Joshua Gombele
ST10448749

📑 Draft Report Part 1: Project Planning & Prototype Development

Introduction

The Contract Monthly Claim System (CMCS) is a proposed .NET Core application designed to streamline the submission and approval of claims for Independent Contractor lecturers. This report outlines the prototype design of the system, including database structure, graphical user interface layouts, project planning, and version control considerations. At this stage, the prototype focuses on design only and does not include functional implementation.

1. Documentation of Design Choices

The design of the CMCS prioritizes simplicity, transparency, and role-based access. The database is structured around four main entities: Lecturers, Claims, Supporting Documents, and Users. This ensures data integrity and enables efficient claim management workflows.

The GUI is based on a role-driven layout:

- Lecturers can create and submit claims, upload supporting documents, and track claim status.
- Programmed Coordinators and Academic Managers can review and approve claims, ensuring accountability.

Assumptions:

- All lecturers are pre-registered in the system with valid credentials.
- Hourly rates are predetermined and stored in the lecturer profile.
- Only PDF and image files are accepted as supporting documents.

Constraints:

- The prototype must be built using .NET Core (MVC or WPF).
- User authentication and document storage will be implemented in later phases.
- The GUI at this stage is non-functional.

Joshua Gombele
ST10448749

2. UML Class Diagram (Database)

The system database consists of the following classes:

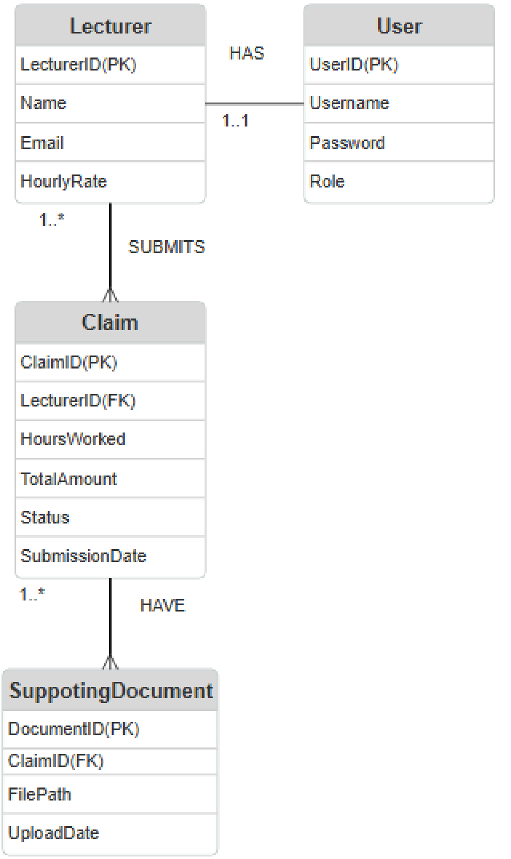
- Lecturer (LecturerID, Name, Email, HourlyRate)
- Claim (ClaimID, LecturerID, HoursWorked, TotalAmount, Status, SubmissionDate)
- SupportingDocument (DocumentID, ClaimID, FilePath, UploadDate)
- User (UserID, Username, PasswordHash, Role)

Relationships:

- A Lecturer can submit many Claims (1..*).
- Each Claim can have multiple Supporting Documents (1..*).
- Each User has a defined role (Lecturer, Coordinator, Manager).
- Each lecturer has one user (1..1)

This design ensures scalability and easy integration with authentication systems.

Joshua Gombele
ST10448749



3. Project Plan

The project plan follows an iterative development approach.

Task	Duration	Dependency
Requirement Analysis & Documentation	1 Day	None
UML Diagram & Database Design	1Day	Documentation
GUI Wireframe Design (WPF/MVC)	2 Days	Database Design
Prototype Compilation & Refinement	1 Day	GUI Design
GitHub Version Control Updates	1 day	All tasks

Estimated Timeline: 6 Day.

4. GUI / UI Design

The prototype GUI is developed in WPF using .NET Core, with an emphasis on user-friendliness and intuitive navigation. Key screens include:

- Login Screen (role-based access)
- Lecturer Dashboard (Submit Claim, Upload Documents, View Claim Status)
- Coordinator Dashboard (Verify Claims, Approve/Reject)
- Manager Dashboard (Final Approval, Generate Reports)

Each screen uses consistent styling, clear buttons, and logical layouts to enhance usability.

5. Version Control

All project files will be managed via GitHub, with at least five commits reflecting project progress. Example commit messages:

- “Initial UML class diagram added”
- “Created lecturer dashboard wireframe”
- “Added project plan draft”
- “Uploaded GUI mockups”
- “Refined documentation for submission”

Conclusion

This prototype design provides a structured foundation for the CMCS. The UML diagram, project plan, and GUI wireframes ensure a clear direction for development, while version control supports iterative improvements. In the next phase, the system will be made functional with database integration and user authentication.

References

- Microsoft, 2024. *Role-based authorization in ASP.NET Core*. [online] Available at: https://learn.microsoft.com/en-us/aspnet/core/security/authorization/roles?view=aspnetcore-9.0 [Accessed 17 Sep. 2025].
- Microsoft, 2023. *Claims-based authorization in ASP.NET Core*. [online] Available at: https://learn.microsoft.com/en-us/aspnet/core/security/authorization/claims?view=aspnetcore-9.0 [Accessed 17 Sep. 2025].
- Duende Software, 2025. *Role-Based Access Control with ASP.NET Core Identity*. [online] Available at: https://duendesoftware.com/learn/role-based-access-control-asp-net-core-identity/ [Accessed 17 Sep. 2025]. duendesoftware.com
- Telerik, 2024. *UX Crash Course: Wireframing*. [online] Available at: https://www.telerik.com/blogs/ux-crash-course-wireframing [Accessed 17 Sep. 2025].
- C-Sharp Corner, (n.d.). *Working With Wireframes*. [online] Available at: https://www.c-sharpcorner.com/UploadFile/ansh06031982/working-with-wireframes/ [Accessed 17 Sep. 2025].
- Github Link: https://github.com/BokosaJG/prog6212st10448749.git