PORTFOLIO OF EVIDENCE (POE)

Programming 2B (PROG6212)

Student Name: Liam Hendricks

Student Number: ST10454685

Module Code: PROG6212

Module Name: Programming 2B

Assessment Type: POE Part 1 — Project Planning and Prototype Development

Date: 09 September 2025

# Introduction

The Contract Monthly Claim System (CMCS) is a proposed .NET Core-based application designed to streamline the submission, approval, and tracking of monthly claims for Independent Contractor (IC) lecturers. This document presents Part 1 of the Portfolio of Evidence (POE), which focuses on project planning and prototype development. It covers design documentation, assumptions and constraints, UML class diagram for the database, a detailed project plan, GUI prototype design, and version control practices.

# Documentation: Design Choices and Structure

The system’s architecture is structured around a central relational database and a modular GUI. The database consists of entities such as Lecturers, Claims, Documents, and Approvals. The design ensures that lecturers can submit claims with supporting documents, programme coordinators and managers can verify and approve claims, and all stakeholders can transparently track claim status.

The GUI layout is designed for simplicity and usability. Key screens include:  
1. Login Page — secure access for all users.  
2. Lecturer Dashboard — for claim submission and document uploads.  
3. Coordinator/Manager Dashboard — for reviewing and approving claims.  
4. Claim Tracking Page — for transparent monitoring of claim status.

# Assumptions and Constraints

Assumptions:  
• All lecturers will have institutional accounts and system access.  
• Coordinators and managers review claims within a set timeframe (e.g., 7 days).  
• All data is stored securely in the institution’s database.

Constraints:  
• File uploads limited to .pdf, .docx, and .xlsx formats.  
• System designed for internal institutional use only.  
• Limited scalability for external contractors not registered within the institution.

# UML Class Diagram for Database

The UML class diagram represents the entities and their relationships within the database. It includes the following classes:  
• Lecturer (LecturerID, Name, Email, BankDetails)  
• Claim (ClaimID, LecturerID, HoursWorked, HourlyRate, TotalAmount, Status)  
• Document (DocumentID, ClaimID, FilePath, UploadDate)  
• Approval (ApprovalID, ClaimID, ApprovedBy, Role, DateApproved, Status)  
The relationships ensure that each claim is linked to a lecturer, documents can be attached to claims, and approvals are linked to claims for accountability.

# Project Plan

The project plan outlines tasks, dependencies, and timeline for developing the CMCS prototype. The tasks are spread across a six-week period for effective execution.

Week 1: Requirements gathering, documentation of assumptions, and constraints.  
Week 2: Design UML class diagram for the database.  
Week 3: Create GUI mockups and wireframes using WPF/MVC.  
Week 4: Draft project plan and refine database structure.  
Week 5: Compile Word report and integrate diagrams/screenshots.  
Week 6: Push source code and documentation to GitHub with 5 descriptive commits.

# GUI Prototype Design

The GUI prototype is designed using WPF (Windows Presentation Foundation). It prioritises user-friendliness, with intuitive navigation and consistent styling across pages. The interface includes simple buttons for claim submission, file uploads, and approval workflows. Although not functional at this stage, it provides a visual representation of the intended user experience.

# Version Control Evidence

Version control is maintained through GitHub. The project repository demonstrates at least 5 commits, each with clear and descriptive commit messages such as 'Initial UML diagram added', 'Created GUI wireframes', and 'Updated project plan'. This ensures code traceability and structured development practices.

# Conclusion

Part 1 of the POE establishes a solid foundation for the Contract Monthly Claim System by addressing planning, design, and prototype requirements. The documentation, UML class diagram, project plan, and GUI prototype ensure that future stages (implementation and automation) can be executed effectively.

