

Software Requirements Specification (SRS)

Uasin Gishu Smart Waste Monitoring System

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to define the requirements, functionalities, and design of the Uasin Gishu Smart Waste Monitoring System (UGSWMS). The system seeks to modernize waste management within Uasin Gishu County, beginning with Eldoret town, by utilizing IoT simulation, real-time dashboards, and digital payment systems.

It will allow administrators to monitor bin fill levels, allocate nearby trucks for collection, and process payments through the M-Pesa Daraja API. The platform will consist of both web and mobile applications to ensure accessibility for administrators, drivers, and residents.

1.2 Scope

The UGSWMS is designed to improve efficiency, accountability, and sustainability in waste management. It will simulate IoT-enabled smart bins that update fill levels dynamically and notify the admin when a bin is full.

Administrators can view all bins and trucks on an integrated Google Maps dashboard and assign the nearest truck to empty full bins. Residents and businesses can make waste management payments through M-Pesa (Till or Paybill) using the Daraja API.

An AI chatbot will be included to provide assistance and handle queries, while real-time reports and analytics will track performance and revenue. The system will first be deployed in Eldoret and later scaled across Uasin Gishu County.

1.3 Definitions, Acronyms, and Abbreviations

- **UGSWMS** – Uasin Gishu Smart Waste Monitoring System
 - **IoT** – Internet of Things
 - **API** – Application Programming Interface
 - **RBAC** – Role-Based Access Control
 - **AI** – Artificial Intelligence
 - **NLP** – Natural Language Processing
 - **M-Pesa Daraja API** – Safaricom's API for mobile money integration
-

2. Overall Description

2.1 Product Perspective

The system will serve as a centralized digital platform connecting residents, drivers, and administrators. It will simulate IoT-enabled waste bins that report their fill levels in real time. When a bin reaches a specified threshold, the system will trigger an alert on the dashboard. The administrator can view all bins and trucks on a Google Maps interface, identify full bins, and assign the nearest truck for collection.

Residents will use the mobile app to report issues, view collection schedules, and make payments. The system backend will run on **Firebase**, while the frontend will be developed using **Flutter**, ensuring both mobile and web compatibility.

2.2 Product Functions

Key system functions include:

- Simulating bin fill levels using dynamic IoT data.
- Displaying bins and trucks on Google Maps with real-time updates.
- Triggering alerts when bins exceed 80% capacity.
- Allowing administrators to assign nearby trucks for collection.
- Enabling residents and businesses to make digital payments via M-Pesa Daraja API.
- Providing AI chatbot assistance for reporting and support.
- Generating reports on collection performance and payments.
- Supporting multiple user roles (Admin, Driver, Resident).

2.3 User Characteristics

- **Administrator:** Manages bins, drivers, and routes; monitors the dashboard; assigns tasks; and generates reports.
- **Driver:** Receives collection assignments, updates pickup status, and navigates routes using Google Maps.
- **Resident/Business User:** Reports waste issues, tracks bin collection status, and makes service payments.

2.4 Constraints

- Real IoT devices will not be used — simulated data will represent bin activity.
- System relies on internet connectivity and Firebase uptime.
- Limited API usage due to free-tier restrictions on Google Maps and Firebase.
- Payment operations will initially use the Daraja API sandbox environment for testing.

2.5 Assumptions and Dependencies

- Users will access the system via internet-connected devices.
- Firebase, Google Maps, and Daraja APIs will remain accessible under free-tier conditions.
- Drivers and admins will have GPS-enabled smartphones.

3. Specific Requirements

3.1 Functional Requirements

User Authentication and Roles

- The system shall allow secure login and logout for all users.
- Role-Based Access Control (RBAC) shall manage permissions for Admins, Drivers, and Residents.

Driver Management

- The system shall store driver details such as name, ID, phone, vehicle, assigned route, and availability status.
- Admins shall assign routes and update driver availability.

