

Optimization of Urban Services using Priority Queue



A

Project Report

Submitted in partial fulfillment of the requirement for the award of degree of

Bachelor of Technology

In

Information Technology

Submitted to

**RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA,
BHOPAL (M.P.)**

Guided By

Prof. Prashant Lakkadwala

Submitted By

Jahn timer Mandloi(0827IT211048)

Jayshree Dave(0827IT211050)

Mahee Dubey(0827IT211065)

Minakshi Soni(0827IT211069)

**DEPARTMENT OF INFORMATION TECHNOLOGY
ACROPOLIS INSTITUTE OF TECHNOLOGY & RESEARCH,
INDORE (M.P.) 452020
2024-2025**

Declaration

We hereby declared that the work, which is being presented in the project entitled **Optimization of UrbanServices using Priority Queue** partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology**, submitted in the department of Information Technology at **Acropolis Institute of Technology and Research, Indore** is an authentic record of our own work carried under the supervision of **Prof. Prashant Lakkadwala**. We have not submitted the matter embodied in this report for the award of any other degree.

Jahnavi Mandloi(0827IT211048)

Jayshree Dave(0827IT211050)

Mahee Dubey(0827IT211065)

Minakshi Soni(0827IT211069)

Prof. Prashant Lakkadwala

Supervisor

Project Approval Form

We hereby recommend that the project **Optimization of UrbanServices using Priority Queue** prepared under our supervision by Jahnavi Mandloi(0827IT211048), Jayshree Dave(0827IT211050), Mahee Dubey(0827IT211065), Minakshi Soni(0827IT211069) be accepted in partial fulfillment of the requirement for the degree of Bachelor of Technology in Information Technology.

Prof. Prashant Lakkadwala

Supervisor

Recommendation concurred in 2024-2025

Prof. Vishal Trivedi

Project Incharge

Prof. Deepak Singh Chouhan

Project Coordinator

Acropolis Institute of Technology & Research

Department of Information Technology



Certificate

The project work entitled **Optimization of UrbanServices using Priority Queue** submitted by Jahnavi Mandloi(0827IT211048), Jayshree Dave(0827IT211050), Mahee Dubey(0827IT211065), Minakshi Soni(0827IT211069) is approved as partial fulfillment for the award of the degree of Bachelor of Technology in Information Technology by Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.).

Internal Examiner

Name:.....

Date:/.../.....

External Examiner

Name:

Date:/.../.....

Acknowledgement

With boundless love and appreciation, we would like to extend our heartfelt gratitude and appreciation to the people who helped us to bring this work to reality. We would like to have some space of acknowledgement for them.

Foremost, our would like to express our sincere gratitude to our supervisor,

Prof. **Prashant Lakkadwala** whose expertise, consistent guidance, ample time spent and consistent advice that helped us to bring this study into success.

To the project in-charge **Prof. Vishal Trivedi** and project coordinator **Prof. Deepak Singh Chouhan** for their constructive comments, suggestions, and critiquing even in hardship.

To the honorable **Prof. Prashant Lakkadwala**, Head, Department of Information Technology for his favorable responses regarding the study and providing necessary facilities.

To the honorable **Dr. S.C. Sharma**, Director, AITR, Indore for his unending support, advice and effort to make it possible.

Finally, we would like to pay our thanks to faculty members and staff of the Department of Information Technology for their timely help and support.

We also like to pay thanks to our **parents** for their eternal love, support and prayers without them it is not possible.

Jahnavi Mandloi(0827IT211048)

Jayshree Dave(0827IT211050)

Mahee Dubey(0827IT211065)

Minakshi Soni(0827IT211069)

Abstract

The Urban Services project presents a smart city initiative that leverages advanced digital technologies to optimize urban resource management. This project addresses critical urban challenges in waste management, water conservation, and citizen engagement. The primary goal is to streamline operations, reduce resource wastage, and promote sustainable practices in urban environments.

The waste management component prioritizes different waste types (e.g., organic, hazardous, recyclable, and electronic) to improve response times and reduce environmental impact. Water management is enhanced through real-time monitoring to detect leaks and enable smart irrigation systems. Citizen engagement is facilitated through a user-friendly platform that allows residents to report issues, provide feedback, and access city services, thereby enhancing communication with city authorities.

Java-based technologies serve as the backbone of this integrated solution, with a priority queue system used to address higher-priority tasks first, ensuring efficient resource allocation. The project includes a real-time dashboard for city officials and administrators to monitor and manage city services effectively. Ultimately, the Urban Services project aims to improve urban living conditions by implementing data-driven approaches to governance and community involvement.

Table of Content

| | |
|------------------------------------|------------|
| Declaration..... | II |
| Project Approval Form | III |
| Acknowledgement | V |
| Abstract..... | VI |
| List of Figures..... | X |
| List of Tables | XI |

Chapter 1: Introduction

| | |
|--------------------------------------|---|
| 1.1 Rationale | 1 |
| 1.2 Existing System | 1 |
| 1.3 Problem Formulation | 2 |
| 1.4 Proposed System..... | 2 |
| 1.5 Objectives | 3 |
| 1.6 Contribution of the Project..... | 3 |
| 1.6.1 Market Potential | 3 |
| 1.6.2 Innovativeness..... | 4 |
| 1.6.3 Usefulness | 4 |
| 1.7 Report Organization..... | 4 |

Chapter 2: Requirement Engineering

| | |
|---|---|
| 2.1 Feasibility Study (Technical, Economical, Operational | 5 |
| 2.2 Requirement Collection | 5 |
| 2.2.1 Discussion | 5 |
| 2.2.2 Requirement Analysis | 5 |

| | |
|---|---------|
| 2.3 Requirements | |
| 2.3.1 Functional Requirements | 5 |
| 2.3.2 Nonfunctional Requirments | 6 |
| 2.4 Hardware & Software Requirements | 6 |
| 2.4.1 Hardware Requirement (Developer & End User)..... | 6 |
| 2.4.2 Software Requirement (Developer & End User) | 6 |
| 2.5 Use-case Diagrams..... | 7 |
| 2.5.1 Use-case Descriptions | 8 |
| Chapter 3: Analysis & Conceptual Design & Technical Architecture | |
| 3.1 Technical Architecture | 9 |
| 3.2 Sequence Diagrams | 9 |
| 3.3 Class Diagrams | 10 |
| 3.4 DFD..... | 11 |
| 3.5 User Interface Design | 11 |
| 3.6 Data Design..... | 12 |
| 3.6.1 Schema Definitions | 12..... |
| 3.6.2 E-R Diagram | 19 |
| Chapter 4: Implementation & Testing | |
| 4.1 Methodology | 20 |
| 4.1.1 Proposed Algorithm | 20 |
| 4.2 Implementation Approache..... | 21 |
| 4.2.1 Introduction to Languages, IDEs Tools and Technoloies..... | 21 |
| 4.3 Testing Apporaches | 23 |
| 4.3.1 Unit Testing | 23 |

| | |
|--|-----------|
| a. Test Cases..... | 23-27 |
| Chapter 5: Results & Discussion | |
| 5.1 User Interface Representation | 28 |
| 5.1.1 Brief Dessiontion of Various Modules | 28 |
| 5.2 Snapshot of System with Brief Description..... | 30-35 |
| 5.3 Database Description | 36 |
| 5.3.1 Snapshot of Database Tables with Brief Description | 36-38 |
| Chapter 6. Conclusion & Future Scope | |
| 6.1 Conclusion | 39 |
| 6.2 Future Scope | 39 |
| REFERENCES..... | 40 |
| Appendix A: Project Synopsis | |
| Appendix B: Guide Interaction Report | |
| Appendix C: User Manual | |

List of Figures

| | |
|---|----|
| Fig.1.1 Block diagram of Optimization of UrbanServices using Priority Queue | 3 |
| Fig.2.1 Use-case diagram..... | 7 |
| Fig.3.2 Sequence diagram | 9 |
| Fig.3.3 Class diagram..... | 10 |
| Fig.3.4 DFD diagram | 11 |
| Fig.3.5 UrbanServices Dashboard..... | 12 |
| Fig.3.6.2 ER Diagram | 19 |
| Fig.5.2(a) UrbanServices Platform:Dashboards | 30 |
| Fig.5.2(b) UrbanServices Platform:Registration and Login | 31 |
| Fig.5.2(c) Waste Management: Report & Feedback Forms..... | 32 |
| Fig.5.2(d) Water Management: Report & Feedback Forms | 33 |
| Fig.5.2(e) Citizen Engagement: Report & Feedback Forms..... | 34 |
| Fig.5.2(f) Waste Management:View Issues and Feedback | 35 |

List of Tables

| | |
|--|-------|
| Table I Test Cases for Optimization of UrbanServices using Priority Queue..... | 23-27 |
| Table II User and Admin Registration | 36 |
| Table III User Report Issue and Feedback for Waste Management..... | 37 |
| Table IV User Report Issue for Water Management | 37 |
| Table V User feedback for Water Management and report issue for Citizen..... | 38 |
| Table VI User Feedback for Citizen Engagement | 38 |

INTRODUCTION

1.1 Rationale

The rapid urbanization of cities has led to an increased demand for efficient public services, especially for waste and water management. Current methods for addressing urban issues often lack transparency, speed, and user involvement, leading to dissatisfaction among citizens and challenges for city administrators. The UrbanServices platform aims to bridge this gap by enabling citizens to report issues, give feedback, while administrators manage and resolve these complaints effectively.

