

SOFTWARE DESIGN DOCUMENT

BY G-26

COLLEGE OF COMPUTING AND INFORMATION SCIENCES

COFFEE SUPPLY CHAIN MANAGEMENT SYSTEM

DESIGN DOCUMENT FOR COFFEE SUPPLY CHAIN MANAGEMENT SYSTEM

GitHub Link: https://github.com/syny-pulse/coffee-chain.git

G-26

Supervisor: Mary Nsabagwa

|  |  |
| --- | --- |
| Name | Registration Number |
| Kisomose Arnold Patrick | 23/U/10079/EVE |
| Mugisha Modern | 23/U/11587/PS |
| Kabalebe Joshua | 23/U/27067/EVE |
| Kukiriza Sinai | 23/U/0646 |
| Wandera Hannington | 23/U/23796 |

**Table of Contents**

[1. Introduction 5](#_Toc199943088)

[1.1 Purpose 5](#_Toc199943089)

[1.2 Scope 5](#_Toc199943090)

[1.2.1 Goals and Objectives 5](#_Toc199943091)

[1.3 Document Overview 7](#_Toc199943092)

[2. System Overview 10](#_Toc199943093)

[3. System Architecture 12](#_Toc199943094)

[3.1 Architectural Design 12](#_Toc199943095)

[3.2 Decomposition Description 13](#_Toc199943096)

[3.3 Design Rationale 17](#_Toc199943097)

[4. Data Design 18](#_Toc199943098)

[4.1 Data Description 18](#_Toc199943099)

[4.2 ERD 25](#_Toc199943100)

[4.3 Data Dictionary 25](#_Toc199943101)

[5. Component Design 35](#_Toc199943102)

[6. Human Interface Design 39](#_Toc199943103)

[6.1 Overview of User Interface 39](#_Toc199943104)

[6.2 Screen Images 39](#_Toc199943105)

[6.3 Screen Objects and Actions 40](#_Toc199943106)

**List of tables**

[Table 4. 1 Users 18](#_Toc199942959)

[Table 4. 2 Companies 19](#_Toc199942960)

[Table 4. 3 FarmerHarvest 20](#_Toc199942961)

[Table 4. 4 ProductRecipes 20](#_Toc199942962)

[Table 4. 5 FarmersOrders 21](#_Toc199942963)

[Table 4. 6 ProcessorRawMaterialInventory 21](#_Toc199942964)

[Table 4. 7 ProcessorFinishedGoodsInventory 22](#_Toc199942965)

[Table 4. 8 RetailerOrders 22](#_Toc199942966)

[Table 4. 9 RetailerOrderItems 23](#_Toc199942967)

[Table 4. 10 Employees 23](#_Toc199942968)

[Table 4. 11 Messages 24](#_Toc199942969)

[Table 4. 12 InventoryTransactions 24](#_Toc199942970)

[Table 4. 13 Users 25](#_Toc199942971)

[Table 4. 14 Companies 26](#_Toc199942972)

[Table 4. 15 FarmerHarvest 27](#_Toc199942973)

[Table 4. 16 Product Recipes 28](#_Toc199942974)

[Table 4. 17 FarmerOrders 28](#_Toc199942975)

[Table 4. 18 ProcessorRawMaterialInventory 29](#_Toc199942976)

[Table 4. 19 ProcessorFinishedGoodsInventory 30](#_Toc199942977)

[Table 4. 20 RetailerOrders 30](#_Toc199942978)

[Table 4. 21 RetailerOrderItems 31](#_Toc199942979)

[Table 4. 22 Employees 32](#_Toc199942980)

[Table 4. 23 Messages 33](#_Toc199942981)

[Table 4. 24 InventoryTransactions 33](#_Toc199942982)

**List of Figures**

[Figure 3. 1 Component diagram for the Coffee Supply System 15](#_Toc199944828)

[Figure 3. 2 Use-case diagram for the Coffee Supply System 16](#_Toc199944829)

[Figure 3. 3 Sequence diagram for the Coffee Supply System 17](#_Toc199944830)

[Figure 4. 1 ERD for the Coffee Supply Database 25](#_Toc199944951)

[Figure 6. 1 Login Page 39](#_Toc199945994)

# Introduction

## Purpose

This Software Design Document (SDD) describes the architecture and system design of the Coffee Supply Chain Management System. The document is intended for stakeholders involved in the development and implementation of the three-tier coffee supply chain platform. This document serves as the primary reference for code development and system implementation, containing all necessary information required by programmers to develop the system components that connect coffee farmers, processors, and retailers in an integrated platform leveraging machine learning for demand prediction and supply chain optimization.

## Scope

The Coffee Supply Chain Management System is designed to streamline the coffee supply chain from farmers in Bukomansimbi to retailers such as Café Javas and Java House. It connects three tiers—farmers (Tier 1), processors (Tier 2), and retailers (Tier 3)—to optimize production, inventory, and distribution processes. The system leverages machine learning (ML) for demand prediction, customer segmentation, workforce distribution in processing facilities, and raw material optimization, alongside features for inventory management, order processing, stakeholder communication, and automated vendor validation.

The system manages the complete coffee lifecycle including Arabica and Robusta varieties with three processing methods (Natural, Washed, Honey) at the farmer level, transformation into four product categories (Drinking Coffee, Roasted Coffee, Coffee Scents, Coffee Soap) at the processor level, and distribution through various retail channels.

