Software Requirements Specification

for

Contract Registry System

**Version 1.0 approved**

**Prepared by Junior R. Gaseb**

**Namibia Wildlife Resorts**

**22 September 2025**

**Table of Contents**

[1. Introduction 1](#_Toc209599479)

[1.1 Purpose 1](#_Toc209599480)

[1.2 Document Conventions 1](#_Toc209599481)

[1.3 Intended Audience and Reading Suggestions 1](#_Toc209599482)

[1.4 Product Scope 1](#_Toc209599483)

[1.5 References 1](#_Toc209599484)

[2. Overall Description 2](#_Toc209599485)

[2.1 Product Perspective 2](#_Toc209599486)

[2.2 Product Functions 2](#_Toc209599487)

[2.3 User Classes and Characteristics 2](#_Toc209599488)

[2.4 Operating Environment 2](#_Toc209599489)

[2.5 Design and Implementation Constraints 2](#_Toc209599490)

[2.6 User Documentation 2](#_Toc209599491)

[2.7 Assumptions and Dependencies 3](#_Toc209599492)

[3. External Interface Requirements 3](#_Toc209599493)

[3.1 User Interfaces 3](#_Toc209599494)

[3.2 Hardware Interfaces 6](#_Toc209599495)

[3.3 Software Interfaces 7](#_Toc209599496)

[3.4 Communications Interfaces 7](#_Toc209599497)

[4. System Features 7](#_Toc209599498)

[4.1 Authentication and Authorization 7](#_Toc209599499)

[4.2 Contract Management Interface 8](#_Toc209599500)

[4.3 File Upload and Access 8](#_Toc209599501)

[4.4 Automated Email Remainders 9](#_Toc209599502)

[4.5 Logging Automatic Emails 9](#_Toc209599503)

[4.6 Backend API 10](#_Toc209599504)

[5. Other Nonfunctional Requirements 10](#_Toc209599505)

[5.1 Performance Requirements 10](#_Toc209599506)

[5.2 Safety Requirements 10](#_Toc209599507)

[5.3 Security Requirements 11](#_Toc209599508)

[5.4 Software Quality Attributes 11](#_Toc209599509)

[5.5 Business Rules 11](#_Toc209599510)

[6. Other Requirements 11](#_Toc209599511)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Junior R. Gaseb | 24/09/2025 | Initial document | 1.0 |
|  |  |  |  |

# Introduction

## Purpose

To track and maintain records of contracts between Namibia Wildlife Resorts (NWR) with external companies. It streamlines the process of contract creation, review, renewal, and expiration tracking. It ensures that contracts are backed up, securely stored, and accessible to authorized personnel.

## Document Conventions

Functional Requirements are labeled FRx.y, where x and y are numbers.

Nonfunctional requirements are labeled NFRx.y, where x and y are numbers.

Headings: Heading2, Times font, bold, size-14

Body Text: Normal text, Arial font, size-11

## Intended Audience and Reading Suggestions

Intended for internal use within the organization, for future reference and traceability.

It will be used by:

* Developers: to design, test, and implement the system.
* IT staff
* End Users (Admin, Managers, Viewers): for awareness if system functions.
* Auditors

## Product Scope

The system provides the following features:

* A web interface for contract CRUD operations (Create, Read, Update, and Delete).
* Role-based access control (admin, manager, viewer (default user with limited permission).
* File uploads for contract documents (specifically PDF, DOC, and DOCX).
* Automated email reminders for contract expirations (90, 60, and 30 days remaining).
* Emails send via SMTP
* Logging of email activities to `backend/logs/reminder\_log.txt`.
* Integration with a MySQL database.
* Deployed in a Windows server/Xampp environment.

## References

* *PHPMailer Documentation:* [*https://github.com/PHPMailer/PHPMailer*](https://github.com/PHPMailer/PHPMailer)
* *XAMPP Setup:* [*https://www.apachefriends.org/*](https://www.apachefriends.org/)
* *Windows Task Scheduler:* [*https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-task-scheduler*](https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-task-scheduler)

# Overall Description

## Product Perspective

The Contract Registry System is a standalone web-based application built using PHP (backend), JavaScript/CSS/HTML (frontend), and MySQL (database). It integrates with PHPMailer for email notifications and uses Windows Task Scheduler for automated script execution.

## Product Functions

* User authentication and authorization.
* CRUD operations on contracts.
* File management (upload, view, download).
* Contract type management.
* Automated contract expiry reminders.
* Logging of email activities.
* API endpoints for contract operations.

## User Classes and Characteristics

Admin: Full access, manage users, roles, contracts.

Manager: Manage contracts (create, edit, delete).

Viewer: Limited to viewing contracts and documents.

## Operating Environment

OS: Windows Sever / Windows 10+

Webserver: Apache (XAMPP)

Database: MySQL

Browser Support: Chrome, Firefox, Edge (Latest version)

Email: SMTP

## Design and Implementation Constraints

Only authenticated users may access contracts.

File uploads restricted to PDF, DOC, DOCX.

Emails must be sent via Gmail SMTP with TLS.

Session-based authentication required.

## User Documentation

*TBD*

## Assumptions and Dependencies

2.7.1 Assumptions

Assumption that the system will be limited to qualified personal.

Users are assumed to know basic web navigation.

That the contracts will be handled by the person that creates them on the system.

