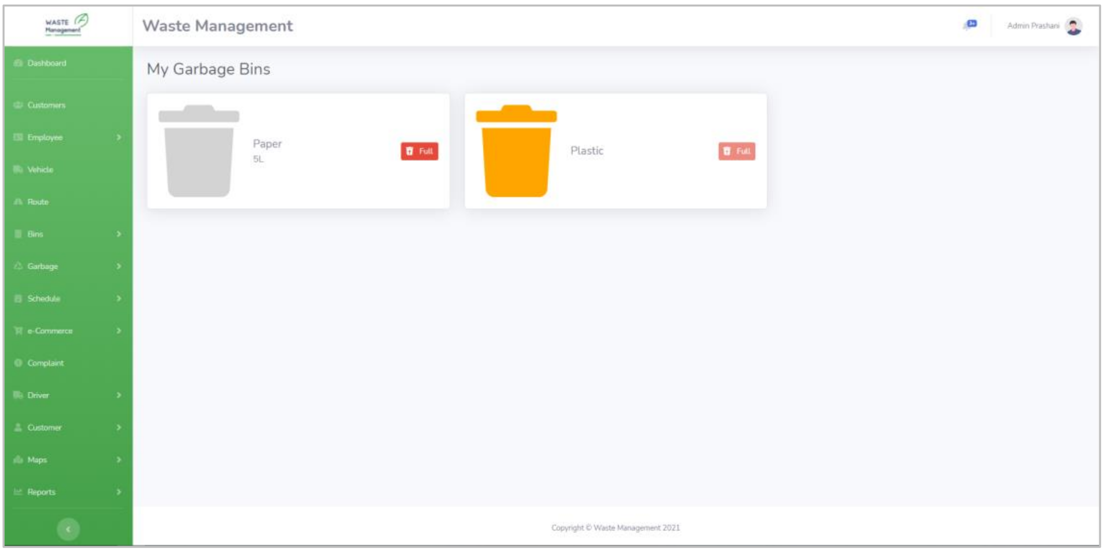


```
function addBinAllocation($arr,$activationCode,$allocationDate)
    {
         $binTypeID = $arr['binTypeID'];
         $cusID = $arr['cusID'];
        $binVolumeID = $arr['binVolumeID'];
        $con = $GLOBALS['con'];
        $sql = "INSERT INTO binallocation (binTypeID,binVolumeID,cusID,activation
Code,allocationDate, binStatus,isbinFull) VALUES('$binTypeID','$binVolumeID','$cu
sID','$activationCode','$allocationDate','Pending','No')";
        $result = $con->query($sql) or die($con->error);
    }
```

Figure 4.12 - Code snippet for bin model – Add bin function

4.5.3 UPDATE BIN STATUS

There are two ways to update bin status. First one is if the bin volume is higher than 80% of the bin size, IoT it will automatically update the bin staus. Other way is staff or customer can update bin status manually through the interface. This second option buit as fail-safe if IoT unit stopped working due to malfunction or fault.



My Garbage Bins

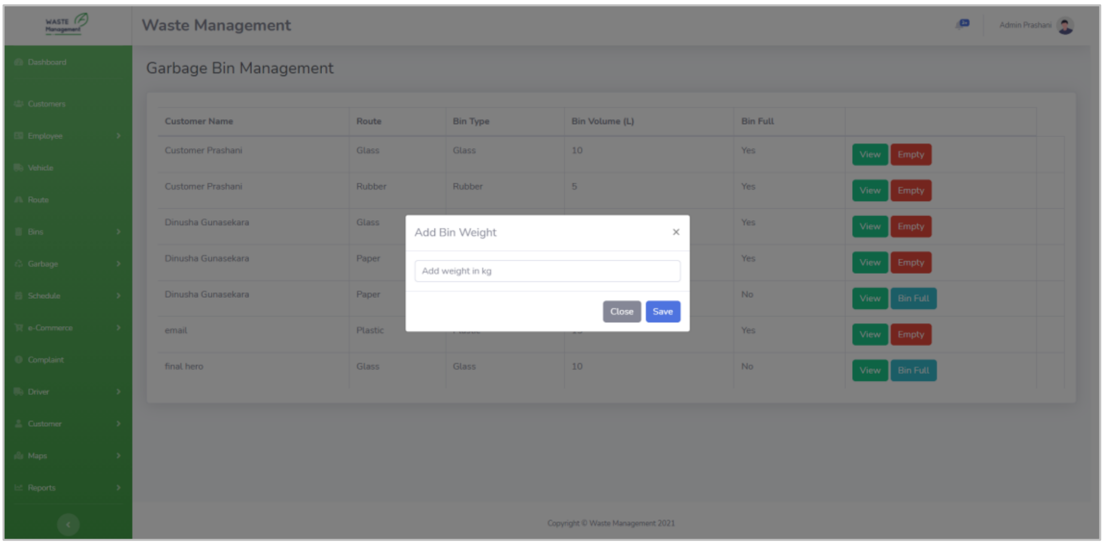
Paper
5L

Plastic

Figure 4.13 - Update bin status – Customer view

Above Figure 4.13 shows customer view of the update bin page. All active garbage bins will be shown under my garbage bins page. Bins that didn’t exceed the limit shows in grey color with bin full button. Customer can use “bin full” button to change the bin status as “full”. After bin marked as full bin icon color changes and button get disabled. Customer doesn’t have permission to change bin full status until driver or staff reset the status.

Below Figure 4.14 shows staff view of the change bin status page. Staff can change bin status as empty or full. When make bin status “empty”, pop-up dialog box appears to add the weight of the garbage as shown in Figure 4.15.



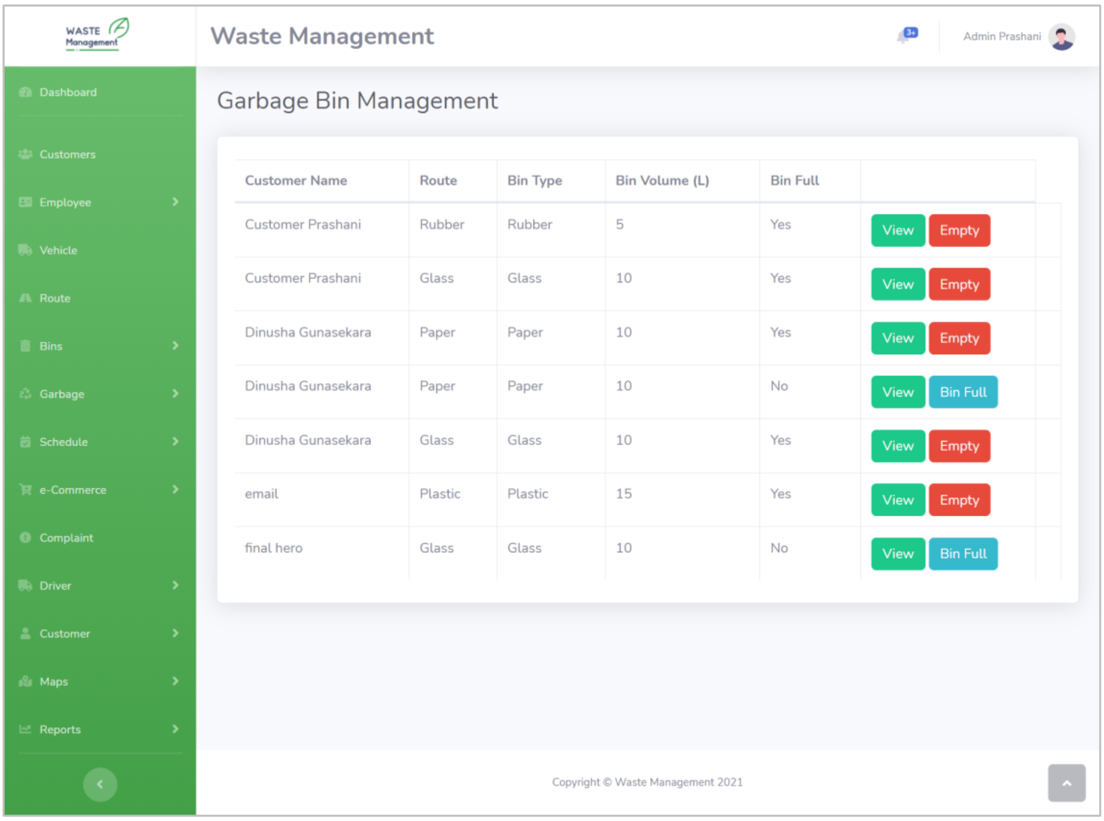
Garbage Bin Management

Customer Name	Route	Bin Type	Bin Volume (L)	Bin Full	
Customer Prashani	Glass	Glass	10	Yes	View Empty
Customer Prashani	Rubber	Rubber	5	Yes	View Empty
Dinusha Gunasekara	Glass			Yes	View Empty
Dinusha Gunasekara	Paper			Yes	View Empty
Dinusha Gunasekara	Paper			No	View Bin Full
email	Plastic	Plastic	10	Yes	View Empty
final hero	Glass	Glass	10	No	View Bin Full

Add Bin Weight

Add weight in kg

Figure 4.14 - Add bin weight pop-up



Garbage Bin Management

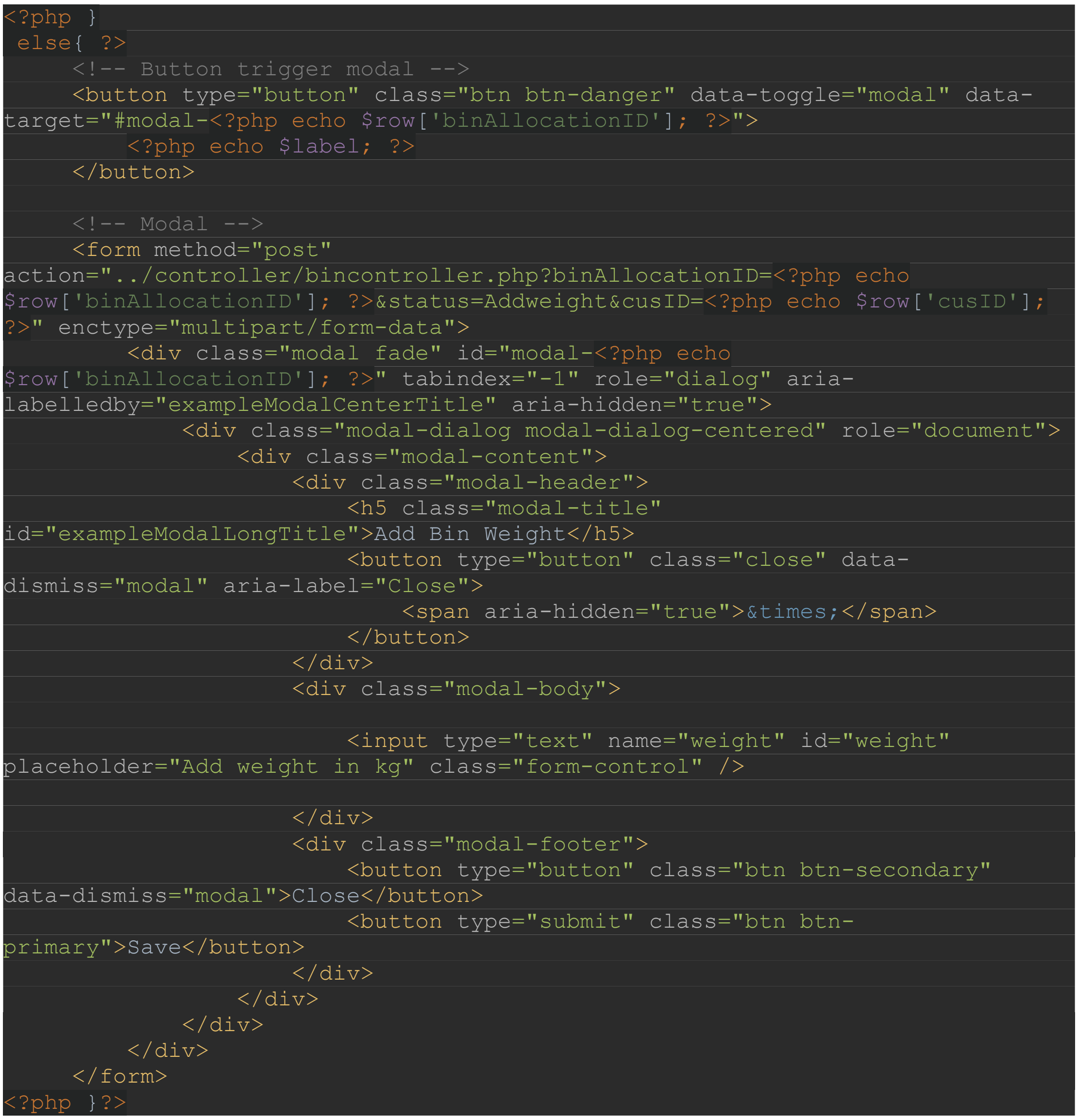
Customer Name	Route	Bin Type	Bin Volume (L)	Bin Full	
Customer Prashani	Rubber	Rubber	5	Yes	View Empty
Customer Prashani	Glass	Glass	10	Yes	View Empty
Dinusha Gunasekara	Paper	Paper	10	Yes	View Empty
Dinusha Gunasekara	Paper	Paper	10	No	View Bin Full
Dinusha Gunasekara	Glass	Glass	10	Yes	View Empty
email	Plastic	Plastic	15	Yes	View Empty
final hero	Glass	Glass	10	No	View Bin Full

Copyright © Waste Management 2021

Figure 4.15 - Update bin status – Staff view

Following Figure 4.16 code snippet shows in staff's change bin status page's how the status get changed and when make bin empty, how the modal appears as a pop up to add the garbage weight.

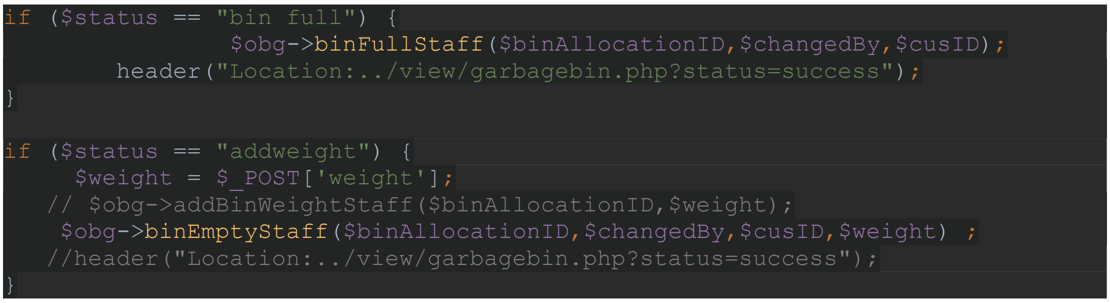
```
<?php
while ($row = $result->fetch_array()) {
    //To check the status
    if (strtolower($row['isbinFull']) == "yes") {
        $label = "Empty";
        $class = "btn btn-danger";
    } else {
        $label = "Bin Full";
        $class = "btn btn-info";
    }
    ?>
<?php if ($label=="Bin Full"){ ?>
     <a href="../controller/bincontroller.php?binAllocationID=<?php echo
$row['binAllocationID']; ?>&status=<?php echo $label; ?>&cusID=<?php echo
$row['cusID']; ?>">
         <button type="button" class="<?php echo $class; ?>">
             <?php echo $label; ?></button></a>
```



```
<?php }
 else{ ?>
     <!-- Button trigger modal -->
     <button type="button" class="btn btn-danger" data-toggle="modal" data-
target="#modal-<?php echo $row['binAllocationID']; ?>">
         <?php echo $label; ?>
     </button>

     <!-- Modal -->
     <form method="post"
action="../controller/bincontroller.php?binAllocationID=<?php echo
$row['binAllocationID']; ?>&status=Addweight&cusID=<?php echo $row['cusID'];
?>" enctype="multipart/form-data">
         <div class="modal fade" id="modal-<?php echo
$row['binAllocationID']; ?>" tabindex="-1" role="dialog" aria-
labelledby="exampleModalCenterTitle" aria-hidden="true">
             <div class="modal-dialog modal-dialog-centered" role="document">
                 <div class="modal-content">
                     <div class="modal-header">
                         <h5 class="modal-title"
id="exampleModalLongTitle">Add Bin Weight</h5>
                         <button type="button" class="close" data-
dismiss="modal" aria-label="Close">
                             <span aria-hidden="true">&times;</span>
                         </button>
                     </div>
                     <div class="modal-body">

                         <input type="text" name="weight" id="weight"
placeholder="Add weight in kg" class="form-control" />

                     </div>
                     <div class="modal-footer">
                         <button type="button" class="btn btn-secondary"
data-dismiss="modal">Close</button>
                         <button type="submit" class="btn btn-
primary">Save</button>
                     </div>
                 </div>
             </div>
         </div>
     </form>
<?php }?>
```

Figure 4.16 - Code snippet for change bin status page

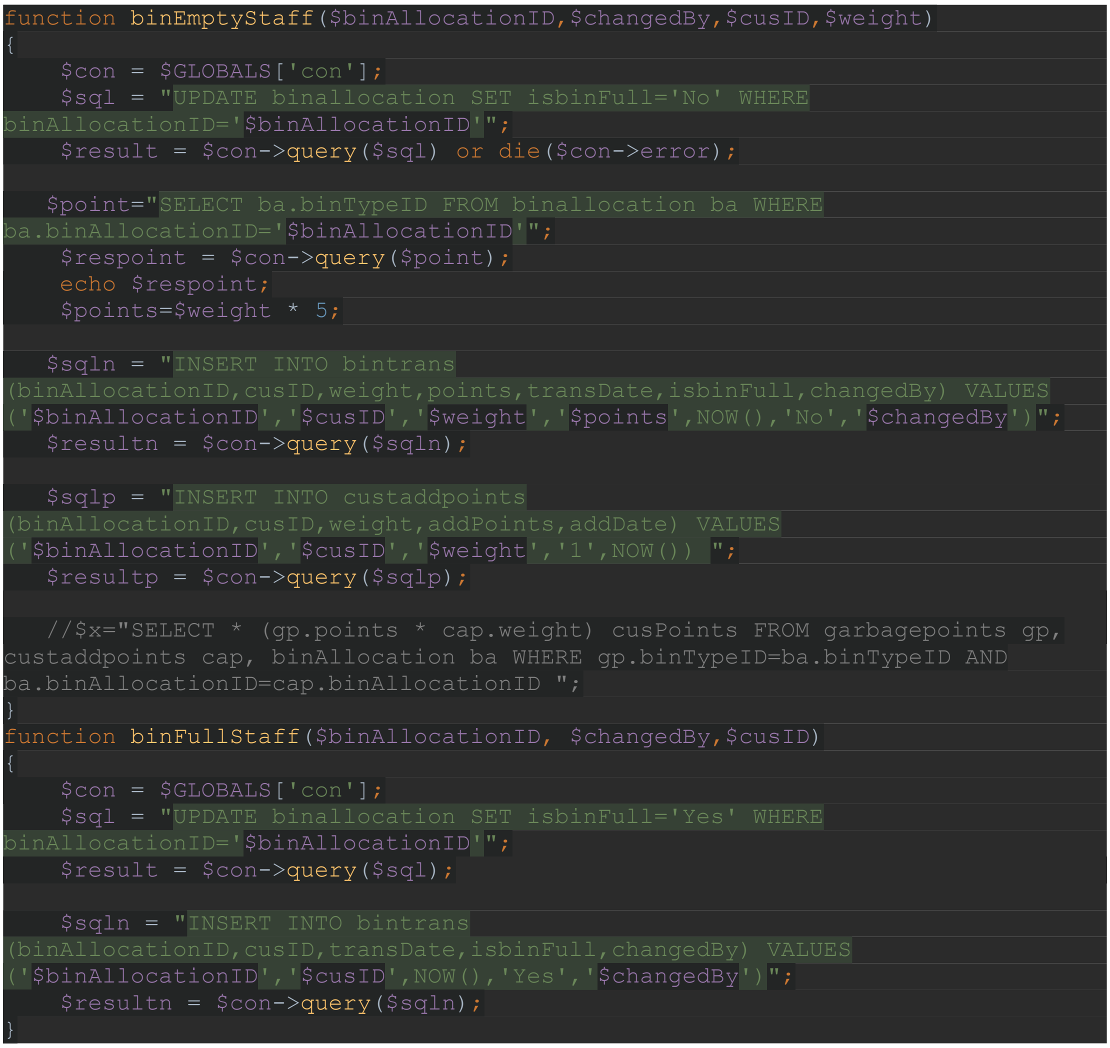
Below code snippet shows how the bin full status and add weight/bin empty functions handled in bin controller.



```
if ($status == "bin full") {
                $obg->binFullStaff($binAllocationID,$changedBy,$cusID);
        header("Location:../view/garbagebin.php?status=success");
}

if ($status == "addweight") {
     $weight = $_POST['weight'];
   // $obg->addBinWeightStaff($binAllocationID,$weight);
    $obg->binEmptyStaff($binAllocationID,$changedBy,$cusID,$weight) ;
   //header("Location:../view/garbagebin.php?status=success");
}
```

Figure 4.17 - Code snippet for bin controller – Bin status functions

Model page shows how the bin status change will be update in database tables. When bin status changed there are multiple tables get updated naming binallocation, bintrans, custaddpoints. Greenpoints will be added to customer when bin get empty along with the garbage weight.

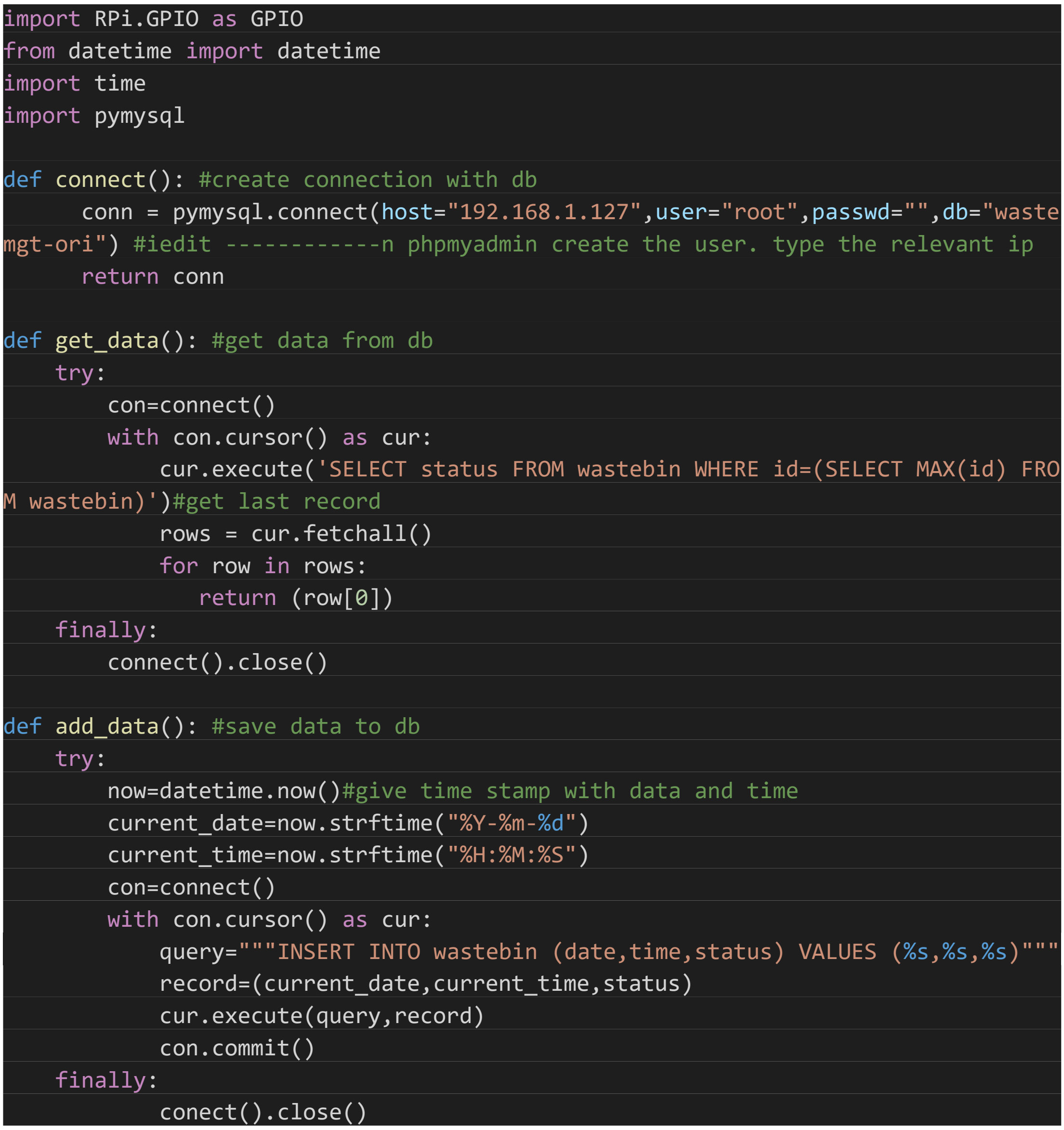


```
function binEmptyStaff($binAllocationID,$changedBy,$cusID,$weight)
{
    $con = $GLOBALS['con'];
    $sql = "UPDATE binallocation SET isbinFull='No' WHERE
binAllocationID='$binAllocationID'";
    $result = $con->query($sql) or die($con->error);

   $point="SELECT ba.binTypeID FROM binallocation ba WHERE
ba.binAllocationID='$binAllocationID'";
    $respoint = $con->query($point);
    echo $respoint;
    $points=$weight * 5;

   $sqln = "INSERT INTO bintrans
(binAllocationID,cusID,weight,points,transDate,isbinFull,changedBy) VALUES
('$binAllocationID','$cusID','$weight','$points',NOW(),'No','$changedBy')";
   $resultn = $con->query($sqln);

   $sqlp = "INSERT INTO custaddpoints
(binAllocationID,cusID,weight,addPoints,addDate) VALUES
('$binAllocationID','$cusID','$weight','1',NOW()) ";
   $resultp = $con->query($sqlp);

   //$x="SELECT * (gp.points * cap.weight) cusPoints FROM garbagepoints gp,
custaddpoints cap, binAllocation ba WHERE gp.binTypeID=ba.binTypeID AND
ba.binAllocationID=cap.binAllocationID ";
}
function binFullStaff($binAllocationID, $changedBy,$cusID)
{
    $con = $GLOBALS['con'];
    $sql = "UPDATE binallocation SET isbinFull='Yes' WHERE
binAllocationID='$binAllocationID'";
    $result = $con->query($sql);

    $sqln = "INSERT INTO bintrans
(binAllocationID,cusID,transDate,isbinFull,changedBy) VALUES
('$binAllocationID','$cusID',NOW(),'Yes','$changedBy')";
    $resultn = $con->query($sqln);
}
```

Figure 4.18 - Code snippet for bin model – Bin status functions

Following code snippets shows in how python script is written in IoT unit to update bin status.

