

Supply Chain Management System

Coffee Industry Solution

Comprehensive Software Design Document

Project Scope

End-to-end coffee supply chain digitization from farm to retail

Architecture

Hybrid MVC + Microservices with ML integration

Key Features

Inventory management, demand prediction, vendor validation

Target Users

Farmers, suppliers, processors, distributors, retailers

Laravel

Python ML

Java

MySQL

Cloud Storage

© 2025 Coffee Supply Chain Management System | Design Document

Group 9

NAME

REG.Number

Tusiime Emmanuel

22/U/3920/EVE

Aine Levi

22/U/2903/EVE

Otim Ronald

22/U/3780/PS

Ssentongo Henry Atanus

22/U/3870/PS

Bengana Anthony

17/U/3569/EVE

Table of Contents

List of Tables	iii
List of figures	iv
1. INTRODUCTION	1
1.1 Purpose	1
1.2 Scope	1
1.3 Overview	1
1.4 Reference Material	2
1.5 Acronyms	2
2. System Overview	2
3. SYSTEM ARCHITECTURE	3
3.1 Architectural Design	3
3.1.1 System Layers and Components	4
3.2 Decomposition Description	5
3.3 Design Rationale	6
4. DATA DESIGN	7
4.1 Data Description	7
4.2 Data Dictionary	7
5. COMPONENT DESIGN	8
5.1 ER - Diagram	8
5.2 Database Schema	8
5.2 Functional/Procedural Description of System Components (Pseudo code)	11
5.2.1 User Authentication	11
5.2.2 Inventory Management function updateInventory(product_id, location_id, new_quantity):	11
5.2.3 Order Processing	12
5.2.4 Vendor Validation (Java Microservice)	12
5.2.5 Machine Learning – Demand Prediction	12
5.2.6 Customer Segmentation	12
5.2.7 Real-time Chat Messaging	12
5.2.8 Report Generation	13
6. HUMAN INTERFACE DESIGN	13
6.1 Overview of User Interface	13
6.1.1 Key Functional Interactions:	13
6.1.2 Feedback Mechanisms:	13
6.2 Screen Images	14
6.3 Screen Objects and Actions	16
7. REQUIREMENTS MATRIX	16
8. APPENDICES	16
Appendix A: Tools and Technologies Used	16
Appendix B: Dataset Sources	17
Appendix D: User Roles Overview	17
8. REFERENCES	17

List of Tables

Table 1 Shows User	8
Table 2 Product	9
Table 3 Categories	9
Table 4 Suppliers	9
Table 5 Inventories	9
Table 6 Location.....	9
Table 7 Orders.....	10
Table 8 Order_Items	10
Table 9 Chat_Messages	10
Table 10 Chat_Messages	10
Table 11 Reports	10
Table 12 Vendors	11
Table 13 Customers	11
Table 14 Customer_Segments.....	11

List of figures

Figure 1 System Overview.....	3
Figure 2 Login / Register Screen.....	14
Figure 3 Supplier and Admin Dashboards	15
Figure 4 Factory manager and Customer interfaces	15

1. INTRODUCTION

1.1 Purpose

The purpose of the Supply Chain Management System (SCMS) is to provide a centralized digital platform to streamline and monitor the end-to-end processes involved in the coffee supply chain. This system is designed to ensure transparency, efficiency, accountability, and collaboration among all key stakeholders—ranging from farmers and suppliers to processors, distributors, and retailers. By digitizing and optimizing operations, the SCMS aims to enhance product quality, ensure timely deliveries, minimize losses, and improve customer satisfaction through data-driven decision-making and predictive analytics.

1.2 Scope

The SCMS covers the entire coffee supply chain lifecycle, including:

- a Sourcing of raw coffee beans
- b Processing and packaging
- c Warehousing and inventory management
- d Order processing and fulfillment
- e Distribution and delivery to retail stores

1.2.1 Key functional modules included in the system:

- a Inventory management
- b Order processing
- c Demand prediction through analytics
- d Customer segmentation and personalization
- e Real-time communication between stakeholders
- f Workforce allocation optimization
- g Automated vendor validation and onboarding

1.3 Overview

Coffee is a globally traded commodity with a complex supply chain. The SCMS is designed to address several challenges in managing this network, such as inefficient inventory tracking, poor communication, inaccurate demand forecasting, and non-automated reporting. By integrating advanced analytics and communication features, the system improves visibility, optimizes resource allocation, and enables better coordination among all stakeholders. Ultimately, the SCMS ensures that high-quality coffee products reach consumers reliably, efficiently, and profitably.