### Goals and Objectives

#### Primary Goals

* Develop an intelligent three-tier coffee supply chain management platform that connects farmers, processors, and retailers in an integrated ecosystem
* Implement machine learning capabilities to predict retail demand and optimize production planning from retail sales back to raw material production
* Create a unified communication platform that enables seamless collaboration between all supply chain participants
* Establish an automated vendor validation system using Java server technology that ensures quality and compliance throughout the network

#### 1.2.1.1 Specific Objectives

* Achieve demand prediction accuracy of at least 80% using historical sales data and ensemble machine learning algorithms (ARIMA, LSTM, Random Forest) to optimize raw material requirements and prevent stockouts.
* Reduce order processing time by 50% through automated workflows and real-time inventory tracking across all three supply chain tiers.
* Minimize inventory holding costs by 25% with dynamic reorder points and safety stock calculations.
* Establish seamless real-time communication channels between farmers, processors, and retailers to reduce response times and improve collaboration.
* Enhance customer satisfaction through personalized recommendations based on segmentation.
* Support 100+ concurrent users with 99.5% uptime and real-time inventory updates (< 5 seconds).
* Provide mobile-responsive interfaces for farmers in rural areas.
* Create seamless data flow where retail sales patterns directly influence processor production schedules and farmer planting decisions through automated demand propagation.
* Implement customer segmentation algorithms using RFM analysis that identify 4-6 distinct customer groups (Premium Coffee Enthusiasts, Daily Coffee Consumers, Seasonal Buyers, Price-Sensitive Shoppers) to enable targeted marketing and improve customer satisfaction by 20%.
* Convert product demand forecasts into precise raw material requirements using recipe decomposition algorithms, optimizing farmer production planning by variety, processing method, and grade.
* Implement automated Java-based vendor validation system that processes applications 60% faster than manual processes while maintaining rigorous quality and compliance standards
  1. Document Overview

This document has been partitioned into different chapters with each describing a particular aspect of the coffee supply chain system design and development as follows:

**Chapter 1. Introduction:** Outlines the purpose of the SDD, its intended audience and the scope of the three-tier coffee supply chain system including the goals and objectives of connecting farmers, processors, and retailers through intelligent demand prediction.

**Chapter 2. System Overview:** Gives a general description of the coffee supply chain functionality, context and design of the system, providing background information on coffee supply chain management and the role of machine learning in optimizing operations from retail demand to farmer production.

**Chapter 3. System Architecture:** Describes the components and modules in the three-tier system and the interaction between them, providing a decomposition of the system into subsystems and the rationale for selecting the layered architectural pattern with Laravel framework and Java vendor validation server.

**Chapter 4. Data Design:** Describes how the coffee supply chain entities are stored, processed, organized using MySQL relational database and has a list of the entities along with their types and descriptions including users, companies, farmer harvests, product recipes, orders, inventory, employees, and communication data.

**Chapter 5. Component Design:** Gives a drill down of the system into different components, modules and sub-components, providing detailed pseudocode algorithms for user authentication, multi-tier inventory management, demand prediction with ML models, order processing between tiers, recipe management, and vendor validation functionalities.

**Chapter 6. User Interface Design:** This describes how the different users (farmers, processors, retailers) view and use the system to perform their desired tasks, includes the screenshots of the interface from the perspective of different user roles and describes the screen images and their related actions for coffee supply chain management.

* 1. **Reference Material**
* Laravel Framework Documentation (v10.x)
* MySQL Database Documentation (v8.0)
* International Coffee Organization (ICO): <https://www.ico.org>
* MIT Supply Chain Dataset: <https://supplychain.mit.edu>
* Kaggle Retail Datasets: <https://www.kaggle.com>
* NOAA: <https://www.noaa.gov>
* FAO: <https://www.fao.org>
* FRED: <https://fred.stlouisfed.org>
  1. **Definitions and Acronyms**

|  |  |
| --- | --- |
| SCM | Supply Chain Management |
| ML | Machine Learning |
| CRUD | Create, Read, Update, Delete |
| MVC | Model-View-Controller |
| ORM | Object-Relational Mapping |
| JSON | JavaScript Object Notation |
| HTTP | Hypertext Transfer Protocol |
| RBAC | Role-Based Access Control |
| ICO | International Coffee Organization |
| ARIMA | AutoRegressive Integrated Moving Average |
| LSTM | Long Short-Term Memory (neural network) |
| RFM | Recency, Frequency, Monetary (customer analysis) |
| API | Application Programming Interface |
| FK | Foreign key |
| UI | User Interface |
| PK | Primary key |
| PDF | Portable Document Format |

# System Overview

The Coffee Supply Chain Management System is a web-based platform designed to streamline and optimize the complete coffee supply chain from raw material production at farms in regions like Bukomansimbi to final product sales at retail outlets like Café Javas and Java House. The system manages three distinct tiers: farmers producing Arabica and Robusta coffee with various processing methods (Natural, Washed, and Honey), processors transforming raw materials into finished products (Drinking Coffee, Roasted Coffee, Coffee Scents, Coffee Soap) and retailers distributing products to end consumers.

The system leverages ensemble machine learning algorithms (ARIMA, LSTM, Random Forest) to predict retail demand and translate these forecasts into precise raw material requirements through recipe decomposition. The system uses real-world data from sources like ICO, NOAA, and FAO to ensure accurate ML predictions and analytics.

The platform supports role-based access for farmers, processors, retailers, and administrators, each with specific dashboards and functionalities tailored to their operational needs. The system has a responsive interface for both desktop and mobile users.

Key system capabilities include demand-driven production planning where retail sales patterns directly influence farmer production decisions, comprehensive recipe management linking raw materials to finished products with precise conversion ratios, multi-tier inventory management with automated reorder points and safety stock calculations, real-time communication platform enabling collaboration between all stakeholders, customer segmentation using RFM analysis for personalized marketing, automated workforce distribution management in processing facilities, intelligent vendor validation using Java server technology with facility visit scheduling, and comprehensive analytics dashboards providing stakeholder-specific insights for data-driven decision making.

The business process flow follows a demand-driven approach:

Retailers → Sales Data → ML Prediction Engine → Processor Demand Forecast → Farmer Production Planning

↓ ↑ ↑ ↑ ↑

End Consumers ← Products ← Processed Goods ← Raw Materials ← Coffee Production

, creating a seamless flow where end consumer demand drives upstream production decisions. The system transforms traditional supply chain management by creating an ecosystem where inventory levels are optimized across all tiers, communication flows seamlessly between stakeholders, and machine learning predictions enable proactive rather than reactive supply chain management.

