

# **Technical Documentation: Web-Based Document Tracking System and Management for National Irrigation Administration Regional Office IV-A (Calabarzon) ff Pila, Laguna**

---

## **System Overview and Architecture**

The Document Tracking System is built on a PHP-based web application architecture following the Model-View-Controller pattern:

- Frontend Layer : HTML/CSS/JavaScript with jQuery for enhanced interactivity
- Controller Layer : PHP controllers handling business logic and request processing
- Model Layer : PHP models for database interactions and data manipulation
- Database Layer : MySQL database storing all system data

## **Key Architectural Components:**

- Authentication System :
  - Role-based access control (Admin, Record Office, Handler, Guest)
  - Session-based authentication with security measures
  - Password recovery with OTP verification
- Document Management :
  - Document upload and storage
  - QR code generation for tracking
  - Document routing between offices and handlers
- Notification System :
  - Real-time notifications for document actions
  - Email notifications via PHPMailer
  - Idle document monitoring via cron jobs
- Reporting Module :
  - Document status reports
  - User activity tracking
  - Performance metrics

## Database Schema:

The system utilizes multiple interconnected tables including:

- User information and authentication
- Document metadata and tracking
- Office and handler relationships
- Notifications and messaging
- Document types and categories

## Summary of Enhancements and Rationale

### Performance Optimizations

- Optimized database queries
  - Created reusable query methods in **OptimizedQueries.php**
  - Used prepared statements and indexed fields for faster execution
  - Reduced redundant SQL calls by consolidating logic
- Database maintenance scripts
  - Added **cleanup.php** to archive old records and optimize tables
  - Implemented **check-idle-documents.php** to detect unprocessed documents
  - Scheduled cleanup using **monitor.php** via cron jobs
- Efficient document retrieval mechanisms
  - Introduced caching and index-based lookups
  - Improved query performance with **ORDER BY**, **LIMIT**, and **WHERE** clauses that use indexed columns

### Security Enhancements

- Secure authentication system
  - Uses strong password hashing with **password\_hash()** and **password\_verify()**
  - Implements session-based login with secure handling of user roles
- Role-based access control
  - Each user role has restricted access based on permissions

- Prevents unauthorized actions and data exposure
- Input validation and sanitization
  - All inputs are validated before database interaction
  - Reduces risk of injection attacks
- Session security measures
  - Session regeneration on login
  - Secure session destruction on logout
  - Role/office verification before page access

