PROG6212 Portfolio of Evidence – Part 1: Contract Monthly Claim System (CMCS)

ST10184131

hollo KABELO MAMPANE

2025

The Contract Monthly Claim System (CMCS) is a .NET web-based application designed to streamline the process of submitting, verifying, and approving monthly claims for Independent Contractor (IC) lecturers. The system improves efficiency, reduces errors, and provides a transparent workflow for both lecturers and administrators. This report outlines the design choices, database structure, GUI prototype, assumptions, constraints, and project plan for Part 1 of the POE.

2. Documentation

2.1 Design Choices, Database Structure, and GUI Layout

The CMCS is designed with accuracy, accountability, and usability in mind.

Database Structure:

Lecturer: Stores lecturer details (ID, name, email).

Claim: Represents monthly claims linked to lecturers, containing claim date, status, and amount.

ClaimLineItem: Details individual teaching sessions/tasks for a claim.

SupportingDocument: Stores uploaded files for verification.

ClaimApproval: Records approval actions by Programme Coordinators and Academic Managers.

ProgrammeCoordinator and AcademicManager: Separate roles for a two-step approval process.

GUI Layout:

Lecturer Dashboard: Submit claims, upload documents, track claim status.

Coordinator Dashboard: Verify claims, check supporting documents.

Manager Dashboard: Approve/reject claims after verification.

Claim Tracking Screen: Clear visual status updates for lecturers.

These choices ensure the system replicates the real-world workflow while remaining intuitive and user-friendly.

2.2 Assumptions and Constraints

Assumptions:

Each lecturer submits their own claim.

Claims must contain at least one line item.

Coordinators always verify claims before managers approve.

Supporting documents are mandatory for processing.

Users have basic digital literacy.

Constraints:

File uploads limited to 5MB.

System requires internet connectivity.

Claims only for the current or previous month.

One claim per lecturer per month.

Role-based access restrictions enforced for security.

## UML Class Diagram

A diagram of a computer flowchart

Description automatically generated

## PROJECT PLAN:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| WEEK | TASK | DEPENDENCY | DELIVERABLE | RESPONSIBLE PERSON | TIMELINE (DATES) | STATUS/NOTES |
| 1 | Gather requirements, identify entities | None | Draft requirements + initial UML sketch | Student | Week 1 (e.g., 16–20 Sept) | Start of project |
| 2 | Finalize UML & document design choices | Week 1 | Completed UML + written design documentation | Student | Week 2 (e.g., 23–27 Sept) | Expand database structure |
| 3 | Create GUI wireframes (Lecturer, Coordinator, Manager views) | Week 2 | Static GUI prototype (non-functional) | Student | Week 3 (e.g., 30 Sept – 4 Oct) | Ensure user-friendly layout |
| 4 | Prepare project plan & draft report (400–500 words) | Week 1-3 | Project plan + draft of Part 1 report | Student | Week 4 (e.g., 7–11 Oct) | Align with rubric |
| 5 | Version control: push all work to GitHub with 5 commits | Week 1-4 | Report, UML diagram, wireframes uploaded | Student | Week 5 (e.g., 14–18 Oct) | Clear commit messages |
| 6 | Final review and submission | Week 1-5 | Final polished Word report + GitHub repo link | Student | Week 6 (e.g., 21–25 Oct) | Submit before deadline |

## GUI Prototype

<https://github.com/Mampane23/CMCS_Prototypefinal.git>

gh repo clone Mampane23/CMCS\_Prototypefinal