Contents

Aonenawo Software Requirements Specification (SRS)

1. Introduction

This SRS document describes the functional and non-functional requirements for Aonenawo. Its purpose is to specify what the system must accomplish (functional) and how the system performs (non-functional). This SRS will be used by developers to implement Aonenawo and by testers to validate the system. It will also help communicate the design to stakeholders and ensure everyone has the same understanding.

1.1 Scope

Aonenawo is a two-sided marketplace and community platform. It allows buyers (students and others) to browse products, interact in forums, and purchase goods from sellers (campus and local businesses). It includes an escrow-based checkout with integrated delivery by student couriers. The initial deployment is web-only (responsive design) targeting UNIMA, with future mobile app versions planned. Key subsystems include user accounts, shop and product catalog, community discussion boards, shopping cart and payment, order management, courier dispatch, and administrative tools.

1.2 Definitions and Acronyms

- User: Any person with an account (student or non-student).
- Seller: A user who registers a business/shop.
- Courier: A user who delivers items (often students); also called delivery personnel.
- Escrow: Holding buyer's payment until delivery conditions are met.
- Community: A discussion group centered on a topic; similar to a forum.
- Post: A user-generated message in a community or personal feed.
- Airtel Money/TNM Mpamba: Malawi's mobile payment services.

2. Overall Description

2.1 Product Perspective

Aonenawo is a standalone web application. It integrates with external services for payments. The user interface is browser-based, supporting modern browsers and mobile devices. Data will be stored on a central server/database.

The system comprises:

- Frontend (Web UI)
- Backend (Server)
- Database
- Payment Gateway Interfaces
- Notification Service
- **Third-Party Services:** (Optional) SMS gateway for OTPs; mapping (e.g., Google Maps API) if showing courier locations.

2.2 User Needs

- Buyers (normal users): Need to browse products and shops, filter by category/price, add to cart, checkout with local payment options, and track orders. They want to interact socially (post updates, comment, follow shops and topics).
- Sellers (business users): Need to register and verify a business, create/manage a shop profile, upload products (with images, descriptions, prices), manage inventory, and view incoming orders.
- Couriers: Need to register as couriers (with ID verification), set availability schedules, receive delivery requests, accept orders, update order status (picked up, delivered), and view earnings.
- Administrators: Need dashboards to manage default communities, moderate content (posts, comments), onboard/verify couriers, handle disputes/refunds, and maintain system configurations.

2.3 Assumptions and Dependencies

- Assumptions: Users have basic internet access on web-enabled devices. Payment services (Airtel Money, Mpamba, banks) are operational and have public APIs. Couriers are students with flexible schedules. Sellers outside Zomba can send to Aonenawo's address in Zomba using local means like CTS and Speed courier.

- Dependencies: Reliable connectivity to payment gateways and any notification services. Compliance with Malawi's e-commerce regulations.

3. System Features and Requirements

3.1 Functional Requirements

The system shall provide the following functionalities:

3.1.1 User Registration & Authentication:

- Users must sign up with email/mobile and password (or social login).
- Upon first login, users indicate if they are registering a *business*. If yes, they are prompted for business details (name, address, type, registration number) before proceeding.
- Students may be asked to verify student status (e.g. university email).
- The system shall support password reset and secure login.

3.1.2 User Roles and Profiles:

- **Buyer (Student/Non-Student):** Has a personal profile showing their posts, followed shops, and joined communities. Can create and edit posts (text/images) on personal feed or in communities.
- **Seller (Business User):** In addition to the above, has a shop profile (name, description, location, logo) and a dashboard to manage product listings and view sales/orders. The shop owner's personal profile and shop are linked to the same account.
- **Courier:** Has a courier profile and dashboard to set availability (days/times). Courier dashboard shows assigned deliveries, statuses, and earnings.

- **Admin:** Has privileged access to moderate content and manage system settings.

3.1.3 Product Catalog & Search:

- Sellers can create/edit products with name, category, price, description, images, and stock quantity.
- Buyers can browse products by category, search by keywords, and apply filters (price range, category, seller rating, etc.).
- Product pages display details and “Add to Cart” option.

3.1.4 Shopping Cart & Checkout:

- Users can add multiple products to a shopping cart and modify quantities.
- At checkout, user reviews items and selects delivery options.
- Payment integration: Users pay via Airtel Money, TNM Mpamba, or VISA debit/credit (via integrated payment gateway).
- Upon successful payment, an order is created in *Pending* state, and funds are held in escrow by Aonenawo. The seller and the user both receive order confirmation notifications.