### **Workflow Improvements**

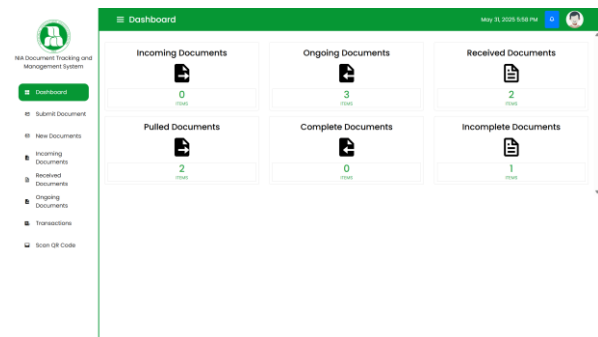
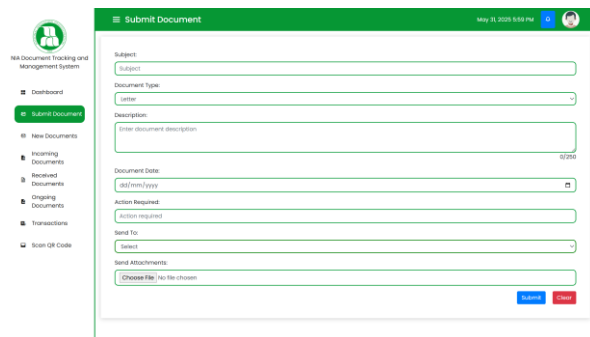
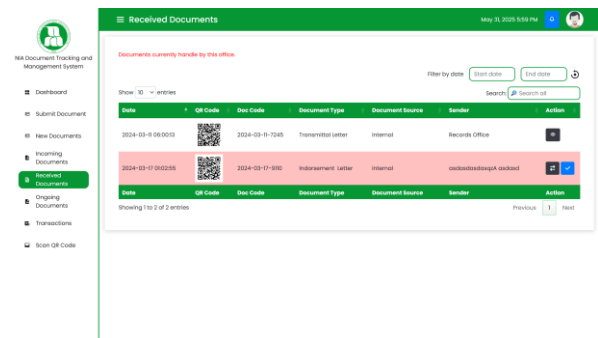
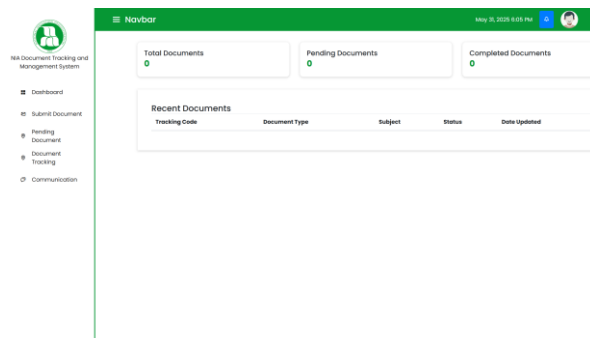
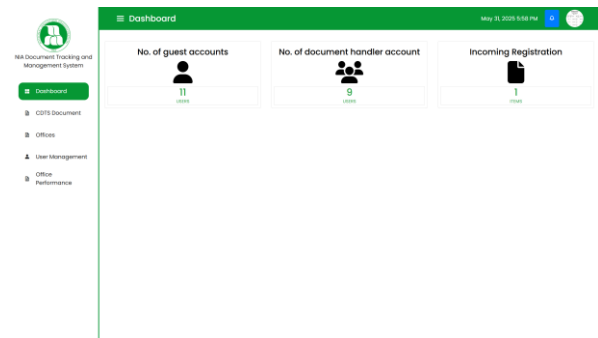
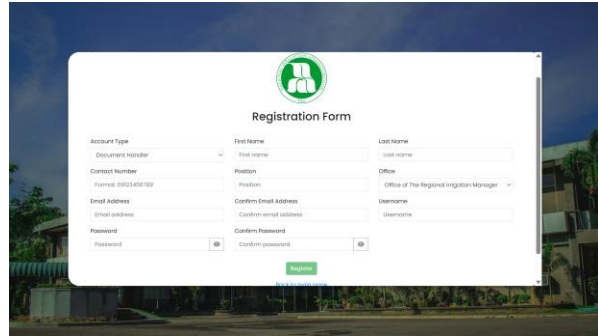
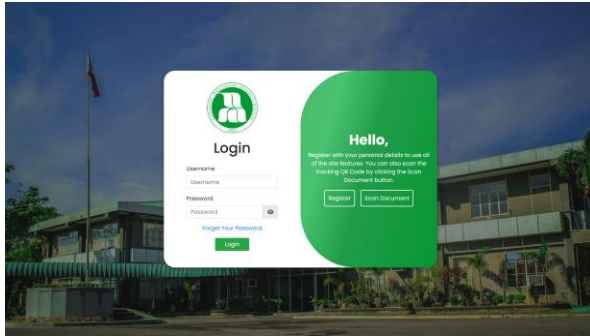
- QR code scanning for rapid tracking
  - Allows handlers and guests to instantly track document status
  - Reduces manual input and increases accuracy
- Automated idle document detection and notification
  - Identifies documents not processed within 7 days
  - Sends alerts to responsible office
- Streamlined document transfer process
  - Enables handlers to pass documents between offices easily
  - Logs every action taken for audit purposes
- Conversation system for document-related communication
  - Users can send messages related to specific documents
  - Messages are stored and linked to document ID