1.4 Reference Material

The following sources may be consulted for further understanding and development of the SCMS:

- a Coffee Supply Chain Industry Reports
- b Inventory and Supply Chain Best Practices
- c Analytics and Machine Learning in Retail
- d ISO Standards for Supply Chain and Vendor Management
- e Software Development Guidelines for ERP Systems
- f Relevant local and international regulatory guidelines

1.5 Acronyms

<i>ACRONYM</i>	<i>DEFINITION</i>
SCMS	Supply Chain Management System
ERP	Enterprise Resource Planning
SKU	Stock Keeping Unit
FORECASTING	The use of historical data and analytics to predict future demand
VENDOR ONBOARDING	The process of validating and registering new suppliers or partners
INVENTORY OPTIMIZATION	The process of maintaining ideal inventory levels to meet demand without overstocking
STAKEHOLDERS	All parties involved in the supply chain, including farmers, processors, wholesalers, retailers, and logistics personnel

2. System Overview

The Supply Chain Management System (SCMS) is a modular, scalable, and service-oriented software solution designed to support the full lifecycle of the coffee supply chain, from bean sourcing to retail distribution. Built with modern technologies, the SCMS ensures efficiency, high availability, and adaptability through the use of microservices, web frameworks, and predictive analytics.

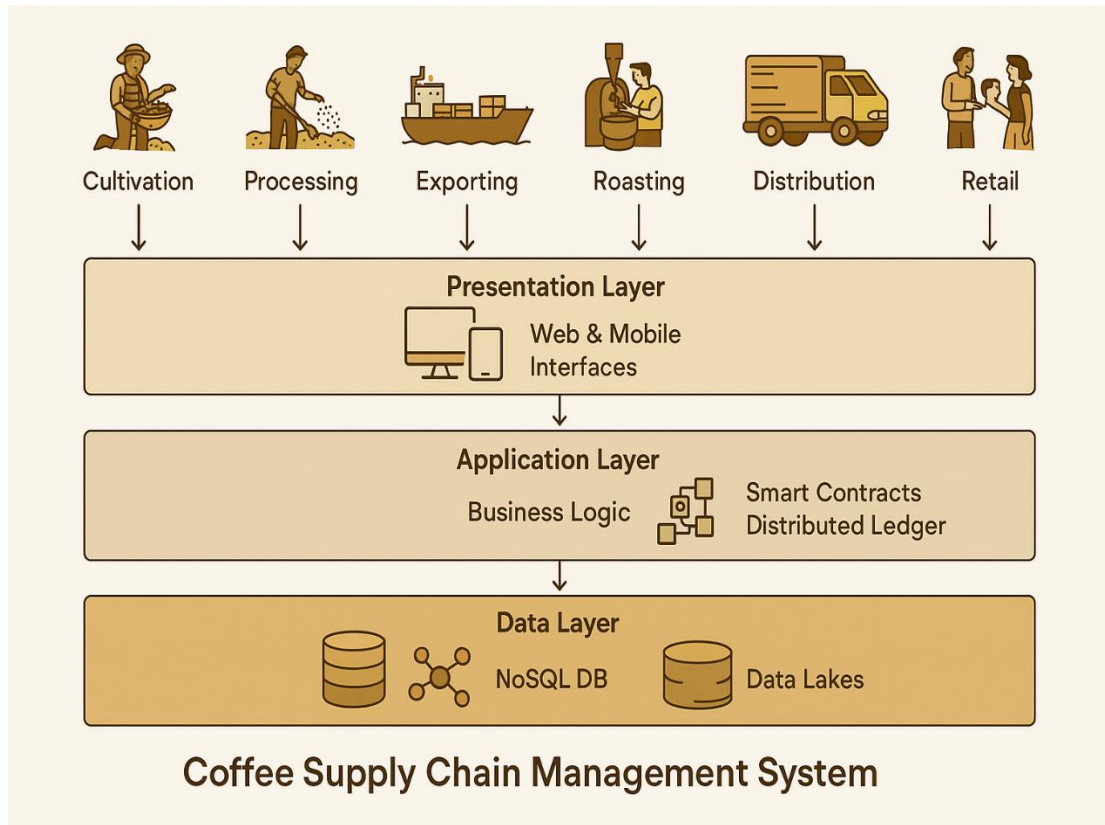


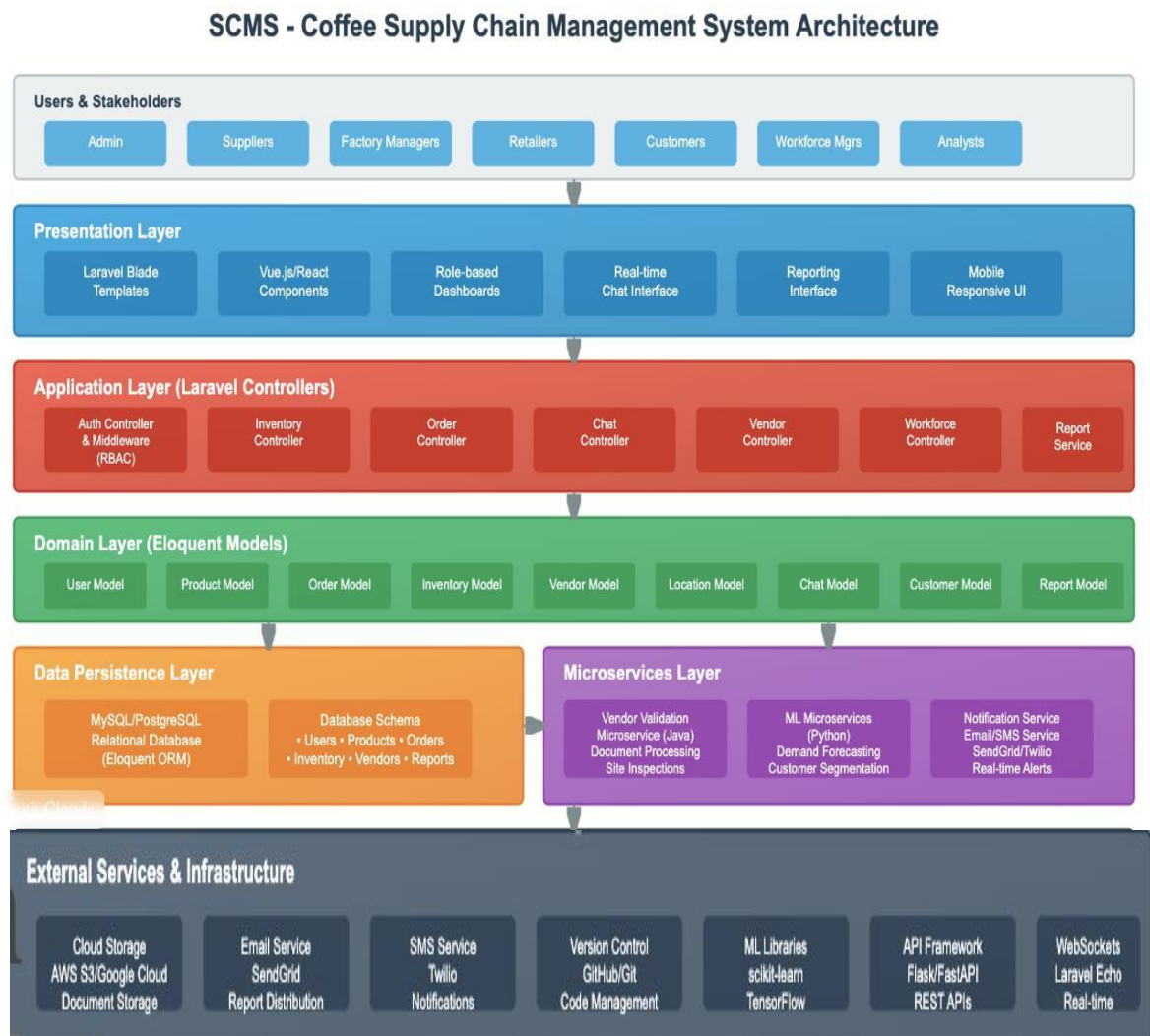
Figure 1 System Overview

3. SYSTEM ARCHITECTURE

3.1 Architectural Design

The architecture of SCMS is layered and hybrid, incorporating both monolithic and microservices design paradigms to balance rapid development and independent scalability.

