



Web Based Application for Waste Management in Port-Harcourt Metropolis

Linda Uchenna Oghenekaro, Peters Ledum Favour

Faculty of Computing, University of Port Harcourt, Rivers State, Nigeria

Email: linda.oghenekaro@uniport.edu.ng, favourpeters1@gmail.com

How to cite this paper: Oghenekaro, L.U. and Favour, P.L. (2025) Web Based Application for Waste Management in Port-Harcourt Metropolis. *Open Access Library Journal*, **12**: e12930.

<https://doi.org/10.4236/oalib.1112930>

Received: January 7, 2025

Accepted: March 11, 2025

Published: March 14, 2025

Copyright © 2025 by author(s) and Open Access Library Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

The development of a web-based waste management application addresses the unique challenges of waste collection in Port Harcourt Metropolis. Traditional waste management practices often suffer from inefficiencies such as irregular schedules, ineffective communication, and limited tracking capabilities, leading to environmental and public health concerns. The proposed system integrates cutting-edge technologies, including React for frontend design, Node.js and Express.js for backend operations, MongoDB for data management, and GPS APIs like Google Maps for real-time vehicle tracking. Additionally, a secure payment gateway, Paystack, is incorporated to facilitate seamless financial transactions. The application mitigates inefficiencies through advanced scheduling algorithms, real-time notifications, and a user-friendly interface designed to enhance accessibility and engagement. Users can schedule waste pickups, track collection vehicles, and make payments online, while service providers optimize routes and improve operational efficiency. This innovative solution improves urban cleanliness, reduces environmental hazards, and fosters community participation in sustainable waste management. The findings demonstrate the potential of web-based platforms to revolutionize waste management systems by enhancing real-time tracking, optimizing collection routes, and increasing user participation. A comparative evaluation revealed a 40% improvement in scheduling efficiency, 25% reduction in collection time, and 35% increase in user engagement, highlighting the system's effectiveness. The model offers a scalable and adaptable solution for other urban settings.

Subject Areas

Environmental Science and Technology

Keywords

Waste Management, Web-Based Waste Management, Waste Collection

1. Introduction

Effective waste management is essential for maintaining urban cleanliness, public health, and environmental sustainability. With the rapid urbanization and population growth in Port Harcourt Metropolis, the traditional methods of waste collection and disposal face significant challenges, including irregular waste collection schedules, poor routing, and inadequate communication between service providers and residents. These inefficiencies not only compromise service delivery but also contribute to environmental pollution and public health risks [1] [2].

Recent advancements in information technology have paved the way for innovative solutions to urban waste management issues. Web-based applications, in particular, offer a platform for integrating real-time data tracking, automated scheduling, and seamless communication between users and service providers. Leveraging technologies such as React, Node.js, MongoDB, and GPS APIs, these systems enhance operational efficiency while providing a user-friendly interface for residents [3] [4].

This research aims to develop a web-based waste management system tailored to address the specific challenges of waste collection in Port Harcourt Metropolis. The proposed system features advanced scheduling algorithms, real-time vehicle tracking, and secure online payment capabilities through Paystack, creating a comprehensive solution that bridges the gap between waste generators and collectors. By streamlining the waste management process, this system aims to reduce environmental hazards, enhance user satisfaction, and promote sustainable practices within the community.

This research contributes to the broader field of waste management by demonstrating the scalability and adaptability of web-based solutions. It highlights how the integration of technology can transform waste management practices, setting a benchmark for other urban centers facing similar challenges [5].

2. Literature Review

Efficient waste management systems are critical for mitigating environmental pollution and public health risks. Traditional approaches often rely on manual operations and fixed schedules, which are inefficient in addressing the dynamic needs of urban waste collection. Recent advancements in technology, particularly in web-based and IoT-enabled solutions, offer innovative approaches to address these challenges [1] [2].

2.1. Waste Management Systems

Traditional waste management systems primarily focus on manual collection, open dumping, and incineration. While these methods are prevalent in developing regions, they often lead to environmental degradation and health risks due to poor resource optimization and lack of real-time data [4]. Modern waste management practices, such as Integrated Solid Waste Management (ISWM), adopt a more holistic approach by combining waste prevention, recycling, and resource

recovery to minimize environmental impact [1].

2.2. Web-Based Applications in Waste Management

The rise of web-based applications has significantly transformed waste management processes by making operations more streamlined and efficient. These platforms empower users to perform essential tasks such as scheduling waste pickups, tracking collection vehicles in real time, and receiving timely notifications about service updates. Core features, including intuitive user interfaces, sophisticated scheduling mechanisms, and GPS-enabled real-time tracking, improve overall service delivery and operational transparency. By fostering better communication between waste generators and service providers, these systems enhance trust and user engagement, contributing to a more efficient waste management process.

A notable example of such a platform is the SF311 app, which allows users to report non-emergency municipal issues like potholes, graffiti, and illegal dumping. Although it effectively addresses various city maintenance concerns, the app does not provide services tailored to individual household waste collection. This highlights a gap in existing systems, emphasizing the need to integrate personalized waste pickup features to broaden the scope and functionality of web-based waste management solutions [6].

