Alexis Ntombiyobuhle Maphosa

[Email address]

Alexis Maphosa

POE part 1

ST10440449

Contract Monthly Claim System (CMCS) Documentation

Overview

The Contract Monthly Claim System (CMCS) is a web-based application designed to manage claims, contracts, and user information for lecturers and programme coordinators. The system is built using the Model-View-Controller (MVC) framework, ensuring a robust and maintainable architecture.

System Architecture

- MVC Framework: The system is built using the MVC framework, which separates concerns into distinct parts, making it easier to manage complex applications and cross-platform compatible.

- Data Handling: A relational database (SQL Server or MySQL) is used to store claims, contracts, and user information. Robust data validation and encryption are implemented to ensure data consistency and security.

- User Authentication: Role-based access control is implemented, with secure login and session management. Passwords are hashed and salted to ensure secure storage.

System Components

- Navigation Bar: A prominent navigation bar is displayed at the top of every page, allowing users to easily move through the system. The links displayed are role-dependent, with lecturers seeing claim-related pages and admins seeing management tools for users, claims, and reports.

- Dashboard: The dashboard is tailored to each role, providing an overview of key statistics and relevant information. Lecturers see claim-related data, while admins see user and claim metrics.

- Main Content Area: The main area of the dashboard is divided into panels or cards, showcasing relevant information and tasks. Lecturers see recent activity, while admins have access to admin information, recent activity, and notifications.

- Tables and Forms: Detailed information is presented in tables and forms, with clear labels and styling. Admins can manage users and claims, while lecturers can submit new claims and update their profile information.

- Profile Pages: Both lecturers and admins have profile pages, displaying personal details and relevant information. Users can update their details and passwords.

Database Structure

- Entities and Attributes:

- Role: RoleID (primary key), RoleName

- User: UserID (primary key), Name, Email, RoleID (foreign key referencing Role)

- Contract: ContractID (primary key), ContractName, StartDate, EndDate, UserID (foreign key referencing User)

- Claim: ClaimID (primary key), ClaimDate, HoursWorked, HourlyRate, ClaimAmount, Status, ContractID (foreign key referencing Contract), UserID (foreign key referencing User)

- Document: DocumentID (primary key), FileName, FileType, UploadDate, UserID (foreign key referencing User)

- Data Types and Constraints: Appropriate data types (datetime, decimal) are used, and constraints are implemented for data consistency (unique, check constraints).

Business Rules

- User Role Selection: Users must select a valid role (Lecturer or Programme Coordinator).

- Contract Association: Contracts are associated with a specific user.

- Claim Association: Claims are associated with a specific contract and user.

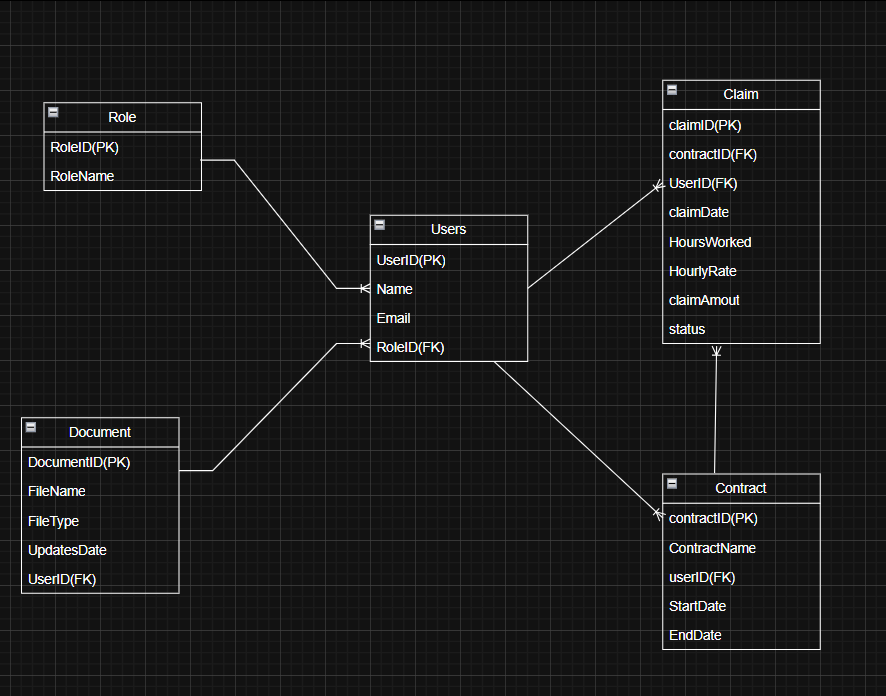
- Document Requirements: Documents (if implemented) must be in a specific format (e.g., PDF) and not exceed a certain size (e.g., 2 GB).

Design Principles

- Consistency: A consistent design language is maintained throughout the system, with common styles and typography used across cards, tables, and forms.

- User Experience: The system is designed to be intuitive and user-friendly, with clear labels and styling.

UML class Diagram for Database



Project Plan

A screenshot of a computer

AI-generated content may be incorrect.

# Bibliography

Anon., n.d. *Edrawmax.* [Online]   
Available at: https://www.edrawmax.com/templates/1001674/  
[Accessed 7 09 2025].

Anon., n.d. *MockFlow.* [Online]   
Available at: https://mockflow.com/glossary/navigation-bar  
[Accessed 8 09 2025].

Anon., n.d. *W3schools.* [Online]   
Available at: https://www.w3schools.com/  
[Accessed 9 2025].