|  |  |
| --- | --- |
| A picture of a winding road and trees  DEVELOPMENT TESTING DOCUMENT  ICS 2302: SOFTWARE ENGINEERING | SMART WASTE MANAGEMENT SYSTEM  AKECH DAU ATEM - SCT211-0535/2022  GATMACH YUOL NYUON – SCT211-0504/2021 |

Table of Contents

[1. Introduction 2](#_Toc187868738)

[2. Testing Objectives 3](#_Toc187868739)

[3. Testing Scope 3](#_Toc187868740)

[4. Testing Methodologies 3](#_Toc187868741)

[4.1 Unit Testing 4](#_Toc187868742)

[4.2 Integration Testing 4](#_Toc187868743)

[4.3 System Testing 4](#_Toc187868744)

[**4**.4. Performance Testing 4](#_Toc187868745)

[4.5 Security Testing 4](#_Toc187868746)

[4.6. User Acceptance Testing (UAT) 5](#_Toc187868747)

[4.7 Compatibility Testing 5](#_Toc187868748)

[5. Test Environment 5](#_Toc187868749)

[6. Test Case Overview 5](#_Toc187868750)

[6.1. Unit Testing 5](#_Toc187868751)

[6.2. Integration Testing 6](#_Toc187868752)

[6.3. System Testing 6](#_Toc187868753)

[6.4. Performance Testing 9](#_Toc187868754)

[6.5. Security Testing 9](#_Toc187868755)

[6.6. User Interface (UI) Testing 10](#_Toc187868756)

[6.7. Compatibility Testing 10](#_Toc187868757)

[7. Testing Schedule 13](#_Toc187868758)

[8. Conclusion 13](#_Toc187868759)

# Introduction

The Smart Waste Management System (SWMS) is a cutting-edge solution that revolutionizes waste collection, transportation, and disposal by integrating advanced technologies such as IoT (Internet of Things) and real-time data analytics. By leveraging these technologies, SWMS enables municipalities and waste management companies to enhance operational efficiency and responsiveness, addressing the challenges posed by traditional waste management practices. The SWMS leverages smart cameras to capture and upload images of waste bins, which are then processed to provide real-time status updates, indicating whether bins are full, partially full, or empty. This facilitates efficient monitoring and data-driven decision-making for waste management. This real-time data enables timely collection schedules, reducing overflowing bins, and helps determine optimal routes for waste collection vehicles, resulting in significant fuel savings and a reduction in greenhouse gas emissions. Furthermore, the system analyzes waste generation patterns, providing valuable insights into seasonal trends and identifying areas with high recycling potential to promote sustainability and recycling efforts. Beyond operational optimization, the SWMS aims to foster community engagement by providing users with an application that delivers notifications on collection schedules, recycling information, and educational resources for waste reduction. This approach encourages active participation from residents in sustainability initiatives and creates a culture of environmental responsibility.

This document outlines the testing strategy, methodologies, and procedures to ensure the system’s functionality, performance, reliability, and security. The testing phase is crucial in the software development lifecycle, ensuring the system operates as intended, meets stakeholder requirements, and provides a seamless user experience. The document provides detailed test cases for various functionalities, expected outcomes for validation, and methods to verify the system's robustness and reliability. A feedback loop is incorporated for continuous improvement, allowing the system to adapt to evolving waste management needs post-deployment. By rigorously testing the Smart Waste Management System (SWMS), the project aims to deliver a robust, transparent, and sustainable waste management solution that benefits municipalities, stakeholders, and communities, paving the way for a cleaner, greener future.

# Testing Objectives

* Validate the core functionalities of the SWMS, including bin status monitoring, notifications, and route optimization.
* Ensure smooth integration of all system components.
* Test the system under various load conditions to assess performance.
* Confirm the security of user data and system interactions.
* Verify the responsiveness and usability of the system across devices.

# Testing Scope

This document covers:

**Functional testing** of core features: bin monitoring, route optimization, real-time notifications, and user dashboards.