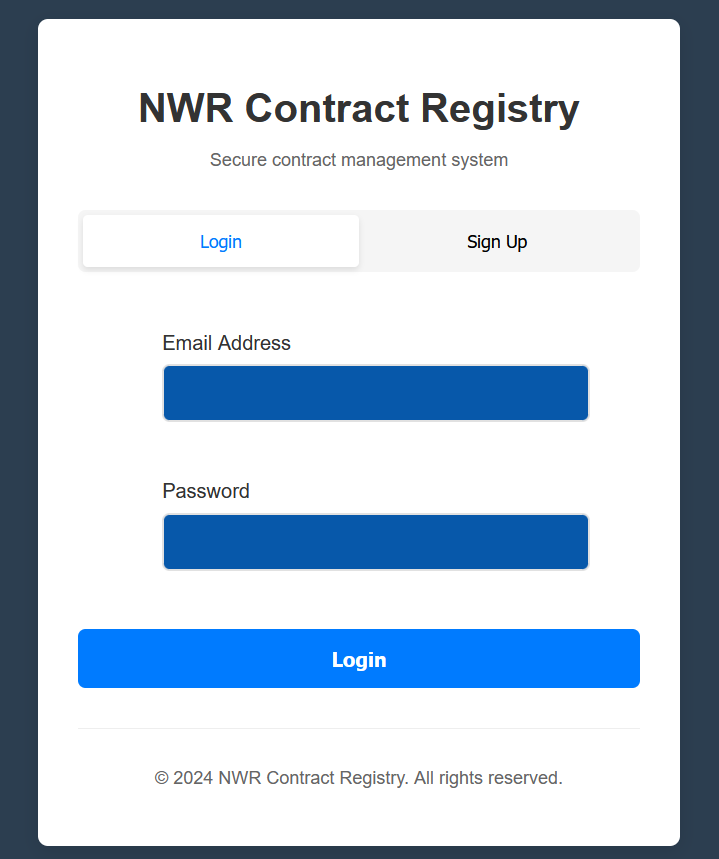
2.7.2 Dependencies

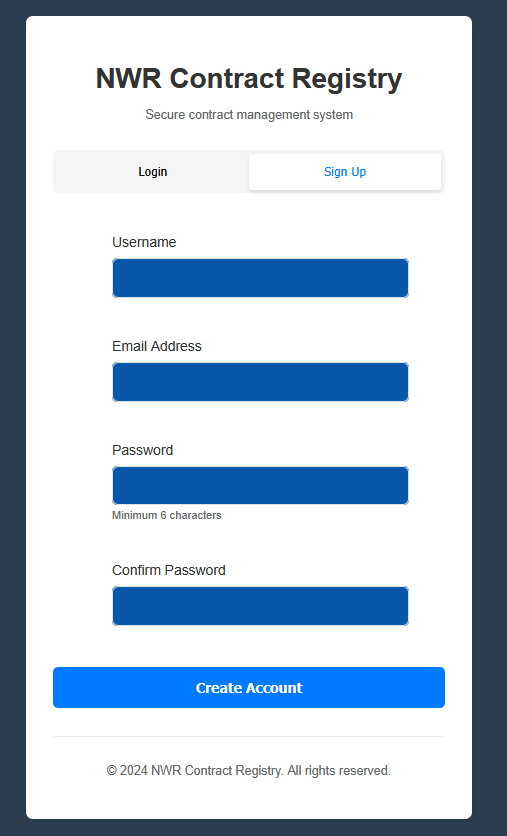
The system depends on PHPMailer, MySQL, and Task Scheduler.

# External Interface Requirements

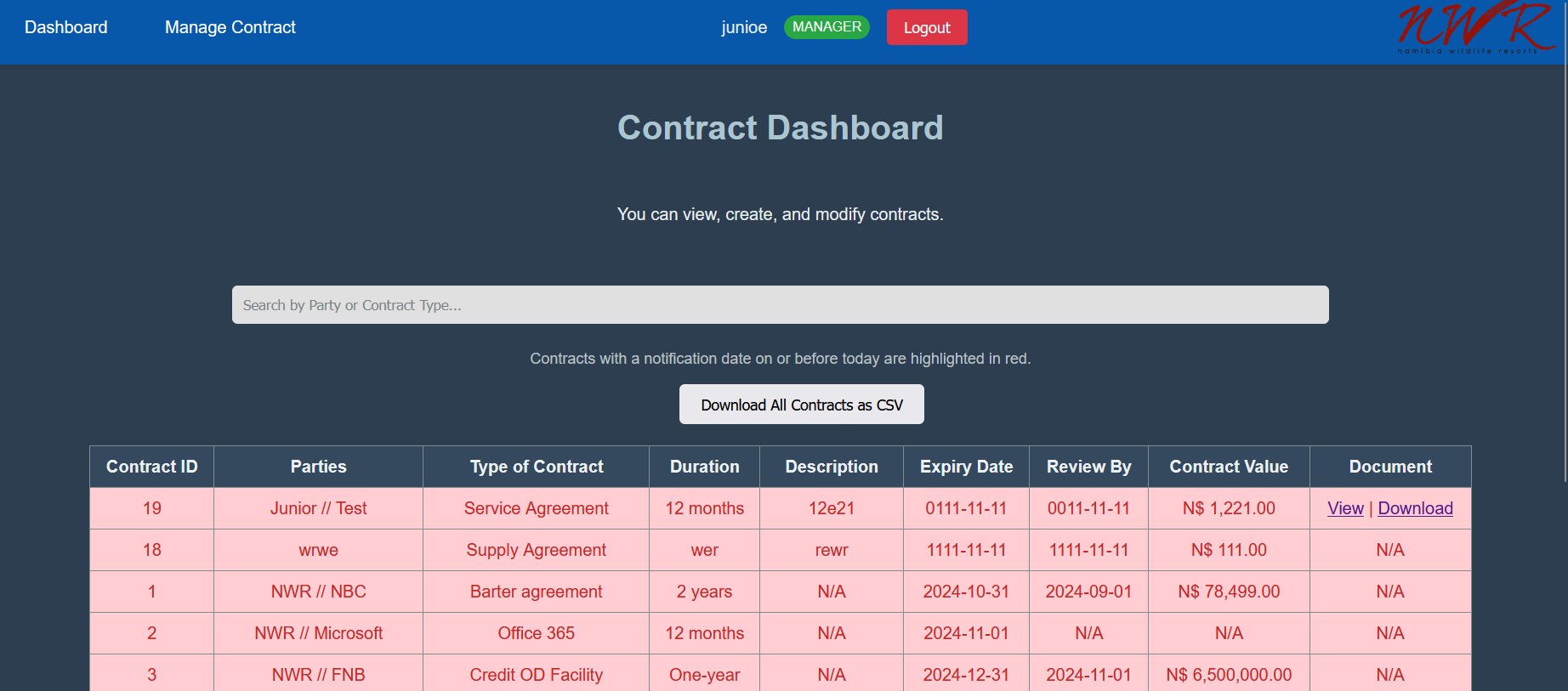
## User Interfaces

Login Page: Authentication (email, password).