Architectural Design Image



3.1.1 System Layers and Components

- Presentation Layer:** Offers role-specific web interfaces using Laravel Blade templates. Supports dynamic UI with Vue.js or React if needed.
- Application Layer:** Laravel Controllers orchestrate input validation, business logic processing, and inter-service communication.
- Domain Layer:** Eloquent Models define and enforce business logic for entities like Users, Orders, Products, and Vendors.
- Persistence Layer:** Relational Database (MySQL/PostgreSQL) stores structured data. Access is facilitated by Eloquent ORM.

e **Microservices Layer:**

- i **Vendor Validation:** Java-based microservice managing vendor eligibility and documentation.
- ii **Machine Learning:** Python-based microservices for forecasting and customer segmentation, accessible via REST APIs.

3.2 Decomposition Description

SCMS is decomposed into core modules and services to ensure separation of concerns, maintainability, and service independence:

3.2.1 Laravel Core Application (MVC)

- a **Models:** Represent business entities (e.g., Users, Inventory, Orders).
- b **Controllers:** Handle routing, logic delegation, and API exposure.
- c **Views:** Render dynamic content using Blade, with optional Vue.js/React components.

3.2.2 Central Database

- a Relational DB (MySQL/PostgreSQL) stores structured data including vendor records, transactions, user logs, etc.
- b Managed via Laravel migrations and Eloquent ORM.

3.2.2 Vendor Validation Microservice (Java)

- a Receives applications and document uploads from Laravel frontend.
- b Parses and evaluates documents using rule-based validation logic.
- c Coordinates site inspections and returns status updates.