**Non-functional testing**: performance, scalability, and security.

**End-to-end testing**: data flow from sensors to the central system and user-facing applications.

**Compatibility testing** across devices and platforms.

# Testing Methodologies

## 4.1 Unit Testing

**Objective**: Validate the functionality of individual components.

**Scope**: Algorithms for bin status detection, API endpoints, and route optimization logic.

**Tools**: Python unit test or PyTest framework.

**Expected Outcome**: Each component performs as expected, handling all edge cases.

## 4.2 Integration Testing

**Objective**: Ensure seamless interaction between system modules.

**Scope**: Integration between sensors, databases, analytics engines, and user interfaces.

**Tools**: Postman for API testing, and integration scripts.

**Expected Outcome**: The data flow between modules is smooth, with no errors or loss.

## 4.3 System Testing

**Objective:** Validate the complete system functionality.

**Scope:** End-to-end workflow from bin sensors to user notifications and dashboards.

**Tools**: Manual testing with predefined test scenarios.

**Expected Outcome**: The system meets all functional requirements.

## **4**.4. Performance Testing

**Objective**: Evaluate system performance under various conditions.

**Scope**: Response time, data transmission rates, and scalability under load.

**Tools**: Apache JMeter, Locust.

**Expected Outcome**: The system handles peak loads efficiently without performance degradation.

## 4.5 Security Testing

**Objective:** Ensure data protection and system security.

**Scope:** Secure communication, data encryption, and protection against common vulnerabilities.

**Tools**: OWASP ZAP, Burp Suite.

**Expected Outcome**: No unauthorized access or data breaches.

## 4.6. User Acceptance Testing (UAT)

**Objective**: Confirm system usability and satisfaction.

**Scope**: Interaction with user interfaces, notifications, and dashboards by end-users.

**Tools**: Surveys and feedback forms.

**Expected Outcome**: Positive feedback from stakeholders, meeting usability expectations.

## 4.7 Compatibility Testing

**Objective**: Ensure the system operates effectively across various devices and platforms.

**Scope**: Compatibility with browsers, operating systems, and devices.

**Tools**: Browser Stack, manual testing.

**Expected Outcome**: Consistent behavior across all supported platforms.

# Test Environment

* + **Frontend**: Web application with Django templates and Bootstrap for responsive design.
  + **Backend**: Django framework, REST API for communication, and PostgreSQL for data management.
  + **Hardware**: IoT-enabled smart bins equipped with sensors for real-time monitoring.
  + **Tools**: Selenium, Postman, JMeter, and PyTest.

# Test Case Overview

## 6.1. Unit Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case ID | Test Case Description | Steps | Expected Result |
| TC-001 | Bin Status Detection | Upload an image of the bin via the dashboard. | The system detects and displays the bin status. |
| TC-002 | Notification System | Simulate a full bin condition. | Email and SMS notifications are sent successfully. |
| TC-003 | Route Optimization | Request optimized route from the dashboard. | The system generates an optimized collection route. |
| TC-004 | User Registration & Login | Register a new user and log in to the system. | The user is registered and redirected to the dashboard. |
| TC-005 | Admin Panel Access | Log in as an admin and access the admin panel. | Admin can manage bins, routes, and notifications. |

## 6.2. Integration Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case ID | Integration Component | Test Scenario | Expected Result |
| IT-001 | IoT Sensors -> Backend | Simulate a bin status update from IoT sensors. | Bin status updates are reflected in real time. |
| IT-002 | Backend -> Notifications | Trigger a notification for a bin status update. | Notifications are sent via email/SMS. real-time |
| IT-003 | Route Optimization Algorithm -> Dashboard | Request an optimized route via the dashboard. | The dashboard displays the optimized route. |

## 6.3. System Testing

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Input | Expected Output | Actual Output | Pass/Fail | Remarks |
| TC-001 | Verify camera data transmission | The Bin camera reports "75% full" | Camera data is transmitted to the server accurately and displayed on the dashboard. | Camera data received and displayed correctly | Pass | None |
| TC-002 | Test bin status detection | Uploaded image showing a full bin | The Bin status is displayed as "Full" | Bin status displayed correctly | Pass | None |
| TC-003 | Test real-time updates | Bin Camera updates to "50% full" | The dashboard reflects "50% full" status in real-time | Real-time update successful | Pass | None |
| TC-004 | Verify route optimization | Bin locations: A, B, C | Optimal route: A → B → C is generated and displayed | Route generated as expected | Pass | None |
| TC-005 | Test notification system (email) | Bin A status updates to "Full" | Email alert sent to "admin@swms.com" | Email received successfully | Pass | None |
| TC-006 | Test notification system (SMS) | Bin A status updates to "Full" | SMS alert sent to "+254-726-466-335" | SMS received successfully | Pass | None |
| TC-007 | Verify user dashboard functionality | The user logs in and accesses the dashboard | All features (bin status, routes, notifications) are accessible and functional. | All features accessible | Pass | None |
| TC-008 | Verify user authentication | Correct username: "user1", password: "pass1234" | The user successfully logs in | Login successful | Pass | None |
| TC-009 | Verify user authentication (invalid) | Incorrect username: "user2", password: "wrong pass" | The user receives an error message: "Invalid credentials" | Error message displayed | Pass | None |
| TC-010 | Test data analytics functionality | Monthly waste data for bins A, B, C | Report shows trends, peak times, and recycling rates | Accurate report generated | Pass | None |
| TC-011 | Verify mobile responsiveness | Access the dashboard via a smartphone browser | UI adapts to mobile resolution with no loss of functionality | Responsive UI displayed | Pass | None |
| TC-012 | Test system performance under load | 100 simultaneous users access the dashboard | The system remains functional and responsive | No performance issues | Pass | None |
| TC-013 | Verify system security | Attempt unauthorized access to admin data | Access denied, error logged | Unauthorized access denied | Pass | None |
| TC-014 | Test data integrity | The camera picture upload shows the bin is "90% full", and data is sent to a server. | The data displayed matches the data received | Data integrity verified | Pass | None |
| TC-015 | Test error handling | Sensor disconnected during data upload | Error message displayed: "Sensor disconnected. Retrying..." | An appropriate error message is displayed | Pass | None |