With the rapid urbanization of cities worldwide, municipalities are facing growing challenges in managing essential civic services such as waste disposal, water supply, road maintenance, and sewage management. Inefficient systems for reporting and addressing these civic issues can result in delays, dissatisfaction, and a lack of accountability, diminishing citizens trust in local authorities.

Traditionally, citizens report issues through manual processes, such as phone calls or physical visits to municipal offices. These methods are not only time-consuming but also prone to errors, leading to delays and unaddressed complaints. In addition, there is no centralized system that allows citizens to track the progress of their complaints in real-time, and many municipalities lack mechanisms for prioritizing issues based on severity.

To address these challenges, the UrbanServices aims to develop a web-based platform that will enable citizens to report issues efficiently, and provide feedback. This system will enhance transparency, improve communication between citizens and municipal authorities, and ensure timely resolution of civic issues.

1.2 Existing System

Traditional systems for managing civic issues are mostly offline, slow, and lack proper tracking mechanisms. Reporting issues often requires citizens to visit government offices or make numerous phone calls, resulting in delays and unaddressed complaints. Additionally, current systems lack a way for citizens to monitor the status of their complaints, which leads to a lack of transparency and trust in the system.

Currently, most municipalities rely on outdated manual systems for reporting civic issues, which include:

- Phone calls or in-person visits to municipal offices for complaints.
- Lack of real-time tracking of complaints, making it difficult for citizens to track the status of reported issues.
- No mechanisms for prioritization of issues based on severity, leading to delays in addressing urgent matters.
- Minimal communication from authorities regarding the resolution status, resulting in frustration and disengagement from citizens.
- These systems do not provide adequate support for municipalities to manage large volumes of issues effectively or keep citizens informed, which often leads to poor service delivery and lack of trust.

1.3 Problem Formulation

The primary problems identified in the current systems are:

- **Inefficient Communication:** Citizens face difficulty in reporting issues, and there is no effective mechanism for receiving timely updates from authorities.
- **Lack of Transparency:** Without real-time tracking and status updates, citizens are left in the dark about the progress of their complaints.
- **Ineffective Prioritization:** There is no automated way to prioritize urgent issues (e.g., water supply disruptions, road accidents), leading to a lack of prompt responses to critical problems.

These challenges need to be addressed to ensure more effective management of civic issues, improved citizen satisfaction, and better allocation of municipal resources.

1.4 Proposed System

The UrbanServices is designed as a web-based platform with the following features:

Issue Reporting: Citizens can report issues related to public services, including descriptions, locations, and attachments (e.g., images).

Feedback Mechanism: After an issue is resolved, citizens can provide feedback, which helps municipal authorities assess the quality of service and improve operations.

Issue Prioritization: The system will automatically categorize complaints based on their severity (e.g., urgent, high, medium, low), ensuring that critical issues are addressed promptly.

Admin Dashboard: Municipal staff will have an intuitive dashboard to view reported issues, and feedback.

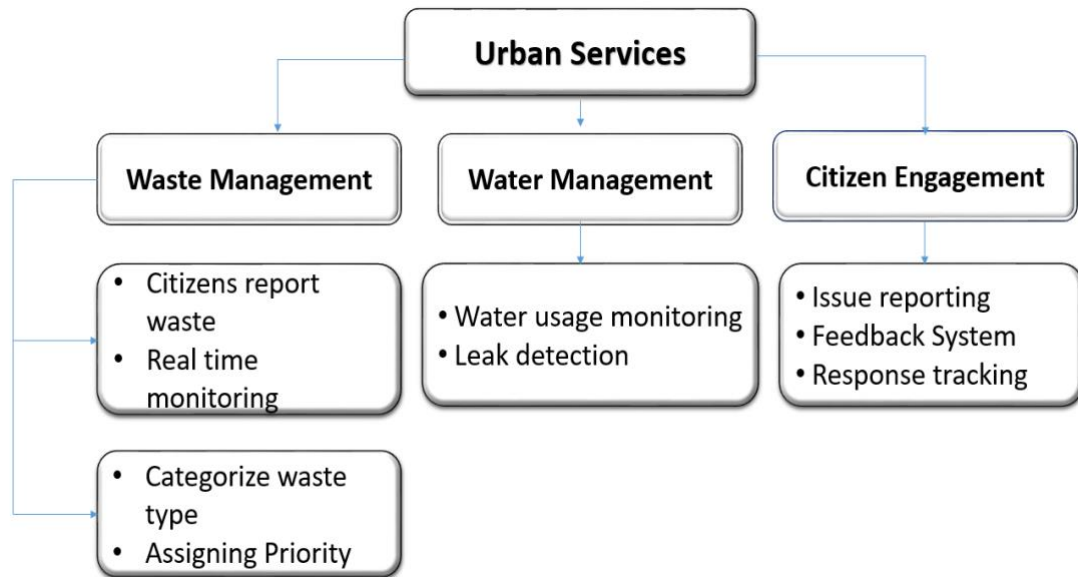


Fig.1 Block Diagram of Optimization of UrbanServices using Priority Queue

1.5 Objectives

The objectives of the UrbanServices platform are:

- To provide a unified platform for reporting and resolving civic issues.
- To enable priority-based issue resolution for efficient management.
- To ensure transparency and accountability in civic services.
- To facilitate communication between citizens and administrators.

1.6 Contribution of the Project

1.6.1 Market Potential

The growing urbanization and demand for smart city solutions make the UrbanServices highly relevant. Municipalities looking to improve service delivery and strengthen citizen-government relationships can greatly benefit from this system. The scalability of the platform ensures it can be implemented in cities of various sizes, providing a comprehensive solution to civic issue management.

As urban populations grow, there is a rising demand for effective civic management platforms. This project addresses the need for improved communication and transparency in urban services, a requirement for cities aiming for smart infrastructure.

1.6.2 Innovativeness

This system introduces a technology-driven approach to civic issue management. Issue prioritization and citizen feedback mechanisms are features rarely seen in traditional reporting systems, making this project innovative and impactful.

The platform uses priority-based algorithms for issue management, improving response times and efficiency. By digitizing civic engagement, it adds value to urban management processes.

1.6.3 Usefulness

By offering citizens an accessible platform to report issues and feedback, the system encourages greater citizen participation in local governance, reduces response times, improves service quality, and builds trust between citizens and municipal authorities.

UrbanServices benefits both citizens and administrators by simplifying issue reporting and management.

1.7 Report Organization

The report is structured into the following chapters:

1. Chapter 1: Introduction – Overview of the project, its objectives, and the rationale.
2. Chapter 2: Requirement Engineering – Discusses the feasibility, requirement gathering, and analysis.
3. Chapter 3: Analysis & Conceptual Design – Details the system architecture, design, and technical components.
4. Chapter 4: Implementation & Testing – Describes the development methodology, tools used, and testing strategies.
5. Chapter 5: Results & Discussion – Presents the results of the implementation and testing, followed by a discussion of findings.
6. Chapter 6: Conclusion & Future Scope – Concludes the report and outlines possible future developments.

REQUIREMENT ENGINEERING

2.1 Feasibility Study

Technical Feasibility: The system is developed using Java Servlets, HTML, CSS, and JavaScript for validation. The backend MySQL database efficiently stores and retrieves user data.

Economic Feasibility: Minimal hardware is required, making this an affordable solution for municipalities.

Operational Feasibility: The platform is user-friendly and easy to deploy, designed to enhance public service interactions and streamline administrative workflows.

2.2 Requirement Collection

2.2.1 Discussion

Requirements were collected through a combination of surveys with citizens and interviews with municipal officials. These discussions identified key challenges, such as the lack of real-time updates and the need for a centralized system for issue tracking. Requirements were gathered from user feedback, analyzing common challenges faced by urban management systems, and studying similar platforms.

2.2.2 Requirement Analysis

The data was analyzed to classify the requirements into essential features and nice-to-haves, ensuring that the critical functionalities were prioritized for implementation.

2.3 Requirements

2.3.1 Functional Requirements

User Registration and Login: Users can create an account, log in securely, and access the system.

Issue Reporting: Users can report various issues, including descriptions, locations, and optional attachment like images.

Admin Dashboard: Municipal staff can view reported issues and feedback, manage resources, and resolve issues in a timely manner.

2.3.2 Non-Functional Requirements

Performance: The system must support at least 500 concurrent users.

Security: User data must be encrypted, and the platform should comply with data protection regulations.

Usability: The user interface must be simple and intuitive, allowing citizens of all age groups and technological proficiency to use the system.

