Republic of the Philippines
Laguna State Polytechnic University
Province of Laguna

Software Requirements Specification (SRS)

Prepared by:

Adelan, Jhanna Shammel
Almario, Cel Rick D.
Bayani, Reigniell Ann L.
Delos Santos, Gladys A.
Javier, Geron Simon A.
Redera, Angelo Nicolas A.
BSCS 3B IS

Submitted to:

Mia V. Villarica, DIT
Associate Dean, CCS, Subject Instructor

1. Introduction and Purpose

The purpose of this Software Requirements Specification (SRS) is to define the requirements for the development and deployment of the "Web-Based Document Tracking System and Management for National Irrigation Administration Regional Office IV-A (CALABARZON) of Pila, Laguna." This document outlines the intended system's purpose, scope, overall description, and specific requirements to ensure a smooth and efficient implementation.

The main objective of the system is to improve the existing manual process of document handling and tracking within the NIA Pila, Laguna office. The current system lacks an automated way to monitor the status, routing, and handling of documents, resulting in delays, confusion, and inefficient workflow. The proposed web-based system provides a platform where documents can be tracked in real-time, securely managed, and easily accessed by relevant users, including administrators, record officers, document handlers, and guests.

2. Overall Description

The Web-Based Document Tracking System is designed to provide a centralized, accessible, and efficient solution for managing and tracking documents within the National Irrigation Administration (NIA) Regional Office IV-A, located in Pila, Laguna. It addresses the limitations of the current manual process by automating the tracking, routing, and status monitoring of documents.

The system allows four types of users:

- **System Admin** – manages offices, users, documents, and generates reports.
- **Administrative Record Office** – creates and tracks documents, receives and processes document transfers.
- **Document Handler** – submits, receives, and transfers documents across offices.
- **Guest** – submits documents, tracks status, and communicates with the admin office.

The system is accessible via a local network through a browser using the URL: http://localhost/documenttracking/index.php.

It provides core features such as:

- Real-time document tracking and status updates
- QR code scanning for document identification

- Centralized communication between guests and admin
- Document submission, transfer, and completion features
- Report generation in PDF and CSV formats

The system uses the **Rapid Application Development (RAD)** model for faster delivery and improved feedback integration. Technologies used include **PHP, MySQL, JavaScript, and Tailwind CSS** for a responsive and modern web interface.

3. Functional and Non-functional Requirements

3.1 Functional Requirements

The system must provide the following functions for each user role:

System Admin:

- View, add, edit, and search document and office records
- Manage user accounts (add, update, archive, retrieve)
- Generate monthly reports (PDF and CSV)
- Monitor office performance automatically
- Scan document QR codes for quick access

Administrative Record Office:

- Create document tracking entries
- Receive and complete document status updates
- Submit and transfer documents to other offices
- Track incoming and ongoing documents
- Use communication tools to coordinate with users
- Generate transaction reports

Document Handler:

- Submit and manage document tracking forms
- Receive and transfer documents
- Mark documents as complete or ongoing
- View document status and details
- Scan document QR codes for processing

Guest:

- Submit documents with required details
- View document status (pending, pulled, or completed)
- Use basic communication features with admin office

3.2 Non-functional Requirements

- **Usability:** The interface must be user-friendly and easy to navigate for all user levels.
- **Security:** System access requires login credentials with OTP (One-Time Password) for user verification.
- **Performance:** The system must support real-time updates with minimal response delay for document status changes.
- **Scalability:** The system should allow future upgrades and additional features without major system changes.
- **Maintainability:** The system must be easy to update, debug, and maintain by IT staff.
- **Reliability:** Data must be accurately stored in the database with backup and recovery measures in place.
- **Responsiveness:** While the current system works best on web browsers, future versions must fully support mobile responsiveness.

4. System Features and Interfaces

4.1 System Features

1. Real-Time Document Tracking

– Allows users to monitor document status (pending, received, completed, or released) in real time.

2. Role-Based User Access

– Four user roles: System Admin, Administrative Record Office, Document Handler, and Guest.

– Each role has specific access and privileges based on their functions.

3. Document Submission and Routing

– Users can submit documents, track their routing across offices, and receive updates on status.

4. QR Code Integration

– Each document is assigned a unique QR code.