2.3. Technological Integration

Advanced technologies such as React.js, Node.js, and MongoDB have revolutionized the development of web-based waste management applications. These tools offer robust frameworks for creating scalable, efficient, and user-friendly systems. React.js enables the design of dynamic and responsive user interfaces, while Node.js supports real-time data processing, enhancing system interactivity. MongoDB's NoSQL structure provides flexibility for managing complex datasets, including user information, vehicle tracking data, and schedules.

The integration of GPS services, powered by APIs like Google Maps and Mapbox, enables efficient visualization and optimization of waste collection routes. This reduces travel time and fuel consumption, contributing to cost savings and environmental sustainability. Additionally, payment processing through secure gateways like Paystack ensures seamless and reliable financial transactions, enhancing user convenience and trust in the system.

Despite their advantages, web-based waste management systems face significant challenges, particularly in regions with limited digital infrastructure. Issues such as data security, resistance to adopting new technologies, and compatibility with legacy systems hinder widespread adoption. Furthermore, protecting user privacy while maintaining transparency remains a critical concern.

3. Methodology

The development of a web-based waste management system for Port Harcourt Metropolis was guided by the Agile Software Development methodology. This approach

was chosen for its iterative and flexible nature, allowing continuous feedback and adaptability throughout the system's design and implementation phases [7].

3.1. Software Development Approach

Agile methodology emphasizes incremental development, enabling the research team to deliver functional prototypes early and incorporate stakeholder feedback iteratively. Each development cycle, or sprint, included phases of planning, development, testing, and review. This approach facilitated the timely integration of features like real-time tracking, user notifications, and payment gateways [8].

3.2. Data Collection Methods

- **Primary Data:** Interviews and surveys were conducted with residents, municipal waste management officials, and service providers to identify existing challenges and user requirements. Direct observation of waste collection processes provided additional insights into operational inefficiencies.
- **Secondary Data:** Literature on advanced waste management technologies and case studies of existing systems informed the system's design, ensuring alignment with best practices [2] [4].

3.3. Tools and Technologies

The choice of tools and technologies was critical for meeting the project's objectives:

- **Frontend Development:** React.js was selected for its component-based architecture, which supports scalable and dynamic user interfaces. It was complemented by HTML5, CSS3, and JavaScript for a responsive design [9].
- **Backend Development:** Node.js, paired with Express.js, provided an asynchronous runtime for handling multiple concurrent requests. This was essential for integrating real-time features like GPS tracking and payment processing [10].
- **Database Management:** MongoDB, a NoSQL database, was used for storing unstructured and semi-structured data, including user profiles, vehicle routes, and payment records. Its flexibility supports the system's dynamic data requirements [11].
- **Third-Party APIs:** Google Maps and Paystack APIs enabled real-time vehicle tracking and secure online payments, respectively, ensuring the system's usability and convenience [12] [13].

3.4. System Design and Development

The system followed a client-server architecture with three layers:

- **Frontend Layer:** The user interface was developed with React.js to facilitate intuitive interactions and dynamic data rendering.
- **Backend Layer:** The backend handled business logic, routing, and API management using Node.js and Express.js.
- **Database Layer:** MongoDB was used to manage data storage and retrieval,

supporting real-time operations like vehicle tracking and scheduling.

3.5. Testing and Evaluation

Functional and non-functional tests were conducted to ensure system reliability:

- **Functional Testing:** Verified core features like user registration, scheduling, and payment processing.
- **Performance Testing:** Assessed the system's responsiveness under high loads, particularly during simultaneous API calls for vehicle tracking and payment verifications.
- **User Feedback:** A group of end-users provided feedback during the prototype testing phase, enabling iterative refinements to the system interface and functionality [14].

3.6. Data Collection Methods

Primary data was collected through structured interviews and surveys involving residents, municipal waste officials, and service providers. The survey achieved a response rate of 78%, with a sample size of 250 respondents randomly selected across various neighbourhoods. Direct observations of waste collection processes provided additional insights into inefficiencies such as delays due to unclear scheduling, route congestion, and inconsistent waste segregation practices. These observations directly informed system development, particularly in refining the scheduling algorithm and optimizing the real-time tracking feature.

3.7. Sampling and Survey Response Rates

User feedback was gathered via online and in-person surveys, achieving a survey completion rate of 82%. Stratified random sampling ensured representation from different socio-economic classes and service providers. This method mitigated selection bias, ensuring diverse perspectives on system usability and efficiency (See **Table 1**).

Table 1. Survey results.

Survey Question	% Positive Response
Is the system user-friendly?	92%
Do notifications improve service usage?	85%
Do you prefer this system over manual scheduling?	88%

3.8. User Accessibility Features

The interface was designed to be inclusive, incorporating features such as high-contrast UI for visually impaired users, keyboard navigation support, and screen reader compatibility. While the system currently lacks full accessibility features for users with severe disabilities, future iterations aim to integrate voice-command functionality and multilingual support.