2.4 Hardware & Software Requirements

2.4.1 Hardware Requirement (Developer & End User)

- Processor (CPU): Intel Core i3 or higher
- Memory (RAM): 4 GB or higher
- Storage: 128 GB HDD/SSD
- Display: 1280x1024 resolution
- Browser: Chrome, Firefox, or Edge (latest versions)

2.4.2 Software Requirement (Developer & End User)

- Programming Language: Java for the backend system.
- Frontend Technologies: HTML, CSS for designing the user interfaces and Javascript for validation.
- Database: MySQL for storing user and city management data.
- Integrated Development Environment (IDE): NetBeans 8.0.2
- Operating System: Windows 10

2.5 Use-case Diagrams

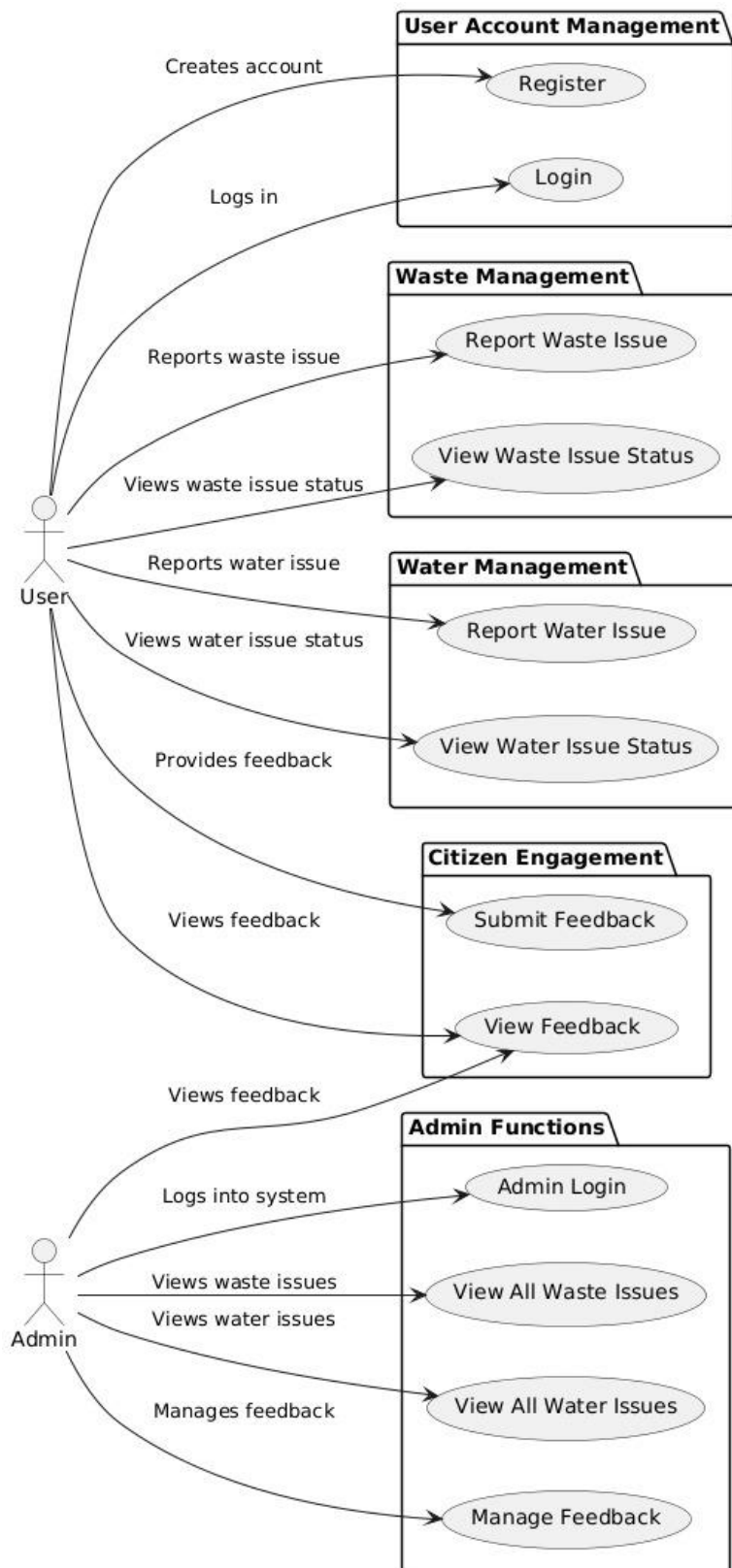


Fig. 2.5 Use case diagram

2.5.1 Use Case Description

1. User Account Management

- Actors: User
- Functions:
 - Register: Users can create an account.
 - Login: Users can log in to access additional functionalities.

2. Waste Management

- Actors: User
- Functions:
 - Report Waste Issue: Users can report issues related to waste management.
 - View Waste Issue Status: Users can check the status of previously reported waste issues.

3. Water Management

- Actors: User
- Functions:
 - Report Water Issue: Users can report issues related to water management.
 - View Water Issue Status: Users can check the status of previously reported water issues.

4. Citizen Engagement

- Actors: User
- Functions:
 - Submit Feedback: Users can provide feedback on various services.
 - View Feedback: Users can view feedback submitted by other users.

5. Admin Functions

- Actors: Admin
- Functions:
 - Admin Login: Admins can log in to manage the system.

ANALYSIS AND CONCEPTUAL DESIGN AND TECHNICAL ARCHITECTURE

3.1 Technical Architecture

The system follows a client-server architecture:

Client-side: The web interface allows users to interact with the system (reporting, tracking, and feedback).

Server-side: The backend processes user inputs, manages databases, and sends updates to users.

Database: A MySQL database stores user data, issues, and feedback.