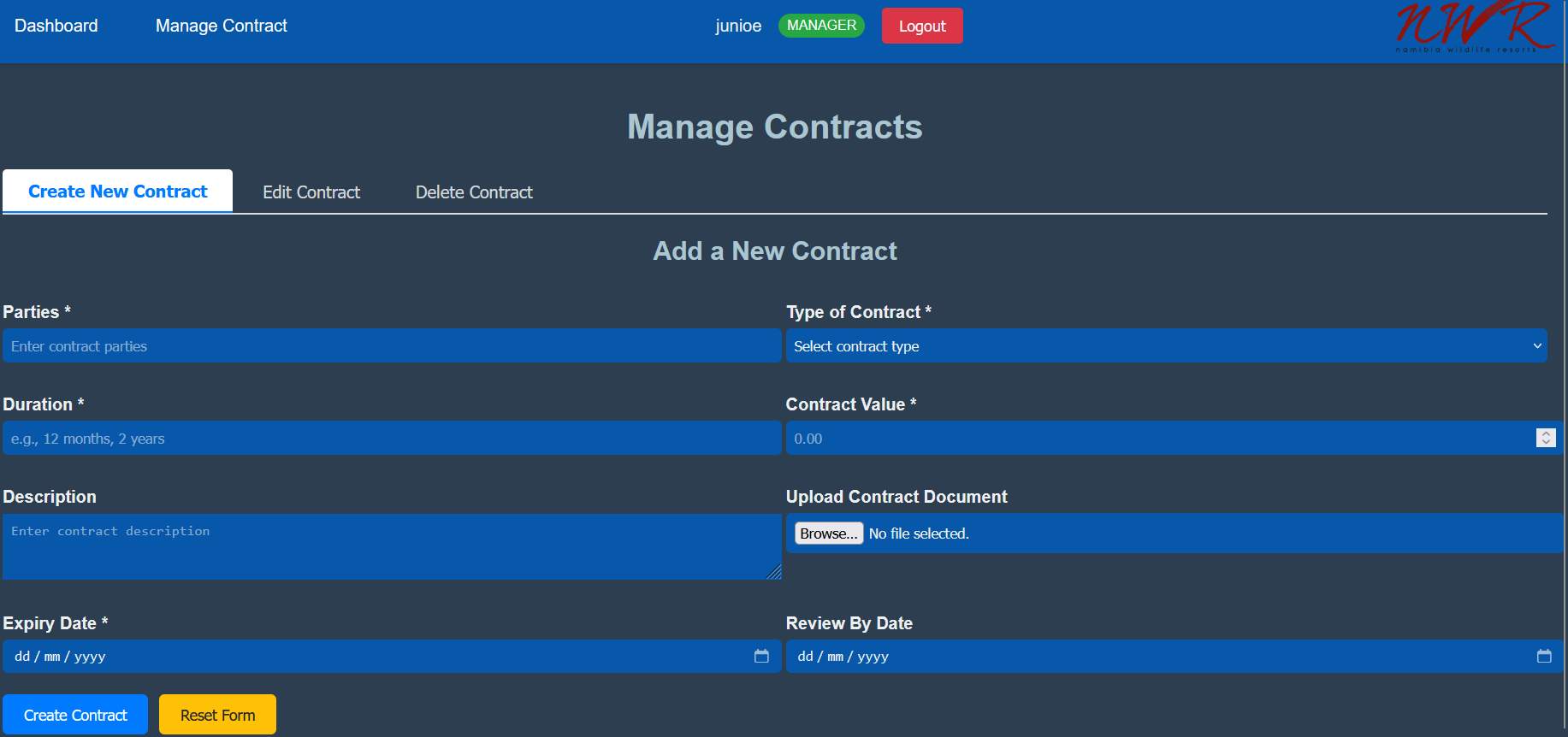


Dashboard: Displays user info, role badge, and navigation links.

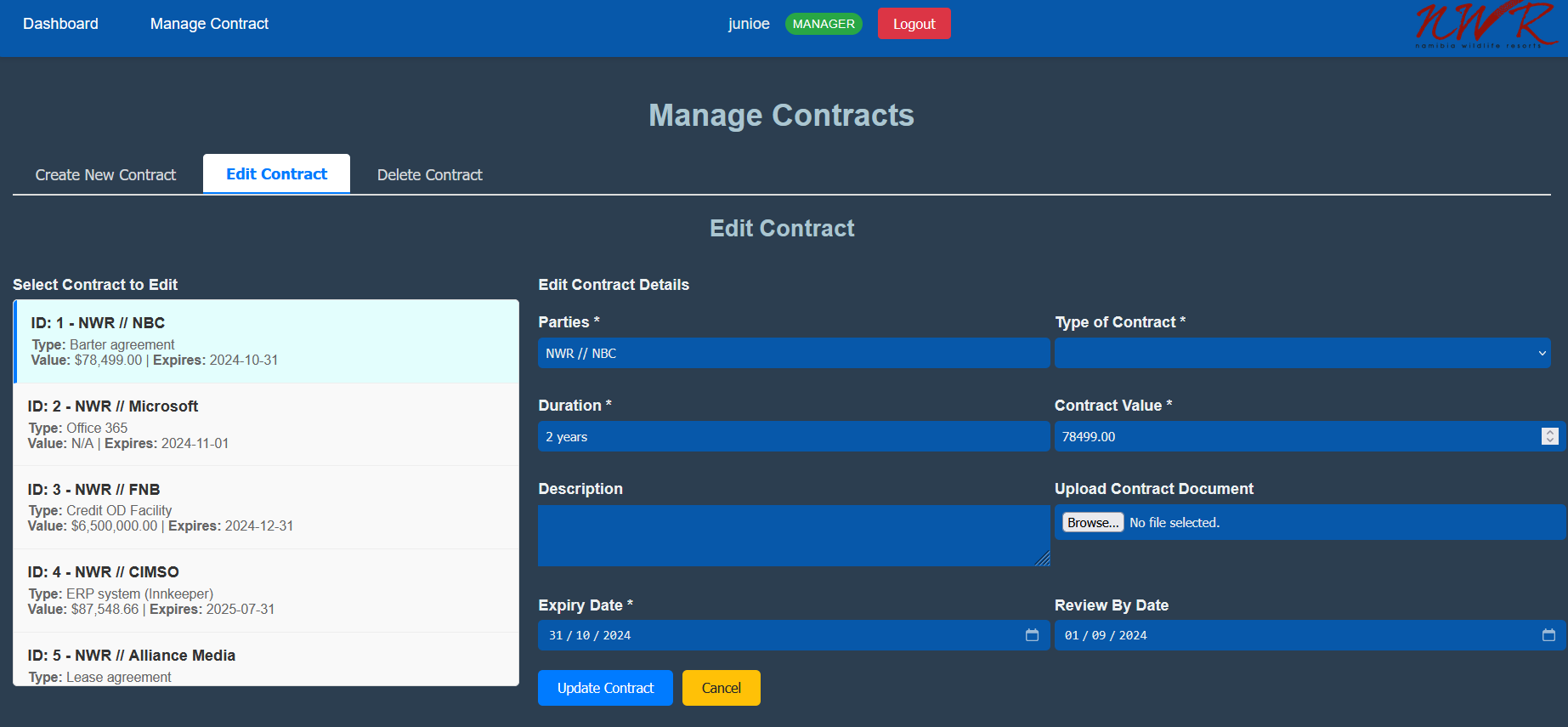


Contract Management: Tabs for Create, Edit, Delete.