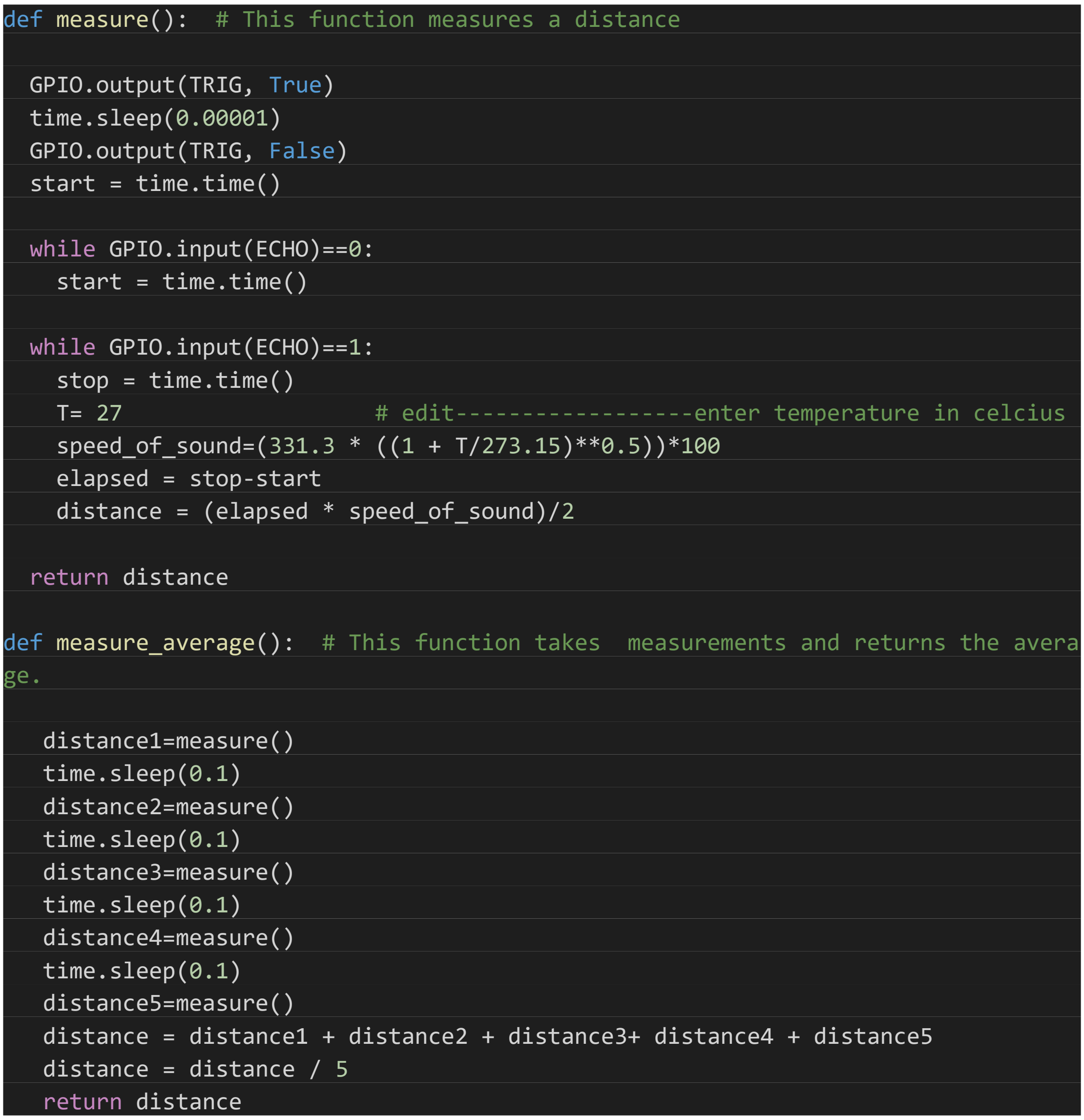
Connection is established to the mysql database through the user defined connect function. get_data() function is used to extract the latest bin status saved in the database. User defined add_data() function is used to save the status of the bin together with the current date and time which had been obtained from python built in function.



```
import RPi.GPIO as GPIO
from datetime import datetime
import time
import pymysql

def connect(): #create connection with db
      conn = pymysql.connect(host="192.168.1.127",user="root",passwd="",db="waste
mgt-ori") #iedit ------------n phpmyadmin create the user. type the relevant ip
      return conn

def get_data(): #get data from db
    try:
        con=connect()
        with con.cursor() as cur:
            cur.execute('SELECT status FROM wastebin WHERE id=(SELECT MAX(id) FRO
M wastebin)')#get last record
            rows = cur.fetchall()
            for row in rows:
               return (row[0])
    finally:
        connect().close()

def add_data(): #save data to db
    try:
        now=datetime.now()#give time stamp with data and time
        current_date=now.strftime("%Y-%m-%d")
        current_time=now.strftime("%H:%M:%S")
        con=connect()
        with con.cursor() as cur:
            query="""INSERT INTO wastebin (date,time,status) VALUES (%s,%s,%s)"""
            record=(current_date,current_time,status)
            cur.execute(query,record)
            con.commit()
    finally:
            conect().close()
```

Figure 4.19 - Python code snippet – Part I

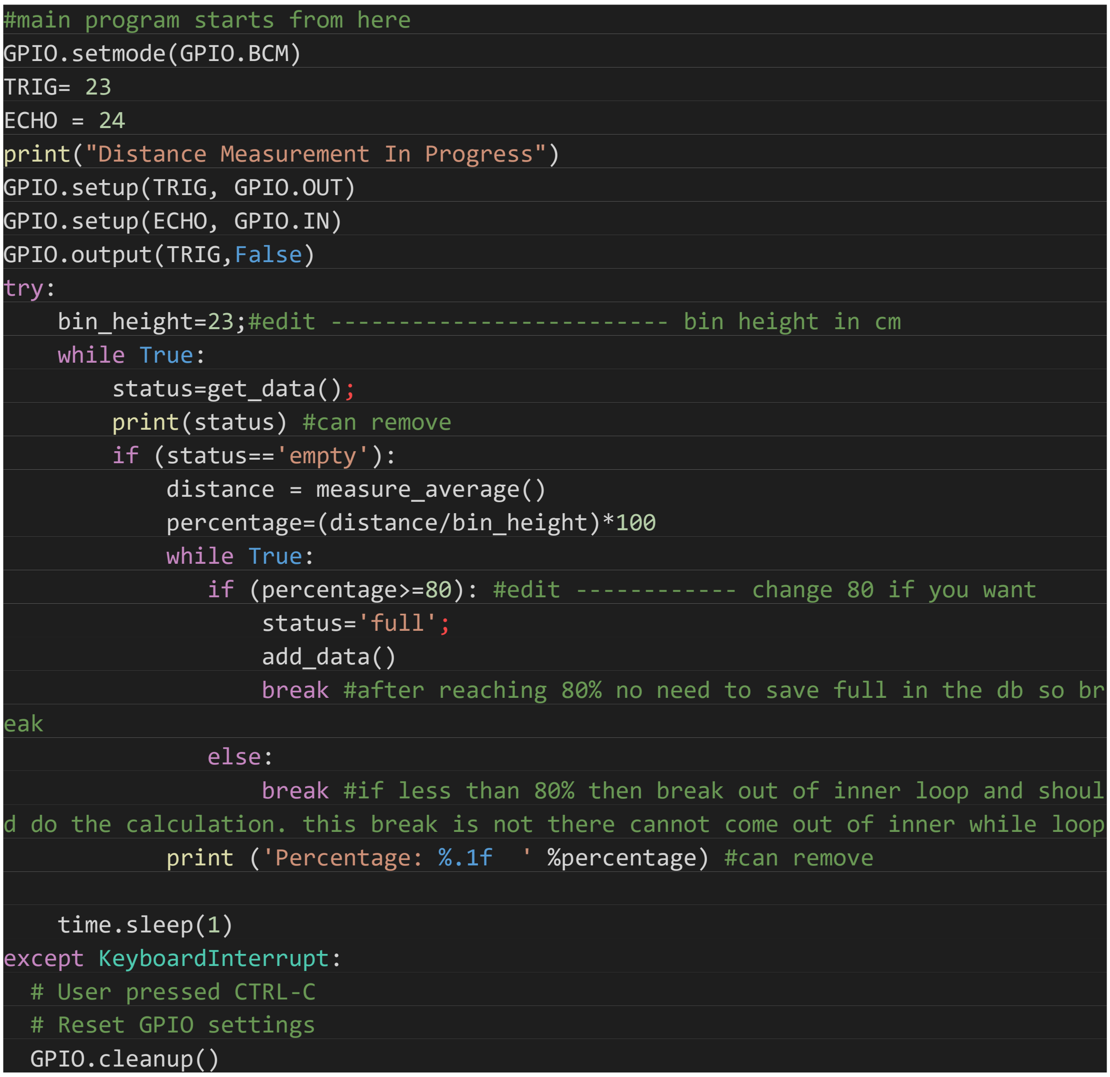
In Figure 4.20 shows measure function is used to measure the distance from the trash in the waste bin to the bin lid. An ultrasonic module is used to get the time it takes for an ultrasonic pulse to hit the trash and bounce back. Then this value had been used to calculate the distance the pulse travelled. measure_average() function is used to calculate the average of five distance measurements to get more precise measurements.



```
def measure():  # This function measures a distance

  GPIO.output(TRIG, True)
  time.sleep(0.00001)
  GPIO.output(TRIG, False)
  start = time.time()

  while GPIO.input(ECHO)==0:
    start = time.time()

  while GPIO.input(ECHO)==1:
    stop = time.time()
    T= 27                   # edit------------------enter temperature in celcius
    speed_of_sound=(331.3 * ((1 + T/273.15)**0.5))*100
    elapsed = stop-start
    distance = (elapsed * speed_of_sound)/2

  return distance

def measure_average():  # This function takes  measurements and returns the avera
ge.

   distance1=measure()
   time.sleep(0.1)
   distance2=measure()
   time.sleep(0.1)
   distance3=measure()
   time.sleep(0.1)
   distance4=measure()
   time.sleep(0.1)
   distance5=measure()
   distance = distance1 + distance2 + distance3+ distance4 + distance5
   distance = distance / 5
   return distance
```

Figure 4.20 - Python code snippet – Part II

In Figure 4.21 shows if the bin is empty only the distance calculation is required to so initially the status of the bin is retrieved from the database. Then the distance to the trash is calculated which is used to get the percentage filled. If it is more than 80% then bin will be in full status. That data is saved in the database which then will be notified to the staff.



```
#main program starts from here
GPIO.setmode(GPIO.BCM)
TRIG= 23
ECHO = 24
print("Distance Measurement In Progress")
GPIO.setup(TRIG, GPIO.OUT)
GPIO.setup(ECHO, GPIO.IN)
GPIO.output(TRIG,False)
try:
    bin_height=23;#edit ------------------------- bin height in cm
    while True:
        status=get_data();
        print(status) #can remove
        if (status=='empty'):
            distance = measure_average()
            percentage=(distance/bin_height)*100
            while True:
               if (percentage>=80): #edit ------------ change 80 if you want
                   status='full';
                   add_data()
                   break #after reaching 80% no need to save full in the db so br
eak
               else:
                   break #if less than 80% then break out of inner loop and shoul
d do the calculation. this break is not there cannot come out of inner while loop
            print ('Percentage: %.1f  ' %percentage) #can remove

    time.sleep(1)
except KeyboardInterrupt:
  # User pressed CTRL-C
  # Reset GPIO settings
  GPIO.cleanup()
```

Figure 4.21 - Python code snippet – Part III

Below image shows the how the ultrasonic sensor connected to Raspberry Pi through the breadboard.

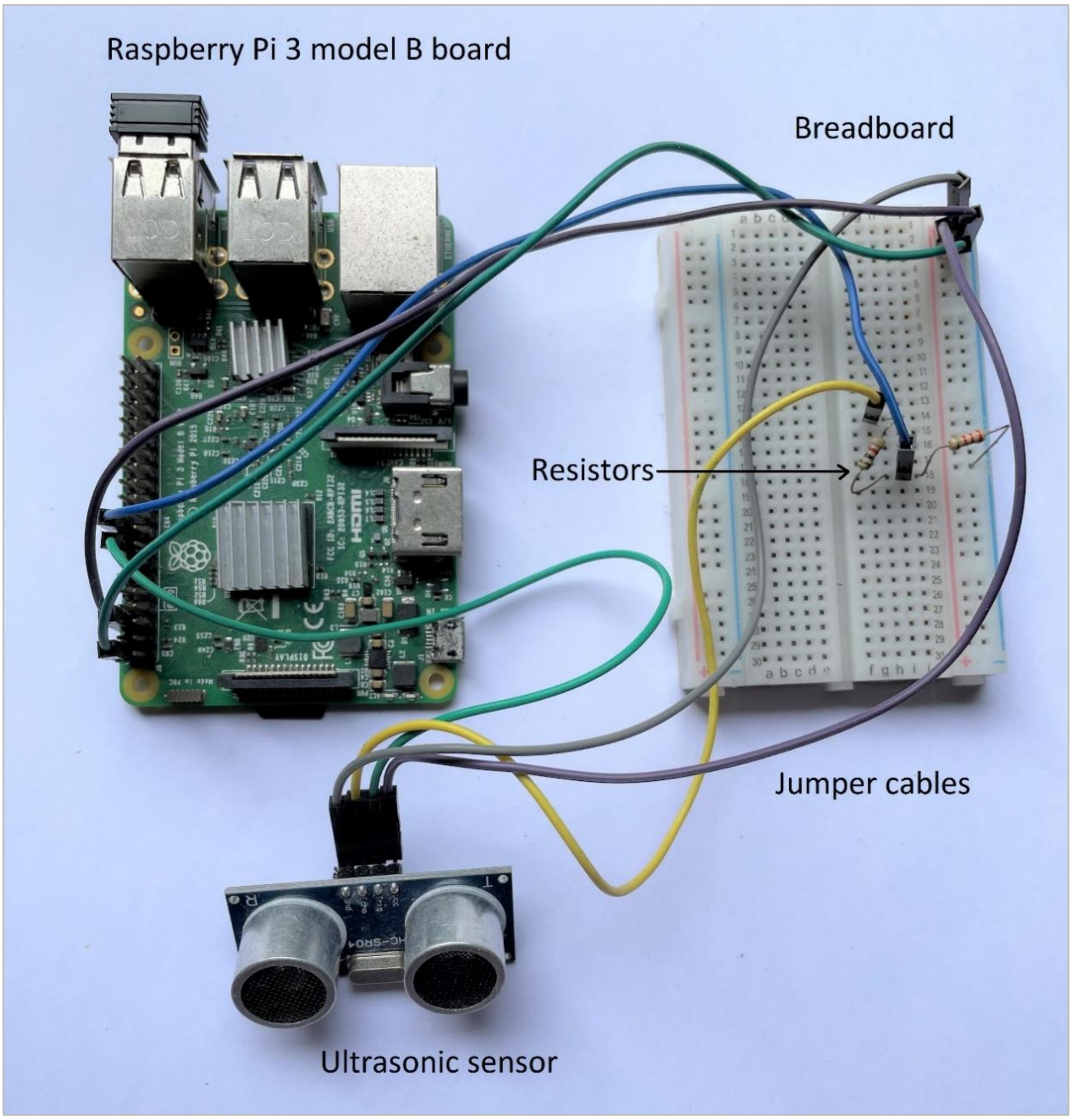


Figure 4.22 - Photo of IoT unit circuit setup

The other important code segments are listed in Appendix G.

TESTING & EVALUATION

Software testing involves the inspection (static testing) or execution (dynamic testing) of a software component with the intent to find any software bugs. A software evaluation is a type of assessment that seeks to determine if software or a combination of software programs is the best possible fit for the needs of a given client (Madhivanan, 2017). The system was tested to find errors and to ensure that it is free from bugs. Then the software was evaluated by potential users to check whether how much the original intended goals had been achieved.

5.1 TESTING

With an understanding on the importance of level 4 in the test process maturity levels introduced by Beizer (Paul Ammann, 2016) developing a high quality software was the goal of testing the system. Thus the testing was done to measure some aspects of its quality such as correctness, performance and usability.

5.1.1 TEST STRATEGY

The project is using an agile approach, with incremental way. At the end of each increment the requirements identified for that time box delivered and tested.

The testing approach was black-box & database level testing. Black box testing was used to check the system from the users' perspective in order to validate the system against the pre-defined requirements. Database level testing was performed to check data integrity and consistency. Test cases & Test designs were created after finishing the requirement analysis phase, and evolved further during design phase and executed during testing phase.

5.1.2 TEST PLAN

The objective of testing is to verify that the functionality of Smart System for Waste Management works according to the requirements. The scope of the system was defined

, test types were identified, test criteria were identified and finally test deliverables were produced. The detailed test plan is presented in Appendix E.

5.1.3 TEST TYPES

In this project, three types of Functional testing was performed to check the functions of application.

- Unit Testing

In the agile approach unit testing is emphasized heavily (Paul Ammann, 2016) thus methods were tested by passing parameters and verifying results. White box testing technique like Decision coverage was used to check the Boolean expressions of the source code. The modules in the system such as scheduling module, e-Commerce module ,report module, notification module was tested in the incremental method.

- Integration Testing

Individual software modules were combined and tested as a group. For integration incremental testing approach was used with bottom up testing strategy. The diagrammatic representation of module integration is presented in Figure 5.1.

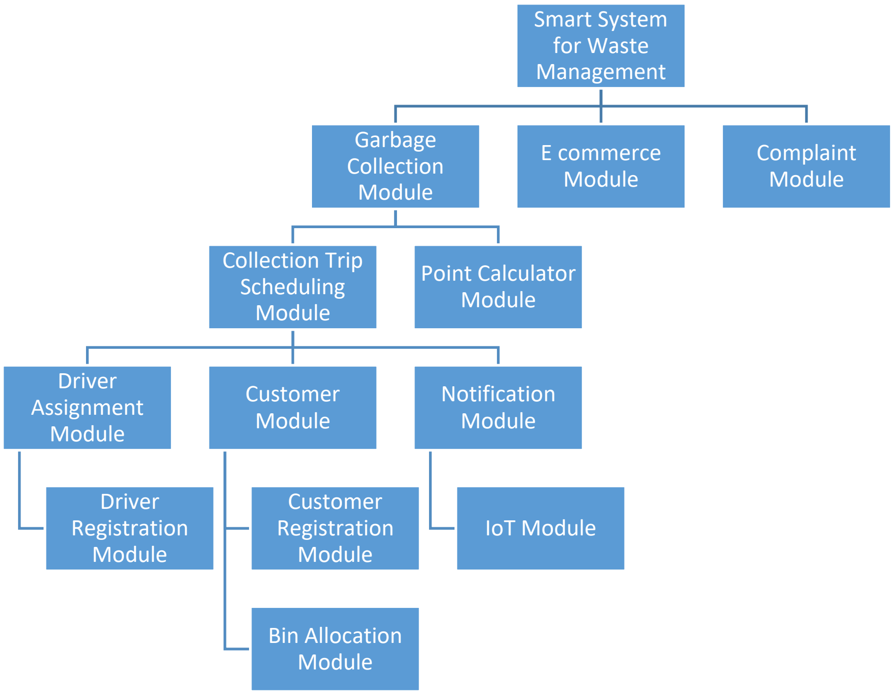

Smart System for Waste Management
Garbage Collection Module
E commerce Module
Complaint Module
Collection Trip Scheduling Module
Point Calculator Module
Driver Assignment Module
Customer Module
Notification Module
Driver Registration Module
Customer Registration Module
IoT Module
Bin Allocation Module

Figure 5.1 - Module integration

- System Testing

Application as a whole is tested for its compliance to functional and non-functional requirements.

5.1.4 AUTOMATION TESTING

Many Software Engineering projects use automation testing to have faster feedback cycle, increase the test coverage and reduce repetitive work done manually. There are many automation frameworks are available such as Selenium, Cypress, Playwright, Webdriver.io but building an automation framework takes time, resources. Apart from that test cases needs to added and maintained regularly to get good ROI from the framework. For success in automation validation, it should have predictable results to produce pass and fail conditions. In Agile projects automation testing mostly use as regression suites & Integrated into CI/CD pipelines to get quick feedback and verify changes haven’t introduced new bugs.

The results of the cost benefit analysis revealed that conducting manual testing for a one time project like this is much more cost effective when compared with automation testing. For example perform manual testing on GUI seems more quicker and more reliable than doing building an automation framework and conducting tests. This project requires exploratory testing and domain knowledge based ad hoc testing.

5.2 TEST CASES

A test case is a set of test inputs, execution conditions, and expected results developed for a particular objective (Anon., 2021). System functionality could be tested with properly planned test cases. The system was decomposed in to modules to reduce the complexity and to minimize the dependency. Test cases of major modules are presented below. (Refer Appendix F for the test case results)

5.2.1 TEST CASES FOR CUSTOMER MODULE

TC ID	Priority	Severity	Description	Test Steps	Expected Result
#1	P1	S1	Verify user able to login with a valid username and valid password.	1. User go to login page 2. Enter valid username and password 3. Click login button	User can login into the dashboard
#2	P1	S1	Verify user cannot login with an invalid username and an invalid password.	1. User go to login page 2. Enter invalid username and password 3. Click login button	1. User unable to login 2. Error message should appears
#3	P1	S1	Verify when the username and password field is blank.	1. User go to login page 2. Keep username and password fields blank 3. Click login button	1. User unable to login 2. Error msg should appears
#4	P2	S1	Verify the messages for invalid login	1. User go to login page 2. Enter invalid username and password 3. Click login button	1. Content of the error msg should be correct
#5	P3	S2	Verify if the data in password field is either visible as asterisk or bullet signs.	1. User go to login page 2. Enter valid username and password	1. data in password field is should visible as asterisk or bullet signs.
#6	P4	S4	Verify UI of the Login page	1. User go to login page	UI of the page should be correctly placed & sized/colored

Table 5.1 - Test cases for customer login

TC ID	Priority	Severity	Description	Test Steps	Expected Result
#1	P1	S1	Verify user able to add customer name	1. User go to registration page 2. Add name	1. User can add customer name
#2	P1	S1	Verify user able to add contact No	1. User go to registration page 2. Add contact no	1. User can add contact No
#3	P1	S1	Verify user able to add email	1. User go to registration page 2. Add email	1. User can add email
#4	P1	S1	Verify user able to add address	1. User go to registration page	1. User can add address

				2. Add address	
#5	P1	S1	Verify user able to add longitude	1. User go to registration page 2. Add longitude	1. User can add longitude
#6	P1	S1	Verify user able to add latitude	1. User go to registration page 2. Add latitude	1. User can add latitude
#7	P3	S3	Verify the clear button	1. User go to registration page 2. Add content 3. Click clear button	1. Field data should get reset
#8	P1	S1	Verify Validations for compulsory fields	1. User go to registration page 2. Check compulsory field validation for all required fields	1. Unable to proceed further if compulsory fields are missing 2. Validation message should shows
#9	P1	S1	Verify after creating a profile staff can view it	1. Go to staff login 2. Go to customer 3. Open created profile	1. Staff can view the created profile with correct details
#10	P1	S1	Verify after creating a profile user can login to it	1. Create a profile 2. Login to system using created profile details	1. User can login to the profile

Table 5.2 - Test cases for customer creation

5.2.2 TEST CASES FOR GARBAGE MODULE

TC ID	Priority	Severity	Description	Test Steps	Expected Result
#1	P1	S1	Staff can view past transactions related to one bin	1. go to garbage bin management window 2. click on view button	able to view past transactions related to one bin
#2	P2	S2	Customer can view past transactions related to one bin	1. go to my bins window 2. click on view button	able to view past transactions related to one bin
#3	P1	S1	Customer can able to make bin full by clicking a button	1. go to my bins window 2. click on bin full button	1. Bin status changed to full 2. Staff receive a notification
#4	P1	S1	Customer can able to make bin full by using IoT unit	1. add garbage to basket until its full	1. Bin status changed to full 2. Staff receive a notification
#5	P1	S1	Staff can able to make bin full by clicking a button	1. go to garbage bin management window 2. click on bin full button	1. Bin status changed to full 2. Staff receive a notification

#6	P1	S1	Staff can able to make bin empty by clicking a button	1. go to garbage bin management window 2. click on bin empty button 3. add weight details	1. Bin status changed to empty 2. weight details can add 3. weight information will be showing in transaction, green points pages
#7	P1	S1	After driver submitting bin weight bin status changed to empty	1. go to garbage bin management window 2. click on bin empty button 3. add weight details	1. Bin status changed to empty 2. weight details can add 3. weight information will be showing in transaction, green points pages

Table 5.3 - Test cases for garbage module

5.2.3 TEST CASES FOR BIN ALLOCATION MODULE

TC ID	Priority	Severity	Description	Test Steps	Expected Result
#1	P1	S1	Customer can request a bin	1. login to system 2. go to bin allocation page 3. click on request a bin button 4. fill the form & save details	1. chose bin should shows under customer my bins window 2. bin status should shows as pending
#2	P1	S1	Staff can assign a bin to a customer	1. login to system 2. go to bin allocation page 3. click on add a bin button 4. fill the form & save details	1. chose bin should shows under customer my bins window 2. newly added bin details should show in bin allocation details page 3. bin status should shows as pending
#3	P2	S2	When requesting a bin previously allocated bin list shows	1. login to system 2. go to bin allocation page 3. click on add a bin button	1. previously added bin details shows in a table
#4	P2	S1	View allocated bins to a given customer	1. login to system 2. go to bin allocation page 3. click on view button	1. all the allocated bin details should show
#5	P1	S1	Active a bin by customer	1. go to my bin page	1. bin status should change to active

				2. click on activate bin button 3. enter activation code received	2. customer able to start using the bin
#6	P1	S1	Try to activate bin by wrong activation code	1. go to my bin page 2. click on activate bin button 3. enter invalid activation code	1. validation message should be appears 2. bin should be in pending status
#7	P1	S1	Active a bin by staff	1. go to bin allocation page 2. click on active button	1. staff should be able to activate bin without activation code
#8	P1	S1	Deactivate a bin by staff	1. go to bin allocation page 2. click on Deactivate button	1. staff should be able to deactivate bin

Table 5.4 - Test cases for bin allocation

5.2.4 TEST CASES FOR E-COMMERCE MODULE

TC ID	Priority	Severity	Description	Test Steps	Expected Result
#1	P1	S1	Customer can view product list	1. go to e-shop	can view product list
#2	P1	S1	Customer can view a product detail	1. go to e-shop 2. click on a product	can view a product detail page
#3	P1	S1	Customer can add a product to a cart	1. go to e-shop 2. click on a product 3. select size/color/qty 4. click add to cart	Product can add to cart
#4	P1	S1	Customer can add multiple same product to a cart	1. go to e-shop 2. click on a product 3. select multiple qty 4. click add to cart	Same product can add to cart multiple times
#5	P1	S1	View shopping cart after adding product	1. go to e-shop 2. click on a product 3. select size/color/qty 4. click add to cart	All added products are in the cart
#6	P2	S1	Continue shopping by adding more items to the cart	1. go to e-shop 2. click on a product 3. select size/color/qty 4. click add to cart 5. go to cart window 6. click continue shopping button	User will be redirect to product list page
#7	P2	S1	Remove added items	1. add items to the cart	Removed products should get removed from the cart

				2. click on remove button	
#8	P1	S1	Go to check out page	1. go to e-shop 2. click on a product 3. select size/color/qty 4. click add to cart 5. go to cart window 6. click on check out button	User will be redirect to checkout page
#9	P1	S1	Try to checkout when sufficient credits are there	1. go to e-shop 2. click on a product 3. select size/color/qty 4. click add to cart 5. go to cart window 6. click on check out button	User can do the payment successfully
#10	P1	S1	Try to checkout when sufficient credits are not there	1. go to e-shop 2. click on a product 3. select size/color/qty 4. click add to cart 5. go to cart window 6. click on check out button	Validation message should be appears
#11	P2	S2	Verify payment completed page and email	1. go to e-shop 2. click on a product 3. select size/color/qty 4. click add to cart 5. go to cart window 6. click on check out button 7. click do the payment button	User can do the payment successfully and will receive an email regarding purchase details
#12	P1	S1	Validate cart is empty after checkout	1. go to e-shop 2. click on a product 3. select size/color/qty 4. click add to cart 5. go to cart window 6. click on check out button 7. click do the payment button	Cart is empty

Table 5.5 - Test cases for e-Shop

5.2.5 TEST CASES FOR VEHICLE MODULE

TC ID	Priority	Severity	Description	Test Steps	Expected Result
#1	P1	S1	Add a new vehicle to the system	1. go to Vehicle Management page 2. click add a vehicle 3. fill the form and submit	New vehicle should be added to the system
#2	P1	S2	Unable to add the same vehicle twice	1. go to Vehicle Management page 2. click add a vehicle 3. fill the form using previously added details and submit	1. Validation message should appears 2. Unable to add the same vehicle twice
#3	P1	S1	View added vehicle details	1. go to Vehicle Management page 2. click view vehicle button	Able to see vehicle details
#4	P2	S2	Update vehicle details	1. go to Vehicle Management page 2. click update vehicle button	Able to update and save vehicle details
#5	P1	S2	Deactivate vehicle	1. go to Vehicle Management page 2. click Deactivate vehicle button	Vehicle get deactivated
#6	P1	S1	Active vehicle	1. go to Vehicle Management page 2. click active vehicle button	Deactivated vehicle get active

Table 5.6 - Test cases for vehicle module

5.3 TEST METRICS

Test Matrix is a quantitative measure that is used to estimate the progress of the testing process.

After completing the testing following metrices used analyze the results.

1. Passed Test cases percentage
2. Defect Density

$$\text{Passed TC percentage customer module} = \frac{\text{No. of Passed Test Cases}}{\text{Total No. of Executed Test Cases}} \times 100\%$$

$$= \frac{27}{30} \times 100\ \%$$

= 90%

Defect density is the number of confirmed defects found in software/ component divided by the size of the software/ component. However, there is no fixed standard for defect density, studies suggest that one Defect per thousand lines of code is generally considered as a sign of good project quality (Swati, 2021).

$$\text{Defect Density of Customer Module } = \frac{\text{Defect count}}{\text{size of the release (LoC)}}$$

$$= \frac{3}{345}$$

= 0.008695 defects/LoC

= 0.000008695 defects/KLoC

UI issues and Validation issue were found in customer module and the defects were injected as coding error in Implementation phase.

5.4 USABILITY EVALUATION

The usability evaluation was done considering the citizens who use this system. To distribute the questionnaire a sample of citizens was selected using simple random sampling. The citizens were allowed to use the system for a period of one week. Then they were contacted through the phone and the purpose of the survey was explained. Afterwards questionnaire was sent as a google form. The results were collected to an Excel sheet and analyzed. The questionnaire was prepared adopting the System Usability Scale (Administration, 2021) according to the current system. The distributed questionnaire is attached in Appendix F.

5.4.1 EVALUATION RESULTS

The Figure 5.2 below indicates that the system has met user expectations as everyone agree that they prefer to use the system frequently. According to the above results most users are satisfied about the user friendliness of the system. As per the users’ feedback the help provided by the system, could be improved in the future. On average 83%

believe that the functions in the system are well integrated and 14% strongly believe that. It is further proved as majority don’t believe that there is too much inconsistency in the system. The results indicate that the system is error free and the system performance is satisfactory up to certain extent. Most of the users believe there is no security issue in this system.

The users expect to more live tracking of the garbage truck. It is obvious from the following comment.

“I wish I can see the truck closer to my home so I can get ready”

Some users prefer to have a mobile app as an improvement of the current system. It is clear by the statement.

