**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 is designed to provide a streamlined, digital solution for managing independent contractor lecturers’ monthly claims. The system will work by allowing lecturers to log in securely and capture their working hours and hourly rate, attach supporting documents, and then submit the claim online. Once submitted, the claim will automatically move through a structured workflow.

At the first stage, Programme Coordinators will be able to review submitted claims, verify the details, and either approve or reject them with comments. If approved, the claim will be escalated to the Academic Manager for a second level of validation. The Manager will have a similar option to approve or reject the claim. Approved claims will then be sent to the HR department, where payment preparation and final processing will take place.

The system will also allow lecturers to track the status of their claims in real time. A clear visual status indicator (e.g., Pending, Approved, or Rejected) will keep users informed at every stage of the process. Lecturers will also be able to view their claim history for previous months, including details of approvals and payments. HR will have access to reporting tools that will allow them to generate summaries of processed claims, manage lecturer records, and ensure accountability.

The purpose of Part 1 is to provide the blueprint for the system. This includes defining assumptions and constraints, designing the database schema (ERD), creating mock-ups of the graphical user interface (GUI), setting up version control, and drafting a realistic project plan. No coding is included at this stage; the focus is on design, structure, and clarity of workflow.

# 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.

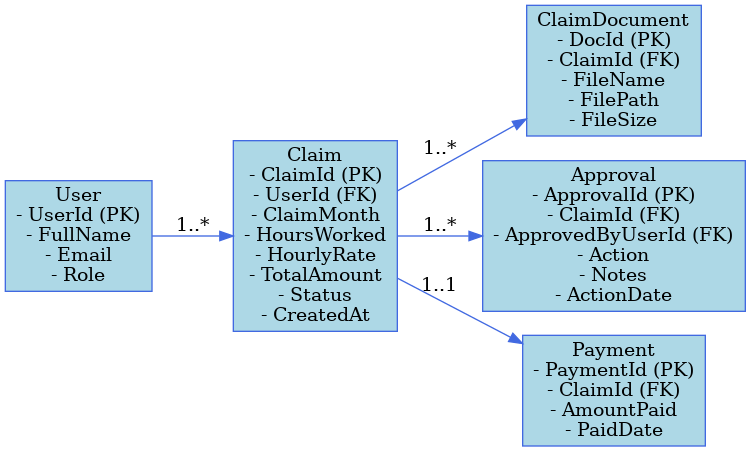


Figure 1

# 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.

A screenshot of a computer

AI-generated content may be incorrect.Visual style: Teal and charcoal theme, simple icons, progress bars, and status badges for Pending, Approved, or Rejected claims.

Figure 2

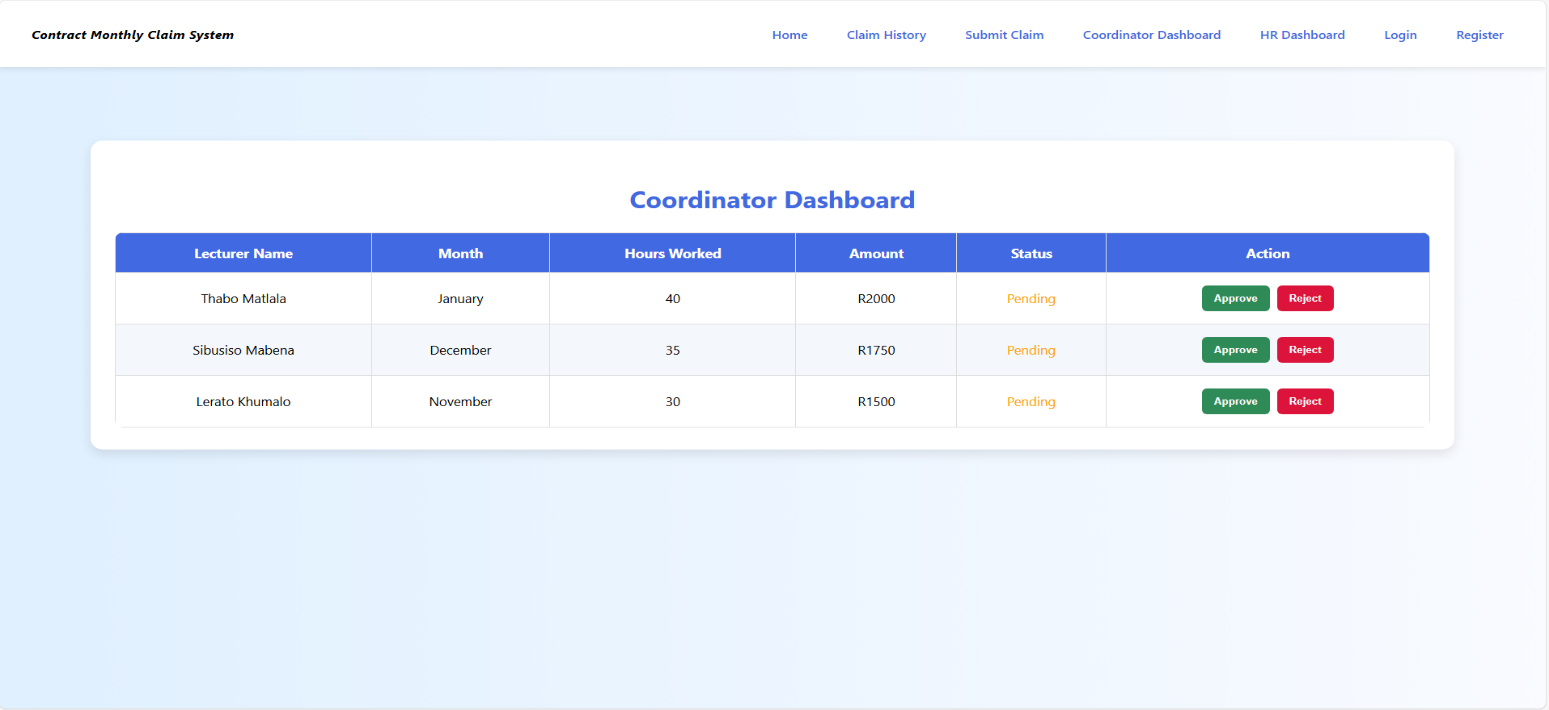
  
Coordinator Dashboard mock-up showing pending claims list with claim details, along with Approve and Reject action buttons for claim processing.

Figure 3

# 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 timeline, the system is well-prepared for implementation in Part 2. The structured workflow ensures that each role (Lecturer, Coordinator, Manager, and HR) has clear responsibilities within the system, which will improve efficiency and transparency compared to manual claim processes. The prototype design demonstrates how claims will move seamlessly from submission to approval and finally to payment, while also giving lecturers the ability to track their progress. The inclusion of document uploads and automated calculations reduces errors and provides accountability at every stage. Version control through GitHub adds an additional layer of professionalism by ensuring that all design work is properly recorded and updated.

In conclusion, Part 1 delivers a comprehensive plan that highlights not only the technical design but also the user experience and operational improvements the system will bring. This foundation ensures that Part 2 can focus on implementing functionality with confidence that the design decisions align with real-world requirements.