BUSITEMA UNIVERSITY
Pursuing Excellence

P.O. Box 236, Tororo, Uganda
Gen: +256 - 45 444 8838
Fax: +256 - 45 4436517
Email: info@adm.busitema.ac.ug
www.busitema.ac.ug

FACULTY OF SCIENCE AND EDUCATION

FINAL YEAR PROJECT REPORT

SMART WASTE MANAGEMENT SYSTEM
(Case study: Ecopick company mbale City)

By

KIPRUTO CALEB

(REG. NO. BU/UP/2023/1493)

0778849761

Kiprutocaleb2023@gmail.com

A PROJECT REPORT SUBMITTED TO THE FACULTY OF SCIENCE AND EDUCATION IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY,

OF BUSITEMA UNIVERSITY.

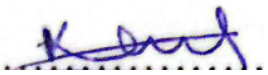
SUPERVISOR: DR. MUTUNGI FREDRICK

DEPARTMENT OF COMPUTER STUDIES

MAY 2026

DECLARATION.

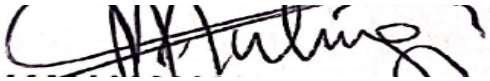
I KIPRUTO CALEB REG NO BU/UP/2023/1493 here by declare that this project report is mine and has never been published or submitted for any academic award in the institutions of higher learning.

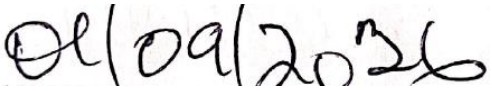
Signature  date 01/09/2026

KIPRUTO CALEB

APPROVAL

This is to acknowledge and approve that this project report is for KIPRUTO CALEB REG NO BU/UP/2023/1493. It was done under my supervision and was completed successfully.

Sign. 

Date. 

DR MUTUNGI FREDRICK

Department of computer studies**.**

Faculty of Science and Education

DEDICATION

First and most importantly, I thank the Almighty God who has enabled me to complete this report with good health.

I then dedicate this research report to my caring and loving parents **MR YEKO GEORGE** and **MRS CLAUDIA CHEPKWEMOI** as well as the entire family, on whose advise, guidance and prayers I relied upon to fulfill this academic milestone. Thank you so much for the support and may the Almighty reward you abundantly.

I also dedicate this report to my dear supervisor **DR Mutungi Fredrick** who guided me throughout the process of making this report successful. Thank you so much sir for imparting knowledge in me.

ACKNOWLEDGEMENT

First and foremost, I thank the Almighty God for giving me strength, wisdom and perseverance, which has enabled me to complete my report successfully.

Special thanks to my Supervisor **DR. Mutungi Fredrick** for his support and guidance to the completion of this research work.

I also extend my appreciation to my lecturers and the entire Faculty of Science and Education especially the department of Computer Studies for equipping me with the necessary knowledge and skills.

I am grateful to my family for their constant prayers, motivation, and emotional support during this academic journey.

I am also grateful to my friends and colleagues, **Kibet Levi**, **Chepkwemoi Damaris, Nyaburu Rose Mary** and **Ayeko Jeremiah** for your support, discussion sessions, and constructive critiques that enriched this project.

May the Almighty God reward all of them abundantly.

Table of Contents

DECLARATION. ii

APPROVAL iii

DEDICATION iv

ACKNOWLEDGEMENT v

List of Tables ix

List of figures x

ACRONYMS xii

ABSTRACT xiii

1.0 Introduction 1

1.1 Background of the study 1

1.2 Problem Statement 2

1.3 Main Objective of the study 3

1.4 Specific Objectives 3

1.5 Scope of the study 3

1.6 Significance of the study 4

CHAPTER TWO: LITERATURE REVIEW 5

2.1 Introduction 5

2.2 Review of Existing Smart Waste Management Systems 5

2.3 System Design 5

2.4 System Implementation 6

2.5 System Testing and Evaluation 6

2.6 Research Gap 7

3.1 Introduction 7

3.2 Type of Research and Development Approach 7

3.3 Area of Study 8

3.4 Population and Sampling methods 8

3.5 Data Collection Methods and Instruments 8

3.6 Data Analysis Methods 8

3.7 System Design. 9

3.8 System Implementation 9

3.9 System Testing and Evaluation 9

3.10 Ethical Considerations 9

4.0 Introduction 10

4.1 Analysis of the Existing System 11
4.1.1 Current System Workflow 11
4.1.2 Strengths of the Existing System 11
4.1.3 Weaknesses of the Existing System 12
4.2 Data Presentation and Analysis 13
4.2.1 Respondent Categories 13
4.2.2 Current Waste Management Methods 14
4.2.3 Required System Modules 15
4.2.4 Security and Access Requirements 16
4.2.5 Expected Improvements 17
4.3 Requirements Analysis 18
4.3.1 User Requirements 18
4.3.2 Functional Requirements 19
4.3.3 Non-Functional Requirements 19
4.3.4 System Requirements 19
4.4: System Design 20
4.4.1 Context flow diagram 20
4.4.2 Data flow diagram 21
4.4.3 Use Case Diagram 22
4.4.4 Entity Relationship (ER) Diagram 23
4.4.6 Database Design 25
4.5 Chapter Summary 27
5.0 Introduction 27
5.1 System Implementation 28
5.1.1 System Deployment 29
5.1.2 System User Interfaces 31
5.1.3 User Login Interface 32
5.1.4 User Access Pages 32
5.5 Responsive Design and Mobile Compatibility 61
5.6 System Performance and Load Testing 63
5.7 Chapter Summary 64
6.1 Introduction 65
6.2 Discussion of Results 65
6.3 Conclusion 68

6.4 Recommendations 68
6.5 Limitations 69
6.6 Future Work 69
REFERENCES 70

List of Tables

Table 1: showing respondent categories 13
Table 2: Current Waste management methods 14
Table 3: Required System Modules 15
Table 4: Showing Security and Access Requirements 16
Table 6: Showing Expected Improvements 17
Table 7: Showing Data Dictionary Showing Description of Processes and Data Stores 22
Table 8: Showing Data Dictionary for External Entities 22
Table 9: Showing Admin Table Schema 25
Table 10: Showing Collector Table Schema 25
Table 11: Showing Resident Table Schema 25
Table 12: Showing Waste Requests Table Schema 26
Table 13: Showing Feedback Table Schema 26
Table 14: Showing Comprehensive Test Cases and Results 55
Table 15: Showing Showing User Acceptance Testing and Feedback 57

List of figures

Figure 1: showing respondent categories....14
Figure 2: showing current waste management methods....15
Figure 3: Showing Required System Modules....16
Figure 4: Showing Security and Access Requirements....17
Figure 5: Showing Expected Improvements....18
Figure 6: Showing Context flow diagram....20
Figure 7: Showing Data flow diagram....21
Figure 8: Showing Use Case diagram....23
Figure 9: Showing Entity Relationship Diagram....24
Figure 10: Showing Smart Waste Management System Home Page....32
Figure 11: showing administrator login page....33
Figure 12: showing administrator dashboard....34
Figure 13: showing registered user's page....35
Figure 14: showing waste collection requests page....35
Figure 15: showing assign collector page....36
Figure 16: showing reports page....37
Figure 17: showing collector login....37
Figure 18: showing collector dashboard....38
Figure 19: showing assigned collection tasks page....39
Figure 20: showing update collection status page....39
Figure 21: showing assigned waste collection schedule page....40
Figure 22: showing report completed collection page....41
Figure 23: showing resident login page....41
Figure 24: showing resident registration page....42
Figure 25: showing resident dashboard....43
Figure 26: showing waste collection request page....44
Figure 27: showing resident request status page....45
Figure 28: showing Smart waste management database....45
Figure 29: showing waste requests table....46
Figure 30: showing collector feedback table....47
Figure 31: Showing Secure Parameterized Query Execution....48
Figure 32: Database Connection and Variable Setup....49
Figure 33: Showing Database Connection Initialization code....51

Figure 34: Showing Session and Authentication Check code 52
Figure 35: Showing Unit Testing..... 54
Figure 36: Showing Integration Testing 54
Figure 37: Showing authentication testing 58
Figure 38: Showing role-based access control testing..... 59
Figure 39: Showing data protection using prepared statements 60

ACRONYMS

SWMS	smart waste management system
DFD	data flow diagram
RAD	rapid application development
UML	unified modeling language
XAMP	cross platform, apache. Mysql.php and perl
WAMP	windows, apache,mysql and php
HTML	hyper text markup language
CSS	cascading style sheets
PHP	hypertext preprocessor
MYSQL	my structured query language
SQL	structured query language
VS CODE	visual studio code

ABSTRACT

The increasing demand for efficient waste collection and management highlighted the limitations of the traditional manual processes used by Ecopick Company, Mbale City. The existing methods relied on phone calls, WhatsApp messages, paper records, and spreadsheets to manage waste collection requests and assignments. These approaches resulted in delayed service delivery, poor communication between residents and waste collectors, loss or duplication of records, and difficulties in monitoring collection activities and generating accurate reports. To address these challenges, this study developed a Smart Waste Management System (SWMS) to provide a centralized, automated, and user-friendly platform for managing waste collection requests, collector assignments, collection records, and reporting.

The project employed the Rapid Application Development (RAD) methodology, which facilitated iterative development, continuous user involvement, rapid prototyping, and timely system refinement throughout the development process. System requirements were gathered through interviews, observation, and document analysis involving administrators, waste collectors, and residents to ensure that the developed system addressed the operational needs of Ecopick Company. The system was designed using structured system design techniques, including Unified Modelling Language (UML) diagrams, data flow diagrams, and an entity relationship diagram to model system processes, user interactions, and database relationships.

The implementation phase utilized HTML, CSS, and JavaScript for the front-end user interface, while PHP and MySQL were used for back-end development and database management. These technologies were selected because of their reliability, scalability, cost-effectiveness, and suitability for developing web-based applications. After development, the system was tested using unit testing, integration testing, user acceptance testing, and security testing with participation from administrators, waste collectors, and residents.

The results demonstrated that the Smart Waste Management System improved the efficiency of waste collection management by reducing manual paperwork, enhancing communication between residents and collectors, improving the accuracy of record management, and enabling real-time monitoring of waste collection requests and activities. The system also simplified report generation, strengthened accountability, and supported informed decision-making through centralized data management. The study concludes that the Smart Waste Management System provides an effective solution for improving waste collection operations and service delivery at Ecopick Company. Its therefore recommented that ecopik company adopts the

developed syetem to replace manual waste management processes and further enhance operational efficiency ,transperancy and environmental sustainability.

CHAPTER ONE: INTRODUCTION

1.0 Introduction

This chapter describes the introductory aspects of the study concerning the development of the Web-Based Waste Collection Management System. It covers the background of the study, outlining the current manual processes used for Waste management at Ecopick Company and the challenges associated with them. The chapter also states the problem that the study aims to solve, specifies the main and specific objectives guiding the system development, defines the geographical and content scope of the project, and highlights the significance of the proposed system in improving Waste Management.

1.1 Background of the study

Waste management is a major global challenge affecting both developed and developing countries. Rapid urbanization, population growth, and increased consumption have increased global waste generation, now exceeding 2.0 billion tonnes annually and projected to reach 3.9 billion tonnes by 2050, with 33% mismanaged **(Rifka Alkhilyatul Ma'rifat, I Made Suraharta, 2024**) . Even with advanced technologies like recycling, scheduled collection, and automated monitoring, many developed nations still struggle and often export waste to countries with weaker waste-handling capacity **(Ahmad et al., 2021)**.

The challenge is more severe in developing countries, where waste systems are underfunded, poorly coordinated, and mostly manual. About 60% of waste remains uncollected and 93% is improperly managed in such regions **(Zohoori & Ghani, 2017)** . Irregular pickups create waste accumulation and breeding grounds for mosquitoes, flies, and rodents. As noted **by (Krystosik et al., 2020)**, weak infrastructure, low public awareness, and poor enforcement worsen public health risks and environmental degradation.

In Uganda, waste management poses significant environmental and health concerns, especially in rapidly growing towns and cities. Many communities depend on manual, uncoordinated waste collection where residents dispose of waste without knowing pickup schedules. Waste trucks move unpredictably, causing delays and missed pickups **(Marshall & Farahbakhsh, 2013)**. Only 8% of Uganda's waste is recycled, while 92% is improperly disposed of, contributing to pollution, blocked drains, and sanitation-related illnesses. In slum areas, only 41.3% of households practice proper waste management **(Whitney G.G., 2008)**.

Addressing these challenges requires modernized systems that improve coordination among residents, collectors, and administrators. A Smart Waste Management System offers a cost-effective solution by digitizing communication and scheduling **(Principle, 2023).** It enables residents to request services, allows collectors to manage assignments, and helps administrators track operations and generate performance reports. Such a system enhances efficiency, accountability, and environmental health, supporting sustainable urban development **(Fatimah et al., 2020)**.

1.2 Problem Statement

Waste management in Mbale City, Uganda, is still largely manual and uncoordinated, leading to irregular waste collection, poor communication between residents, collectors, and accumulation of waste in communities. Studies in similar urban local governments show that over 60% of households experience irregular waste collection **s**ervices, while approximately 45% of waste generated remains uncollected on time, leading to illegal dumping and environmental pollution. This results in environmental pollution and increased health risks.

Residents are often unaware of collection schedules, while collectors operate without proper coordination, causing missed pickups and inefficient service delivery. Research indicates that more than 55% of residents do not receive timely collection updates**,** and about 30-35% of scheduled waste pickups are either delayed or missed entirely due to poor communication systems between stakeholders.

Administrators also lack digital tools to monitor, schedule, and track waste collection activities effectively, leading to poor accountability and delays. In many manual systems, there is an estimated 25–30% loss or inconsistency in operational records, making it difficult to properly track waste collection performance and enforce accountability.

Although existing studies highlight waste management challenges in urban areas, there is limited implementation of integrated digital systems tailored to local contexts such as Mbale **(Siwawa1 & Phuluwa2, 1970)**. This creates a gap in efficient communication and coordination within waste management processes.

Therefore, the proposed web application seeks to improve communication, streamline operations, and enhance the efficiency of waste collection services in Mbale City.

1.3 Main Objective of the study

The main objective of this study is to develop and evaluate a web application, the Smart Waste Management System for improving coordination between residents, collectors and administrators.

1.4 Specific Objectives

i. To conduct an in-depth review of the existing literature and determine the requirements for developing a web-based waste collection management system.
ii. To design a web-based waste management system based on the identified requirements
iii. To implement the designs of the web-based waste collection management system
iv. To test the functionality of a web-based waste management system

1.5 Scope of the study

The research focuses on the development and implementation of the Smart Waste Management System using PHP technology.

It covers the design, development, testing and development of a system that enables clients to report the wastes to be collected and allows the collectors to come and pick

Depth of investigation: The study examined the challenges associated with the traditional manual system, the benefits of digital transformation waste management tracking and the feasibility of implementing a centralized web-based platform for waste management.

The system was tested to evaluate its effectiveness.

Geographical Location: The study was conducted in Ecopick company along republic street mbale in mbale municipality mbale city.

Time frame: The research, system development and testing took a period of three months, covering requirement analysis, system design, development, testing and deployment.

1.6 Significance of the study

The system provides clients and waste collectors a reliable tool to monitor collections effectively, thereby improving waste collection activity tracking and overall well-being, the system eradicates the manual system way of collection which is full of drawbacks.

The system can be applied in water and sewage corporation management.

Waste management intends to carryout recycling of plastic waste materials and other wastes hence keeping the environment clean

Improved waste management efficiency, mitigating health risks and environmental pollution.

Cost-effective deployment, considering economic constraints and sustainability.

Enhanced community awareness and engagement in waste management practices.

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction

This chapter reviews existing literature related to smart waste management systems and the processes, technologies, and methodologies used in their development. The purpose of the literature review is to examine existing knowledge, identify gaps in current waste management practices, and provide a foundation for the design and implementation of the proposed Smart Waste Management System. The chapter is organized according to the study objectives and covers literature on requirements identification, system design, system implementation, and system testing. It further highlights the research gaps that justify the need for the proposed system.

2.2 Review of Existing Smart Waste Management Systems

Effective waste management requires an understanding of existing challenges, user needs, and operational workflows. Existing Literature shows that waste management in many developing countries remains inefficient due to inconsistent collection schedules, manual reporting methods, and limited communication between residents and collectors **(Guerrero et al., 2013).** Note that poor waste coordination contributes to environmental pollution and disease outbreaks. Other researchers emphasize that many cities lack digital systems capable of tracking waste generation, scheduling collections, and monitoring service delivery**(Li et al., 2011).**

In several African urban settings, the absence of modern data collection tools hinders municipalities from analyzing waste volumes, planning collection routes, and responding to service delays. Literature also highlights stakeholder challenges, including limited resident awareness, ineffective municipal communication, and lack of record-keeping systems **(Science, 2024)**. These studies underscore the need for digital, user-friendly platforms that can enhance coordination, streamline requests, and improve administrative monitoring.

2.3 System Design

Various studies have proposed digital waste management platforms to improve coordination and service delivery. Many systems adopt web-based or mobile-based interfaces to allow real-

time communication between stakeholders. Common design approaches include modular system architectures, use case modelling, entity-relationship diagrams (ERD's), and data flow diagrams (DFD's) to represent system processes and data interactions.