The system is built using modern web technologies with Laravel serving as the primary backend framework, MySQL as the relational database for managing complex supply chain relationships, Java server technology for automated vendor validation processes, and integrated Python-based machine learning models for predictive analytics. The architecture ensures scalability to handle multiple concurrent users across all tiers, security through role-based access control and encrypted communications, and maintainability through modular design patterns.

# System Architecture

* 1. Architectural Design

The system follows a layered architectural pattern with clear separation of concerns optimized for coffee supply chain management. The architecture consists of five primary layers: Presentation Layer (Laravel Blade templates and responsive frontend components for farmers, processors, and retailers), Application Layer (Laravel controllers and middleware handling business logic), Machine Learning Layer (Python-based ensemble models for demand prediction and customer segmentation), Business Logic Layer (Laravel services and models managing supply chain operations), and Data Access Layer (Laravel Eloquent ORM and MySQL database with coffee-specific schema).

The system is decomposed into seven major subsystems that collaborate to manage the complete coffee supply chain: User Management Subsystem handles authentication, authorization, and role-based access for farmers, processors, retailers, and administrators; Multi-Tier Inventory Management Subsystem manages stock levels for raw materials at farmer level and finished products at processor/retailer levels; Order Processing Subsystem handles order creation, tracking, and fulfillment between all three tiers; Recipe Management Subsystem maintains product formulations linking raw materials to finished products with conversion ratios; Analytics and ML Subsystem provides demand prediction, customer segmentation, and raw material optimization; Communication Subsystem manages real-time messaging and notifications between stakeholders; and Vendor Management Subsystem handles automated validation using Java server technology and facility visit scheduling.

These subsystems collaborate through well-defined APIs and shared data models specific to coffee supply chain operations. The User Management Subsystem provides authentication services to all other subsystems with role-based permissions. The Multi-Tier Inventory Management Subsystem feeds data to the Analytics Subsystem for demand prediction and raw material optimization. The Recipe Management Subsystem enables conversion of product demand into raw material requirements. The Order Processing Subsystem integrates with inventory management across all tiers and communication systems for stakeholder notifications. The Analytics Subsystem provides demand forecasts and optimization insights to all operational subsystems.

## Decomposition Description

**User Management Subsystem:** Contains User entity, Company entity and Authentication Controller. Manages user registration for farmers, processors, and retailers, role-based authentication, company profile management, and access control using Laravel's built-in authentication with custom role definitions for supply chain participants.

**Multi-Tier Inventory Management Subsystem:** Includes FarmerHarvest entity, ProcessorRawMaterial entity, ProcessorFinishedGoods entity, and Inventory Controller. Handles raw material tracking by coffee variety, processing method, and grade at farmer level; raw material and finished product inventory at processor level; and finished product inventory at retailer level with automated reorder calculations.

**Order Processing Subsystem:** Comprises FarmerOrder entity, RetailerOrder entity, RetailerOrderItem entity, and Order Controller. Manages order lifecycle between farmers and processors for raw materials, and between processors and retailers for finished products, including supplier coordination, delivery tracking, and automated status updates.

**Recipe Management Subsystem:** Contains ProductRecipe entity, and Recipe Controller. Maintains detailed formulations for all four product categories (Drinking Coffee, Roasted Coffee, Coffee Scents, Coffee Soap) with precise requirements for coffee varieties, processing methods, grades, and conversion ratios from raw materials to finished products.

**Analytics and ML Subsystem:** Contains SalesData entity, CustomerSegment entity, DemandForecast entity, ML Service, and Analytics Controller. Implements ensemble machine learning models (ARIMA, LSTM, Random Forest) for demand prediction, RFM-based customer segmentation, and raw material optimization algorithms that translate product demand into precise farming requirements.

**Communication Subsystem:** Includes Message entity, Notification entity, and Communication Controller. Provides real-time messaging capabilities between farmers, processors, and retailers with context-aware notifications for order updates, quality feedback, and demand forecast sharing.

**Vendor Management Subsystem:** Contains VendorApplication entity, ValidationRule entity, FacilityVisit entity, and Vendor Controller implemented with Java server technology. Manages automated vendor application processing, financial and compliance validation checks, reputation scoring, and facility visit scheduling with integration to Laravel-based main system.

These subsystems interact via APIs and shared data models, with the User Management Subsystem providing authentication services to all others.

