

Software Requirements Specification

AgroSphere – Smart Agricultural Support & Sustainable Solution



Submitted by

1. Sheraz Rafiq (F22BINFT1M01083)
2. Muhammad Naveed (F22BINFT1M01112)

Submitted to

Dr. Musarat Karim

Department of Information Technology

Faculty of Computing

The Islamia University of Bahawalpur

Meeting Details

Sr No	Details	Date	Supervisor Signature

--	--	--

Summary

This document provides the Software Requirements Specification for "AgroSphere – Smart Agricultural Support & Sustainable Solution," a web-based platform connecting farmers with agricultural service providers through AI-powered crop diagnosis, product marketplaces, service bookings, and community engagement features.

Table of Contents

1. Introduction

1.1 Purpose

1.2 Scope

1.3 Product Perspective

1.4 User Characteristics

1.5 Similar Apps and Systems/Literature Review

1.6 Proposed Technologies

2. Requirements

2.1 Functional Requirements

2.1.1 Crop Disease Diagnosis

2.1.2 Product Marketplace

2.1.3 Solar Service Booking

2.1.4 Event Management

[2.1.5 User Registration](#)

[2.2 Non-Functional Requirements](#)

3. Use Cases and Flow of Processes

[3.1 Use Case 1: AI Crop Disease Diagnosis](#)

[3.2 Use Case 2: Product Comparison & Purchase](#)

[3.3 Use Case 3: Solar Service Booking](#)

[3.4 Use Case 4: Event Registration](#)

[3.5 Use Case 5: Role-Based Registration](#)

4. References

1. Introduction

AgroSphere is a comprehensive web-based platform designed to connect farmers with agricultural service providers including pesticide companies, field officers, and solar irrigation installers through AI-powered crop diagnosis and digital service management.

1.1 Purpose

To provide farmers with fast, reliable agricultural solutions through AI diagnosis, expert assistance, product access, and service connections in a unified digital platform.

1.2 Scope

The system includes AI crop diagnosis, pesticide product marketplace, solar service booking, event management, and multi-role user management. Excludes payment processing, logistics, and mobile app in initial phase.

1.3 Product Perspective

A standalone web platform that can integrate with third-party services and may expand to mobile applications.

1.4 User Characteristics

Farmers (basic tech skills), Pesticide Companies (moderate), Field Officers (moderate), Solar Installers (basic), Admin (advanced).

1.5 Similar Apps and Systems/Literature Review

Plantix (AI diagnosis only), AgroStar (product sales only), FarmLogs (data management only). AgroSphere integrates all these features.

1.6 Proposed Technologies

Frontend: React.js, Tailwind CSS; Backend: Node.js, Python; Database: MongoDB; AI: TensorFlow/PyTorch; Cloud: AWS S3.

2. Requirements

2.1 Functional Requirements

2.1.1 Crop Disease Diagnosis

- **ID:** FR001
- **Purpose:** Farmers upload crop images for AI diagnosis
- **Users:** Farmer, Field Officer
- **Input:** Crop image, symptoms, location
- **Output:** Diagnosis, confidence score, recommendations

2.1.2 Product Marketplace

- **ID:** FR002
- **Purpose:** Companies list products; farmers compare and contact
- **Users:** Pesticide Company, Farmer
- **Input:** Product details, pricing, offers
- **Output:** Product catalog, comparison view

2.1.3 Solar Service Booking

- **ID:** FR003
- **Purpose:** Farmers book solar installation/maintenance
- **Users:** Solar Installer, Farmer
- **Input:** Service type, location, requirements
- **Output:** Service booking, quotations

2.1.4 Event Management

- **ID:** FR004
- **Purpose:** Promote agricultural events and training
- **Users:** All roles
- **Input:** Event details, registration links
- **Output:** Event calendar, notifications

2.1.5 User Registration

- **ID:** FR005
- **Purpose:** Secure registration with role-based access
- **Users:** All roles
- **Input:** User details, role selection
- **Output:** User account, dashboard access