Scholars highlight the importance of designing systems with clear user roles-such as residents, collectors, and administrators-to ensure efficiency and control **(Simpson, 2021).** System architectures often follow a three-tier structure: the presentation layer for user interfaces, the application layer for business logic, and the database layer for data storage. Literature also highlights the need for responsive interfaces, intuitive navigation, and secure authentication mechanisms to ensure usability and safety **(Rajesh Prabu Vincent De Paul, 2025).**

2.4 System Implementation

System implementation in related studies typically involves the use of web technologies such as HTML, CSS, JavaScript, PHP, and MySQL due to their accessibility, flexibility, and cost-effectiveness. PHP is widely used in similar systems because it supports server-side processing and integrates seamlessly with MySQL databases. MySQL is commonly adopted for storing structured data, including user profiles, waste records, and system logs.

Researchers also emphasize the importance of implementing backend logic that supports real-time updates, task assignment, and status tracking. In modern smart waste systems, implementation may include features such as notification systems, automated scheduling, and digital dashboards. Many studies advocate for lightweight, scalable systems that can be deployed easily on local servers using platforms such as WAMP or XAMPP, especially in resource-constrained settings.

2.5 System Testing and Evaluation

System testing is widely recognized as an essential phase of software development. Literature identifies several testing approaches relevant to smart waste management systems, including functional testing to ensure the system performs intended tasks, usability testing to evaluate user experience, and security testing to verify data protection. Software evaluation frameworks often focus on accuracy, reliability, load performance, ease of use, and user satisfaction.

Studies show that end-user testing is crucial for systems involving diverse stakeholders such as residents, collectors, and administrators. User acceptance testing (UAT) helps ensure that the system meets real-world needs and aligns with operational workflows. Researchers also highlight the importance of iterative testing in improving performance and identifying bugs before deployment.

2.6 Research Gap

Although various studies have explored digital approaches to waste management, many existing systems focus broadly on waste monitoring and do not support real-time coordination between residents, collectors, and administrators **(Lee et al., 2015)**. In Uganda, and particularly in Mbale City, the adoption of practical and locally adaptable digital tools remains limited. Current systems often lack integrated communication features, automated reporting, and user-friendly platforms suitable for low-resource settings **(Stockdale, 2020)**. This study addresses these gaps by proposing a Smart Waste Management System designed to support service requests, task assignment, and real-time tracking within the local context.

CHAPTER THREE: METHODOLOGY

3.1 Introduction

This chapter presents the methodological approach used in the development of the Smart Waste Management System. It describes the research design, study area, population, sampling techniques, data collection methods, data analysis procedures, system design processes, implementation strategy, system testing, and ethical considerations. The methodology ensures that the system is designed, developed, and evaluated systematically and reliably.

3.2 Type of Research and Development Approach

The study adopts a system development research design, which focuses on identifying a problem, gathering requirements, designing a solution, and developing a working system. The Rapid Application Development (RAD) model was used in the development process. This was chosen because it supports quick prototyping, continuous user involvement, and iterative improvements, which are essential for creating a practical and user-friendly waste management

system. The model allows the system to evolve based on feedback from residents, collectors, and administrators during development.

3.3 Area of Study

The study was conducted in Ecopick company in Mbale City, an urban area facing significant waste collection and coordination challenges. The system targets key stakeholders involved in waste management.

3.4 Population and Sampling methods

Purposive sampling was used to select participants who have relevant experience and interact directly with waste management activities. Population and samples included 1 administrator, 4 selected waste collectors, and 5 residents familiar with the challenges of requesting waste collection services. The technique was suitable because it ensured that the study captured insights from those most knowledgeable about the existing waste management workflows.

3.5 Data Collection Methods and Instruments

Data was collected using interviews, questionnaires, and observations. Interviews were conducted with municipal administrators to understand the structural and operational challenges. Questionnaires were distributed to residents and waste collectors to gather information on their experiences, expectations, and system requirements. Direct observation of waste collection processes helped identify inefficiencies and validate information obtained from respondents. Document analysis was also used to review municipal waste management reports and policies.

3.6 Data Analysis Methods

The data collected from interviews and observations was analyzed using qualitative methods, specifically thematic analysis, to identify common patterns and challenges in waste management. Questionnaire data was analyzed using simple descriptive statistics such as percentages and frequency counts to summarize user needs and system requirements. These analyses informed the design specifications and functional components of the Smart Waste Management System.

3.7 System Design.

The system design phase translated the identified requirements into a structured blueprint of the Smart Waste Management System.

3.8 System Implementation

The implementation phase involved developing the system using web-based technologies. HTML, CSS, and JavaScript will be used to create responsive and user-friendly interfaces for residents, collectors, and administrators. PHP was used for backend logic to process requests, manage user authentication, and handle system operations. MySQL served as the database management system for storing user information, collection records, and system logs. The system will be hosted on a local server environment using WAMP/XAMPP during development to support testing and debugging.

3.9 System Testing and Evaluation

System testing was conducted to ensure that the Smart Waste Management System functions as intended.

Unit testing was performed on individual modules such as login functions, request submission, and task assignment.

Integration testing was followed to verify that modules work correctly when combined.

User Acceptance Testing (UAT) involves residents, collectors, and administrators evaluating the system to determine whether it met their needs and expectations. Feedback from testing was used to refine and improve the system's performance, usability, and reliability.

3.10 Ethical Considerations

The ethical issues that were considered during data collection and analysis include the following.

Respect of intellectual property:

I never copied or plagiarize other people`s work during my research instead I considered text citation and referencing to acknowledge the source of information.

Integrity: I had to be consistent in my actions and words during the research process and kept my promises and agreements with all the users of waste management system

Honesty:

I honestly presented data results, procedures, publications and reasons for collecting data for waste management system.

Confidentiality:

I protected the sensitive information that was provided to me by the respondents to prevent unauthorized access.

Objectivity:

I had no bias in all aspects especially natural bias in reporting data and ensured proper sampling of the respondents during data collection.

CHAPTER FOUR: DATA ANALYSIS, PRESENTATION AND DESIGN

4.0 Introduction

This chapter presents, analyses, and discusses the findings of the study on the development of a Smart Waste Management System for Ecopick Company Mbale. The chapter focuses on the analysis of the existing manual system, presentation of collected data, interpretation of findings, and identification of system requirements for the proposed system.

Data was collected from administrators (1), waste collectors (3), Community leaders (1) and residents (5) through interviews and observation. The findings helped identify challenges in the current waste management processes, including delayed waste collection, poor communication, inefficient record management, lack of real-time monitoring, and difficulties in report generation. The chapter also presents stakeholder expectations and translates the findings into functional and non-functional requirements for the proposed system.

4.1 Analysis of the Existing System

4.1.1 Current System Workflow

At Ecopick Company Mbale, waste collection operations are managed using manual methods such as phone calls, WhatsApp messages, paper records, and Microsoft Excel spreadsheets. Residents submit waste collection requests by calling the office, sending WhatsApp messages, or visiting the company offices physically.

Administrators manually record customer information and collection requests before assigning waste collectors through phone calls or handwritten schedules. After completing collection activities, collectors report back to administrators using phone calls or WhatsApp messages.

Although the system enables the company to provide waste collection services, the investigation revealed several limitations. Waste collection requests are difficult to track, communication delays occur frequently, report generation is time-consuming, and there is no real-time monitoring of collection activities. As the number of customers increases, the current manual system becomes increasingly difficult to manage efficiently.

4.1.2 Strengths of the Existing System

Despite its limitations, the current system possesses several strengths including:

Simplicity

The system is easy to understand and operate by administrators and waste collectors with basic computer knowledge.

Low Training Requirements

The current system requires minimal technical skills since most activities are performed manually.

Existing Communication Channels

Residents can easily communicate with Ecopick through phone calls and WhatsApp messages.

Offline Accessibility

The system can operate without internet access because most records are stored manually.

Environmental Conservation

The company contributes to environmental cleanliness through regular waste collection and disposal.

Public Health Protection

Proper waste collection reduces environmental pollution and prevents disease outbreaks.

Low Operational Costs

The system requires limited investment in advanced technology and infrastructure.

Community Participation

Residents actively participate by reporting waste collection needs and providing feedback on services.

4.1.3 Weaknesses of the Existing System

The current system exhibits several weaknesses that affect operational efficiency.

Slow Information Retrieval

Customer information and collection records take time to retrieve because they are stored in different files and spreadsheets.

Data Duplication

The same information is often recorded in multiple places, leading to inconsistencies.

Risk of Data Loss

Paper records can be misplaced, damaged, or lost.

Delayed Reporting

Administrators spend significant time compiling reports manually.

Poor Communication

Communication gaps between administrators, collectors, and residents frequently occur.

Lack of Real-Time Monitoring

Administrators cannot track collection activities as they happen.

Weak Security Controls

Manual records and spreadsheets provide limited protection against unauthorized access.

Missed Collection Requests

Some requests are overlooked due to poor record management.

Poor Scalability

The current system struggles to handle increasing numbers of customers and collection requests.

4.2 Data Presentation and Analysis

This section presents and analyses findings obtained from administrators, waste collectors, Community leaders and residents regarding the current waste management system, challenges faced, and expectations for the proposed system.

4.2.1 Respondent Categories

Table 4.1 presents the categories and frequency of stakeholders who participated in the study.

Table 4.1: Respondent Categories

Table 1: showing respondent categories

Respondent Category	Frequency
Administrators	1
Waste Collectors	3
Residents	5
Community leaders	1

The findings indicated that all stakeholder groups participated in the study, with waste collectors providing the highest number of responses.

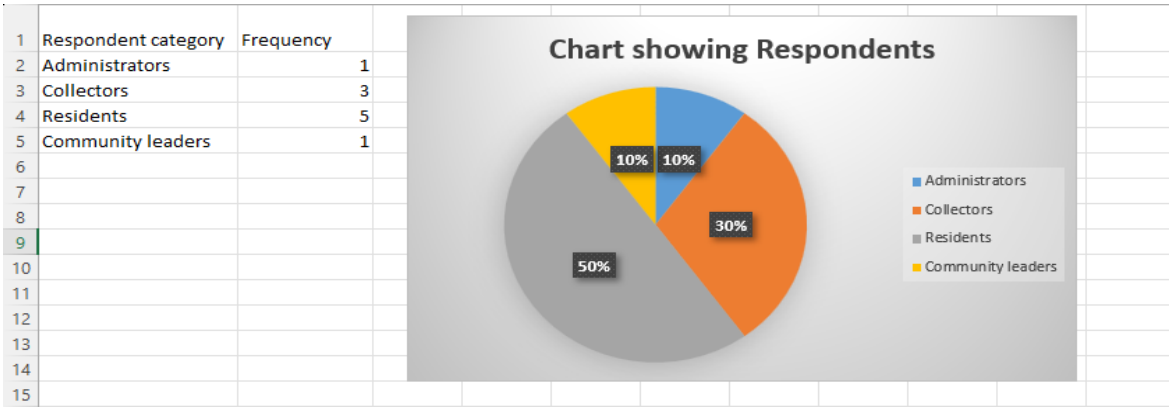


Figure 1: showing respondent categories

Analysis: The high participation of waste collectors and residents ensured that the views of the primary users of the system were adequately represented. This feedback provides a reliable basis for designing a system that addresses actual operational challenges.

4.2.2 Current Waste Management Methods

Table 4.2 illustrates the methods currently used by Ecopick Company for waste management operations.

Current Waste Management Methods

Table 2: Current Waste management methods

Method	**Frequency**
Phone Calls	5
WhatsApp Messages	4
Paper Records	3
Excel Spreadsheets	3

The results revealed that Ecopick Company mainly relies on manual communication and record management methods.

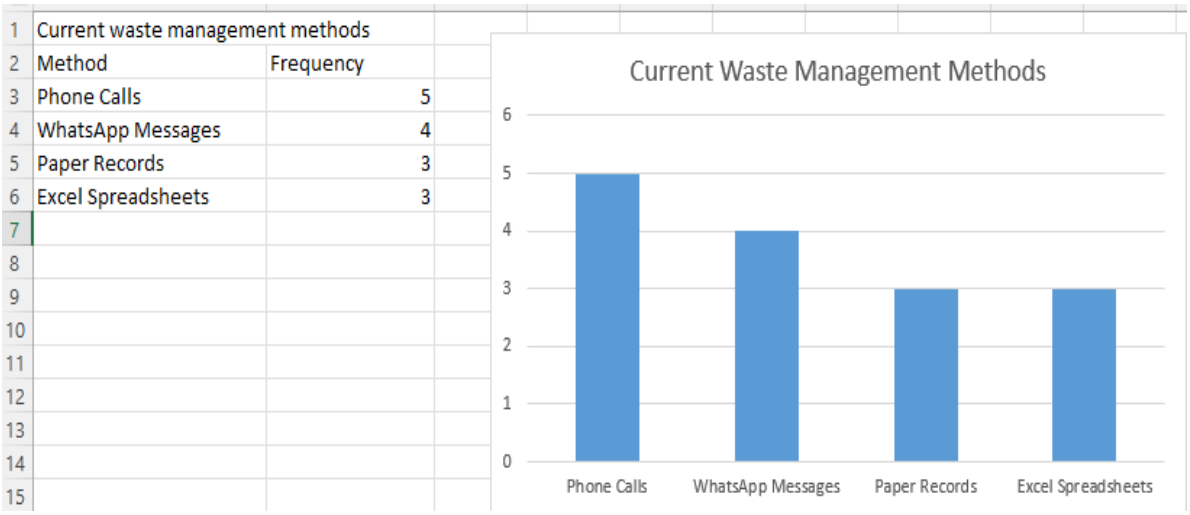


Figure 2: showing current waste management methods

Analysis: The dependence on phone calls, WhatsApp, paper files, and spreadsheets creates fragmented information storage, making monitoring and management difficult. This justifies the need for a centralized computerized system.

4.2.3 Required System Modules

Table 4.3 outlines the modules identified as essential by stakeholders.

Table 3: Required System Modules

Module	**Frequency**
Waste Collection Requests	5
Task Assignment	5
Real-Time Tracking	4
Reporting	4
Notifications	4
Feedback Management	3

The findings highlighted the need for a centralized system integrating all core waste management activities.

	Required module	Frequency
2	Resident registration	5
3	Request collections	4
4	Collector task assignment	3
5	Real-time monitoring	5
6	Reporting	4
7	Notification	4

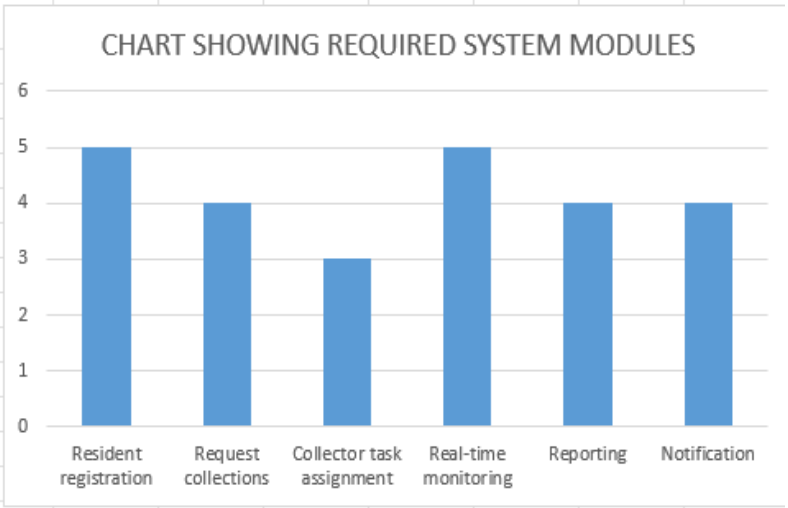


Figure 3: Showing Required System Modules

Analysis: The high demand for request management, task assignment, and tracking modules demonstrates the need for automation of daily waste collection processes.

4.2.4 Security and Access Requirements

Table 4.4 displays the security features prioritized by respondents.

Table 4: Showing Security and Access Requirements

Requirement	**Frequency**
Role-Based Access Control	4
Password Protection	4
Restricted Access	3

Security was considered essential for protecting customer and operational information.

	Requirement	Frequency
2	Role-based access control	4
3	Password protection	4
4	Restricted user access	3

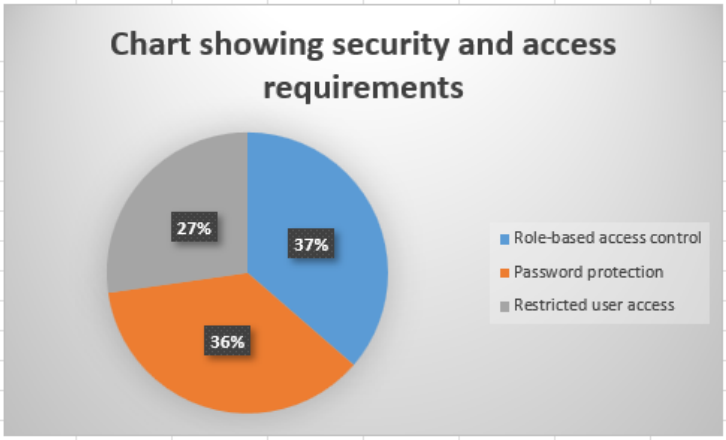


Figure 4: Showing Security and Access Requirements

Analysis: Stakeholders emphasized the need for user authentication and role-based access to ensure confidentiality and prevent unauthorized access.

4.2.5 Expected Improvements

Table 4.5 details operational improvements expected from the proposed system.

Table 5: Showing Expected Improvements

Improvement	Frequency
Improved Communication	5
Reduced Delays	5
Faster Reporting	4
Improved Efficiency	4
Real-Time Monitoring	4
Better Service Delivery	3

Respondents expected improved efficiency, communication, and transparency through the proposed system.

