

**Department of Computer Engineering**

**Academic Term: First Term 2023-24**

**Class: T.E /Computer Sem – V / Software Engineering**

<b>Practical No:</b>	<b>1</b>
<b>Title:</b>	<b>Software Requirement Specification</b>
<b>Date of Performance:</b>	<b>26/07/2023</b>
<b>Roll No:</b>	<b>9614</b>
<b>Team Members:</b>	<b>Mudabbir(9589),Muhammad(9588), Nathan(9597)</b>

**Rubrics for Evaluation:**

<b>Sr. No</b>	<b>Performance Indicator</b>	<b>Excellent</b>	<b>Good</b>	<b>Below Average</b>	<b>Total Score</b>
1	On time Completion & Submission (01)	01 (On Time )	NA	00 (Not on Time)	
2	Theory Understanding(02)	02(Correct )	NA	01 (Tried)	
3	Content Quality (03)	03(All used)	02 (Partial)	01 (rarely followed)	
4	Post Lab Questions (04)	04(done well)	3 (Partially Correct)	2(submitted)	

**Signature of the Teacher:**

# EXPERIMENT NO 1

## 1. Abstract

In India, most of the population is dependent on agriculture and Farmer is the backbone of the Indian economy. The system consists of smartphone applications. The System can be used by farmers on the Android application on mobile devices. The paper proposes the utilization of information mining to give suggestions to ranchers on crops, crop data, and distinguishing proof of suitable manure. The smartphone is used very commonly by everyone who is educated or uneducated. This application can provide different languages, which is Hindi and English, These languages is commonly used and easy to use on the system.

## 2. Introduction

### 2.1 Purpose

Agriculture is a prime food source in India and Farmer is the backbone of the Indian economy. Today's words are Modern World in that every person uses a smartphone when the person is educated or uneducated not depending on education which is related to technology. Now day by day farmer rates are reduced and new farmers do not have proper knowledge about farming. This application's purpose is to provide farming-related information. Seeds information is more important than farming because when you know about seeds then you can use them in farming according to a farm.

### 2.2 Scope

Any software, any program, or any web application can never be said complete at all times in all senses because, in the field of information technology, most things change more frequently according to the need that arises in the changed condition. Likewise in this program also there is a scope for improvements. The proposed system may need the following modification for better performance:

1. There is a scope to provide the facility to take a backup of the database from the proposed system from time to time so that the data can be restored.
2. Payment option in purchasing time.

### 2.3 Definitions, Acronyms, Abbreviations

Not applicable.

## 2.4 References

1. DR. Maheshkumar, M. Pavithra "Forming Assistant Web Service" [www.ijraset.com](http://www.ijraset.com), IC Value: 45.98, Volume 5 Issue IV, ISSN: 2321-9653, April 2017.
2. Prof. Aradhana D, Shiva Prasad K S, Shrivaiashnavi J K, P. Sowmya, Tina Agarwal "AGRICULTURE BASED ANDROID APPLICATION" [www.ijraset.com](http://www.ijraset.com), ISSN: 2349-3224, Volume 3 Issue 2, May 2016.
3. Minwoo Ryu, Jaeseok Yun, Ting Miao, Il-Yeup Ahn, Sung-Chan Choi, Jaeho Kim "Design and Implementation of a "Connected Farm for Smart Farming System" DOI:10.1109/ICSENS.2015.7370624, November 2015.
4. Santosh G. Karkhile, Sudarshan G. Ghuge "A Modern Farming Techniques using Android Application" International Journal of Innovative Research in Science, Engineering and Technology (An ISO 3297: 2007 Certified Organization) Vol. 4, Issue 10, October 2015.
5. Kiran Shinde, Jerrin Andrei, Amey Oke "Web Based Recommendation System for Farmers" IJRITCC, Available @ <http://www.ijritcc.org>, March 2015.

## 2.5 Developer's Responsibilities

The developer is responsible for (a) developing the system, (b) conducting any user training that might be needed for using the system, and (c) maintaining the system for a period of one year after installation.