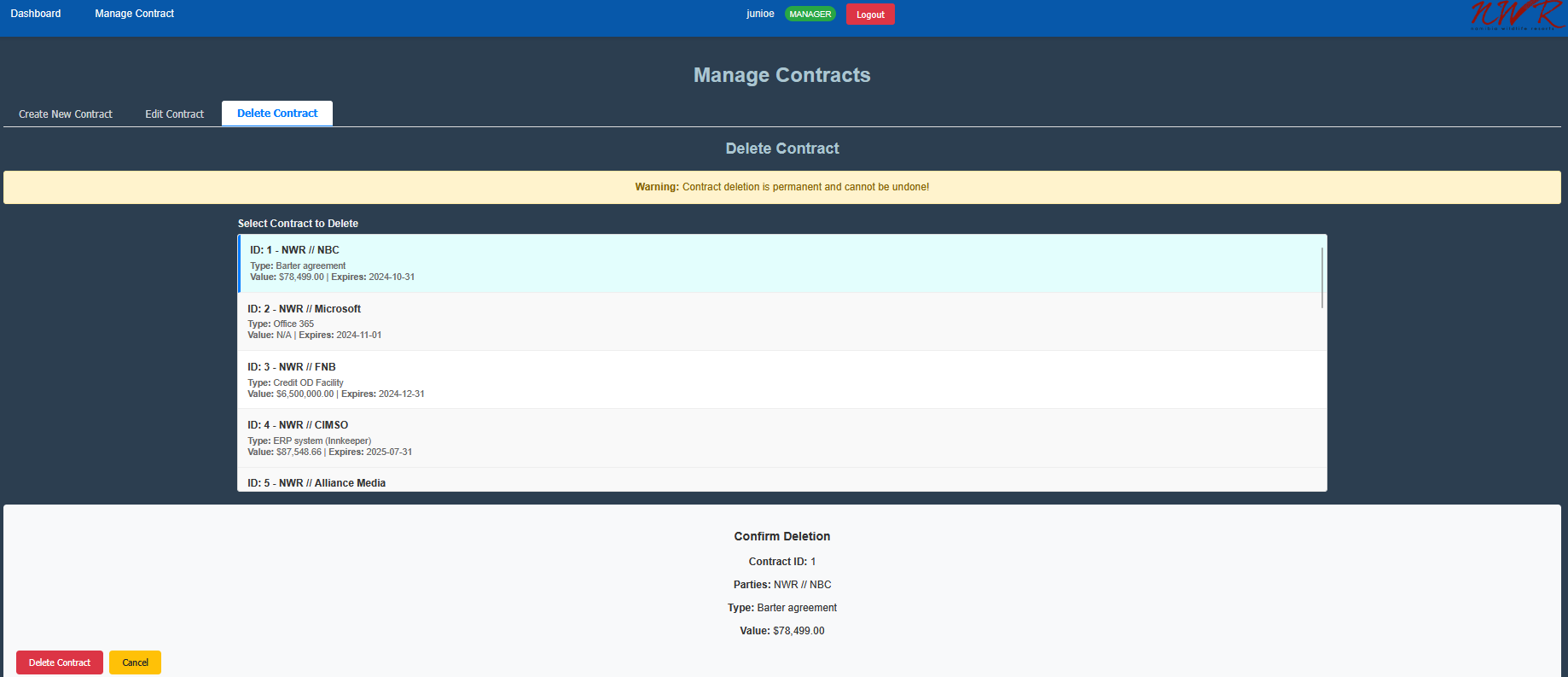
Create tab



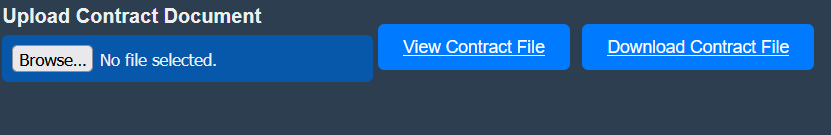
Edit tab



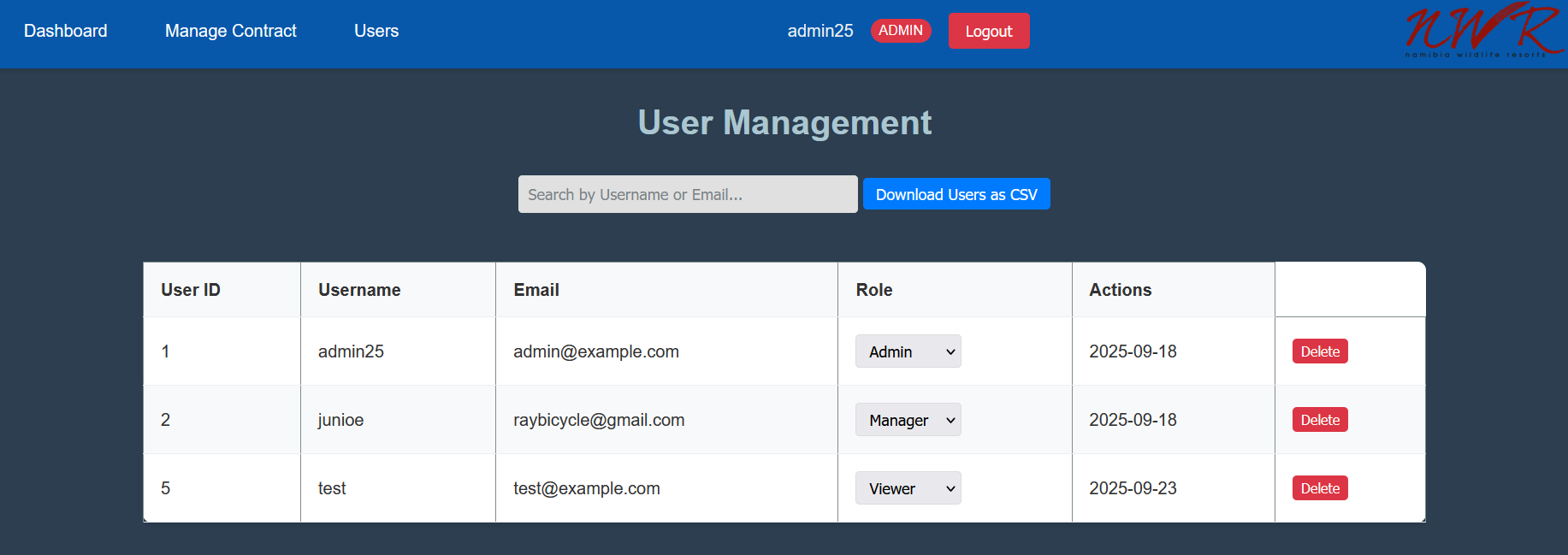
Delete tab



File Management: Uploads with 'View' and 'Download' buttons.