– Users can scan QR codes to instantly view and update document details.

5. Communication System

– Internal messaging feature between guests and the record section for better coordination.

6. Report Generation

– System Admin and Record Office users can generate monthly document reports.

– Reports can be exported as PDF or CSV files.

7. User and Office Management

– Admin can create, edit, archive, and retrieve user accounts and office details.

– Includes a searchable user and office list.

8. Performance Tracking

– Automatic performance monitoring for each division office based on document activities.

4.2 User Interfaces

- **Login Page:** Secure login with username, password, and OTP.
- **Admin Dashboard:** View stats, manage documents, users, and generate reports.
- **Record Office Dashboard:** Create tracking, manage incoming documents, and handle communication.
- **Document Handler Dashboard:** Submit, transfer, and mark documents.
- **Guest Interface:** Submit documents and communicate with the admin.

5. Assumptions and Constraints

5.1 Assumptions

- Users (Admin, Document Handler, Record Office, and Guest) are trained to use basic web applications.
- The system will be hosted and accessed through a **local server (localhost)** or a **secured intranet** within NIA Pila, Laguna.
- Each user will have access to their assigned account credentials and email for OTP verification.
- Documents submitted to the system are encoded and tagged correctly with the required information and document type.
- Offices involved in document processing have stable internal communication and assigned staff for handling submissions.

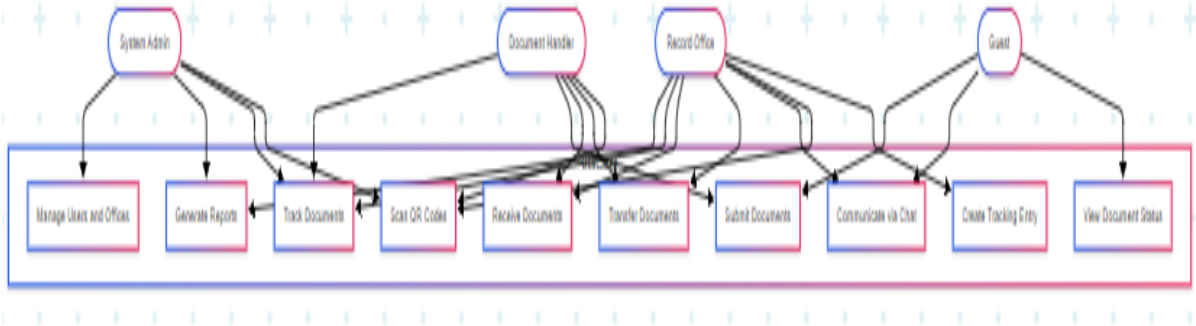
5.2 Constraints

- The system is **not optimized for mobile use** at this stage, though mobile responsiveness is recommended for future versions.
- The platform requires a working local server setup (e.g., XAMPP) to run PHP

and MySQL databases.

- Access is limited to **authorized users only**, with no support for anonymous or public access.
- All files and user data must be stored within the **local database**, with manual or scheduled backups required.
- Integration with external email APIs is limited; email notifications are dependent on the server's mail configuration.
- System performance may vary depending on the **local network's speed and server capacity**.

6. Use Case Diagram



7. Testing Tool Documentation

To ensure that the Web-Based Document Tracking System functions correctly and meets its requirements, the development team used **PHPUnit** as the main testing tool.

7.1 Tool Used: PHPUnit

PHPUnit is a unit testing framework for PHP, which helps in testing individual components or functions of the system to verify they work as expected. It is widely used in PHP-based projects for its reliability, simplicity, and compatibility with modern development practices.

7.2 Purpose and Benefits

- **Accurate Testing of Logic:** PHPUnit was used to validate functions such as user login, document submission, and tracking logic, ensuring each feature works correctly.
- **Error Detection:** It helped identify issues during early development, allowing

the team to fix bugs before deployment.

- **Automated Testing:** PHPUnit allowed repetitive tests to be automated, saving time and reducing manual effort.
- **Quality Assurance:** The consistent use of PHPUnit helped maintain the integrity and reliability of the core system features.

7.3 Testing Approach

The team applied **black box testing** using PHPUnit on core functions. Ten IT experts were involved in testing the system features, and all test cases received a **100% pass rate**, confirming the system met its functional requirements successfully.