# References

Sommerville, I. (2016). \*Software Engineering\* (10th ed.). Pearson.  
Pressman, R. S. (2019). \*Software Engineering: A Practitioner's Approach\* (9th ed.). McGraw-Hill.  
Microsoft Docs. (2024). \*.NET Documentation\*. Retrieved from https://learn.microsoft.com/

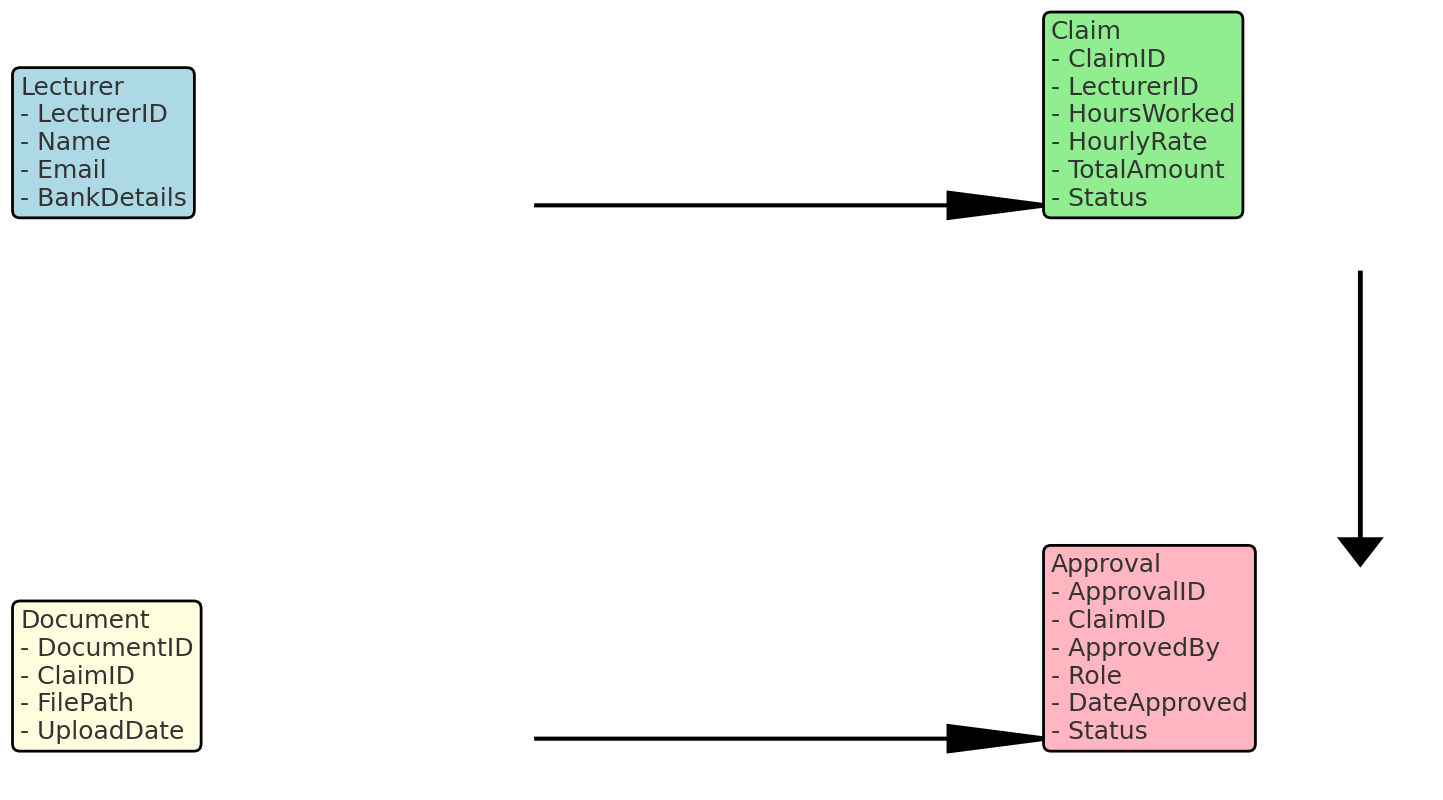


Figure 1: UML Class Diagram for CMCS Database.