User Management (Admins only): Manage user roles and accounts.



## Hardware Interfaces

Standard server hardware with minimum 4 GB RAM, 2-core CPU.  
Client device: Any modern PC with a stable internet connection and browser.

## Software Interfaces

Backend: PHP (index.php, auth\_handler.php, controllers).

Database: MySQL (contracts, users, contractTypes).

Email: PHPMailer (SMTP).

## Communications Interfaces

HTTP/HTTPS: between client and server.

SMTP for email reminders.

# System Features

Authentication & Authorization (FR1.1–FR1.6)

Contract Management (FR2.1–FR2.6)

File Upload & Access (FR3.1–FR3.5)

Automated Email Reminders (FR4.1–FR4.5)

Logging of Email Activities (FR5.1–FR5.5)

Backend API (FR6.1–FR6.4)

## Authentication and Authorization

4.1.1 Description and Priority

High priority feature, to limit and grant permissions to users, so that only qualified personnel are able to created contracts.

4.1.2 Stimulus/Response Sequences

* Stimulus: User submits login form (email/password).
* Response: System verifies credentials; on success, redirects to dashboard.html with session set; on failure, shows error and redirects to index.php.
* Stimulus: User clicks logout.
* Response: Destroys session and redirects to index.php.
* Stimulus: Unauthenticated user accesses protected page.
* Response: Redirects to index.php.
* Stimulus: Non-admin accesses users.php.
* Response: Shows "Access Denied" message.

4.1.3 Functional Requirements

FR1.1: Check user login status via auth\_handler.php?action=check.

FR1.2: Redirect unauthenticated users to frontend/index.php.

FR1.3: Display an "Access Denied" message for non-managers/admins.

FR1.4: Show user info (username, role badge) in the navbar.

FR1.5: Show "Users" nav link only for admins.

FR1.6: Provide a logout button redirecting to index.php.

## Contract Management Interface

4.1.1 Description and Priority

The main feature of the system, creating, updating and editing contract information. To track when it ends, when it needs to be reviewed and renewed. It is of the Highest priority.

4.1.2 Stimulus/Response Sequences

* Stimulus: Manager clicks "Create New Contract" tab and submits form.
* Response: Validates input; uploads file if provided; creates record in database; shows success message; resets form.
* Stimulus: Manager selects contract in "Edit" tab, updates fields/file, submits.
* Response: Validates; updates database; repopulates form with new data (including file links); shows success.
* Stimulus: Manager selects contract in "Delete" tab, confirms.
* Response: Deletes record; refreshes list; shows success.

4.1.3 Functional Requirements

FR2.1: Three tabs: Create, Edit, Delete, styled with CSS (active tab highlighted).

FR2.2: Create tab: Form with fields (parties, type, duration, value, description, file upload, expiry date, review date).

FR2.3: Edit tab: List contracts; select one to populate a form; show "View"/"Download" buttons for files.

FR2.4: Delete tab: List contracts; select one to confirm deletion.

FR2.5: Display success/error messages for CRUD actions, auto-hiding after 5 seconds.

FR2.6: Format contract value as USD currency (e.g., $1,234.56).

## File Upload and Access

4.1.1 Description and Priority

Uploading contract documents to ensure that they are backed up and able to be reviewed by multiple people at once. This is medium priority, and it keeps data secure and intact, in the unlikely event of a fire or physical damage to the contracts.

4.1.2 Stimulus/Response Sequences

* Stimulus: User uploads file in create/update form.
* Response: Validates file type/size; moves to Uploads/ with unique name; saves path in database; returns updated contract data.
* Stimulus: User clicks "View"/"Download" in edit form.
* Response: Serves file via download.php (view opens in new tab, download saves file).

4.1.3 Functional Requirements

FR3.1: Accept PDF, DOC, DOCX files via <input type="file">.

FR3.2: Save files to backend/Uploads/ with unique names (e.g., parties\_1234567890.pdf).

FR3.3: Store file path in contracts.filepath (e.g., Uploads/filename.ext).

FR3.4: Display "View" and "Download" buttons in the Edit tab when filepath exists.

FR3.5: Serve files securely via backend/index.php=... to prevent direct access.

File size limit?

## Automated Email Remainders

4.1.1 Description and Priority

Reminders to review and negotiate new contracts. The feature allows for a timely reminder to the appropriate individuals with automatic reminders without user intervention. Ensuring that all contracts are kept active. It is of medium priority.

4.1.2 Stimulus/Response Sequences

* Stimulus: Task Scheduler runs send\_contract\_reminders.php daily.
* Response: Queries expiring contracts; checks days left and frequency; sends emails via PHPMailer; logs success/failure.

4.1.3 Functional Requirements

FR4.1: Query contracts with expiry dates within 90, 60, 30, or 3 days from today (September 22, 2025, 15:21 CAT).

FR4.2: Send reminders:

* + - 90 days: Weekly (Mondays).
    - 60 days: Weekly (Mondays).
    - 30 days: Daily.

FR4.3: Use PHPMailer with Gmail SMTP (smtp.gmail.com, port 587, TLS).

FR4.4: Email content: Plain text with contract ID, type, expiry date, and days left.

FR4.5: Run daily via Windows Task Scheduler (C:\xampp\php\php.exe backend/send\_contract\_reminders.php).

FR4.5: Email sent to the user (‘manager’) that creates the contract.

## Logging Automatic Emails

4.1.1 Description and Priority

Log email reminder activities for debugging, to track errors and bugs. Medium Priority.

