Software Requirements Specification

for

Farm-Direct : An AI Powered Online Market-Placed For Farmers

Version 1.0 approved

Prepared by : Hanuman Bavane 14108
Darshan Bele 14110
Shrutika Ghodake 14142
Mayuri Hirade 14152

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This document specifies the software requirements for **farmDirect**, a mobile-first platform that provides direct market access to farmers, enabling them to list produce, negotiate prices, and sell directly to buyers with integrated logistics, payments, and traceability. It also incorporates multi-language communication, a structured negotiation system, voice & image-based product capture, offline operation with SMS integration, role-based access control (RBAC), delivery tracking, and a built-in voice assistant.

1.2 Document Conventions

- Requirements are labeled as REQ-XX.
- High (H), Medium (M), Low (L) priority marked.
- Functional and Non-functional requirements are separated.

1.3 Intended Audience and Reading Suggestions

- Developers: For implementation details.
- Farmers: End users with minimal technical background.
- Customers & Retailers: Buyers of farm products.
- Researchers & Agri Experts: For providing expert advice modules.
- Delivery Partners: For logistics and tracking.

1.4 Product Scope

The application leverages digital marketplaces (AgriConnect model) to create a transparent ecosystem. It ensures:

- Direct farmer-to-consumer transactions.
- Negotiation system for fair pricing.
- Role-Based Access Control (RBAC) for security.
- Offline SMS-based ordering in low-connectivity regions.
- Voice assistant for accessibility.

1.5 Product Functions

- Farmer Module
- Register/login securely
- Add crop details (type, quantity, price, harvest date)
- Manage orders and payments
- Track sales through dashboard
- Buyer Module
- Register/login securely
- Search, filter, and compare produce
- Place orders and make secure payments
- Rate/review farmers and transactions
- Admin Module
- Manage disputes between farmers and buyers
- Verify farmer credentials and listings
- Generate reports and analytics
- Value-Added Features
- Real-time chat between farmers and buyers
- AI-powered crop price suggestions and demand prediction
- Geolocation-based recommendations for local buyers
- Multilingual support for regional accessibility

1.6 Target Users

Primary Users:

- Farmers (small, medium, and large-scale)
- Buyers (consumers, retailers, wholesalers, restaurants)

Secondary Users:

- Government agencies (monitoring compliance)
- Platform administrators (managing services and security)

1.7 Benefits

- For Farmers: Higher profit margins, wider market access, better price transparency.
- For Buyers: Fresh produce, competitive prices, and direct sourcing.
- For Ecosystem: Reduced exploitation, digital empowerment of rural communities, and efficient supply chain.

1.8 Operating Environment

- Web Application: Accessible on desktops and mobile browsers.
- Mobile Optimization: Responsive UI with possible hybrid app (React Native) for better adoption among rural farmers.
- Cloud Hosting: Ensures scalability for increasing user base

1.9 Constraints

- Internet availability in rural areas may limit accessibility.
- Requires multilingual support to accommodate farmers with limited English proficiency.
- Seasonal fluctuations may cause variations in platform activity.

1.10 Assumptions and Dependencies

- Farmers will have access to basic smartphones and internet.
- Buyers are willing to adopt direct-purchase platforms.
- Payment gateways (Razorpay/UPI/PayPal) remain reliable for rural transactions**.**

1.11 References

- AGRI CONNECT: A Transparent Digital Marketplace
- Smart Farming Technologies: IoT-based overview

2. Overall Description

2.1 Product Perspective

The proposed system is a mobile-first digital marketplace designed to extend the AgriConnect model of transparent transactions and incorporate the AI-driven advancements demonstrated by platforms like Ninjacart, DeHaat, and AgriBazaar. Unlike traditional agricultural supply chains where farmers face 30–50% loss in income due to middlemen and inefficiencies**,** this solution empowers farmers to directly connect with buyers, retailers, and consumers**,** thereby ensuring fair pricing, reduced food wastage, and improved profitability**.**

2.2 Product Functions

- Registration & RBAC (Farmer, Buyer, Delivery Partner, Admin).
- Product Listing: via text, image, or voice.
- Negotiation System: Buyer & farmer bargain with AI-suggested fair price.
- Multilingual Chat/Voice Assistant.
- Offline SMS Mode for farmers in low-connectivity zones.
- AI-driven Demand Forecasting for price stability.
- Delivery Tracking (GPS-based).
- Blockchain-backed transaction logs for transparency.

2.3 User Classes

- Farmer: Semi-literate, prefers voice/SMS-based interaction.
- Buyer/Retailer: Businesses seeking bulk or retail produce.
- Delivery Agent: Handles logistics.
- Admin: Maintains system integrity, fraud detection.