3.2.3 Machine Learning Microservices (Python)

- a Implements forecasting using historical sales and behavioral data.
- b Provides segmentation insights to optimize product offerings.
- c Built using scikit-learn, TensorFlow, Flask or FastAPI.

3.2.4 Reporting and Notification Services

- a Scheduled or triggered generation of sales, inventory, and performance reports.
- b Notifications sent through Twilio (SMS) and SendGrid (Email).

3.2.5 External File Storage

- a Stores vendor PDFs and compliance files securely via Google Cloud Storage.
- b File references linked in the central database for quick retrieval.

3.2.6 User Management and Access Control

- a Implements Laravel's built-in authentication and role-based access (RBAC).
- b User roles include: Admin, Supplier, Factory Manager, Retailer, and Analyst.

3.3 Design Rationale

The design choices for SCMS were made to achieve modularity, performance, scalability, and long-term maintainability:

- a **Laravel MVC Framework:** Offers rapid development and clear separation of concerns with a well-supported PHP ecosystem.
- b **Microservices Architecture:** Vendor validation and ML processing are offloaded to independent services for better fault isolation, parallel scalability, and technology flexibility.
- c **RESTful Communication:** Enables modular inter-service interactions and smooth integration with third-party systems.
- d **Relational Database:** MySQL/PostgreSQL provides ACID compliance and is optimized for complex transactional workloads.
- e **Cloud Storage for Documents:** Separates large files from relational data to reduce DB load and improve scalability.
- f **Eloquent ORM:** Streamlines object-relational mapping, enhancing consistency and developer productivity.
- g **Security Focus:** Leverages Laravel's authentication and RBAC to enforce strict access control.
- h **Scalability & Maintainability:** Loosely coupled architecture with clearly defined responsibilities makes SCMS adaptable to evolving business needs.

4. DATA DESIGN

4.1 Data Description

The system transforms real-world coffee supply chain processes into structured digital data through an entity-relationship model. Each actor or component in the supply chain (users, products, suppliers, orders, etc.) is mapped into entities and stored in a **MySQL/PostgreSQL relational database** using Laravel's **Eloquent ORM**. Here's how data is organized and processed:

- a **Users:** Role-based records that include suppliers, retailers, admins, and other stakeholders. Stored in the users table with authentication credentials and access roles.
- b **Products:** Each coffee product is uniquely identified by a SKU. Stored in the products table with links to suppliers and categories.
- c **Inventory:** Records current stock at various locations. Includes location ID and product quantity.
- d **Orders:** Tracks sales and movement of products through orders and order_items tables.
- e **Locations:** Warehouses, factories, and retail stores. Used to assign workforce, stock, and order deliveries.
- f **ChatMessages:** Enables communication between users. Supports real-time/asynchronous coordination.
- g **VendorApplications:** Submitted by potential suppliers, including document links and approval status.
- h **Machine Learning Outputs:** Demand forecasts and customer segmentation stored for visualization and decision support.
- i **External Storage:** Vendor documents stored in cloud storage (AWS S3/Google Cloud), with paths linked in the database.

4.2 Data Dictionary

Entity	Type	Description
User	Object	Represents all system users with roles (Admin, Supplier, etc.).
→ id	BIGINT	Unique identifier
→ name	VARCHAR	Full name
→ email	VARCHAR	Unique login
→ password	VARCHAR	Hashed password
→ role	VARCHAR	Role in system
Product	Object	Represents coffee products with SKU, price, and category.
Category	Object	Grouping for products (e.g., Arabica, Robusta).
Supplier	Object	Vendor details and approval status.
Inventory	Object	Tracks product quantities at specific locations.
Location	Object	Supply chain physical site.
Order	Object	Customer orders, including order items.
OrderItem	Object	Product quantity, unit price per order.
ChatMessage	Object	Messaging log between users.
Workforce	Object	Assigned personnel per location and shifts.
Report	Object	Summarized insights (inventory, sales, ML output).
VendorApplication	Object	Application for supplier approval.
Customer	Object	Buyers (individuals, cafes, retailers).
CustomerSegment	Object	Classification based on purchasing patterns.

5. COMPONENT DESIGN

5.1 ER - Diagram



5.2 Database Schema

Below is the detailed database schema for the Supply Chain Management System (SCMS) for coffee. The schema is designed for implementation with Laravel migrations and reflects the entities and relationships defined in the ERD.