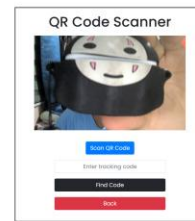
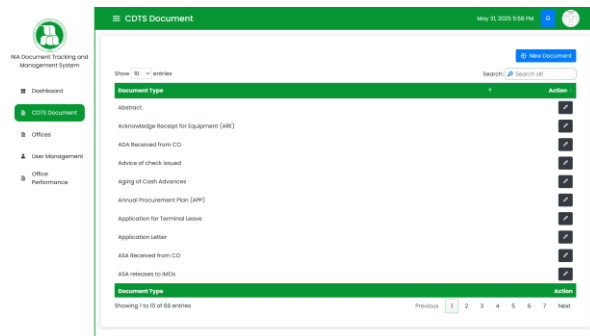
### **User Experience (UX)**

- Intuitive, role-specific dashboards
  - Tailored views for Admin, Handler, Record Office, Guest, and Sysadmin
  - Displays only relevant information per user type
- Real-time notifications
  - In-app alerts when documents are received or transferred
  - Email notifications for key events using PHPMailer

- Simplified document tracking for guests
  - Dedicated dashboard for external users
  - QR code scanner access for quick lookup

## Updated UI/UX Screenshots





## Testing Approach and Results

### Testing Methodology:

- Unit Testing : PHPUnit tests for individual components
- Integration Testing : Testing interactions between system components
- User Acceptance Testing : Role-based testing scenarios
- Performance Testing : Database query optimization and load testing

### Test Coverage:

- Model functionality tests (BaseModelTest, DocumentTest, etc.)
- Configuration validation (ConfigTest)
- Database connectivity and operations (DatabaseTest)
- Business logic validation (ActionTakenTest, DocumentTrackingTest)
- Redirect logic testing (RedirectLogicTest)

### Continuous Integration:

- GitHub Actions workflow for automated testing (.github/workflows/php-test.yml)
- Test-driven development approach

## Technologies and Frameworks Used

### Backend:

- PHP : Core programming language
- MySQL : Database management system
- PHPUnit : Testing framework

### Libraries:

- PHPMailer : Email functionality

- phpqrcode : QR code generation
- TCPDF : PDF generation for reports
- Composer : Dependency management

**Frontend:**

- HTML/CSS/JavaScript : Core frontend technologies
- jQuery : JavaScript library for DOM manipulation
- DataTables : Enhanced table displays
- SweetAlert : User-friendly alerts and notifications
- Boxicons : Icon library

**Development Environment:**

- Laragon : Local development environment
- Git : Version control
- GitHub : Repository hosting and CI/CD

**Developer Notes / Installation Instructions****System Requirements:**

- PHP 7.4 or higher
- MySQL 5.7 or higher
- Apache/Nginx web server
- Composer for dependency management

**Installation Steps:**

- Clone the Repository :
- Install Dependencies :
- Database Setup :
  - Create a MySQL database
  - Import the document-tracking-db.sql file
  - Configure database connection in config/database.php
- Directory Permissions : Ensure write permissions for:
  - assets/qr-codes/

- assets/uploaded-pdf/
  - assets/user-profile/
- Cron Job Setup : Configure a daily cron job to run:
- Web Server Configuration :
  - Configure your web server to point to the project directory
  - Ensure URL rewriting is enabled if needed

### **Common Issues and Troubleshooting:**

- QR Code Generation : Ensure GD library is enabled in PHP
- Email Notifications : Configure SMTP settings correctly in PHPMailer
- File Uploads : Check directory permissions and PHP upload limits
- Database Connectivity : Verify database credentials and server access

### **Development Guidelines:**

- Use prepared statements for database queries
- Implement proper error handling and logging
- Write unit tests for new functionality
- Document code changes and new features