

Here is the **visual interaction flow diagram** for the **Kalemie Agro Poultry Management System**:

**🔗 Module Dependency Summary**

Here’s how the modules depend on each other:

**1. Core System Components**

* **Required by all modules** for authentication, user roles, and configuration.

**2. API Layer**

* **Connects** Web App, Mobile App, and Public Website to backend modules.
* **Used by** IoT Sensors, Admin Console, and external integrations.

**3. Admin Console**

* **Depends on** Core System for user and role management.
* **Controls** access to HR, Finance, Inventory, and Operations.

**4. Operations**

* **Depends on** Inventory (for feed, medicine).
* **Feeds data to** Analytics & Reporting.
* **Receives input from** IoT Sensors.

**5. Commerce & Inventory**

* **Linked to** Operations (egg/meat production).
* **Feeds data to** Finance and Analytics.

**6. HR & Finance**

* **Depends on** Inventory and Operations for cost tracking.
* **Feeds data to** Analytics.

**7. Analytics & Reporting**

* **Central module** that pulls data from all others.
* **Used by** Admin Console, Web App, and Mobile App.

**8. IoT Sensors & Integration**

* **Feeds real-time data** to Operations and Analytics.
* **Triggers alerts** via Mobile App and Web App.

**9. Web App & Mobile App**

* **Depend on** API Layer for data access.
* **Used by** staff and managers to interact with all modules.

**10. Public Website**

* **Uses API Layer** to display public data (products, blog).
* **Independent** from internal modules but shares branding.

A

**Highlights of the Chart**

* **Core System Components** is the backbone, supporting all other modules.
* **API Layer** connects frontend interfaces (Web App, Mobile App, Public Website) with backend modules.
* **Admin Console** manages access and configuration for HR, Inventory, and Operations.
* **Operations** interacts heavily with Inventory, HR, IoT Sensors, and Analytics.
* **Analytics & Reporting** aggregates data from nearly all modules and feeds insights to the Web and Mobile Apps.
* **IoT Sensors** provide real-time data to Operations and Analytics.
* **Users** access the system via Web App, Mobile App, or Public Website.

A

The **Core System Components** module is the **foundation** of the Kalemie Agro Poultry Management System. It provides essential services and infrastructure that all other modules rely on. Here's a detailed explanation of its role and responsibilities:

**🔧 Core System Components – Key Functions**

**1. Authentication & Authorization**

* Manages user login, logout, and session control.
* Implements **Role-Based Access Control (RBAC)** to ensure users only access what they’re permitted to (e.g., farm workers vs. managers vs. admins).

**2. User Management**

* Handles creation, editing, and deletion of user accounts.
* Assigns roles and permissions.
* Tracks user activity and audit logs.

**3. System Configuration**

* Stores global settings (e.g., language preferences, timezone, branding).
* Manages environment variables and system-wide parameters.

**4. Security & Compliance**

* Ensures secure data handling (encryption, password policies).
* Supports GDPR-like compliance features (data access, deletion requests).

**5. Error Handling & Logging**

* Centralized logging of system errors and events.
* Helps in debugging and monitoring system health.

**6. Notifications & Messaging**

* Provides internal messaging between modules.
* Supports alerts, reminders, and system-wide announcements.

**7. Multilingual Support**

* Enables English/French interface switching.
* Ensures consistent language handling across modules.

**🔗 Dependencies**

All other modules depend on Core System Components for:

* Secure access
* User identity and permissions
* System-wide settings
* Logging and notifications

For example:

* **Operations** needs user roles to assign tasks.
* **Analytics** pulls user activity logs.
* **Mobile App** uses authentication to sync data.
* **Admin Console** uses it to manage users and roles.

Would you like a Django model structure or code snippet showing how this module could be implemented?

Category, product, customer, customer\_order, ordered\_product, payment : cash/credit/check