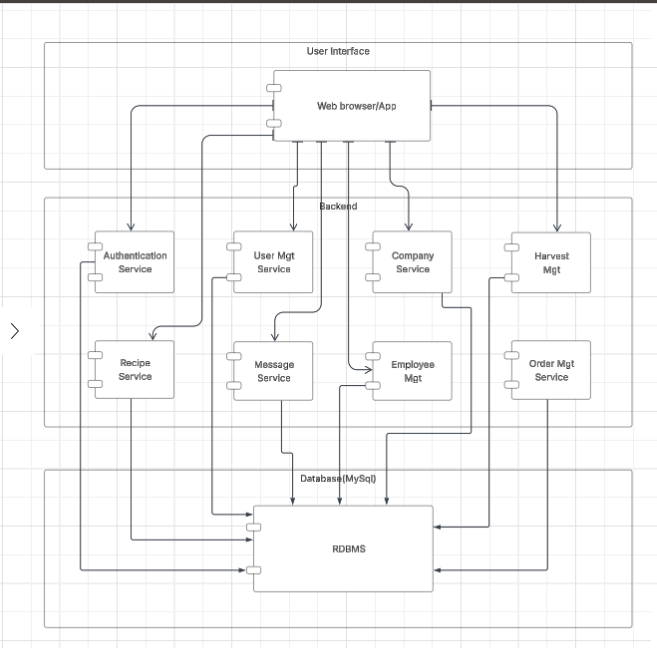
****

Figure 3. 1 Component diagram for the Coffee Supply System

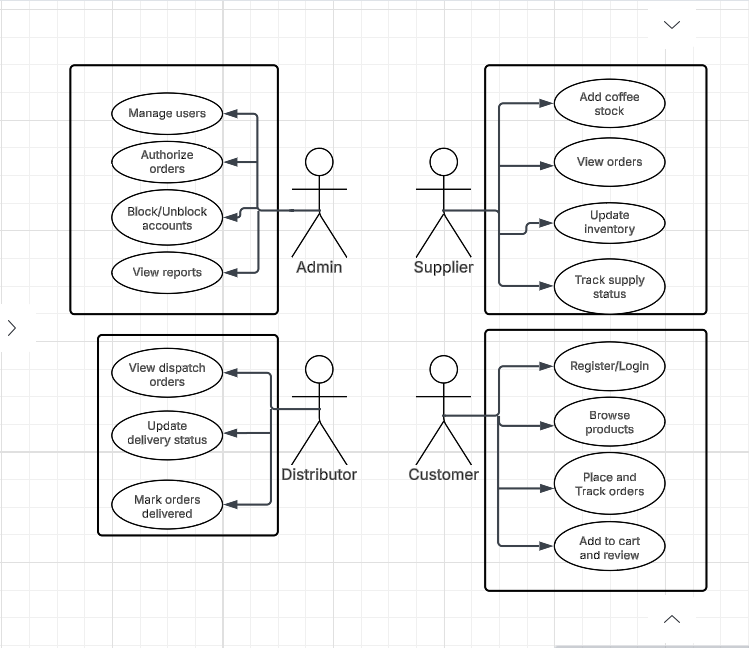


Figure 3. 2 Use-case diagram for the Coffee Supply System

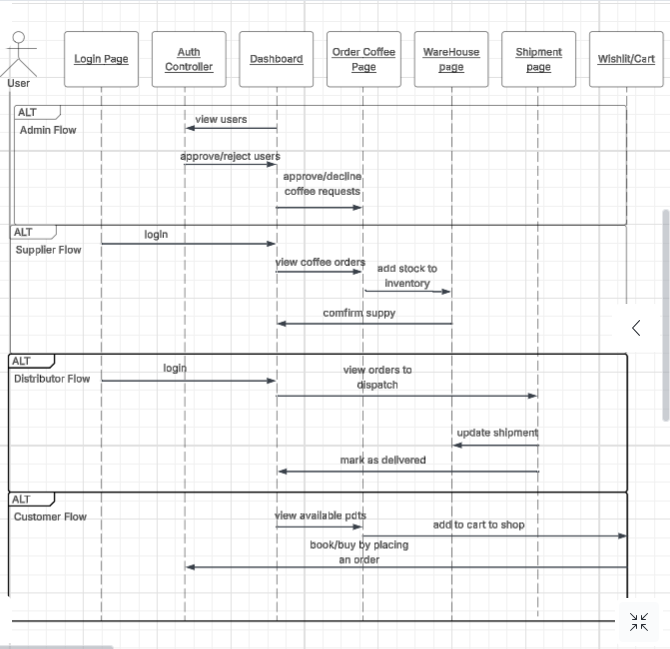


Figure 3. 3 Sequence diagram for the Coffee Supply System

## Design Rationale

The layered architecture was selected to ensure separation of concerns while handling the complexity of coffee supply chain operations with multiple transformation stages, diverse stakeholder types, and varying technical capabilities. Laravel's MVC pattern naturally supports this architecture while providing robust features for authentication, database operations, and API development essential for coffee supply chain integration.

MySQL was chosen as the primary database due to its reliability, performance, and excellent integration with Laravel's Eloquent ORM. The relational model effectively handles the complex relationships between coffee farmers, processors, roasters, distributors, retailers, and the multiple transformation stages inherent in coffee supply chain management. The database design accommodates coffee-specific data requirements including origin tracking, quality parameters, seasonal patterns, and certification management.

The integration of machine learning capabilities was designed to be flexible, allowing for either direct implementation using PHP-based ML libraries or integration with Python-based models through APIs specifically trained on coffee industry data. This approach provides the best of both worlds: Laravel's web development capabilities and Python's rich ML ecosystem with access to coffee consumption pattern analysis and agricultural forecasting models.

The modular subsystem design allows for independent development and testing of coffee supply chain components while maintaining clear interfaces between subsystems. This modularity is particularly important for coffee supply chains that may need to integrate with existing systems used by farmers, processors, or retailers, allowing for gradual adoption and minimal disruption to established workflows.

The user interface architecture adapts to the diverse technical capabilities of coffee supply chain participants, from farmers using basic smartphones in rural areas to sophisticated roasters with advanced analytics needs, ensuring optimal user experience across all stakeholder types while maintaining consistent underlying functionality.

# Data Design

## Data Description

Table 4. 1 Users

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| id | Unique user identifier |
| name | User’s full name |
| email | User’s email address |
| password | Hashed password |
| user\_type | User role |
| company\_id | Reference to Companies table |
| phone | Contact number |
| address | User address |
| status | Account status |
| created\_at | Creation timestamp |
| updated\_at | Last update timestamp |

Table 4. 2 Companies

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| company\_id | Unique company identifier |
| company\_name | Company name |
| email | Company email |
| company\_type | Company role |
| phone | Contact number |
| address | Company address |
| registration\_number | Business registration |
| acceptance\_status | Vendor validation status |
| financial\_risk\_rating | Financial risk score |
| reputational\_risk\_rating | Reputation risk score |
| compliance\_risk\_rating | Compliance risk score |
| created\_at | Creation timestamp |
| updated\_at | Last update timestamp |