4.1.2 Stimulus/Response Sequences

* Stimulus: Email send attempt in reminders script.
* Response: Append timestamped message to reminder\_log.txt (success or failure with error info).

4.1.3 Functional Requirements

FR5.1: Write logs to backend/logs/reminder\_log.txt.

FR5.2: Create logs/ directory if missing with write permissions.

FR5.3: Log format: YYYY-MM-DD HH:MM:SS - Message (e.g., 2025-09-22 15:21:00 - Sent reminder to user@example.com).

FR5.4: Rename log file to ` reminder\_log\_YYYYMMDDHHMMSS.txt ` if the file size exceeds 10MB, creates new `reminder\_log.txt` and logs to new file.

FR5.5: Log success (Sent reminder...) and failure (Failed to send...) with PHPMailer errors.

## Backend API

4.1.1 Description and Priority

REST-like API for contract CRUD operations. High Priority.

4.1.2 Stimulus/Response Sequences

* Stimulus: Frontend sends GET/POST to /backend/index.php?action=...
* Response: Authenticates; processes request; returns JSON (e.g., list of contracts or success message with data).

4.1.3 Functional Requirements

FR6.1: Endpoint: /backend/index.php?action=...

* list: Return all contracts (JSON).
* view&id=X: Return contract by ID.
* create: Create contract (POST, FormData).
* update&id=X: Update contract (POST, FormData).
* delete&id=X: Delete contract (DELETE).

FR6.2: Return updated contract data after update (e.g., {"message": "Contract updated", "contract": {...}}).

FR6.3: Handle file uploads in create and update actions.

FR6.4: Restrict create, update, delete to managers/admins via Auth::requireRole('manager'). dsf

# Other Nonfunctional Requirements

## Performance Requirements

System must handle at least 50 concurrent users.

Response time < 2s for CRUD operations.

## Safety Requirements

Contracts stored digitally as backup to prevent data loss.

Logs rotated to prevent storage exhaustion.

## Security Requirements

Passwords hashed (SHA256 or bcrypt).

Session-based authentication.

Role-based authorization.

Secure file serving (no direct URL access).

## Software Quality Attributes

Availability: 99% uptime within internal network.

Maintainability: Modular PHP controllers, separated frontend/backend.

Portability: Runs on any XAMPP-enabled Windows machine.

Usability: Intuitive tabbed interface, minimal training required.

## Business Rules

Only Admins can manage users.

Only Managers/Admins can create/edit/delete contracts.

Viewers may only view/download contracts.

# Other Requirements

File size limit: 10 MB per contract file.

Daily reminders must run automatically without manual intervention.

**Appendix A: Glossary**

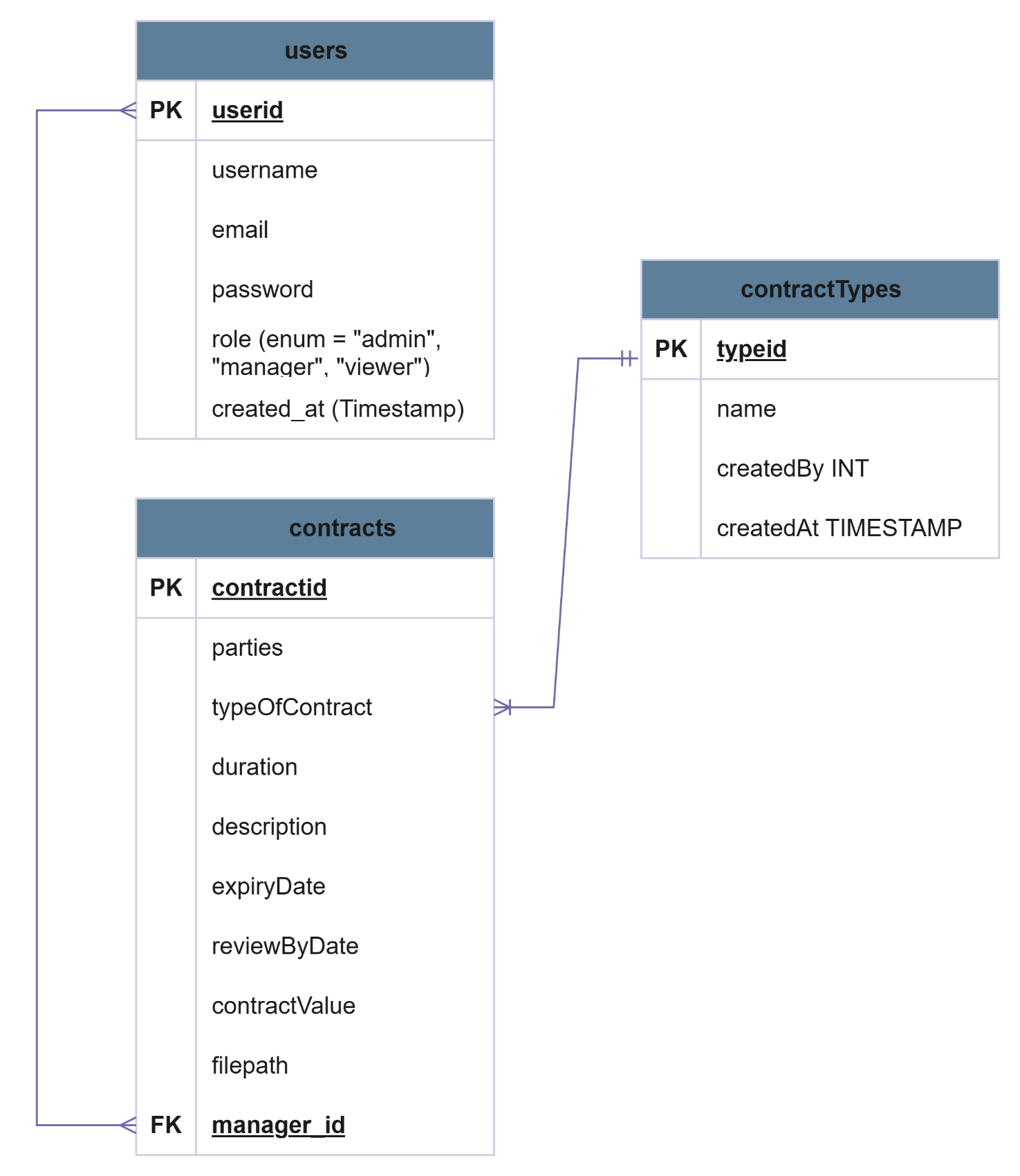
CRUD: Create, Read, Update, Delete.