## 6.4. Performance Testing

|  |  |  |
| --- | --- | --- |
| Test Case ID | Test Scenario | Expected Outcome |
| PT-001 | Test bin status updates under 100 simultaneous updates. | Updates are processed without delays or errors. |
| PT-002 | Simulate 50 concurrent users accessing the dashboard. | The system responds without crashing or performance degradation. |
| PT-003 | Upload large images for bin status detection. | The system processes the images without timeout errors. |

## 6.5. Security Testing

|  |  |  |
| --- | --- | --- |
| Test Case ID | Test Scenario | Expected Outcome |
| ST-001 | Attempt SQL Injection during login. | The system denies access and logs the attack attempt. |
| ST-002 | Access restricted pages without login. | The user is redirected to the login page. |
| ST-003 | Test data encryption during data transmission. | Data is encrypted and secure during transmission. |

## 6.6. User Interface (UI) Testing

|  |  |  |
| --- | --- | --- |
| Test Case ID | Test Scenario | Expected Outcome |
| UI-001 | Test dashboard responsiveness on different devices. | Dashboard adjusts layout appropriately for mobile, tablet, and desktop. |
| UI-002 | Test navigation links in the navbar. | All navigation links direct users to the correct pages. |
| UI-003 | Verify the visual consistency of the system. | Colors, fonts, and alignment are consistent across all pages. |

### 

## 6.7. Compatibility Testing

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Environment/Configuration | Expected Outcome | Actual Outcome | Pass/Fail | Remarks |
| CT-001 | Test on Windows | OS: Windows 11, Browser: Chrome 114 | The system loads correctly, all features are functional | Features load without issues | Pass | None |
| CT-002 | Test on macOS | OS: macOS Monterey, Browser: Safari 15 | The system loads correctly, all features are functional | Features load without issues | Pass | None |
| CT-003 | Test on Linux | OS: Ubuntu 22.04, Browser: Firefox 103 | The system loads correctly, all features are functional | Features load without issues | Pass | None |
| CT-004 | Test on Android smartphone | Device: Samsung Galaxy S22, Browser: Chrome Mobile | UI adapts to mobile resolution, with no functionality issues | Responsive UI rendered correctly | Pass | None |
| CT-005 | Test on iOS smartphone | Device: iPhone 14, Browser: Safari Mobile | UI adapts to mobile resolution, with no functionality issues | Responsive UI rendered correctly | Pass | None |
| CT-006 | Test on tablets | Device: iPad Air, Browser: Safari | UI adapts to tablet resolution, no functionality issues | Responsive UI rendered correctly | Pass | None |
| CT-007 | Test with multiple screen resolutions | 1920x1080 (Full HD), 1366x768, 768x1024 | The layout adjusts correctly, and no content overflow | Layout scales correctly | Pass | None |
| CT-008 | Test on outdated browser versions | Browser: Internet Explorer 11 | The system displays an error message for an unsupported browser | Error message displayed | Pass | None |
| CT-009 | Test on Edge browser | Browser: Microsoft Edge (Latest Version) | All features functional | Features load without issues | Pass | None |
| CT-010 | Test across different devices simultaneously | Devices: Desktop (Windows), Tablet (iPad), Smartphone (iPhone) | The system performs consistently without conflicts | Features functional across devices | Pass | None |
| CT-011 | Test network performance | Mobile Data (3G, 4G, 5G) | The system remains accessible with varying load times | System accessible | Pass | Slow on 3G (not critical) |
| CT-012 | Test cross-browser compatibility | Browsers: Chrome, Firefox, Edge, Safari | Features load and function correctly in all browsers | Features consistent across browsers | Pass | None |

## Testing Schedule

|  |  |  |
| --- | --- | --- |
| Phase | Timeline | Activities |
| Test Planning | Day 1 - Day 2 | Define testing strategy, and create test cases. |
| Functional Testing | Day 3 - Day 7 | Execute functional test cases. |
| Integration Testing | Day 8 - Day 10 | Test component integration. |
| Performance Testing | Day 11 - Day 13 | Perform load and stress testing. |
| Security Testing | Day 14 - Day 15 | Test for vulnerabilities and data security. |
| UI Testing | Day 16 - Day 17 | Verify UI responsiveness and usability. |
| Bug Fixing & Retest | Day 18 - Day 20 | Resolve bugs and re-execute failed test cases. |

# Conclusion

The Smart Waste Management System has undergone comprehensive testing to ensure its functionality, performance, security, and usability meet the highest standards. The testing process focused on validating all key features, ensuring seamless integration of smart bins, real-time data processing, and route optimization. Rigorous performance and security assessments confirmed the system's reliability under various conditions, while usability testing ensured an intuitive and efficient user experience.

With all critical objectives achieved, the system is now positioned as a robust, secure, and user-friendly solution for modern waste management challenges, ready to optimize operational efficiency and promote sustainability.