

Software Requirements Specification

Candijay Fishery and Agricultural household Registration Management and Decision Support System

Signature
Developer
Review Date

Alma Mae Auxtero

06/13/2022

DOCUMENT CHANGE RECORD

Revision No.	Description of Change	Approved By	Date Approved
01	Initial publication	Ria H. Borromeo, Ph.D.	

Table of Contents

1. Introduction	4
1.1 Purpose.....	4
1.2 Document Conventions.....	4
1.3 Intended Audience and Reading Suggestions.....	4
1.4 Project Scope.....	5
1.5 References.....	5
2. Use Cases.....	5
2.1 Product Perspective.....	5
2.2 Product Features.....	5
2.3 User Classes and Characteristics.....	6
2.4 Design and Implementation Constraints.....	6
2.5 Assumptions and Dependencies.....	6
3. Functional Requirements.....	6
4. Non-Functional Requirements.....	6
4.1 Usability requirements.....	7
4.2 Reliability and up-time requirements.....	7
4.3 Security requirements.....	7
4.4 Performance and scalability requirements.....	8
4.5 Maintainability and upgradability requirements.....	8
4.6 Supportability and operability requirements.....	8
5. Environmental requirements.....	8
5.1 System hardware requirements.....	8
5.2 System software requirements.....	8
5.3 Application program interfaces (APIs) requirements.....	9
5.4 Data import and export requirements.....	9

1. Introduction

1.1 Purpose

Software Requirements Specification or SRS is a document that is purposely made to give a comprehensive definition or description of the functional requirements of a specific application to be developed. This document is a helpful guideline for designers and developers since this becomes their monitoring tool all throughout the development cycle. Thus, this SRS describes the specifications of *Candijay Fishery and Agricultural Household Registration Management and Decision Support System (C-FARMS)*, a decision support system for managing the Municipal Agriculture Office process of farmer registration and program beneficiaries selection.

1.2 Document Conventions

Main Headings:

Font: Calibri Face: Bold Size: 14

Sub-sections:

Font: Calibri Face: Bold Size: 12

Other Text Explanations:

Font: Calibri Face: Normal Size Size: 12

1.3 Intended Audience and Reading Suggestions

This document is prepared for casual users, developers, and all other project stakeholders.

- Users: The casual users who wish to read this document can obtain a clear view of what the project can do.
- Developers: The developers can easily review through this document the methodologies to be used for the target improvements on the project.

Each section discusses different important matters. And the suggested way of reading this document is to read it sequentially from the first section to the last.

Section 2 provides an overall description of the product- *Candijay Fishery and*

Agricultural Household Registration Management and Decision Support System.

Section 3 describes the overview of the use cases of the system. Section 4 discusses the non-functional requirements of the system. And section 5 discusses the environmental requirements of the system.

1.4 Project Scope

This document will be limited to the set of software requirements involved to develop the web application- Candijay Fishery and Agricultural Household Registration Management and Decision Support System. This encapsulates detailed definitions of all the functionalities and features that the site will provide. This will include every single detail of the project requirements except the programming language or the development tool to be used in the development and creation of the project.

1.5 References

Software Requirement Specification

Thomas Ngyuyen Hornbeck

[Readysset Format by Tigris](#)

2. Use Cases

The detailed information regarding the use cases of the system is written in a separate [use case suite](#) document. To provide an overview regarding the use cases in the system, the following sections describe the (1) product perspective, (2) product features, (3) user classes and characteristics, (4) design and implementation constraints, and (5) assumptions and dependencies.

2.1 Product Perspective

Candijay Fishery and Agricultural Household Registration Management and Decision Support System is a web application that is made to help the Municipal Agriculture Office (MAO) in their record keeping of farm and farmer records. In effect, farmer households will be effectively served by the MAO through its programs and beneficiaries will be fairly chosen.

2.2 Product Features

Candijay Fishery and Agricultural Household Registration Management and Decision Support System will allow three types of users- administrators, employers and interns. These users should sign up or register into the site to access information and add data to the site. The website provides features on posting internship vacancies and applicants' profile setting, as well as, notifications where applicable.