	Expected improvements	Frequency
2	Improved coordination	6
3	Faster request processing	4
4	Reduced paper work	3
5	Faster reporting	4
6	Real time monitoring	3

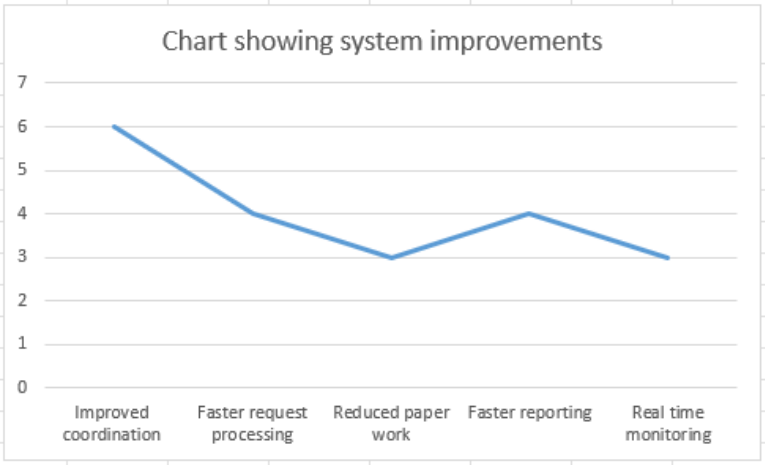


Figure 5: Showing Expected Improvements

Analysis: The high expectation for improved communication and reduced delays indicates the importance of implementing notifications, tracking, and automated scheduling.

4.3 Requirements Analysis

To understand the actual needs of users at Ecopick Company Mbale, a requirements analysis was conducted. The purpose was to identify specific features required for the proposed Smart Waste Management System.

4.3.1 User Requirements

The following requirements were identified:

i. A system that is easy to learn and use.

ii. A system that allows residents to submit collection requests online.

iii. A system that provides real-time updates on collection activities.

iv. A system that supports secure login and user authentication.

v. A system that provides attractive and user-friendly interfaces.

vi. A system that validates information during data entry.

vii. A system that securely stores resident and collection records.

viii. A system that generates reports automatically.

ix. A system that improves communication between administrators, collectors, and residents.

x. A system that supports monitoring of waste collection activities.

4.3.2 Functional Requirements

The functional requirements describe the specific services and operations that the Smart Waste management system is expected to perform. It should Support user authentication, allow residents and collectors to Register. Manage waste collection requests, assign collection tasks, Track collection activities, generate reports, manage feedback, send notifications and Search, update, and delete records.

4.3.3 Non-Functional Requirements

The non-functional requirements define the quality and performance aspects of the Smart Waste Management system. The system should be Fast and responsive, Reliable and available when needed. It should be Secure through role-based access control, User-friendly, Scalable for future expansion, Maintainable and easy to update.

4.3.4 System Requirements

Hardware Requirements

- Computer or Laptop (RAM 4GB or higher)
- Hard Disk (Minimum 80GB)
- Stable Power Supply
- Internet-enabled Device

Software Requirements

- XAMPP Server
- Windows 10/11
- Google Chrome or Mozilla Firefox
- MySQL Database
- PHP Programming Language

4.4: System Design

The requirements determined were used to design the system. The design development methods in the study are rapid application development (RAD). Design stages include; context diagram, data flow diagram, system modelling using UML (use case) and entity relationship diagram (ERD)

4.4.1 Context flow diagram

The context flow diagram is a visual representation that illustrate the interactions between the system being developed and external entities such as users, other systems, or organizations. It helped in understanding and analyzing how data flow into and out of the Smart Waste Management System, thereby providing a clear overview of the system's boundaries and its interactions with external actors.

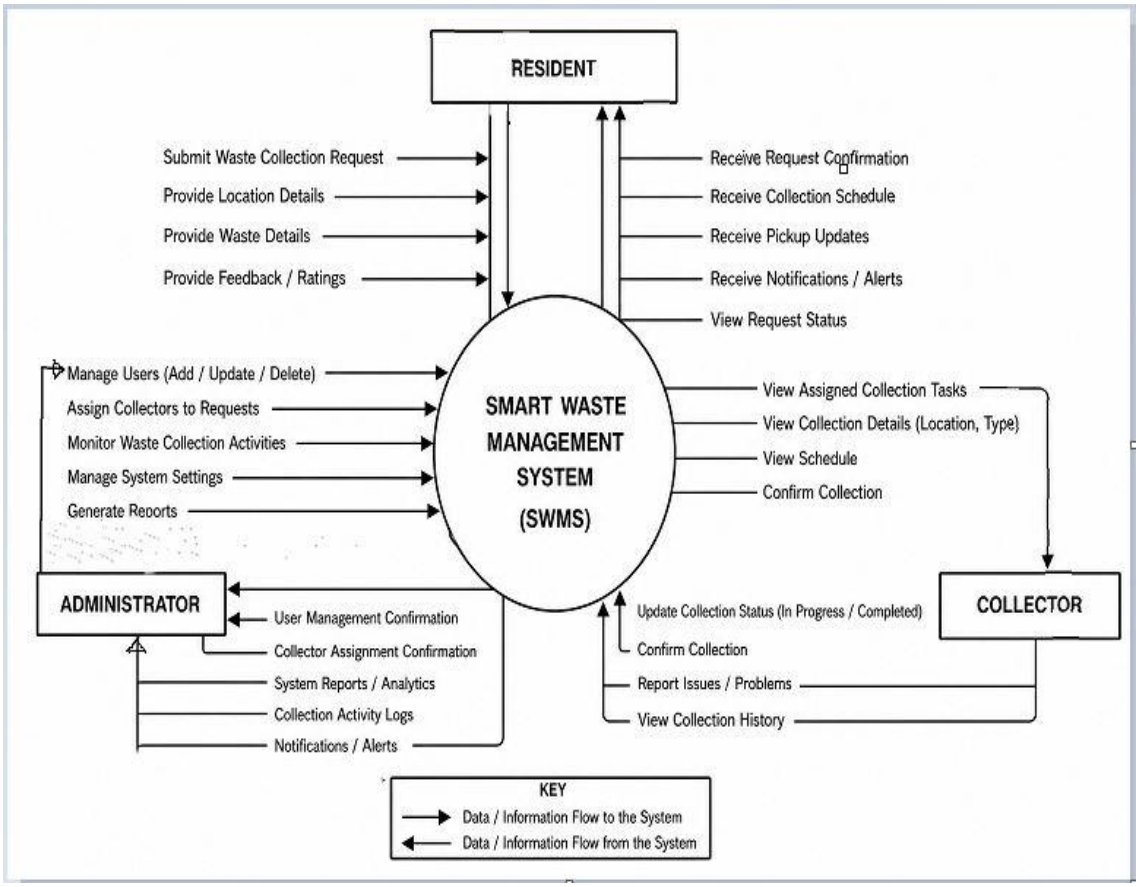


Figure 6: Showing Context flow diagram

4.4.2 Data flow diagram

A data flow diagram is a graphical representation of how data flow through the Smart Waste Management System. It was used to visualize data processing activities, showing how data move between processes, data stores, and external entities. The DFD is particularly useful in understanding system processes, identifying data sources and destinations, and defining the transformations that data undergo within the system, as illustrated in the diagrams.

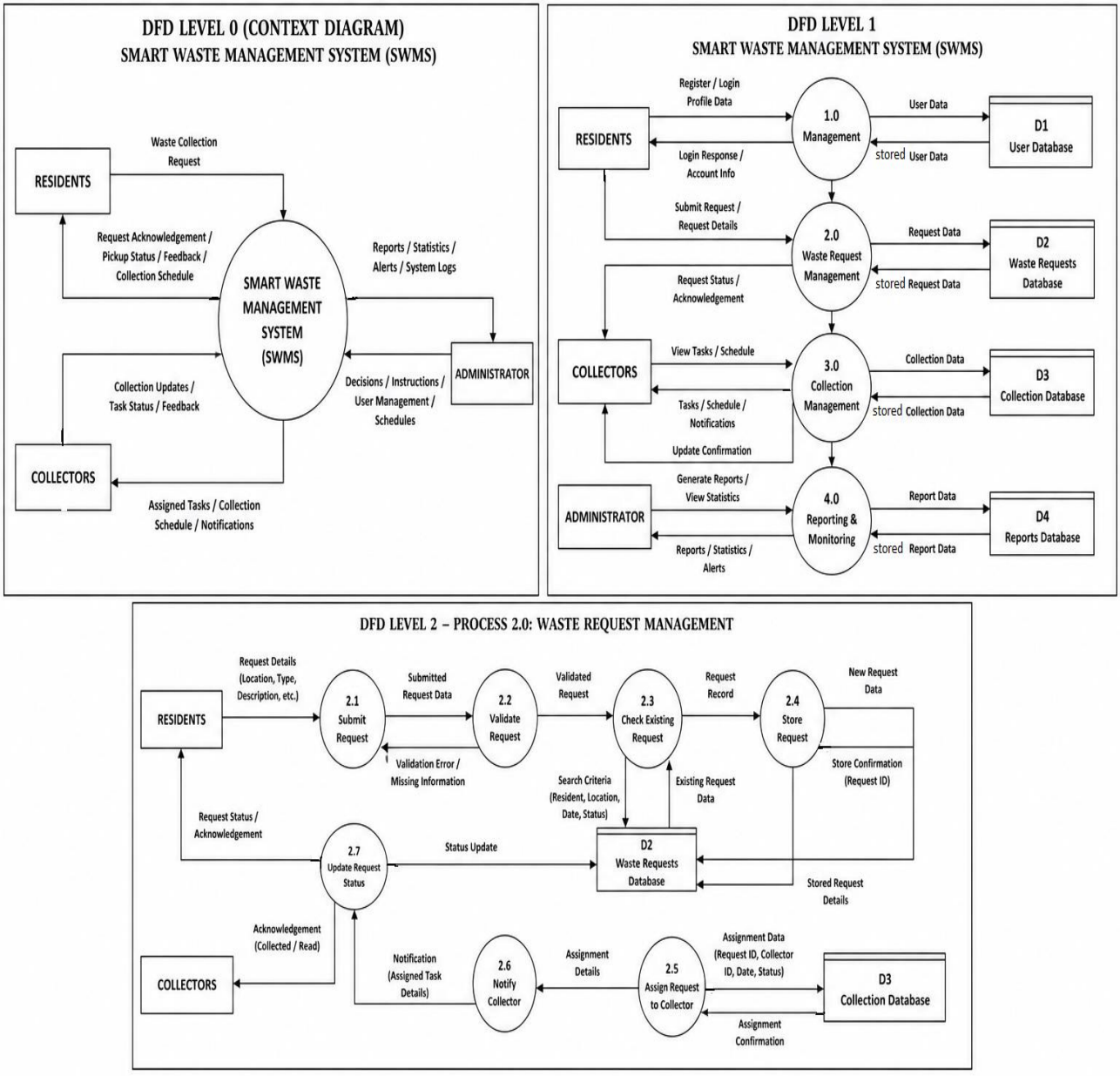


Figure 7: Showing Data flow diagram

Table 6: Showing Data Dictionary Showing Description of Processes and Data Stores

Process No.	Process Name	Description	Inputs	Outputs
1.0	Manage Requests	Handles waste collection requests from residents	Request Details	Stored Requests
2.0	Manage Collectors	Assigns and manages collectors	Collector Details	Task Assignments
3.0	Manage Collections	Updates collection status	Collection Updates	Collection Records
4.0	Generate Reports	Produces operational reports	Collection Data	Reports

Table 7: Showing Data Dictionary for External Entities

System User	Description	Requirements
Admin	Manages the entire system	Login, manage users, assign tasks, generate reports
Collector	Performs waste collection activities	Login, view tasks, update collection status
Resident	Requests waste collection services	Login, submit requests, view status, provide feedback

4.4.3 Use Case Diagram

A use case diagram is a behavioral diagram in Unified Modeling Language (UML) that represent the functionality of the system from the users' perspective. It illustrates how different users (referred to as actors) interacts with the system to accomplish specific tasks or achieve

particular goals. The diagram helped define the system's expected behavior and clarify user requirements.

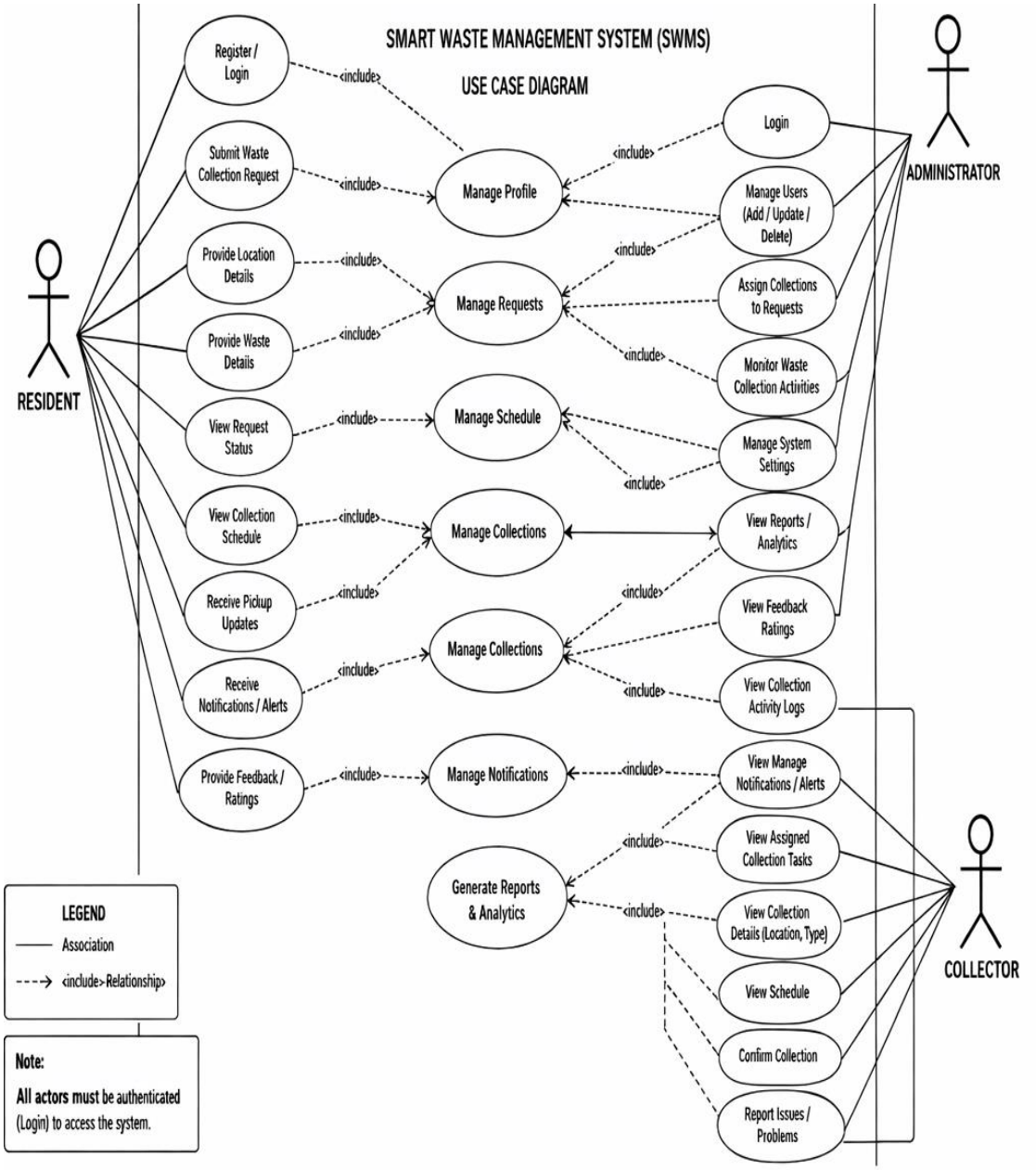


Figure 8: Showing Use Case diagram

4.4.4 Entity Relationship (ER) Diagram

An Entity Relationship Diagram (ERD) is a type of flowchart that illustrate how entities-such as people, objects, or concepts-will relate to each other within the proposed system. The ERD

helped define the structure of the database by showing the entities, their attributes, and the relationships that exist among them in the Smart Waste Management System.