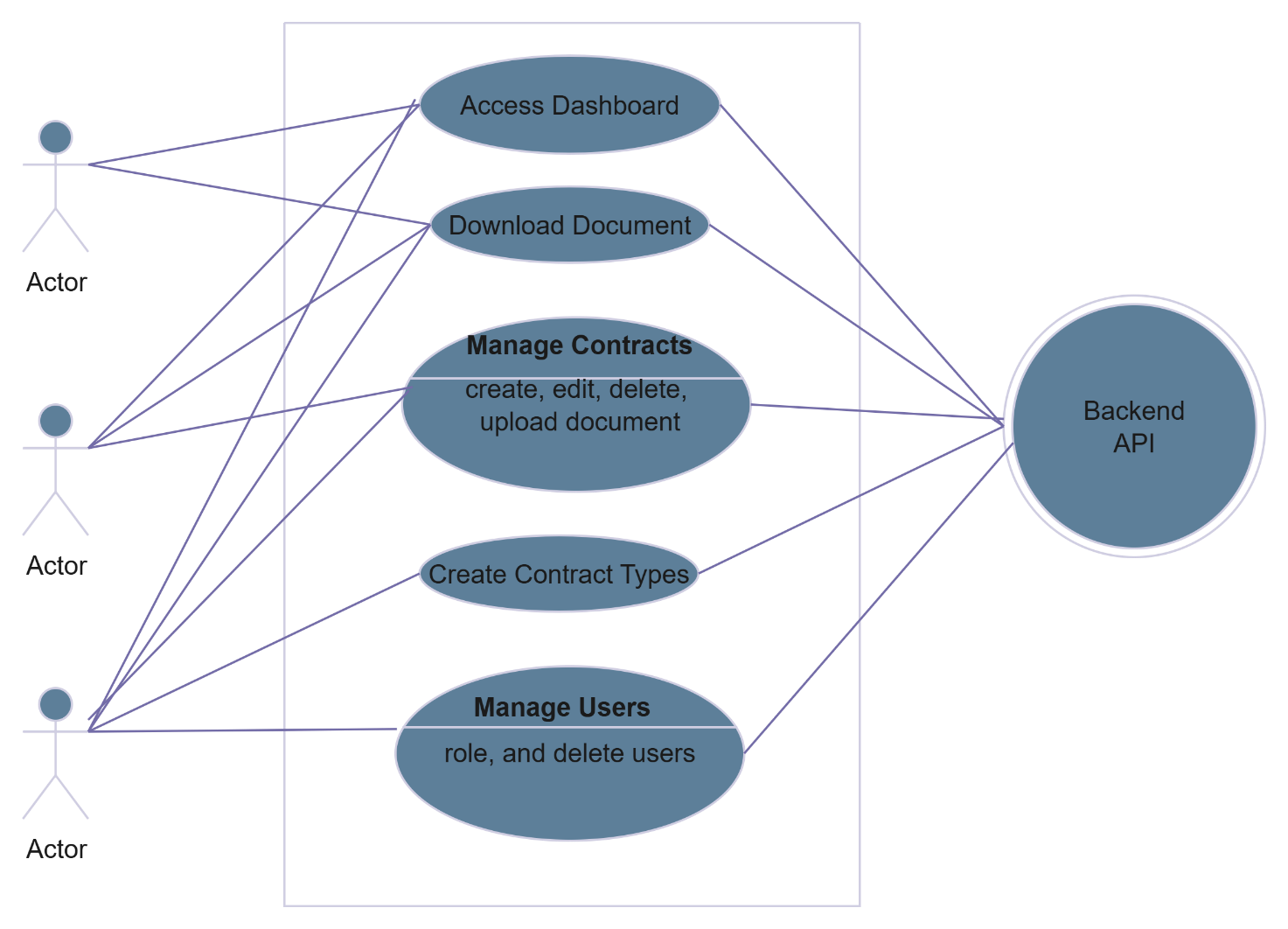
SMTP: Simple Mail Transfer Protocol.