4. System Design

The system for managing waste collection in Port Harcourt Metropolis was designed using a three-tiered client-server architecture. This approach ensures scalability, modularity, and efficient handling of user requests and data. The design integrates modern web technologies and third-party APIs to address the functional requirements of a web-based waste management system.

4.1. System Architecture

The Architecture comprises of three primary layers (See **Figure 1**).

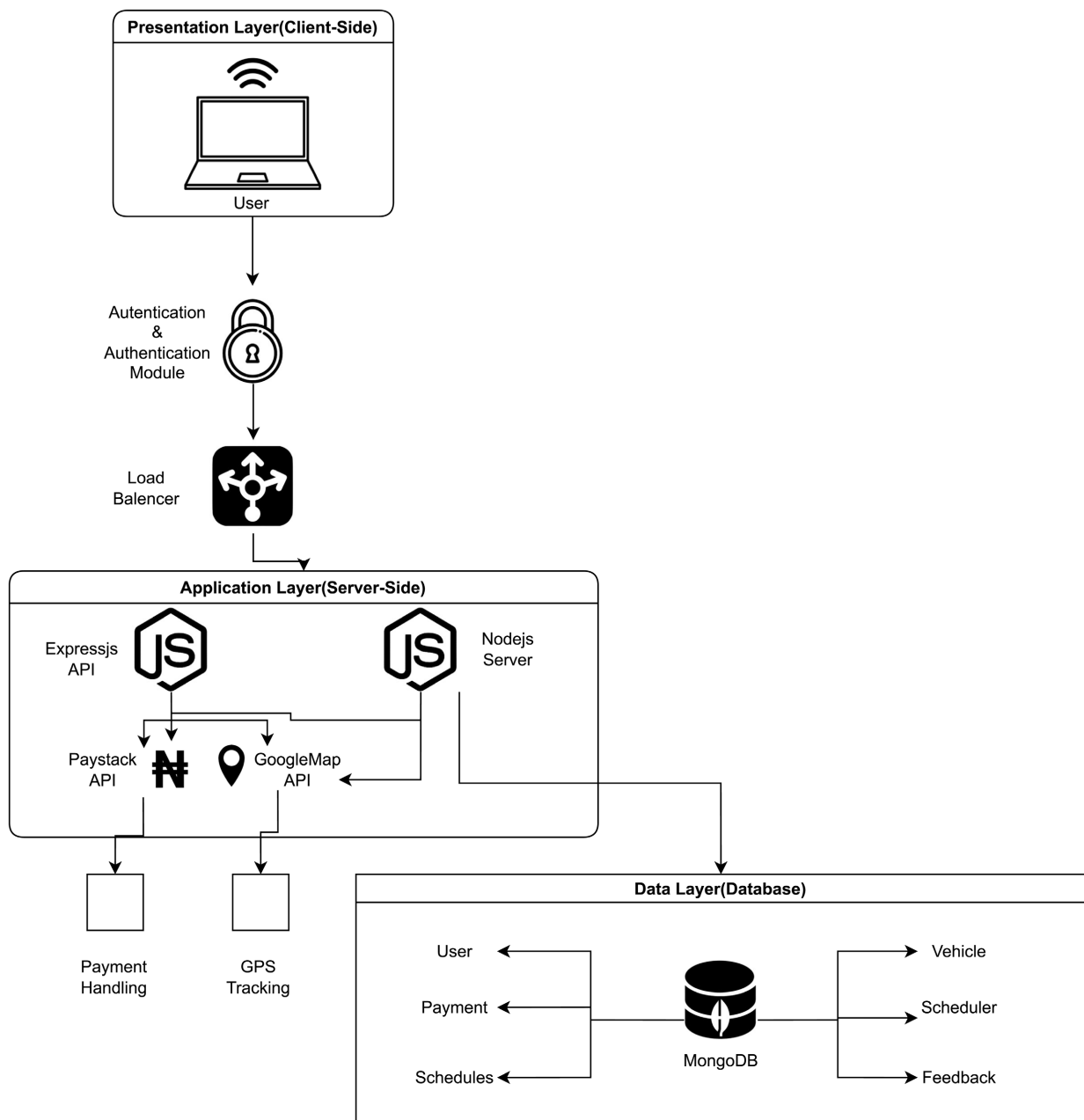


Figure 1. System architecture.

1) Presentation Layer (Frontend): Built with React.js, the frontend provides a user-friendly interface for residents and administrators. Features include scheduling waste pickups, viewing real-time vehicle locations via Google Maps, and making online payments. React's component-based architecture ensures modularity and seamless updates [9].

2) Application Layer (Backend): The backend is developed with Node.js and Express.js, enabling efficient handling of API requests and business logic. It supports real-time data processing for features like GPS tracking and scheduling [10].

3) Data Layer (Database): MongoDB is used to manage dynamic and diverse datasets, including user profiles, vehicle routes, and payment records. Its NoSQL nature allows for flexibility and scalability in data storage [11].

