Date 18/1/24

Preparing an SRS using Rational Requisite Pro

Objectives:

- → Understanding the IEEE standard of SRS.
- → Preparing Requirement management plan.
- → Preparing SRS document for E-AGRI
- → Finding features of the Software systems
- → Finding Usecase requirements
- → Finding Stakeholders requirements
- → Finding Supplementary requirements
- → Mapping Features with other requirements
- → Preparing Attribute matrix
- → Preparing Traceability matrix

E-AGRI

1. INTRODUCTION

1.1) Purpose

The purpose of the E-AGRI software is to create a platform that connects farmers directly with end customers, eliminating the need of intermediaries. The Software aims to empower farmers by enabling them to sell their agricultural products directly to consumers, thereby reducing dependence on multinational corporations (MNCs) and minimizing Agricultural waste.

Teacher's Signature:

1.2) Document Conventions

SRS version: 1.0
Date : 25/01/2024
Author : Sanjai Sk

1.3) Intended Audience and Reading Suggestions

This document is intended for developers, testers, project managers, and stakeholders involved in the E-AGRI project. It provides an overview of the project requirements and serves as a reference for system development.

1.4) Project Scope

The E-AGRI software will include features for direct transactions b/w farmers and customers, information dissemination on agricultural development, and tools to enhance the overall agriculture ecosystem.

1.5) References

→ Krisol ERP
→ CropIn SmartFarm
→ PanAgro
→ Satya FIMS
→ Aahaar ERP

2. OVERALL DESCRIPTION

2.1) Product Perspective

The E-AGRI software is a standalone system that act as a bridge b/w farmers and consumers. It is not dependent on external systems but may integrate with external databases for agricultural information.

2.2) Product Features

1. Direct Transaction: Allow farmers to list and sell their agricultural products directly to customers.
2. Information Hub: Provide agricultural information, tips, and updates to users.
3. User Profiles: Farmers and customers can create profile with relevant information.
4. Payment Gateway: Facilitate seure online transactions.
5. Rating and Feedback: Allow users to rate and provide feedback on transactions.

2.3) User Classes and Characteristics

Farmer: Primary users who list and sell agricultural products.

Customers: Individuals or business who purchase products from farmers.

2.4) Operating Environment

The software will be accenible through web browsers on desktop and mobile devices. It should be compatible with major browsers.

2.5) Design and Implementation Constraints

The software should be developed using HTML, CSS, Javascript, react, node JS, MYSQL

2.6) Assumptions and Dependencies

→ Users have accen to the internet for online transactions.

→ Farmers and customers are responsible for their account information.

3. SYSTEM FEATURES

3.1) Functional Requirements

3.1.1) Direct transaction

→ Farmers can list their products with detail

→ Customers can browse and purchase products

→ A secure payment gateway for transactions

3.1.2) Information Hub

→ Agricultural news and updates

→ Tips for farmers on crop cultivation

→ Weather information.

4. EXTERNAL INTERFACE REQUIREMENTS

4.1) User Interfaces

→Intuitive and user-friendly interfaces for farmers and customers.

→Responsive design for various devices.

4.2) Hardware Interfaces

Standard Hardware for accessing web applications (desktops, laptops, tablets, smartphones).

4.3) Software Interfaces

Compatibility with major web browsers (Chrome, FireFox, Safari, Edge, Brave)

4.4) Communications Interfaces

Secure data transfer over HTTPS.

5. NONFUNCTIONAL REQUIREMENTS

5.1) Performance Requirements

→Response time for transaction: [1 min]

→Concurrent user support : [1000]

5.2) Safety Requirements

→Encryption of sensitive user data.

→Regular data backups.

5.3) Security Requirements

→ User authentication and authorization.

→ Protection against common web vulnerabilities (SQL injection, XSS).

5.4) Software Quality Attributes

Reliability: The system should be available and reliable during peak usage.

Usability: Intuitive interfaces for both farmers and customers.

Maintainability: Code should be well-documented and modular.

Result:-

Thus, I was successfully completed my software requirement specifications for my E-AGRI software using Rational Requisite Pro.