“I don’t use the computer very much. If I can access this system through my phone it is easy for me”

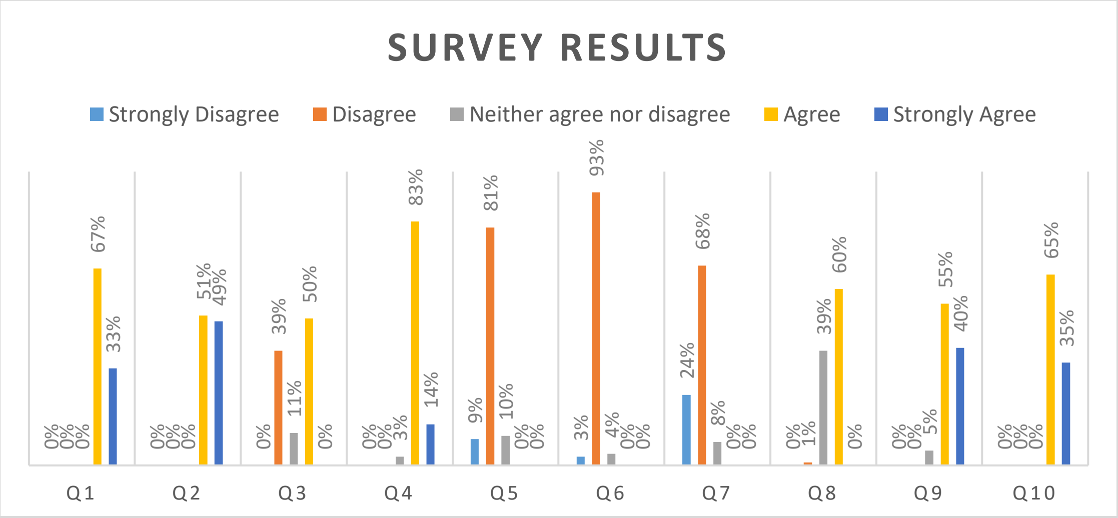


Figure 5.2 - System usability survey results analysis

CONCLUSION

The aim of the project was to implement a smart system for waste management to overcome the downsides in traditional door-step waste collection system. After a domain analysis and analyzing the solutions taken in other countries, a system was designed for the Sri Lankan context. It was implemented as a web based solution as the system should be accessible to citizens and truck drivers from anywhere. The system provides real time garbage bin status which is vital to lessen unhygienic environment and to reduce fuel, labor cost too. The system facilitates the staff for bin management, garbage management and schedule management. The implemented system provides a platform for communication with citizens and truck drivers so that citizens are able to make requests and complaints online, the truck drivers get job lists instantly and manage leave easily. This system can be used to improve the job profile of traditional garbage collectors, which will lead to improve living condition of them. This system acts as collaboration hub between customers, drivers, staff and garbage to streamline the waste management process so it will be beneficial to customers, drivers and environment too.

The system was implemented using latest web technologies such as PHP, MYSQL, AJAX and JQUERY. Bootstrap had been used as the front end framework and MVC had been used as the back end framework. Raspberry Pi module was used with sensors to the bin status real time. The system was tested using black box testing and database level testing approach in every increment. many stages using appropriate testing mechanisms for each stage to enhance the quality. For each of the module, passed test case percentage is 90%, which is an indication of minimization of bugs. The system was tested by a sample of users and by their feedback it is evident that the project succeeded.

This waste management system help the waste collection authorities by monitoring the domestic wastage in the area at regular intervals to avoid unhygienic and unhealthy environment. As it monitors the garbage bin status automatically it minimizes the frequent checking of garbage collection and hence reduces the overall expenditure associated with the garbage collection by minimizing the collection trips of garbage trucks. In the future, various types of sensors such as temperature sensors, humidity sensors can be included to get the precise output. This proposed product is an attempt to improve the current waste collection system in Sri Lanka and pave a path for an eco-friendly environment.

6.1 LESSONS LEARNT

This course component has given a chance to apply the theoretical knowledge gained in the degree program. The concepts leant in Database Management had been used in great extent throughout the project. The concepts in Quality Assurance was applied to ensure the product meets quality requirements. Mapping domain requirement to a computerized system was a complex task and gained a lot of experience in providing a computerized solution to an existing manual process in waste management.

The programming knowledge was vital to implement the system with new technologies and finally deliver a quality product in an acceptable manner. This component had helped me to apply my programming skills for a real world scenario and I learnt new tools and technologies such as Ajax, jQuery. It was interesting to study the MVC architecture and implement the system according to that design pattern. Python Programming was useful in implementing the IoT module.

This course component has helped to develop the soft skills such as technical writing, communication skills, time management and problem solving. Preparing the dissertation has increased the technical writing skills and by developing the system according to a schedule I learnt to manage activities efficiently and in a timely manner.

6.2 FUTURE IMPROVEMENTS

With a project addressing a real world scenario there is always room for improvement.

- Integrating a mobile app
  It will be convenient for the drivers if a mobile app is available. Then they can access the application anytime using their smart phones and a better experience can be delivered as screen size and touchscreen make web application much more difficult to use through mobile phones.
- Real time fleet tracking

The system evaluation summery indicates that the users prefer to find the current location of the garbage truck. Integrating a GPS tracking module to the truck will enable moment to moment route location of the garbage truck.

- Improve the Notification module

  Email notification is already integrated in the system and also integrating SMS notification could be beneficial to drivers and customers.

- Improve report module

  More reports can be generated by the system to derive meaningful information from the existing data. An accounting module could be integrated to get a comprehensive idea about the cash flow of the organization.

REFERENCES

Admin, E., 2017. *STATUS OF WASTE MANAGEMENT IN SRI LANKA.* [Online]
Available at: https://efl.lk/status-waste-management-sri-lanka/
[Accessed 3 January 2020].

Administration, U. G. S., 2021. *System Usability Scale (SUS).* [Online]
Available at: https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html
[Accessed 24 February 2021].

Alam, T. C. M. D. D. M., 2017. *Internet Of Things (IOT) Based Waste Management System.* Chittagong, Bangladesh, s.n.

Ana Isabel Molina, M. Á. R. O., 2009. A methodological approach for user interface development of collaborative applications: A case study. *Science Direct,* 74(9), pp. 754-776.

Anon., 2014. *System use case diagrams.* [Online]
Available at: https://www.togaf-modeling.org/models/application-architecture/system-use-case-diagrams.html
[Accessed 11 March 2021].

Anon., 2017. *Software Design Strategies.* [Online]
Available at:
https://www.tutorialspoint.com/software_engineering/software_design_strategies.htm
[Accessed 9 April 2021].

Anon., 2019. *newsfirst.* [Online]
Available at: https://www.newsfirst.lk/2019/08/08/collection-of-garbage-in-colombo-halted-solution-not-yet-provided/

Anon., 2021. *Chart.js.* [Online]
Available at: https://www.chartjs.org/
[Accessed 3 March 2021].

Anon., 2021. *Maps JavaScript API.* [Online]
Available at: https://developers.google.com/maps/documentation/javascript/overview
[Accessed 14 May 2021].

Anon., 2021. *PHPMailer.* [Online]
Available at: https://en.wikipedia.org/wiki/PHPMailer
[Accessed 30 January 2021].

Anon., 2021. *test cases in software testing - EverSundays.* [Online]
Available at: https://eversundays.com/aii540d/test-cases-in-software-testing
[Accessed 19 April 2021].

Anon., n.d. *Bootstrap (front-end framework) - Wikipedia.* [Online]
Available at: https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)
[Accessed 14 January 2020].

Anon., n.d. *DataTables.* [Online]
Available at: https://datatables.net/
[Accessed 3 March 2021].

Anon., n.d. *Normalization of Database.* [Online]
Available at: https://www.studytonight.com/dbms/database-normalization.php
[Accessed 21 February 2021].

Anon., n.d. *Systems architecture.* [Online]
Available at: https://en.wikipedia.org/wiki/Systems_architecture
[Accessed 14 January 2021].

Bhatia, A., 2016. *Waste Management: How India Is Drowning In Garbage.* [Online]
Available at: https://swachhindia.ndtv.com/waste-management-india-drowning-garbage-2147/
[Accessed 4 February 2021].

Dassanayake, M., 2011. *Successful Integrated Urban Planning Approach to Solid Waste Management in Sri Lanka,* s.l.: s.n.

Flow, T., 2020. *Trash Flow.* [Online]
Available at: https://www.trashflow.com/
[Accessed 27 December 2020].

Jewett, T., n.d. *Models and languages.* [Online]
Available at: http://www.tomjewett.com/dbdesign/dbdesign.php?page=models.html
[Accessed 19 April 2021].

Jordan, A., 2012. *Connected bins reduce Netherlands council's carbon footprint by a fifth.* [Online]

Available at: https://www.iot-now.com/2012/09/17/7543-connected-bins-reduce-netherlands-councils-carbon-footprint-by-a-fifth/
[Accessed 4 February 2021].

Logics, W., 2020. *Waste Logics.* [Online]
Available at: https://www.wastelogics.com/
[Accessed 27 December 2020].

Madhivanan, 2017. *Software Testing vs Software Evaluation [closed].* [Online]
Available at: https://stackoverflow.com/questions/16614368/software-testing-vs-software-evaluation/24553346
[Accessed 3 June 2021].

Matt, 2012. *Ultrasonic Distance Measurement Using Python – Part 1.* [Online]
Available at: https://www.raspberrypi-spy.co.uk/2012/12/ultrasonic-distance-measurement-using-python-part-1/
[Accessed 27 December 2020].

MSE, 2021. *Chapter 6 - Shaping A Greener Future With Science And Technology.* [Online]
Available at: https://www.towardszerowaste.gov.sg/zero-waste-masterplan/chapter6/
[Accessed 4 February 2021].

Ong, S., 2015. *Global Waste Management Outlook,* s.l.: United Nations Environmental Program.

Pardini, K., 2020. A Smart Waste Management Solution Geared towards Citizens. *Sensors 2020,* 22 April.

Paul Ammann, J. O., 2016. *Introduction to Software Testing.* 2nd ed. s.l.:Cambridge University Press.

S.A. Mahajan, A. K. A. S. M. S. ,. S., 2017. Smart Waste Management System using IoT. *International Journal of Advanced Engineering Research and Science (IJAERS,* pp. 93-95.

Sensoneo, 2021. *sensoneo.* [Online]
Available at: https://sensoneo.com/
[Accessed 18 May 2021].

Swati, S. t. m., 2021. *All In One Guide On Defect Density – Its Importance And How To Calculate It..* [Online]

Available at: https://www.softwaretestinghelp.com/defect-density/
[Accessed 27 February 2021].

WM Intellectual Property Holdings, L., 2020. *EXPLORE THE MY WM APP.* [Online]
Available at: https://www.wm.com/us/en/my-wm-mobile-app
[Accessed 21 February 2021].

Admin, E., 2017. *STATUS OF WASTE MANAGEMENT IN SRI LANKA.* [Online]
Available at: https://efl.lk/status-waste-management-sri-lanka/
[Accessed 3 January 2020].

Administration, U. G. S., 2021. *System Usability Scale (SUS).* [Online]
Available at: https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html
[Accessed 24 February 2021].

Alam, T. C. M. D. D. M., 2017. *Internet Of Things (IOT) Based Waste Management System.* Chittagong, Bangladesh, s.n.

Ana Isabel Molina, M. Á. R. O., 2009. A methodological approach for user interface development of collaborative applications: A case study. *Science Direct,* 74(9), pp. 754-776.

Anon., 2014. *System use case diagrams.* [Online]
Available at: https://www.togaf-modeling.org/models/application-architecture/system-use-case-diagrams.html
[Accessed 11 March 2021].

Anon., 2017. *Software Design Strategies.* [Online]
Available at:
https://www.tutorialspoint.com/software_engineering/software_design_strategies.htm
[Accessed 9 April 2021].

Anon., 2019. *newsfirst.* [Online]
Available at: https://www.newsfirst.lk/2019/08/08/collection-of-garbage-in-colombo-halted-solution-not-yet-provided/

Anon., 2021. *Chart.js.* [Online]
Available at: https://www.chartjs.org/
[Accessed 3 March 2021].

Anon., 2021. *Maps JavaScript API.* [Online]
Available at: https://developers.google.com/maps/documentation/javascript/overview
[Accessed 14 May 2021].

Anon., 2021. *PHPMailer.* [Online]
Available at: https://en.wikipedia.org/wiki/PHPMailer
[Accessed 30 January 2021].

Anon., 2021. *test cases in software testing - EverSundays.* [Online]
Available at: https://eversundays.com/aii540d/test-cases-in-software-testing
[Accessed 19 April 2021].

Anon., n.d. *Bootstrap (front-end framework) - Wikipedia.* [Online]
Available at: https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)
[Accessed 14 January 2020].

Anon., n.d. *DataTables.* [Online]
Available at: https://datatables.net/
[Accessed 3 March 2021].

Anon., n.d. *Normalization of Database.* [Online]
Available at: https://www.studytonight.com/dbms/database-normalization.php
[Accessed 21 February 2021].

Anon., n.d. *Systems architecture.* [Online]
Available at: https://en.wikipedia.org/wiki/Systems_architecture
[Accessed 14 January 2021].

Bhatia, A., 2016. *Waste Management: How India Is Drowning In Garbage.* [Online]
Available at: https://swachhindia.ndtv.com/waste-management-india-drowning-garbage-2147/
[Accessed 4 February 2021].

Dassanayake, M., 2011. *Successful Integrated Urban Planning Approach to Solid Waste Management in Sri Lanka,* s.l.: s.n.

Flow, T., 2020. *Trash Flow.* [Online]
Available at: https://www.trashflow.com/
[Accessed 27 December 2020].

Jewett, T., n.d. *Models and languages.* [Online]
Available at: http://www.tomjewett.com/dbdesign/dbdesign.php?page=models.html
[Accessed 19 April 2021].

Jordan, A., 2012. *Connected bins reduce Netherlands council's carbon footprint by a fifth.* [Online]
Available at: https://www.iot-now.com/2012/09/17/7543-connected-bins-reduce-netherlands-councils-carbon-footprint-by-a-fifth/
[Accessed 4 February 2021].

Logics, W., 2020. *Waste Logics.* [Online]
Available at: https://www.wastelogics.com/
[Accessed 27 December 2020].

Madhivanan, 2017. *Software Testing vs Software Evaluation [closed].* [Online]
Available at: https://stackoverflow.com/questions/16614368/software-testing-vs-software-evaluation/24553346
[Accessed 3 June 2021].

Matt, 2012. *Ultrasonic Distance Measurement Using Python – Part 1.* [Online]
Available at: https://www.raspberrypi-spy.co.uk/2012/12/ultrasonic-distance-measurement-using-python-part-1/
[Accessed 27 December 2020].

MSE, 2021. *Chapter 6 - Shaping A Greener Future With Science And Technology.* [Online]
Available at: https://www.towardszerowaste.gov.sg/zero-waste-masterplan/chapter6/
[Accessed 4 February 2021].

Ong, S., 2015. *Global Waste Management Outlook,* s.l.: United Nations Environmental Program.

Pardini, K., 2020. A Smart Waste Management Solution Geared towards Citizens. *Sensors 2020,* 22 April.

Paul Ammann, J. O., 2016. *Introduction to Software Testing.* 2nd ed. s.l.:Cambridge University Press.

S.A. Mahajan, A. K. A. S. M. S. ,. S., 2017. Smart Waste Management System using IoT. *International Journal of Advanced Engineering Research and Science (IJAERS,* pp. 93-95.

Sensoneo, 2021. *sensoneo.* [Online]
Available at: https://sensoneo.com/
[Accessed 18 May 2021].

Swati, S. t. m., 2021. *All In One Guide On Defect Density – Its Importance And How To Calculate It.*. [Online]
Available at: https://www.softwaretestinghelp.com/defect-density/
[Accessed 27 February 2021].

WM Intellectual Property Holdings, L., 2020. *EXPLORE THE MY WM APP.* [Online]
Available at: https://www.wm.com/us/en/my-wm-mobile-app
[Accessed 21 February 2021].

APPENDIX A - SYSTEM DOCUMENTATION

This document gives step by step instructions to install Waste Management System.

7.1 SYSTEM REQUIREMENTS

7.1.1 HARDWARE REQUIREMENTS

Hardware	**Minimum Requirement**
Processor	3.6 GHz Intel i3 Processor
Memory	4 GB RAM
Hard Disk	40GB Capacity
Internet	Standard Internet Connection

Table 7.1 - Hardware requirement

7.1.2 SOFTWARE REQUIREMENTS

Software	**Minimum Requirement**
Operating System	Windows 7, Windows 8.1, Windows 10
XAMPP	XAMPP V5.6.31 or Separate Installation of Apache 2.4.26 /MySQL 5.5.32 / PHP 5.6.31 / phpMyAdmin 4.7.0 / Mercury Mail Transport System v4.63
Web Browser	Google Chrome Version 92.0.4515.107 Firefox 89.0.2
PDF Reader	Adobe Acrobat Reader

Table 7.2 - Software requirement

7.2 INSTALLATION

- Download and install XAMPP for Windows from https://www.apachefriends.org/index.html. Give installation path to C:\xampp of the computer.

- Install a given web browser above.

- Open the CD and copy the wastemgt folder and paste into “C:\xampp\htdocs”

7.2.1 DATABASE INSTALLATION

- Open the web browser and type the URL http://localhost/phpmyadmin/

- Create a new database named “wastemgt” and in “Import” tab browse CD for waste management database.

- After selecting wastemgt in CD click Go.

7.2.2 SYSTEM LAUNCHING

- Open XAMPP control panel and start Apache & MySql services.

- Open web brower and type URL http://localhost/wastemgt and press enter button.

APPENDIX B - DESIGN DOCUMENTATION

8.1 USER MANAGEMENT

The diagram in Figure 8.1 is a detailed use case diagram of the user management process.

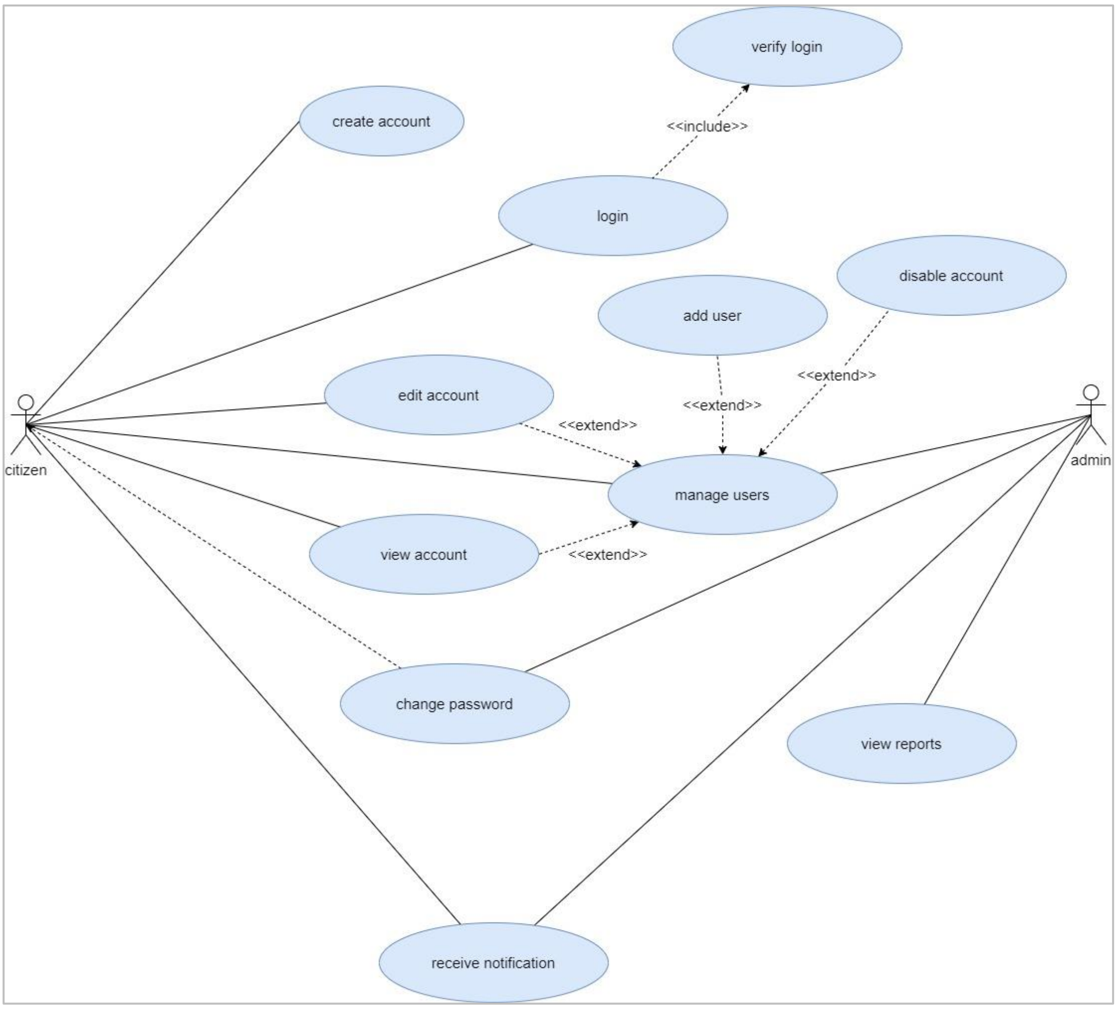


Figure 8.1 - Use case diagram; User management process

8.2 E-COMMERCE SITE

Using the e-Commerce site customer able to buy eco-friendly products by spending green points. Figure 8.2 shows the use case diagram of e-Commerce site process.

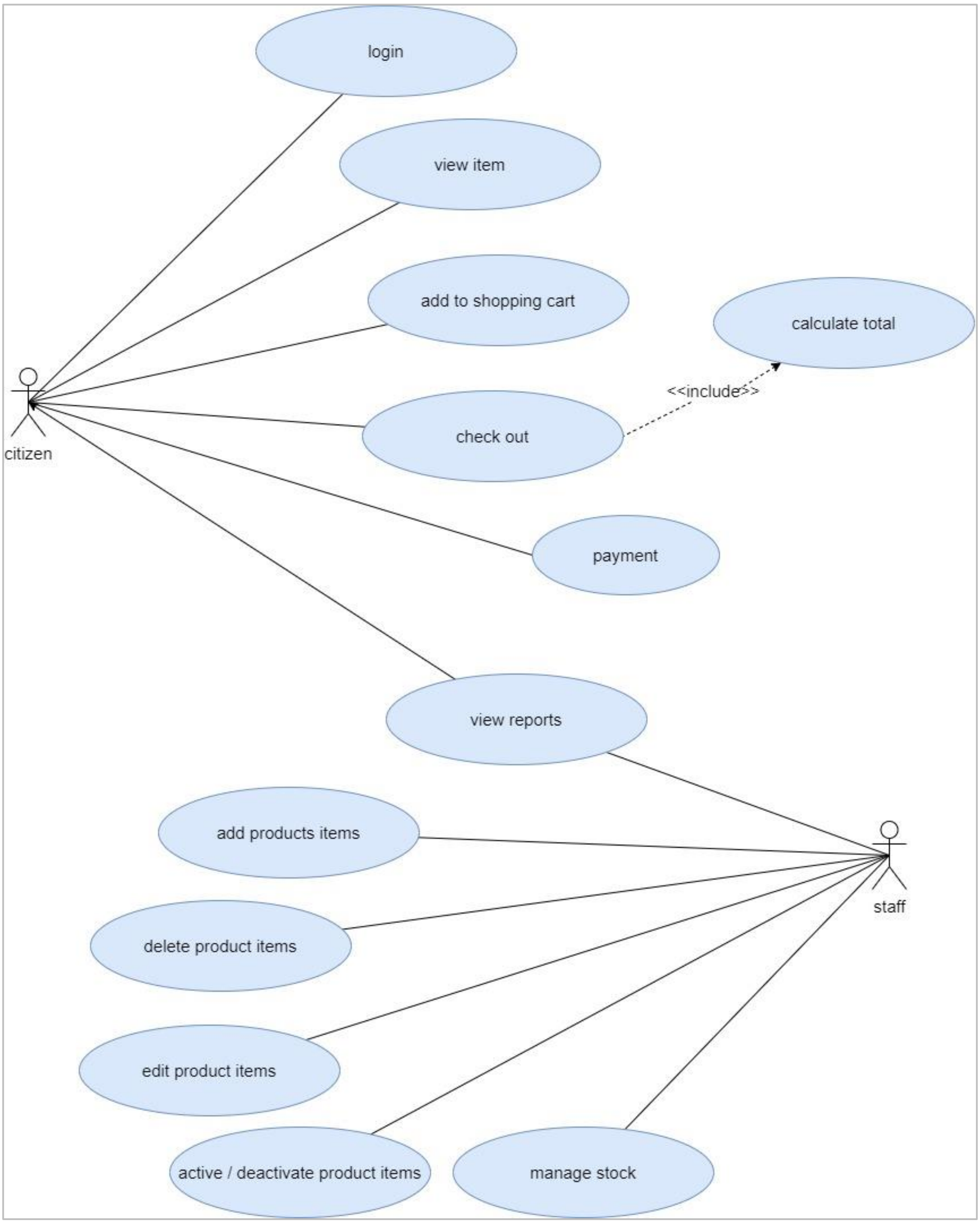


Figure 8.2 - Use case diagram; e-Commerce site

8.3 NOTIFICATION MODULE

Notification module automatically send notification to users, staff and drivers when there are specific event occurred. Activities in notification module is depicted in Figure 8.3.

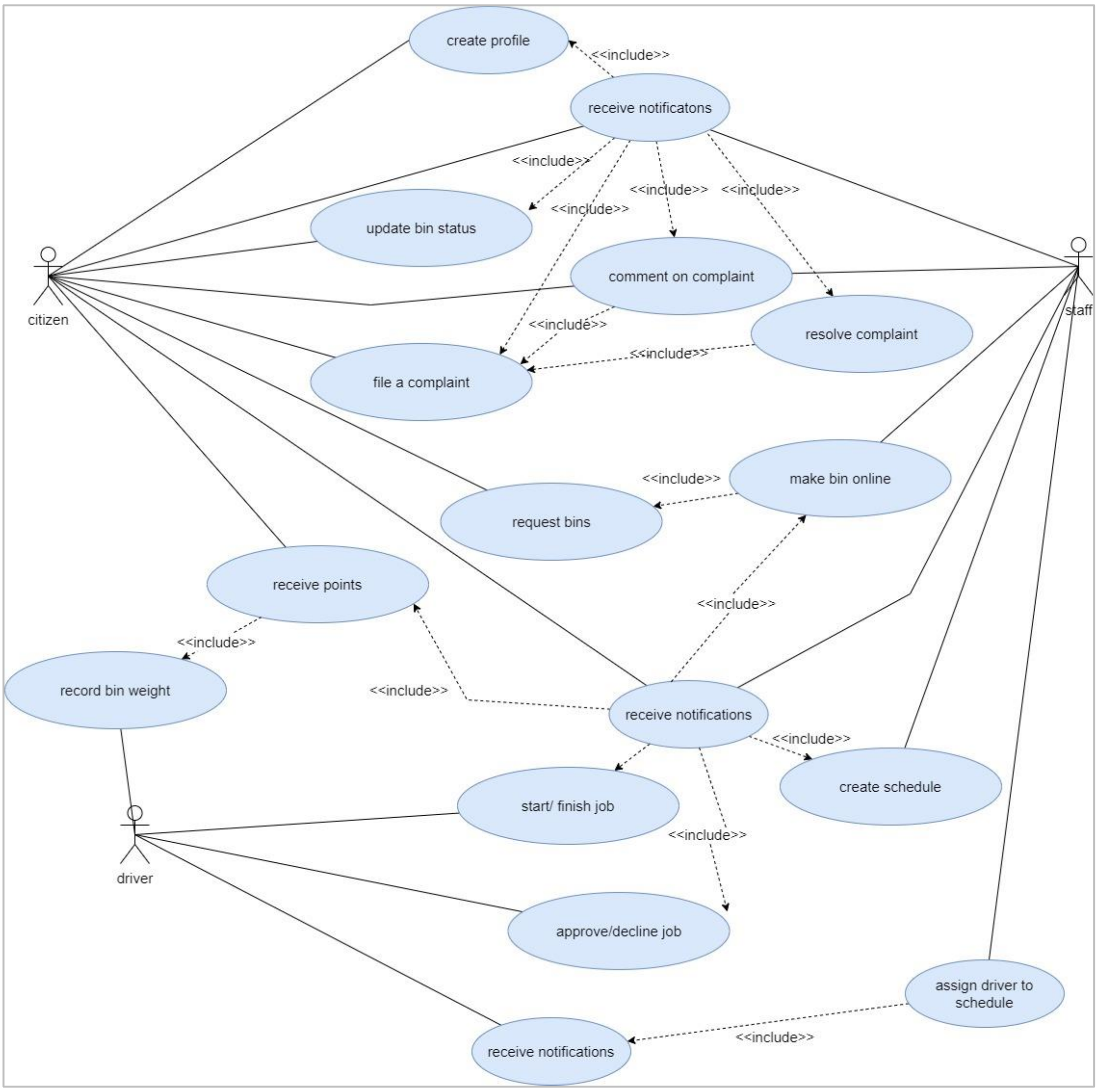


Figure 8.3 - Use case diagram: Notification module

8.4 SCHEDULE MANAGEMENT

Scheduling plays important role in the system as it connect to users, garbage bins, drivers. Figure 8.4 shows how schedule management handles in the system.

