System Decomposition

Overview

The system follows a **Layered Architecture** comprising three main layers:

1. **Presentation Layer** (Frontend)
2. **Application Layer** (Backend API)
3. **Data Layer** (Database)

Each layer is further broken down into functional components/modules with specific responsibilities. Below is the decomposition of the system components, particularly highlighting the backend's role.

System Components

1. User Management Module

Layer: Backend (Application Layer)
Responsibilities:

- Handle user registration and authentication.
- Manage user profiles (shop owners, customers, government officials).
- Manage user roles and permissions.

2. Shop Registration & Verification Module

Layer: Backend
Responsibilities:

- Register spaza shops.
- Allow shop owners to upload verification documents.
- Notify government officials for verification.
- Update shop verification status.

3. Product Stocking and Supplier Comparison Module

Layer: Backend
Responsibilities:

- Display supplier list and products.
- Compare product prices.
- Help shop owners stock their stoks/products efficiently.

4. Complaint & Review System

Layer: Backend
Responsibilities:

- Allow customers to submit reviews and complaints.
- Authenticate complaints using government portal.
- Use AI-based system to flag or prioritize complaints.

5. Messaging & Notification Module

Layer: Backend
Responsibilities:

- Integrate WhatsApp or SMS or email API for user notifications.
- In-app notifications for verification updates, complaint statuses, etc.

6. Admin Dashboard Module

Layer: Backend
Responsibilities:

- Allow government users to view complaints and verifications.
- Flag fake reports.
- View system usage and analytics.

Component Interaction

Each module communicates with:

- The **Frontend** via RESTful API endpoints.
- The **Database** via an ORM or direct SQL queries.
- External services (e.g., WhatsApp/ email API) using HTTP requests or SDKs.

This modular decomposition ensures separation of concerns, maintainability, and scalability.