3.2 Sequence Diagrams

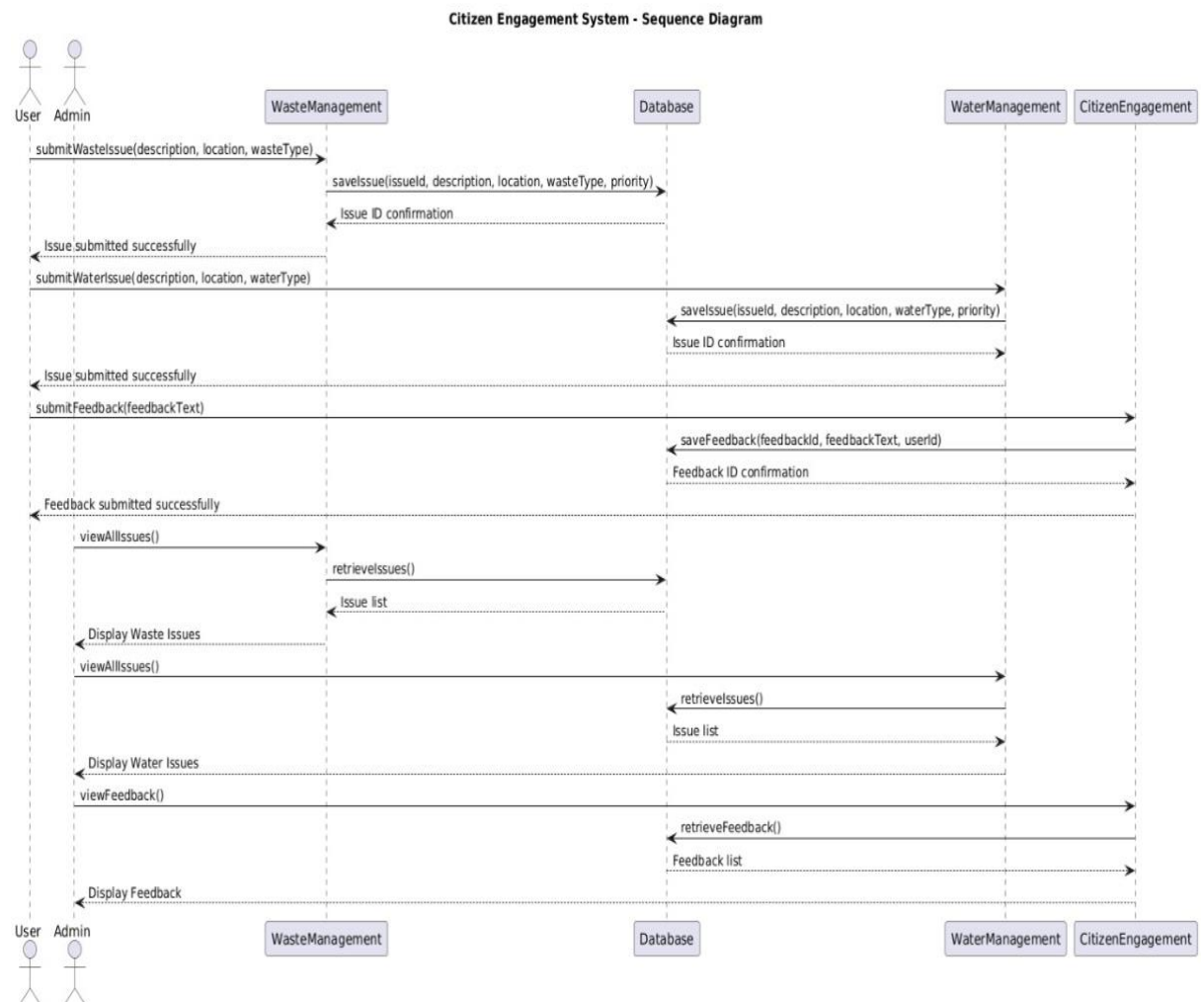


Fig. 3.2 Sequence diagram

3.3 Class Diagrams

Citizen Engagement System - Module-Based Class Diagram

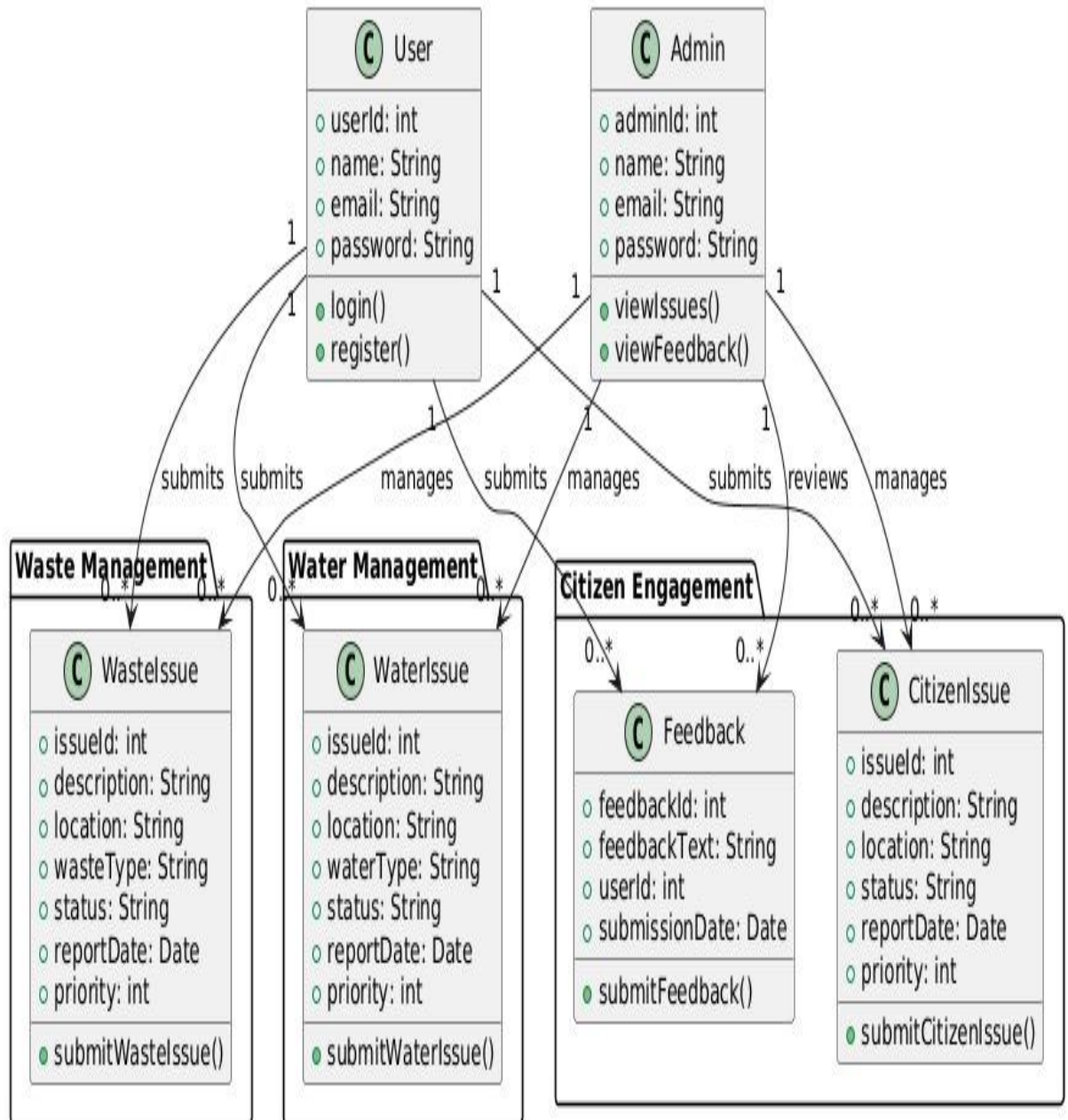


Fig. 3.3 Class diagram

3.4 Data Flow Diagrams (DFD)

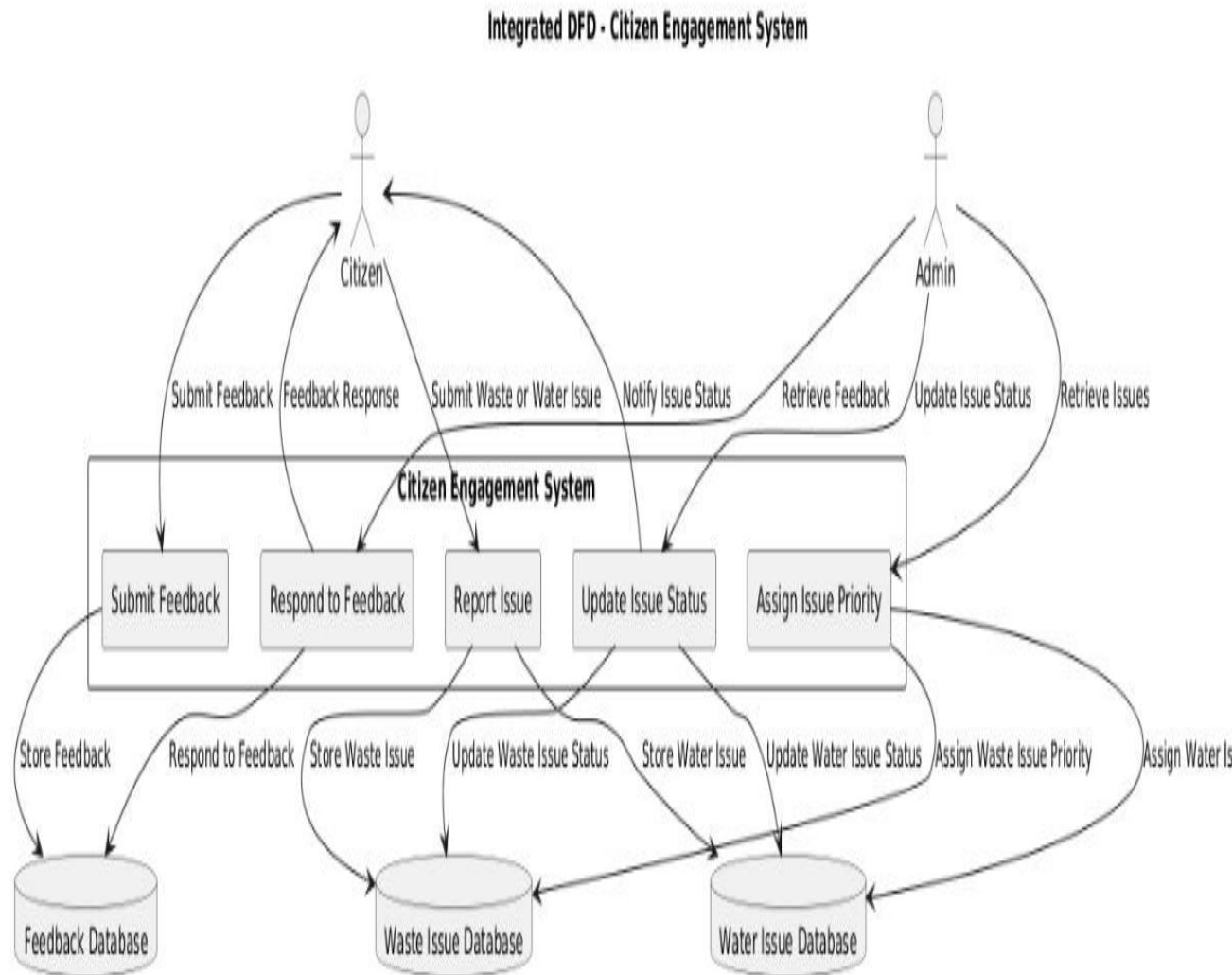


Fig.3.4 Data flow diagram

3.5 User Interface Design

The UI is designed with simplicity and accessibility in mind. It includes:

Responsive Design: The interface adjusts to different screen sizes

Simple UI: Users can process for reporting issues and feedback.

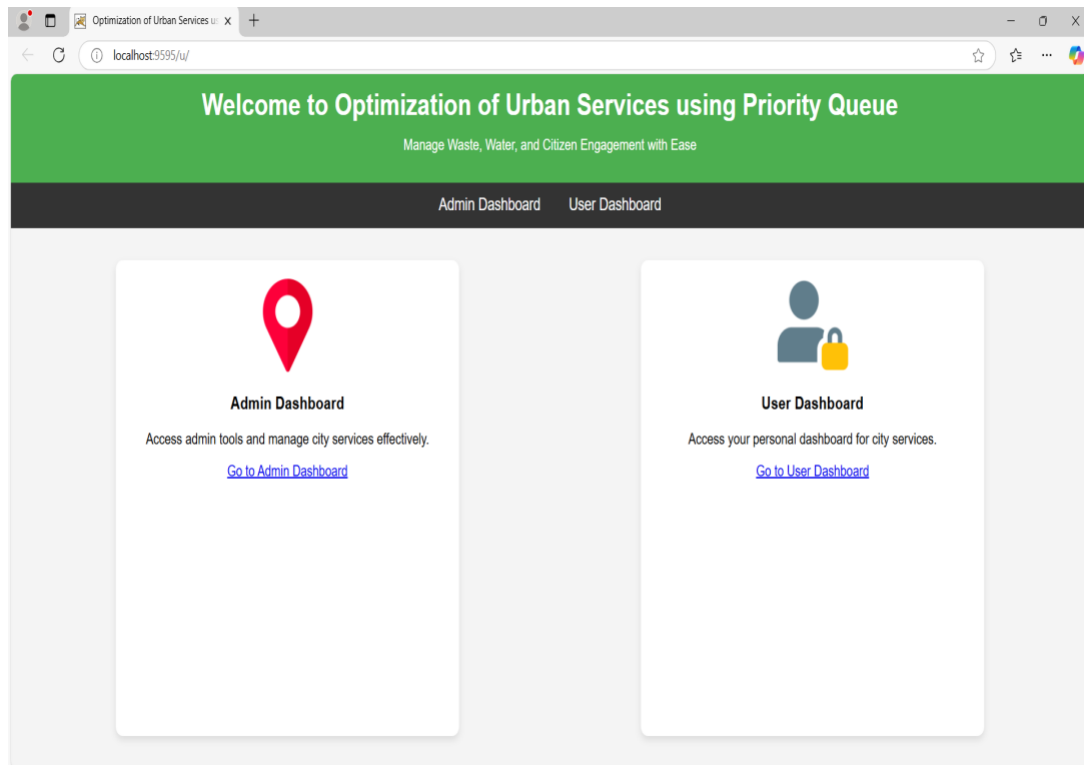


Fig.3.5 UrbanServices Dashboard

3.6 Data Design

3.6.1 Schema Definitions

1. User Registration Table (regis)

SQL Query:

```
CREATE TABLE IF NOT EXISTS regis (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(50) NOT NULL,  
  password VARCHAR(100) NOT NULL,  
  email VARCHAR(100) UNIQUE,  
  contact VARCHAR(15),  
  address VARCHAR(255)  
);
```

Explanation:

- **Purpose:** Stores information for each registered user.
- **Fields:**
 - id: Unique identifier for each user, automatically incremented.
 - name: User's full name, up to 50 characters.
 - password: Hashed password, ensuring security.
 - email: User's email, unique to prevent duplicates.
 - contact: Optional phone contact information.
 - address: Optional address field for user location.