2.3 User Classes and Characteristics

There are three specific user classes; the administrator/s, manager and encoders.

- Administrator/s: This user class would be the one to add user accounts and has the ability to access all functionality of the system.
- Managers: This user class may add, edit and delete data into the system. However, user accounts may only be viewed by this user class.
- Encoders: May only add and download data from the system.

2.4 Design and Implementation Constraints

This web application enforces user authentication for security purposes. At present, this does not allow offline reporting. However, constant updates and additional features will be made by the developer.

2.5 Assumptions and Dependencies

It is required that users should be logged in to the site to read, add and update data in the system.

3. Functional Requirements

Detailed information regarding the set of functional requirements that will be implemented in the system is written in a separate use case document. The use cases documented are organized by functional area, priority, and user class.

4. Non-Functional Requirements

4.1 Usability requirements

The difficulty of performing high-frequency use cases determines the usability of the system. It depends on the number of clicks, the knowledge that the user must have at each step, the decisions that the user must make at each step, and the mechanics of each step (e.g. choosing from options instead of typing the desired value directly).

Another criterion for usability is learnability, which indicates how easy it is for users to perform specific tasks the first time they use an application. In order to achieve this, user interface should be as familiar as possible to users who have used similar web applications.

4.2 Reliability and up-time requirements

The system must be very reliable due to the importance of data and the damages incorrect or incomplete data can do.

- The system is available 100% for the user and is used 24 hrs a day and 365 days a year.
- The system shall be operational 24 hours a day and 7 days a week.
- The system will be developed in such a way that it may fail once in a year.
- Even if the system fails, the system will be recovered back up within an hour or less.
- The accuracy of the system is limited by the accuracy of the speed at which the different users use the system.
- The system shall provide 100% access reliability.

4.3 Security requirements

The system must have security features so as to protect itself and hardware on which it is going to be operated.

- Access will be controlled with google accounts.
- Access to functions will be restricted or authorized based on user type.
- Data access should be restricted or authorized based on user type.

4.4 Performance and scalability requirements

The system must be designed in a way such that response time is minimal and reacts quickly to its environment.

- The landing page or splash page should be able to load within a few seconds of key press.
- The information is refreshed real-time.
- The system shall respond to a user immediately.
- The system is cable of handling 100 users at a time.
- The resources are modified according to the user requirements.

4.5 Maintainability and upgradability requirements

The system should be easy to extend. This implies that the code should be written in a way that it favors the implementation of new functions. The system must have minimal downtime or disruption whenever new versions of the product are deployed. Data file formats that include enough meta-data should be used in order to reliably transform existing customer data during an upgrade.

4.6 Supportability and operability requirements

The system is a web-based application and is built in HTML, Javascript, CSS, NodeJS, ReactJS, and PostgreSQL, therefore, making it platform and OS-independent. The system is easily supported and operable.

5. Environmental Requirements

5.1 System hardware requirements

The system needs wired and wireless network infrastructure. MAC, Unix and Windows and mobile phones can be used to access the site.

5.2 System software requirements

The client operating system can be opened in any type of operating system, as long as it has a browser. Moreover, the system needs to be opened in a JavaScript-compatible browser. In terms of network software and protocols for systems communication, TCP/IP, HTTP, and HTTPS are needed.

5.3 Application program interfaces

All pages of the system are following a consistent theme and clear structure. The occurrence of errors should be minimized through the use of checkboxes, radio buttons, and scroll-down select in order to reduce the amount of text input from user. Use of Javascript for data validation before submission. Use of data tables to display information to give a clear structure that is easy to understand. The error message should be located beside the error input which clearly highlights and tells the user how to solve it. If system error, it should provide the contact methods. Each user type will have its own interface and functions.

5.4 Data import and export requirements

System must provision for data import and export.

- The system will utilize a GraphQL query language, which is designed to make APIs fast, flexible, and developer-friendly.
- The system will store data in a PostgreSQL database.