2.4 Operating Environment

- Frontend: React.js
- Backend: Node.js with Express
- Database: MySQL
- API: RESTful APIs
- Hosting: AWS/Azure cloud with Docker & Nginx.
- Mobile Support: React Native (Android/iOS) optional.
- Messaging: SMS Gateway integration for offline orders/alerts.
- Payments & Security: UPI/Razorpay, JWT-based RBAC, HTTPS/TLS.

2.5 Design and Implementation Constraints

- Must work in low-network areas (offline + SMS).
- Support low-end devices (lightweight, voice-first).
- MySQL scalability limits → needs caching.
- Security & RBAC required.
- AI/ML & GPS depend on external APIs.
- Payment compliance (RBI/PCI-DSS).

2.6 User Documentation

- Farmer Guide: Multilingual manual + voice tutorials.
- Buyer Guide: Using marketplace, negotiation, payments, delivery tracking.
- Delivery Guide: Order handling & GPS tracking steps.
- Admin Guide: RBAC, fraud checks, system monitoring.
- Help & FAQs: In-app text + voice support.
- Video/Voice Tutorials: For low-literacy users (listing, SMS, voice assistant).

2.7 Assumptions and Dependencies

- Farmers have basic smartphones/feature phones; SMS works in low-network areas.
- Users adopt digital payments (UPI, wallets) for secure transactions.
- System depends on SMS gateway, payment gateway, GPS APIs, and cloud hosting.
- AI/ML modules (price prediction, image recognition, negotiation) rely on external APIs/models.
- Government agri policies and market regulations influence pricing and compliance.

3. External Interface Requirements

3.1 User Interfaces

- Farmer UI: Simple dashboard, multilingual support, large icons, voice/image crop listing, SMS fallback.
- Buyer UI: Product search, negotiation chat, payment gateway, delivery tracking.
- Delivery Partner UI: Order acceptance, GPS-based tracking, status updates.
- Admin UI: RBAC-based control panel, fraud detection, transaction logs.
- Accessibility: Voice assistant, text-to-speech, low-literate friendly design..

3.2 Hardware Interfaces

- Smartphones (Android/iOS): For farmers, buyers, and delivery partners (camera, mic, GPS enabled).
- Feature Phones: For SMS-based offline access.
- GPS Devices: Integrated in smartphones for delivery tracking.
- Cloud Servers: Hosting backend, database, and AI services.
- Admin Systems (PC/Laptop): For monitoring, analytics, and RBAC management.

3.3 Communications Interfaces

- REST APIs (HTTPS): For secure app–server communication.
- WebSockets: Real-time negotiation, chat, and price updates.
- SMS Gateway: Offline product listing, orders, and OTP in low-network areas.
- GPS/Map APIs: Delivery tracking and location-based services.
- Payment APIs: UPI/Razorpay for secure digital payments.
- Cloud Services: Sync AI modules (price prediction, image recognition, voice assistant).

4. System Features

The system offers multilingual support, AI-based negotiation, voice and image product listing, and offline SMS mode for low-network areas. It includes delivery tracking, RBAC security, a voice assistant, and an integrated chatbot for 24/7 support, creating a farmer-friendly and AI-driven marketplace.

4.1 Multilingual Communication

REQ-1: Support 10+ Indian languages + English.

REQ-2: Text-to-speech and speech-to-text integration**.**

4.2 AI-Enabled Negotiation

REQ-3: Farmers and buyers negotiate digitally.

REQ-4: AI recommends fair price based on demand/supply data**.**

4.3 Voice & Image Product Listing

REQ-5: Farmers upload crops via voice description or photo upload.

REQ-6: AI auto-identifies crop from image (e.g., “tomatoes, 50kg”).

4.4 Offline SMS Mode

REQ-7: Farmers can send structured SMS (“LIST Tomato 50kg 2000Rs”).

REQ-8: Orders sync when back online**.**

4.5 Delivery Tracking

REQ-9: GPS-based tracking for delivery agents.

REQ-10: Customers get real-time ETA.

4.6 RBAC Security

REQ-11: Farmers, Buyers, Delivery, Admin have distinct permissions.

REQ-12: Payments & sensitive data secured by role restrictions.

4.7 Voice Assistant

REQ-13: Farmers can ask “What is today’s onion price?”