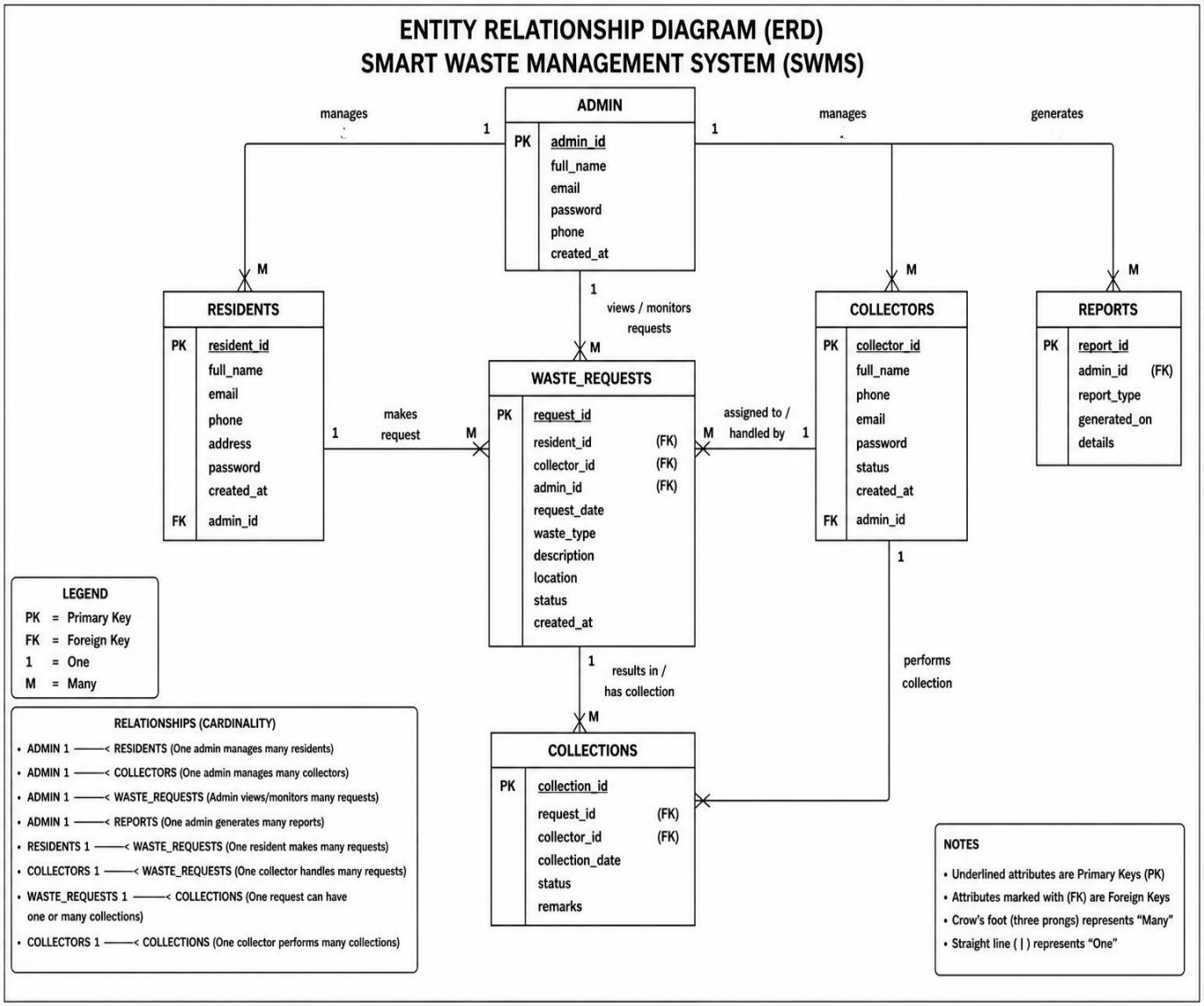


Figure 9: Showing Entity Relationship Diagram

4.4.6 Database Design

The database consists of several tables that store system information.

Table 8: Showing Admin Table Schema

Field Name	**Data Type**
id	int (11)
Username	varchar (50)
password	varchar (250)

Table 9: Showing Collector Table Schema

Field Name	**Data Type**
id	int (11)
full name	varchar (100)
phone	varchar (20)
username	varchar (50)
password	varchar (200)
created at	timestamp

Table 10: Showing Resident Table Schema

Field Name	**Data Type**
id	int (11)
full_name	varchar (100)
phone	varchar (20)

location	varchar (225)
password	varchar (100)
created_at	timestamp

Table 11: Showing Waste Requests Table Schema

Field Name	**Data Type**
id	int (11)
resident_id	int (11)
waste_type	varchar (100)
pickup_date	date
status	varchar (50)
created_at	timestamp

Table 12: Showing Feedback Table Schema

Field Name	**Data Type**
id	int (11)
resident_id	int (11)
message	text
feedback date	timestamp

4.5 Chapter Summary

This chapter presented and analyzed the data collected from the field, which informed the design of the Smart Waste Management System for Ecopick Company Mbale. The investigation revealed significant challenges including delayed waste collection, poor communication, manual record management, lack of real-time monitoring, and inefficient reporting. These findings justified the need for a computerized Smart Waste Management System.

Based on stakeholder feedback, user, functional, and non-functional requirements were established to ensure usability, security, efficiency, and scalability. These requirements were translated into a database structure consisting of five main tables for managing administrators, residents, collectors, waste requests, and feedback records. The next chapter presents system implementation, testing, conclusions, and recommendations.

CHAPTER FIVE: SYSTEM IMPLEMENTATION AND TESTING

5.0 Introduction

This chapter presents the implementation and testing of the Smart Waste Management System (SWMS) developed for Ecopick Company, Mbale District. It explains how the system was developed from the identified user requirements, system analysis, and design specifications into a functional web-based application.

The chapter describes the technologies, development environment, database implementation, system deployment procedures, and testing methods used to ensure that the developed system operates correctly and satisfies the requirements of the major users including administrators, residents, and waste collectors.

The Smart Waste Management System was developed to address challenges associated with the existing manual waste collection processes such as communication through phone calls and WhatsApp, paper-based records, poor tracking of waste collection activities, delays in service delivery, and difficulties in generating accurate reports.

The implemented system provides a centralized platform where residents can submit waste collection requests, administrators can manage users and assign collection tasks, and collectors can view assigned tasks and update collection status. The system improves communication, monitoring, record keeping, and efficiency in waste management operations.

5.1 System Implementation

The Smart Waste Management System was implemented in a Windows development environment using web-based technologies. Visual Studio Code was used as the Integrated Development Environment (IDE) for writing, debugging, and maintaining the system source code.

The frontend user interfaces were developed using HTML, CSS, and JavaScript to provide interactive and user-friendly system pages. PHP was used for backend processing, handling user authentication, business logic, and communication between the application and the database.

MySQL, managed through phpMyAdmin, was used as the database management system for storing and managing system information including user accounts, waste collection requests, collector assignments, collection records, notifications, and generated reports.

The system was deployed on a local server environment using XAMPP/WAMP, which provided the Apache web server, PHP runtime environment, and MySQL database services. Users were able to access the system through a web browser using a localhost address during the development and testing phase.

The Smart Waste Management System consists of three main user modules:

Administrator Module

The administrator module provides system management functions. The administrator is responsible for controlling system operations including:

- Managing resident and collector accounts.
- Viewing submitted waste collection requests.
- Assigning collectors to waste collection tasks.
- Monitoring collection activities.
- Managing waste records.
- Generating waste collection reports.
- Viewing system notifications and updates.

Resident Module

The resident module allows residents to interact with the waste collection system. Residents can:

- Create an account and log into the system.
- Submit waste collection requests.
- Specify waste type, description, and collection location.
- View request status.
- Receive updates about collection schedules.
- View completed collection history.

Collector Module

The collector module supports waste collectors in managing assigned collection activities. Collectors can:

- Log into the system.
- View assigned waste collection tasks.
- Access resident collection details.
- Confirm completed waste collection.
- Update collection status.
- View collection history.

5.1.1 System Deployment

System deployment refers to the process of installing, configuring, and making the final application available for use within its intended environment. The Smart Waste Management System was deployed using a web-based client-server architecture where users access the application through a web browser while data is stored and processed through the server and database.

The deployment process involved configuring the server environment, transferring system files, creating the database, importing database tables, updating system configurations, and performing final verification tests.

The following steps were followed during system deployment.

Server Requirements

Server Requirements: The system ran on a standard WAMP environment. The hosting server was required to meet the following minimum software specifications:

• Operating System: Linux (Ubuntu 22.04 LTS or similar)

• Web Server: Apache Web Server 2.4+

• Database Management System: MySQL 8.0+ or Maria DB 10.4+

• Scripting Language Environment: PHP 8.1+ with extensions for MySQL, PDO, and session.

Step-by-Step Deployment Process

Database Migration

• **Exporting Schema:** The local database (**waste_db**) was exported into a single file named **database.sql** using the phpMyAdmin export tool. The exported database contained all necessary tables required for the Smart Waste Management System, including users, waste requests, collections, and other related system records.

• **Creating Server Database:** A clean, empty database was created on the live hosting server's control panel (cPanel) under the same name (**waste_db**) to maintain compatibility with the system configuration.

• **Importing Tables:** The **database.sql** file was uploaded and executed on the live server to recreate all the necessary tables such as users, waste_requests, collections, and other required tables for the Smart Waste Management System.

File Transfer and Directory Setup

• **Compressing Code:** All Smart Waste Management System project files, scripts, and code directories were compressed into a single **.zip** file to simplify the transfer process.

• **Uploading Files:** The files were transferred to the server's public web directory (**public_html**) using an FTP client such as FileZilla or the hosting account's file manager interface.

• **Extracting Code:** The compressed project folder was extracted directly inside the web directory to make the Smart Waste Management System accessible through a web browser.

• **Folder Permissions:** The file permissions for the **uploads/** folder were updated to allow the system to store uploaded waste-related images and other required files on the server.

Configuration Update

• **Updating Credentials:** The main database connection file was edited directly on the live server to establish a proper connection between the Smart Waste Management System application and the database.

• **Live Server Details:** The local login credentials (**localhost, root, and an empty password**) were replaced with the live hosting provider's production details, secure database username, and strong generated password.

Live Testing and Verification

• **URL Verification:** The system administrator accessed the Smart Waste Management System through the configured system web domain to confirm that the application was available and accessible.

• **Functional Checks:** System functions such as user registration, login, waste collection request submission, collector assignment, and collection status updates were tested to confirm that the server paths and database connections were fully operational and secure.

5.1.2 System User Interfaces

Home Page

The home page is the first interface displayed when the Smart Waste Management System is accessed. It provides navigation options for different users including **Administrator,**

Resident, and Collector. Each user selects their respective login option, which directs them to the appropriate login page.

This page serves as the main entry point to the system and improves usability by clearly separating user roles. It allows users to easily access the system based on their responsibilities in the waste management process.

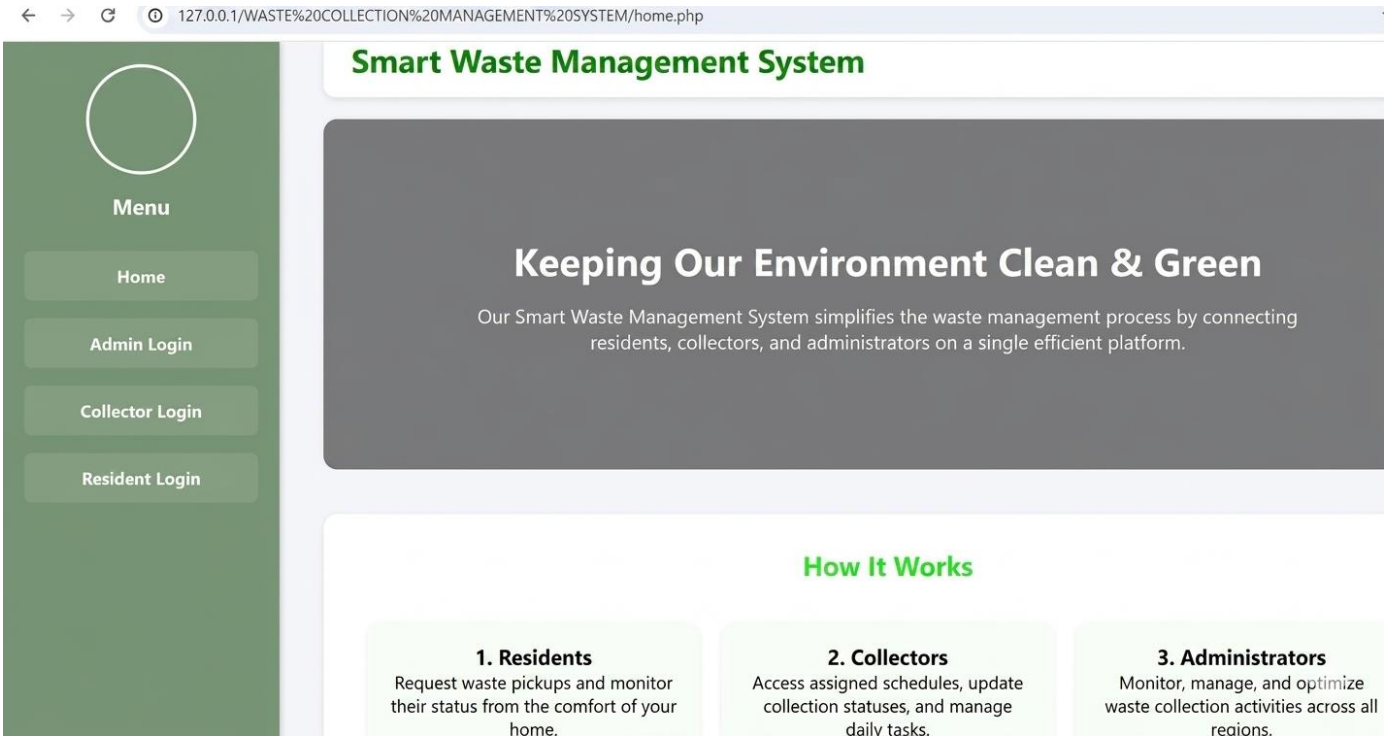


Figure 10: Showing Smart Waste Management System Home Page

5.1.3 User Login Interface

The login page allowed users to enter their registered email/username and password for authentication. Only users with valid login credentials were allowed to access the Smart Waste Management System. After successful authentication, users were redirected to their respective dashboards according to their assigned roles, which included Administrator, Resident, and Collector.

5.1.4 User Access Pages

This section showed all the interfaces contained in the Smart Waste Management System for different categories of users. Each user accessed specific interfaces according to their assigned roles to ensure proper security, authorization, and effective system management.

Administrator Login

The administrator login interface allowed the system administrator to securely access the administrative section of the Smart Waste Management System. After successful authentication, the administrator was redirected to the admin dashboard where system management activities were performed.

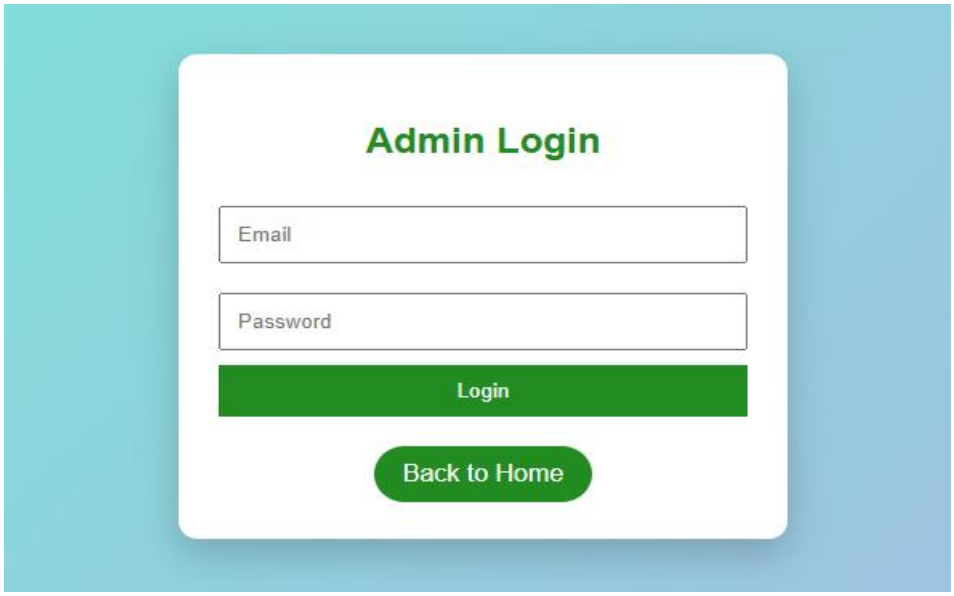


Figure 11: showing administrator login page

Administrator Side

The administrator had full control over the Smart Waste Management System and was responsible for managing the overall waste collection operations. The administrator was able to manage registered residents and collectors, view submitted waste collection requests, assign collectors to collection tasks, monitor waste collection activities, manage system records, and generate waste management reports.

The administrator was also able to monitor the overall performance of the system, update system information, and ensure that waste collection activities were properly coordinated between residents and collectors.