Table 4. 3 FarmerHarvest

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| harvest\_id | Unique harvest identifier |
| company\_id | Reference to Companies table |
| coffee\_variety | Coffee type |
| processing\_method | Processing method |
| grade | Quality grade |
| quantity\_kg | Total harvested quantity. |
| available\_quantity\_kg | Available quantity |
| harvest\_date | Harvest date |
| availability\_status | Stock status |
| quality\_notes | Quality-related notes |
| created\_at | Creation timestamp |
| updated\_at | Last update timestamp |

Table 4. 4 ProductRecipes

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| recipe\_id | Unique recipe identifier |
| product\_name | Product category |
| recipe\_name | Specific recipe name |
| coffee\_variety | Coffee type |
| processing\_method | Processing method |
| required\_grade | Required quality grade |
| percentage\_composition | Ingredient percentage. |
| created\_at | Creation timestamp |
| updated\_at | Last update timestamp |

Table 4. 5 FarmersOrders

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| order\_id | Unique order identifier. |
| farmer\_company\_id | Reference to Companies table (farmer) |
| coffee\_variety | Coffee type |
| processing\_method | Processing method |
| grade | Quality grade |
| quantity\_kg | Ordered quantity |
| unit\_price | Price per kg |
| total\_amount | Total order value |
| expected\_delivery\_date | Expected delivery date |
| actual\_delivery\_date | Actual delivery date |
| order\_status | Order status |
| notes | Additional notes |
| created\_at | Creation timestamp |
| updated\_at | Last update timestamp |

Table 4. 6 ProcessorRawMaterialInventory

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| inventory\_id | Unique inventory identifier |
| processor\_company\_id | Reference to Companies table |
| coffee\_variety | Coffee type |
| processing\_method | Processing method |
| grade | Quality grade |
| current\_stock\_kg | Current stock |
| reserved\_stock\_kg | Reserved stock |
| available\_stock\_kg | Current stock minus reserved stock |
| average\_cost\_per\_kg | Cost per kg |
| last\_updated | Last update timestamp |

Table 4. 7 ProcessorFinishedGoodsInventory

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| inventory\_id | Unique inventory identifier |
| processor\_company\_id | Reference to Companies table |
| recipe\_id | Reference to ProductRecipes table |
| Product\_name | Product category |
| product\_variant | Specific product variant |
| current\_stock\_units | Current stock |
| reserved\_stock\_units | Reserved stock |
| available\_stock\_units | Current stock minus reserved stock |
| production\_cost\_per\_unit | Cost per unit |
| selling\_price\_per\_unit | Selling price per unit |
| last\_updated | Last update timestamp |

Table 4. 8 RetailerOrders

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| order\_id | Unique order identifier |
| order\_number | Readable order number |
| processor\_company\_id | Reference to Companies table (processor) |
| total\_amount | Total order value |
| expected\_delivery\_date | Expected delivery date |
| actual\_delivery\_date | Actual delivery date |
| order\_status | Order status |
| shipping\_address | Delivery address |
| notes | Additional notes |
| created\_at | Creation timestamp |
| updated\_at | Last update timestamp |

Table 4. 9 RetailerOrderItems

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| item\_id | Unique item identifier |
| order\_id | Reference to Retailer Orders table |
| recipe\_id | Reference to Product Recipes table |
| product\_name | Product category |
| product\_variant | Specific product variant |
| quantity\_units | Ordered quantity |
| unit\_price | Price per unit |
| line\_total | Total for item |
| created\_at | Creation timestamp |

Table 4. 10 Employees

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| employee\_id | Unique employee identifier |
| processor\_company\_id | Reference to Companies table |
| employee\_name | Employee’s full name |
| employee\_code | Unique employee code |
| skill\_set | Employee skills |
| primary\_station | Primary role |
| current\_station | Current role |
| availability\_status | Availability |
| shift\_schedule | Shift type |
| hourly\_rate | Pay rate |
| hire\_date | Hire date |
| status | Employment status |
| created\_at | Creation timestamp |
| updated\_at | Last update timestamp |

Table 4. 11 Messages

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| message\_id | Unique message identifier |
| sender\_company\_id | Reference to Companies table (sender) |
| receiver\_company\_id | Reference to Companies table (receiver) |
| sender\_user\_id | Reference to Users table (sender) |
| receiver\_user\_id | Reference to Users table (receiver) |
| subject | Message subject |
| message\_body | Message content |
| message\_type | Message type |
| is\_read | Read status |

Table 4. 12 InventoryTransactions

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| transaction\_id | Unique transaction identifier |
| company\_id | Reference to Companies table. |
| transaction\_type | Transaction type |
| inventory\_type | Inventory type |
| inventory\_item\_id | Reference to inventory table |
| quantity\_change | Quantity added or removed |
| unit\_cost | Cost per unit |
| reference\_type | Reference type |
| reference\_id | Reference to related record |
| notes | Additional notes |
| processed\_by | Reference to Users table (processor) |
| transaction\_date | Transaction timestamp |