3.1.5 Order Management:

- **Seller’s View:** Seller sees new orders in their dashboard, including item details and buyer info. The seller marks the item as ready for pickup. If the seller is on-campus or in Zomba, the order is flagged for Aonenawo courier pickup. If off-campus, seller ships item to Aonenawo address, then marks it ready.
- **Courier Assignment:** The system automatically notifies available couriers of orders based on location and schedule. Couriers can accept or decline orders.
- **Delivery Tracking:** Once a courier accepts, they update order status: *Picked Up* when item is collected, and *Delivered* when handed to buyer.
- **Post-Delivery:** Upon delivery confirmation, escrowed funds (minus courier commission, if any) are released to the seller’s account. The system notifies buyer and seller of completion.

3.1.6 Escrow and Refunds:

- Aonenawo holds payment until delivery. If the buyer reports an issue or if the seller fails to deliver in a reasonable time, the system triggers a refund workflow. Admin can approve refunds, returning funds to buyer and releasing seller.

3.1.7 Communities and Social Features:

- **Communities:** The platform includes default communities (e.g. #General, #Books, #DormDeals). Users can join or leave communities. Any user can create a new community (with admin approval) by specifying a name and description.
- **Topics & Posts:** Within each community, users can start topics and post messages (text/images). Other users can comment and react (like, etc.) on these posts.
- **Personal Feed:** Users can make personal posts on a general feed (viewable by their followers or community members). Posts support comments and reactions.

3.1.8 Following and Notifications:

- Users can follow shops or other users. Following a shop means the user sees new products or announcements from that shop in their feed.
- Users can opt to follow community topics.
- System sends notifications (in-app or email/SMS) for important events: new order, delivery status changes, community replies, etc.

3.1.9 Profiles:

- Every user has a profile page listing their posts, community memberships, shops followed, and (if applicable) their shop or courier statistics.
- Sellers' profiles show their shop(s) and products.
- Couriers' profiles (internal) show completed deliveries and earnings.

3.1.10 Courier Management:

- Couriers register by agreeing to terms (including background check via student ID/transcript). Admin verifies credentials.
- Couriers set weekly availability slots (e.g. Mon/Wed 9am–5pm).

- System matches orders to couriers who are available and within range.
- Couriers earn a set commission per delivery; the system tracks earnings per courier and total deliveries completed.

3.1.11 Administrative Functions:

- Admins can create/edit default communities and moderate user-generated content (remove spam, ban users).
- Admins handle disputes: they can initiate refunds or adjust balances.
- Admins oversee courier onboarding and can deactivate couriers who violate policies.

3.2 External Interface Requirements

3.2.1 Payment Gateway

The system will integrate APIs from local payment providers (Airtel Money, TNM Mpamba) and VISA card payments (via banks or services like PayChangu and OneKhusa). These interfaces must securely process transactions and notify the platform of payment results.

3.2.2 Mobile Money APIs

Implementation may use a unified gateway (e.g. PayChangu or OneKhusa) to handle multiple mobile money schemes seamlessly.

3.2.3 SMS/Email Services

For sending OTPs, alerts, and notifications.

3.2.4 Mapping/Geolocation (optional)

If displaying courier routes or shop locations on a map, a mapping API (Google Maps or similar) may be used.

3.2.5 Hardware/Software

The system runs on web servers and requires modern web browsers. No special hardware is needed for users.

3.3 Non-Functional Requirements

- **Performance:** The system should be responsive. For example, 95% of page requests should load within 2 seconds under normal load. The platform should handle at least 500 concurrent users without degradation.
- **Security:** All user data (credentials, payments) must be encrypted (HTTPS/TLS). Only authenticated users can access their data. Admin and seller interfaces require strong authentication. Payment data must comply with payment industry standards.
- **Reliability & Availability:** Aim for 99% uptime. The system should gracefully recover from failures and maintain data integrity.
- **Usability:** Interfaces should be intuitive and mobile-friendly. A user onboarding tutorial or help section will guide new users. Documentation for sellers and couriers will be available.
- **Scalability:** The design should allow horizontal scaling of the application and database to accommodate growth to thousands of users as Aonenawo expands beyond campus.
- **Localization:** All text is in English (Malawi's official language for higher education), prices in MKW. Currency formatting and date/time adhere to local conventions.
- **Legal/Regulatory Compliance:** The system must comply with Malawi's Electronic Transactions Act, protect customer data per privacy laws, and use escrow/holding funds in accordance with Payment Systems Act.
- **Maintainability:** Code should be modular and documented. Use version control (e.g., Git) for source code, and have a clear process for bug fixes and updates.