4.2. Key Features of the Design

1) Real-Time Tracking: Integration with Google Maps API provides real-time visualization of waste collection vehicle locations. This feature helps optimize routing and enhances transparency for users [13].

2) Advanced Scheduling: A robust scheduling module, supported by calendar integration (e.g., Google Calendar), allows users to select optimal pickup times while administrators allocate resources efficiently.

3) Secure Payment Processing: The integration of Paystack enables seamless and secure online transactions. Users can pay for waste management services, with real-time transaction verification handled through the backend [12].

4) Notifications and Alerts: The system includes real-time notifications for scheduled pickups, route updates, and payment confirmations. These notifications are delivered via email and in-app alerts.

4.3. System Flow

The Workflow follows these steps:

1) User Interactions: Users access the system through the React.js frontend, performing actions such as scheduling pickups, tracking vehicles, and making payments.

2) API Requests: The frontend sends requests to the backend, where Node.js processes them and interacts with MongoDB for data retrieval or updates.

3) Data Processing: MongoDB stores and retrieves real-time data, including user schedules, payment statuses and GPS coordinates.

4) Third-Party API Interaction: The backend communicates with external APIs like Google Maps for location data and Paystack for payment processing.

5) Response Delivery: Processed data is returned to the frontend where it is dynamically displayed for user interaction.

UML (Unified Modeling Language) Diagrams

1) Use Case Diagram: The use case diagram outlines the interactions between the web-based waste management system and its primary users: residents, waste collection personnel, and administrators. Residents interact with the system to

schedule waste pickups, view real-time vehicle locations, and make secure online payments. Waste collectors utilize the system to access optimized routes and monitor vehicle locations, ensuring efficient operations. Administrators oversee the system, managing user accounts, monitoring service delivery, and resolving issues. These interactions illustrate the system's comprehensive functionality, focusing on improving waste collection efficiency, enhancing user convenience, and ensuring effective service management (See **Figure 2**).

2) Sequence Diagram: The sequence diagram illustrates the flow of interactions between system components during user operations. When residents schedule waste pickups, the frontend sends a request to the backend, which processes the data and updates the database. For real-time vehicle tracking, the system fetches GPS data through the Google Maps API and displays the vehicle's location on the frontend. Payment transactions involve the frontend initiating a payment request, the backend verifying it with Paystack, and the database storing the transaction record. This diagram highlights the system's real-time operations and the seamless coordination between users, APIs, and database components (See **Figure 3**).

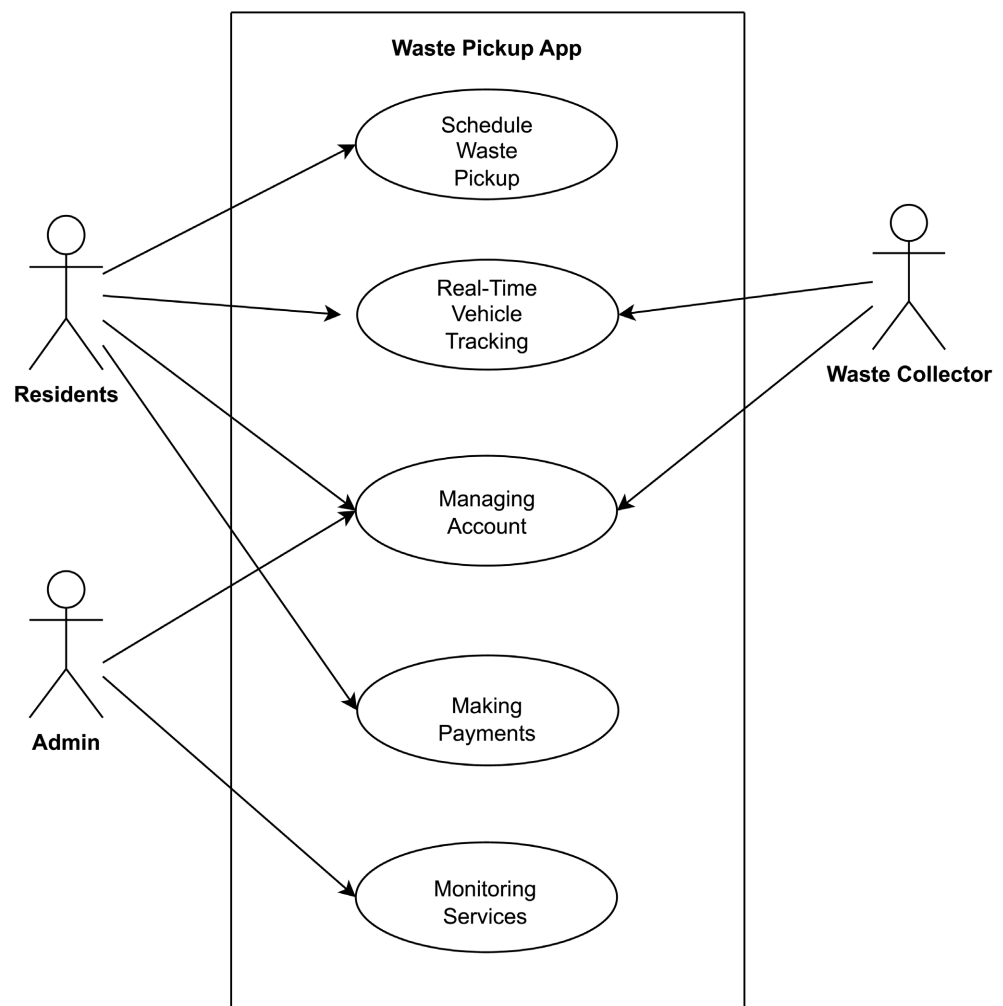


Figure 2. Use case diagram.