## ERD

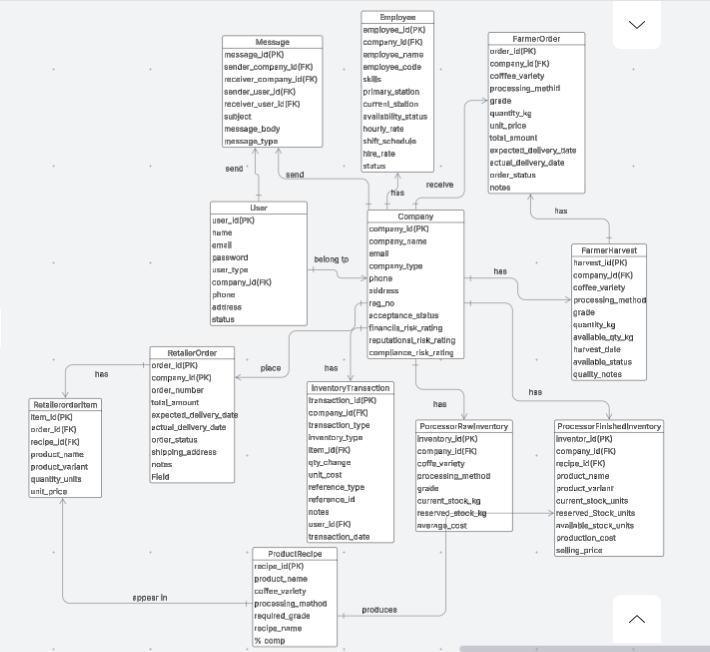
****

Figure 4. 1 ERD for the Coffee Supply Database

## Data Dictionary

Table 4. 13 Users

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Size** | **Constraint** |
| id | INT |  | PK, AUTO\_INCREMENT |
| name | VARCHAR | 25 | NOT NULL |
| email | VARCHAR | 40 | NOT NULL |
| password | VARCHAR | 60 | NOT NULL |
| user\_type | ENUM('farmer', 'processor', 'retailer', 'admin') |  | NOT NULL |
| company\_id |  |  | FK |
| phone | INT |  | NULL |
| address | VARCHAR | 20 | NULL |
| status | ENUM('active', 'inactive', 'pending'), DEFAULT 'pending') |  | NOT NULL |
| created\_at | TIMESTAMP |  | NOT NULL |
| updated\_at | TIMESTAMP |  | NULL |

Table 4. 14 Companies

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Size** | **Constraint** |
| company\_id | INT |  | PK, AUTO\_INCREMENT |
| company\_name | VARCHAR | 50 | NOT NULL |
| email | VARCHAR | 50 | NOT NULL |
| company\_type | ENUM('farmer', 'retailer') |  | NOT NULL |
| phone | INT |  | NOT NULL |
| address | TEXT |  | NOT NULL |
| registration\_number | VARCHAR | 50 | NOT NULL |
| acceptance\_status | ENUM('accepted', 'rejected', 'pending', 'visit\_scheduled'), DEFAULT 'pending') |  | NOT NULL |
| financial\_risk\_rating | DECIMAL | (3,1) | NOT NULL |
| reputational\_risk\_rating | DECIMAL | (3,1) | NOT NULL |
| compliance\_risk\_rating | DECIMAL | (3,1) | NOT NULL |
| created\_at | TIMESTAMP |  | NOT NULL |
| updated\_at | TIMESTAMP |  | NULL |

Table 4. 15 FarmerHarvest

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Size** | **Constraint** |
| harvest\_id | INT |  | PK, AUTO\_INCREMENT |
| company\_id | INT |  | FK |
| coffee\_variety | ENUM('arabica', 'robusta') |  | NOT NULL |
| processing\_method | ENUM('natural', 'washed', 'honey') |  | NOT NULL |
| grade | ENUM('grade\_1', 'grade\_2', 'grade\_3', 'grade\_4', 'grade\_5') |  | NOT NULL |
| quantity\_kg | DECIMAL | (10,2) | NOT NULL |
| available\_quantity\_kg | DECIMAL | (10,2) | NOT NULL |
| harvest\_date | DATE |  | NOT NULL |
| availability\_status | ENUM('available', 'reserved', 'sold\_out', 'expired') |  | NOT NULL |
| quality\_notes | TEXT |  | NULL |
| created\_at | TIMESTAMP |  | NOT NULL |
| updated\_at | TIMESTAMP |  | NULL |

Table 4. 16 Product Recipes

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Size** | **Constraint** |
| recipe\_id | INT |  | PK, AUTO\_INCREMENT |
| product\_name | ENUM('drinking\_coffee', 'roasted\_coffee', 'coffee\_scents', 'coffee\_soap') |  | NOT NULL |
| recipe\_name | VARCHAR |  | NOT NULL |
| coffee\_variety | ENUM('arabica', 'robusta') |  | NOT NULL |
| processing\_method | ENUM('natural', 'washed', 'honey') |  | NOT NULL |
| required\_grade | ENUM('grade\_1', 'grade\_2', 'grade\_3', 'grade\_4', 'grade\_5') |  | NOT NULL |
| percentage\_composition | DECIMAL | (5,2) | NOT NULL |
| created\_at | TIMESTAMP |  | NOT NULL |
| updated\_at | TIMESTAMP |  | NULL |

Table 4. 17 FarmerOrders

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Size** | **Constraint** |
| order\_id | INT |  | PK, AUTO\_INCREMENT |
| farmer\_company\_id | INT |  | FK |
| coffee\_variety | ENUM('arabica', 'robusta') |  | NOT NULL |
| processing\_method | ENUM('natural', 'washed', 'honey') |  | NOT NULL |
| grade | ('grade\_1', 'grade\_2', 'grade\_3', 'grade\_4', 'grade\_5') |  | NOT NULL |
| quantity\_kg | DECIMAL | (10,2) | NOT NULL |
| unit\_price | DECIMAL | (8,2) | NOT NULL |
| total\_amount | DECIMAL | (12,2) | NOT NULL |
| expected\_delivery\_date | DATE |  | NOT NULL |
| actual\_delivery\_date | DATE |  | NULL |
| order\_status | ENUM('pending', 'confirmed', 'processing', 'shipped', 'delivered', 'cancelled'), DEFAULT 'pending') |  | NOT NULL |
| notes | TEXT |  | NULL |
| created\_at | TIMESTAMP |  | NOT NULL |
| updated\_at | TIMESTAMP |  | NULL |

Table 4. 18 ProcessorRawMaterialInventory

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Size** | **Constraint** |
| inventory\_id | INT |  | PK, AUTO\_INCREMENT |
| processor\_company\_id | INT |  | FK |
| coffee\_variety | (ENUM('arabica', 'robusta') |  | NOT NULL |
| processing\_method |  |  | NOT NULL |
| grade | ENUM('grade\_1', 'grade\_2', 'grade\_3', 'grade\_4', 'grade\_5') |  | NOT NULL |
| current\_stock\_kg | DECIMAL | (10,2) | NOT NULL |
| reserved\_stock\_kg | DECIMAL | (10,2) | NOT NULL |
| available\_stock\_kg | DECIMAL | (10,2) | NOT NULL |
| average\_cost\_per\_kg | DECIMAL | (10,2) | NULL |
| last\_updated | TIMESTAMP |  | NULL |