2. Admin Registration Table (adminregis)**SQL Query:**

```
CREATE TABLE IF NOT EXISTS adminregis (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(50) NOT NULL,  
  password VARCHAR(100) NOT NULL,  
  email VARCHAR(100) UNIQUE,  
  contact VARCHAR(15),  
  address VARCHAR(255)  
);
```

Explanation:

- **Purpose:** Manages details of registered administrators.
- **Fields:**
 - id: Unique identifier for each admin, with auto-increment.
 - name: Admin's name, up to 50 characters.
 - password: Hashed password for secure login.
 - email: Admin's email, unique for login purposes.
 - contact: Optional contact number for the admin.

- address: Optional address.

3. Waste Management Issues Report Table (report_issues1)

SQL Query:

```
CREATE TABLE IF NOT EXISTS report_issues1 (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(255) NOT NULL,
  contact VARCHAR(10) NOT NULL,
  address TEXT NOT NULL,
  issue TEXT NOT NULL,
  location VARCHAR(255) NOT NULL,
  waste_type VARCHAR(50) NOT NULL,
  status VARCHAR(50) DEFAULT 'Open',
  image BLOB,
  timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Explanation:

- **Purpose:** Records waste management issues reported by users.
- **Fields:**
 - id: Unique issue identifier.
 - name: Name of the person reporting the issue.
 - contact: Contact number for follow-up.
 - address: Location details of the reporter.
 - issue: Detailed description of the waste issue.
 - location: Specific location for the issue within the service area.
 - waste_type: Type of waste (e.g., plastic, organic) to prioritize based on type.
 - status: Tracks the issue's status, defaulting to "Open."

- image: Optional image provided by the reporter.
- timestamp: Auto-populates with the report submission date and time.

4. Waste Management Feedback Table (feedback)

SQL Query:

```
CREATE TABLE IF NOT EXISTS feedback (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(255) NOT NULL,
  email VARCHAR(255) NOT NULL,
  feedbackType VARCHAR(100) NOT NULL,
  comments TEXT NOT NULL,
  submitted_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Explanation:

- **Purpose:** Stores feedback provided by users.
- **Fields:**
 - id: Unique feedback identifier.
 - name: Name of the user giving feedback.
 - email: Contact email for potential follow-up.
 - feedbackType: Category of feedback, such as “Complaint” or “Suggestion.”
 - comments: Detailed feedback comments.
 - submitted_at: Submission date and time for tracking.

5. Water Management Issues Report Table (water_issues)

SQL Query:

```
CREATE TABLE IF NOT EXISTS water_issues (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(255) NOT NULL,  
  contact VARCHAR(10) NOT NULL,  
  address TEXT NOT NULL,  
  issue TEXT NOT NULL,  
  location VARCHAR(255) NOT NULL,  
  issue_type VARCHAR(50) NOT NULL,  
  status VARCHAR(50) DEFAULT 'Open',  
  image BLOB,  
  timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

Explanation:

- **Purpose:** Manages reports related to water management issues.
- **Fields:**
 - id: Unique identifier for each water issue.
 - name: Name of the reporter.
 - contact: Contact number.
 - address: Address of the person reporting.
 - issue: Description of the water-related issue.
 - location: Location details for resolving the issue.
 - issue_type: Type of water issue (e.g., leakage, low pressure).
 - status: Current status, with a default of “Open.”
 - image: Optional image to illustrate the issue.
 - timestamp: Automatically records when the issue is reported.

6. Water Management Feedback Table (water_feedback)

SQL Query:

```
CREATE TABLE IF NOT EXISTS water_feedback (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(255) NOT NULL,  
  email VARCHAR(255) NOT NULL,  
  feedbackType VARCHAR(50) NOT NULL,  
  comments TEXT NOT NULL,  
  timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

Explanation:

- **Purpose:** Stores feedback on water services.
- **Fields:**
 - id: Unique feedback entry ID.
 - name: Feedback giver's name.
 - email: Email for follow-up.
 - feedbackType: Type of feedback related to water services.
 - comments: Detailed feedback.
 - timestamp: Records the submission time.

7. Citizen Engagement Table (citizen_engagement)

SQL Query:

```
CREATE TABLE IF NOT EXISTS citizen_engagement (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(255) NOT NULL,  
  contact VARCHAR(50) NOT NULL,  
  address TEXT NOT NULL,
```

```
issue_type VARCHAR(50) NOT NULL,  
image BLOB,  
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

Explanation:

- **Purpose:** Tracks citizen engagement issues.
- **Fields:**
 - id: Unique identifier for each issue reported.
 - name: Reporter's name.
 - contact: Contact number for follow-up.
 - address: Address for reference.
 - issue_type: Type of engagement issue (e.g., public safety, community event).
 - image: Optional image upload.
 - created_at: Submission date.

8. Citizen Engagement Feedback Table (feedback2)

SQL Query:

```
CREATE TABLE IF NOT EXISTS feedback2 (  
id INT AUTO_INCREMENT PRIMARY KEY,  
name VARCHAR(255) NOT NULL,  
email VARCHAR(255) NOT NULL,  
feedbackType VARCHAR(255) NOT NULL,  
comments TEXT NOT NULL,  
submitted_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP);
```

Explanation:

- **Purpose:** Stores additional or secondary feedback.

- **Fields:**
 - id: Primary key for unique identification.
 - name: Name of the feedback provider.
 - email: Contact email.
 - feedbackType: Type/category of feedback.
 - comments: Detailed comments on feedback.
 - submitted_at: Timestamp for when feedback is submitted.

3.6.2 ER Diagram

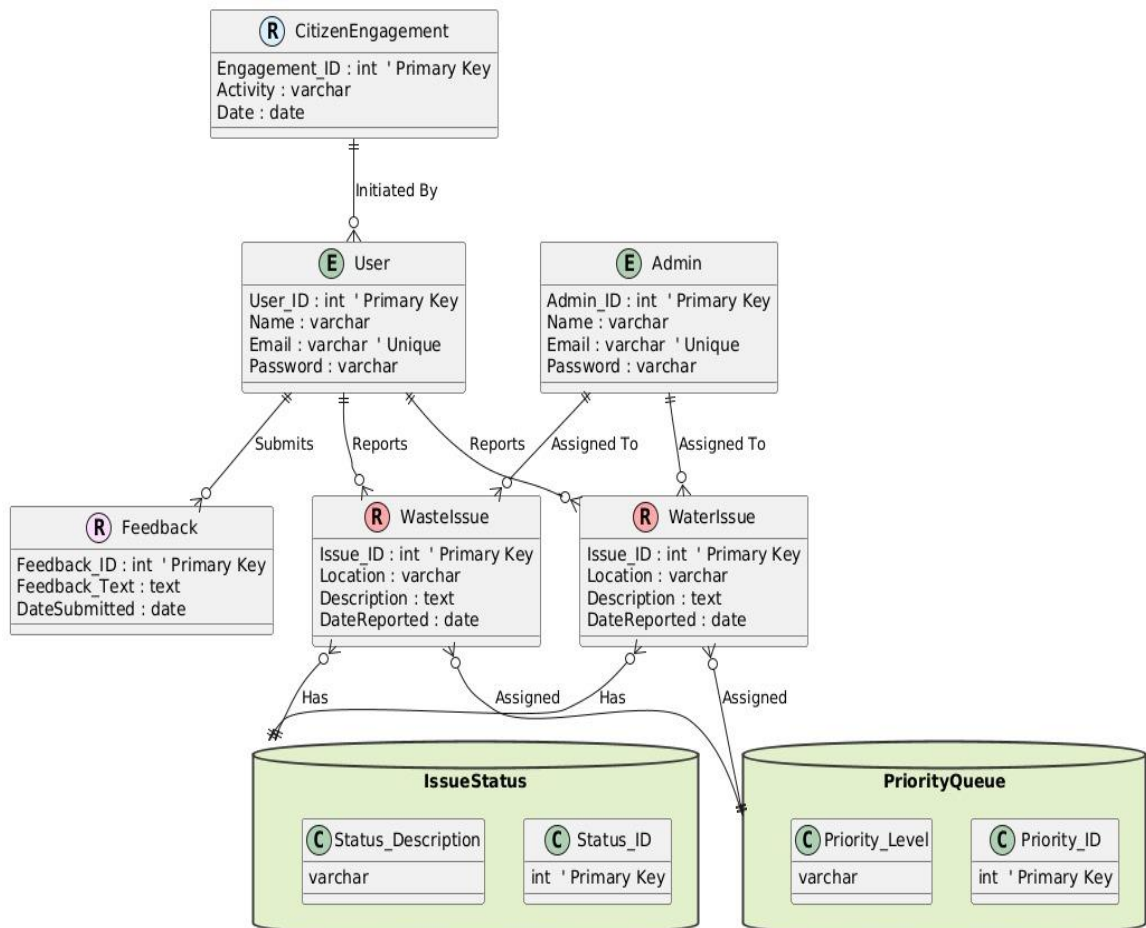


Fig.3.6.2 ER diagram

IMPLEMENTATION AND TESTING

4.1 Methodology

An Agile Development methodology was employed, allowing iterative development and frequent testing and feedback cycles. This approach ensures that the project meets user requirements and allows flexibility for modifications during the development process.

4.1.1 Proposed Algorithm

1. User Registration and Login

- User registers by providing details; system validates and stores in Users table.
- User logs in; credentials are verified, granting access to the dashboard.

2. Issue Reporting

- User fills issue form (description, location, image), system validates and saves to Issues table with "Pending" status.

3. Status Tracking

- User views all reported issues and their statuses on the dashboard.
- System sends notifications of any status updates.

4. Feedback Submission

- After resolution, user submits feedback (rating/comments), stored in Feedback table.

5. Admin Issue Management

- Admin reviews and updates issue statuses (e.g., "In Progress," "Resolved") via the admin dashboard.
- System notifies users of updates.

6. Notification System

- Notifications sent to users for any status changes via in-app or email, ensuring transparency.

4.2 Implementation Approach

The development used the following technologies:

The project uses HTML, CSS, JavaScript, and Java Servlets, with MySQL for the database. Validation is handled via JavaScript.

4.2.1 Introduction to Languages, IDEs, tools and Technology

Java: Java is a widely-used, high-level programming language known for its portability, security, and robustness. It follows the object-oriented programming (OOP) paradigm, making it easy to organize and structure code through objects and classes. Java programs are compiled into bytecode, which can be executed on any device with a Java Virtual Machine (JVM), enabling "write once, run anywhere" functionality. This platform independence is a key advantage, allowing Java applications to run on diverse systems without modification. Java prioritizes security with features like the Java sandbox and security manager, which protect against unauthorized code execution and ensure safe program operation. It includes strong memory management, exception handling, and multithreading support, making it suitable for building scalable and reliable applications. Java's rich standard library offers a vast array of utilities and APIs for common tasks, from basic I/O operations to advanced networking and GUI development. Its combination of simplicity, performance optimization through JIT compilation, and a robust development ecosystem makes Java a preferred choice for enterprise applications, Android development, and web services.

2.HTML: HTML (Hypertext Markup Language) is the standard language used to create and design web pages. It provides a set of elements or tags that structure the content of a webpage. Each HTML tag represents a different part of the content, such as headings, paragraphs, links, images, and forms. Tags are enclosed in angle brackets, `< >`, and usually come in pairs: an opening tag and a closing tag, with content nested between them. For example, `<p>` and `</p>` are tags used to define a paragraph. HTML is not a programming language but rather a markup language that defines the structure and layout of a webpage, including text, images, and other elements. Modern web development often combines HTML with CSS (Cascading Style Sheets) for styling and JavaScript for interactivity, forming the core technologies of the World Wide Web.

3.CSS: CSS (Cascading Style Sheets) is a fundamental technology used for styling and formatting web pages written in HTML and XML. It allows developers to control the appearance of web pages by defining styles for various HTML elements. CSS separates the content of a webpage from its design, enabling a more flexible and efficient approach to web development. Styles are applied to HTML elements using selectors, which can target elements based on their type, class, ID, or other attributes. Each CSS rule consists of a selector and a declaration block, where properties and values define how the selected elements should be styled. For example, CSS can define fonts, colors, margins, padding, and layout, making it possible to create visually appealing and responsive web pages.

4.NetBeans: NetBeans is an open-source Integrated Development Environment (IDE) primarily used for Java development, though it supports other languages like PHP, HTML5, C++, and JavaScript. Known for its intuitive user interface, NetBeans offers a comprehensive suite of tools that streamline coding, debugging, and project management. One of its key features is the drag-and-drop GUI builder, which makes designing user interfaces much easier. Additionally, NetBeans supports numerous plugins, allowing developers to extend its functionality based on specific needs. The IDE also provides real-time error detection and syntax highlighting, which improves coding accuracy and efficiency. Its robust code refactoring tools enable quick and safe modifications across large projects. NetBeans integrates seamlessly with version control systems like Git, allowing collaborative work and efficient code management. Furthermore, it's cross-platform, running on Windows, macOS, and Linux, which increases its accessibility.

5.JDBC: JDBC (Java Database Connectivity) is an essential Java API that enables Java applications to interact with relational databases. It provides a uniform interface for accessing and managing data stored in databases, regardless of the specific database management system (DBMS) being used. JDBC allows developers to perform database operations such as querying data, inserting or updating records, and executing stored procedures directly from Java code.

The core components of JDBC include the DriverManager, which handles the loading and registering of database drivers, and the Connection interface, which represents a connection to a specific database. Developers use the Connection interface to create

Statement objects for executing SQL queries and updates, or PreparedStatement objects for precompiled queries that improve performance and security.

6.Servlet: Servlets are Java programming language components that dynamically process and respond to requests from web clients, usually browsers, by generating web content. They are server-side programs running inside a web server that handle client requests and provide responses. Servlets are part of the Java Enterprise Edition (Java EE) technology stack and are typically used to implement web applications.

4.3 Testing Approaches

4.3.1 Unit Testing

a. Test Cases

Each module was tested individually to ensure its functionality, with test cases including:

User Registration (validations on email and password).

Issue Reporting (mandatory fields).

TABLE I
TEST CASES FOR OPTIMIZATION OF URBANSERVICES USING
PRIORITY QUEUE

| Test Case Id | Test scenario | Test steps | Test data | Expected result | Actual result |
|---------------------|--------------------------------|---|---------------------------------------|--|----------------------|
| UU01 | Create admin registration page | 1. Open website and clicked on admin registration | Information of admin data to be added | Data is successfully added in adminregis table | As expected |

| | | | | | |
|------|---|---|---|--|-------------|
| UU02 | Create user registration page | 1. Open website and clicked on user registration | Information of user data to be added | Data is successfully added in regis table | As expected |
| UU03 | Create login page for admin | 1.open website and clicked on admin login 2. enter username and password | Admin enter valid data | Data is fetch from adminregis table | As expected |
| UU04 | Create login page for user | 1.open website and clicked on user login 2. enter username and password | User enter valid data | Data is fetch from regis table | As expected |
| UU05 | Report issue page for user for waste management | User login successfully then they report issue for waste | User filed necessary data to require for report issue | Data is added successfully in report_issues1 table | As expected |