Table 1 Shows User

Field	Type	attributes
Id	BIGINT	PRIMARY, AUT INCREMENT
Name	VARCHAR(255)	NOT NUL
Email	VARCHAR(255)	UNIQUE, NOT NULL
password	VARCHAR(255)	NOT NULL
Role	VARCHAR(50)	NOT NULL
contact_info	TEXT	NULLABLE
created at	TIMESTAMP	
updated at	TIMESTAMP	

Table 2 Product

Field	Type	attributes
Id	BIGINT	PRIMARY, AUT INCREMENT
Name	VARCHAR(255)	NOT NUL
description	TEXT	NULLABLE
Sku	VARCHAR(100)	UNIQUE, NOT NULL
Price	DECIMAL (10,2)	NOT NULL
category_id	BIGNIT	FOREIGN, NOT NULL
supplier_id	BIGNIT	FOREIGN, NOT NULL
created at	TIMESTAMP	
updated at	TIMESTAMP	

Table 3 Categories

Field	Type	Attributes
Id	BIGINT	PRIMARY, AUT INCREMENT
Name	VARCHAR(255)	NOT NUL
created at	TIMESTAMP	
updated at	TIMESTAMP	

Table 4 Suppliers

Field	Type	Attributes
Id	BIGINT	PRIMARY, AUT INCREMENT
Name	VARCHAR(255)	NOT NUL
contact_info	TEXT	NULLABLE
Status	VARCHAR(30)	NOT NULL (pending/approved/rejected)
created at	TIMESTAMP	
updated at	TIMESTAMP	

Table 5 Inventories

Field	Type	Attributes
Id	BIGINT	PRIMARY, AUTO_INCREMENT
producr_id	BIGINT	FOREIGN, NOT NULL
Quantity	INTEGER	NOT NULL
location_id	BIGINT	FOREIN, NOT NULL
created_at	TIMESTAMP	
updated_at	TIMESTAMP	

Table 6 Location

Field	Type	Attributes
Id	BIGINT	PRIMARY, AUTO_INCREMENT
Name	VARCHAR(255)	NOT NULL
Address	TEXT	NULLABLE
Type	VARCHAR	NOT NULL (factory/warehouse/retailstore)
created_at	TIMESTAMP	
updated_at	TIMESTAMP	

Table 7 Orders

Field	Type	Attributes
Id	BIGINT	PRIMARY, AUT INCREMENT
customer_id	BIGINT	FOREIGN, NOT NUL
Status	VARCHAR(30)	NOT NULL (pending/processing/shipped/delivered)
order_date	DATETIME	NOT NULL
total_amount	DECIMAL (10,2)	NOT NULL
created at	TIMESTAMP	
updated at	TIMESTAMP	

Table 8 Order_Items

Field	Type	Attributes
Id	BIGINT	PRIMARY, AUT INCREMENT
order_id	BIGINT	FOREIGN, NOT NUL
product_id	BIGINT	FOREIGN, NOT NULL
Quantity	INTEGER	NOT NULL
Price	DECIMAL(10,2)	NOT NULL
created at	TIMESTAMP	
updated at	TIMESTAMP	

Table 9 Chat_Messages

Field	Type	Attributes
Id	BIGINT	PRIMARY, AUTO_INCREMENT
sender_id	BIGINT	FOREIGN (users), NOT NULL
reciever_id	BIGINT	FOREIGN (users), NOT NULL
Message	TEXT	NOT NULL
Timestamp	DATETIME	NOT NULL
created at	TIMESTAMP	
updated at	TIMESTAMP	

Table 10 Chat_Messages

Field	Type	Attributes
Id	BIGINT	PRIMARY, AUT INCREMENT
Name	VARCHAR(255)	NOT NUL
Role	VARCHAR (100)	NOT NULL
supply_center_id	BIGINT	FOREIGN(locations), NOT NULL
shift_times	VARCHAR(100)	NULLABLE
created at	TIMESTAMP	
updated at	TIMESTAMP	

Table 11 Reports

Field	Type	Attributes
Id	BIGINT	PRIMARY, AUT INCREMENT
stakeholder_id	BIGINT	FOREIGN(locations), NOT NULL
report_type	VARCHAR(100)	NOT NULL
generated_date	DATETIME	NOT NULL
Context	TEXT	NOT NULL
created at	TIMESTAMP	
updated at	TIMESTAMP	

Table 12 Vendors

Field	Type	Attributes
Id	BIGINT	PRIMARY, AUT INCREMENT
supply_id	BIGINT	FOREIGN(locations), NOT NULL
pdf_path	VARCHAR(255)	NOT NULL
Status	VARCHAR(30)	NOT NULL (pending/approved/rejected)
submission_date	DATETIME	
created at	TIMESTAMP	
updated at	TIMESTAMP	

Table 13 Customers

Field	Type	attributes
Id	BIGINT	PRIMARY, AUT INCREMENT
Name	VARCHAR(255)	NOT NULL
contact_info	TEXT	NULLABLE
segment_id	BIGINT	FOREIGN, NULLABEL
created at	TIMESTAMP	
updated at	TIMESTAMP	

Table 14 Customer_Segments

Field	Type	Attributes
Id	BIGINT	PRIMARY, AUT INCREMENT
Name	VARCHAR(255)	NOT NUL
Description	TEXT	NULLABLE
created at	TIMESTAMP	
updated at	TIMESTAMP	