Table 4. 19 ProcessorFinishedGoodsInventory

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Size** | **Constraint** |
| inventory\_id | INT |  | PK, AUTO\_INCREMENT |
| processor\_company\_id | INT |  | FK, NOT NULL |
| recipe\_id | INT |  | FK, NOT NULL |
| Product\_name | ENUM('drinking\_coffee', 'roasted\_coffee', 'coffee\_scents', 'coffee\_soap') |  | NOT NULL |
| product\_variant | VARCHAR | 10 | NOT NULL |
| current\_stock\_units | DECIMAL | (10,2) | NOT NULL |
| reserved\_stock\_units | DECIMAL | (10,2) | NOT NULL |
| available\_stock\_units | DECIMAL | (10,2) | NOT NULL |
| production\_cost\_per\_unit | DECIMAL | (10,2) | NULL |
| selling\_price\_per\_unit | DECIMAL | (10,2) | NULL |
| last\_updated | TIMESTAMP |  | NULL |

Table 4. 20 RetailerOrders

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Size** | **Constraint** |
| order\_id | INT |  | PK, AUTO\_INCREMENT |
| order\_number | VARCHAR | 20 | UNIQUE, NOT NULL |
| processor\_company\_id | INT |  | FK, NOT NULL |
| total\_amount | DECIMAL | (12,2) | NOT NULL |
| expected\_delivery\_date | DATE |  | NOT NULL |
| actual\_delivery\_date | DATE |  | NULL |
| order\_status | ENUM('pending', 'confirmed', 'processing', 'shipped', 'delivered', 'cancelled') |  | NOT NULL |
| shipping\_address | TEXT |  | NOT NULL |
| notes | TEXT |  | NULL |
| created\_at | TIMESTAMP |  | NOT NULL |
| updated\_at | TIMESTAMP |  | NULL |

Table 4. 21 RetailerOrderItems

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Size** | **Constraint** |
| item\_id | INT |  | PK, AUTO\_INCREMENT |
| order\_id | INT |  | FK, NOT NULL |
| recipe\_id | INT |  | FK, NOT NULL |
| product\_name | ENUM('drinking\_coffee', 'roasted\_coffee', 'coffee\_scents', 'coffee\_soap') |  | NOT NULL |
| product\_variant | VARCHAR | 10 |  |
| quantity\_units | DECIMAL | (10,2) |  |
| unit\_price | DECIMAL | (8,2) |  |
| line\_total | DECIMAL | (12,2) |  |
| created\_at | TIMESTAMP |  | NOT NULL |

Table 4. 22 Employees

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Size** | **Constraint** |
| employee\_id | INT |  | PK, AUTO\_INCREMENT |
| processor\_company\_id | INT |  | FK, NOT NULL |
| employee\_name | VARCHAR | 20 | NOT NULL |
| employee\_code | VARCHAR | 30 | NOT NULL, UNIQUE |
| skill\_set | SET('grading', 'roasting', 'packaging', 'logistics', 'quality\_control', 'maintenance') |  | NOT NULL |
| primary\_station | ENUM('grading', 'roasting', 'packaging', 'logistics', 'quality\_control', 'maintenance') |  | NOT NULL |
| current\_station | (ENUM('grading', 'roasting', 'packaging', 'logistics', 'quality\_control', 'maintenance') |  | NULL |
| availability\_status | ENUM('available', 'busy', 'on\_break', 'off\_duty', 'on\_leave'), DEFAULT 'available') |  | NOT NULL |
| shift\_schedule | ENUM('morning', 'afternoon', 'night', 'flexible'), DEFAULT 'morning') |  | NOT NULL |
| hourly\_rate | DECIMAL | (8,2) | NULL |
| hire\_date | DATE |  | NOT NULL |
| status | ENUM('active', 'inactive', 'terminated'), DEFAULT 'active') |  | NOT NULL |
| created\_at | TIMESTAMP |  | NOT NULL |
| updated\_at | TIMESTAMP |  | NULL |

Table 4. 23 Messages

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Size** | **Constraint** |
| message\_id | INT |  | PK, AUTO\_INCREMENT |
| sender\_company\_id | INT |  | FK, NOT NULL |
| receiver\_company\_id | INT |  | FK, NOT NULL |
| sender\_user\_id | INT |  | FK, NOT NULL |
| receiver\_user\_id | INT |  | FK, NULL |
| subject | VARCHAR | 100 |  |
| message\_body | TEXT |  |  |
| message\_type | ENUM('general', 'order\_inquiry', 'quality\_feedback', 'delivery\_update', 'system\_notification') |  |  |
| is\_read | BOOLEAN |  | NOT NULL |

Table 4. 24 InventoryTransactions

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data Type** | **Size** | **Constraint** |
| transaction\_id | INT |  | PK, AUTO\_INCREMENT |
| company\_id | INT |  | FK, NULL |
| transaction\_type | ENUM('stock\_in', 'stock\_out', 'adjustment', 'transfer', 'production', 'waste') |  | NOT NULL |
| inventory\_type | ENUM('raw\_material', 'finished\_goods') |  | NOT NULL |
| inventory\_item\_id | INT |  | FK, NOT NULL |
| quantity\_change | DECIMAL | (10,2) | NOT NULL |
| unit\_cost | DECIMAL | (8,2) | NULL |
| reference\_type | ENUM('farmer\_order', 'retailer\_order', 'production', 'adjustment', 'waste') |  | NULL |
| reference\_id | INT |  | NULL |
| notes | TEXT |  | NULL |
| processed\_by | INT |  | NULL, FK |
| transaction\_date | TIMESTAMP |  | NOT NULL |