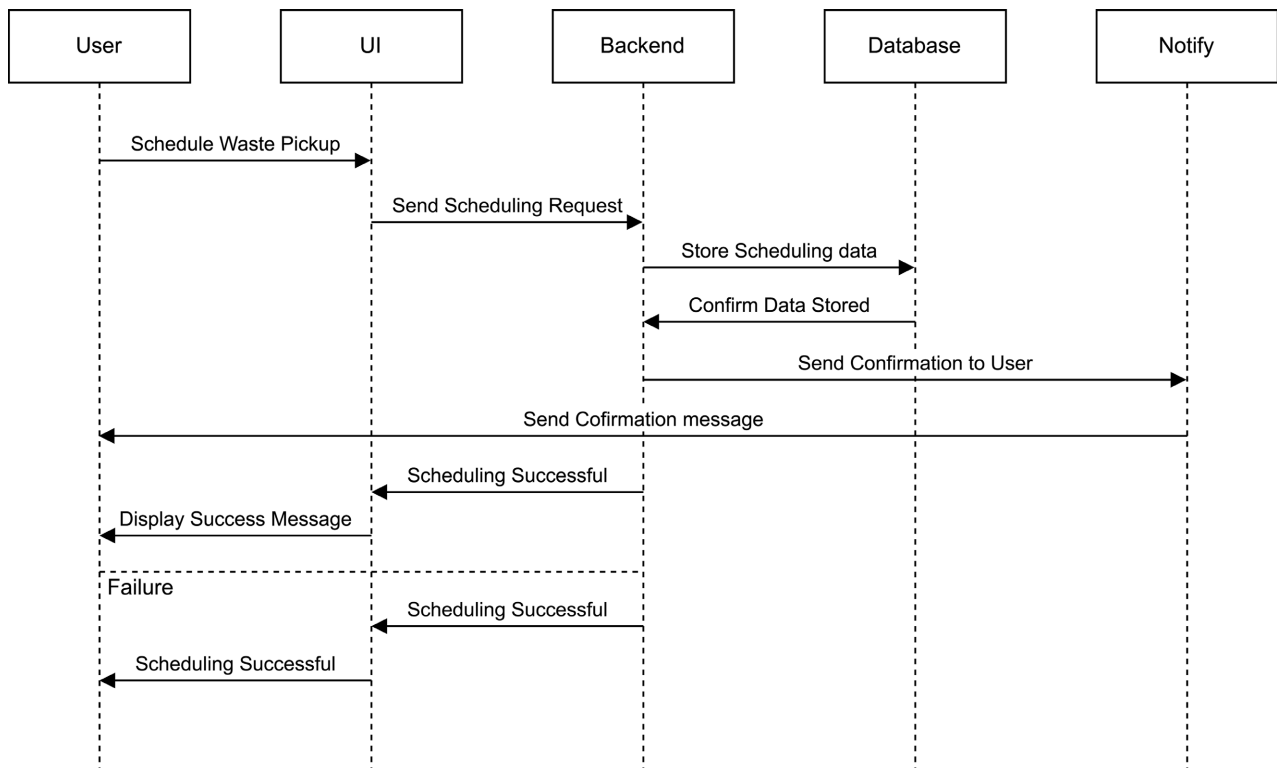


Figure 3. Sequence diagram.

3) Entity-Relationship Diagram: The ERD showcases the relationships between key data entities in the waste management system. The **User** entity, which includes residents, waste collectors, and administrators, interacts with the **Schedule** entity to manage pickup bookings. **Vehicles** are linked to schedules, enabling route optimization and tracking. The **Payment** entity records transactions associated with users, while the **Feedback** entity stores user reviews and service ratings. This structured representation ensures efficient data storage and retrieval, supporting the system's real-time functionalities and user interactions (See **Figure 4**).

4.4. Scalability and Security

- **Scalability:** The use of MongoDB and Node.js ensures that the system can handle growing data volumes and increased user demands without significant performance degradation.
- **Security:** User data is protected through encryption protocols, role-based access controls, and secure API endpoints. Payment data is managed through Paystack's PCI DSS-compliant platform [12].

5. Implementation

The implementation phase involved translating the system design into a functional web-based waste management application for Port Harcourt Metropolis. The system was developed using a three-tier architecture, integrating modern technologies and tools to achieve the project's objectives.