REQ-14: Assistant helps in navigation & updates.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- *The system shall support **5,000+ concurrent users** with an average response time <3 seconds.*
- *Synchronization of offline/SMS orders must complete within **5 minutes** once connectivity is restored.*
- *The platform shall provide **99.5% uptime** excluding scheduled maintenance.*
- *The farmer dashboard and marketplace homepage should load within **5 seconds** on a 3G+ network.*
- *System performance should scale to **1 million API calls/month**.*

5.1 Safety Requirements

- *Automatic **data backup and restore** to prevent permanent loss.*
- *All critical transactions (orders, payments) must have rollback and error-handling mechanisms.*
- *Failed payments should trigger retries and log entries for later resolution.*
- *GPS-based delivery updates must validate integrity to prevent false entries.*
- *The platform must comply with **government food safety and trade regulations**.*

5.2 Security Requirements

- ***JWT authentication** for all users; **bcrypt hashing** for password storage.*
- ***Role-Based Access Control (RBAC):** farmers, buyers, delivery partners, and admins have distinct permissions.*
- ***KYC verification** required for farmers and sellers before trading.*
- *All communication must use **HTTPS/TLS encryption**.*

- *Sensitive data (e.g., payments) secured using **AES-256 encryption**.*
- *Admins must use **multi-factor authentication (MFA)**.*
- ***API Logger** will track all system activity to detect suspicious behavior.*

5.3 Software Quality Attributes

- ***Usability:** Multilingual UI with text-to-speech and voice inputs for semi-literate farmers.*
- ***Reliability:** System must recover from failures within **30 minutes**.*
- ***Maintainability:** Modular MERN stack design for independent updates.*
- ***Scalability:** Must scale to **50,000 users within 3 years**.*
- ***Interoperability:** REST APIs for integration with payment and logistics systems.*
- ***Portability:** Compatible with Android (8+), iOS (12+), and all major browsers.*
- ***Testability:** At least **80% unit test coverage** on core modules.*

5.4 Business Rules

- *Farmers must complete **KYC verification** before listing products.*
- *Buyers must complete payment before delivery scheduling.*
- *Negotiation system can only be enabled if allowed by the seller.*
- *Each successful order must generate an **invoice** (downloadable/email).*
- *Refunds should be processed within **7 business days**.*
- *Disputes must be resolved by admin within **72 hours**.*
- *Large transactions (>₹50,000) require enhanced verification (compliance with RBI).*

6. Other Requirements

- ***Database:*** *MongoDB (with Mongoose ORM).*
- ***Frontend:*** *React.js + Tailwind CSS.*
- ***Backend:*** *Node.js + Express.js.*
- ***Hosting:*** *Cloud-based (AWS/Azure/GCP TBD).*
- ***Payment:*** *Razorpay API with UPI, net banking, cards.*
- ***Notifications:*** *Email/SMS alerts via Nodemailer & SMS gateway.*
- ***Internationalization:*** *Support for* ***10+ Indian languages + English****.*
- ***Legal Compliance:*** *RBI, IT Act 2000, GDPR (for global expansion).*

Appendix A: Glossary

- ***Farm-Direct:*** *AI-powered digital marketplace connecting farmers, buyers, and sellers.*
- ***RBAC (Role-Based Access Control):*** *Security model limiting access based on user roles.*
- ***Story System:*** *Ledger + Event Management for farmers to track crop activities and finances.*
- ***Negotiation System:*** *Real-time buyer–farmer price discussion.*
- ***Invoice Management:*** *Auto-generation of invoices after successful payment.*
- ***Expert Connect:*** *Module allowing farmers to consult agricultural experts.*
- ***API Logger:*** *Tracks all API activity for monitoring and security.*
- ***KYC (Know Your Customer):*** *Verification required before trade.*
- ***Razorpay:*** *Payment gateway used for secure transactions.*
- ***SMS Gateway:*** *Offline ordering mechanism for low-connectivity regions.*

Appendix B: Analysis Models

1) ***Use Case Diagrams***
   - *Farmer ↔ Buyer (product listing, negotiation, payment).*
   - *Seller ↔ Farmer (input purchase, ledger update).*
2) ***ER Model***
   - *Entities: Farmer, Buyer, Seller, Product, Order, Payment, Invoice, Expert.*
3) ***System Architecture***
   - *MERN stack with modular services (Auth, Product, Payment, Expert Connect, Logger).*
4) ***DFD (Level 0 & 1)***
   - *User requests → API Gateway → Database → Response.*
   - *Modules: Story, Negotiation, Invoice, Payment, Expert Connect.*
5) ***Class Diagram***
   - *User (Farmer, Buyer, Seller, Admin)*
   - *Product, Order, Invoice, Story, Payment, ExpertSession, APILogger.*

Appendix C: To Be Determined List

- ***Final cloud provider (AWS/Azure/GCP).***
- ***Exact list of supported regional languages.***
- ***Maximum scaling limits (concurrent users, orders/day).***
- ***AI modules (basic recommendation vs. predictive analytics).***
- ***Data retention period for invoices/logs.***
- ***Final KYC provider/integration method.***
- ***SMS gateway provider for offline ordering.***
- ***Export compliance for international expansion.***