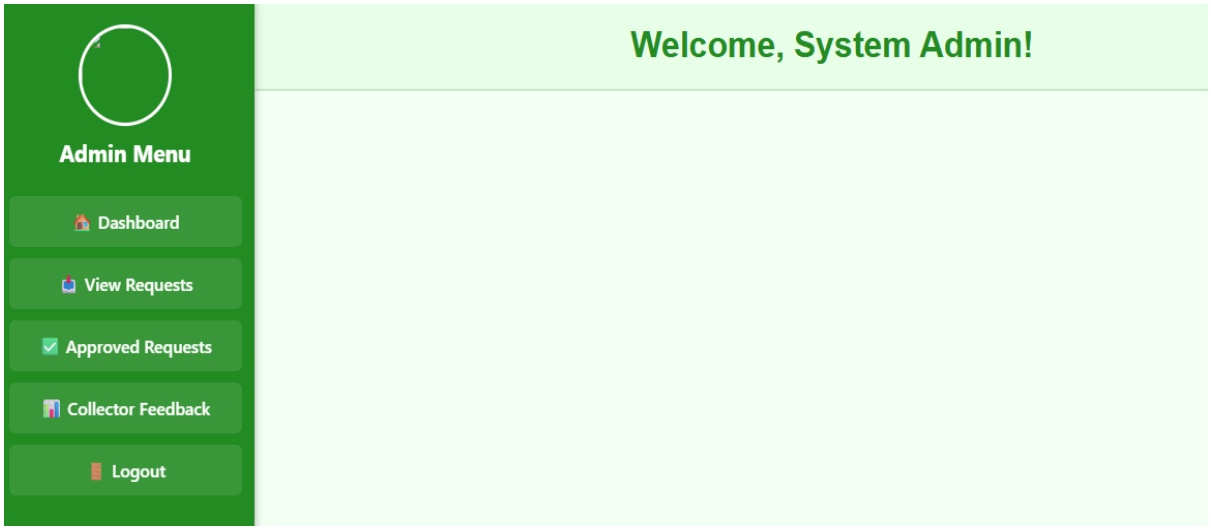
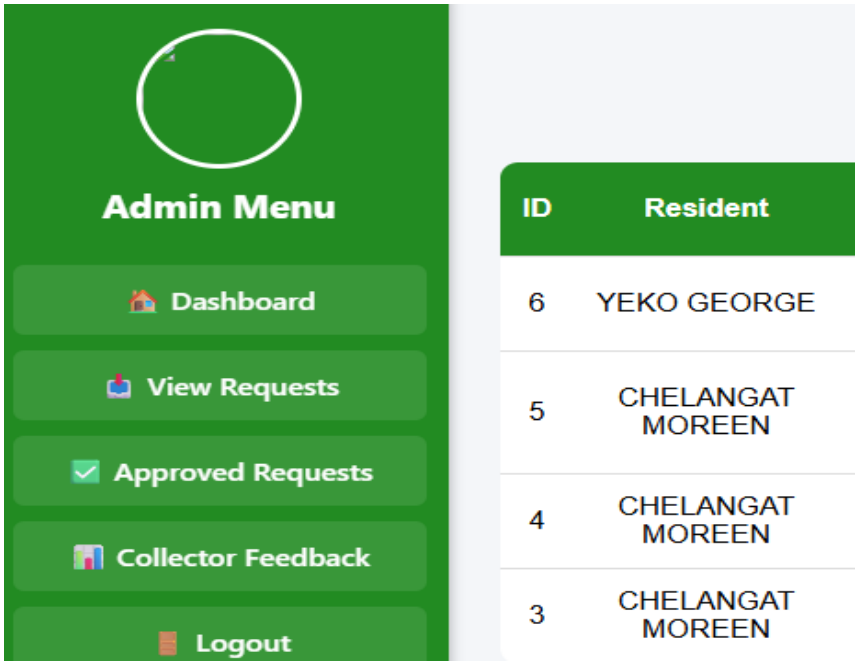


Figure 12: showing administrator dashboard

Administrator Pages

Registered Users Page

This interface allowed the administrator to view and manage all registered users within the Smart Waste Management System. The administrator was able to view registered residents and collectors together with their details. This helped in maintaining accurate user records and ensuring that only authorized users accessed the system.



Collector Name
SANDE JOSHUA
KWEM EMMA

Figure 13: showing registered user's page

Waste Collection Requests Page

This interface allowed the administrator to view all waste collection requests submitted by residents. The administrator was able to monitor request details such as waste type, description, location, request date, and current request status.

The page helped the administrator to track incoming waste collection requests and take necessary actions such as assigning collectors for waste pickup.

Figure 14: showing waste collection requests page

Assign Collector Page: This interface allowed the administrator to assign available waste collectors to submitted waste collection requests. After reviewing resident requests, the administrator selected a suitable collector and updated the collection assignment details.

This helped improve coordination between residents, collectors, and the company by ensuring that waste collection tasks were properly distributed.

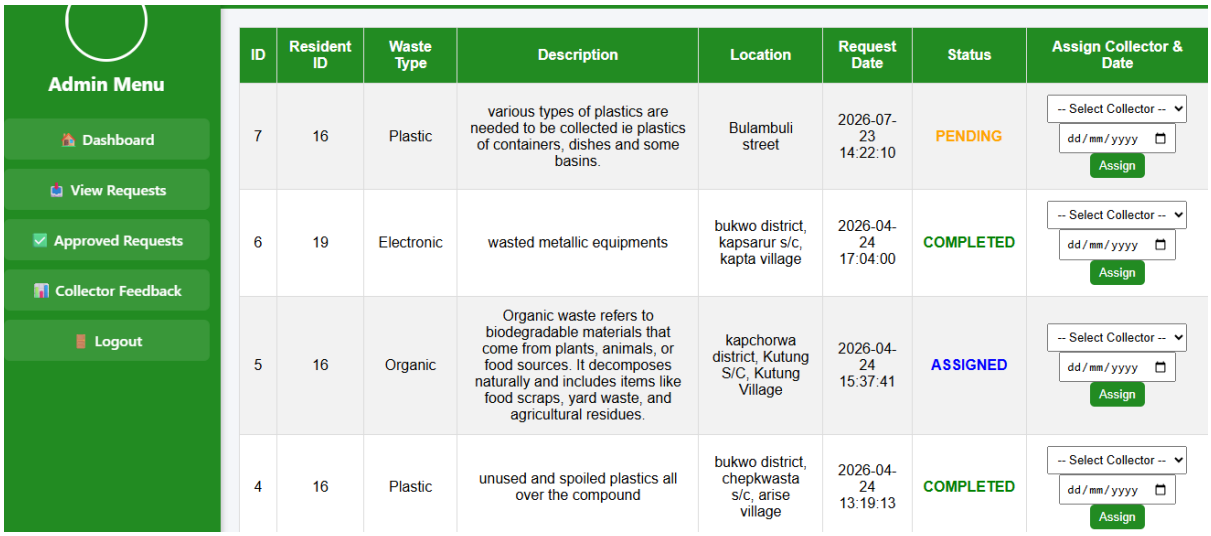


Figure 15: showing assign collector page

Reports Page

The reports interface allowed the administrator to generate and view waste management reports from the system. The administrator was able to monitor collection activities, completed requests, pending requests, and overall system performance.

The generated reports supported decision-making and helped Ecopick Company to evaluate waste collection operations effectively.

Figure 16: showing reports page

Collector Side

Collector Login

This interface allowed waste collectors to securely log into the Smart Waste Management System using their registered email/username and password. After successful authentication, collectors were redirected to their respective dashboard where they could access assigned waste collection activities.

The login interface ensured that only authorized collectors accessed collection tasks assigned to them by the administrator.

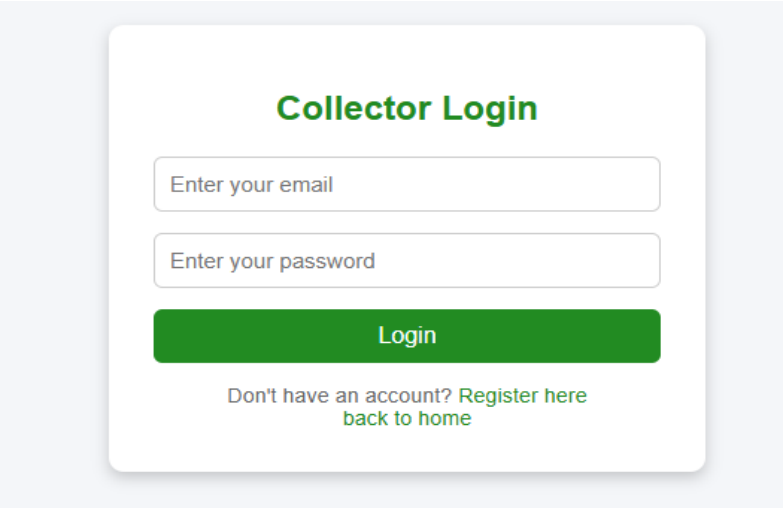


Figure 17: showing collector login

Collector Dashboard

The collector dashboard provided waste collectors with access to their assigned waste collection activities. Through this interface, collectors were able to view assigned waste collection requests, check resident collection details, monitor pickup schedules, and update the status of completed waste collection activities.

The dashboard helped collectors to organize their daily collection tasks and improved coordination between the administrator and waste collection teams.

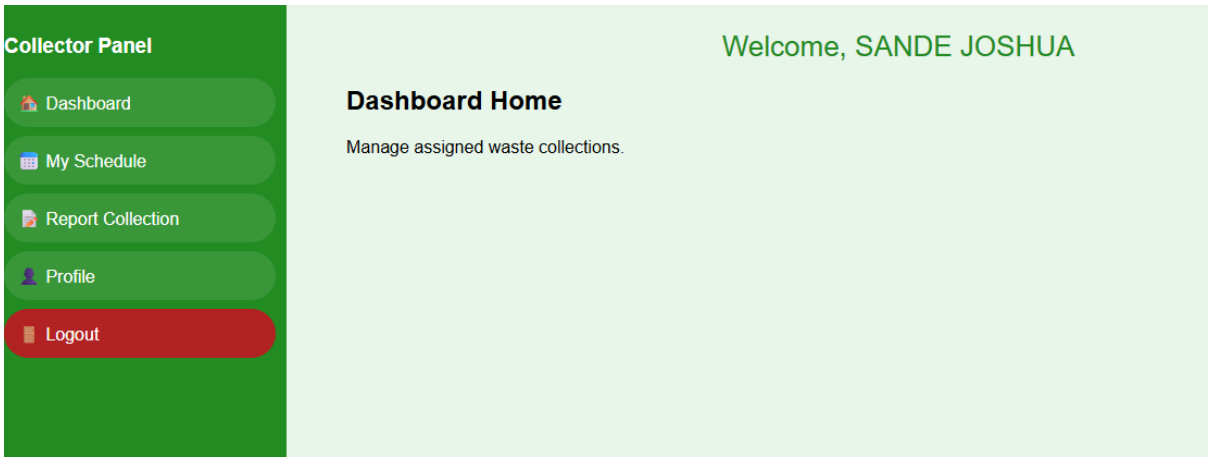


Figure 18: showing collector dashboard

View Assigned Collection Tasks Page

This interface allowed collectors to view waste collection requests assigned to them by the administrator. The page displayed important information such as resident details, waste type, collection location, request date, and current collection status.

The interface helped collectors identify pending tasks and plan their collection activities effectively.

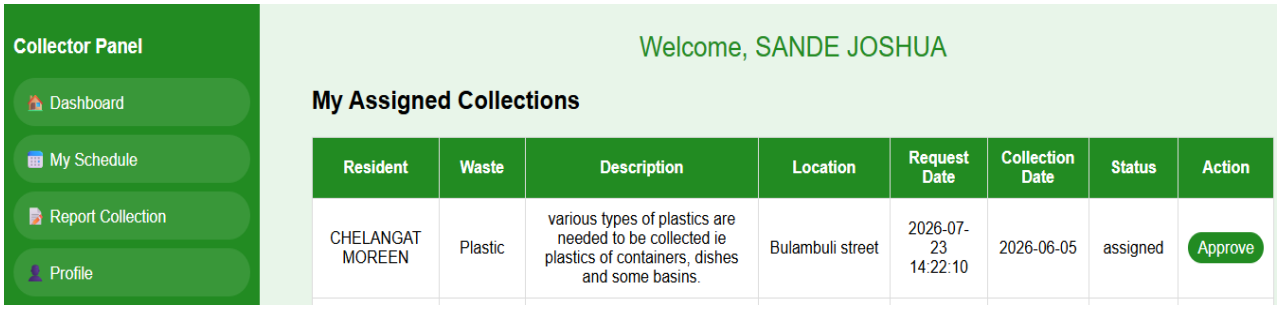


Figure 19: showing assigned collection tasks page

Update Collection Status Page

This interface allowed collectors to update the progress of waste collection activities after completing assigned tasks. Collectors were able to change the status of requests from assigned or pending to completed after successful waste pickup.

This helped maintain accurate collection records and enabled the administrator to monitor completed waste management activities.

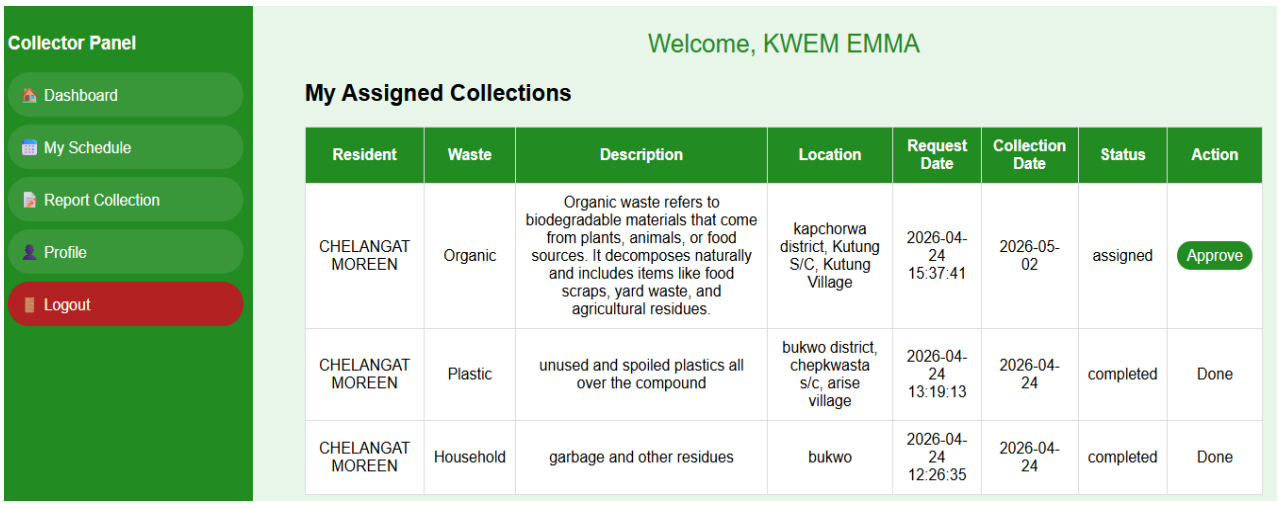


Figure 20: showing update collection status page

View Assigned Waste Collection Schedule Page

This interface allowed collectors to view their assigned waste collection schedules and manage their allocated waste pickup activities. After logging into the system, collectors were able to view waste collection tasks assigned to them by the administrator.

The page displayed important collection details including resident information, waste type, description, collection location, request date, and current collection status. From the same interface, collectors were able to update the collection status after successfully completing waste pickup activities.

This feature helped collectors organize their assigned tasks, monitor pending collections, and provide timely updates on completed waste collection activities. It also improved coordination between the administrator and collectors by ensuring that collection progress was accurately recorded in the system.

Figure 21: showing assigned waste collection schedule page

Report Completed Collection Page

This interface allowed collectors to submit reports after completing assigned waste collection activities. After finishing a waste pickup, the collector selected the completed job from the available assigned tasks, entered the collection date, provided a message or additional details about the completed activity, and submitted the report through the system.

The submitted completion report helped the administrator track completed waste collection activities and maintain accurate records of waste management operations. This feature improved communication between collectors and administrators by providing updates on completed collection tasks.

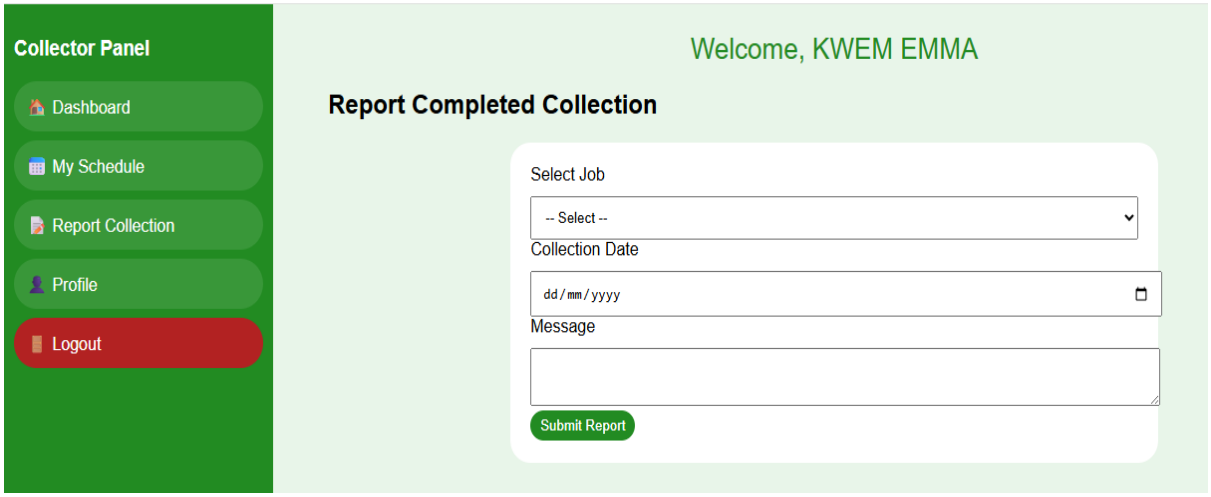


Figure 22: showing report completed collection page

Resident Side

Resident Login

This interface allowed residents to log into the Smart Waste Management System using their registered email and password. After successful authentication, residents were redirected to their dashboard where they could submit waste collection requests, monitor request progress, view approved collections, and receive updates from Ecopick Company.

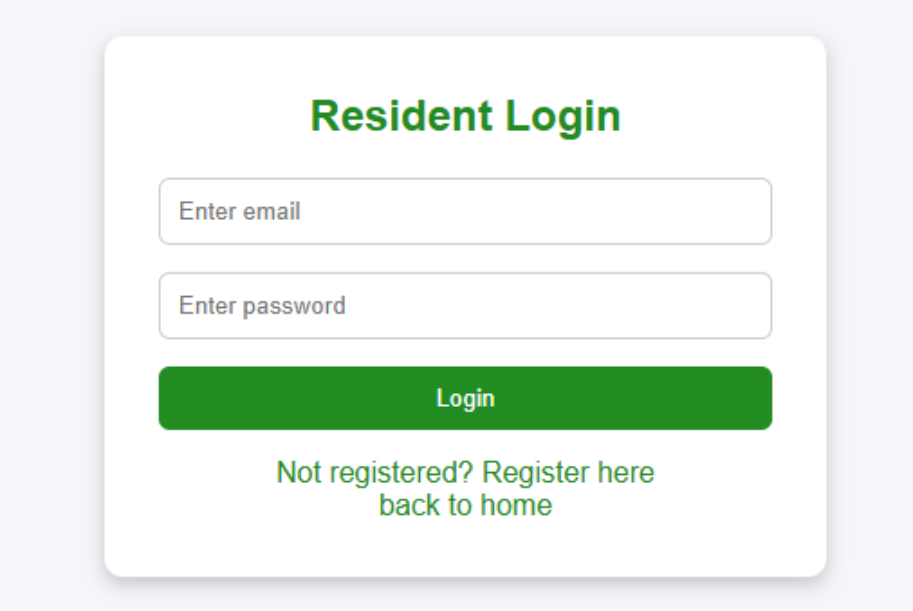


Figure 23: showing resident login page

Resident Registration Page

The resident registration page allowed new residents to create an account in the system by providing their personal details and location information. The registration process captured important details such as the resident's name, email address, phone number, and residential location.