# Component Design

Below are pseudocode algorithms for key components, adapted to the Coffee Supply Chain Management System:

**User Authentication Component:**

FUNCTION authenticateCoffeeUser(email, password)

BEGIN

user = findCoffeeUserByEmail(email)

IF user EXISTS AND verifyPassword(password, user.password) THEN

generateAuthToken(user)

logLoginActivity(user.id, user.role\_id)

loadUserDashboardPreferences(user.role\_id)

checkCertificationExpiry(user.id)

RETURN success WITH user\_data AND auth\_token AND dashboard\_config

ELSE

logFailedAttempt(email, getClientIP())

RETURN failure WITH error\_message

END IF

END

**Inventory Management Component:**

FUNCTION updateInventoryLevel(inventory\_id, quantity\_change, transaction\_type)

BEGIN

LOCK inventory\_record WHERE inventory\_id

current\_inventory = getCurrentInventory(inventory\_id)

IF transaction\_type = 'stock\_out' AND current\_inventory.available\_stock\_kg < quantity\_change THEN

UNLOCK inventory\_record

RETURN insufficient\_stock\_error

END IF

UPDATE inventory SET

current\_stock\_kg = current\_stock\_kg + quantity\_change,

reserved\_stock\_kg = reserved\_stock\_kg + IF(transaction\_type = 'reservation', quantity\_change, 0),

last\_updated = CURRENT\_TIMESTAMP

logInventoryTransaction(inventory\_id, quantity\_change, transaction\_type)

checkReorderPoint(inventory\_id)

UNLOCK inventory\_record

RETURN success

END

**Demand Prediction ML Component:**

FUNCTION predictDemand(product\_id, forecast\_period)

BEGIN

historical\_data = getSalesHistory(product\_id, LAST\_24\_MONTHS)

external\_data = getExternalData(ICO, NOAA, FAO)

features = prepareFeatures(historical\_data, external\_data, time\_features, lag\_variables)

arima\_model = loadARIMAModel()

lstm\_model = loadLSTMModel()

rf\_model = loadRandomForestModel()

arima\_pred = arima\_model.predict(features, forecast\_period)

lstm\_pred = lstm\_model.predict(features, forecast\_period)

rf\_pred = rf\_model.predict(features, forecast\_period)

prediction = ensembleAverage(arima\_pred, lstm\_pred, rf\_pred)

confidence\_interval = calculateConfidenceInterval(prediction)

storePrediction(product\_id, prediction, confidence\_interval, forecast\_period)

RETURN prediction\_result

END

**Order Processing Component:**

FUNCTION processNewOrder(order\_data, tier)

BEGIN

validateOrderData(order\_data)

order = createOrderRecord(order\_data, tier)

FOR each item IN order\_data.items DO

IF tier = 'retailer' THEN

reserveInventory(item.recipe\_id, item.quantity\_units, order.order\_id)

ELSE IF tier = 'processor' THEN

reserveFarmerHarvest(item.harvest\_id, item.quantity\_kg, order.order\_id)

END IF

IF reservation\_failed THEN

rollbackOrder(order.order\_id)

RETURN insufficient\_inventory\_error

END IF

END FOR

calculateOrderTotal(order.order\_id)

sendOrderConfirmation(order.customer\_id, order.order\_id)

notifySupplier(order.supplier\_id, order.order\_id)

scheduleDelivery(order.order\_id)

RETURN order\_confirmation

END

**Vendor Validation Component:**

FUNCTION validateVendorApplication(application\_id)

BEGIN

application = getApplication(application\_id)

validation\_score = 0

financial\_score = evaluateFinancialStability(application.financial\_data)

reputation\_score = checkReputation(application.company\_name, application.references)

compliance\_score = verifyCertifications(application.certifications)

total\_score = (financial\_score + reputation\_score + compliance\_score) / 3

IF total\_score >= VALIDATION\_THRESHOLD THEN

scheduleFieldVisit(application\_id)

updateApplicationStatus(application\_id, 'visit\_scheduled')

sendApprovalNotification(application.contact\_email)

ELSE

updateApplicationStatus(application\_id, 'rejected')

sendRejectionNotification(application.contact\_email, validation\_reasons)

END IF

RETURN validation\_result

END

# Human Interface Design

## Overview of User Interface

The system provides a responsive web-based interface accessible via modern browsers, optimized for both desktop and mobile devices. Role-based dashboards cater to farmers, processors, retailers, and admins, offering tailored functionalities. Users interact through intuitive forms, interactive tables, real-time charts, messaging interfaces, and notification panels. The interface includes context-sensitive help, success/error messages, and progress indicators for long-running operations.

## Screen Images



Figure 6. Login Page

|  |  |
| --- | --- |
| Login Screen | Displays company branding, email/password fields, “Remember Me” option, forgot password link, and vendor registration link. |
| Farmer Dashboard | Shows production recommendations, inventory levels, order status, price trends, and quality metrics. |
| Processor Dashboard | Includes production planning, recipe management, raw material sourcing, inventory tracking, and order fulfillment views. |
| Retailer D  ashboard | Features sales analytics, inventory management, customer segmentation, order processing, and POS integration. |
| Analytics Dashboard | Displays interactive charts for sales trends, demand forecasts, and customer segmentation. |
| Chat Interface | Provides real-time messaging with participant lists, message history, and file-sharing capabilities. |

## Screen Objects and Actions

|  |  |  |
| --- | --- | --- |
| Object | Action | Result |
| User profile dropdown | Click | View profile, logout |
| KPI cards | Click | view detailed reports, e.g., sales trends, inventory levels |
| Submit/Cancel buttons | Click | Process or discard form data |
| File upload areas | Click | Upload certifications for vendor validation |
| Menu items | Click | Navigate to modules like inventory, orders, analytics |