

Programming 2B (PROG6212)



Name: Sthobile
Surname: Mkhabela
Student no.: ST10448774
Assignment type: POE Part 1

Table of Contents

Question 1	3
Question 2	5
Question 3	6
Project Plan for Contract Monthly Claim System Prototype.....	6
Question 4	8
AI Declaration.....	9
References.....	10

Table of Figures

Figure 1: UML Class Diagram of Contract Monthly Claim System (CMCS)	5
---	---

Question 1

Design choices

When designing the Contract Monthly Claim System (CMCS), my main goal was to create something that would make the claiming process easier for lecturer while ensuring that programme Coordinators and Academic Managers could review and approve claims in a clear and structured manner. Thus, I made the following choices:

- Made use of a role-based approach
This approach ensures that a user sees the part of the system that is applicable to them depending on the kind of user (lecturer, coordinator, or academic manager) they are.
- The prototype is designed using MVC (Model-View-Controller) design principle using .NET Core
This separates the logic, database, and interface, making the system easier to understand, maintain, and possibly expand in the future.
- Simple application layout
The structure of the system made it easier for the application navigation to be simplified, thus accommodating lecturers and managers since they may not have the same level of technical expertise.

The database for the CMCS is designed to capture the essential information needed for the claim process. At the centre of the database is the lecturer, who can submit multiple Claims. Each claim contains details such as the hours worked, hourly rate, and the total amount. Claims can also have supporting documents, which are files the lecturer uploads to support the work they are claiming for. Once a claim is submitted, it first goes to the programme coordinator, who checks if the details are correct. After that, it is sent to the academic manager for final acceptance or rejection. I also included an audit log table to keep track of every action (like when a claim is submitted, approved, or rejected), which helps with accountability and transparency. Even though the current prototype does not use a real back-end database, having this structure in place is useful for future development.

For the GUI, I designed three main role-based layouts to keep the system simple and focused for each type of user. Lecturer will have a form to submit claims, upload documents, and track progress, which makes their process straightforward. Programme Coordinators can review claims, and then the claims are sent to the Academic Manager. Academic Managers have the final say on approvals and can generate reports, which reflects their higher responsibility.

In terms of assumptions, I assumed that:

- All lecturers, that make use of the system, are independent contractors who are paid based on an hourly rate.
- Users have internet access

- Each user only has one role, meaning that a user cannot be both a lecturer and a programme coordinator/academic manager.
- Information (such as lecturer details) stored in the system is accurate

The constraints, that were identified, are as follows:

- The system will not include a real database or file storage system
- No advanced security measures (like encryption or multi-factor authentication) will be implemented at this stage
- Performance is not a priority at this stage, and thus cannot be tested
- There is no user testing as this is the GUI is non-functional

(Gido, Clements, Baker, Harinarain, & Eresia-eke, 2022)

(The IIE, Chapter 1-Project Management Concepts(Part 1) [INSY6212 ARC], 2014)