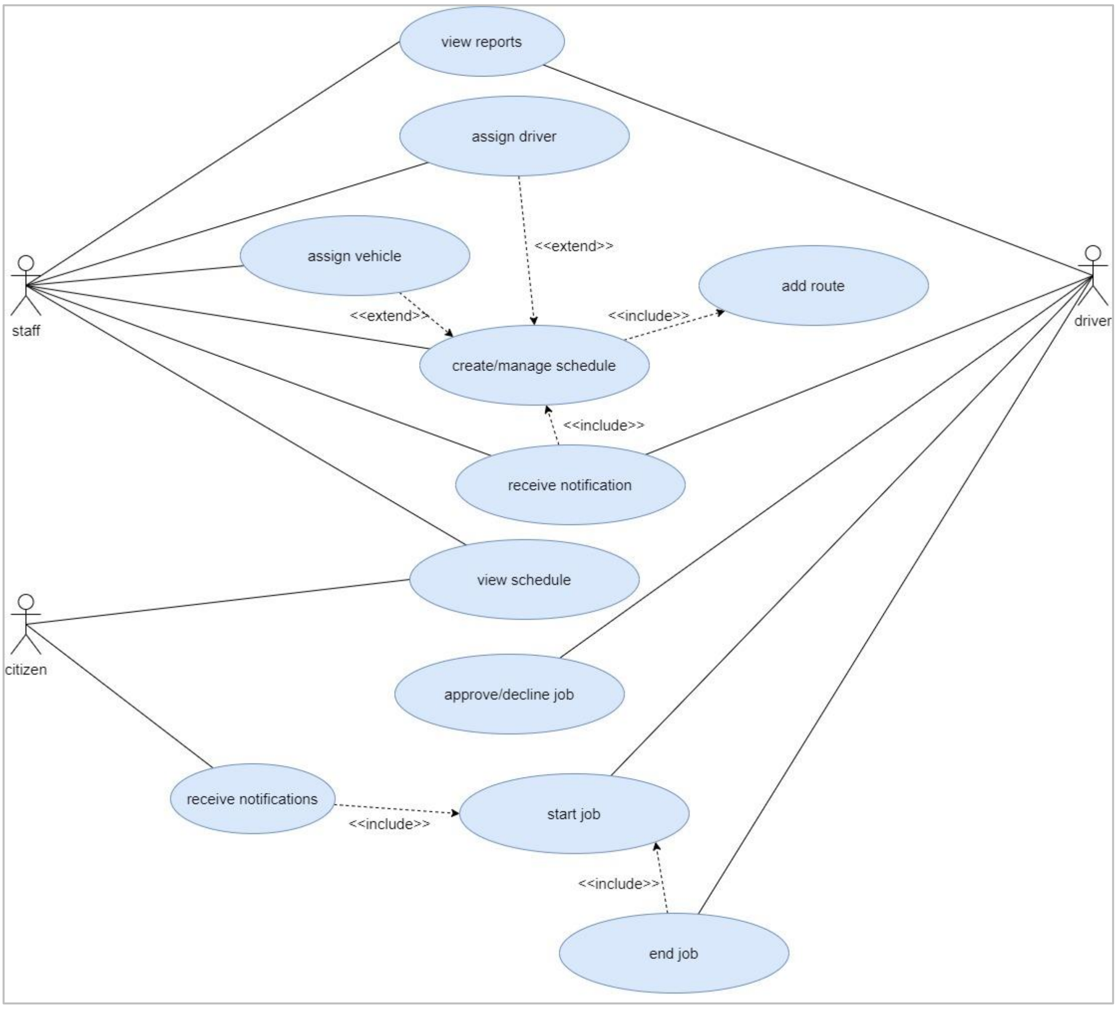


Figure 8.4 - Use case diagram: Schedule management

APPENDIX C - USER DOCUMENTATION

Different user levels has different system privileges so profiles are different in each and every user level. This user documentation assumes administrator role which has all the system privileges.

9.1 LOGIN FORM

Following Figure 9.1 shows the login form which used to login into the system. User can use email address and password to login into system. Validations in the user form check for correct user name and password. User can change the password using forgot password link.

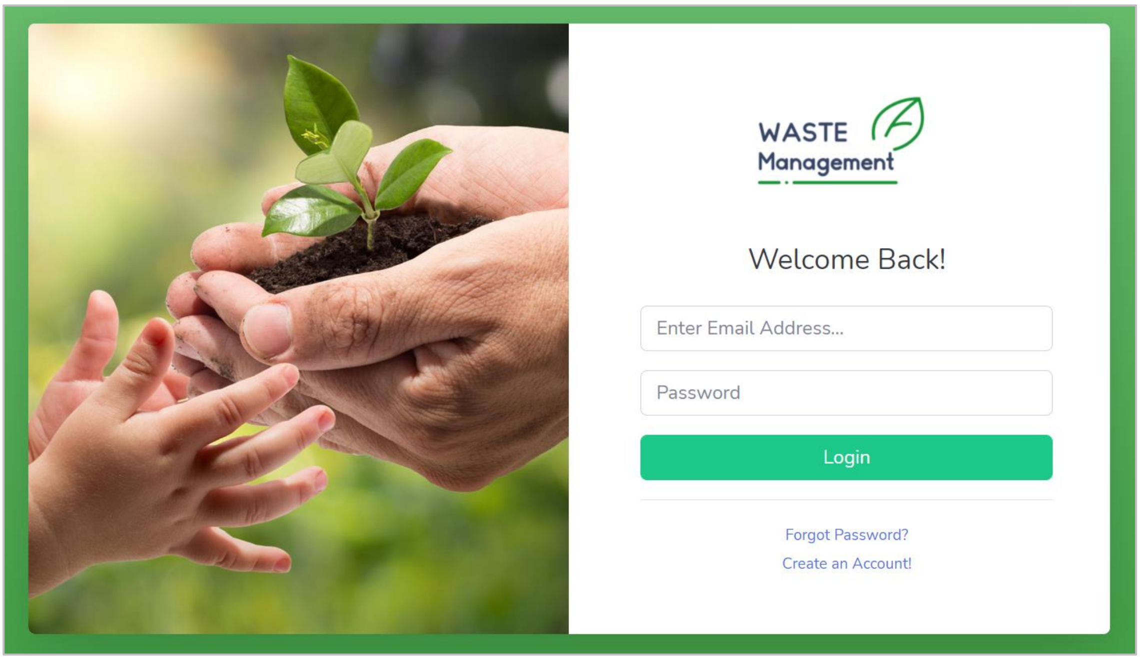
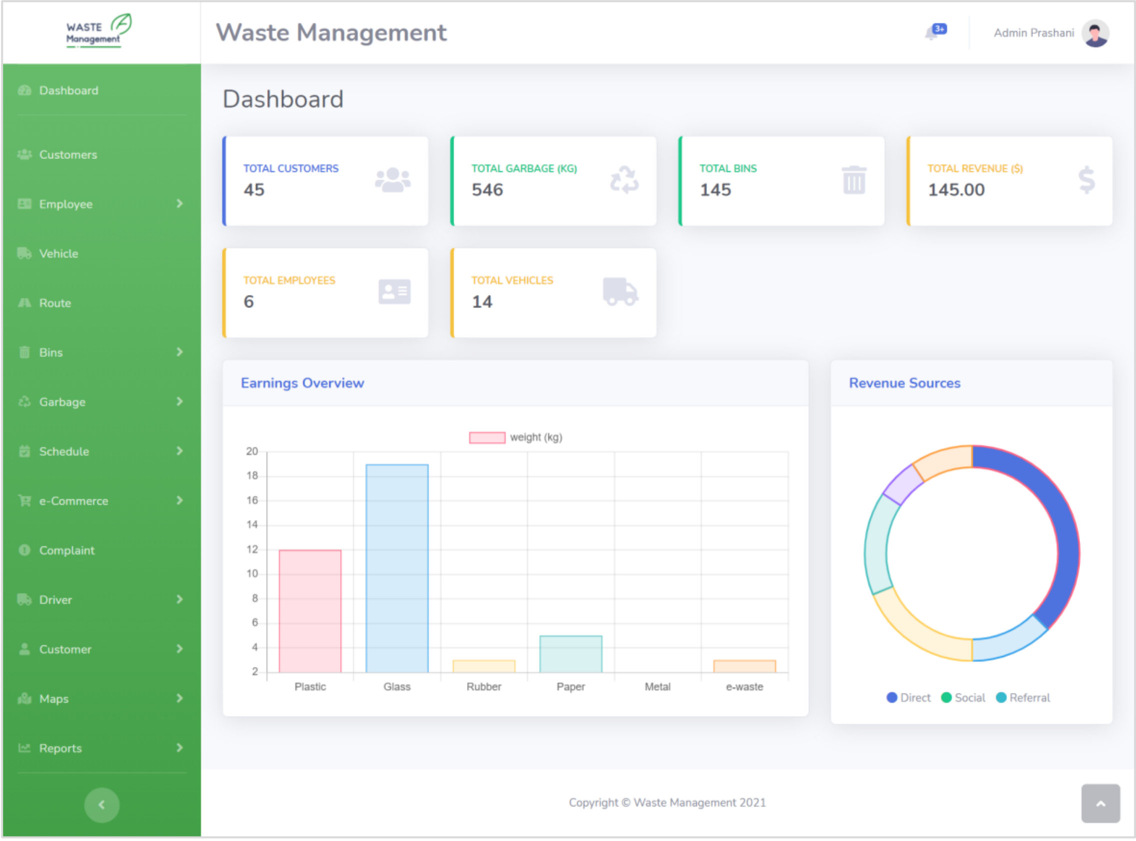


Figure 9.1 - Login form

9.2 DASHBOARD

Dashboard page consists of top bar, left navigation bar, body and footer. All the modules in the system can access through the system dashboard. The modules that appear on the dashboard may differ according to user privileges. Left side navigation panel grouped all available modules accordingly for easy access. When clicked on profile picture/user name on top left side of the window, sign-out button will be available.



Dashboard

TOTAL CUSTOMERS 45

TOTAL GARBAGE (KG) 546

TOTAL BINS 145

TOTAL REVENUE ($) 145.00

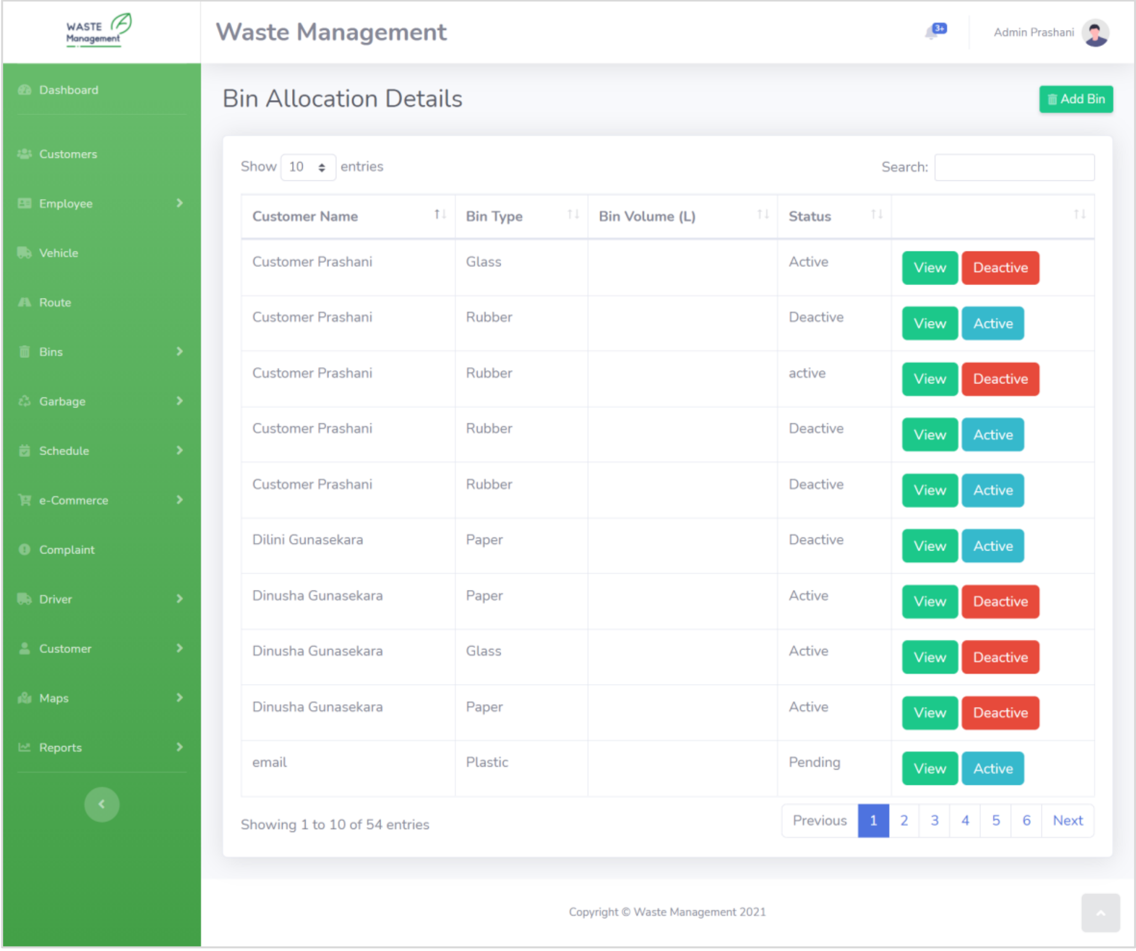
TOTAL EMPLOYEES 6

TOTAL VEHICLES 14

Figure 9.2 - Dashboard

9.3 BIN ALLOCATION

Bin Allocation page is available for users who log in with staff member credentials. The user can get details of the bins such as the bin type, bin volume and the user is able to allocate the individual bins to the customers. Figure 9.3 illustrates the bin allocation page.



Bin Allocation Details

Show 10 entries Search:

Customer Name	Bin Type	Bin Volume (L)	Status	
Customer Prashani	Glass		Active	View Deactive
Customer Prashani	Rubber		Deactive	View Active
Customer Prashani	Rubber		active	View Deactive
Customer Prashani	Rubber		Deactive	View Active
Customer Prashani	Rubber		Deactive	View Active
Dilini Gunasekara	Paper		Deactive	View Active
Dinusha Gunasekara	Paper		Active	View Deactive
Dinusha Gunasekara	Glass		Active	View Deactive
Dinusha Gunasekara	Paper		Active	View Deactive
email	Plastic		Pending	View Active

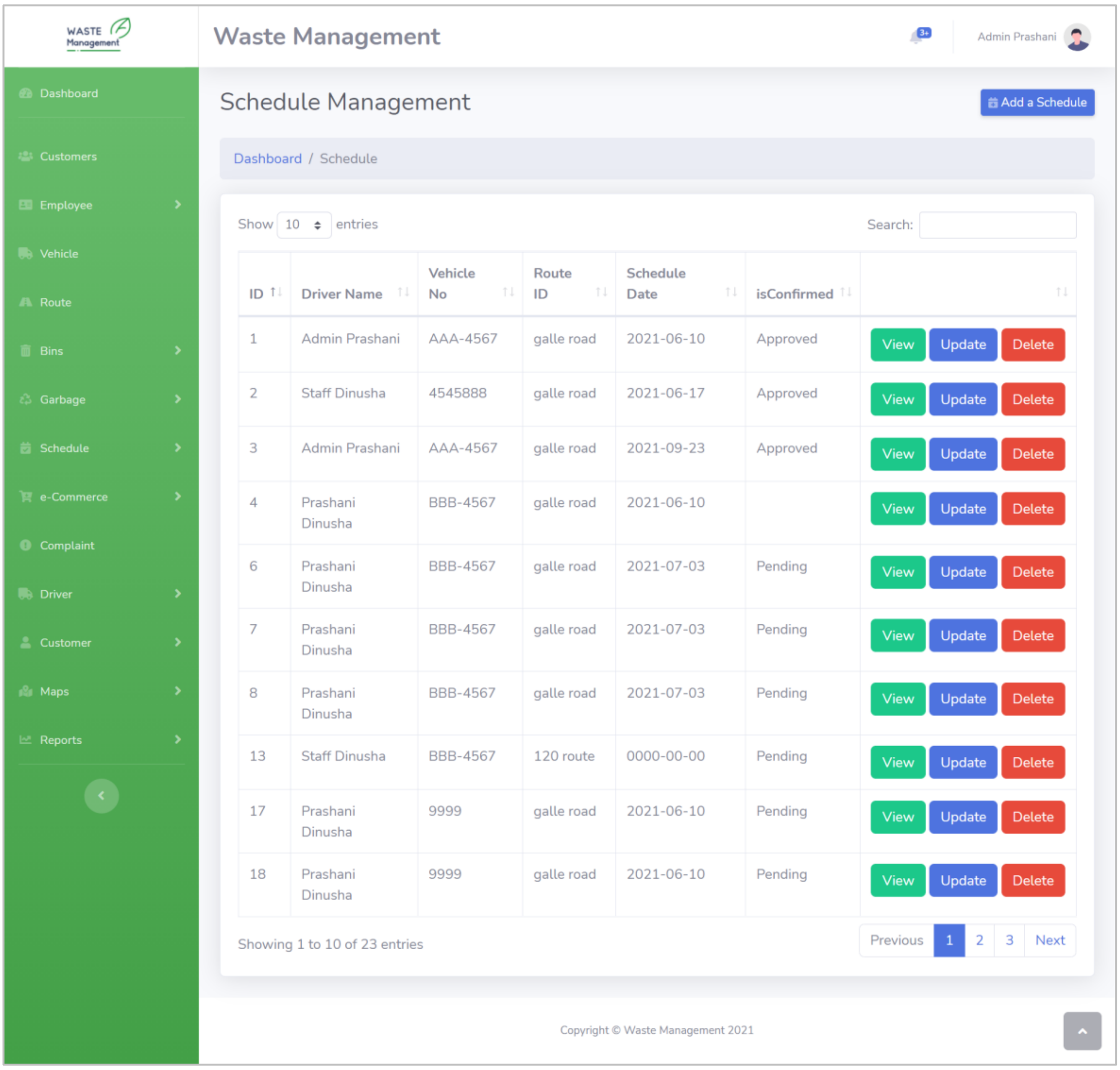
Showing 1 to 10 of 54 entries

Copyright © Waste Management 2021

Figure 9.3 - Bin allocation detail page

9.4 MANAGE SCHEDULE

This page allows the staff member to get a comprehensive view of all the details required to create a schedule namely the routes, Trash collection requests on that route and the driver availability. The staff member can create a schedule through the interface shown in Figure 9.4.



Schedule Management

Show 10 entries

Search:

ID	Driver Name	Vehicle No	Route ID	Schedule Date	isConfirmed	
1	Admin Prashani	AAA-4567	galle road	2021-06-10	Approved	View Update Delete
2	Staff Dinusha	4545888	galle road	2021-06-17	Approved	View Update Delete
3	Admin Prashani	AAA-4567	galle road	2021-09-23	Approved	View Update Delete
4	Prashani Dinusha	BBB-4567	galle road	2021-06-10		View Update Delete
6	Prashani Dinusha	BBB-4567	galle road	2021-07-03	Pending	View Update Delete
7	Prashani Dinusha	BBB-4567	galle road	2021-07-03	Pending	View Update Delete
8	Prashani Dinusha	BBB-4567	galle road	2021-07-03	Pending	View Update Delete
13	Staff Dinusha	BBB-4567	120 route	0000-00-00	Pending	View Update Delete
17	Prashani Dinusha	9999	galle road	2021-06-10	Pending	View Update Delete
18	Prashani Dinusha	9999	galle road	2021-06-10	Pending	View Update Delete

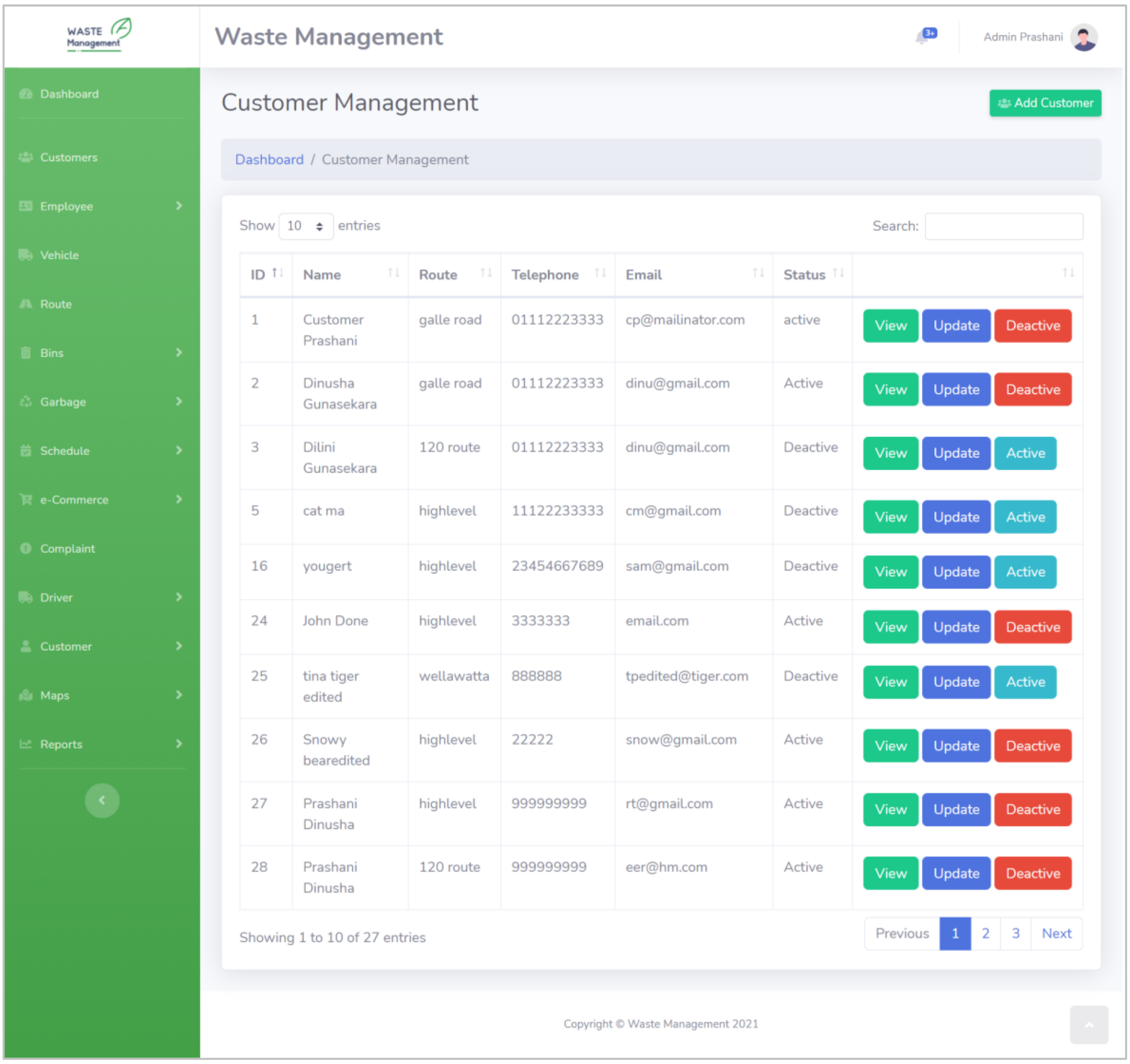
Showing 1 to 10 of 23 entries

Previous 1 2 3 Next

Figure 9.4 - Manage schedule page

9.5 CUSTOMER PAGE

All the customer related details shows under customer management page. Most important fields are shown on the table view and the further details can be viewed using the “View” button. Using “Add new customer” button in top right corner button can create a new customers. There are options to update customer details and delete customers as well.



Customer Management

Show 10 entries

ID	Name	Route	Telephone	Email	Status	
1	Customer Prashani	galle road	01112223333	cp@mailinator.com	active	View Update Deactive
2	Dinusha Gunasekara	galle road	01112223333	dinu@gmail.com	Active	View Update Deactive
3	Dilini Gunasekara	120 route	01112223333	dinu@gmail.com	Deactive	View Update Active
5	cat ma	highlevel	11122233333	cm@gmail.com	Deactive	View Update Active
16	yougert	highlevel	23454667689	sam@gmail.com	Deactive	View Update Active
24	John Done	highlevel	3333333	email.com	Active	View Update Deactive
25	tina tiger edited	wellawatta	888888	tpedited@tiger.com	Deactive	View Update Active
26	Snowy bearedited	highlevel	22222	snow@gmail.com	Active	View Update Deactive
27	Prashani Dinusha	highlevel	999999999	rt@gmail.com	Active	View Update Deactive
28	Prashani Dinusha	120 route	999999999	eer@hm.com	Active	View Update Deactive

Showing 1 to 10 of 27 entries

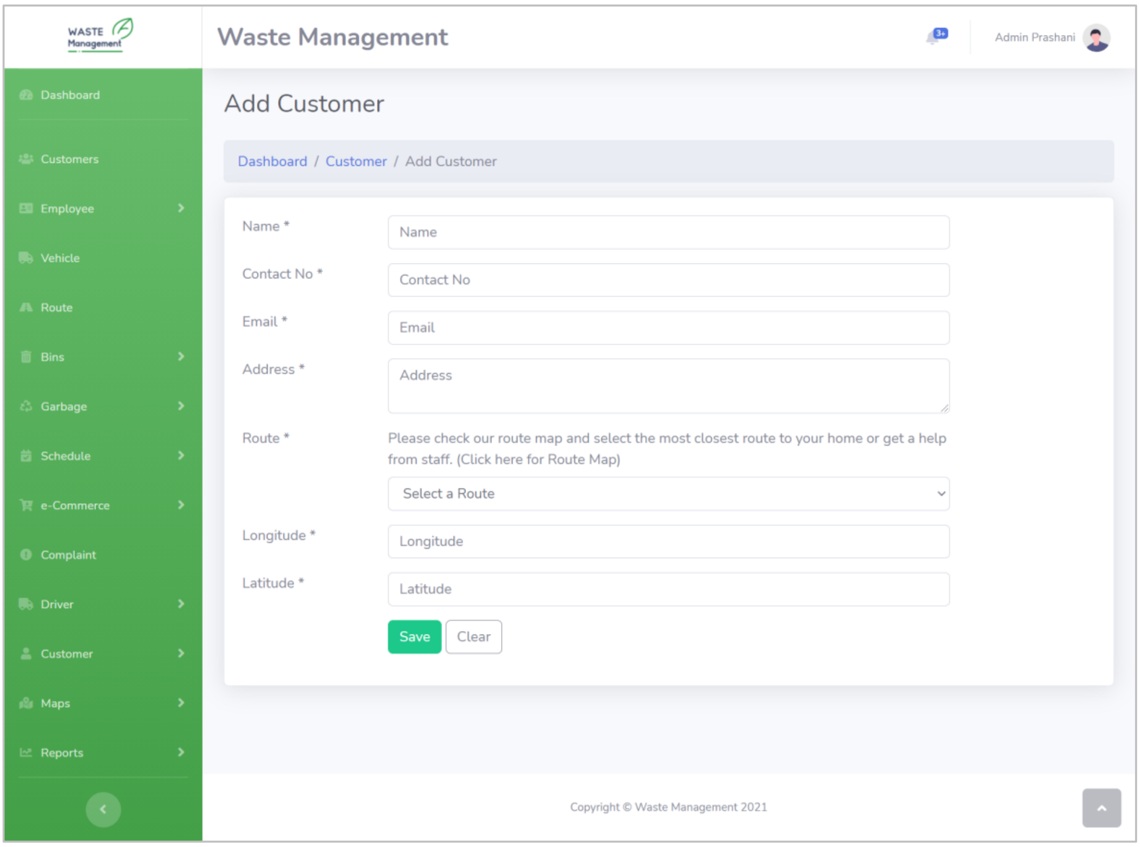
Previous 1 2 3 Next

Copyright © Waste Management 2021

Figure 9.5 - Customer management page

9.5.1 ADD CUSTOMER FORM

New customer details can entered the system via this form. There are validations for invalid data and if invalid data entered an error message is shown and the user will be unable to submit the form. Duplicate email addresses cannot be entered to the system as the system verifies the new email address with the existing ones in the database.



Add Customer

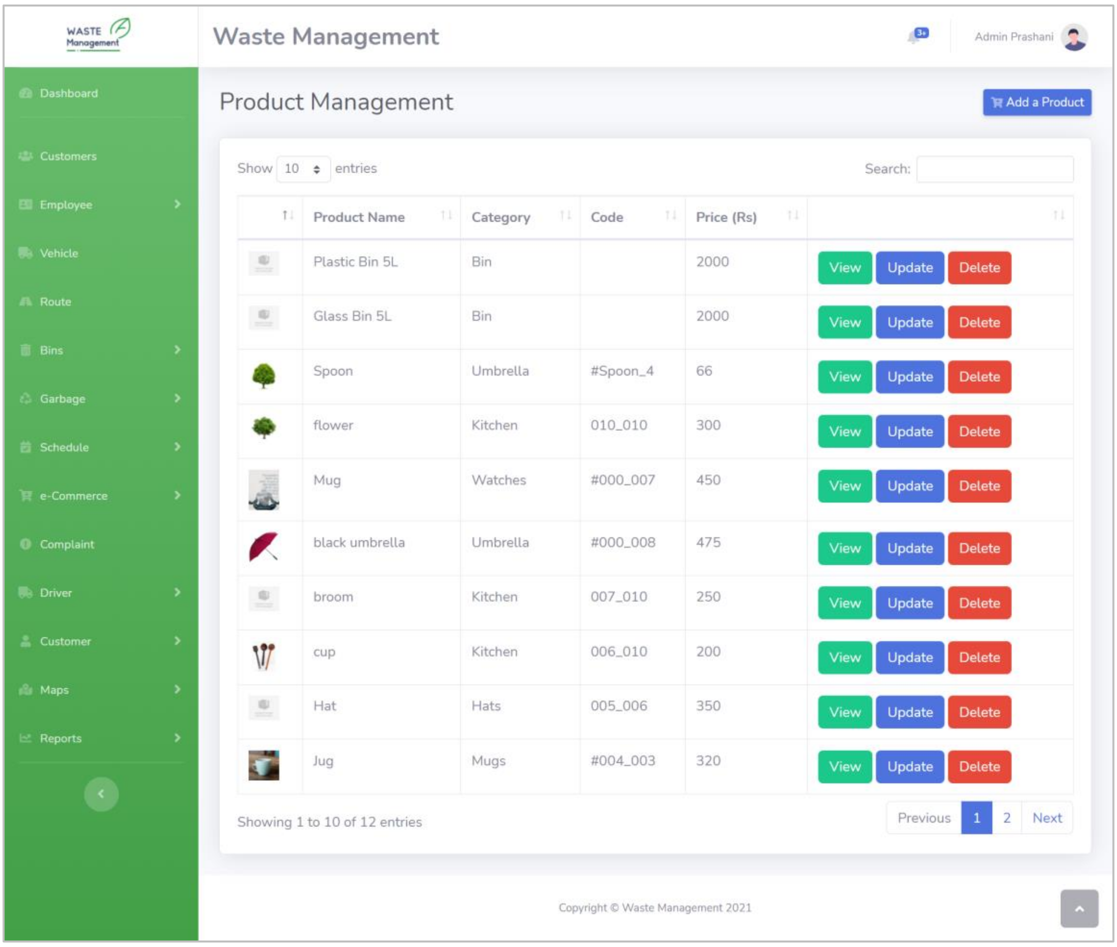
Dashboard / Customer / Add Customer

Name *	Name
Contact No *	Contact No
Email *	Email
Address *	Address
Route *	Please check our route map and select the most closest route to your home or get a help from staff. (Click here for Route Map) Select a Route
Longitude *	Longitude
Latitude *	Latitude

Figure 9.6 - Add customer form

9.6 PRODUCT PAGE

The products shown in eshop page are added through product management page. Product category, product type, size, price shows in Product page and user can view, update and delete the products.



