**PROG6212 Portfolio of Evidence – Part 1**

**Project Title:** Contract Monthly Claim System (CMCS)  
**Student Name:** Mampe Shaan Motsholane  
**Module:** PROG6212 – Programming  
**Lecturer:** M Nkoane  
**Date:** 09 September 2025

**INTRODUCTION**

The Contract Monthly Claim System (CMCS) is designed to help streamline the monthly claim submission process for Independent Contractor (IC) lecturers. Currently, the process is time-consuming, paper-based, and prone to errors. The CMCS prototype aims to digitize claim submissions, approvals, and record-keeping to improve efficiency and accuracy.

* Documentation of the project problem and objectives
* UML class diagram for the database design
* A realistic project plan with a Gantt chart
* Graphical User Interface (GUI) wireframes

**DOCUMENTATION**

**Problem Description**

Currently, Independent Contractor lecturers submit their monthly claims manually, often via email or paper-based forms. This results in delays, missing documents, and difficulties in tracking claims. Programme Coordinators and Academic Managers also face challenges when approving and verifying claims because there is no centralized system.

**Objectives**

* To provide a user-friendly system for lecturers to submit claims online.
* To allow Programme Coordinators to verify and approve claims efficiently.
* To provide Academic Managers with tools to monitor, approve, and generate reports on lecturer claims.
* To reduce errors, delays, and inefficiencies in the claim process.

**Scope of the Prototype**

* Lecturers claim submission
* Programme Coordinator approval
* Academic Manager approval
* Basic reporting functions

**UML CLASS DIAGRAM**

The UML class diagram represents the structure of the CMCS database.

**Classes**

* **Lecturer** (LecturerID, Name, Surname, Email)
* **Claim** (ClaimID, LecturerID, HoursWorked, HourlyRate, Status, DateSubmitted)
* **ProgrammeCoordinator** (CoordinatorID, Name, Email)
* **AcademicManager** (ManagerID, Name, Email)

**Relationships**

Lecturer

* Attributes: LecturerID, Name, Email, Password
* Role: Submits claims, uploads documents.
* Relationship: *One Lecturer can submit many Claims*.

Claim

* Attributes: ClaimID, Month, HoursWorked, Amount, Status, DateSubmitted
* Role: Core entity that goes through submission → verification → approval.
* Relationship: *Each Claim belongs to one Lecturer* and may have many Supporting Documents.

SupportingDocument

* Attributes: DocumentID, FileName, FilePath, UploadDate
* Role: Attached to Claims for verification.
* Relationship: *One Claim can have multiple documents*.

Programme Coordinator

* Attributes: CoordID, Name, Email, Password
* Role: Verifies Lecturer claims (approve/reject at verification stage).

Academic Manager

* Attributes: ManagerID, Name, Email, Password
* Role: Gives final approval and generates reports.

(Below is the logical Database View)

