**PROG 6212**

**ST10441421**



**Odirile Masemola**

**DISD 2**

Table of Contents

[1. Introduction 2](#_Toc207001756)

[2. Assumptions and Constraints 2](#_Toc207001757)

[3. Database Design (ERD) 2](#_Toc207001758)

[4. GUI Design 3](#_Toc207001759)

[5. Project Plan 3](#_Toc207001760)

[6. Version Control 4](#_Toc207001761)

[7. Conclusion 4](#_Toc207001762)

# 1. Introduction

This Portfolio of Evidence documents the design and planning of the Contract Monthly Claim System (CMCS). The system will allow independent contractor lecturers to submit monthly claims, which will then be reviewed by coordinators and managers, and finally processed by HR for payment.

The purpose of Part 1 is to provide the blueprint for the system, including assumptions, constraints, database design, GUI design, version control, and a project plan. No coding is included at this stage.

# 2. Assumptions and Constraints

* The system will be built using ASP.NET Core MVC.
* Database: SQL Server with Entity Framework Core.
* Authentication will use ASP.NET Identity.
* File uploads allowed: PDF, DOCX, XLSX (max 5 MB).
* Roles include: Lecturer, Coordinator, Manager, HR.
* Claims must include: Hours worked, Hourly rate, and supporting documents.
* All calculations (e.g., Total Amount) are automated.
* The prototype UI will be non-functional for Part 1 but will show layouts and workflow.

# 3. Database Design (ERD)

The system will have the following entities:

* User – stores lecturer, coordinator, manager, and HR details.
* Claim – stores claim month, hours worked, hourly rate, total amount, and status.
* ClaimDocument – stores uploaded files linked to a claim.
* Approval – stores approval or rejection actions with notes.
* Payment – stores payment details for approved claims.

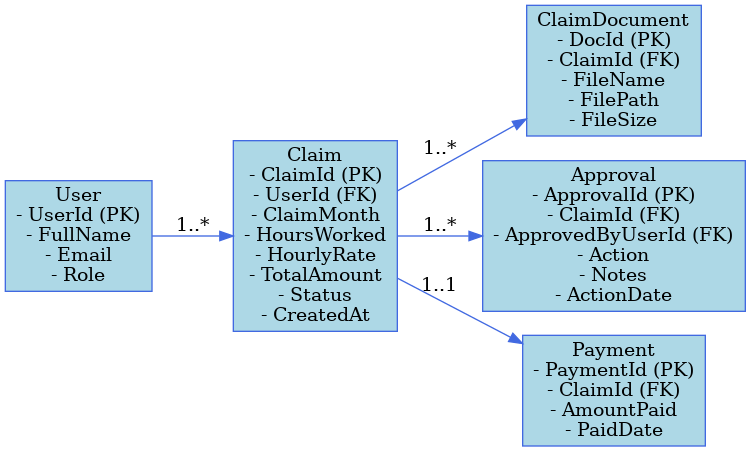


Figure 1

# 4. GUI Design

The GUI will follow a modern, responsive design with a clean card-based interface.

* Lecturer Dashboard: Claim submission form, upload area, and claim history.
* Coordinator Dashboard: List of pending claims, Approve/Reject buttons.
* Manager Dashboard: Higher-level approvals.
* HR Dashboard: Export reports and manage payments.

Visual style: Teal and charcoal theme, simple icons, progress bars, and status badges for Pending, Approved, or Rejected claims.

*(Insert GUI mockup screenshots here – can be hand-drawn or digital)*

# 5. Project Plan

|  |  |  |
| --- | --- | --- |
| Week | Task | Deliverable |
| 1 | Project setup and design | Repo created, UML draft |
| 2 | Database design | ERD complete |
| 3 | GUI mock-ups | Screens ready |
| 4 | Documentation | Report draft |
| 5 | GitHub commits | Evidence uploaded |
| 6 | Final report | Part 1 submission |

# 6. Version Control

GitHub will be used to manage the project. At least 5 commits will be made in Part 1, including:

1. Initial commit – repository created
2. Added UML diagram
3. Added GUI mockups
4. Added Part 1 written report
5. Final update for Part 1

# 7. Conclusion

Part 1 of the CMCS project sets a strong foundation for development. By defining the database, designing the GUI, and planning the project, the system is well-prepared for implementation in Part 2. The use of GitHub ensures proper version control and professional documentation of progress.