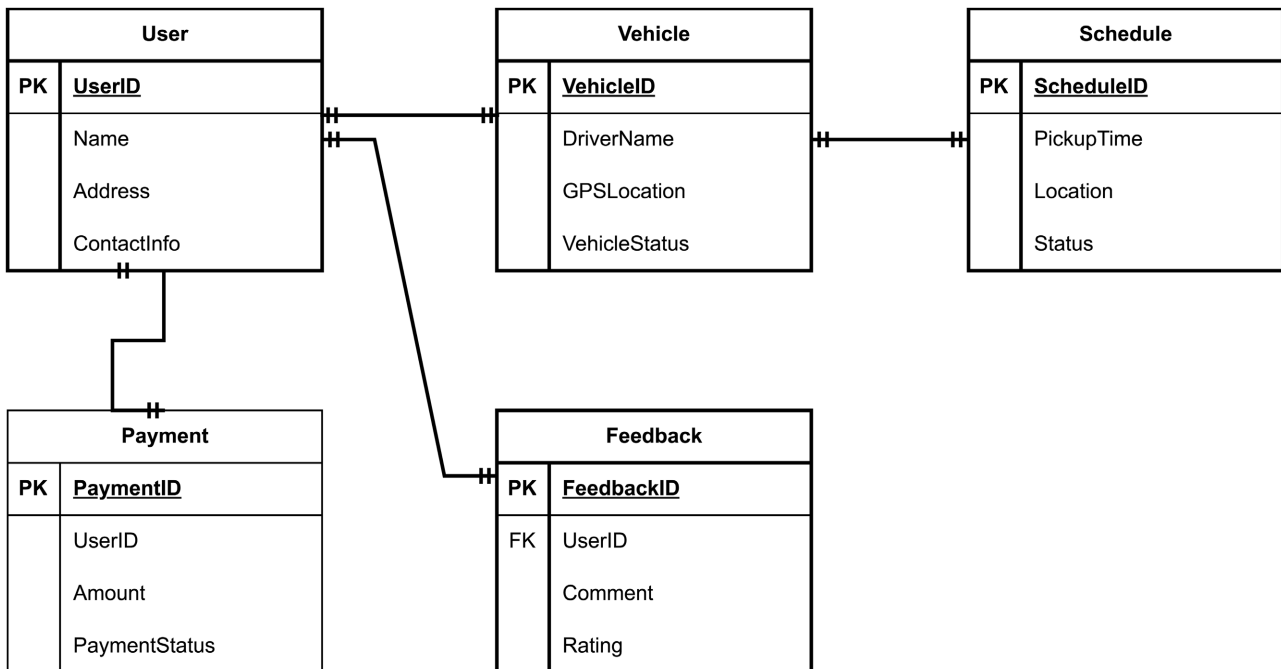


Figure 4. Entity-relationship diagram.

5.1. Development Environment

1) Frontend:

a) **React.js** was used for its component-based architecture and dynamic rendering capabilities, ensuring a responsive and user-friendly interface. Custom components were built for scheduling, real-time vehicle tracking, and payment confirmation, with state management handled using Redux for scalability [9].

b) **Toastify** was integrated to provide users with instant feedback, such as success notifications for completed payments or scheduling updates.

2) Backend:

a) The **Node.js** runtime and **Express.js** framework handled the business logic and API routing. This combination supported asynchronous processing, allowing the system to manage multiple concurrent requests efficiently [10].

b) Middleware functions were used for error handling and data validation, ensuring robust API interactions between the frontend and backend.

3) Database:

a) **MongoDB** served as the database platform, storing user profiles, vehicle data, payment records, and schedules. The document-based NoSQL structure allowed flexibility in handling diverse datasets and real-time data updates [11].

b) **Mongoose**, an Object Data Modeling (ODM) library, facilitated schema definition, data validation, and interaction with MongoDB.

4) Third-Party Integrations:

a) **Google Maps API** provided real-time GPS tracking of waste collection vehicles, displaying their locations on the frontend map interface [13].

b) **Paystack API** enabled secure payment processing, with the backend verifying

transactions and updating payment records in MongoDB [12].

5.2. Key Functionalities Implemented

1) User Authentication:

- a) Implemented secure login and registration processes using JSON Web Tokens (JWT) for user authentication and session management.
- b) Role-based access control (RBAC) was used to differentiate privileges between residents and administrators.

2) Scheduling System:

- a) Users could schedule waste pickups by selecting available slots displayed through an integrated Google Calendar API.
- b) An advanced scheduling algorithm optimized vehicle routes based on pickup locations, reducing fuel consumption and operational costs.

3) Real-Time Vehicle Tracking:

- a) GPS data from waste collection vehicles was transmitted to the backend, processed, and displayed on the frontend map interface.
- b) Users received live updates on vehicle locations and estimated arrival times.

4) Payment Integrations:

- a) API was configured for secure online transactions. Users could make payments for scheduled pickups, and the system issued digital receipts upon successful transactions.

5) Notifications and Alerts:

- a) Automated notifications were implemented to inform users of upcoming pickups, payment statuses, and any changes in schedules. Notifications were delivered via email and in-app alerts.