**\***

PROGRAMME COORDINATION

CoordinationID - int **PK**

ClaimID - **FK**

Name - string

Email - string

Password - string

SUPPORTING DOCUMENT

DocumentID - int **PK**

ClaimID – int **FK**

FileName - string

UploadAt - date

**1**

CLAIM

ClaimID – int **PK**

LectureID - int **FK**

Month - string

TotalHours - int

TotalAmount - decimal

Status - string

DateSubmitted- date

LECTURE

LectureID – int **PK**

Name - string

Email - string

Password - string

**1**

**\***

ACADEMICMANAGER

ManagerID – int **PK**

ClaimID – int **FK**

Name - string

Email - string

Password - string

*CLAIM*

*CLAIM*

VERIFIES

APPROVES

**1**

**PROJECT PLAN**

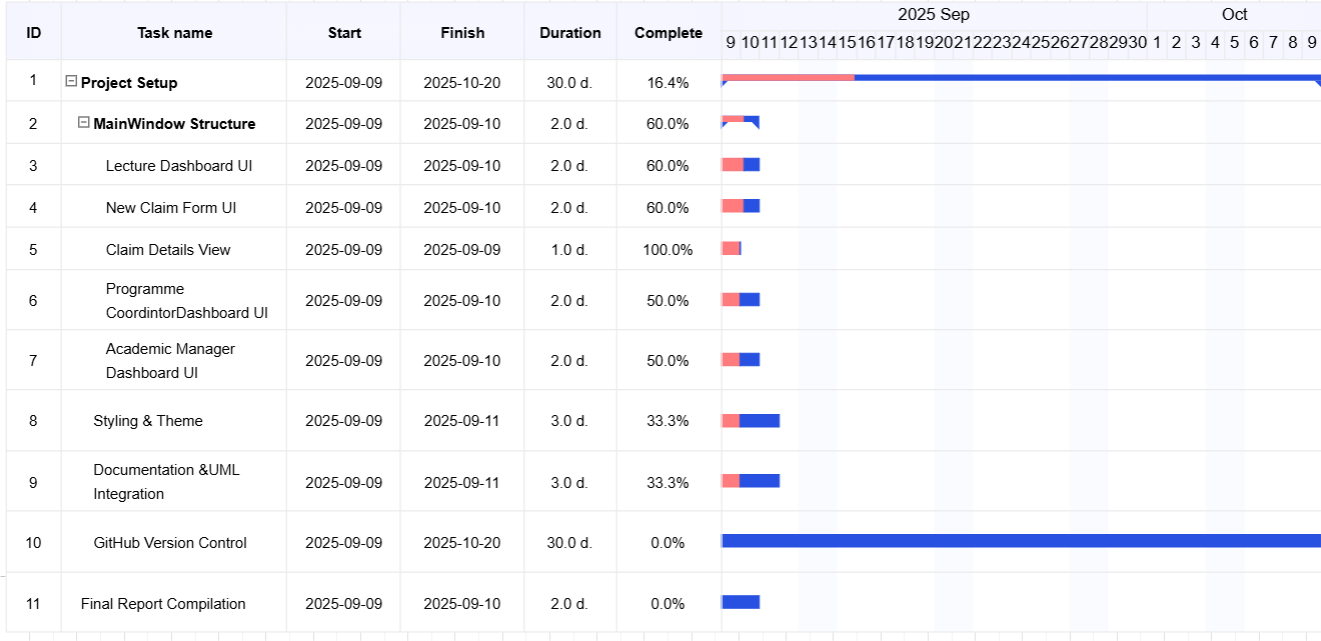
The project plan outlines the tasks, dependencies, and estimated time required to complete the CMCS prototype.

Week 1: Tasks 1–3 (setup + lecturer dashboard).

Week 2**:** Tasks 4–7 (new claim form + coordinator + manager dashboards).

Week 3: Tasks 8–11 (styling, documentation, UML, final report).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task ID** | **Task Name** | **Task Description** | **Dependencies** | **Duration** |
| T1 | Project Setup | Create Visual Studio project, configure .NET WPF solution, add GitHub repo. | None | Day 1 |
| T2 | MainWindow Structure | Add TabControl in MainWindow, define placeholders for Lecturer, Coordinator, Manager dashboards. | T1 | 2 Days |
| T3 | Lecture Dashboard UI | Design Lecturer tab: claim history (ListView), buttons for *Submit, Upload, Track*. | T2 | 2 Days |
| T4 | New Claim Form UI | Create form tab with inputs: month, hours worked, claim amount (readonly). | T2 | 2 Days |
| T5 | Claim Details View | Create Lecturer claim details tab showing static claim info. | T2 | 1 Day |
| T6 | Programme CoordintorDashboard UI | Add tab: pending claims list, verify/reject buttons. | T2 | 2 Days |
| T7 | Academic Manager Dashboard UI | Add tab: approval list, approve/reject/report buttons. | T2 | 2 Days |
| T8 | Styling & Theme | Apply light/dark green theme, improve readability, apply consistent formatting. | T3 – T7 | 2 Days |
| T9 | Documentation &UML Integration | Write documentation, update UML diagram, ensure consistency with project. | Parallel with T3-T7 | 3 Days |
| T10 | GitHub Version Control | Commit after project setup, UI drafts, lecturer UI, coordinator UI, final polish. | Throughout project | Continuous |
| T11 | Final Report Compilation | Combine documentation, UML, project plan, screenshots of prototype. | T3-T9 | 2 Days |



**REFERENCE**

For the Gnatt Chart I used

EDrawMax. (2025) *Wondershare EDrawMax* Available at: <https://www.edrawmax.com/online/en/> (Accessed: 27 August 2025).