Figure 2: GUI Mockup of Lecturer Dashboard.

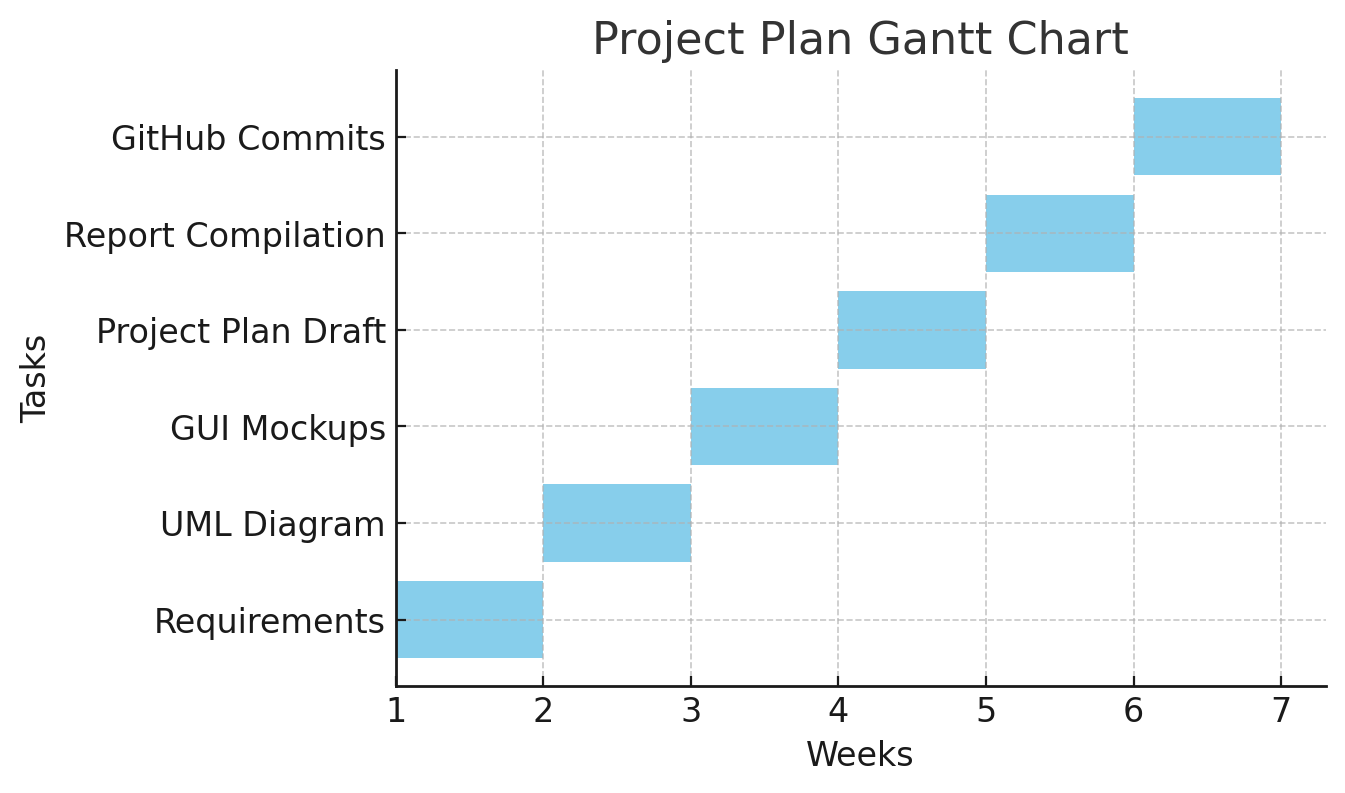


Figure 3: Gantt Chart Project Plan for CMCS Prototype.