Krishikendra - Agricultural Supply Chain Management System

SRS Document:

Document: System Requirement Specification Document

Title: Krishikendra - Agricultural Supply Chain Management System

Objective:

The objective of Krishikendra is to establish an efficient agricultural supply chain management system that connects farmers, distributors, retailers, and consumers. The system aims to simplify and optimize the process of procuring, storing, transporting, and selling agricultural products, ensuring a smooth flow of goods from farmers to end consumers.

Scope:

The Krishikendra system will provide functionalities for farmer registration, crop listing, order placement and management, inventory management, transportation and logistics, pricing and payment, quality control, reporting and analytics, and user management
It will ensure fair and transparent transactions, improve supply chain operations, and offer valuable insights for better decision-making.

functional requirements:

1. Farmer Registration:

- Farmers should be able to register their details in the system, providing personal information, farming practices, and contact information.

2. Crop Listing:

- Farmers should be able to list the crops they have available for sale, specifying crop type, quantity, quality, and pricing.

3. Order Placement and Management:

- Distributors and retailers should be able to place orders for agricultural products, and the system should manage and track orders, including order confirmation, status updates, and delivery scheduling.

4. Inventory Management:

- Farmers should be able to manage their inventory within the system, updating crop availability, tracking stock levels, and ensuring quality control.

5. Reporting and Analytics:

- The system should generate reports and provide analytics on key metrics such as crop demand, sales, inventory levels, and profitability, to support decision-making.

Non-functional Requirements:

- Usability: The system should be intuitive and user-friendly, ensuring ease of use for all stakeholders.

- Availability: The system should be available and accessible to users at all times, with minimal downtime.

- Compatibility: The system should be compatible with different devices, browsers, and operating systems.

- Reliability: The system should be reliable, with robust data storage and backup mechanisms.

- Scalability: The system should be scalable to accommodate a growing number of users and increasing data volume.

- Security: The system should ensure the security and confidentiality of user data and transactions.

- Performance: The system should perform efficiently, with fast response times and minimal latency.

- User Privacy: The system should comply with data privacy regulations and protect user privacy.

Functional Operations:

- Farmer Registration
- Crop Listing
- Order Placement
- Order Management
- Inventory Management
- Transportation and Logistics
- Pricing and Payment
- Quality Control
- Reporting and Analytics
- User Management

Business Entities and Table Queries:

Business Entities:

- Farmer (FarmerID [PK], Name, Address, ContactNumber, Email)
- Crop (CropID [PK], CropType, Quantity, Quality, Price)
- Order (OrderID [PK], farmer id [fk], crop id[fk], Quantity, Status, DeliveryDate)
- User (UserID [PK], Username, Password, Role)

Table Creation Queries:

```
CREATE TABLE Farmer (
FarmerID INT PRIMARY KEY,
Name VARCHAR(100),
Address VARCHAR(200),
ContactNumber VARCHAR(20),
Email VARCHAR(100)
);

CREATE TABLE Crop (
CropID INT PRIMARY KEY,
FarmerID INT,
CropType VARCHAR(100),
Quantity INT,
Quality VARCHAR(50

),
Price DECIMAL(10, 2),
FOREIGN KEY (FarmerID) REFERENCES Farmer(FarmerID)
);

CREATE TABLE Order (
OrderID INT PRIMARY KEY,
DistributorID INT,
RetailerID INT,
```

```
CropID INT,
Quantity INT,
Status VARCHAR(50),
DeliveryDate DATE,
FOREIGN KEY (DistributorID) REFERENCES User(UserID),
FOREIGN KEY (RetailerID) REFERENCES User(UserID),
FOREIGN KEY (CropID) REFERENCES Crop(CropID)
);

CREATE TABLE User (
UserID INT PRIMARY KEY,
Username VARCHAR(50),
Password VARCHAR(50),
Role VARCHAR(50)
);
```