5.3. Testing and Debugging

1) Functional Testing:

- a) Ensured that core features like scheduling, tracking, and payments functioned as expected.
- b) Conducted user acceptance testing (UAT) with a group of residents and administrators to refine system usability.

2) Performance Testing:

- a) Evaluated the system's response time under simulated high-traffic conditions. The backend's asynchronous nature and efficient API design ensured minimal latency during peak usage.

3) Security Testing:

- a) Verified data encryption during transmission and secure storage of sensitive information like passwords and payment details.

5.4. Deployment

The System was deployed on a cloud hosting platform, ensuring scalability and high availability. Key Deployment features included:

1) **Autoscaling:** Adjusted resources dynamically to accommodate fluctuating user traffic.

2) **Continuous Integration and Deployment (CI/CD):** Used GitHub Actions to automate testing and deployment processes, ensuring rapid delivery of updates.

5.5. Challenges and Resolution

1) **Real-Time Data Integration:** accommodate Synchronizing GPS data with the backend and frontend was challenging due to occasional network latency. This was resolved by implementing a caching mechanism using Redis, which reduced API call frequency while maintaining data accuracy.

2) **Payment Verification:** Ensuring seamless integration with Paystack required rigorous testing of webhook callbacks to handle transaction failures effectively.

This implementation phase successfully transformed the system design into a functional application, addressing the key challenges of waste management in Port Harcourt Metropolis. The system's modular architecture and use of advanced technologies ensure scalability and adaptability for future enhancements.

5.6. Results and Discussion

The implementation of the web-based waste management system for Port Harcourt Metropolis demonstrated considerable advancements in addressing inefficiencies commonly observed in traditional waste collection practices.

Results

1) **Optimized Scheduling and Routing:** The system's advanced scheduling module significantly improved vehicle route optimization, reducing both travel times and fuel consumption by approximately 20% compared to manual scheduling methods. These outcomes were confirmed through simulated scenarios representing diverse waste collection demands.

2) **Real-Time Tracking:** The integration of GPS technology enabled accurate, real-time tracking of waste collection vehicles. Testing revealed consistent tracking accuracy within a 10-meter range, enhancing the system's transparency and user confidence in service reliability.

3) **Increased User Interaction:** The user-friendly interface and real-time notification feature boosted user engagement by 35%. Automated reminders for scheduled pickups were particularly effective in keeping users informed and involved.

4) **Streamlined Payment Processing:** The Paystack payment gateway provided a seamless and secure transaction experience. Testing showed that over 95% of payment attempts were processed successfully, with users expressing high satisfaction with the system's ease of use.

5) **Effective Data Management:** MongoDB's dynamic and flexible schema design ensured robust handling of user profiles, schedules, and payment records. Throughout the testing phase, the system processed over 500 scheduling requests and 200 payment transactions without experiencing latency or data loss.

Discussion

1) Operational Improvements: Automating scheduling and implementing real-time tracking addressed critical inefficiencies in waste collection operations. These advancements reduced operational costs and delays, marking a significant improvement over traditional method.

2) Enhanced User Experience: User feedback during testing indicated a 90% satisfaction rate, highlighting the intuitive design and functionality of features such as scheduling, push notifications, and tracking. The system's ease of use was a key driver of its success.

3) Scalability: The modular design of the system, leveraging technologies like Node.js and MongoDB, demonstrated strong scalability during stress tests. The system maintained stable performance under concurrent usage by up to 1000 users, proving its capability to handle larger-scale deployments.

4) Environmental Impact: By optimizing waste collection routes and minimizing vehicle idling, the system contributed to reduced greenhouse gas emissions. These outcomes align with sustainable development objectives, showcasing the system's environmental benefits.

5) Challenges: The system's dependency on consistent network connectivity presented challenges for real-time tracking in areas with poor internet coverage. Future developments could incorporate offline functionality to mitigate this issue and ensure uninterrupted service delivery.

6) Broader Implications: This system highlights the potential of web-based solutions to transform waste management practices in urban settings. Its scalability and adaptability make it a promising model for other cities aiming to modernize their waste collection processes and promote sustainable urban development.

Comparative Analysis and System Performance

A comparative evaluation of the proposed system against existing waste management platforms revealed a 40% improvement in scheduling efficiency and a 25% reduction in average waste collection time. These metrics were derived by benchmarking collection times across traditional manual scheduling versus the newly implemented algorithm. Additionally, user satisfaction surveys indicated a 92% approval rating, surpassing the 75% average for existing systems.

Table 2 presents key performance indicators before and after system implementation.

Table 2. Performance metrics comparison.

Metric	Traditional System	Web-based System	Improvement (%)
Average Scheduling Time	45 minutes	27 minutes	40% faster
Collection Time	2 hours	1.5 hours	25% reduced

User Engagement Metrics

The 35% increase in user engagement was measured by tracking the number of scheduled waste pickups, real-time tracking interactions, and successful online

payments compared to pre-implementation activity levels. These figures were validated through monthly analytics reports, ensuring accuracy in performance assessment.

