# Coffee Supply Chain Management System

## Detailed Functional Specification Document

### 1. Introduction & Purpose

The Coffee Supply Chain Management System is a comprehensive web application designed to streamline inventory and operational workflows for coffee roasteries and retail shops. The system addresses several key challenges in the coffee industry:

#### 1.1 Business Context

* Managing complex coffee supply chains from green beans to retail distribution
* Tracking inventory across multiple locations
* Coordinating roasting operations and quality control
* Ensuring accurate dispatch and delivery confirmation
* Maintaining precise billing and reporting systems

#### 1.2 Problem Statement

Current coffee supply chain management faces several challenges: - Lack of real-time inventory visibility - Difficulty in tracking coffee quality and grades - Inefficient dispatch confirmation processes - Complex billing calculations based on coffee grades - Limited reporting capabilities

#### 1.3 Solution Overview

Our system provides: - Centralized inventory management - Role-based access control - Real-time tracking and notifications - Automated billing calculations - Comprehensive reporting and analytics

### 2. System Overview

#### 2.1 Core Functionality

* **Inventory Management**
  + Green coffee tracking with minimum thresholds
  + Retail product inventory management
  + Automated restock notifications
  + Quality grade tracking
* **Order Processing**
  + Multi-step order workflow
  + Status tracking (pending, roasted, dispatched, delivered)
  + Quantity validation
  + Automated inventory updates
* **Dispatch Confirmation**
  + Two-way confirmation process
  + Quantity discrepancy handling
  + Digital signature capture
  + Automated inventory adjustments
* **Roasting Operations**
  + Batch planning and scheduling
  + Production tracking
  + Loss calculation
  + Quality control integration
* **Shop Management**
  + Multi-location support
  + Stock level monitoring
  + Staff management
  + Performance analytics
* **Billing System**
  + Grade-based pricing
  + Split revenue calculations
  + Automated billing cycles
  + Payment tracking

#### 2.2 Technical Stack

* **Frontend**
  + React with TypeScript
  + Zustand state management
  + shadcn/ui components
  + TanStack Query for data fetching
  + Wouter for routing
  + Zod validation
* **Backend**
  + Node.js runtime
  + Express.js server
  + PostgreSQL database
  + Drizzle ORM
  + WebSocket for real-time updates

### 3. User Roles & Access Control

#### 3.1 Available Roles

1. **Roastery Owner**
   * Full system access
   * Global configuration management
   * Financial report access
   * User management
   * Pricing control
2. **Retail Owner**
   * Access to owned shops
   * Shop performance monitoring
   * Staff management
   * Inventory oversight
   * Order management
3. **Roaster**
   * Roasting operation access
   * Green coffee inventory management
   * Quality control
   * Batch planning
   * Production reporting
4. **Shop Manager**
   * Single shop management
   * Staff supervision
   * Inventory control
   * Order creation
   * Basic reporting
5. **Barista**
   * Basic inventory access
   * Order creation
   * Stock level checking
   * Dispatch confirmation
   * Basic reporting

#### 3.2 Permission Matrix

| Feature | Roastery Owner | Retail Owner | Roaster | Shop Manager | Barista |
| --- | --- | --- | --- | --- | --- |
| User Management | Full | Shop Only | No | No | No |
| Inventory Management | Full | Shop Only | Full | Shop Only | View |
| Order Management | Full | Shop Only | View | Shop Only | Create |
| Financial Reports | Full | Shop Only | No | Basic | No |
| Roasting Operations | Full | No | Full | No | No |
| Dispatch Confirmation | Full | Full | Create | Confirm | Confirm |

### 4. Core Features

#### 4.1 Inventory Management

##### Green Coffee Tracking

* Stock level monitoring
* Minimum threshold alerts
* Producer information
* Country of origin
* Grade tracking (Specialty, Premium, Rarity)
* Quality metrics

##### Retail Inventory

* Shop-specific stock levels
* Package size tracking
  + Small bags (200g)
  + Large bags (1kg)
* Automated restock notifications
* Historical stock levels
* Transfer tracking