| | | | | | |
|------|---|---|---|--|-------------|
| UU06 | Feedback page for user for waste management | User login successfully then they provide Feedback for waste management | User filed necessary data to require for feedback | Data is added successfully in feedback table | As expected |
| UU07 | View issues page for admin for waste management | Admin see applicant name and other information with the issue | Admin process for the issue solution | Successfully processed | As expected |
| UU08 | View feedback page for admin for waste management | Admin see applicant name and other information with the feedback | Admin process for the feedback | Successfully processed | As expected |
| UU09 | Report issue page for user for water management | User login successfully then they report issue for water | User filed necessary data to require for report issue | Data is added successfully in water_issues table | As expected |

| | | | | | |
|------|---|---|---|--|-------------|
| UU10 | Feedback page for user for water management | User login successfully then they provide Feedback for water management | User filed necessary data to require for feedback | Data is added successfully in water_ feedback table | As expected |
| UU11 | View issues page for admin for water management | Admin see applicant name and other information with the issue | Admin process for the issue solution | Successfully processed | As expected |
| UU12 | View feedback page for admin for water management | Admin see applicant name and other information with the feedback | Admin process for the feedback | Successfully processed | As expected |
| UU13 | Report issue page for user for citizen engagement | User login successfully then they report issue | User filed necessary data to require for report issue | Data is added successfully in citizen_engagement table | As expected |

| | | | | | |
|------|--|--|--|---|----------------|
| UU14 | Feedback page for user for citizen engagement | User login successfully then they provide Feedback | User filed necessary data to require for feedback | Data is added successfully in feedback2 table | As expected |
|------|--|--|--|---|----------------|

RESULTS AND CONCLUSION

5.1 User Interface Representation

5.1.1 Brief Description of Various Modules

1. Admin Registration Module

- Description: Provides an interface for admins to register on the platform.
- Expected Outcome: Admin information is saved in the admin regis table.

2. User Registration Module

- Description: Allows users to register on the platform.
- Expected Outcome: User details are stored in the regis table.

3. Admin Login Module

- Description: Provides a login interface for admins with a username and password authentication.
- Expected Outcome: Valid login data fetches admin details from the admin regis table.

4. User Login Module

- Description: Provides a login interface for users with a username and password authentication.
- Expected Outcome: Valid login data fetches user details from the regis table.

5. Waste Management - Issue Reporting Module

- Description: Allows logged-in users to report waste management issues.
- Expected Outcome: Issue details are saved in the report_issues1 table.

6. Waste Management - Feedback Module

- Description: Allows users to submit feedback for waste management services.
- Expected Outcome: Feedback data is stored in the feedback table.

7. Waste Management - Admin Issue Viewing Module

- Description: Enables admins to view waste management issues reported by users.

- Expected Outcome: Admin can access user-submitted issue information and begin resolution.

8. Waste Management - Admin Feedback Viewing Module

- Description: Enables admins to view feedback from users on waste management.
- Expected Outcome: Admin can process user feedback for improvement.

9. Water Management - Issue Reporting Module

- Description: Allows users to report issues related to water management.
- Expected Outcome: Issue details are saved in the water issues table.

10. Water Management - Feedback Module

- Description: Allows users to submit feedback regarding water management services.
- Expected Outcome: Feedback is stored in the water feedback table.

11. Water Management - Admin Issue Viewing Module

- Description: Enables admins to view reported issues related to water management.
- Expected Outcome: Admin can access and process user-submitted water issues.

12. Water Management - Admin Feedback Viewing Module

- Description: Enables admins to review feedback related to water management.
- Expected Outcome: Admin can address feedback to improve water management services.

13. Citizen Engagement - Issue Reporting Module

- Description: Allows users to report various issues concerning citizen engagement.
- Expected Outcome: Issue data is stored in the citizen engagement table.

14. Citizen Engagement - Feedback Module

- Description: Allows users to provide feedback related to citizen engagement and expected outcome is saved in feedback2 table.

5.2 Snapshot of System with Brief Description

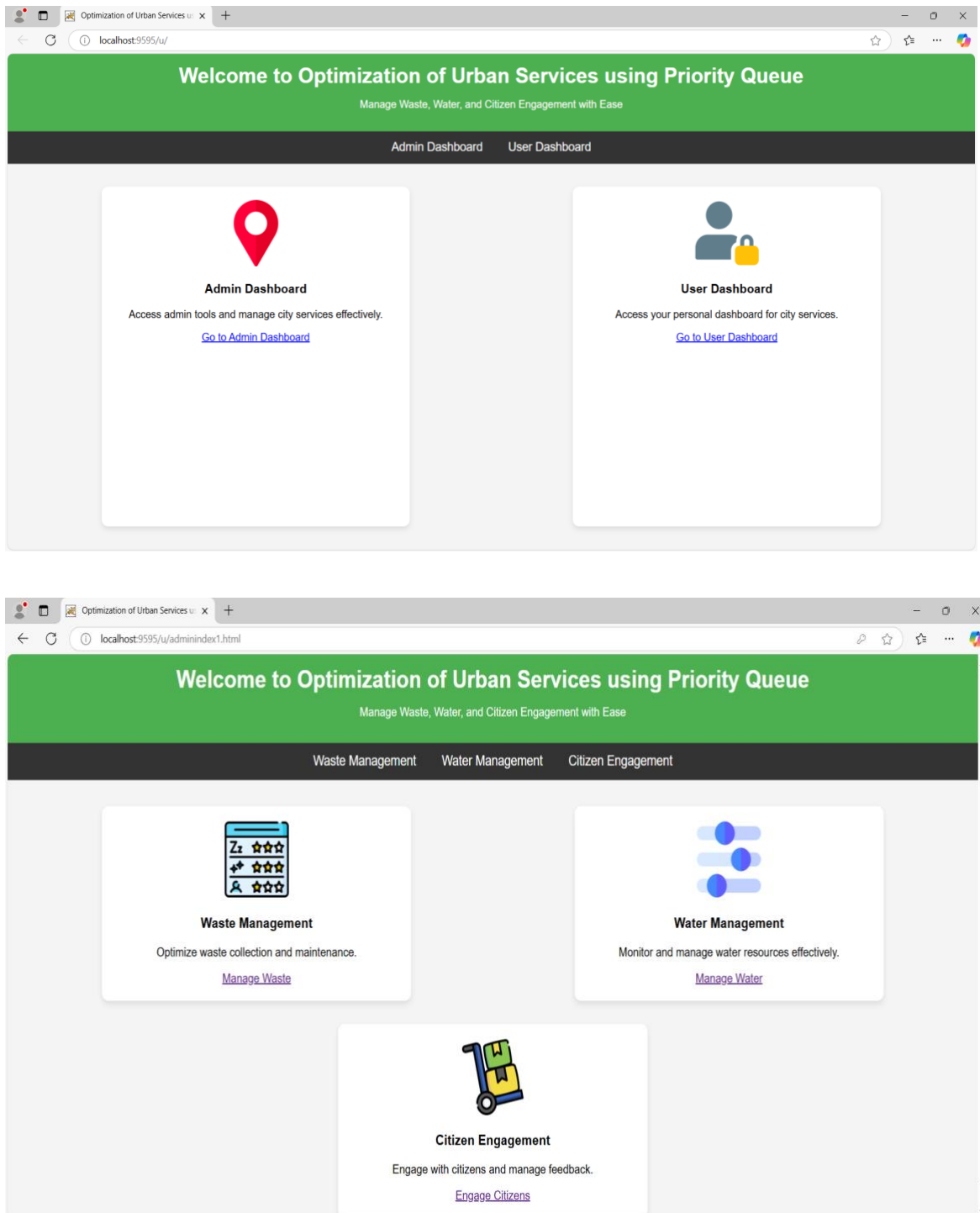


Fig.5.2(a) UrbanServices Platform:Dashboards

Registration Form

Name:

Email:

Password:

Contact:

Address:

Register

Login

Username:

Password:

Login

Don't have an account? [Register here](#)

Fig.5.2(b) UrbanServices Platform:Registration and Login

The image displays a web browser window with the title 'Waste Management System' and the URL 'localhost:9595/uy/wasteManagement.html'. The main content area features a light blue background with a large green diagonal stripe. At the top, the heading 'Waste Management System' is centered. Below it are two green buttons: 'Report Issue' with the text 'Click to report a waste issue in your community.' and 'Feedback' with the text 'Provide feedback on our waste management services.'

Below the main content, two detailed form views are shown side-by-side. The left form is titled 'Report an Issue' and contains the following fields: 'Your Name' (text input), 'Contact Number' (text input), 'Address' (text input), 'Issue Description' (text area), 'Location' (text input), 'Waste Type' (dropdown menu), and 'Upload Image (Optional)' (file upload button). A green 'Submit Report' button is at the bottom. The right form is titled 'Submit Feedback' and contains: 'Name' (text input), 'Email' (text input), 'Feedback Type' (dropdown menu), and 'Comments' (text area). A green 'Submit Feedback' button is at the bottom.

Fig.5.2(c) Waste Management: Report & Feedback Forms

The image displays a web application for a Water Management System. The top section features a header with the title "Water Management System" and two main action buttons: "Report Water Issue" and "Feedback". Below this, the application is split into two panels. The left panel, titled "Submit Your Feedback", contains a form with fields for "Your Name", "Your Email", "Feedback Type" (a dropdown menu), and "Your Comments". The right panel, titled "Report an Issue", contains a more detailed form with fields for "Your Name", "Contact Number", "Address", "Issue Description", "Location", "Waste Type" (a dropdown menu), and an "Upload image (Optional)" section with a "Choose File" button. Both panels have a "Submit" button at the bottom.

Water Management System

Report Water Issue

Click to report a water-related issue in your community.

Feedback

Provide feedback on our water management services.

Submit Your Feedback

Your Name

Your Email

Feedback Type

Select Feedback Type

Your Comments

SUBMIT FEEDBACK

Report an Issue

Your Name

Contact Number

Address

Issue Description

Location

Waste Type

Select Issue Type

Upload image (Optional)

Choose File
No file chosen

Submit Report

Fig.5.2(d) Water Management: Report & Feedback Forms

+

ment.html

CITIZEN ENGAGEMENT SYSTEM

Report Issue

Report any civic issues such as water supply, road problems, etc.

Feedback

Provide feedback on various civic services.

Submit Your Feedback

Your Name

Your Email

Feedback Type

Select Feedback Type

Your Comments

SUBMIT FEEDBACK

Citizen Engagement - Report Issue

Submit Your Civic Issue

Full Name

Contact Number

Address

Issue Type

Waste Management

Upload Image (Optional)

Choose File
No file chosen

Submit Report

Fig.5.2(e) Citizen Engagement: Report & Feedback Forms

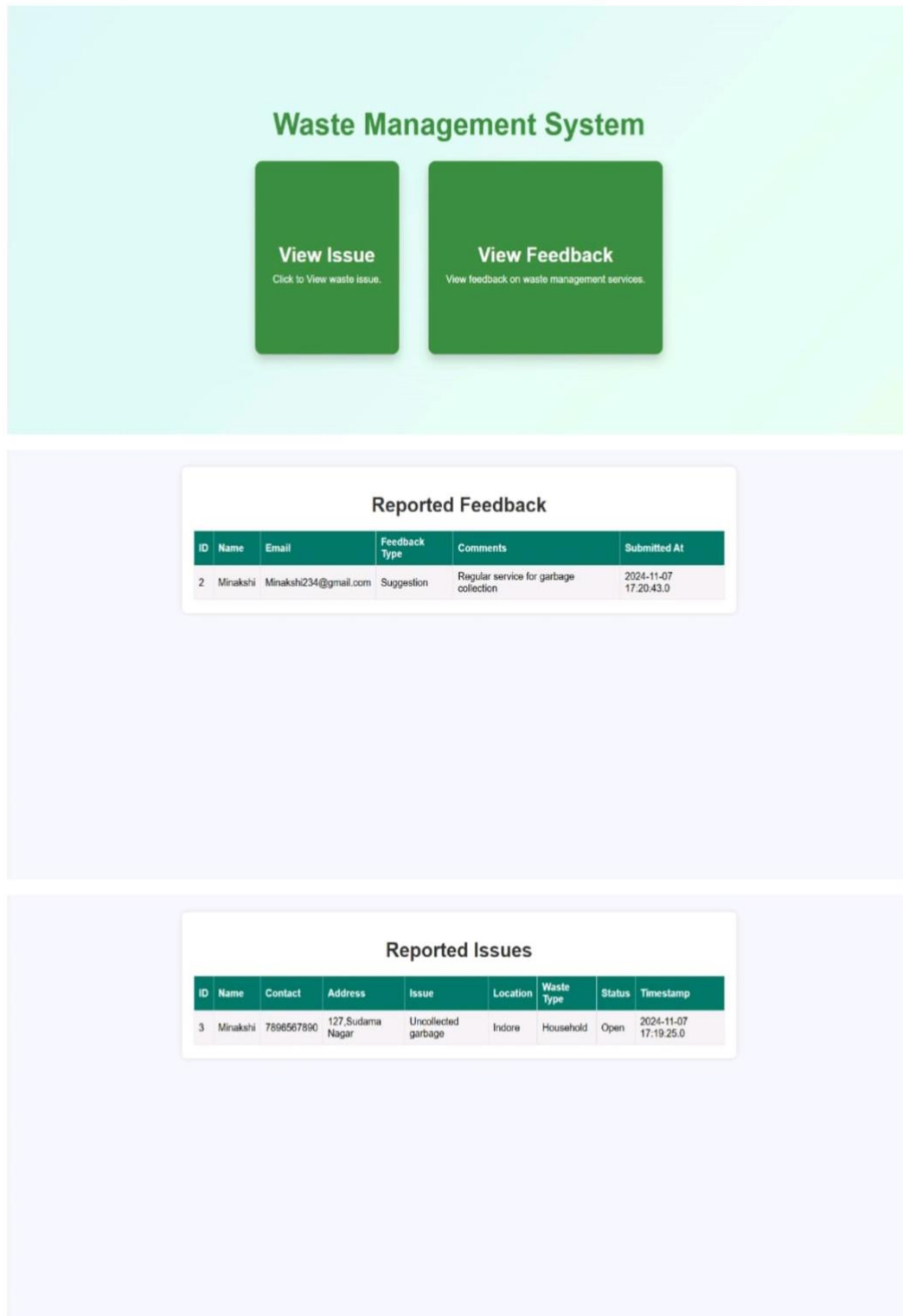


Fig 5.2(f) Waste Management:View Issues and Feedback

5.3 Database Description

The database schema includes tables such as regis,adminregis, report_issues1,feedback,water_issues,water_feedback,citizen_engagement,feedback2 which store essential information of users, issue details and user feedback.

5.3.1 Snapshot of database table with brief description

TABLE II
USER AND ADMIN REGISTRATION