**GUI / UI PROTOTYPE**

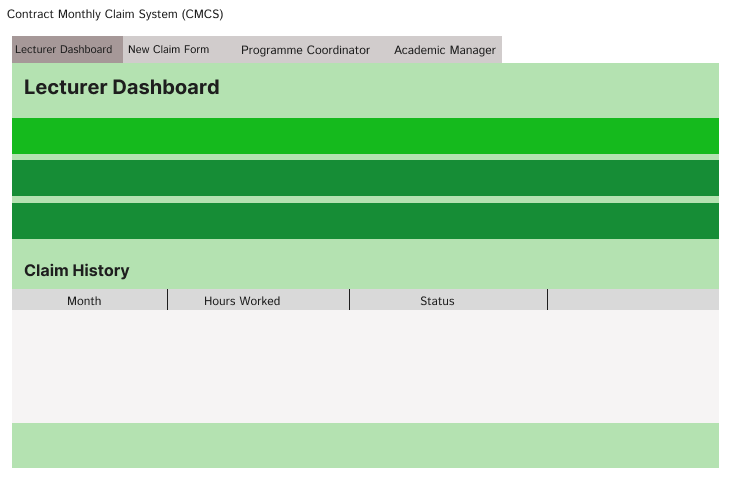
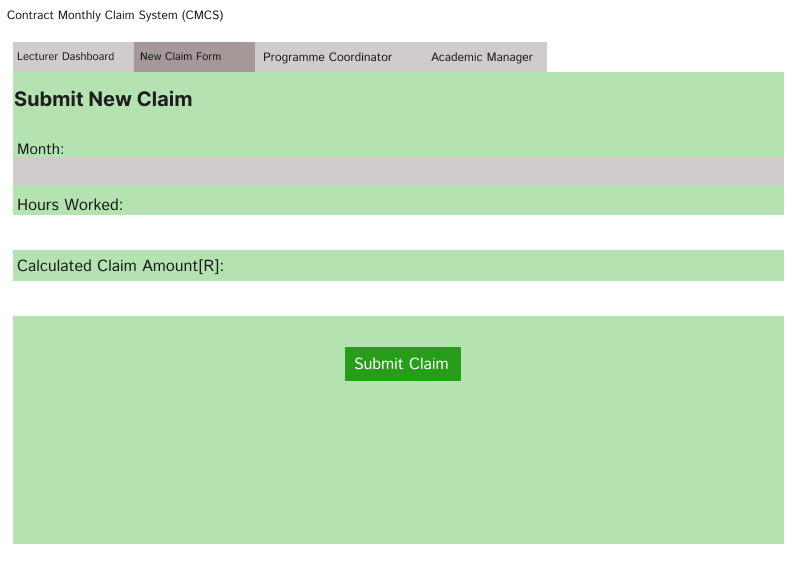
**Lecturer Dashboard:** Shows claim history and option to submit a new claim.

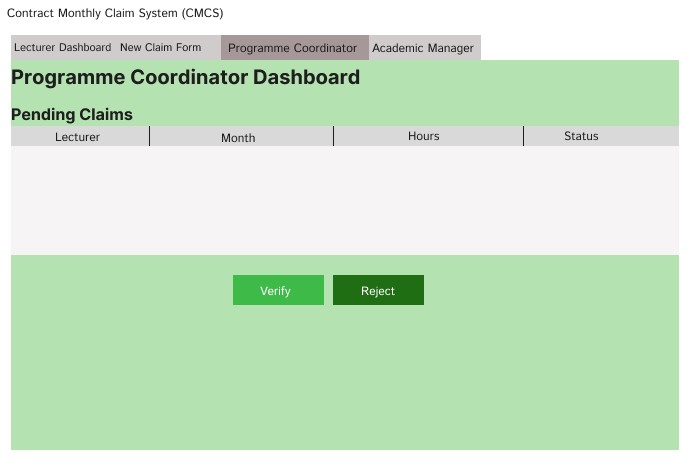
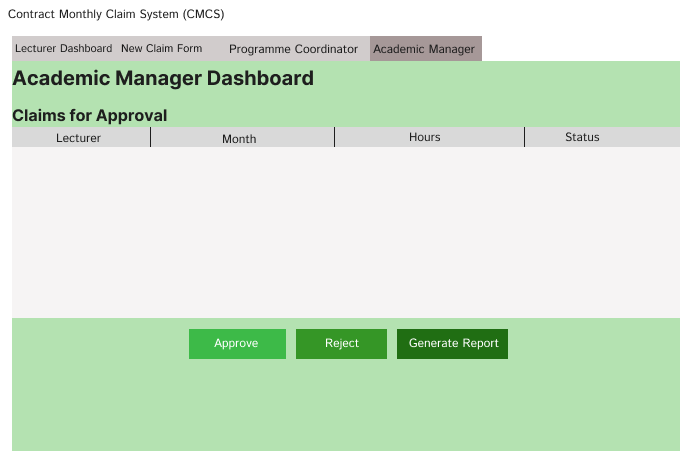
**New Claim Form:** Simple form to capture hours worked and calculate claim amount.

**Claim Details (Lecturer View):** Displays claim information and status.

**Programme Coordinator Dashboard:** Shows pending claims for verification.

**Academic Manager Dashboard:** Provides approval functionality and reporting view.





A computer screen with a green and white box

AI-generated content may be incorrect.