#### 4.2 Order Management

##### Order Creation

* Multi-step validation
* Shop-specific pricing
* Quantity checks
* Delivery scheduling

##### Status Tracking

* Pending
* Roasted
* Dispatched
* Delivered
* Status change logging
* Notification system

##### Dispatch Confirmation

* Two-way verification
* Quantity reconciliation
* Digital signatures
* Discrepancy resolution
* Automated inventory updates

#### 4.3 Roasting Operations

##### Batch Planning

* Production scheduling
* Resource allocation
* Capacity planning
* Priority management

##### Production Tracking

* Real-time monitoring
* Loss calculation
* Quality metrics
* Yield analysis
* Batch status updates

##### Quality Control

* Grade verification
* Cupping scores
* Defect tracking
* Sample management

#### 4.4 Reporting & Analytics

##### Monthly Reports

* Sales analysis
* Inventory turnover
* Production efficiency
* Quality metrics
* Financial performance

##### Shop Performance

* Sales trends
* Stock efficiency
* Order patterns
* Staff productivity
* Customer satisfaction

##### Billing Management

* Revenue calculations
* Split payments
* Grade-based pricing
* Payment tracking
* Invoice generation

### 5. Data Model

#### 5.1 Core Entities

##### Users Table

{  
 id: serial("id").primaryKey(),  
 username: text("username").notNull().unique(),  
 password: text("password").notNull(),  
 role: text("role", { enum: userRoles }).notNull(),  
 isActive: boolean("is\_active").notNull().default(true),  
 isPendingApproval: boolean("is\_pending\_approval").notNull().default(true),  
 createdAt: timestamp("created\_at").defaultNow()  
}

##### Shops Table

{  
 id: serial("id").primaryKey(),  
 name: text("name").notNull(),  
 location: text("location").notNull(),  
 isActive: boolean("is\_active").notNull().default(true),  
 desiredSmallBags: integer("desired\_small\_bags").notNull().default(20),  
 desiredLargeBags: integer("desired\_large\_bags").notNull().default(10),  
 createdAt: timestamp("created\_at").defaultNow()  
}

[Additional tables omitted for brevity]

#### 5.2 Relationships

* User-Shop Assignments (Many-to-Many)
* Shop-Inventory Relations (One-to-Many)
* Order-Shop Relations (One-to-Many)
* Batch-Coffee Relations (One-to-Many)

### 6. UI/UX Components

#### 6.1 Navigation

* Main Navigation Bar
* Shop Selector
* Breadcrumb Navigation
* Quick Actions Menu

#### 6.2 Dashboard Views

* Retail Overview
* Roasting Dashboard
* Analytics Dashboard
* Reports View

#### 6.3 Interactive Elements

* Order Forms
* Confirmation Dialogs
* Stock Level Indicators
* Progress Tracking
* Real-time Updates

### 7. Security & Authentication

#### 7.1 Authentication

* Session-based Authentication
* Password Hashing
* Token Management
* Session Timeout

#### 7.2 Authorization

* Role-based Access Control
* Shop Access Verification
* API Route Protection
* Resource Permission Checks

### 8. Error Handling & Logging

#### 8.1 Client-side Error Handling

* Error Boundaries
* Form Validation
* Network Error Recovery
* User Feedback

#### 8.2 Server-side Error Handling

* Request Validation
* Database Error Handling
* Transaction Rollbacks
* Error Logging

### 9. Deployment & Infrastructure

#### 9.1 Hosting

* Replit Platform
* PostgreSQL Database
* Environment Configuration
* Automatic Migrations

#### 9.2 Scaling

* Database Connection Pooling
* Caching Strategies
* Query Optimization
* Resource Management

### 10. Future Enhancements

* Mobile Application
* Advanced Analytics
* External System Integration
* Automated Quality Control
* Customer Portal
* API Extensions
* Reporting Enhancements
* Machine Learning Integration