IoT Bin Simulation

- The system shall simulate gradual bin fill levels using dynamic data updates every 5–10 seconds.
- Each bin's fill rate shall vary to reflect different waste-generation zones (e.g., residential, market, industrial).
- The system shall apply thresholds to classify bins as Empty (0–30%), Half Full (31–80%), and Full (81–100%).
- Alerts shall trigger only after a bin remains above 80% capacity for several consecutive updates to avoid false notifications.
- Once a bin is assigned for pickup, alert repetition shall be suspended until the bin is cleared or refilled.
-

Real-Time Dashboard

- The admin dashboard shall display bins with color indicators: Empty, Half Full, Full.
- The dashboard shall update smoothly to reflect gradual changes in bin fill levels without abrupt transitions.
- The dashboard shall automatically refresh in real time without requiring manual page reloads.
- Alerts shall appear only when a bin meets or exceeds the defined persistence threshold.

Truck Allocation

- Admins shall assign the nearest available truck to full bins.
- Truck and bin locations shall be shown dynamically on Google Maps.
- The system shall compute and suggest the three closest trucks to each full bin using real-time coordinates.

Route Tracking and Visualization

- The system shall integrate Google Maps API to display bin and truck positions.
- Routes shall update dynamically based on driver assignments and changes in bin status.
- Drivers shall view optimized pickup routes on their mobile interface.

Digital Payments

- Residents and businesses shall make payments via M-Pesa (Till).
- Payments shall be processed securely through the Safaricom Daraja API.
- All transaction details (amount, user, timestamp, and status) shall be stored securely in Firebase.
- The system shall support transaction verification and payment receipts.

AI Chatbot

- The chatbot shall assist users in reporting issues, answering FAQs, and navigating system features.
- It shall use Hugging Face NLP models to provide contextual and accurate responses.
- The chatbot shall handle both resident and admin-level queries related to waste management.

Notifications

- Drivers shall receive push notifications for new collection tasks and assigned bins.
- Admins shall be notified of completed pickups, system alerts, and received payments.
- Notifications shall be rate-limited to prevent spam during frequent updates.

Reports and Analytics

- The system shall generate reports on revenue, bin collection frequency, driver performance, and service efficiency.
 - Reports shall be exportable in multiple formats (PDF, Excel).
 - Admins shall view graphical dashboards showing waste trends and system activity over time.
-

3.2 Non-Functional Requirements

Performance

- IoT data shall update every 5–10 seconds, simulating realistic bin fill patterns.
- The system shall throttle database writes to maintain efficiency while ensuring near real-time responsiveness.
- Dashboard updates shall occur automatically, with smooth visual transitions between bin states.
- System response time for user actions (login, payment, data load) shall not exceed three seconds under normal conditions.

Scalability

- The architecture shall support integration with real IoT sensors in future deployments.
- The system shall allow expansion to additional towns and counties without significant redesign.
- Database and backend resources shall auto-scale to handle increased data loads.

Security

- All communications shall be encrypted using HTTPS.
- Firebase Authentication shall ensure secure login and access control.
- Sensitive user and payment data shall be encrypted both in transit and at rest.
- Access rights shall be restricted according to RBAC permissions.

Usability

- The interface shall be responsive, simple, and intuitive across both web and mobile platforms.
- Icons, color codes, and visual indicators shall ensure quick understanding of bin statuses.
- Tooltips and chatbot support shall guide users and simplify onboarding.

Availability

- The system shall maintain at least **95% uptime** during regular operations.
- Firebase cloud infrastructure shall ensure reliability, redundancy, and automatic backups.
- Offline caching shall allow users temporary access to basic features during connectivity interruptions.

Maintainability

- The codebase shall follow modular programming practices for easy debugging and future upgrades.
- Comprehensive documentation shall be provided for all modules and APIs.
- The system shall support version control and continuous integration for updates.

Compatibility

- The system shall be compatible with major browsers (Chrome, Edge, Firefox) and Android OS version 8.0 or higher.
- APIs shall follow RESTful standards to enable integration with third-party systems.

Reliability

- The system shall recover automatically from temporary network or service interruptions.
 - Regular data backups shall prevent data loss.
 - IoT simulation shall continue seamlessly during normal uptime, with accelerated-time mode available for demos.
-

4. Conclusion

The Uasin Gishu Smart Waste Monitoring System provides a sustainable, efficient, and scalable solution to waste management challenges through IoT simulation, real-time dashboards, AI-powered communication, and secure mobile payments. By integrating cloud services and open APIs, the system lays the foundation for future expansion and real IoT device integration across the county.