5.2 Functional/Procedural Description of System Components (Pseudo code)

5.2.1 User Authentication

```
function login(email, password):
    user = DB.findUserByEmail(email)
    if hash(password) == user.password:
        return session.start(user.id)
    else:
        return error("Invalid credentials")
```

5.2.2 Inventory Management

```
function updateInventory(product_id, location_id, new_quantity):

    inventory = DB.findInventory(product_id, location_id)
    if inventory exists:
        inventory.quantity = new_quantity
        DB.save(inventory)
    else:
        DB.createInventory(product_id, location_id, new_quantity)
```

5.2.3 Order Processing

```
function placeOrder(customer_id, items[]):
  order = DB.createOrder(customer_id, current_date)
  for item in items:
    DB.createOrderItem(order.id, item.product_id, item.quantity, item.price)
  total = calculateOrderTotal(order.id)
  order.total_amount = total
  DB.save(order)
```

5.2.4 Vendor Validation (Java Microservice)

```
function validateVendorApplication(application_id, pdf_path):
  document = parsePDF(pdf_path)
  if checkCompliance(document) and checkFinancials(document):
    scheduleInspection(application_id)
    return updateStatus(application_id, "approved")
  else:
    return updateStatus(application_id, "rejected")
```

5.2.5 Machine Learning – Demand Prediction

```
function forecastDemand(product_id, location_id):
  data = getHistoricalSales(product_id, location_id)
  prediction = LSTM_Model.predict(data)
  DB.storeForecast(product_id, location_id, prediction)
  return prediction
```

5.2.6 Customer Segmentation

```
function segmentCustomers():
  customers = DB.getCustomerFeatures()
  segments = KMeans.cluster(customers)
  for customer in customers:
    customer.segment_id = segments[customer.id]
    DB.save(customer)
```

5.2.7 Real-time Chat Messaging

```
function sendMessage(sender_id, receiver_id, message):
  chat = new ChatMessage(sender_id, receiver_id, message, now())
  DB.save(chat)
```


5.2.8 Report Generation

```
function generateReport(report_type, target_user):  
    data = collectData(report_type)  
    formatted = formatAsPDF(data)  
    emailReport(target_user.email, formatted)  
    DB.saveReport(target_user.id, report_type, formatted)
```

6. HUMAN INTERFACE DESIGN

6.1 Overview of User Interface

From the user's perspective, the SCMS interface is designed for **clarity, responsiveness, and role-specific functionality**. It adapts dynamically based on the authenticated user's role—Admin, Supplier, Factory Manager, Retailer, Customer, or Workforce Manager. Here's how users interact with the system:

6.1.1 Key Functional Interactions:

- a) **Login Page:** Users input email and password to securely access the system. Unauthorized access is denied with appropriate feedback.
- b) **Dashboard:** After login, each user sees a dashboard tailored to their role:
 - i. **Admin:** See system-wide KPIs, manage users, review reports, and monitor ML insights.
 - ii. **Supplier:** Submit and track vendor applications, communicate via chat, manage supplied products.
 - iii. **Factory Manager:** Monitor inventory, assign workforce, communicate with suppliers and delivery agents.
 - iv. **Retailer:** Place orders, view product availability, track order status.
 - v. **Customer:** Browse products, place orders, and view delivery timelines.
 - vi. **Workforce Manager:** Allocate shifts, track staff activity by location.

6.1.2 Feedback Mechanisms:

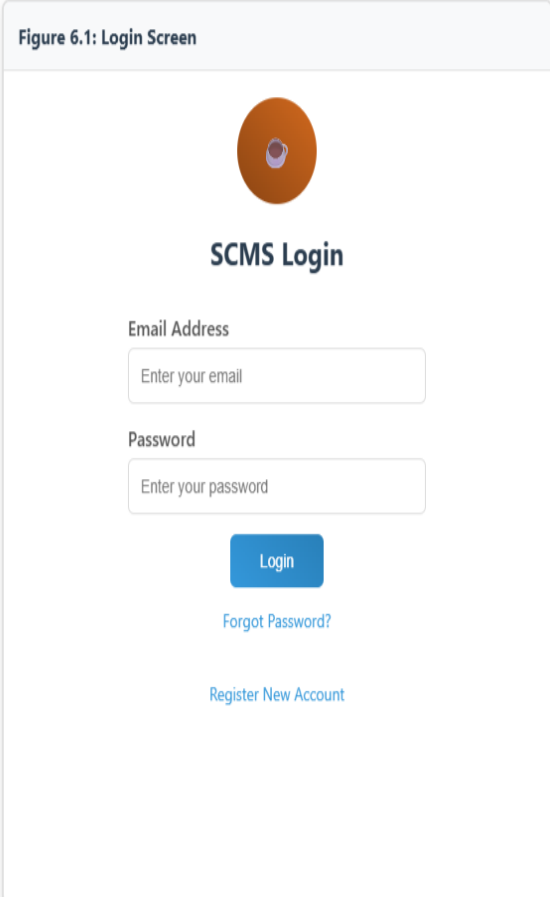
- a) **Real-time Notifications:** Alerts for order updates, stock levels, vendor application status, and new messages.
- b) **Status Tags:** Clear visual tags like *Pending*, *Shipped*, *Delivered*, *Approved* to show current states.
- c) **Form Validation:** Instant feedback on errors (e.g., "Email already exists", "Quantity cannot be negative").
- d) **Loading Indicators:** Shown during data fetching or report generation.