# 3. General Description

## 3.1 Functions Overview

The system diagram considers three main parts which are the user, farmer application, and admin. The Admin and user can access the application with the help of User ID and Password, which is different for every person that's providing security and avoiding duplication of the User ID or User Name. The user can open the application by using the ID & password and give information about farming according to the seasons. The Admin also opens the Dashboard by using the admin ID & password. Admin provided the authority to add, edit, or update & delete the data. The application provides interface communication to the user and admin. One more thinks Admin can sell their own products by using the application. That purpose providing a marketing interface for admin and user.

## 3.2 User Characteristics

The cornerstone notation for data modeling is the entity-relationship diagram. Three components are used in relational database design. These are Entities, Attributes, and Relationships. The user modules have one of the Android application interfaces provided to the user. This application has many options for helping the user just as login, registration, Farming information, seed information, Soil information, Animal information, Weather, item porches, and last one is contact information. This application provides two languages which are English and Hindi. All about the information provided in the listed form. Login and registration is the common process for accessing application information which is already discussed. Farming information is seeds & soil information provided to the user or farmer. Which type of seed grows better way in which type of soil and side by side Animal farming gives more profitability to the farmer. Weather reporting is just information providing users or farmers with today's weather according to state and city.

### 3.3 General Constraints

Not Applicable

### 3.4 General Assumptions and Dependencies

Not applicable.

## Software Requirements

### 4.1 Inputs and Outputs

The cornerstone notation for data modeling is the entity-relationship diagram. All types of authority in the application are provided to the admin to handle or manage the application. All database management, information management, information insert, information update, or information deletes any type of

authority in the hands of admin. Login, dashboard, customer, item information there is an interface provided to the admin. Admin Simple by using Admin ID & password enter the admin dashboards on those dashboards there are options for baseboards, Customer, and item information. In customer, there list of

customers registered in the application, and all details on the list form can see you. Admin can delete someone's personal information. One option is Item Information in that there are three sub-parts which are item insert, product sale

information, and product sale. Item inserts to insert the information about

farming or product related. Product sale information and product sale that according to marketing add and remove products and also any user can buy the product that's one notification given to the Admin and all about that information.

#### 4.2 Functional Requirements

Maintenance: The application should be at least managed once a month.

#### 4.3 External Interface Requirements

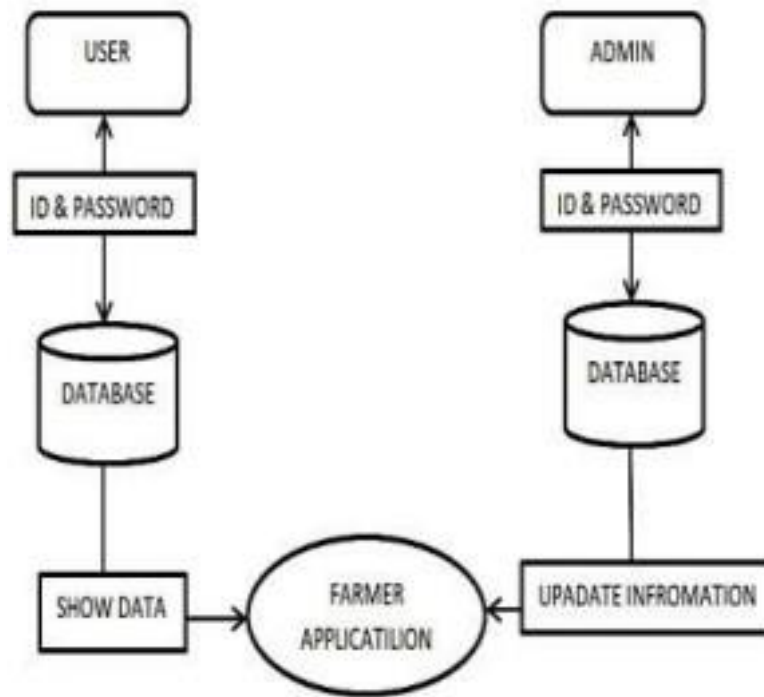
Not Applicable

#### 4.4 Performance Constraints

Not Applicable

#### 4.5 Design Constraints

The system diagram considers three main parts which are the user, farmer application, and admin. The Admin and User can access the application with the help of User ID and Password, which is different for every person that's providing security and avoids duplication of the User ID or User Name. The user can open the application by using the ID & password and give information about farming according to the seasons. The Admin also opens the Dashboard by using the admin ID & password. Admin provided the authority to add, edit, or update & delete the data. The application provides interface communication to the user and admin. One more thinks Admin can sell their own products by using the application. That purpose providing a marketing interface for admin and user.