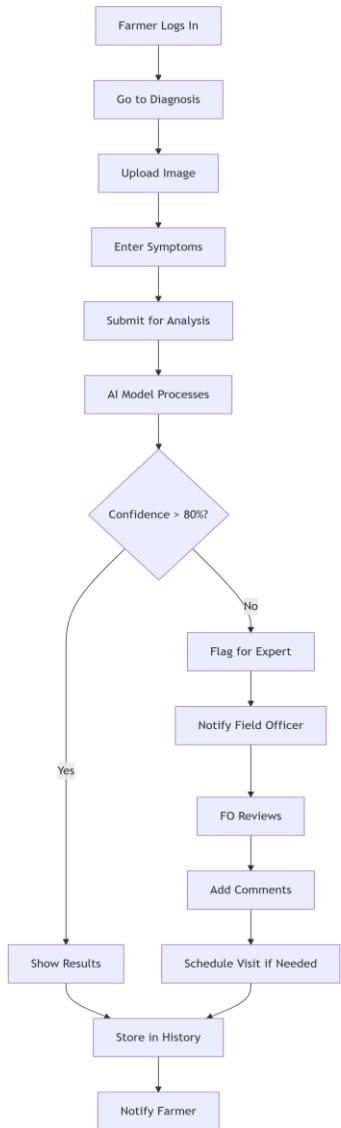
2.2 Non-Functional Requirements

- Performance: Page load < 3s, AI response < 10s
- Availability: 99.5% uptime
- Security: SSL, role-based access
- Usability: Responsive, multilingual

3. Use Cases and Flow of Processes

3.1 Use Case 1: AI Crop Disease Diagnosis

Flow Diagram:



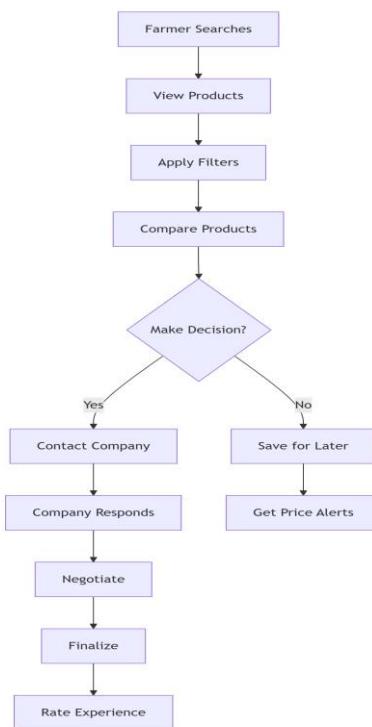
Use Case Table:

ID	UC001
Name	AI Crop Disease Diagnosis
Description	Farmer uploads crop image for AI diagnosis with expert fallback
Requirement(s)	FR001
Actor(s)	Farmer, Field Officer

Precondition	Farmer logged in, has crop image
Postcondition	Diagnosis stored, notifications sent
Basic Flow	<ol style="list-style-type: none"> 1. Upload image+symptoms 2. AI analyzes 3. If confidence high, show results 4. Else notify FO 5. FO reviews 6. Notify farmer
Alternative Flow	FO directly diagnoses without AI
Exceptions	Invalid image, AI unavailable

3.2 Use Case 2: Product Comparison & Purchase

Flow Diagram:

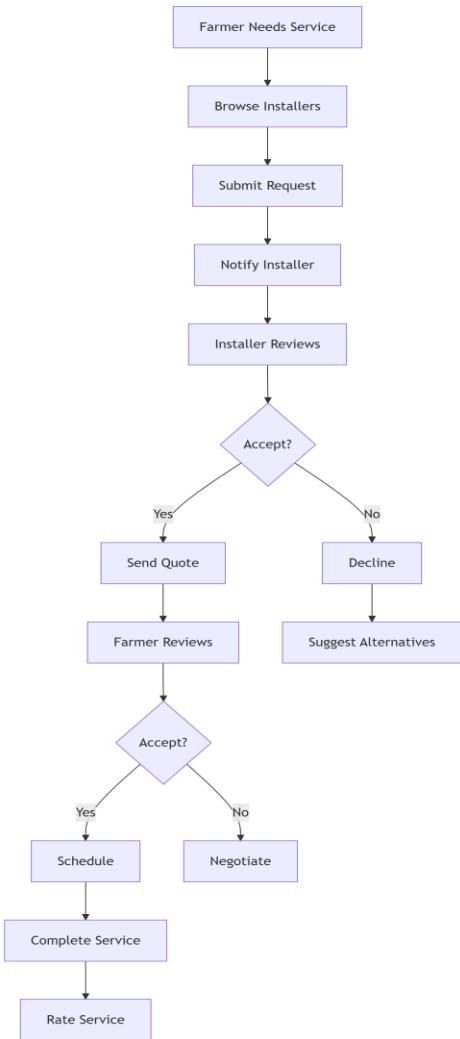


Use Case Table:

ID	UC002
Name	Product Comparison & Purchase
Description	Farmer compares pesticide products and contacts companies
Requirement(s)	FR002
Actor(s)	Farmer, Pesticide Company
Precondition	Products listed, farmer logged in
Postcondition	Contact established, potential sale
Basic Flow	<ol style="list-style-type: none"> 1. Search products 2. Filter and compare 3. Contact company 4. Negotiate 5. Rate experience

3.3 Use Case 3: Solar Service Booking

Flow Diagram:



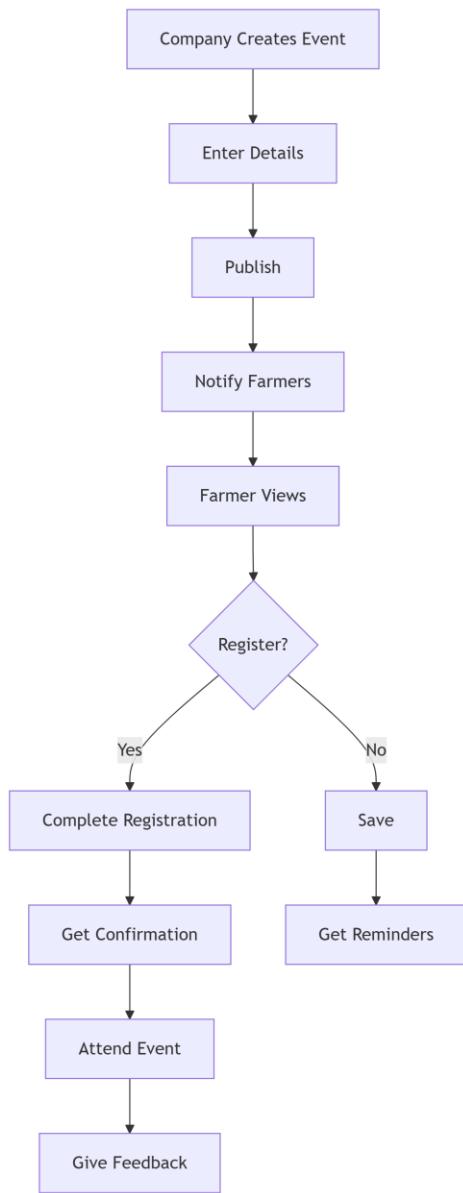
Use Case Table:

ID	UC003
Name	Solar Service Booking
Description	Farmer requests solar irrigation services
Requirement(s)	FR003
Actor(s)	Farmer, Solar Installer
Precondition	Installers registered, farmer logged in

Postcondition	Service booked or declined
Basic Flow	<ol style="list-style-type: none">1. Submit request2. Installer responds3. Quote negotiation4. Schedule service5. Rate service

3.4 Use Case 4: Event Registration

Flow Diagram:



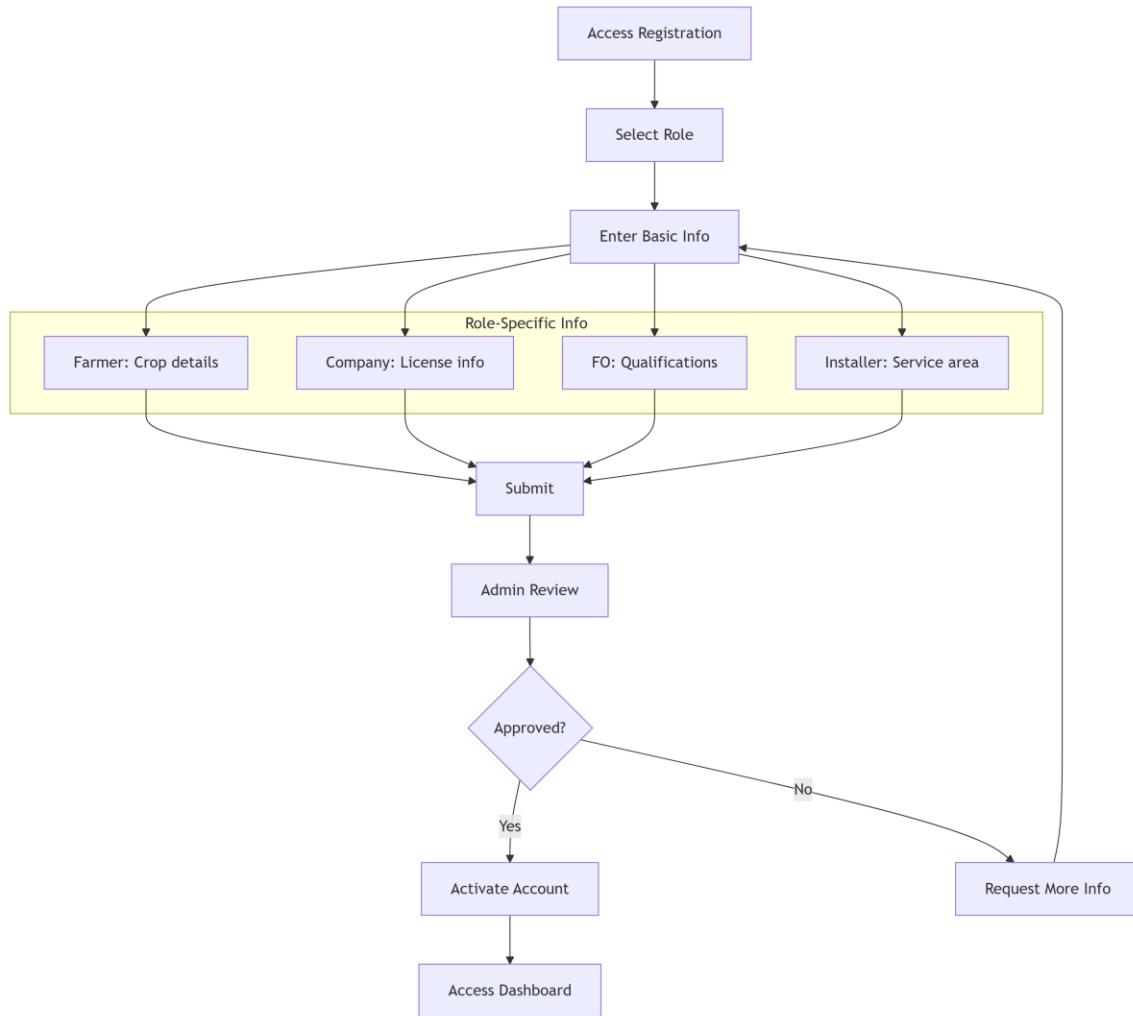
Use Case Table:

ID	UC004
Name	Event Registration
Description	Farmers register for agricultural events
Requirement(s)	FR004

Actor(s)	Company, Farmer
Precondition	Event created, farmer logged in
Postcondition	Registration completed or saved
Basic Flow	<ol style="list-style-type: none"> 1. View event 2. Register interest 3. Get confirmation 4. Attend event 5. Provide feedback

3.5 Use Case 5: Role-Based Registration

Flow Diagram:



Use Case Table:

ID	UC005
Name	Role-Based Registration
Description	User registers with specific role and permissions
Requirement(s)	FR005
Actor(s)	All roles, Admin

Precondition	New user, valid email
Postcondition	Account created with role permissions
Basic Flow	<ol style="list-style-type: none"> 1. Select role 2. Enter details 3. Submit for approval 4. Admin reviews 5. Account activated

4. References

1. Project Proposal: "AgroSphere – Smart Agricultural Support & Sustainable Solution"
2. IEEE Std 830-1998, Software Requirements Specifications
3. Plantix App: <https://plantix.net>
4. React.js Documentation: <https://reactjs.org>
5. MongoDB Documentation: <https://docs.mongodb.com>