After successful registration, residents could log into the system and access waste management services including requesting waste collection and tracking their collection activities.

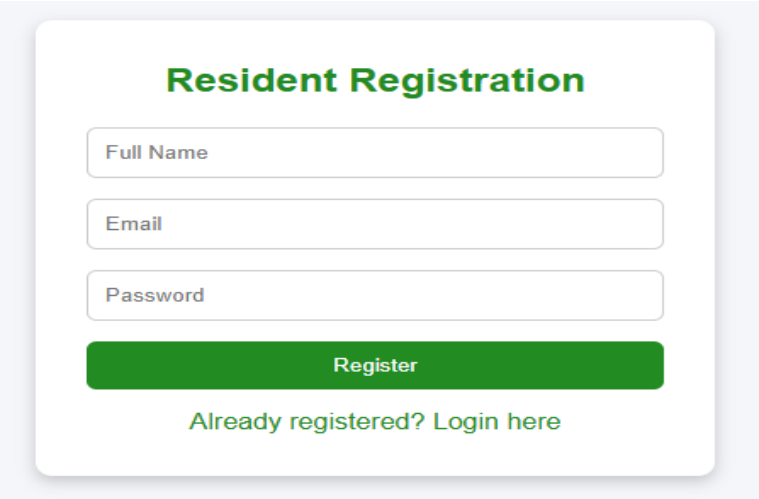


Figure 24: showing resident registration page

Resident Dashboard

The resident dashboard provided residents with a central platform for interacting with the Smart Waste Management System. Through the dashboard, residents could submit waste collection requests, view their previous requests, monitor collection status, view assigned collection schedules, and update their profile information.

The dashboard improved communication between residents and Ecopick Company by providing a simple and organized way of accessing waste collection services without relying on manual methods such as phone calls and physical records.

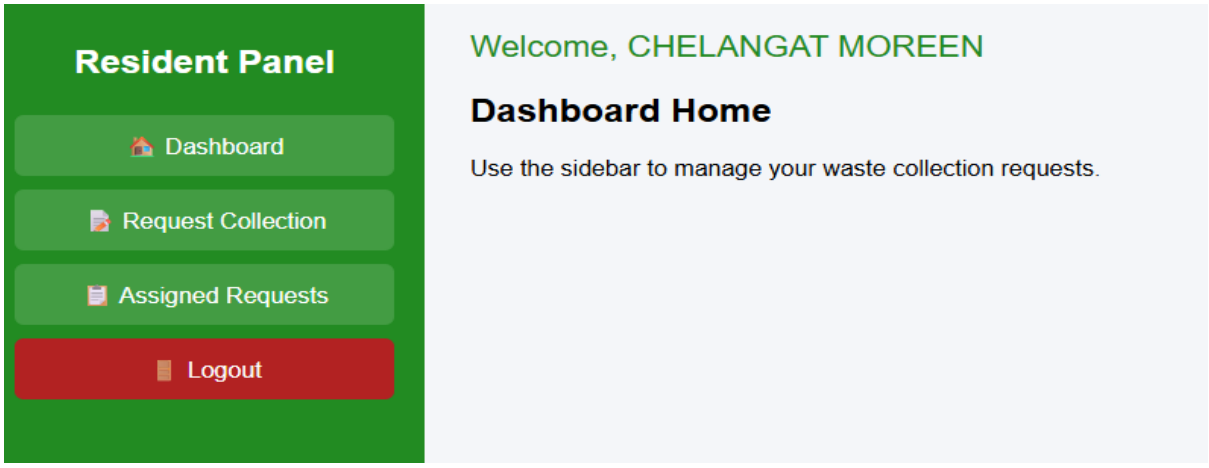


Figure 25: showing resident dashboard

Request Waste Collection Page

The waste collection request page allowed residents to request waste pickup services from Ecopick Company. Residents provided information such as waste type, description of waste, collection location, and preferred collection details.

Once submitted, the request was stored in the system and made available to the administrator for review and assignment to available collectors.

This helped to reduce delays and improve coordination between residents, administrators, and waste collectors.

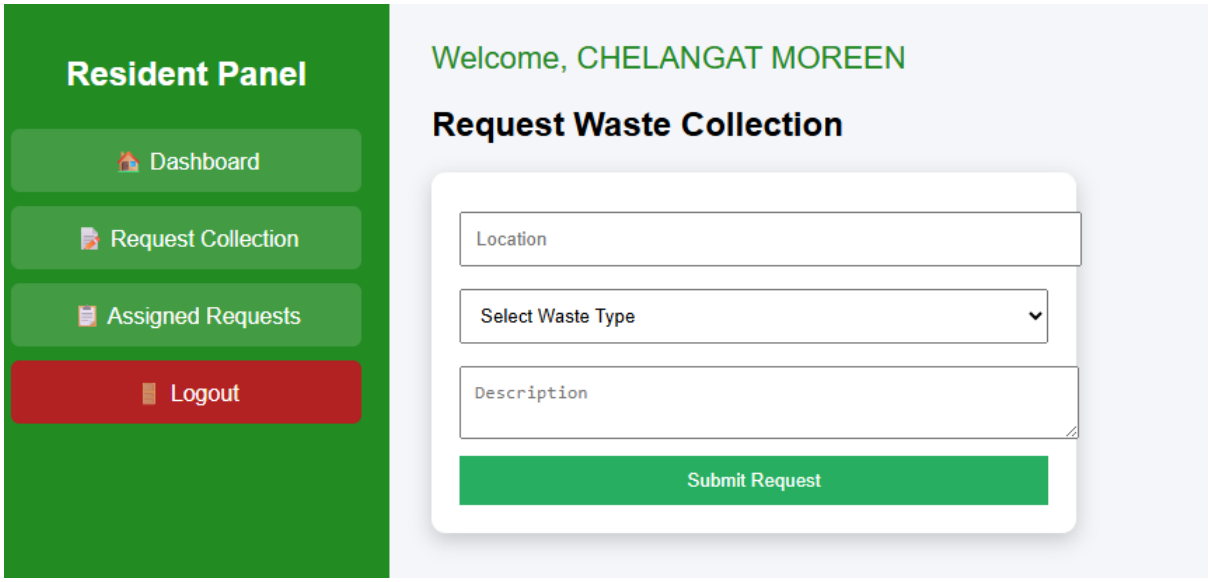


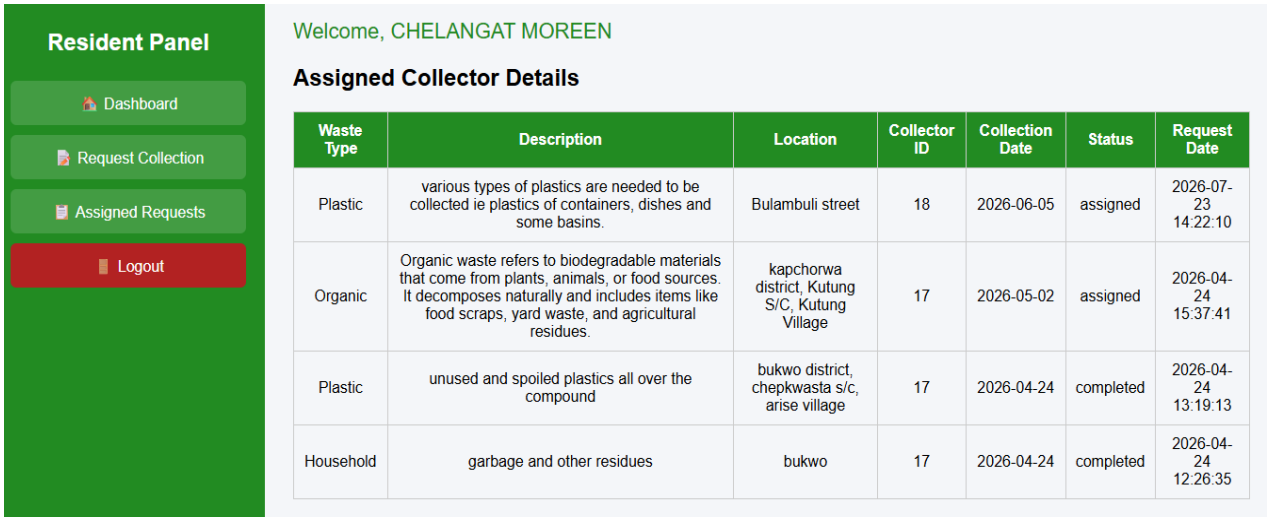
Figure 26: showing waste collection request page

View Collection Request Status Page

This page allowed residents to track the progress of their submitted waste collection requests. Residents could view important details such as request date, waste type, assigned collector, collection date, and current status.

The system displayed different stages of the request, including pending, assigned, and completed.

This improved transparency by allowing residents to monitor their requests without contacting Ecopick offices manually.



Waste Type	Description	Location	Collector ID	Collection Date	Status	Request Date
Plastic	various types of plastics are needed to be collected ie plastics of containers, dishes and some basins.	Bulambuli street	18	2026-06-05	assigned	2026-07-23 14:22:10
Organic	Organic waste refers to biodegradable materials that come from plants, animals, or food sources. It decomposes naturally and includes items like food scraps, yard waste, and agricultural residues.	kapchorwa district, Kutung S/C, Kutung Village	17	2026-05-02	assigned	2026-04-24 15:37:41
Plastic	unused and spoiled plastics all over the compound	bukwo district, chepkwasta s/c, arise village	17	2026-04-24	completed	2026-04-24 13:19:13
Household	garbage and other residues	bukwo	17	2026-04-24	completed	2026-04-24 12:26:35

Figure 27: showing resident request status page

key back-end components

All system data including users, children, sponsors, donations, and sponsorship records were stored in the MySQL database in tabular form. The database ensured data consistency, integrity, and easy retrieval of information when needed. Tables were linked using primary and foreign keys to maintain relationships between data and the data contained in the tables.

Database Design

A relational database was designed using MySQL to store data such as resident details, collector information, waste collection requests, collection records, administrator details, and system activity records for the Smart Waste Management System.

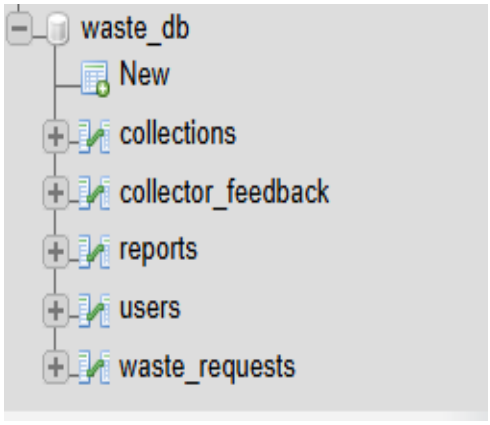


Figure 28: showing Smart waste management database.

Waste Requests Table

The waste requests table stores waste collection requests submitted by residents. It maintains details such as waste type, description, location, request status, and assigned collector information.

	request_id	resident_id	waste_type	description	request_date	status	collector_id	collection_time	created_at	location	collection_date
Delete	3	16	Household	garbage and other residues	2026-04-24 12:26:35	completed	17	*NULL*	2026-04-24 12:26:35	bukwo	2026-04-24
Delete	4	16	Plastic	unused and spoiled plastics all over the compound	2026-04-24 13:19:13	completed	17	*NULL*	2026-04-24 13:19:13	bukwo district, chepkwasta s/c, arise village	2026-04-24
Delete	5	16	Organic	Organic waste refers to biodegradable materials th...	2026-04-24 15:37:41	assigned	17	*NULL*	2026-04-24 15:37:41	kapchorwa district, Kutung S/C, Kutung Village	2026-05-02
Delete	6	19	Electronic	wasted metallic equipments	2026-04-24 17:04:00	completed	18	*NULL*	2026-04-24 17:04:00	bukwo district, kapsarur s/c, kapta village	2026-04-28
Delete	7	16	Plastic	various types of plastics are needed to be collect...	2026-07-23 14:22:10	assigned	18	*NULL*	2026-07-23 14:22:10	Bulambuli street	2026-06-05

Figure 29: showing waste requests table

Collections Table

The collections table stores completed and ongoing waste collection activities. It keeps records of collection dates, assigned collectors, and collection status.

Reports Table

The reports table stores generated waste management reports used by administrators for monitoring collection activities and evaluating system performance.

Collector Feedback Table

The collector feedback table stores feedback information related to collection activities, helping administrators monitor service delivery and identify areas for improvement.

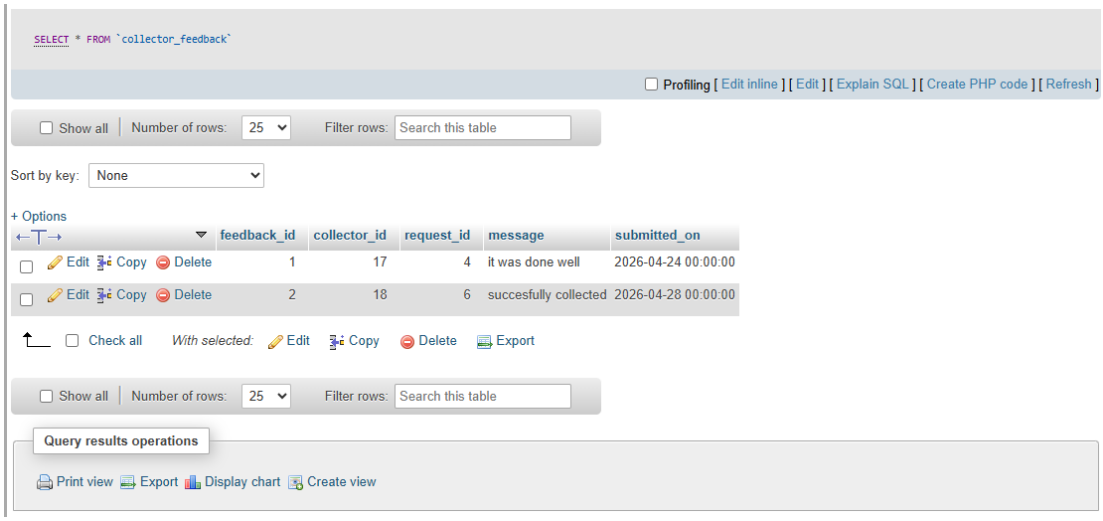


Figure 30: showing collector feedback table

SQL Injection Prevention and Prepared Statements

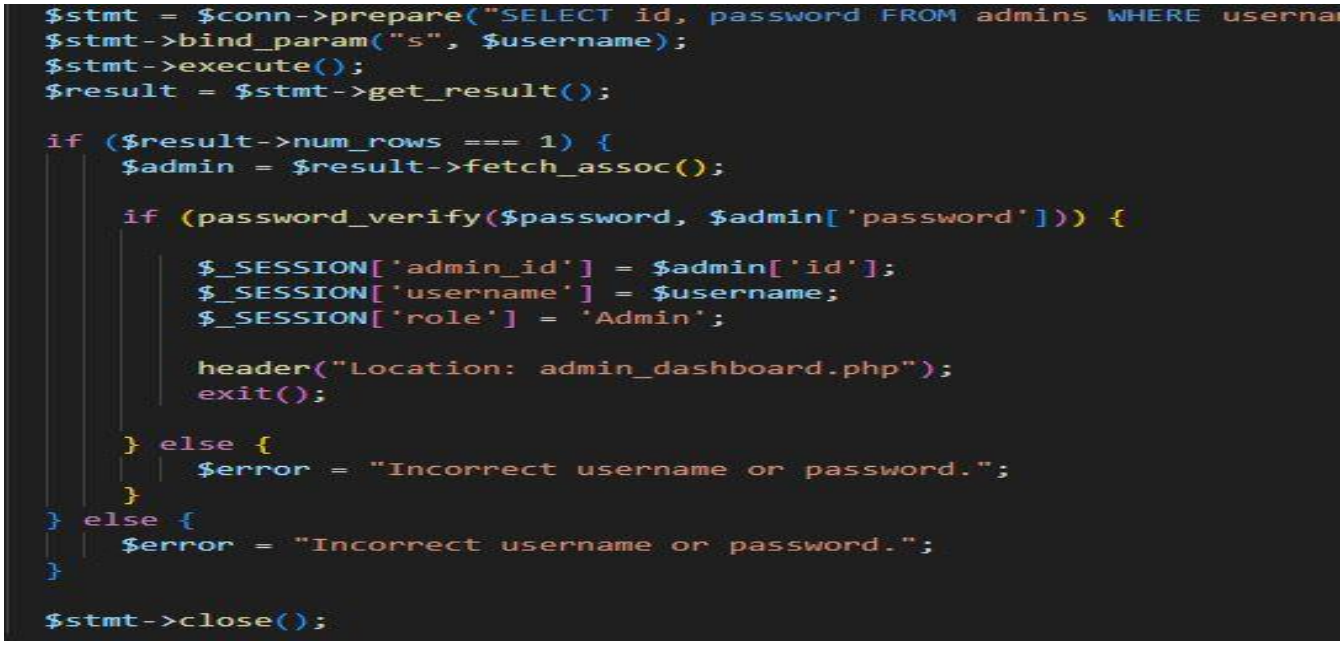
Security was a key focus during back-end development. The system used **Prepared Statements** via PHP Data Objects (PDO) or MySQL for all database actions.