**Figure 1: System Diagram.**

### 1. Software Requirement Specification:

a) Evaluate the importance of a well-defined Software Requirement Specification (SRS) in the software development lifecycle and its impact on project success.

Ans: A well-defined Software Requirement Specification (SRS) is a foundational document in the software development lifecycle that outlines the detailed requirements, functionalities, and constraints of a software project. Its importance cannot be overstated, as it serves as a critical reference point for all stakeholders involved in the development process. Here's an evaluation of the importance of a well-defined SRS and its impact on project success:

1. **Clarity and Shared Understanding:** A well-defined SRS ensures that all stakeholders, including developers, testers, designers, and clients, have a common understanding of the project's objectives, functionalities, and scope. This reduces misunderstandings and minimizes the risk of misaligned expectations.
2. **Reduced Ambiguity:** Clear and detailed requirements in the SRS leave little room for interpretation. This clarity helps prevent scope creep, where new requirements are added without proper assessment, leading to project delays and budget overruns.
3. **Basis for Estimations:** Developers use the SRS to estimate the effort, resources, and time required for each phase of the project. Accurate estimations enable

better project planning and resource allocation.

4. Risk Management: A comprehensive SRS allows project managers to identify potential risks and challenges early in the project lifecycle. This enables them to develop mitigation strategies and contingency plans to address these risks.

5. Foundation for Design and Development: The SRS serves as a blueprint for the design and development phases. It guides the creation of architecture, user interfaces, data models, and other technical components.

6. Quality Assurance and Testing: Testers use the SRS to create test plans and test cases, ensuring that all required functionalities are thoroughly tested. This leads to higher software quality and reliability.

7. Change Control: When changes to requirements are requested, the SRS serves as a baseline against which these changes can be evaluated. This helps in assessing the impact of changes on the project timeline and resources.

8. Client Communication: The SRS serves as a document of agreement between the client and the development team. It provides a clear reference for clients to review and validate whether the delivered software aligns with their expectations.

9. Project Monitoring and Reporting: A well-defined SRS facilitates effective project monitoring by providing a measurable set of requirements. Progress can be tracked against these requirements, ensuring that the project stays on track.

10. Alignment with Business Goals: A thorough SRS ensures that the software being developed aligns with the business goals and objectives of the client or organization. This connection enhances the value delivered by the software.

11. Project Documentation: The SRS acts as a comprehensive source of documentation for the project. It provides a historical record of project decisions, requirements, and functionalities, aiding future maintenance and

enhancements.

Impact on Project Success: A well-defined SRS significantly contributes to project success by minimizing risks, preventing scope creep, enhancing communication, and ensuring that the final product meets stakeholder expectations. When the SRS is properly crafted, it improves the chances of delivering the software on time, within budget, and with the required quality, ultimately leading to higher client satisfaction and positive project outcomes.

b) Analyse a given SRS document to identify any ambiguities or inconsistencies and propose improvements to enhance its clarity and completeness.

Ans: Let's analyze this portion of the farmer helper site for ambiguities, inconsistencies, and areas for improvement:

Objective Clarity:

- Issue: The objective is mentioned but lacks clear and concise language.
- Improvement: Reframe the objective to clearly state that the project aims to develop a website that connects village farmers with city wholesalers, facilitating online sales and providing training if needed.

Existing System Description:

- Issue: The description is brief and could be more comprehensive.
- Improvement: Provide more details about the challenges faced by farmers in the existing system and how the manual selling process is cumbersome. Explain how the reliance on seasonal markets impacts the farmers' income.

Proposed System Description:

- Issue: The description is somewhat repetitive and could be more explicit about the website's benefits.
- Improvement: Clearly outline how the proposed system (website) will address the challenges mentioned in the existing system. Highlight features like online sales, training for farmers, and convenience for both farmers and wholesalers.

