Contract Monthly Claim System (CMCS)

Author: st10383430
Module: PROG6212 POE Part 3

Overview

The Contract Monthly Claim System (CMCS) is a robust WPF-based application designed to streamline the monthly claim submission, approval, and processing workflows for lecturers, program coordinators, and HR departments. It ensures accuracy, transparency, and efficiency in managing contractual claims.

Features

Lecturer View

- Automated Claim Submission:

  Input hours worked and hourly rate to auto-calculate the claim amount.

  Real-time validation for accurate data entry.

- Claim Status Tracking:

  View claim status updates in real-time.

  Program Coordinator and Academic Manager View

- Automated Verification and Approval:

  Review submitted claims against predefined policies.

  Approve or reject claims with detailed feedback.

- Streamlined Approval Workflow:

  Notifications for pending claims.

HR View

- Claim Processing Automation:

  Generate detailed invoices and reports for payment processing.

- Lecturer Data Management:

  Update lecturer profiles, including contact details and personal information.

  General Features

- User Authentication:

  Role-based access (Lecturer, Coordinator, HR).

- Secure Data Handling:

  Data encryption and secure storage of sensitive information.

  Responsive and User-Friendly Interface:

  Designed with intuitive navigation and a clean layout.

Architecture

The CMCS application follows a layered architecture to ensure modularity and scalability:

Presentation Layer (WPF):

Interactive UI designed using XAML.

Implements the MVVM pattern for efficient data binding and separation of concerns.

Application Layer:

Handles business logic, including claim calculations and workflow automation.

Data Access Layer:

Uses Entity Framework Core for database interactions.

Database Layer:

Stores lecturer data, claims, approvals, and logs using SQL Server or Azure SQL Database.

Technology Stack

1. Front-End: WPF (XAML, C#)
2. Back-End: .NET Core

3. Database: SQL Server / Azure SQL
4. Authentication: ASP.NET Identity
5. Reporting: Crystal Reports / SSRS (optional for HR-generated reports)

Setup Instructions

Prerequisites

1. Visual Studio 2022 or later
2. .NET Core SDK 6.0 or higher
3. SQL Server (Local or Azure SQL Database)
4. Git
5. Steps to Run the Application

Clone the Repository:

git clone https://github.com/your-repo/st10383430_PROG6212_POE

cd st10383430_PROG6212_POE

Configure the Database:

Update the connection string in appsettings.json or App.config to point to your database.

Run Migrations:

dotnet ef migrations add InitialCreate

dotnet ef database update

Launch the Application:

Open the solution in Visual Studio.

Build and run the project.

Usage Instructions

For Lecturers:

1. Log in with your lecturer credentials.
2. Navigate to the "Submit Claim" section.
3. Enter hours worked and hourly rate.
4. Submit the claim for approval.

For Coordinators and Managers:

1. Log in with your coordinator credentials.
2. Navigate to the "Approve Claims" section.
3. Review pending claims and approve/reject them.

For HR:

1. Log in with HR credentials.
2. Generate reports for approved claims.
3. Manage lecturer data from the dashboard.

Contributing

Fork the repository.

Create a feature branch (git checkout -b feature-name).

Commit your changes with descriptive messages.

Push to your forked repo and create a pull request.

Version Control

GitHub Repository: Link to repository

Regular commits with descriptive messages were made throughout development.

Screenshots

Lecturer View: Submit Claim

Coordinator View: Approve Claims

HR View: Generate Reports
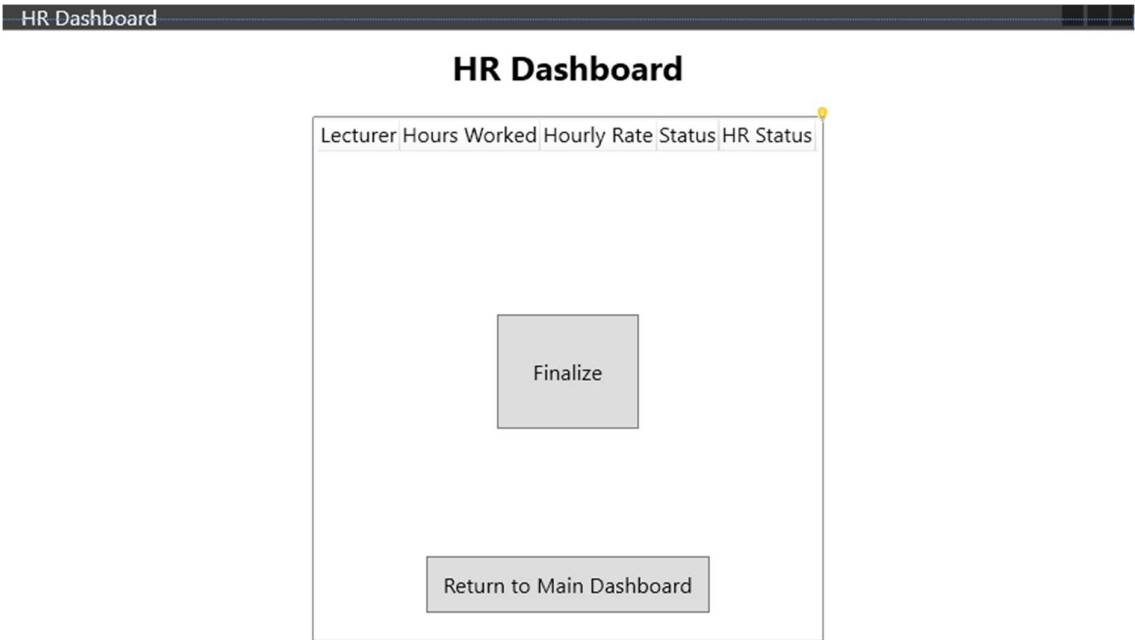
PowerPoint Presentation

The accompanying PowerPoint presentation highlights the system's features, benefits, and functionality.

Conclusion

The CMCS is a powerful and user-friendly system designed to eliminate inefficiencies in claim management processes. Its modular and scalable architecture ensures future enhancements can be integrated seamlessly.