This method secures the system because:

It separates the SQL query structure from the user data.

The database treats all user input strictly as text parameters.

It stops attackers from inserting malicious SQL commands into input fields.



```
$stmt = $conn->prepare("SELECT id, password FROM admins WHERE usernam
$stmt->bind_param("s", $username);
$stmt->execute();
$result = $stmt->get_result();

if ($result->num_rows === 1) {
    $admin = $result->fetch_assoc();

    if (password_verify($password, $admin['password'])) {

        $_SESSION['admin_id'] = $admin['id'];
        $_SESSION['username'] = $username;
        $_SESSION['role'] = 'Admin';

        header("Location: admin_dashboard.php");
        exit();

    } else {
        $error = "Incorrect username or password.";
    }
} else {
    $error = "Incorrect username or password.";
}

$stmt->close();
```

Figure 31: Showing Secure Parameterized Query Execution

Code Explanation

SQL Injection Defense. The snippet begins by using a prepared statement ($conn->prepare) to lookup the administrator accounts table. The query structure separates the SQL command from the user input by using a placeholder.

Binding Parameters. The $stmt->bind_param ("s", $username) method safely binds the input string variable to the placeholder. This completely neutralizes malicious injection inputs.

Retrieving Data. The code runs the statement and collects the records into memory using $stmt->get_result ().

User Account Validation

Account Check. The outer if condition checks if exactly one row ($result->num_rows === 1) matches the provided username.

Generic Error Messages. If no account matches the username, the code branches to the final else statement. It sets a vague error message: "Incorrect username or password." This is a critical security design pattern. It prevents malicious attackers from guessing valid usernames in the system.

Secure Password Verification

The password_verify () Function. Because passwords are saved as secure cryptographic hashes, they cannot be read directly as plain text. The code uses the built-in password_verify ()

function. This compares the plain text password typed into the login form against the encrypted database string ($admin ['password']).

Authentication Logic. If the password match is successful, the code opens up session storage variables ($_SESSION) to log the admin in and remembers their admin_id, username, and access privileges ('Admin').

Secure Page Redirection. Once authorized, the header () function securely forwards the user straight to the system's management panel (admin_dashboard.php), and the code calls exit () to immediately stop processing further layout scripts.

Password Hashing and Secure Storage

The system used secure password hashing to protect user accounts. Plain text passwords were never saved directly into the MySQL database tables. This ensures that even if an unauthorized person gains access to the database, they cannot read the user passwords.

The Hashing Mechanism

Algorithm Used. The system used the **password hash ()** function built into PHP, which relies on the strong **BCRYPT** algorithm.

Automatic Salting. The algorithm automatically adds a unique random string (called a salt) to each password before hashing it. This prevents identical passwords from looking the same in the database.

Verification. During login, the system used **password verify ()** to check if the entered password matches the stored secure hash.

```php
<?php
session_start();

// Database connection
$conn = mysqli_connect("localhost", "root", "", "waste_db");
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$message = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = mysqli_real_escape_string($conn, $_POST['name']);
    $email = mysqli_real_escape_string($conn, $_POST['email']);
    $password = password_hash($_POST['password'], PASSWORD_DEFAULT);
```

Figure 32: Database Connection and Variable Setup

Code Explanation

Database Connection and Variable Setup

Session Initialization: The session_start () function starts a user session, allowing the system to maintain user information during interaction with the application.

Database Connection: The code establishes a connection to the **waste_db** database using the MySQLi library. The connection uses the local server (localhost) with the default MySQL username and password settings.

Connection Validation: The if statement checks whether the database connection was successful. If the connection fails, the system stops execution using die() and displays the specific connection error message.

Form Data Processing: The code checks whether the request method is **POST**, which allows the system to receive and process user input submitted through forms, such as registration or login details.

Data Security: The mysqli_real_escape_string () function is used to prevent harmful characters from affecting database queries, while password_hash () securely encrypts user passwords before storing them in the database.

Form Request Handling and Data Cleansing

Post Verification. The script checks if the page received a POST request, ensuring code execution only occurs when a user actually clicks the registration form's submit button.
Data Sanitization. It extracts user inputs like name, email, phone number, and location from the $_POST array. The code uses the trim () function on each item to strip away any accidental leading or trailing blank spaces.

Secure Password Hashing

Supervisor Compliance. Your supervisor explicitly asked to "Explain how passwords are hashed and stored." This line directly implements that security measure.
The password hash () Function. The code avoids saving plain, readable text passwords into the database. Instead, it processes the user's password using PHP's built-in password hash () function.
The PASSWORD_DEFAULT Algorithm. It uses the strongest up-to-date standard algorithm available in PHP (currently shifting between bcrypt and Argon2 depending on the server

version). This algorithm automatically creates a unique cryptographically secure string that cannot be easily reversed or guessed by attackers, fully meeting standard data protection rules.

Database-Connection-Module

This module established a connection between the PHP application and the MySQL database, enabled data storage and retrieval operations across the system.

```php
// Database connection
$conn = mysqli_connect("localhost", "root", "", "waste_db");
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
```

Figure 33: Showing Database Connection Initialization code

Code Explanation

Database Connection Initialization

Purpose. This line establishes a live link between the PHP application and the MySQL database server.

The MySQL Class. It creates a new object using the MySQL () library, which is a built-in PHP tool used to talk to MySQL databases.

Connection Parameters. The code passes four specific pieces of information to log into the database server:

"Localhost". The host server where the database lives (running on the same computer as the code).

"Root": The default administrator username for the database server.

"". An empty password, which is common for local development environments.

"Waste_db". The specific name of the database that stores your system's tables (like children and donation logs).

Connection Error Checking

The Condition Check. The if statement automatically looks at the $conn->connect_error property. If the database login fails for any reason (like a wrong password or the server being turned off), this property will hold an error message.

System Termination. If an error is found, the code triggers the die() function. This instantly stops the entire program from running. It prevents the website from crashing later on when trying to read or write data that isn't there.

Error Feedback. The die () function prints out the exact text "Connection failed: " followed by the technical error message so the developer knows exactly why the system could not connect.

Waste Disposal Request Submission Handler

Overview: This PHP script handles the secure submission of a waste collection request from an authenticated resident. It verifies the user's session, sanitizes incoming form data, and inserts a new request into a database using secure prepared statements.

request_collection.php

```php
<?php
session_start();

if(!isset($_SESSION['resident_id'])){
    header("Location: login.php");
    exit();
}

include "db.php";

$resident_id = $_SESSION['resident_id'];
$message = "";

if(isset($_POST['submit'])){

    $location = trim($_POST['location']);
    $waste_type = trim($_POST['waste_type']);
    $description = trim($_POST['description']);
    $request_date = date("Y-m-d");

    /* PREPARED STATEMENT FOR SAFETY */
    $stmt = $conn->prepare("
        INSERT INTO waste_requests
        (resident_id, location, waste_type, description, request_date, status, collector_id, collection_dat
        VALUES (?, ?, ?, ?, ?, 'pending', NULL, NULL)
    ");

    $stmt->bind_param("issss", $resident_id, $location, $waste_type, $description, $request_date);

    if($stmt->execute()){
        $message = "✓ Your request has been submitted successfully.";
    } else {
```

Figure 34: Showing Session and Authentication Check code

Code Explanation

Session and Authentication Check

The script opens with a PHP tag and initializes a session to access stored user information. It checks if the unique resident identification exists in the active session data. If this identifier is

missing, the script redirects the user directly to the login page and halts all further execution to block unauthorized access.

Database Connection and Initialization

The file links an external database configuration script to establish a connection with the data server. It retrieves the authenticated resident's identification from the session and stores it locally. Additionally, it prepares an empty message variable to hold user notifications later on.

Form Submission Verification

The script evaluates if a form submission has occurred by looking for a specific post request trigger. This prevents the processing code from running prematurely when a user simply loads the webpage.

Form Data Collection and Sanitization

Once a submission is confirmed, the code extracts the location, type of waste, and text description from the incoming request. It applies a trimming function to strip any accidental blank spaces from these inputs. It also generates the current calendar date automatically using a standard year-month-day layout.

Secure Database Insertion

The handler compiles an insertion template targeting the waste requests registry table. It uses a secure prepared statement structure to neutralize potential database injection threats. Instead of passing user inputs directly into the database command, it utilizes safe placeholder markers for the user data while hardcoding default values for status tracking.

Parameter Binding and Execution

The script links the sanitized user variables explicitly to the designated placeholder markers, declaring the exact data types for each entry (one whole number and four text segments). It then executes the compiled command. If the database registers the new entry successfully, the code stores a confirmation message to notify the user.

1. Unit Testing

Unit testing was conducted on individual modules of the **Smart Waste Management System (SWMS)**, including login, resident registration, waste collection requests, collector task assignment, and collection recording. Each module was tested independently to ensure that it functions correctly, processes data accurately, and stores information properly in the database.

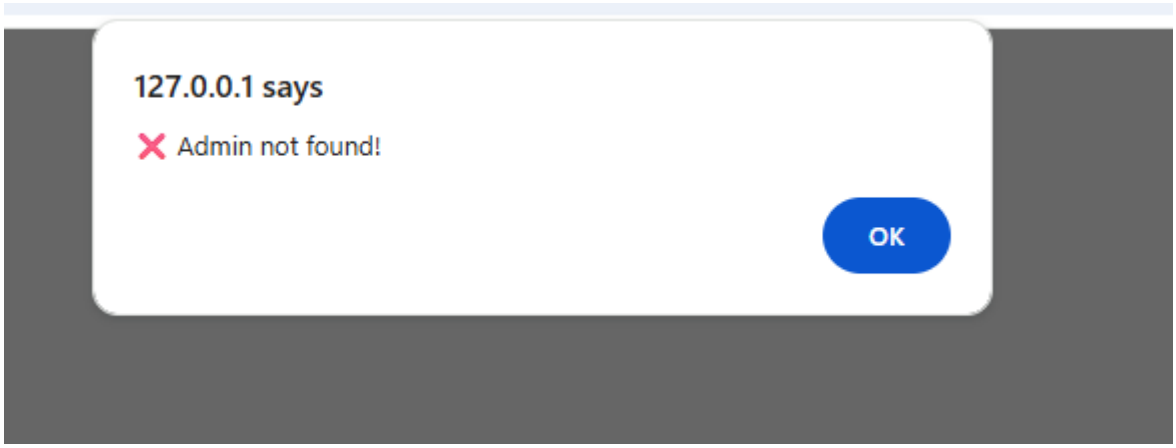


Figure 35: Showing Unit Testing

2. Integration Testing

Integration testing was carried out after combining all modules of the **Smart Waste Management System (SWMS)** to ensure they work together effectively. The testing verified proper data flow between modules, such as linking residents, waste collection requests, collectors, and administrators. It ensured that the complete system operates smoothly without errors.

ID	Resident ID	Waste Type	Description	Location	Request Date	Status	Assign Collector & Date
6	19	Electronic	wasted metallic equipments	bukwo district, kapsarur s/c, kapta village	2026-04-24 17:04:00	COMPLETED	-- Select Collector -- dd/mm/yyyy Assign

Figure 36: Showing Integration Testing

Comprehensive Test Cases and Results

The table below presents the detailed test cases conducted during system testing, including the modules tested, expected outcomes, actual results obtained, and the status of each test. The results indicate that all tested functionalities operated as intended and met the specified system requirements.

Table 13: Showing Comprehensive Test Cases and Results

Test ID	**Module Tested**	**Test Input / Action**	**Expected Outcome**	**Actual Outcome**	**Status**
TC-01	User Login	Enter valid username/email and password.	User logs in successfully and is redirected to their respective dashboard.	User accessed the correct dashboard successfully.	Pass
TC-02	User Login	Enter invalid password with a valid username/email.	System rejects login and displays an error message.	"Invalid Credentials" message displayed successfully.	Pass
TC-03	Waste Collection Request	Submit a waste collection request with complete details such as waste type and location.	Request is saved successfully in the database.	Waste collection request recorded successfully.	Pass
TC-04	Waste Request Search/View	Search or view submitted waste collection requests.	System displays the correct request details.	Correct waste request information displayed.	Pass
TC-05	Collector Task Assignment	Admin assigns a pending waste request to a collector.	Assigned collector receives the task and request status	Collector assignment completed	Pass

Test ID	Module Tested	Test Input / Action	Expected Outcome	Actual Outcome	Status
			changes to assigned.	successfully and status updated.	
TC-06	Collection Completion	Collector marks an assigned waste collection task as completed.	System updates collection status and records completion details.	Collection status updated successfully.	Pass
TC-07	Access Control	Attempt to access admin or collector dashboard without logging in.	System blocks unauthorized access and redirects user to login page.	Unauthorized access was blocked successfully.	Pass

User Acceptance Testing (UAT)

UAT Overview and Methodology

User Acceptance Testing was carried out to ensure that the **Smart Waste Management System (SWMS)** met the actual needs of the end users at **Ecopick Company, Mbale City**. The system was deployed on a local network environment for testing. Three main groups of users participated in the testing: the System Administrator, Waste Collectors, and Residents.

Each user performed common system tasks such as logging in, submitting waste collection requests, assigning collection tasks, and recording completed collections. After testing, users evaluated different sections of the system using a rating scale of 1 to 5, where 1 represented Poor and 5 represented Excellent.

UAT Results and Feedback

The table below summarizes the feedback and acceptance status from the actual users who tested the **Smart Waste Management System (SWMS)** at **Ecopick Company, Mbale City**.

Table 14: Showing Showing User Acceptance Testing and Feedback

User Role	**Number of Testers**	**Target Task Tested**	**Average Rating (1-5)**	**User Feedback / Comments**	**Acceptance Status**
Administrator	1	Managing user accounts, assigning collectors, monitoring waste requests, and generating collection reports.	4.8	"The system makes it easier to monitor waste collection activities and generate reports compared to manual records."	Accepted
Waste Collectors	2	Logging in, viewing assigned collection tasks, updating collection status, and recording completed pickups.	4.5	"The task assignment feature helps me know the waste collections I am responsible for and update their progress easily."	Accepted
Residents	2	Creating accounts, submitting waste collection requests, and viewing request status.	4.6	"The system provides an easier way to request waste collection and track the progress of submitted requests."	Accepted

Summary:

The UAT results showed that all user groups accepted the system, with high ratings indicating that the **Smart Waste Management System** was easy to use, met user requirements, and improved waste collection management processes.

UAT Conclusion

All targeted user groups successfully completed their assigned tasks during User Acceptance Testing. The system achieved an overall satisfaction rating above **4.5 out of 5**. Based on the positive feedback received from administrators, collectors, and residents, the **Smart Waste Management System (SWMS)** was accepted as suitable for improving waste collection management processes at **Ecopick Company, Mbale City**.

Security Testing

Security Testing Overview

Security testing was performed to ensure that user information and waste collection records stored in the **Smart Waste Management System (SWMS)** were protected. The testing verified that only registered users could access the system and that users could only access functions permitted according to their roles, including Administrator, Collector, and Resident.

Security Test Areas and Results

The security testing focused on three main areas: **Authentication, Access Control, and Data Protection.**

• Authentication Testing

The system was tested to ensure that users entered valid login credentials before accessing the system. Tests included entering blank fields, incorrect passwords, and invalid email addresses. The login module successfully rejected unauthorized login attempts and only allowed users with valid credentials to access their dashboards.

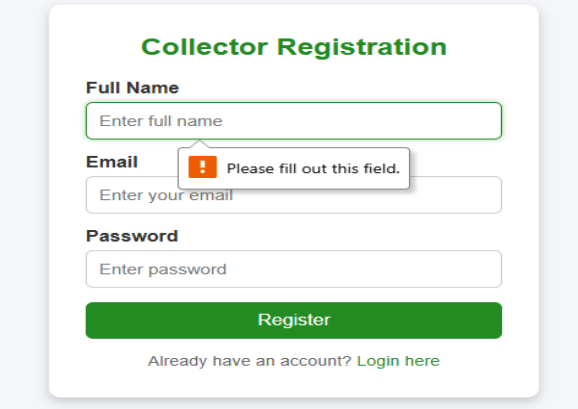


Figure 37: Showing authentication testing

Login page with email and password fields

Invalid login message such as **"Invalid Credentials"**

Successful login redirect to dashboard

• Access Control Testing (Role-Based Security)

Access control testing was conducted to verify that users could only access pages assigned to their roles. For example, residents and collectors were prevented from accessing the administrator dashboard by manually entering the admin page URL in the browser.

The system successfully blocked unauthorized access attempts and redirected users back to the login page or their permitted dashboard.

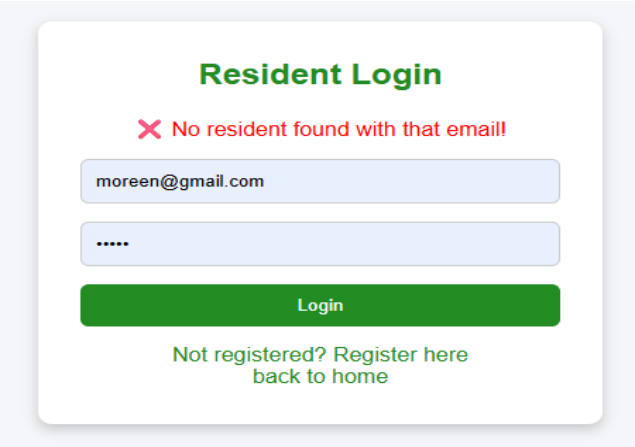


Figure 38: Showing role-based access control testing

This demonstrates that:

Only administrators access admin pages.