Clarity of Process:

- Issue: The process of how farmers will use the website is not detailed.
- Improvement: Describe step-by-step how a farmer would use the website to list products, manage orders, and interact with wholesalers. Mention the registration process, product listing, and communication features.

Technical Details and Implementation:

- Issue: The document lacks information about the technical aspects of the website's development.
- Improvement: Add a section that briefly explains the technologies,



programming languages, and frameworks that will be used to build the website. This will provide a better understanding of the project's technical foundation.

#### User Roles and Interfaces:

- Issue: The document briefly mentions farmers and wholesalers but doesn't detail their interactions with the website.
- Improvement: Provide separate sections that define the roles of farmers, wholesalers, and potentially computer professionals. Explain how each role will interact with the website, including registration, product listing, purchasing, and support.

#### Benefits and Impact:

- Issue: The document could benefit from a clearer explanation of the benefits of the proposed system.
- Improvement: Elaborate on how the website will improve farmers' income, reduce travel time, increase access to markets, and make the purchasing process more efficient for wholesalers.

#### Training Plan:

- Issue: While the document mentions training, it lacks a comprehensive plan.
- Improvement: Dedicate a section to the training plan, including details about scheduling, content, and delivery methods. Clarify how the training will help farmers become proficient in using the website.

Remember, a well-defined SRS should be detailed, comprehensive, and clear in its description of the project's objectives, functionalities, and processes. It should leave little room for ambiguity or interpretation to ensure that all stakeholders share a common understanding of the project's scope and goals.

c) Compare and contrast different techniques for requirement elicitation, such as interviews, surveys, and use case modeling, and determine their effectiveness in gathering user needs?

Ans: Requirement elicitation is the process of gathering and understanding user needs and requirements for a software project. Different techniques, such as interviews, surveys, and use case modeling, can be employed for this purpose. Let's compare and contrast these techniques and determine their effectiveness in gathering user needs:

#### 1. Interviews:

- Description: Interviews involve direct conversations between stakeholders and analysts. They can be structured (following a predefined set of questions) or unstructured (allowing for open-ended discussions).
- Advantages:
  - Rich and detailed information can be obtained.
  - Immediate clarification of responses is possible.
  - Personal interactions can establish rapport and trust.

- Disadvantages:
  - Time-consuming, especially for large user groups.
  - Potential for bias due to the interviewer's influence.
  - May not be representative of all user opinions.

## 2. Surveys:

- Description: Surveys involve distributing questionnaires to a larger number of stakeholders, allowing them to respond at their convenience.
- Advantages:
  - Efficient for collecting data from a large user base.
  - Anonymity may encourage honest responses.
  - Standardized questions facilitate quantitative analysis.
- Disadvantages:
  - Limited depth in responses due to fixed-choice questions.
  - Lack of context for certain responses.
  - Lower response rates and potential for incomplete or biased data.

## 3. Use Case Modeling:

- Description: Use case modeling involves creating scenarios that describe how users interact with the system. Use cases depict user actions and system responses.
- Advantages:
  - Visual representation provides a clear understanding of interactions.
  - Helps in identifying system functionalities from a user's perspective.
  - Easier to prioritize requirements based on critical use cases.
- Disadvantages:
  - Requires a good understanding of system behavior to create accurate use cases.
  - May not capture all possible scenarios.
  - Focuses more on system behavior rather than user needs in certain cases.

## Effectiveness in Gathering User Needs:

- Interviews: Interviews are highly effective in gathering in-depth insights and clarifications. They are particularly useful for capturing nuanced requirements and building rapport with users. However, they can be time-intensive and might not be feasible for large user groups.
- Surveys: Surveys are efficient for gathering a broad range of opinions quickly. They are effective for identifying common preferences or trends among users. However, they might lack the depth of understanding provided by interviews and might not capture context or specific scenarios.
- Use Case Modeling: Use case modeling is effective in capturing user interactions and system behavior. It helps in understanding how users will interact with the system and identifying key functionalities. However, it may not capture the full spectrum of user needs and may focus more on system behavior than user goals.

In practice, a combination of these techniques might be the most effective approach. Interviews can provide deeper insights, surveys can help validate and quantify findings, and use case modeling can help visualize interactions. The choice of technique depends on factors such as the project's scope, time constraints, user base, and available resources.