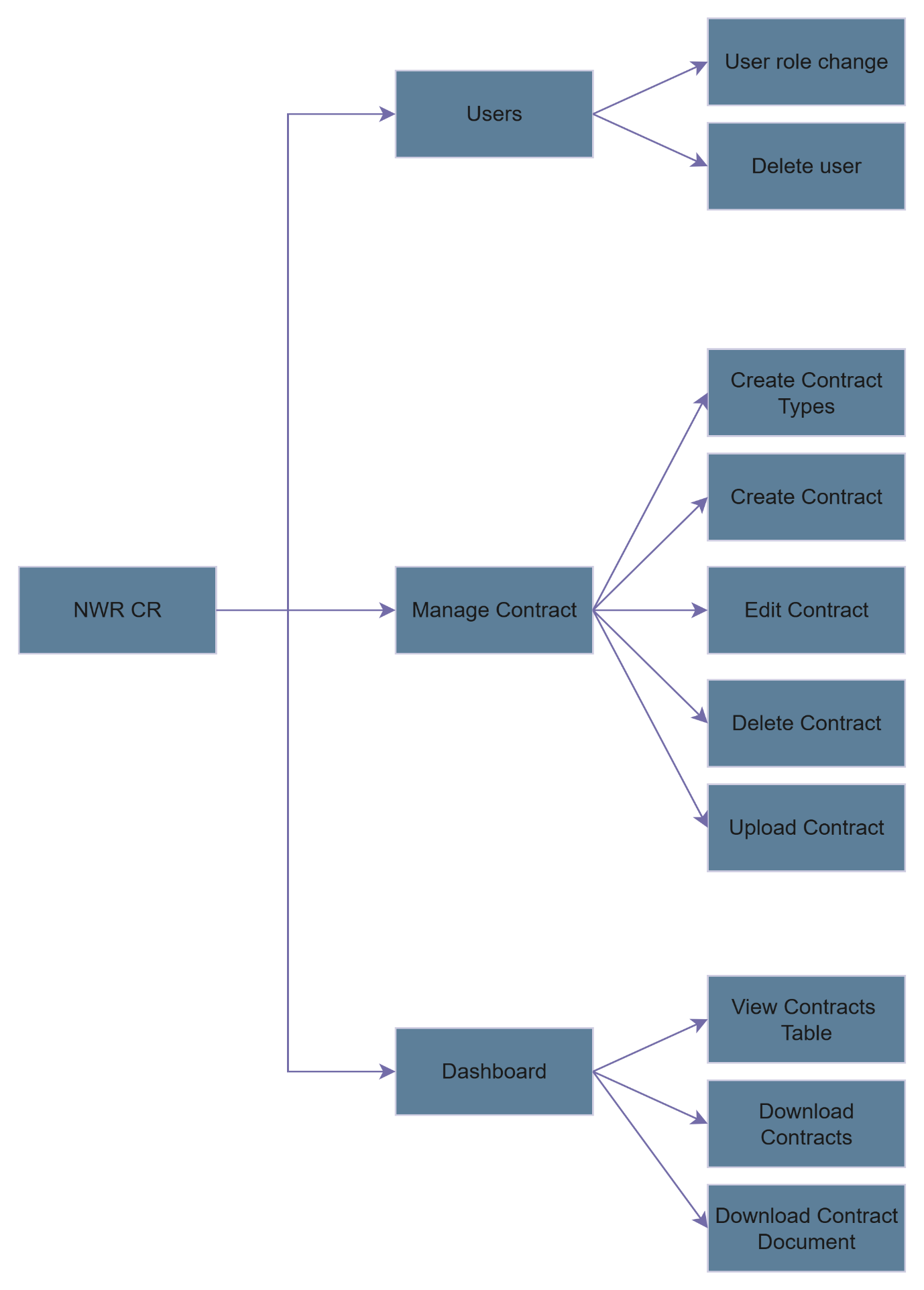
ERD: Entity Relationship Diagram

NWR\_CR: Namibia Wildlife Resorts Contract Registry

**Appendix B: Analysis Models**

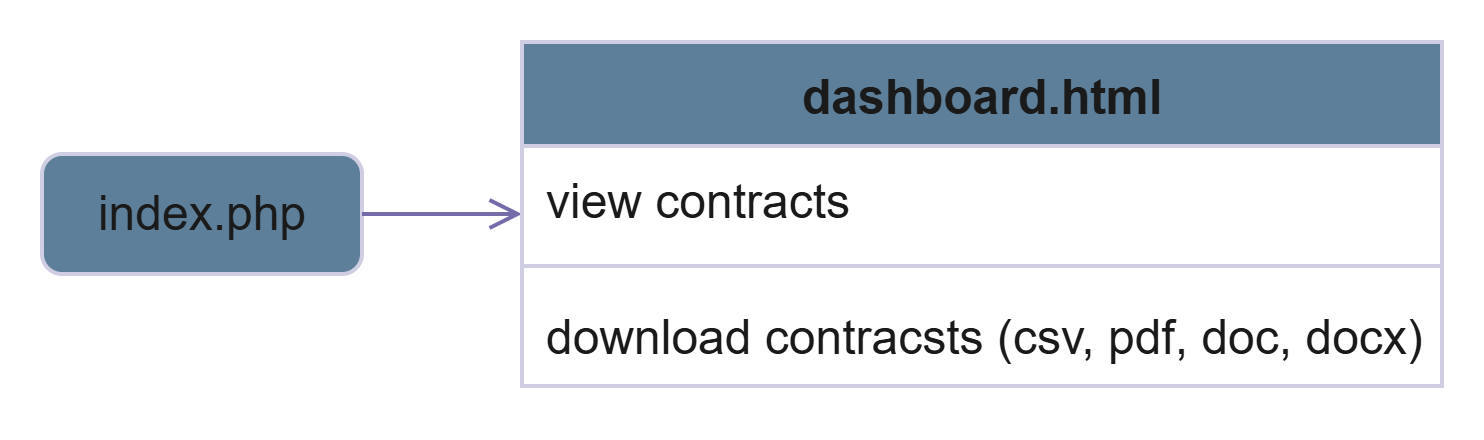
Entity-Relation Diagram

Use Case Diagram

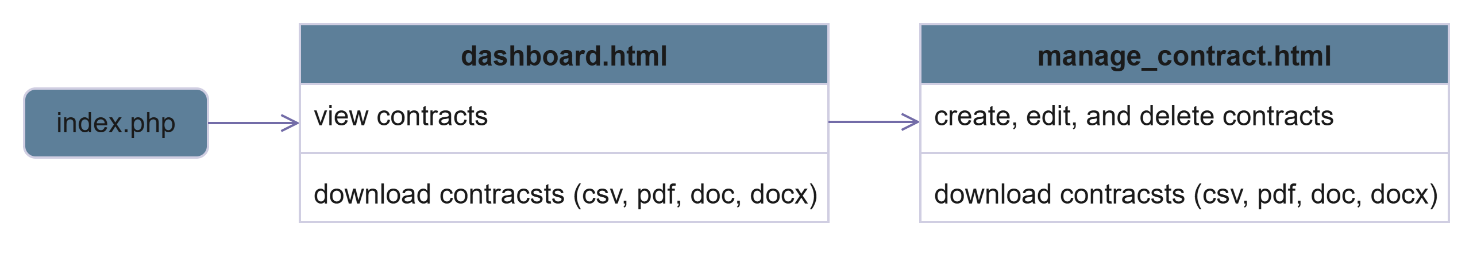
App Flow

User Flow

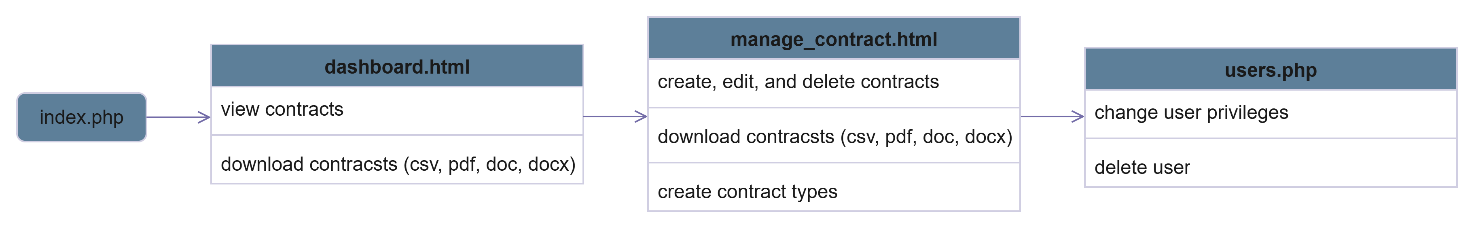
1.1 Viewer



1.2 Manager/Admin



1.3 Admin



**Appendix C: To Be Determined List**

User documentation: User manuals.

Official logging policy