

**Software Requirements Specification**

**For**

**Farm Management System**

**Prepared By:**

**Mohammed Nihal Mantri**

**PES1UG22CS356**

**Safdar Hakeem Ahmed**

**PES1UG23CS822**

**Synopsis/Purpose**

• The main objective of the project is to design and develop a user friendly-system

• Easy to use and an efficient computerized system.

• To develop an accurate and flexible system, it will eliminate data redundancy.

• To study the functioning of Farm management System.

• To make a software fast in processing, with good user interface.

• To make software with good user interface so that user can change it and it should be used for a long time without error and maintenance.

• To provide synchronized and centralized farmer and seller database.

• Computerization can be helpful as a means of saving time and money.

• To provide better Graphical User Interface (GUI).

• Less chances of information leakage.

• Provides Security to the data by using login and password method.

• To provide immediate storage and retrieval of data and information.

• Improving arrangements for farmers co-ordination.

• Reducing loss.

**Introduction**

In the evolving world of agriculture, effective farm management is pivotal for achieving operational excellence and maximizing profitability. The Farm Management System (FMS) emerges as a transformative business solution designed to address the multifaceted needs of modern farming enterprises. By leveraging advanced technology, this product offers

a comprehensive suite of tools to enhance farm management practices, streamline operations, and drive business growth.

A Farm Management System is not merely a software product; it is a strategic asset for agricultural businesses aiming to optimize their operations. This system integrates various functionalities, including real-time data analytics, resource management, and financial tracking, into a unified platform. Its primary goal is to provide farmers with the insights and tools needed to make informed decisions, manage resources efficiently, and boost overall productivity.

**Overall Description**

The Farm Management System (FMS) is a cutting-edge solution designed to revolutionize agricultural operations by integrating advanced technology with farm management practices. This comprehensive system serves as a crucial business tool for modern farming enterprises, aiming to enhance efficiency, productivity, and profitability across various aspects of farm management.

**External Interface Requirements**

**Web Portal:** for Farmers: A user-friendly web application where farmers can create and manage their profiles, access agricultural information, and receive updates and notifications. This portal would use HTML, CSS, and JavaScript for the frontend, with Python-Flask serving the backend.

**Mobile Application:** A companion app for smartphones to allow farmers to access their profiles, receive alerts, and interact with the system on the go.

**API Integration**: The system may provide APIs for integration with other agricultural databases, or external applications that might need to access or update farmer data.

**Data Export and Import**: Features to import data from or export data to external systems or spreadsheets for analysis, sharing, or backup purposes.

**Analysis Models**

**Data Flow Diagrams:** To illustrate the flow of data through the system.

**Entity-Relationship Diagrams:** To describe the database schema.

**System Features**

**User-Friendly Design**

* Intuitive Interface: A clean, easy-to-navigate graphical user interface (GUI) designed with simplicity in mind to ensure that users of all tech levels can operate it efficiently.
* Customizable Dashboard: Allows users to customize their dashboard for quick access to relevant information and functions.

**Efficiency and Accuracy**

* Real-Time Data Processing: Fast processing capabilities to handle large volumes of data with minimal delay.
* Flexible Data Management: Accurate data entry and retrieval processes to minimize errors and ensure up-to-date information.

**Data Management**

* Centralized Database: A synchronized, centralized database for farmers and sellers to avoid data redundancy and ensure consistency.
* Data Redundancy Elimination: Mechanisms to detect and remove duplicate entries, reducing redundancy.

**System Functionality**

* Farm Management Functions: Tools for tracking farm activities, crop management, and resource allocation.
* Reporting and Analytics: Comprehensive reporting tools for performance analysis, financial tracking, and other key metrics.

**Other Nonfunctional Requirements**

**Scalability Requirements**

Requirement: The platform must be able to support up to 10,000 concurrent users without degradation in performance.

Verification: Load testing to ensure the system can handle high user volumes.

**Availability Requirements**

Requirement: The platform must be available 24/7 with minimal downtime, allowing for scheduled maintenance outside peak hours.

**Usability Requirements**

Requirement: The platform must be intuitive, with user-friendly navigation and accessibility features, including support for screen readers.

**Other Requirements**

User requirements address the need for user-friendly experiences, including training resources, feedback mechanisms, and customization options.

Performance requirements include scalability to handle increased loads, such as more users or larger datasets, and performance testing to ensure responsiveness under various conditions.

Security requirements involve robust measures for protecting data, including incident response procedures and maintaining audit trails for tracking system and user activities.

  Appendices

Appendix A: Glossary

**Appendix A: Glossary**

the glossary simplifies technical terms and concepts, making them more accessible for farmers who may not be familiar with complex technology language

| **Term** | **Definition** |
| --- | --- |
| **API (Application Programming Interface** | A set of rules that allows different software programs to talk to each other and share information. |
| **Authentication** | Checking that someone is who they say they are, usually by entering a username and password. |
| **Centralized Database** | A single place where all the important data is stored and can be accessed by everyone who needs it. |
| **Compliance** | Following laws and rules to make sure everything is done correctly, like protecting your personal information. |
| **Data Encryption** | Changing data into a secret code so that only authorized people can read it and keep it safe from others. |
| **Farmer Profile** | A personal record for each farmer that includes their details, farm information, and what crops they grow. |
| **Graphical User Interface (GUI)** | The visual part of a computer program that lets you interact with it using things like buttons and pictures. |
| **Integration** | Making different software programs work together and share information. |

**Appendix B: Field Layouts**

This appendix contains the layouts and descriptions of key fields in your project, including user registration forms, task creation forms, and database schema fields.

**1.User Registration Form Layout**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Description** | **Data Type** | **Validation Rules** |
| Username | Unique identifier for the user | String | Must be unique; max length: 30 characters |
| Email | User's email address | String | Must be a valid email format |
| Password | Password for user authentication | String | Min length: 8 characters; must include numbers |
| Phone Number | User's contact number | String | Must be a valid phone number format |

**Appendix C: Requirement Traceability Matrix (RTM)**

Here’s a Requirement Traceability Matrix (RTM) for a farmer management system, outlining how each requirement is mapped to its design, development, and testing phases:

| **Requirement ID** | **Requirement Description** | **Design Specification** | **Development Task** | **Testing Scenario** | **Status** |
| --- | --- | --- | --- | --- | --- |
| **REQ-001** | System must allow users to create a unique farmer  profile. | Profile creation form with unique ID field. | Implement profile creation with unique ID generation. | Test profile creation for uniqueness and proper ID assignment. | Completed |
| **REQ-002** | Users must be able to input farm details, including type and size. | Farm details section in profile form. | Develop fields for farm type, size, and description. | Verify proper input and storage of farm details. | Completed |
| **REQ-00**  **3** | System must support multiple languages for user interface. | Language selection dropdown in UI. | Implement multi-language support and translations. | Test language switch functionality and translation accuracy. | In Progress |
| **REQ-004** | Users must be able to view and update their contact information. | Contact information section in user profile. | Implement contact information update feature. | Test update functionality and correct display of updated info. | Completed |
| **REQ-005** | The system must handle data synchronization for multiple users. | Design for real-time data synchronization. | Develop real-time synchronization features. | Test data synchronization among multiple users. | In Progress |