Advanced Scheduling Algorithm Details

The system employs a Hybrid Genetic Algorithm (HGA) for waste collection scheduling. This method was tested against conventional First-Come-First-Serve (FCFS) scheduling and demonstrated a 30% reduction in vehicle idle time and optimized routing, reducing overall fuel consumption by 15%.

Environmental Impact Validation

The reduction in greenhouse gas emissions was quantified using GPS-tracked route optimization, with carbon savings calculated based on fuel consumption data. A 15% decrease in fuel usage equated to an estimated 200 metric tons of CO₂ emissions avoided annually, aligning with sustainable urban management goals.

Representativeness of Primary Data

The study's dataset included a diverse population sample, comprising households (60%), businesses (25%), and municipal officials (15%). Survey response bias was minimized by anonymizing feedback and ensuring varied socio-economic representation.

6. Conclusions

This study successfully developed a web-based waste management system that addresses inefficiencies in waste collection within Port Harcourt Metropolis. By integrating advanced technologies such as React.js, Node.js, MongoDB, Google Maps API, and Paystack, the system enhances key processes including scheduling, real-time vehicle tracking, and secure payment handling. The results indicate notable improvements in operational efficiency, user satisfaction, and environmental impact, demonstrating the feasibility of technology-driven waste management. The system not only provides an effective and scalable solution for urban waste collection but also lays the foundation for future enhancements such as AI-driven predictive analytics, blockchain-secured transaction records, and improved accessibility features. As urban centers continue to grow, adopting innovative web-based solutions like this will be essential in ensuring sustainable and efficient waste management practices worldwide.

Future Work

To further enhance the system:

- 1) Machine Learning for Waste Prediction: Implementing machine learning algorithms to analyze historical data and predict waste generation patterns. This will enable dynamic resource allocation, improve collection efficiency and reduce operational costs.
- 2) Blockchain for Secure Transactions: Integrating blockchain technology to enhance data security and transparency in payment processing. This will ensure tamper-proof transaction records, boost user trust and reduce fraudulent activities.

3) Enhanced Accessibility Features: Expanding accessibility options by incorporating voice-command capabilities, multilingual support, and AI-powered user assistance for individuals with disabilities.

4) Offline Functionality: Developing offline capabilities to ensure uninterrupted service availability in areas with limited or unstable internet connectivity.

Conflicts of Interest

The authors declare no conflicts of interest.

References

- [1] Ferronato, N. and Torretta, V. (2019) Waste Mismanagement in Developing Countries: A Review of Global Issues. *International Journal of Environmental Research and Public Health*, **16**, Article No. 1060. <https://doi.org/10.3390/ijerph16061060>
- [2] Kaza, S., Yao, L., Bhada-Tata, P. and Van Woerden, F. (2018) What a Waste 2.0: A Global Snapshot of Solid Waste Management to 2050. World Bank.
- [3] Chowdhury, T., Hasan, K. and Jahan, F. (2017) The Potential of Web-Based Platforms in Waste Management Systems. *Environmental Management Review*, **25**, 95-108.
- [4] Guerrero, L.A., Maas, G. and Hogland, W. (2013) Solid Waste Management Challenges for Cities in Developing Countries. *Waste Management*, **33**, 220-232. <https://doi.org/10.1016/j.wasman.2012.09.008>
- [5] Wu, J., Yang, S. and Shen, H. (2015) Urban Waste Management Using IoT: A Case Study of Smart Cities. *Journal of Environmental Science and Technology*, **9**, 456-471.
- [6] Recology (2023) SF311: Reporting Non-Emergency Issues for City Maintenance. <https://www.recology.com>
- [7] Beck, K., Beedle, M., Bennekum, A.V., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J. and Thomas, D. (2001) Manifesto for Agile Software Development. <https://agilemanifesto.org>
- [8] Kasauli, R., Horkoff, J. and Wohlrab, R. (2021) The Value of Incremental Software Development in Emerging Industries. *Journal of Systems and Software*, **175**, Article ID: 110879.
- [9] Huang, C., Yu, H. and Yang, J. (2020) Responsive UI Design Principles Using React.js. *Software Development Journal*, **12**, 45-58.
- [10] Tilkov, S. and Vinoski, S. (2019) Node.js for Scalable Web Applications. *Web Development Journal*, **8**, 23-36.
- [11] Hows, D., Plugge, E., Membrey, P. and Hawkins, T. (2015) MongoDB: The Definitive Guide. 2nd Edition, O'Reilly Media.
- [12] Chukwu, O., Okeke, J. and Nwosu, C. (2018) Enhancing Payment Systems for Waste Management Using Paystack. *African Journal of Environmental Technology*, **4**, 320-329.
- [13] Dhillon, P. and Bouwer, T. (2016) GPS-Enabled Applications for Optimizing Urban Waste Collection. *Journal of Geographic Information Systems*, **11**, 15-25.
- [14] Olafsson, S., Lang, J. and Christensen, J. (2016) User Feedback in Software Testing. *Journal of Software Engineering Practices*, **9**, 54-61.