Product Management

Show 10 entries

Search:

	Product Name	Category	Code	Price (Rs)	
	Plastic Bin 5L	Bin		2000	View Update Delete
	Glass Bin 5L	Bin		2000	View Update Delete
	Spoon	Umbrella	#Spoon_4	66	View Update Delete
	flower	Kitchen	010_010	300	View Update Delete
	Mug	Watches	#000_007	450	View Update Delete
	black umbrella	Umbrella	#000_008	475	View Update Delete
	broom	Kitchen	007_010	250	View Update Delete
	cup	Kitchen	006_010	200	View Update Delete
	Hat	Hats	005_006	350	View Update Delete
	Jug	Mugs	#004_003	320	View Update Delete

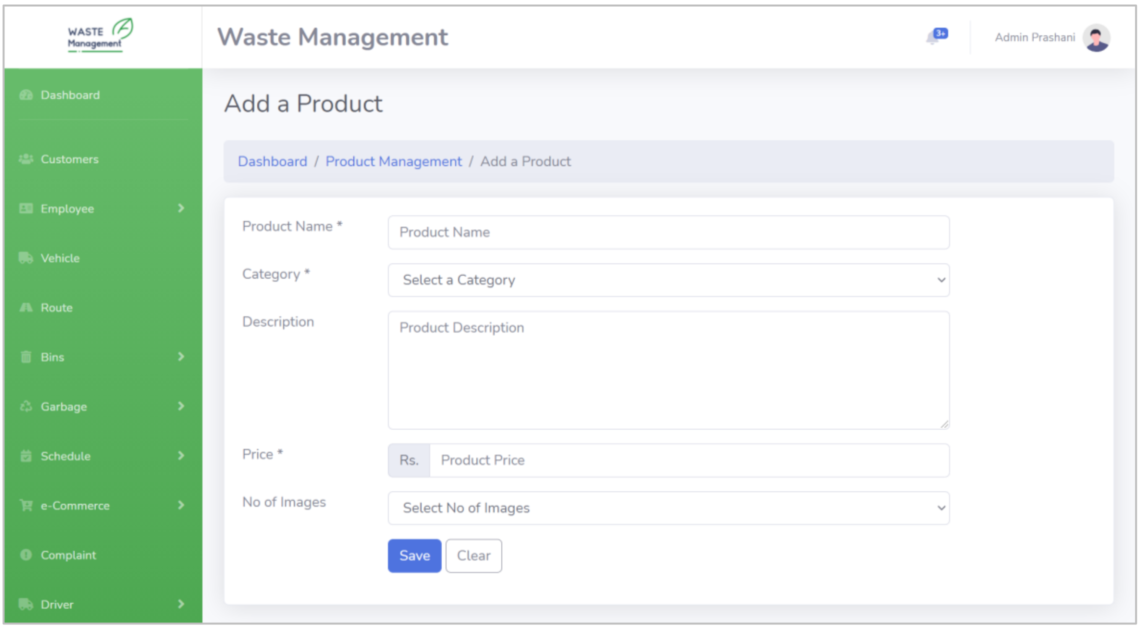
Showing 1 to 10 of 12 entries

Previous 1 2 Next

Figure 9.7 - Product management page

9.6.1 ADD A PRODUCT FORM

In Figure 9.7 shows add a product form. User can select product type and can search category and size or add new category and size. There are validations to check empty or correct data types and if user add invalid data validation message shows and can’t proceed further until correct the error.



Add a Product

Dashboard / Product Management / Add a Product

Product Name * Product Name

Category * Select a Category

Description Product Description

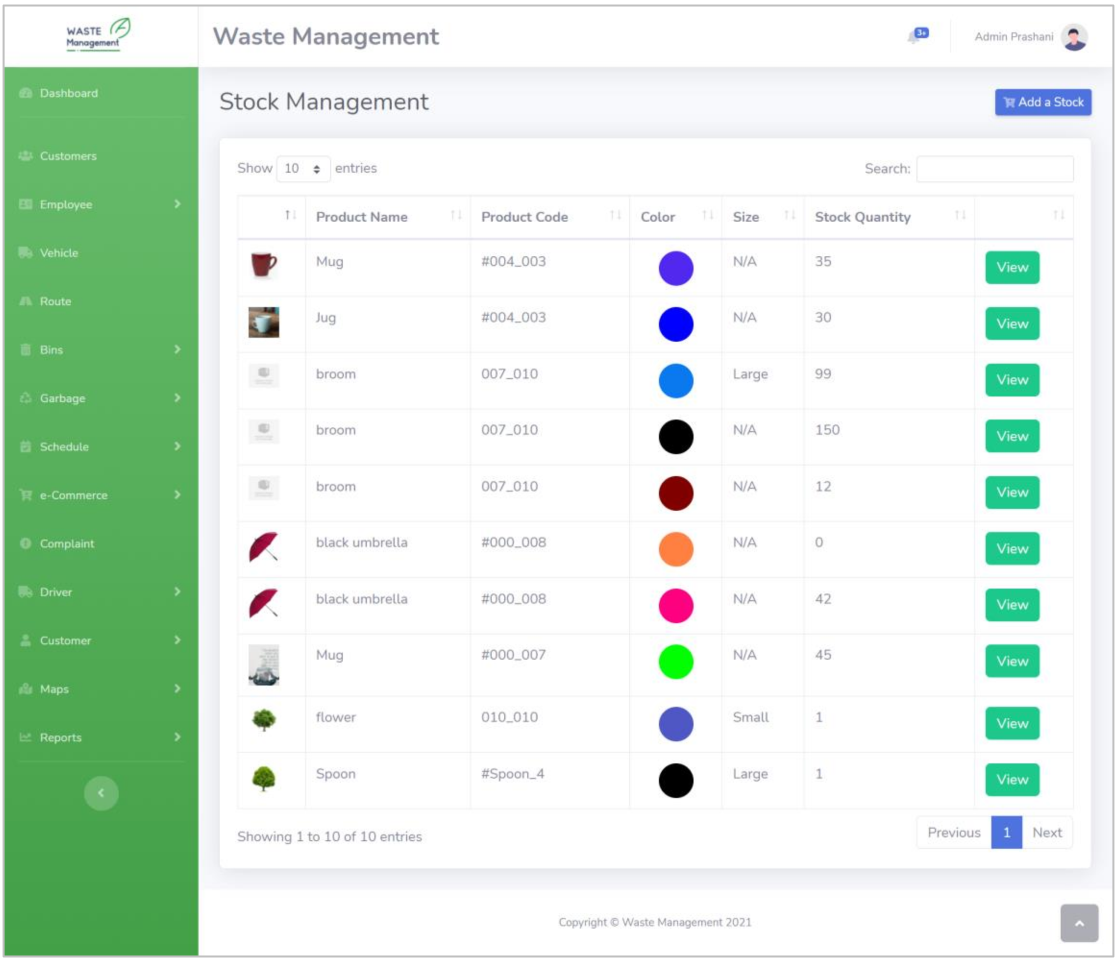
Price * Rs. Product Price

No of Images Select No of Images

Figure 9.8 - Add a product form

9.7 STOCK PAGE

Stock details about the products are shown in stock page. Administrator can add a stock using the button in the upper left corner. The stock page shows product name, product code, color, size and stock quantity. Only “view stock” option is available for users other than the administrator. Only the administrator has the privilege to add, update or delete stock.



Stock Management

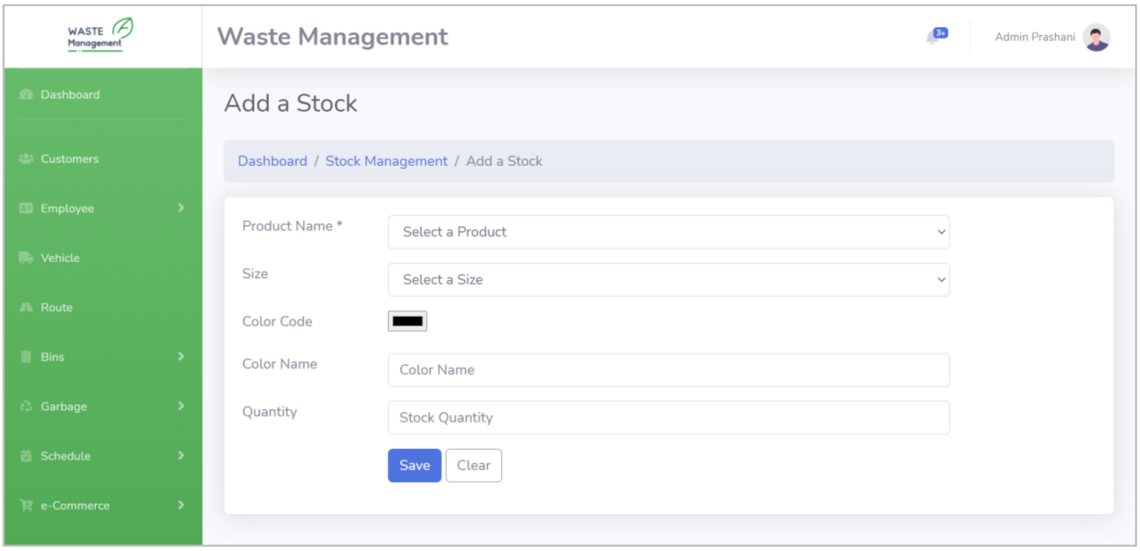
	Product Name	Product Code	Color	Size	Stock Quantity	
	Mug	#004_003	●	N/A	35	View
	Jug	#004_003	●	N/A	30	View
	broom	007_010	●	Large	99	View
	broom	007_010	●	N/A	150	View
	broom	007_010	●	N/A	12	View
	black umbrella	#000_008	●	N/A	0	View
	black umbrella	#000_008	●	N/A	42	View
	Mug	#000_007	●	N/A	45	View
	flower	010_010	●	Small	1	View
	Spoon	#Spoon_4	●	Large	1	View

Showing 1 to 10 of 10 entries

Copyright © Waste Management 2021

Figure 9.9 - Stock management page

9.7.1 ADD A STOCK FORM



Add a Stock

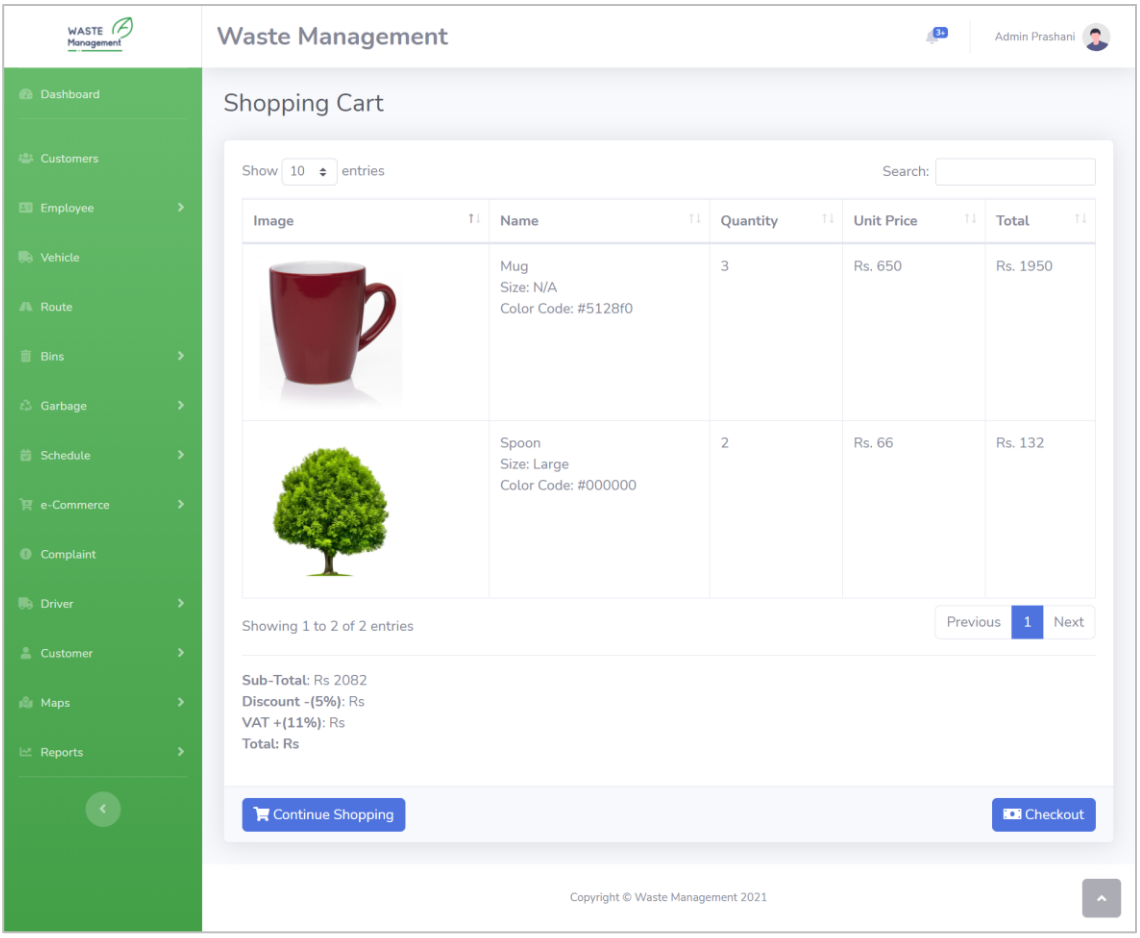
Product Name *	Select a Product
Size	Select a Size
Color Code	■
Color Name	Color Name
Quantity	Stock Quantity

Figure 9.10 - Add a stock form

Stock form page can be used to add stock to the system. Products can be selected using drop down menu and if stock date not added current date is taken as the stock date. A size already in the database is selected or a new size can be added. User can either select a color or give a color name. The color name will be hidden if the user selects a color code. There is validations to check empty or incorrect data types and if user add invalid data validation message shows and can’t proceed further until correct the error.

9.8 E-SHOP PAGE

e-Shop is the place where customers can spend their green points and purchase eco-friendly products. Below screenshots shows different screens related to e-Shop.



Shopping Cart

Show 10 entries

Search:

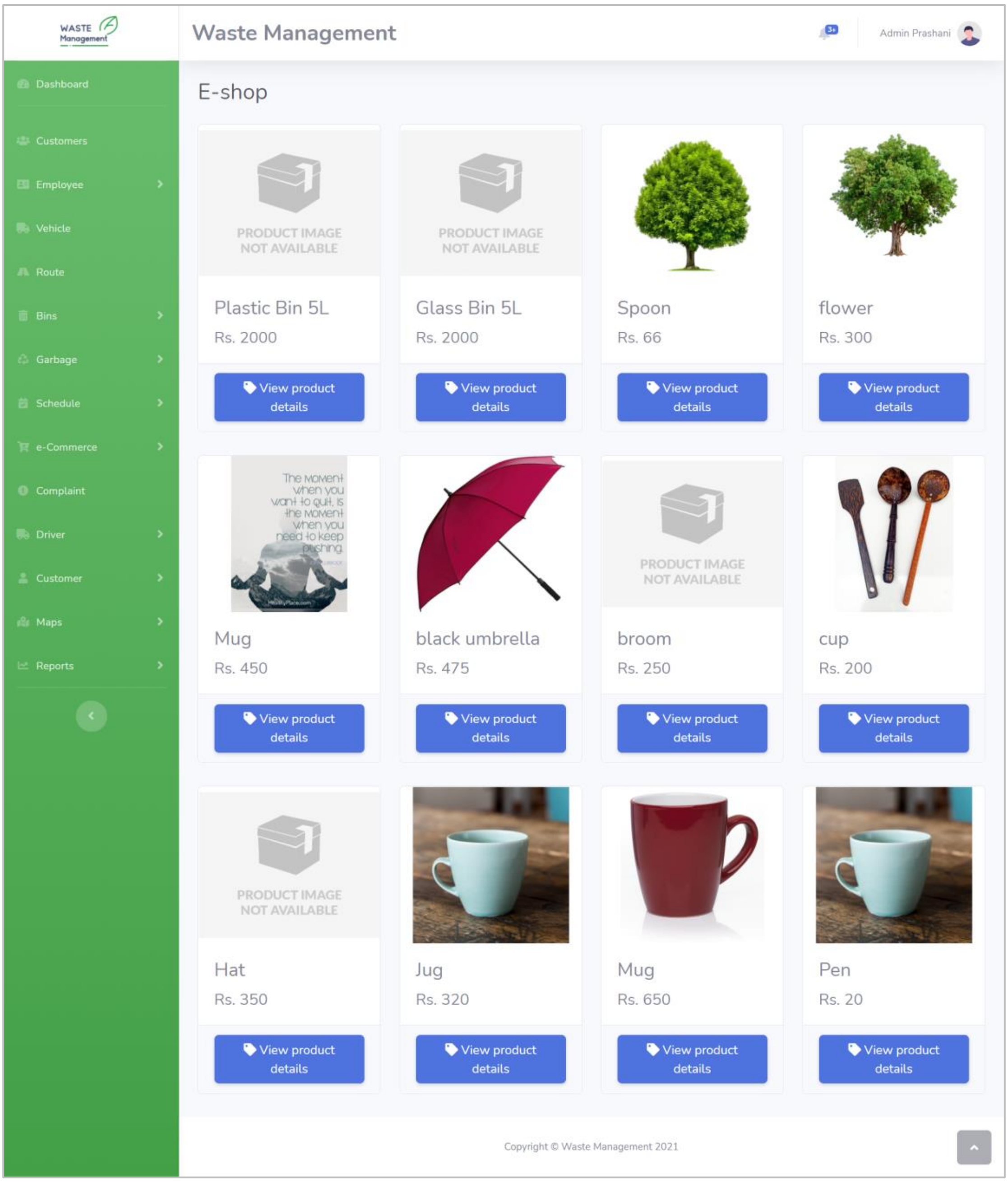
Image	Name	Quantity	Unit Price	Total
	Mug Size: N/A Color Code: #5128f0	3	Rs. 650	Rs. 1950
	Spoon Size: Large Color Code: #000000	2	Rs. 66	Rs. 132

Showing 1 to 2 of 2 entries

Sub-Total: Rs 2082
Discount -(5%): Rs
VAT +(11%): Rs
Total: Rs

Copyright © Waste Management 2021

Figure 9.11 - Shopping cart



Waste Management

E-shop

Plastic Bin 5L
Rs. 2000

Glass Bin 5L
Rs. 2000

Spoon
Rs. 66

flower
Rs. 300

Mug
Rs. 450

black umbrella
Rs. 475

broom
Rs. 250

cup
Rs. 200

Hat
Rs. 350

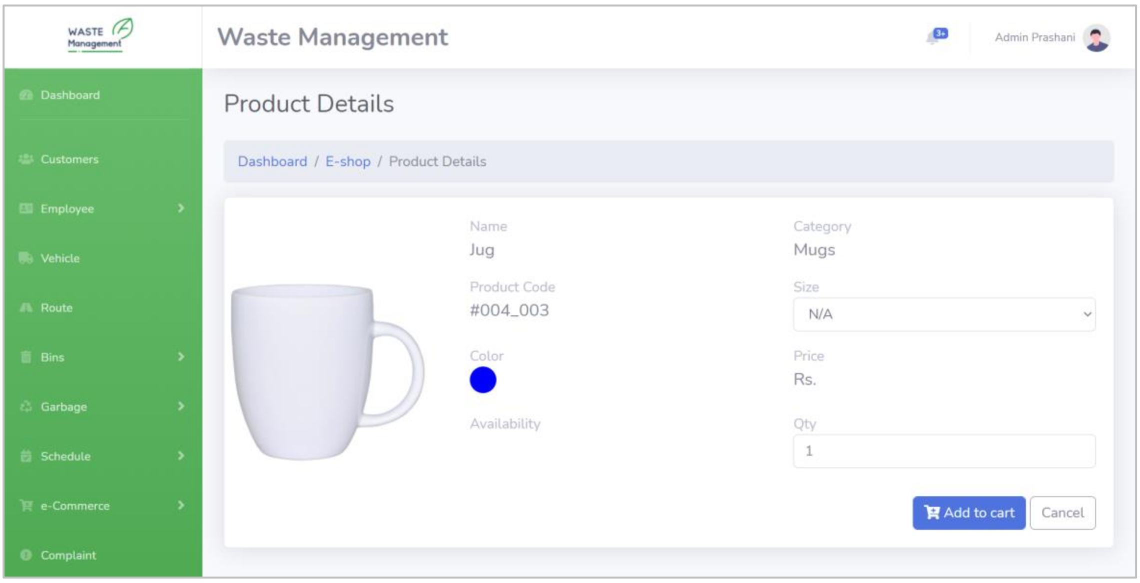
Jug
Rs. 320

Mug
Rs. 650

Pen
Rs. 20

Copyright © Waste Management 2021

Figure 9.12 - e-Shop



Product Details

Dashboard / E-shop / Product Details

Name
Jug

Category
Mugs

Product Code
#004_003

Size
N/A

Color
●

Price
Rs.

Availability

Qty
1

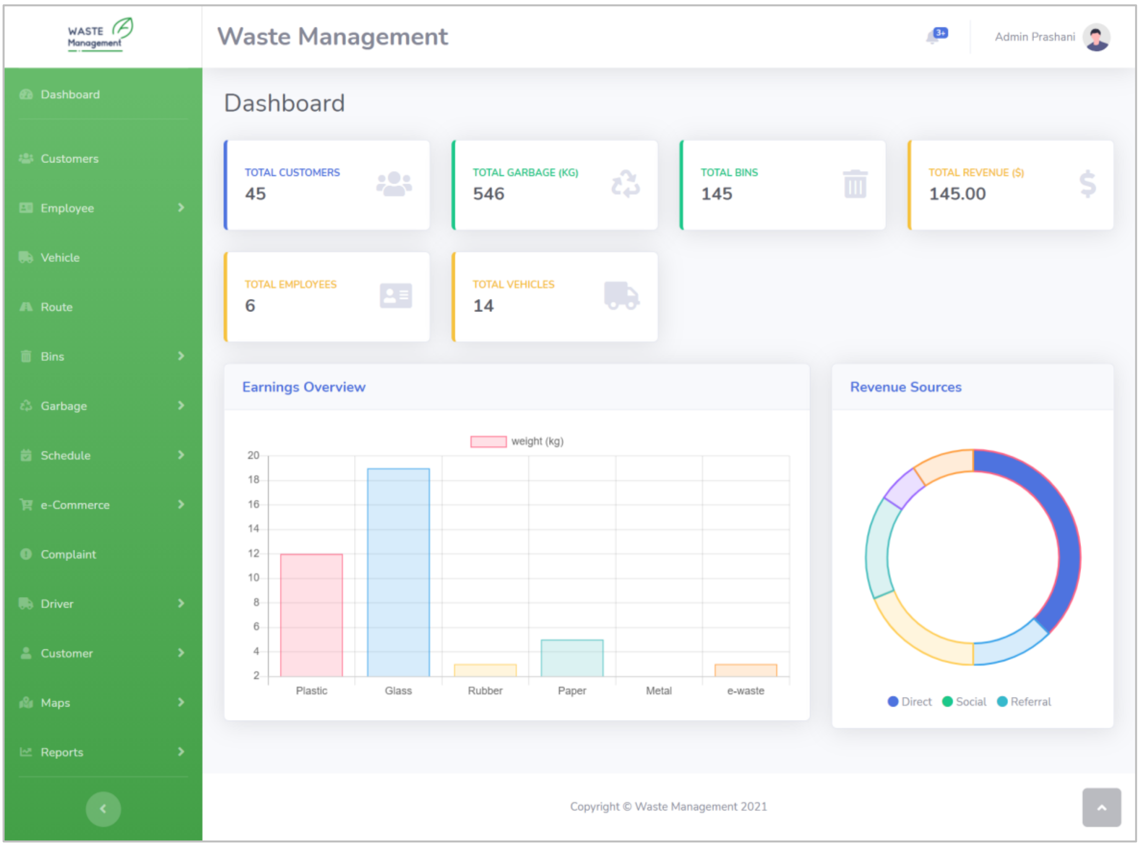
Figure 9.13 - Product details page

APPENDIX D - MANAGEMENT REPORTS

The most crucial component of the system is the report module. It determines the success or failure of the system in terms of usefulness to the end users. If the generated reports provide a clear view of the current situation or supports for decision making, then end users will be satisfied with the system. The report module in the current system provides numerous reports for all the users who interact with the system. Ex: staff, customers and drivers. The reports are generated in the form of charts and tables with filtering option.

10.1 DASHBOARD

Dashboard provides a comprehensive summery of all important aspects of the system like garbage, customers, vehicles, employees. When staff or admin login to the system then can have overall idea about the system with a quick galance.



Dashboard

TOTAL CUSTOMERS 45

TOTAL GARBAGE (KG) 546

TOTAL BINS 145

TOTAL REVENUE ($) 145.00

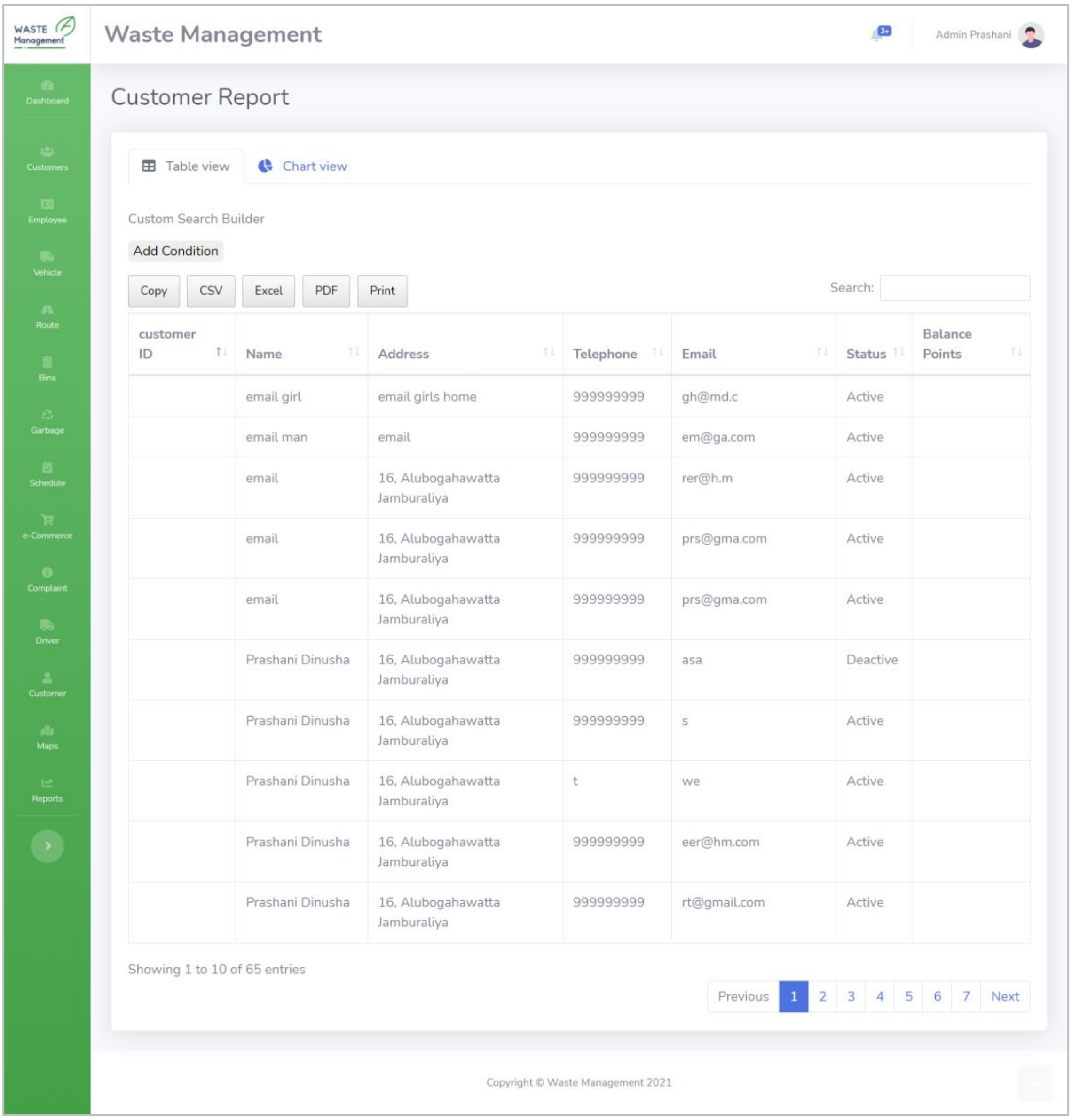
TOTAL EMPLOYEES 6

TOTAL VEHICLES 14

Figure 10.1 - Dashboard

10.2 CUSTOMERS’ DETAIL REPORT

Customer details are presented in tabular form with filtering from one or more fields. Reports can be downloaded in csv, excel or pdf format.