6.2 Screen Images

Supply Chain Management System - Interface Mockups

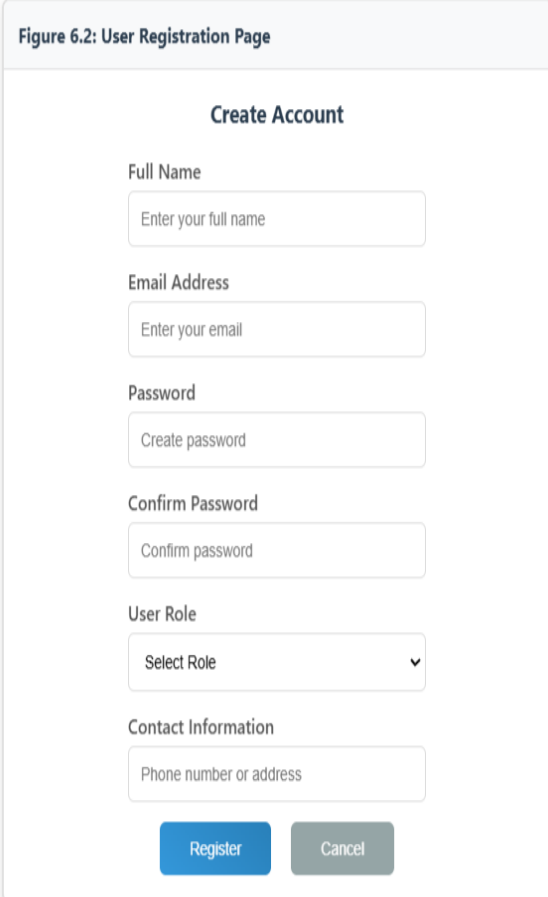
Group 9 Design Document Visualizations

Figure 6.1: Login Screen



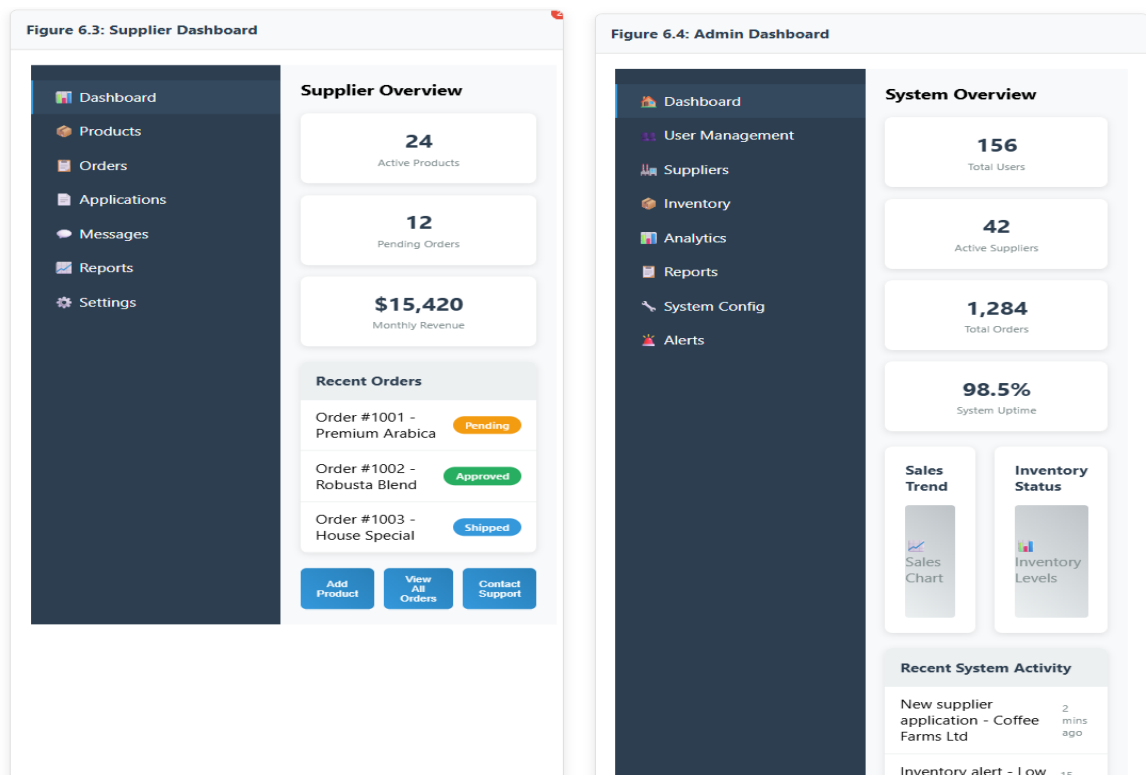
The login screen features a central orange circular logo with a white stylized 'S' inside. Below the logo, the text 'SCMS Login' is displayed in a bold, dark blue font. The form includes two input fields: 'Email Address' with the placeholder 'Enter your email' and 'Password' with the placeholder 'Enter your password'. A blue 'Login' button is positioned below the password field. At the bottom, there are two links: 'Forgot Password?' and 'Register New Account'.