Collectors access collector pages.

Residents access resident pages.

• Data Protection Testing

The system was also tested to ensure that stored information such as resident details, waste requests, and collection records was protected from unauthorized access and database attacks.

Prepared statements were used during database operations to prevent SQL injection attacks and improve data security.

	user_id	name	email	password	role	created_at
Edit Copy Delete	17	KWEM EMMA	emma@gmail.com	$2y$10$.f5tdAyr2LnNnqpXFIsC6Ob7mjCGQU0wlxcFceCXvur...	collector	2026-04-24 12:29:45
Edit Copy Delete	18	SANDE JOSHUA	josh2@gmail.com	$2y$10$8nHts4PDVf534Vp0ZCzmjOUn96uYwx6GgcMZk4OQtrM...	collector	2026-04-24 15:41:05
Edit Copy Delete	16	CHELANGAT MOREEN	moreen@gmail.com	$2y$10$xqEOGo8Sw3.KGrWFnc59H.2kP.CXnEHsdHN9JoZaloM...	resident	2026-04-24 12:24:41
Edit Copy Delete	15	System Admin	admin@gmail.com	$2y$10$CaeLE9ZvxZHajPHdotn7JOX2CiwzkGQ93lLeZqHM7KS...	admin	2026-04-18 13:09:58
Edit Copy Delete	19	YEKO GEORGE	yekogeorge@gmail.com	$2y$10$CuoKGx35wCS21RbOSl6ameQF9Ve8cHg2EeBjUu5OjpK...	resident	2026-04-24 17:01:30

Figure 39: Showing data protection using prepared statements

This proves that the system securely handles database transactions.

5.4 System Error Handling Mechanisms

Error handling was an important part of the **Smart Waste Management System (SWMS)** design. It ensured that when unexpected inputs, software errors, or database problems occurred, the system responded properly without crashing or exposing sensitive information. The system implemented three main levels of error handling.

Graceful Connection Failures (Database Level)

• **The Mechanism:**
The system used conditional validation checks immediately after initializing the database connection. These checks verified whether the PHP application successfully connected to the MySQL database named waste_db.

• **The Action:**
If the system failed to establish a connection with the database server, the program stopped execution using the die () function and displayed a simple connection error message.

• **Why it mattered:**
This prevented users from viewing technical database details such as server paths, database credentials, or system configuration information. It also helped administrators identify connection problems during maintenance.

Input Validation and Parameter Integrity Checks (Application Level)

• **The Mechanism:**
The system performed input validation on information entered by users, including login details, resident registration data, waste collection requests, and collection records. The system used validation checks and status messages to monitor successful and failed operations.

• **The Action:**
When users submitted incomplete information or entered incorrect details, the system detected the problem and displayed clear messages such as “Please fill all required fields” or “Invalid login credentials.”

• **Why it mattered:**
This improved user experience by guiding users to provide correct information while preventing invalid data from being stored in the database. It also reduced security risks caused by incorrect or harmful inputs.

Secure Query Execution Failure (Database Engine Level)

• **The Mechanism:**
The system used MySQL prepared statements ($stmt->execute ()) when inserting, updating, or retrieving waste management records. This separated SQL commands from user-provided data.

• **The Action:**
If a database operation failed due to issues such as duplicate records, missing information, or database restrictions, the system captured the error and displayed a controlled message instead of exposing raw database errors.

• **Why it mattered:**
This protected the internal database structure, prevented SQL injection attacks, and ensured that failures did not interrupt the normal operation of the Smart Waste Management System.

5.5 Responsive Design and Mobile Compatibility

The **Smart Waste Management System (SWMS)** was designed to be responsive and accessible on different devices. The user interface automatically adjusted according to the

screen size, allowing administrators, collectors, and residents to access the system using desktop computers, tablets, and smartphones.

Core Framework and Grid Layout

• **Flexible Layouts:**
The system interface was developed using CSS responsive techniques, including flexible layouts and media queries, to allow pages such as dashboards, forms, and tables to adjust according to different screen sizes.

• **Automatic Resizing:**
Instead of using fixed-width elements, the system used flexible sizing methods that allowed menus, input forms, and information tables to expand or reduce depending on the user's device screen.

Mobile Compatibility for Users

• **Residents:**
Residents were able to access the system using mobile devices to create accounts, submit waste collection requests, provide collection locations, and monitor the status of their requests.

• **Collectors:**
Collectors could use smartphones or tablets to log in, view assigned waste collection tasks, update collection status, and confirm completed waste pickups while in the field.

• **Administrators:**
The administrator dashboard was designed to adapt to smaller screens. Large tables containing users, requests, and collection reports could be viewed through horizontal scrolling, while navigation menus adjusted for easier mobile access.

• **Touch-Friendly Inputs:**
Buttons, forms, dropdown menus, and input fields were designed with suitable sizes to support easy interaction on touchscreen devices and reduce accidental selections.

Cross-Browser Optimization

• **Consistency:**
The responsive design was tested on different browsers including Google Chrome, Mozilla

Firefox, and Microsoft Edge to ensure that users experienced consistent system performance regardless of the browser or device used.

5.6 System Performance and Load Testing

Performance testing was conducted to evaluate the speed, reliability, and stability of the **Smart Waste Management System (SWMS)** under different operating conditions. The purpose of the testing was to confirm that the system could handle daily waste management activities without delays, failures, or performance problems.

Performance Testing Environment

• **Testing Environment:**

Performance testing was conducted using the local Apache and MySQL server environment to simulate the real operational environment of the system.

• **Database Size:**

The test database was populated with sample records including resident accounts, collector profiles, waste collection requests, and completed collection records to simulate realistic system usage.

Key Performance Metrics and Results

Response Time (Speed)

• **Page Loading:**

The average loading time for important system pages such as dashboards, waste request pages, and collection records remained within an acceptable range, allowing users to access information quickly.

• **Form Submission:**

Submitting forms such as waste collection requests and collection completion updates was processed successfully within a short time due to optimized database queries and secure prepared statements.

Load Testing (Stress/Users)

• **Simulated Traffic:**

The system was tested with multiple users accessing different functions simultaneously,

including residents submitting requests, collectors updating tasks, and administrators monitoring activities.

• **Server Performance:**
During testing, the system remained stable and continued responding correctly without major delays or crashes.

• **Error Rate:**
No major system failures occurred during testing, and all major pages loaded successfully under the simulated user activities.

Performance Optimization Techniques

Database Optimization

• **Database Indexing:**
Primary keys and foreign keys such as user_id, resident_id, collector_id, and request_id were properly structured to improve searching, updating, and retrieving records efficiently.

• **Query Optimization:**
Prepared SQL statements were used to improve database security and performance during data processing.

Image and File Optimization

• **Image Handling:**
Where image uploads were required, such as profile images, the system was configured to reduce unnecessary file sizes to improve loading speed and storage efficiency.

5.7 Chapter Summary

In conclusion, Chapter Five presented the implementation, testing, and evaluation of the Smart Waste Management System (SWMS). The system was successfully developed using PHP, MySQL, HTML, CSS, and JavaScript to automate waste collection activities at Ecopick Company, Mbale City.

The implementation focused on developing key modules including user authentication, resident registration, waste collection requests, collector task assignment, collection tracking,

and reporting. Security measures such as prepared statements, role-based access control, and error handling mechanisms were implemented to protect system data.

Furthermore, responsive design principles were applied to ensure that administrators, collectors, and residents could access the system using different devices. Performance testing confirmed that the system was stable, responsive, and capable of supporting daily waste management operations.

With successful implementation, testing, and user evaluation, the Smart Waste Management System was considered ready for deployment and practical use in improving waste collection coordination, monitoring, and reporting.

CHAPTER SIX: DISCUSSION, CONCLUSION, RECOMMENDATIONS, LIMITATIONS AND FUTURE WORK

6.1 Introduction

This chapter presents the discussion of the results obtained from the design and implementation of the Smart Waste Management System (SWMS) for Ecopick Company, Mbale City. It also provides conclusions drawn from the study, recommendations for improvement, limitations encountered during system development, and suggestions for future enhancements. The discussion is based on the specific objectives of the study and supported by findings from the reviewed literature.

6.2 Discussion of Results

The findings from the system development, implementation, and testing processes were discussed in relation to the study objectives and the literature reviewed in Chapter Two.

Objective 1: To identify the system requirements for the Smart Waste Management System at Ecopick Company

The study began by examining the existing waste management processes used at Ecopick Company. The findings showed that the company relied mainly on manual methods such as phone calls, WhatsApp communication, paper records, and spreadsheets to manage waste collection activities.

These methods created several challenges, including delays in responding to collection requests, poor communication between residents and collectors, difficulties in tracking collection activities, loss of records, and challenges in generating accurate reports.

To address these challenges, system requirements were gathered from administrators, waste collectors, and residents through interviews and observations. The functional requirements identified included user registration and login, waste collection request submission, collector assignment, collection status updates, waste monitoring, and report generation. Non-functional requirements included security, usability, reliability, and system performance.

The findings confirmed that the existing manual waste management approach was inefficient and that there was a need for a digital system to improve waste collection coordination, accountability, and service delivery at Ecopick Company.

Objective 2: To design the proposed Smart Waste Management System

After identifying the system requirements, the proposed Smart Waste Management System was designed to address the challenges identified in the existing waste collection process.

The system design involved the development of system models including use case diagrams, context diagrams, data flow diagrams, and entity relationship diagrams to represent system processes, user interactions, and database relationships.

The design focused on creating a user-friendly interface for the three main users of the system: administrators, waste collectors, and residents. Different modules were designed, including user management, waste collection request management, collector assignment, collection tracking, and report generation.

Security features such as user authentication and role-based access control were incorporated to ensure that users could only access functions assigned to their roles.

The design phase provided a clear structure for developing a system that could effectively improve waste collection management at Ecopick Company.

Objective 3: To implement the designed Smart Waste Management System

The Smart Waste Management System was implemented using PHP, MySQL, HTML, CSS, and JavaScript as described in the methodology chapter. A centralized database named waste_db was developed to store information related to users, residents, collectors, waste requests, and collection records.

Interfaces were created to allow residents to submit waste collection requests, administrators to manage users and assign collectors, and collectors to view assigned tasks and update collection status.

During implementation, attention was given to data validation, security, usability, and proper database management. Prepared statements were used to prevent SQL injection, while role-based access control was implemented to protect system information.

The implemented system automated major waste management activities, reduced dependence on manual records, and improved the efficiency of waste collection operations at Ecopick Company.

Objective 4: To test and validate the functionality and performance of the developed system

The developed Smart Waste Management System was tested using different testing approaches, including unit testing, integration testing, user acceptance testing, and security testing.

The testing results showed that the major modules, including user login, resident registration, waste collection requests, collector assignment, collection completion, and report generation, functioned correctly.

During user acceptance testing, administrators, collectors, and residents successfully completed their assigned tasks and provided positive feedback about the usability and effectiveness of the system.

Security testing confirmed that authentication and role-based access controls were working correctly, preventing unauthorized users from accessing restricted sections of the system.

The testing results validated that the Smart Waste Management System improved efficiency, transparency, and coordination between residents, collectors, and administrators.

6.3 Conclusion

The study successfully achieved its objectives by identifying requirements, designing, implementing, and testing a Smart Waste Management System for Ecopick Company, Mbale City.

The developed system addressed challenges associated with manual waste management practices by providing a centralized digital platform for managing waste collection requests, collector assignments, collection monitoring, and reporting.

The system improved communication between residents and waste collectors, enhanced record accuracy, increased accountability, and supported better decision-making through automated reporting.

Therefore, the developed Smart Waste Management System provides a practical solution for improving waste collection management and service delivery at Ecopick Company.

6.4 Recommendations

Based on the findings and testing results of the study, the following recommendations were made:

- Ecopick Company was recommended to adopt the Smart Waste Management System to improve the management and monitoring of waste collection activities.
- Regular training of administrators and waste collectors was recommended to ensure effective and efficient use of the system.
- The company was advised to continuously maintain and update the system to improve security, performance, and additional functionalities.
- Improvement of internet connectivity and availability of adequate computer and mobile devices was recommended to support smooth system operation.
- Integration of the system with communication services such as SMS and email notifications was recommended to improve communication between residents, collectors, and administrators.

- The company was encouraged to use system-generated reports for planning, monitoring performance, and improving waste collection services.

6.5 Limitations

Despite the success of the project, several limitations were encountered during the development of the Smart Waste Management System:

- Limited time and resources restricted extensive testing with a large number of users and real-time waste collection activities.
- Some users had limited computer knowledge, which affected the speed of testing and feedback collection.
- The system depended on internet availability, making access difficult in areas experiencing poor network connectivity.
- Advanced features such as GPS-based waste tracking, mobile applications, and automatic notifications were not implemented due to time and resource limitations.

6.6 Future Work

To further improve the Smart Waste Management System, future work should focus on:

- Developing a mobile application for residents, collectors, and administrators to improve accessibility.
- Integrating GPS tracking to allow real-time monitoring of waste collection vehicles and collector locations.
- Adding SMS and email notification features to alert residents about collection schedules and request updates.
- Implementing online payment options for waste collection services.
- Expanding the system to support multiple waste management companies and different locations.
- Enhancing offline functionality to allow users to record waste collection activities without an active internet connection.

- Integrating data analytics dashboards to provide insights on waste collection trends and service performance.

REFERENCES

Ahmad, R. W., Salah, K., Jayaraman, R., Yaqoob, I., & Omar, M. (2021). Blockchain for Waste Management in Smart Cities: A Survey. *IEEE Access*, *9*, 131520–131541. https://doi.org/10.1109/ACCESS.2021.3113380

Fatimah, Y. A., Govindan, K., Murniningsih, R., & Setiawan, A. (2020). Industry 4.0 based sustainable circular economy approach for smart waste management system to achieve sustainable development goals: A case study of Indonesia. *Journal of Cleaner Production*, *269*. https://doi.org/10.1016/j.jclepro.2020.122263

FRED, T. (2025). *Assessment of metal contaminants and human health risks in food crops at the uncontrolled Mbale Dumpsite, Uganda*. *October*.

Guerrero, L. A., Maas, G., & Hogland, W. (2013). Solid waste management challenges for cities in developing countries. *Waste Management*, *33*(1), 220–232. https://doi.org/10.1016/j.wasman.2012.09.008

Krystosik, A., Njoroge, G., Odhiambo, L., Forsyth, J. E., Mutuku, F., & LaBeaud, A. D. (2020). Solid Wastes Provide Breeding Sites, Burrows, and Food for Biological Disease Vectors, and Urban Zoonotic Reservoirs: A Call to Action for Solutions-Based Research. *Frontiers in Public Health*, *7*(January), 1–17. https://doi.org/10.3389/fpubh.2019.00405

Lee, D., Ratti, C., & Chair, L. J. V. (2015). *Transforming waste management systems through location tracking and data sharing Signature redacted Signature redacted Transforming waste management systems through location tracking and data sharing by*. 1–156.

Li, N., Sepúlveda, N., & Li, N. (2011). IEEE Xplore Full-Text PDF_李玲琪 自聚焦 写波导. In *Proceedings of the 2011 IEEE International Conference on Robotics and Biomimetics*

(Vol. 17, Issue 1, pp. 1343–1348). https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7781053%0Ahttps://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5485276%0Ahttps://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9511469%0Ahttps://ieeexplore-ieee-org.uproxy.library.dc-uoit.ca/

Marshall, R. E., & Farahbakhsh, K. (2013). Systems approaches to integrated solid waste management in developing countries. *Waste Management*, *33*(4), 988–1003. https://doi.org/10.1016/j.wasman.2012.12.023

Principle, C. (2023). *Non-Contact Adaptive Voltage Sensor Based on Electric Field Coupling Principle*.

Rajesh Prabu Vincent De Paul. (2025). Building Scalable API-Led Connectivity Using Three-Tier Architecture Patterns. *Journal of Computer Science and Technology Studies*, *7*(7), 598–605. https://doi.org/10.32996/jcsts.2025.7.7.67

Rifka Alkhilyatul Ma'rifat, I Made Suraharta, I. I. J. (2024). *Rapid Urbanization and its Impact on Municipal Solid Waste Management in the Greater Accra Region of Ghana.* *2*, 306–312.

Science, I. (2024). *an Evaluation of the State of Record Management Amongst*.

Simpson, M. (2021). *The role of the administrator in the management of data quality in higher education: case study in a health sciences faculty*. *March*, 1–219.

Siwawa1, D. I. V., & Phuluwa2, and D. H. S. (1970). *Pr ep rin t n ot pe er re v iew Pr ep rin t n ot pe er re v ed*. 1–2.

Stockdale, A. J. (2020). *Hepatitis B and C in Malawi : Epidemiology , Disease Burden and Opportunities for a Public Health Treatment Programme*. *March*.

Whitney G.G., F. D. R. (2008). THE EFFECTS OF POOR SANITATION ON THE COMMUNITy OF KABALAGALA SUB- COUNTY, MAKINDyE DIVISION, KAMPALA, CENTRAL UGANDA. 管理现代化, *76*(3), 61–64.

Zohoori, M., & Ghani, A. (2017). Municipal Solid Waste Management Challenges and Problems for Cities in Low-Income and Developing Countries. *International Journal of Science and Engineering Applications*, *6*(2), 039–048.

https://doi.org/10.7753/ijsea0602.1002