Customer Report

Custom Search Builder

customer ID	Name	Address	Telephone	Email	Status	Balance Points
	email girl	email girls home	999999999	gh@md.c	Active	
	email man	email	999999999	em@ga.com	Active	
	email	16, Alubogahawatta Jamburaliya	999999999	rer@h.m	Active	
	email	16, Alubogahawatta Jamburaliya	999999999	prs@gma.com	Active	
	email	16, Alubogahawatta Jamburaliya	999999999	prs@gma.com	Active	
	Prashani Dinusha	16, Alubogahawatta Jamburaliya	999999999	asa	Deactive	
	Prashani Dinusha	16, Alubogahawatta Jamburaliya	999999999	s	Active	
	Prashani Dinusha	16, Alubogahawatta Jamburaliya	t	we	Active	
	Prashani Dinusha	16, Alubogahawatta Jamburaliya	999999999	eer@hm.com	Active	
	Prashani Dinusha	16, Alubogahawatta Jamburaliya	999999999	rt@gmail.com	Active	

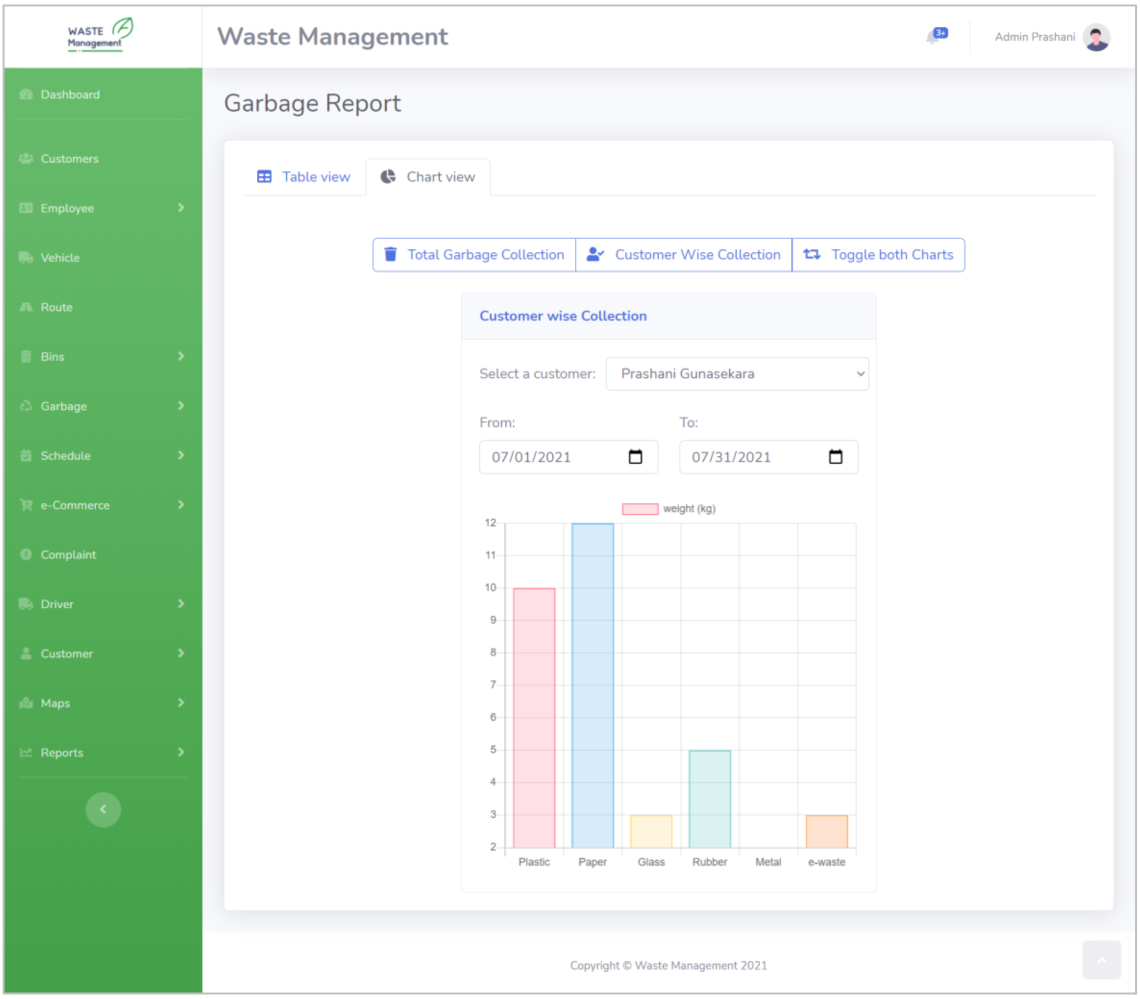
Showing 1 to 10 of 65 entries

Copyright © Waste Management 2021

Figure 10.2 - Customer report

10.3 GARBAGE REPORT

There are two types of reports are available in garbage module as total garbage collection report and customer wise reports. Both can be generated for a preferred time period. Also staff can view above details the tabular format and filter or print the data.



Garbage Report

Customer wise Collection

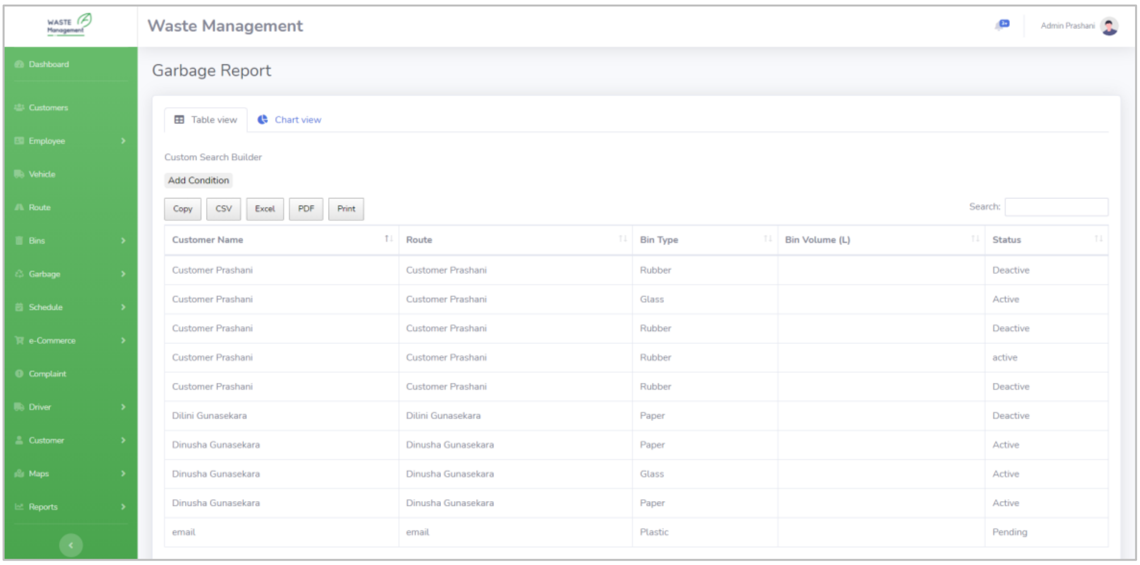
Select a customer: Prashani Gunasekara

From: 07/01/2021

To: 07/31/2021

Copyright © Waste Management 2021

Figure 10.3 - Customer wise garbage collection



Garbage Report

Custom Search Builder

Customer Name	Route	Bin Type	Bin Volume (L)	Status
Customer Prashani	Customer Prashani	Rubber		Deactive
Customer Prashani	Customer Prashani	Glass		Active
Customer Prashani	Customer Prashani	Rubber		Deactive
Customer Prashani	Customer Prashani	Rubber		active
Customer Prashani	Customer Prashani	Rubber		Deactive
Dilini Gunasekara	Dilini Gunasekara	Paper		Deactive
Dinusha Gunasekara	Dinusha Gunasekara	Paper		Active
Dinusha Gunasekara	Dinusha Gunasekara	Glass		Active
Dinusha Gunasekara	Dinusha Gunasekara	Paper		Active
email	email	Plastic		Pending

Figure 10.4 - Garbage report tabular format

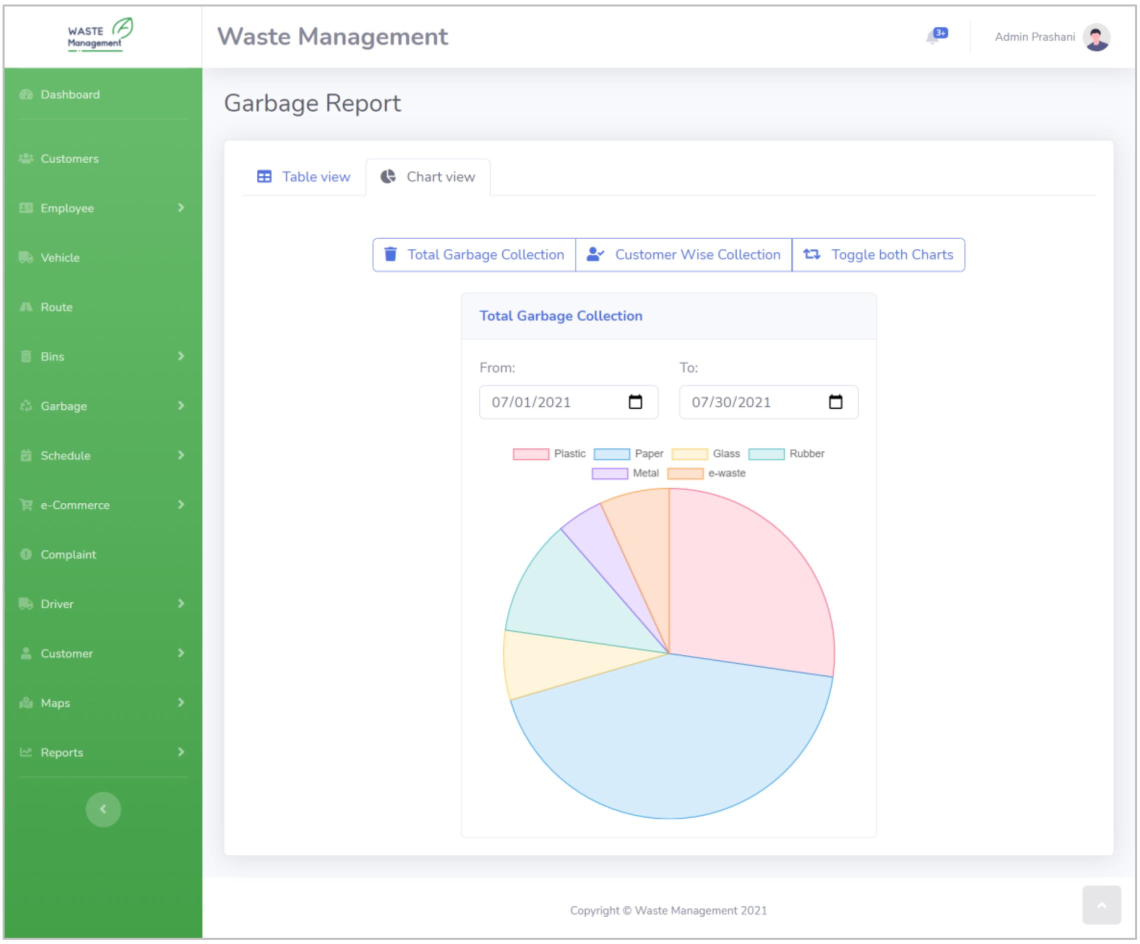
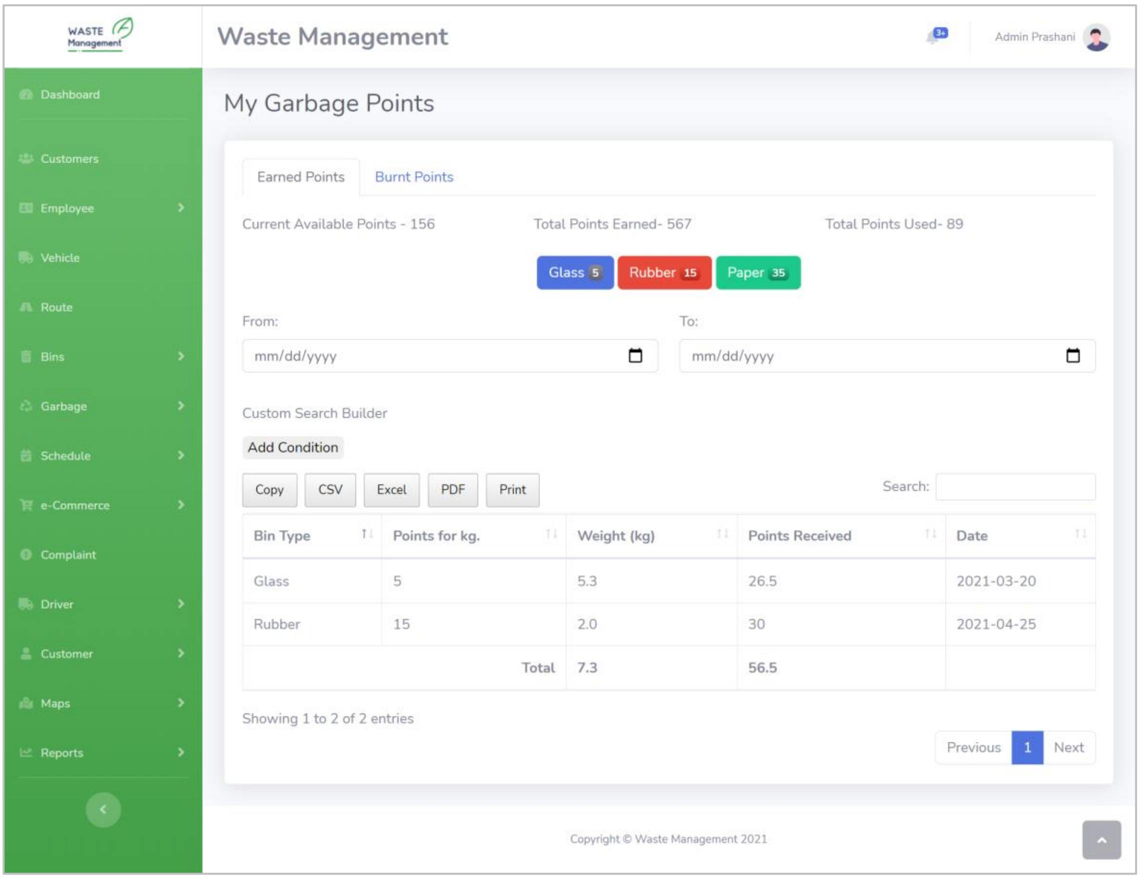


Figure 10.5 - Total garbage collection pie chart

10.4 GREEN POINTS REPORT

This report summarizes pointed earned and points spent in a given time period. This report is available for both customers and staff where staff can filter the records for a single customer.



My Garbage Points

Current Available Points - 156

Total Points Earned- 567

Total Points Used- 89

From: mm/dd/yyyy

To: mm/dd/yyyy

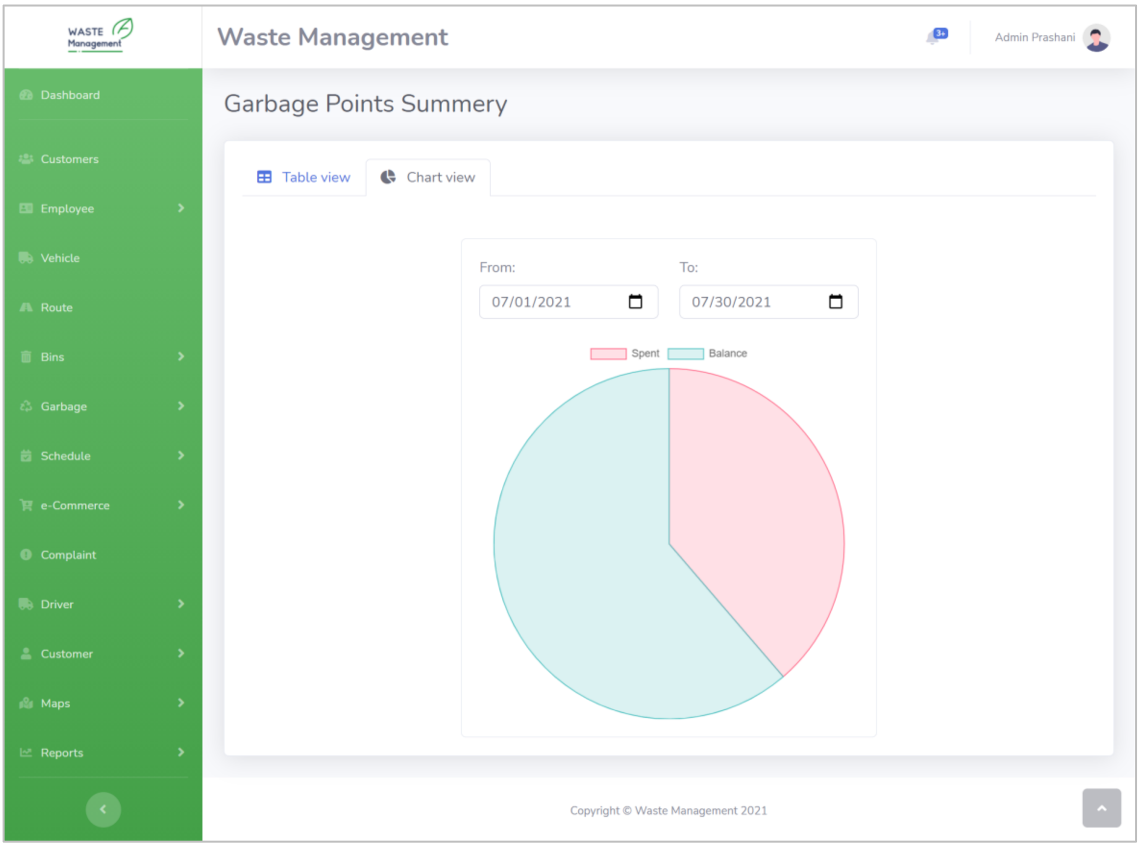
Custom Search Builder

Bin Type	Points for kg.	Weight (kg)	Points Received	Date
Glass	5	5.3	26.5	2021-03-20
Rubber	15	2.0	30	2021-04-25
	Total	7.3	56.5	

Showing 1 to 2 of 2 entries

Copyright © Waste Management 2021

Figure 10.6 - Green points earned details



Garbage Points Summery

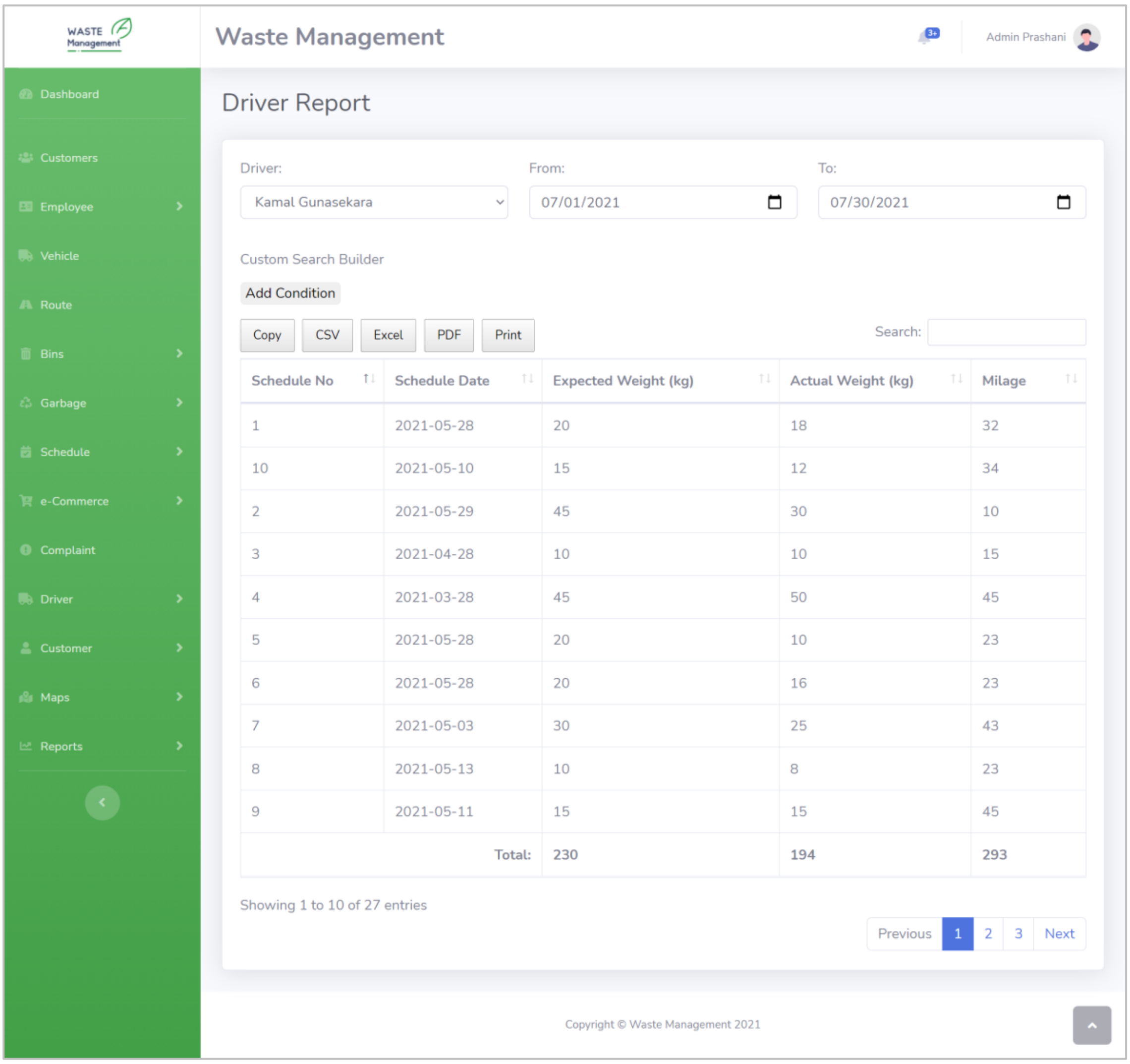
From: 07/01/2021

To: 07/30/2021

Figure 10.7 - Green points spending summery

10.5 DRIVER REPORT

Number of trips completed in each month can be viewed. A comparison of expected vs actual garbage collected from each type can be visualized.



Driver Report

Driver: Kamal Gunasekara

From: 07/01/2021

To: 07/30/2021

Custom Search Builder

Search:

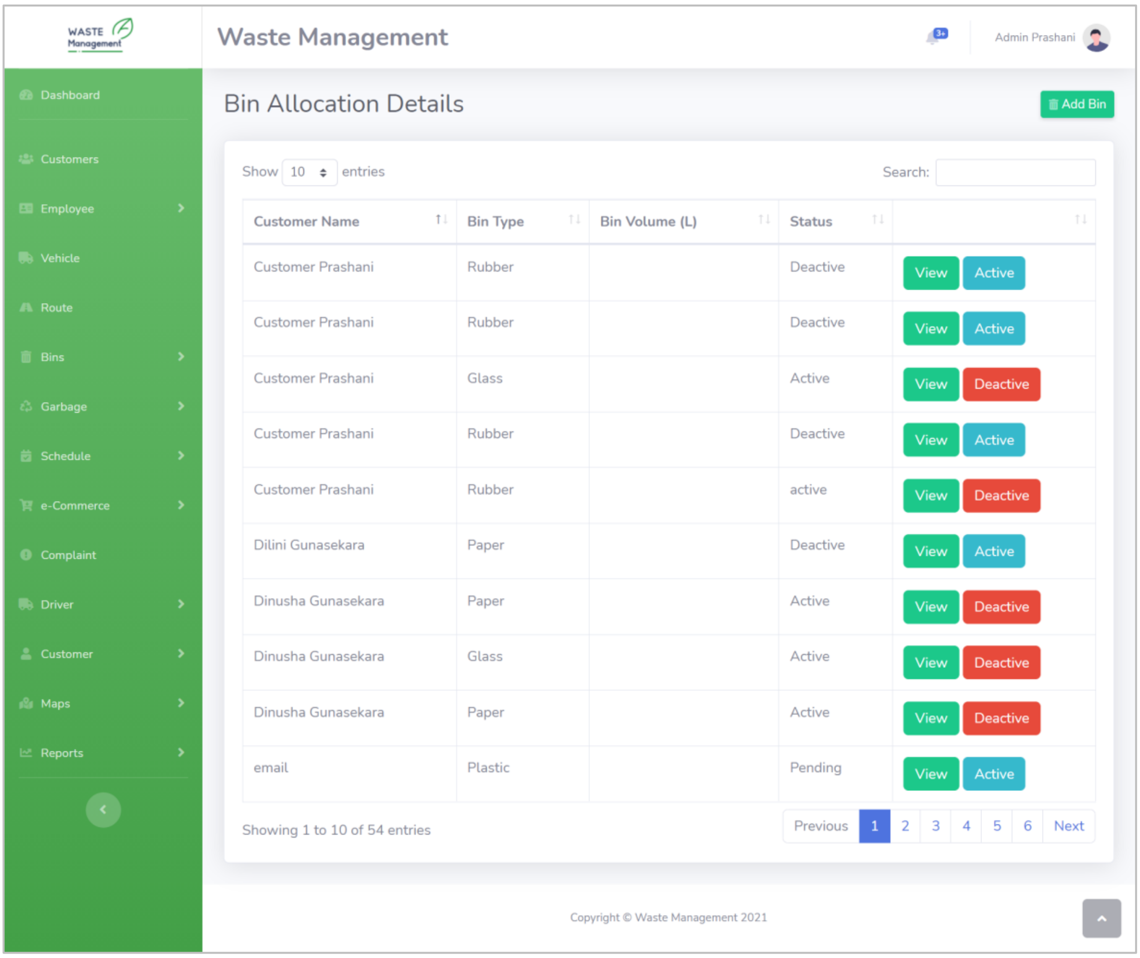
Schedule No	Schedule Date	Expected Weight (kg)	Actual Weight (kg)	Milage
1	2021-05-28	20	18	32
10	2021-05-10	15	12	34
2	2021-05-29	45	30	10
3	2021-04-28	10	10	15
4	2021-03-28	45	50	45
5	2021-05-28	20	10	23
6	2021-05-28	20	16	23
7	2021-05-03	30	25	43
8	2021-05-13	10	8	23
9	2021-05-11	15	15	45
	Total:	230	194	293

Showing 1 to 10 of 27 entries

Figure 10.8 - Driver report

10.6 BIN DETAIL REPORT

The staff can view the number of bins and types of bins allocated for each user.



Bin Allocation Details

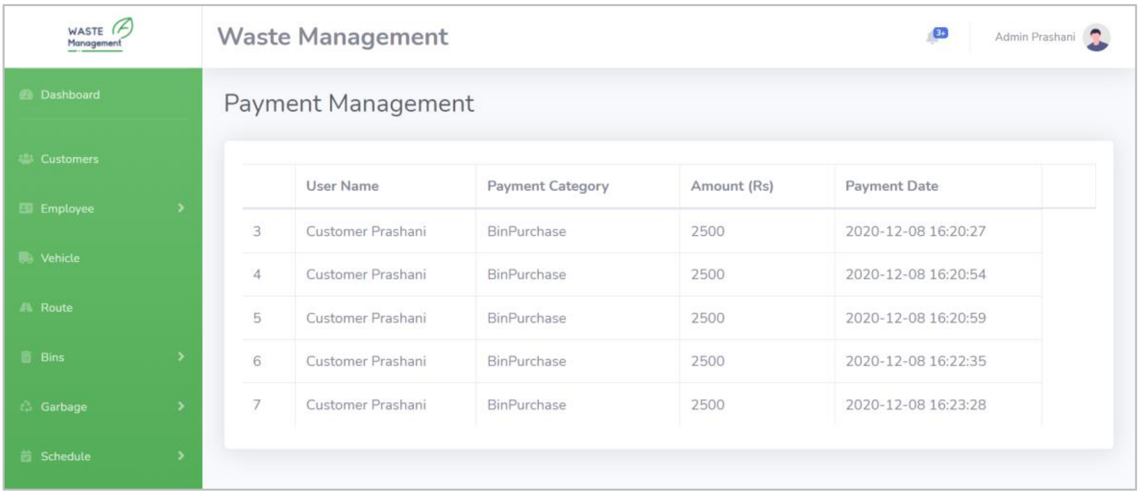
Customer Name	Bin Type	Bin Volume (L)	Status	
Customer Prashani	Rubber		Deactive	View Active
Customer Prashani	Rubber		Deactive	View Active
Customer Prashani	Glass		Active	View Deactive
Customer Prashani	Rubber		Deactive	View Active
Customer Prashani	Rubber		active	View Deactive
Dilini Gunasekara	Paper		Deactive	View Active
Dinusha Gunasekara	Paper		Active	View Deactive
Dinusha Gunasekara	Glass		Active	View Deactive
Dinusha Gunasekara	Paper		Active	View Deactive
email	Plastic		Pending	View Active

Showing 1 to 10 of 54 entries

Figure 10.9 - Bin allocation details

10.7 SALES REPORT

The staff can view monthly sales done through the ecommerce site.



Payment Management

	User Name	Payment Category	Amount (Rs)	Payment Date
3	Customer Prashani	BinPurchase	2500	2020-12-08 16:20:27
4	Customer Prashani	BinPurchase	2500	2020-12-08 16:20:54
5	Customer Prashani	BinPurchase	2500	2020-12-08 16:20:59
6	Customer Prashani	BinPurchase	2500	2020-12-08 16:22:35
7	Customer Prashani	BinPurchase	2500	2020-12-08 16:23:28

Figure 10.10 - Sales report

10.8 USER REGISTRATION REPORT

Number of customers registered for a given time period is available in this report.