Figure 6.2: User Registration Page



The registration page is titled 'Create Account' in a bold, dark blue font. It contains several input fields: 'Full Name' (placeholder: 'Enter your full name'), 'Email Address' (placeholder: 'Enter your email'), 'Password' (placeholder: 'Create password'), and 'Confirm Password' (placeholder: 'Confirm password'). Below these is a 'User Role' dropdown menu with the text 'Select Role' and a downward arrow. A 'Contact Information' field with the placeholder 'Phone number or address' is located at the bottom. At the very bottom, there are two buttons: a blue 'Register' button and a grey 'Cancel' button.

Figure 2 Login / Register Screen



6.3 Screen Objects and Actions

Each screen contains key objects and user actions as follows:

- Login Screen:** Input fields for username and password; buttons to log in, register, or recover password. Actions: user enters credentials and submits to access the system.
- Register Screen:** Fields for user details and a register button. Actions: user provides information to create an account.
- Admin Dashboard:** Navigation menu and control buttons. Actions: admin manages users, views reports, and configures settings.
- Customer Dashboard:** Profile info and order history. Actions: customer views profile, checks orders, and requests support.
- Factory Dashboard:** Production and inventory displays. Actions: monitor production and update inventory.
- Supplier/Retailer Dashboard:** Order and delivery panels. Actions: manage orders and track shipments.
- Workforce Dashboard:** Task lists and attendance. Actions: update tasks and record attendance.

7. REQUIREMENTS MATRIX

Requirement Description	System Components	Associated Data Structures
User authentication and role-based access	Laravel Auth Module, Role Middleware	users table (id, email, password, role), Sessions
Inventory management (CRUD)	Inventory Controller, Inventory Views	inventories, products, locations
Order processing (CRUD, status tracking)	Order Controller, Order Dashboard	orders, order_items, customers
Real-time/asynchronous chat	Chat Controller, WebSockets (Laravel Echo), Messaging UI	chat_messages, users
Analytics dashboards and ML recommendations	ML Microservices (Python), Dashboard Components	orders, sales data, customer_segments, ML API responses
Workforce distribution management	Workforce Controller, Shift Scheduler UI	workforces, locations
Automated reporting (scheduled emails)	Report Service, Email Scheduler (Laravel Scheduler)	reports, users, orders, inventories
Vendor validation via Java microservice	Java Microservice, Vendor Application Controller	vendor_applications, Cloud PDFs (via S3), suppliers

8. APPENDICES

Appendix A: Tools and Technologies Used

Tool/Technology	Purpose
Laravel (PHP)	Backend development using MVC pattern
MySQL / PostgreSQL	Relational database management
Eloquent ORM	Laravel's built-in ORM for database access
Python (Flask / FastAPI)	Machine Learning microservices
Java	Vendor validation microservice
AWS S3 / Google Cloud Storage	Secure external document storage
SendGrid / Twilio	Email and SMS notifications
Figma / Draw.io / dbdiagram.io	UI mockups, ERD, and architecture diagrams
VS Code / IntelliJ / PyCharm	Development environments
GitHub / Git	Version control and team collaboration

Appendix B: Dataset Sources

Dataset	Source	Purpose
Coffee Sales Data	Kaggle	Used for demand forecasting ML model
E-commerce Behavior Data	Kaggle	Used for customer segmentation (K-means clustering)
Inventory Demand Forecasting Dataset	Kaggle	Supports inventory optimization and predictive restocking

Appendix D: User Roles Overview

Role	Permissions
Admin	Full system access, user management, reports, analytics
Supplier	Product management, vendor application, communication
Factory Manager	Inventory control, workforce management
Retailer/Wholesaler	Product browsing, order placement, status tracking
Customer	Product browsing, personal order placement
Workforce Manager	Shift scheduling, resource allocation

8. REFERENCES

Chopra, S., & Meindl, P. (2016). Supply Chain Management: Strategy, Planning, and Operation (6th Edition). Pearson.

Christopher, M. (2016). Logistics & Supply Chain Management (5th Edition). Pearson.

Han, J., Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques (3rd Edition). Morgan Kaufmann.