```
mysql> desc regis;
```

| Field | Type | Null | Key | Default | Extra |
|----------|--------------|------|-----|---------|----------------|
| id | int | NO | PRI | NULL | auto_increment |
| name | varchar(50) | NO | | NULL | |
| password | varchar(100) | NO | | NULL | |
| email | varchar(100) | YES | UNI | NULL | |
| contact | varchar(15) | YES | | NULL | |
| address | varchar(255) | YES | | NULL | |

6 rows in set (0.00 sec)

```
mysql> desc adminregis;
```

| Field | Type | Null | Key | Default | Extra |
|----------|--------------|------|-----|---------|----------------|
| id | int | NO | PRI | NULL | auto_increment |
| name | varchar(50) | NO | | NULL | |
| password | varchar(100) | NO | | NULL | |
| email | varchar(100) | YES | UNI | NULL | |
| contact | varchar(15) | YES | | NULL | |
| address | varchar(255) | YES | | NULL | |

6 rows in set (0.00 sec)

```
mysql> _
```

TABLE III

USER REPORT ISSUE AND FEEDBACK FOR WASTE MANAGEMENT

MySQL 8.0 Command Line Client

```
mysql> use urbans;
Database changed
mysql> desc report_issues1;
```

| Field | Type | Null | Key | Default | Extra |
|------------|--------------|------|-----|-------------------|-------------------|
| id | int | NO | PRI | NULL | auto_increment |
| name | varchar(255) | NO | | NULL | |
| contact | varchar(10) | NO | | NULL | |
| address | text | NO | | NULL | |
| issue | text | NO | | NULL | |
| location | varchar(255) | NO | | NULL | |
| waste_type | varchar(50) | NO | | NULL | |
| status | varchar(50) | YES | | Open | |
| image | blob | YES | | NULL | |
| timestamp | timestamp | YES | | CURRENT_TIMESTAMP | DEFAULT_GENERATED |

10 rows in set (0.25 sec)

```
mysql> desc feedback;
```

| Field | Type | Null | Key | Default | Extra |
|--------------|--------------|------|-----|-------------------|-------------------|
| id | int | NO | PRI | NULL | auto_increment |
| name | varchar(255) | NO | | NULL | |
| email | varchar(255) | NO | | NULL | |
| feedbackType | varchar(100) | NO | | NULL | |
| comments | text | NO | | NULL | |
| submitted_at | timestamp | YES | | CURRENT_TIMESTAMP | DEFAULT_GENERATED |

6 rows in set (0.04 sec)

TABLE IV

USER REPORT ISSUE FOR WATER MANAGEMENT

```
mysql> desc water_issues;
```

| Field | Type | Null | Key | Default | Extra |
|------------|--------------|------|-----|-------------------|-------------------|
| id | int | NO | PRI | NULL | auto_increment |
| name | varchar(255) | NO | | NULL | |
| contact | varchar(10) | NO | | NULL | |
| address | text | NO | | NULL | |
| issue | text | NO | | NULL | |
| location | varchar(255) | NO | | NULL | |
| issue_type | varchar(50) | NO | | NULL | |
| status | varchar(50) | YES | | Open | |
| image | blob | YES | | NULL | |
| timestamp | timestamp | YES | | CURRENT_TIMESTAMP | DEFAULT_GENERATED |

10 rows in set (0.00 sec)

```
mysql> desc water_feedback;
```


TABLE V
USER FEEDBACK FOR WATER MANAGEMENT AND REPORT ISSUE
FOR CITIZEN ENGAGEMENT

MySQL 8.0 Command Line Client

```
mysql> desc water_feedback;
```

| Field | Type | Null | Key | Default | Extra |
|--------------|--------------|------|-----|-------------------|-------------------|
| id | int | NO | PRI | NULL | auto_increment |
| name | varchar(255) | NO | | NULL | |
| email | varchar(255) | NO | | NULL | |
| feedbackType | varchar(50) | NO | | NULL | |
| comments | text | NO | | NULL | |
| timestamp | timestamp | YES | | CURRENT_TIMESTAMP | DEFAULT_GENERATED |

6 rows in set (0.02 sec)

```
mysql> desc citizen_engagement;
```

ERROR 1146 (42S02): Table 'urbans.citizen' doesn't exist

```
mysql> desc citizen_engagement;
```

| Field | Type | Null | Key | Default | Extra |
|------------|--------------|------|-----|-------------------|-------------------|
| id | int | NO | PRI | NULL | auto_increment |
| name | varchar(255) | NO | | NULL | |
| contact | varchar(50) | NO | | NULL | |
| address | text | NO | | NULL | |
| issue_type | varchar(50) | NO | | NULL | |
| image | blob | YES | | NULL | |
| created_at | timestamp | YES | | CURRENT_TIMESTAMP | DEFAULT_GENERATED |

7 rows in set (0.02 sec)

TABLE VI
USER FEEDBACK FOR CITIZEN ENGAGEMENT

```
mysql> desc feedback2;
```

| Field | Type | Null | Key | Default | Extra |
|--------------|--------------|------|-----|-------------------|-------------------|
| id | int | NO | PRI | NULL | auto_increment |
| name | varchar(255) | NO | | NULL | |
| email | varchar(255) | NO | | NULL | |
| feedbackType | varchar(255) | NO | | NULL | |
| comments | text | NO | | NULL | |
| submitted_at | timestamp | YES | | CURRENT_TIMESTAMP | DEFAULT_GENERATED |

6 rows in set (0.00 sec)

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

The UrbanServices provides an efficient and transparent way for citizens to report issues and track their resolution, enhancing communication between citizens and municipal authorities and improving civic management.

The UrbanServices project successfully addresses urban management challenges by providing a platform for efficient reporting and tracking of civic issues. The priority-based queue system allows for timely resolution, fostering trust and engagement between citizens and city authorities.

6.2 Future Scope

Future improvements could include:

- Mobile App Development for easier access.
- Integration with Social Media to allow reporting through platforms like Twitter or Facebook.
- Advanced Analytics for data-driven decision-making, helping municipalities optimize their resources and improve service delivery.

REFERENCES

- [1] Patel, S. et al., "A Study on Smart City Projects in India", Journal of Urban Technologies, 2023.
- [2] Bhardwaj, R., "Waste Management through Smart Technologies", Environmental Research Journal, 2022.
- [3] Gupta, P., "Leveraging IoT for Water Management in Urban Areas", Journal of Smart Systems and Technologies, 2021.
- [4] Sharma, A. et al., "The Role of Citizen Engagement in Urban Governance", International Journal of Public Administration, 2021.
- [5] Mohan, K., "Applications of Java in Smart City Solutions", Java Technologies and Future Systems, 2023.