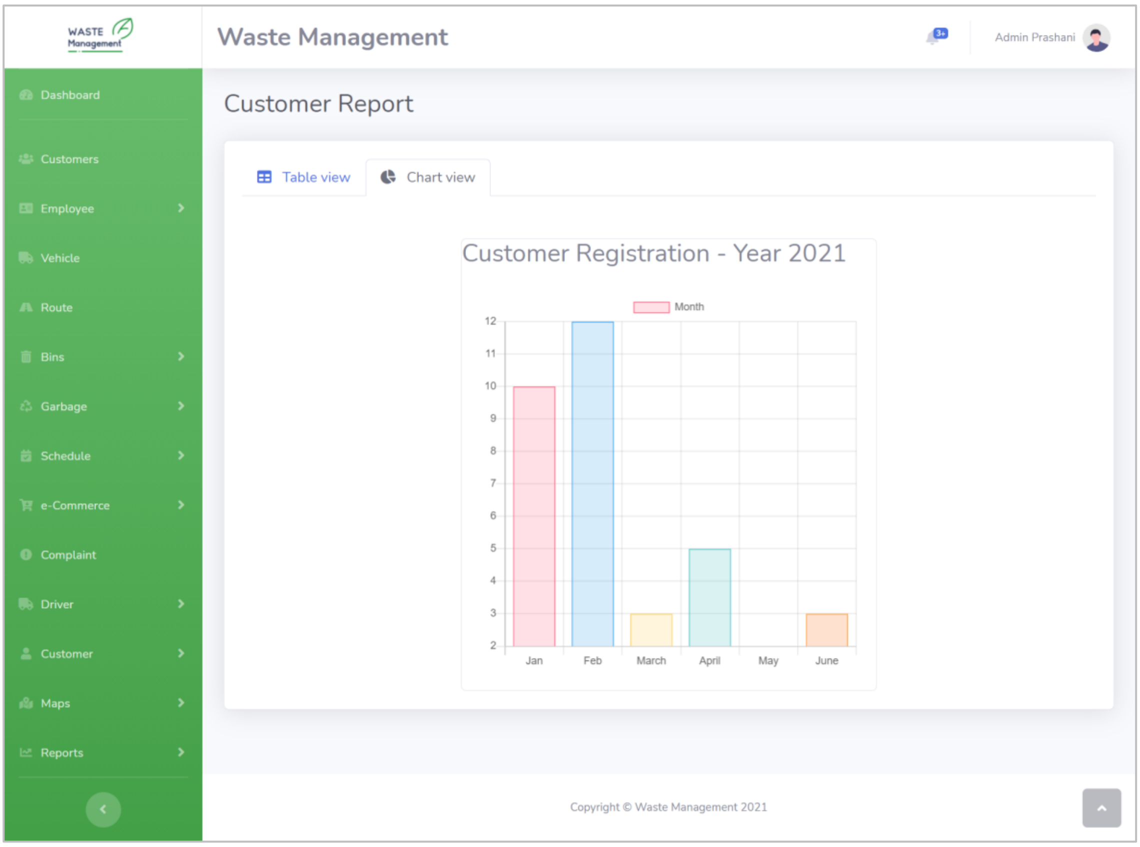


Figure 10.11 - Customer registration report

APPENDIX E - TEST PLAN

11.1 TEST OBJECTIVES

The objective of the test is to verify that the functionality of Smart System for Waste Management works according to the requirements.

11.2 SCOPE OF TESTING

- Functions to be tested
    - GUI
    - All the feature of the system
    - Search and Filters Logic
- Functions not to be tested
    - Everything that not mentioned above

11.3 TEST APPROACH

The project is using an agile approach, with incremental way. At the end of each increment the requirements identified for that time box will be delivered and will be tested.

The testing approach would be to do black-box & database level testing. Test cases & Test designs will be created after finish the requirement analysis phase, will be evolve further during design phase and executed during testing phase.

11.4 ENTRY CRITERIA

- Have completely understood the requirement
- source code must be tested while developing

- Have sound knowledge of functionality
- Review test scenarios, test cases

11.5 TEST TYPES

In this project, there are 3 types of testing will be conducted.

- Functional Testing: Functional testing will be performed to check the functions of application.
- Integration Testing: Individual software modules are combined and tested as a group
- System Testing: Application as a whole is tested for its compliance to functional and non-functional requirements

11.6 EXIT CRITERIA

- All the high priority/severity test cases has been executed
- Deliverables are ready
- High severity/ priority bugs are fixed

APPENDIX F - TEST RESULTS

Test cases given in evaluation stage along with test results are given in this chapter.

12.1 TEST CASES FOR CUSTOMER LOGIN

Test cases with expected results and actual results of the customer login are displayed in below Table 12.1.

TC Id	Priority	Severity	Description	Test Steps	Expected Result	Actual Result	Status
#1	P1	S1	Verify user able to login with a valid username and valid password.	1. User go to login page 2. Enter valid username and password 3. Click login button	User can login into the dashboard	User login to the dashboard	Pass
#2	P1	S1	Verify user cannot login with an invalid username and an invalid password.	1. User go to login page 2. Enter invalid username and password 3. Click login button	1. User unable to login 2. Error message should appears	1. Validation message appears 2. Unable to login to the system	Pass
#3	P1	S1	Verify when the username and password field is blank.	1. User go to login page 2. Keep username and password fields blank 3. Click login button	1. User unable to login 2. Error msg should appears	1. Validation message appears 2. Unable to login to the system	Pass
#4	P2	S1	Verify the messages for invalid login	1. User go to login page 2. Enter invalid username and password 3. Click login button	1. Content of the error msg should be correct	1. Validation message appears 2. Unable to login to the system	Pass
#5	P3	S2	Verify if the data in password field is either visible as asterisk or bullet signs.	1. User go to login page 2. Enter valid username and password	1. data in password field is should visible as asterisk or bullet signs.	password field is visible as asterisk	Pass

#6	P4	S4	Verify UI of the Login page	1. User go to login page	UI of the page should be correctly placed & sized/colored	UI correctly placed & sized/colored	Pass

Table 12.1 - Login test case

12.2 TEST CASES FOR CUSTOMER CREATION

Test cases with expected results and actual results of the customer creation are displayed in below Table 12.2.

TC Id	Priority	Severity	Description	Test Steps	Expected Result	Actual Result	Status
#1	P1	S1	Verify user able to add customer name	1. User go to registration page 2. Add name	1. User can add customer name	can add customer name	Pass
#2	P1	S1	Verify user able to add contact No	1. User go to registration page 2. Add contact no	1. User can add contact No	can add contact No	Pass
#3	P1	S1	Verify user able to add email	1. User go to registration page 2. Add email	1. User can add email	can add email	Pass
#4	P1	S1	Verify user able to add address	1. User go to registration page 2. Add address	1. User can add address	can add address	Pass
#5	P1	S1	Verify user able to add longitude	1. User go to registration page 2. Add longitude	1. User can add longitude	can add longitude	Pass
#6	P1	S1	Verify user able to add latitude	1. User go to registration page 2. Add latitude	1. User can add latitude	can add latitude	Pass
#7	P3	S3	Verify the clear button	1. User go to registration page 2. Add content 3. Click clear button	1. Field data should get reset	Data get reset	Pass
#8	P1	S1	Verify Validations for	1. User go to registration page	1. Unable to proceed further if compulsory	1. Unable to proceed further if	Pass

			compulsory fields	2. Check compulsory field validation for all required fields	fields are missing 2. Validation message should shows	compulsory fields are missing 2. Validation message shows	
#9	P1	S1	Verify after creating a profile staff can view it	1. Go to staff login 2. Go to customer 3. Open created profile	1. Staff can view the created profile with correct details	Can view the profile	Pass
#10	P1	S1	Verify after creating a profile user can login to it	1. Create a profile 2. Login to system using created profile details	1. User can login to the profile	Can login into the system	Pass

Table 12.2 - Customer creation test cases

12.3 TEST CASES FOR GARBAGE MODULE

Test cases with actual vs expected results for the garbage module are displayed in below Table 12.3.

TC Id	Priority	Severity	Description	Test Steps	Expected Result	Actual Result	Status
#1	P1	S1	Staff can view past transactions related to one bin	1. go to garbage bin management window 2. click on view button	able to view past transactions related to one bin	Can view past transactions	Pass
#2	P2	S2	Customer can view past transactions related to one bin	1. go to my bins window 2. click on view button	able to view past transactions related to one bin	Can view past transactions	Pass
#3	P1	S1	Customer can able to make bin full by clicking a button	1. go to my bins window 2. click on bin full button	1. Bin status changed to full 2. Staff receive a notification	Can change bin status to full	Pass
#4	P1	S1	Customer can able to make bin full by using IoT unit	1. add garbage to basket until its full	1. Bin status changed to full 2. Staff receive a notification	Can change bin status to full	Pass

#5	P1	S1	Staff can able to make bin full by clicking a button	1. go to garbage bin management window 2. click on bin full button	1. Bin status changed to full 2. Staff receive a notification	Can change bin status to full	Pass
#6	P1	S1	Staff can able to make bin empty by clicking a button	1. go to garbage bin management window 2. click on bin empty button 3. add weight details	1. Bin status changed to empty 2. weight details can add 3. weight information will be showing in transaction, green points pages	1. Can change bin status to empty 2. weight details get recorded	Pass
#7	P1	S1	After driver submitting bin weight bin status changed to empty	1. go to garbage bin management window 2. click on bin empty button 3. add weight details	1. Bin status changed to empty 2. weight details can add 3. weight information will be showing in transaction, green points pages	1. Can change bin status to empty 2. weight details get recorded	Pass

Table 12.3 - Test cases for garbage module

12.4 TEST CASES FOR BIN ALLOCATION MODULE

Test cases with actual and expected results for the bin allocation module are displayed in below Table 12.4.

TC Id	Prior ity	Seve rity	Description	Test Steps	Expected Result	Actual Result	Stat us
#1	P1	S1	Customer can request a bin	1. login to system 2. go to bin allocation page 3. click on request a bin button 4. fill the form & save details	1. chose bin should shows under customer my bins window 2. bin status should shows as pending	Can request a bin	Pass

#2	P1	S1	Staff can assign a bin to a customer	1. login to system 2. go to bin allocation page 3. click on add a bin button 4. fill the form & save details	1. chose bin should shows under customer my bins window 2. newly added bin details should show in bin allocation details page 3. bin status should shows as pending	Can assign a bin	Pass
#3	P2	S2	When requesting a bin previously allocated bin list shows	1. login to system 2. go to bin allocation page 3. click on add a bin button	1. previously added bin details shows in a table	previously allocated bin list shows	Pass
#4	P2	S1	View allocated bins to a given customer	1. login to system 2. go to bin allocation page 3. click on view button	1. all the allocated bin details should show	Can View allocated bins to a given customer	Pass
#5	P1	S1	Active a bin by customer	1. go to my bin page 2. click on activate bin button 3. enter activation code received	1. bin status should change to active 2. customer able to start using the bin	Can activate bin	Pass
#6	P1	S1	Try to activate bin by wrong activation code	1. go to my bin page 2. click on activate bin button 3. enter invalid activation code	1. validation message should be appears 2. bin should be in pending status	Unable to activate bin	Pass
#7	P1	S1	Active a bin by staff	1. go to bin allocation page 2. click on active button	1. staff should be able to activate bin without activation code	Can activate bin	Pass
#8	P1	S1	Deactivate a bin by staff	1. go to bin allocation page 2. click on Deactivate button	1. staff should be able to deactivate bin	Can deactivate bin	Pass

Table 12.4 - Bin allocation module test case

12.5 TEST CASES FOR E-COMMERCE MODULE

Test cases with actual and expected results for the bin allocation module are displayed in below Table 12.5.

TC Id	Priority	Severity	Description	Test Steps	Expected Result	Actual Result	Status
#1	P1	S1	Customer can view product list	1. go to e-shop	can view product list	can view product list	Pass
#2	P1	S1	Customer can view a product detail	1. go to e-shop 2. click on a product	can view a product detail page	can view a product detail page	Pass
#3	P1	S1	Customer can add a product to a cart	1. go to e-shop 2. click on a product 3. select size/color/qty 4. click add to cart	Product can add to cart	Product can add to cart	Pass
#4	P1	S1	Customer can add multiple same product to a cart	1. go to e-shop 2. click on a product 3. select multiple qty 4. click add to cart	Same product can add to cart multiple times	Same product can add to cart multiple times	Pass
#5	P1	S1	View shopping cart after adding product	1. go to e-shop 2. click on a product 3. select size/color/qty 4. click add to cart	All added products are in the cart	All added products are in the cart	Pass
#6	P2	S1	Continue shopping by adding more items to the cart	1. go to e-shop 2. click on a product 3. select size/color/qty 4. click add to cart 5. go to cart window 6. click continue shopping button	User will be redirect to product list page	Redirected to product list page	Pass
#7	P2	S1	Remove added items	1. add items to the cart 2. click on remove button	Removed products should get removed from the cart	Removed products get removed	Pass

						from the cart	
#8	P1	S1	Go to check out page	1. go to e-shop 2. click on a product 3. select size/color/qty 4. click add to cart 5. go to cart window 6. click on check out button	User will be redirect to checkout page	redirect to checkout page	Pass
#9	P1	S1	Try to checkout when sufficient credits are there	1. go to e-shop 2. click on a product 3. select size/color/qty 4. click add to cart 5. go to cart window 6. click on check out button	User can do the payment successfully	can do the payment successfully	Pass
#10	P1	S1	Try to checkout when sufficient credits are not there	1. go to e-shop 2. click on a product 3. select size/color/qty 4. click add to cart 5. go to cart window 6. click on check out button	Validation message should be appears	Validation message be appears	Pass
#11	P2	S2	Verify payment completed page and email	1. go to e-shop 2. click on a product 3. select size/color/qty 4. click add to cart 5. go to cart window 6. click on check out button 7. click do the payment button	User can do the payment successfully and will receive an email regarding purchase details	can do the payment successfully and receive an email regarding purchase details	Pass

#12	P1	S1	Validate cart is empty after checkout	1. go to e-shop 2. click on a product 3. select size/color/qty 4. click add to cart 5. go to cart window 6. click on check out button 7. click do the payment button	Cart is empty	Cart is empty	Pass

Table 12.5 - e-Commerce module test case

12.6 TEST CASES FOR VEHICLE MODULE

Test cases with actual and expected results for the vehicle module are displayed in below Table 12.6.

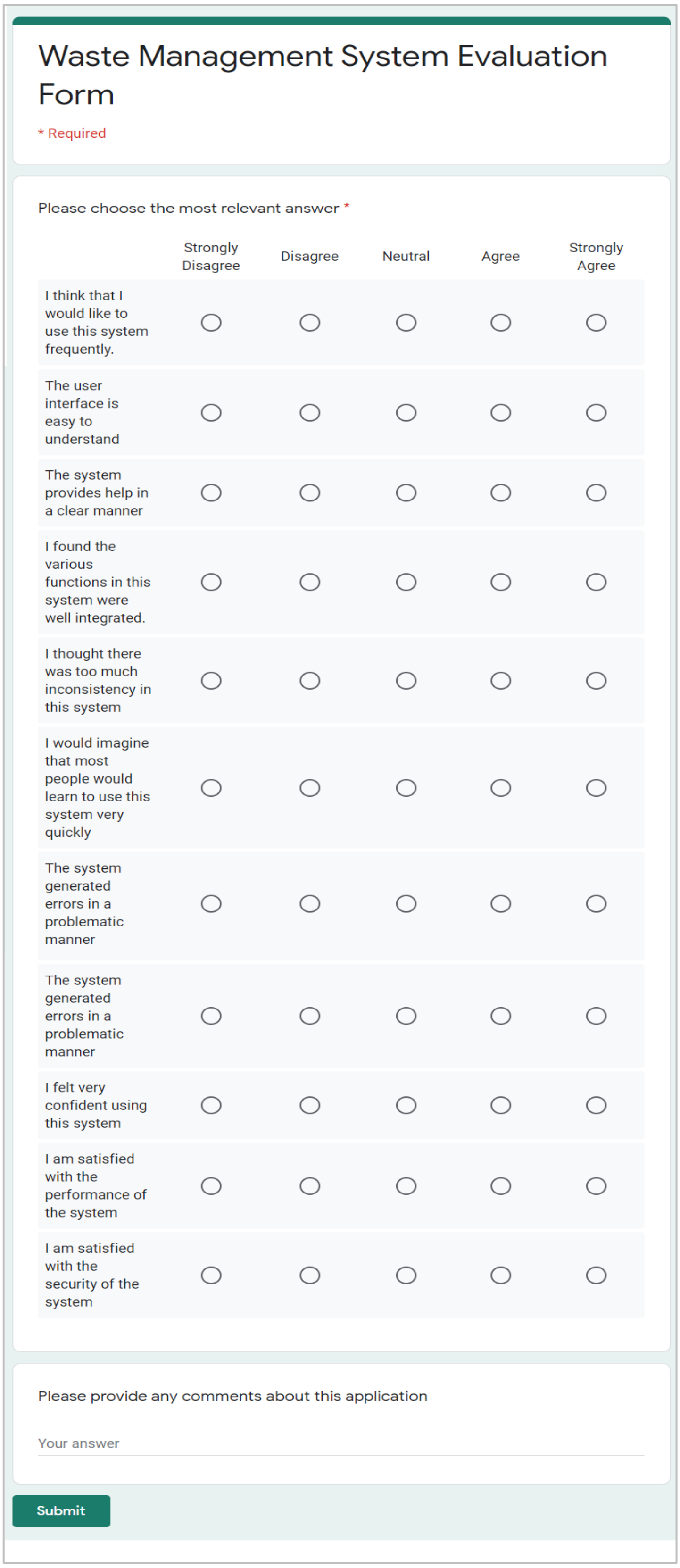
TC Id	Priority	Severity	Description	Test Steps	Expected Result	Actual Result	Status
#1	P1	S1	Add a new vehicle to the system	1. go to Vehicle Management page 2. click add a vehicle 3. fill the form and submit	New vehicle should be added to the system	New vehicle added	Pass
#2	P1	S2	Unable to add the same vehicle twice	1. go to Vehicle Management page 2. click add a vehicle 3. fill the form using previously added details and submit	1. Validation message should appears 2. Unable to add the same vehicle twice	Unable to add a vehicle	Pass
#3	P1	S1	View added vehicle details	1. go to Vehicle Management page 2. click view vehicle button	Able to see vehicle details	View vehicle details	Pass

#4	P2	S2	Update vehicle details	1. go to Vehicle Management page 2. click update vehicle button	Able to update and save vehicle details	Update Vehicle details	Pass
#5	P1	S2	Deactivate vehicle	1. go to Vehicle Management page 2. click Deactivate vehicle button	Vehicle get deactivated	Vehicle get deactivated	Pass
#6	P1	S1	Active vehicle	1. go to Vehicle Management page 2. click active vehicle button	Deactivated vehicle get active	Deactivated vehicle get active	Pass

Table 12.6 - Test cases for vehicle module

12.7 USER ACCEPTANCE EVALUATION FORM

Sample user acceptance form given in Figure 12.1.



Waste Management System Evaluation Form

* Required

Please choose the most relevant answer *

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I think that I would like to use this system frequently.	○	○	○	○	○
The user interface is easy to understand	○	○	○	○	○
The system provides help in a clear manner	○	○	○	○	○
I found the various functions in this system were well integrated.	○	○	○	○	○
I thought there was too much inconsistency in this system	○	○	○	○	○
I would imagine that most people would learn to use this system very quickly	○	○	○	○	○
The system generated errors in a problematic manner	○	○	○	○	○
The system generated errors in a problematic manner	○	○	○	○	○
I felt very confident using this system	○	○	○	○	○
I am satisfied with the performance of the system	○	○	○	○	○
I am satisfied with the security of the system	○	○	○	○	○

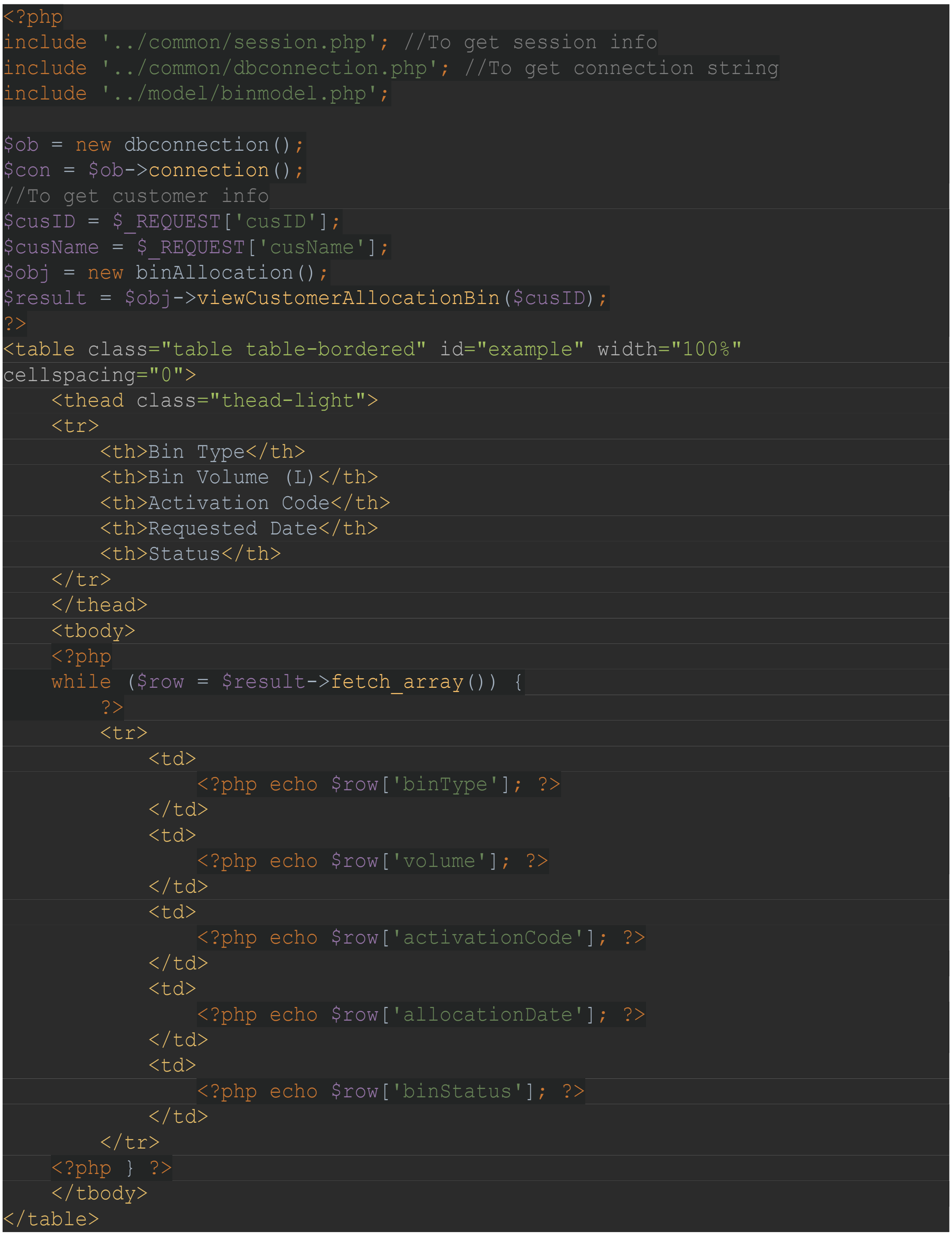
Please provide any comments about this application

Your answer

Figure 12.1 - Sample user acceptance evaluation form

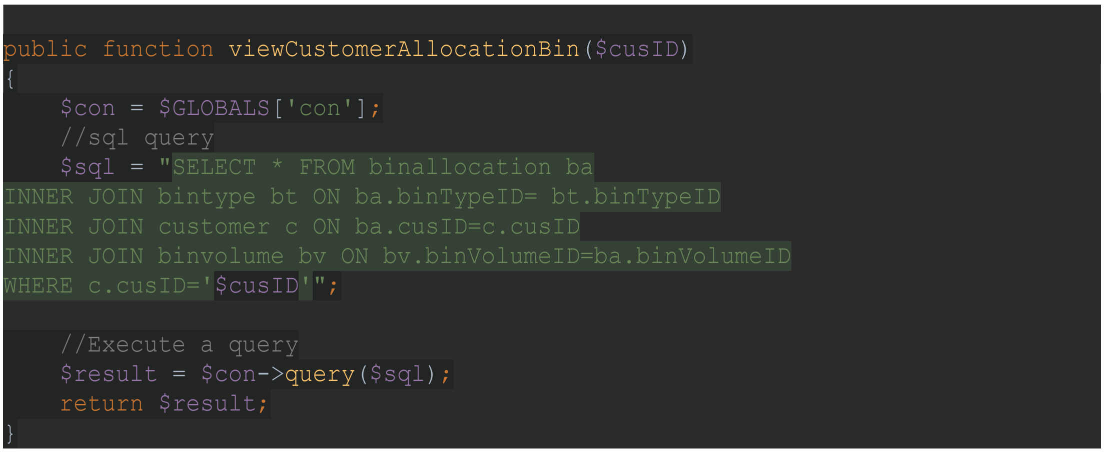
APPENDIX G - CODE LISTING

13.1 CODE SNIPPET FOR VIEW CUSTOMER ALLOCATED BINS



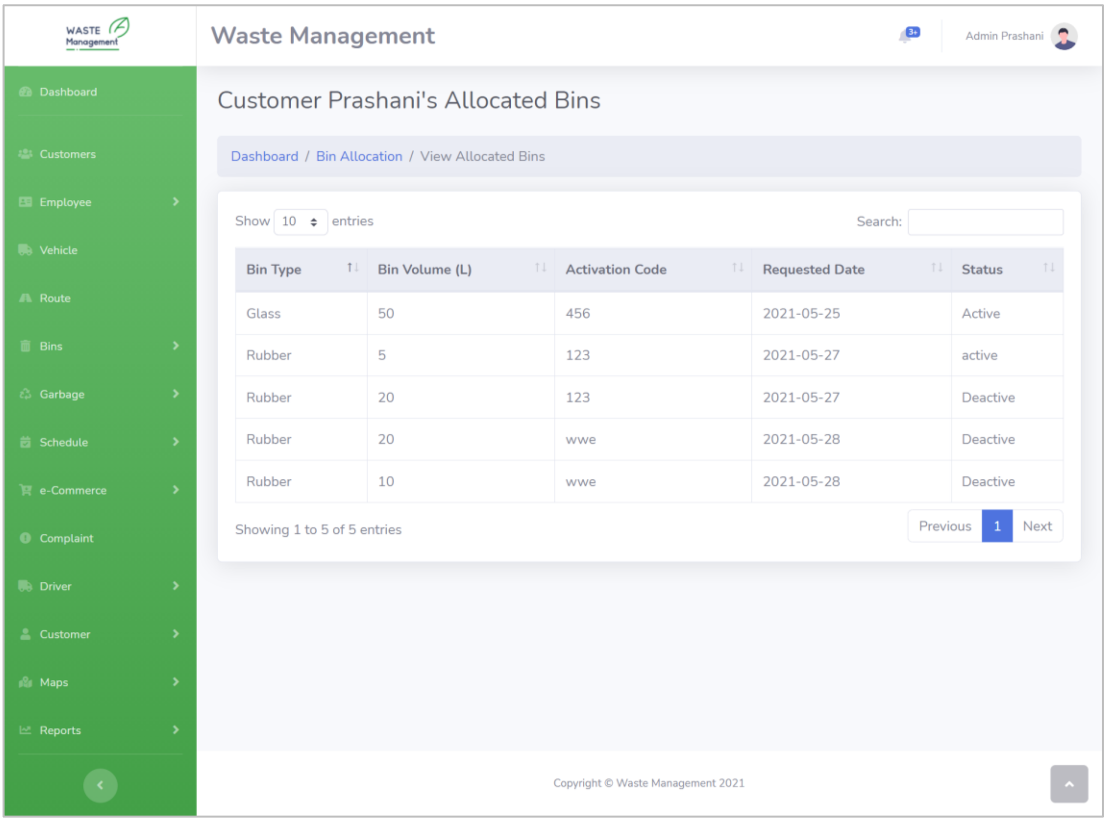
```
<?php
include '../common/session.php'; //To get session info
include '../common/dbconnection.php'; //To get connection string
include '../model/binmodel.php';

$ob = new dbconnection();
$con = $ob->connection();
//To get customer info
$cusID = $_REQUEST['cusID'];
$cusName = $_REQUEST['cusName'];
$obj = new binAllocation();
$result = $obj->viewCustomerAllocationBin($cusID);
?>
<table class="table table-bordered" id="example" width="100%"
cellspacing="0">
    <thead class="thead-light">
    <tr>
        <th>Bin Type</th>
        <th>Bin Volume (L)</th>
        <th>Activation Code</th>
        <th>Requested Date</th>
        <th>Status</th>
    </tr>
    </thead>
    <tbody>
    <?php
    while ($row = $result->fetch_array()) {
        ?>
        <tr>
            <td>
                <?php echo $row['binType']; ?>
            </td>
            <td>
                <?php echo $row['volume']; ?>
            </td>
            <td>
                <?php echo $row['activationCode']; ?>
            </td>
            <td>
                <?php echo $row['allocationDate']; ?>
            </td>
            <td>
                <?php echo $row['binStatus']; ?>
            </td>
        </tr>
    <?php } ?>
    </tbody>
</table>
```

Figure 13.1 - Code snippet for view allocated bins



```
public function viewCustomerAllocationBin($cusID)
{
    $con = $GLOBALS['con'];
    //sql query
    $sql = "SELECT * FROM binallocation ba
INNER JOIN bintype bt ON ba.binTypeID= bt.binTypeID
INNER JOIN customer c ON ba.cusID=c.cusID
INNER JOIN binvolume bv ON bv.binVolumeID=ba.binVolumeID
WHERE c.cusID='$cusID'";

    //Execute a query
    $result = $con->query($sql);
    return $result;
}
```

Figure 13.2 - Code snippet for bin model customer allocated bins



Customer Prashani's Allocated Bins

Dashboard / Bin Allocation / View Allocated Bins

Show 10 entries Search:

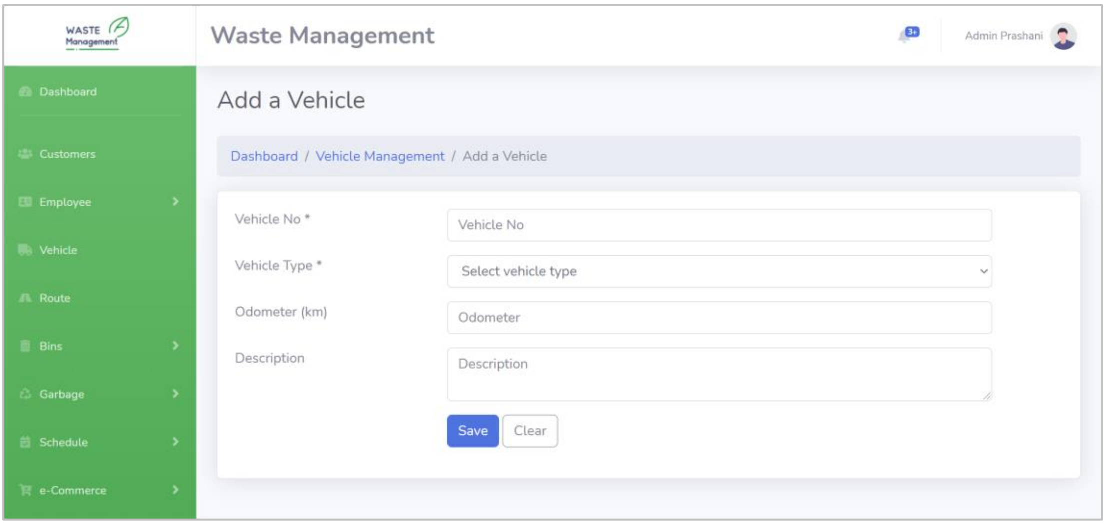
Bin Type	Bin Volume (L)	Activation Code	Requested Date	Status
Glass	50	456	2021-05-25	Active
Rubber	5	123	2021-05-27	active
Rubber	20	123	2021-05-27	Deactive
Rubber	20	wwe	2021-05-28	Deactive
Rubber	10	wwe	2021-05-28	Deactive

Showing 1 to 5 of 5 entries

Copyright © Waste Management 2021

Figure 13.3 - Screen view – Allocated bins

13.2 CODE SNIPPET FOR ADD/UPDATE VEHICLE



Add a Vehicle

Dashboard / Vehicle Management / Add a Vehicle

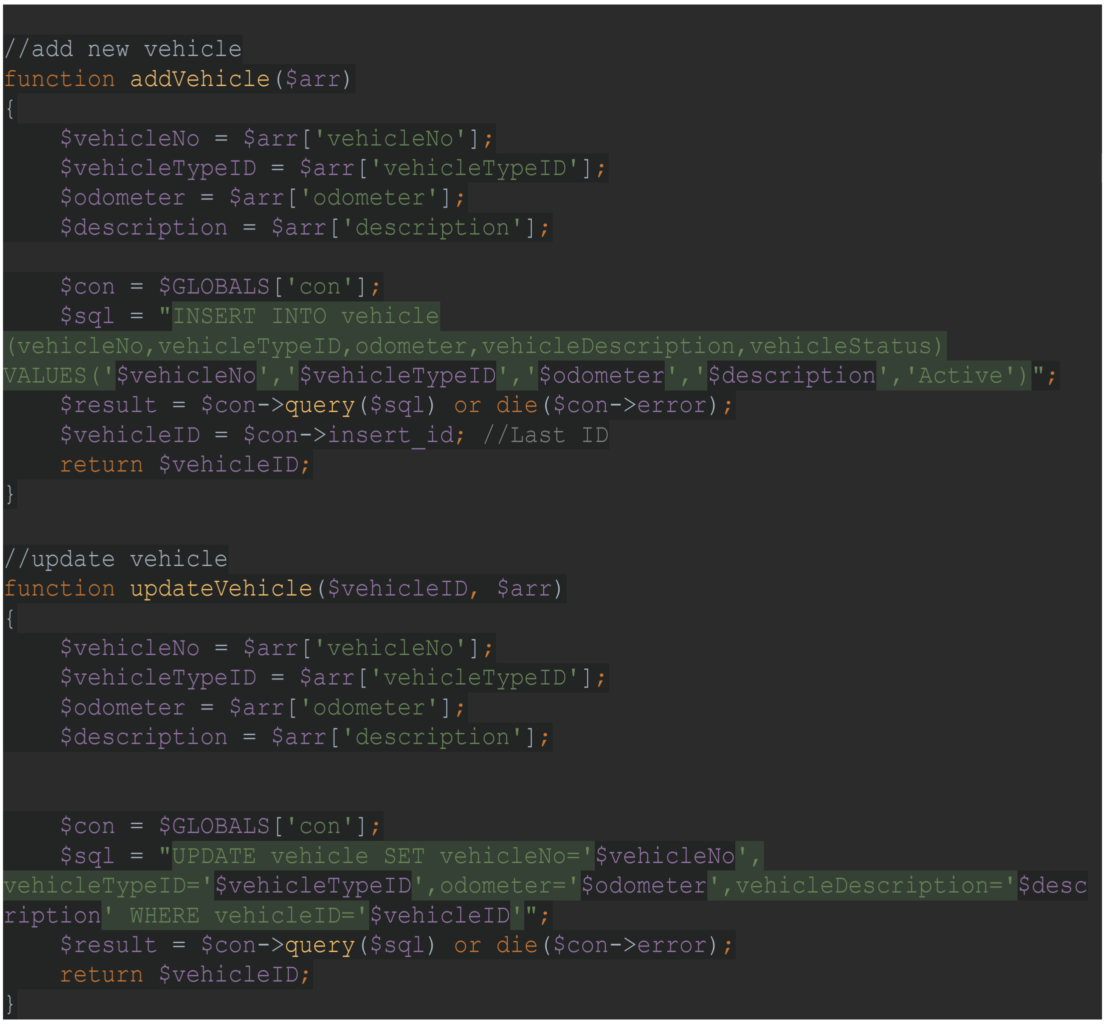
Vehicle No * — Vehicle No

Vehicle Type * — Select vehicle type

Odometer (km) — Odometer

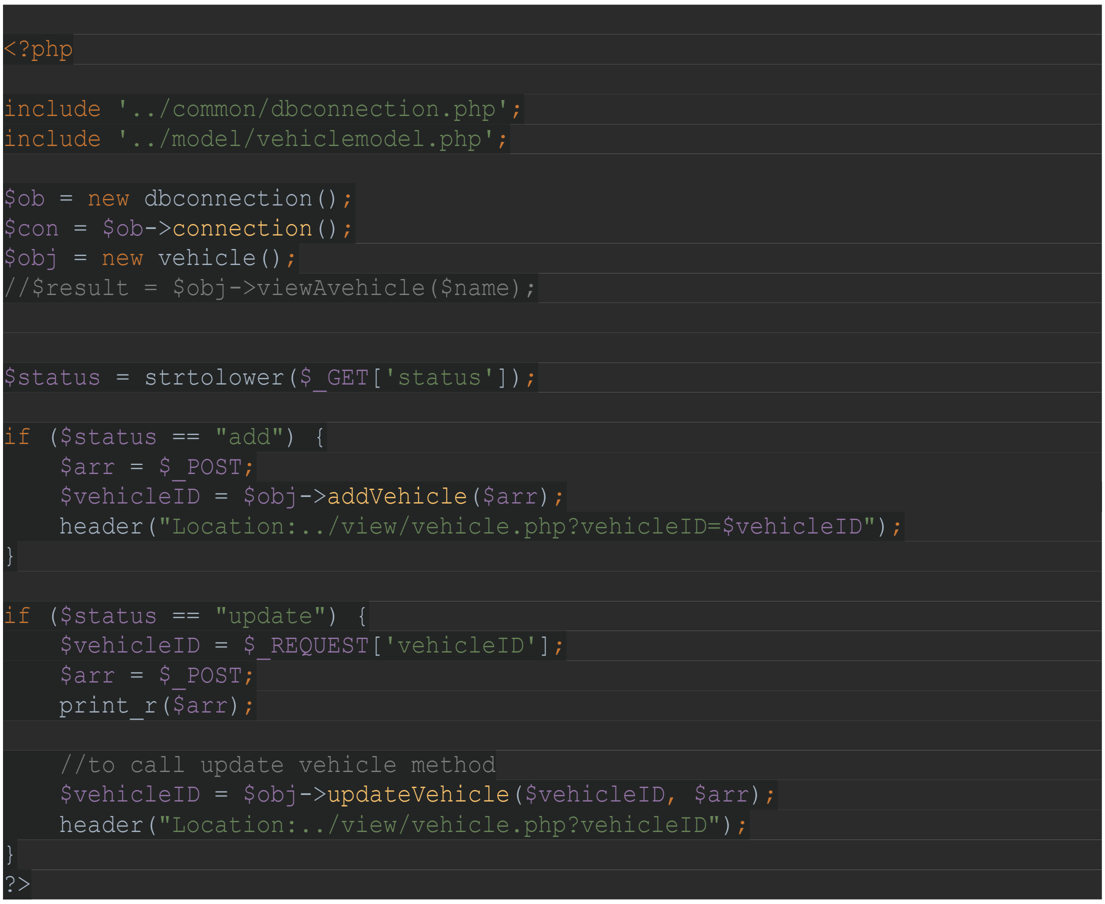
Description — Description

Figure 13.4 - Screen view – Add vehicle



```
//add new vehicle
function addVehicle($arr)
{
    $vehicleNo = $arr['vehicleNo'];
    $vehicleTypeID = $arr['vehicleTypeID'];
    $odometer = $arr['odometer'];
    $description = $arr['description'];

    $con = $GLOBALS['con'];
    $sql = "INSERT INTO vehicle
(vehicleNo,vehicleTypeID,odometer,vehicleDescription,vehicleStatus)
VALUES('$vehicleNo','$vehicleTypeID','$odometer','$description','Active')";
    $result = $con->query($sql) or die($con->error);
    $vehicleID = $con->insert_id; //Last ID
    return $vehicleID;
}

//update vehicle
function updateVehicle($vehicleID, $arr)
{
    $vehicleNo = $arr['vehicleNo'];
    $vehicleTypeID = $arr['vehicleTypeID'];
    $odometer = $arr['odometer'];
    $description = $arr['description'];


    $con = $GLOBALS['con'];
    $sql = "UPDATE vehicle SET vehicleNo='$vehicleNo',
vehicleTypeID='$vehicleTypeID',odometer='$odometer',vehicleDescription='$desc
ription' WHERE vehicleID='$vehicleID'";
    $result = $con->query($sql) or die($con->error);
    return $vehicleID;
}
```

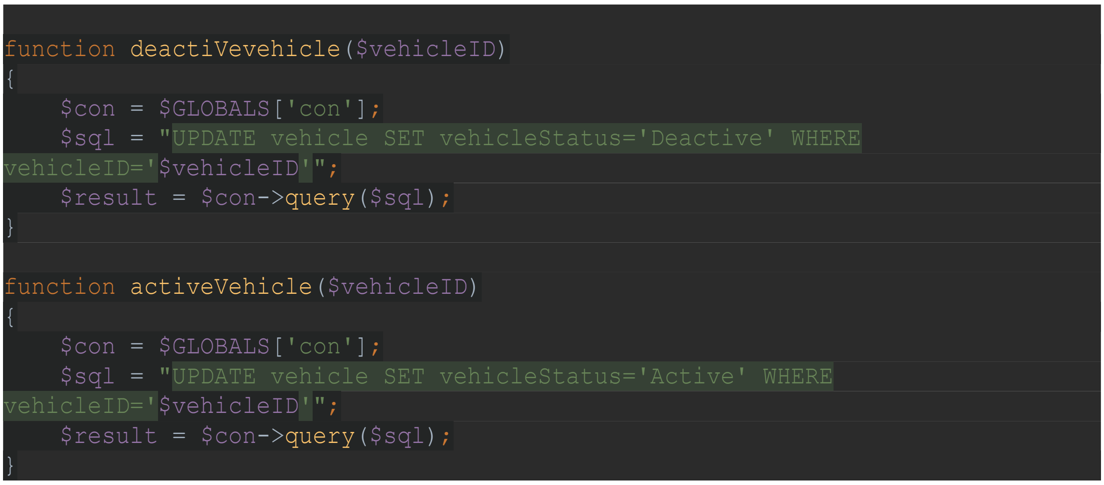
Figure 13.5 - Code snippet for vehicle model add/update



```php
<?php

include '../common/dbconnection.php';
include '../model/vehiclemodel.php';

$ob = new dbconnection();
$con = $ob->connection();
$obj = new vehicle();
//$result = $obj->viewAvehicle($name);


$status = strtolower($_GET['status']);

if ($status == "add") {
    $arr = $_POST;
    $vehicleID = $obj->addVehicle($arr);
    header("Location:../view/vehicle.php?vehicleID=$vehicleID");
}

if ($status == "update") {
    $vehicleID = $_REQUEST['vehicleID'];
    $arr = $_POST;
    print_r($arr);

    //to call update vehicle method
    $vehicleID = $obj->updateVehicle($vehicleID, $arr);
    header("Location:../view/vehicle.php?vehicleID");
}
?>
```

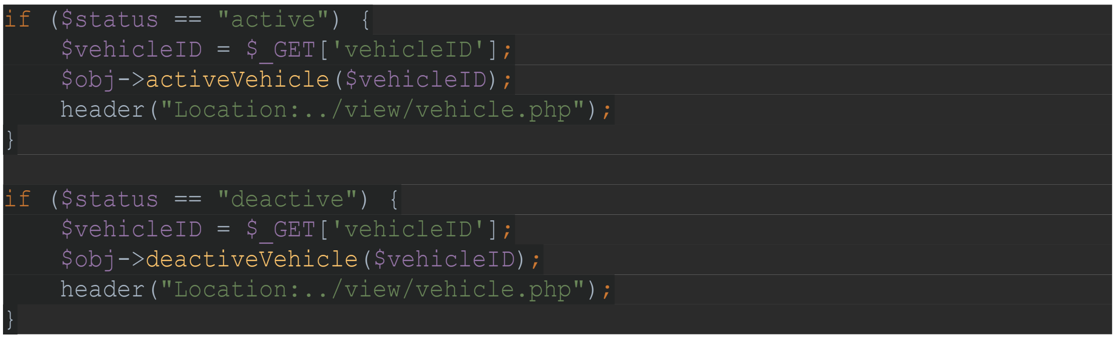
Figure 13.6 - Code snippet for vehicle controller add/update

13.3 CODE SNIPPET FOR ACTIVE/DEACTIVATE VEHICLE



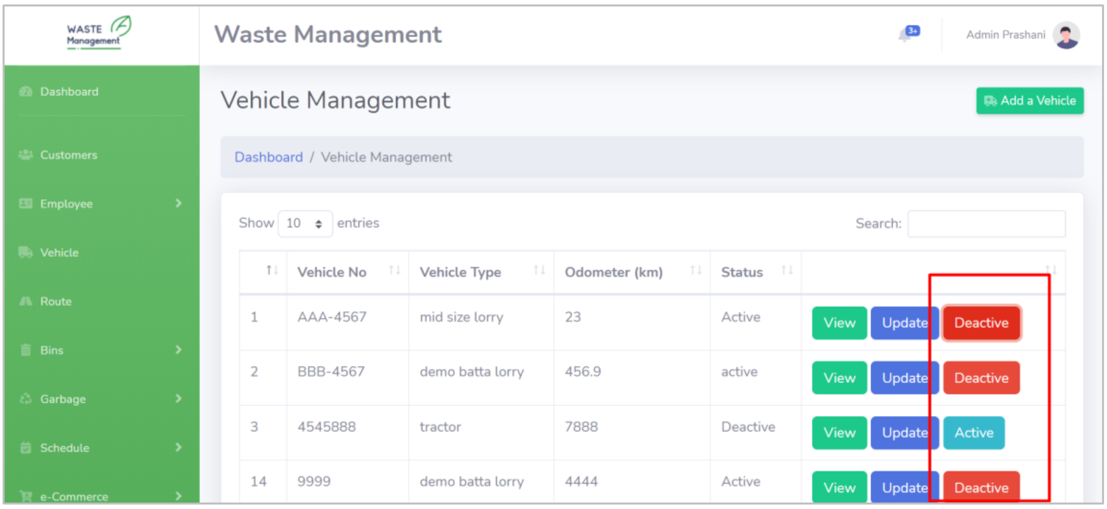
```php
function deactiVevehicle($vehicleID)
{
    $con = $GLOBALS['con'];
    $sql = "UPDATE vehicle SET vehicleStatus='Deactive' WHERE
vehicleID='$vehicleID'";
    $result = $con->query($sql);
}

function activeVehicle($vehicleID)
{
    $con = $GLOBALS['con'];
    $sql = "UPDATE vehicle SET vehicleStatus='Active' WHERE
vehicleID='$vehicleID'";
    $result = $con->query($sql);
}
```

Figure 13.7 - Code snippet for vehicle model active/deactivate vehicle



```
if ($status == "active") {
    $vehicleID = $_GET['vehicleID'];
    $obj->activeVehicle($vehicleID);
    header("Location:../view/vehicle.php");
}

if ($status == "deactive") {
    $vehicleID = $_GET['vehicleID'];
    $obj->deactiveVehicle($vehicleID);
    header("Location:../view/vehicle.php");
}
```

Figure 13.8 - Code snippet for vehicle controller active/deactivate vehicle



Vehicle Management

Dashboard / Vehicle Management

Show 10 entries

Search:

	Vehicle No	Vehicle Type	Odometer (km)	Status	
1	AAA-4567	mid size lorry	23	Active	View Update Deactive
2	BBB-4567	demo batta lorry	456.9	active	View Update Deactive
3	4545888	tractor	7888	Deactive	View Update Active
14	9999	demo batta lorry	4444	Active	View Update Deactive

Figure 13.9 - Screen view – Active/deactivate vehicle

13.4 CODE SNIPPET FOR BIN MAP VIEW

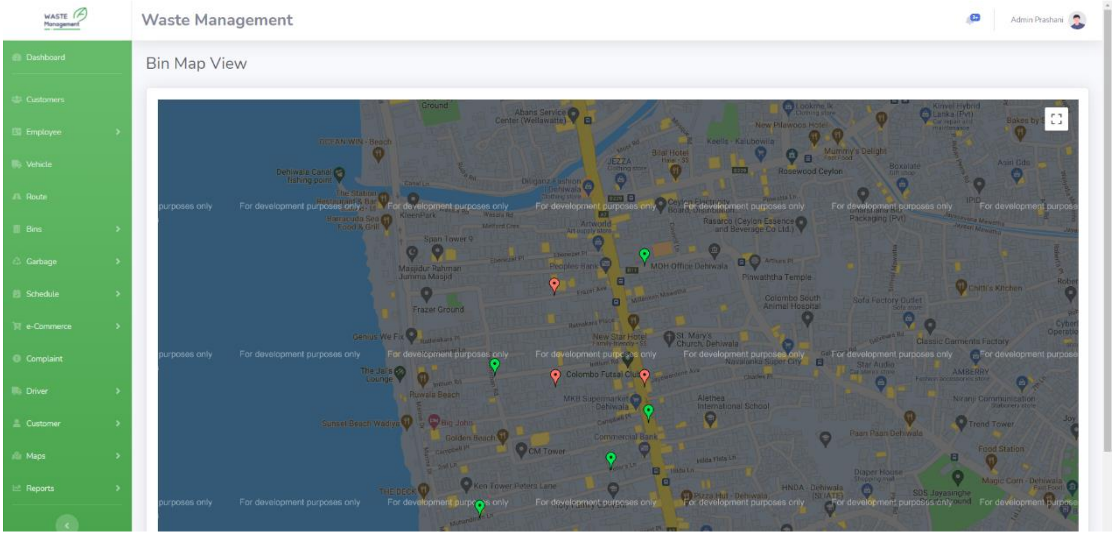
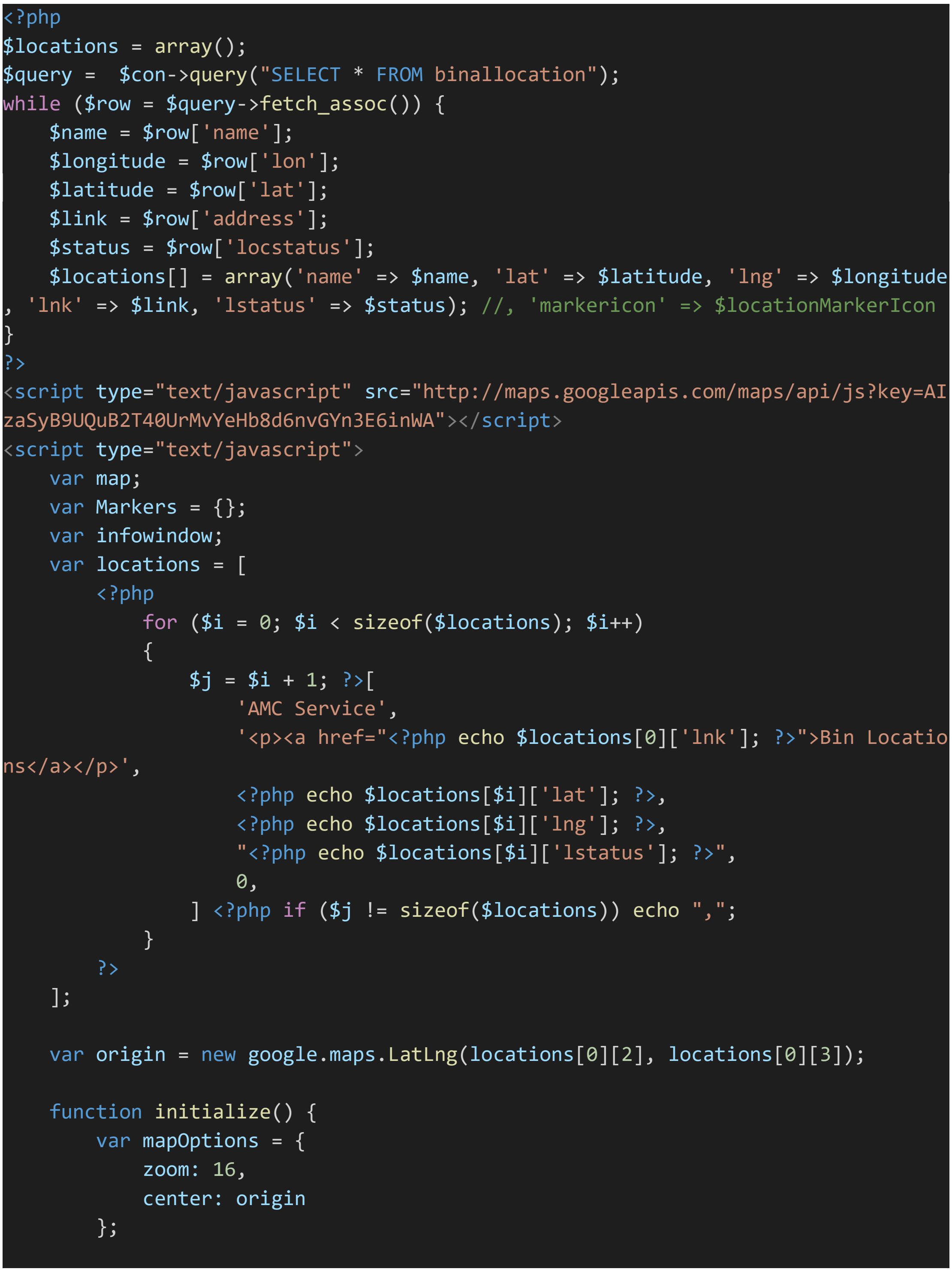
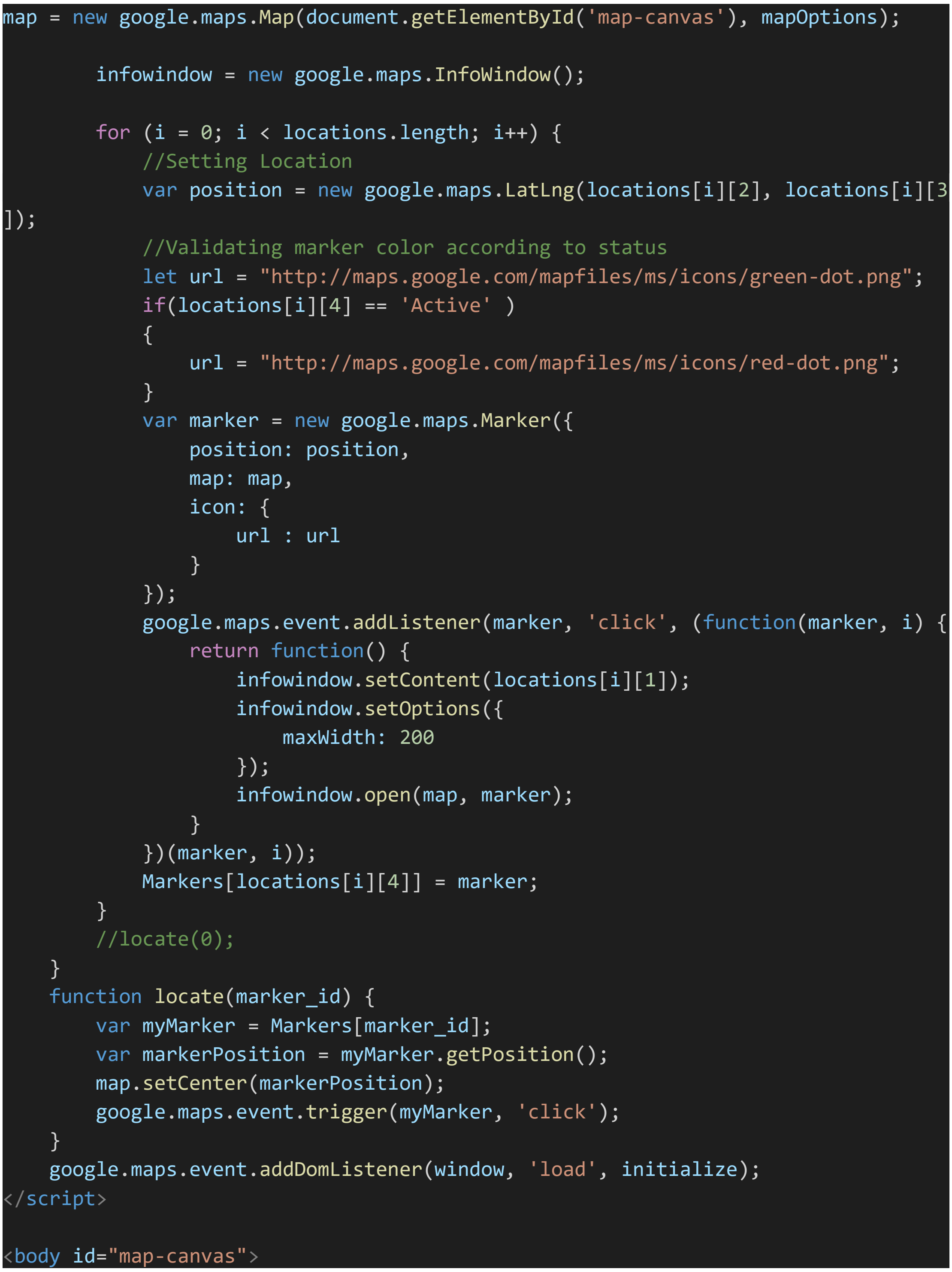


Figure 13.10 - Screen view – Bin map view



```
<?php
$locations = array();
$query =  $con->query("SELECT * FROM binallocation");
while ($row = $query->fetch_assoc()) {
    $name = $row['name'];
    $longitude = $row['lon'];
    $latitude = $row['lat'];
    $link = $row['address'];
    $status = $row['locstatus'];
    $locations[] = array('name' => $name, 'lat' => $latitude, 'lng' => $longitude
, 'lnk' => $link, 'lstatus' => $status); //, 'markericon' => $locationMarkerIcon
}
?>
<script type="text/javascript" src="http://maps.googleapis.com/maps/api/js?key=AI
zaSyB9UQuB2T40UrMvYeHb8d6nvGYn3E6inWA"></script>
<script type="text/javascript">
    var map;
    var Markers = {};
    var infowindow;
    var locations = [
        <?php
            for ($i = 0; $i < sizeof($locations); $i++)
            {
                $j = $i + 1; ?>[
                    'AMC Service',
                    '<p><a href="<?php echo $locations[0]['lnk']; ?>">Bin Locatio
ns</a></p>',
                    <?php echo $locations[$i]['lat']; ?>,
                    <?php echo $locations[$i]['lng']; ?>,
                    "<?php echo $locations[$i]['lstatus']; ?>",
                    0,
                ] <?php if ($j != sizeof($locations)) echo ",";
            }
        ?>
    ];

    var origin = new google.maps.LatLng(locations[0][2], locations[0][3]);

    function initialize() {
        var mapOptions = {
            zoom: 16,
            center: origin
        };
```



```
map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);

        infowindow = new google.maps.InfoWindow();

        for (i = 0; i < locations.length; i++) {
            //Setting Location
            var position = new google.maps.LatLng(locations[i][2], locations[i][3
]);
            //Validating marker color according to status
            let url = "http://maps.google.com/mapfiles/ms/icons/green-dot.png";
            if(locations[i][4] == 'Active' )
            {
                url = "http://maps.google.com/mapfiles/ms/icons/red-dot.png";
            }
            var marker = new google.maps.Marker({
                position: position,
                map: map,
                icon: {
                    url : url
                }
            });
            google.maps.event.addListener(marker, 'click', (function(marker, i) {
                return function() {
                    infowindow.setContent(locations[i][1]);
                    infowindow.setOptions({
                        maxWidth: 200
                    });
                    infowindow.open(map, marker);
                }
            })(marker, i));
            Markers[locations[i][4]] = marker;
        }
        //locate(0);
    }
    function locate(marker_id) {
        var myMarker = Markers[marker_id];
        var markerPosition = myMarker.getPosition();
        map.setCenter(markerPosition);
        google.maps.event.trigger(myMarker, 'click');
    }
    google.maps.event.addDomListener(window, 'load', initialize);
</script>

<body id="map-canvas">
```

Figure 13.11 - Code snippet for map bin view

GLOSSARY

AJAX - Method of building interactive applications for the Web that process user requests immediately.

Bootstrap - Front-end website and web app development framework

CSS –Cascading Style Sheets used to format the layout of Web pages

HTML – Hypertext Markup Language, a standardized system for tagging text files to World Wide Web pages.

jQuery - A cross-platform JavaScript library designed to simplify the client-side scripting of HTML

MySQL - Open source relational database management system (RDBMS) based on Structured Query Language (SQL)

UML - Unified Modeling Language. Standardized modeling language enabling developers to construct and document artifacts of a software system

XAMPP - Free and open source cross-platform web server developed by Apache Friends

Green Points – Points received by customer based on garbage weight they handed over to the company.

INDEX