Question 2

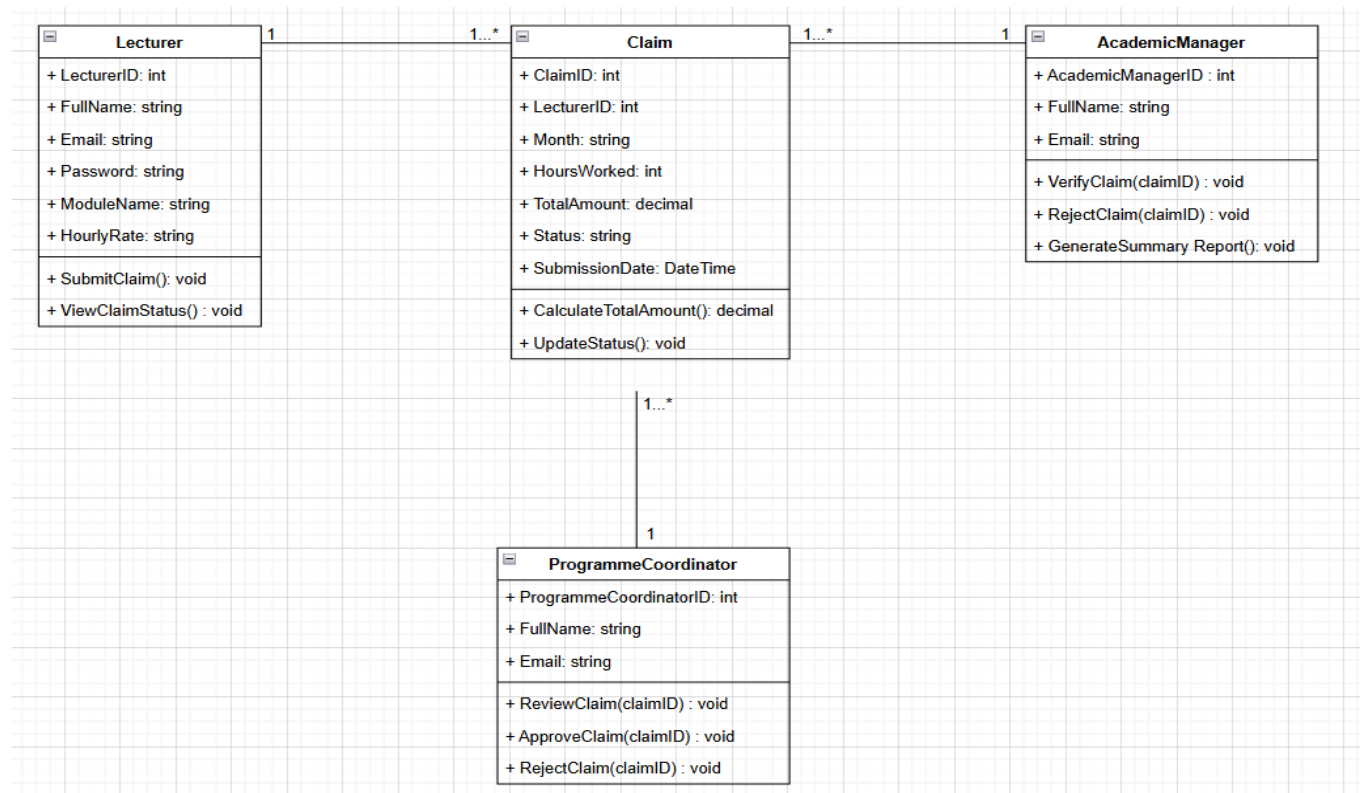


Figure 1: UML Class Diagram of Contract Monthly Claim System (CMCS)

(The IIE, 2012)

Question 3

Project Plan for Contract Monthly Claim System Prototype

Project Duration: 1 July to 20 September (2.5 months)

Project Objective: To design and develop a non-functional GUI prototype for the CMCS that allows lecturers too submit claims, and for coordinators and managers to review and approve them.

Tasks

1. Project Initiation

- Define objectives and scope
- Identify stakeholders
- Create the project schedule

Deliverable: Project Charter and Project Schedule

2. Requirements Gathering

- Meet up with stakeholders (Lecturers, Coordinators, Managers)
- Document functional and non-functional requirements
- Define system constraints and success criteria

Deliverable: Requirements Specification Document

3. System Analysis

- Analyse requirements for feasibility
- Draw up workflows for how claims will be submitted, verified and approved
- Develop use case description
- Identify risks and dependencies

Deliverable: System Analysis Report and Use Case Descriptions

4. System Design

- Create UML Diagrams
- Conceptual Database schema design
- Develop GUI mock-up – MVC (.NET Core)

Deliverable: System Design Document (UML, Database Schema, GUI Mock-ups)

5. Prototype Development

- Build non-functional GUI
- Implement front-end navigation between pages
- Integrate placeholder data

Deliverable: GUI Prototype (with navigation and sample data)

6. Testing and refinement

- Reviewing of UI
- Fix design inconsistencies (If there are any)

- Ensure prototype aligns with requirements

Deliverables: Testing Feedback Report and Updated Prototype

7. Documentation and submission Prep

- Write a project report
- Prepare demo
- Finalise submission package

Deliverable: Final Project Report, Demo Slides and Submission Package

Timeline and dependencies

Phase	Dates	Duration	Dependency
Project Initiation	1 July – 3 July	3 days	
Requirements Gathering	4 July – 10 July	1 week	None
System Analysis	11 July – 17 July	1 week	Requirements Gathering
System design	18 July – 31 July	2 weeks	System Analysis
Prototypes development	1 August – 28 August	4 weeks	System Design
Testing and refinement	29 August – 11 September	2 weeks	Prototype Development
Documentation and submission Prep	12 September – 20 September	1 week	Testing and Refinement

(Gido, Clements, Baker, Harinarain, & Eresia-eke, 2022)

(The IIE, Chapter 1-Project Management Concepts(Part 1) [INSY6212 ARC], 2014)

Question 4

GitHub Repository link:

https://github.com/MkhabelaSthobile/ST10448774_PROG6212_POEPart1.git

(OpenAI, 2025)

Note: AI was used for this question

AI Declaration

For PROG6212 POE PART 1, I have used AI in the following ways:

- For the layout and interface of the system
- For guidance on how to ensure that the GUI is non-functional (in terms of the Views)

References

- Gido, J., Clements, J., Baker, R., Harinarain, N., & Eresia-eke, C. 2022. *Successful Project Management in South Africa*. 2nd ed. London: Cengage.
- OpenAI. 2025. *ChatGPT*, GPT-5. [App]. Available at: <https://chat.openai.com/> [Accessed 12 September 2025]
- The IIE. 2012. Information System [INSY6112 Module Manual]. *The Independent Institute of Education: Unpublished*.
- The IIE. 2014. Chapter 1-Project Management Concepts(Part 1) [INSY6212 ARC]. The Independent Institute of Education:Unpublished.
- Whova. 2025. *A Complete Guide To Successful Event Project Management*. Available on Whova at: https://whova.com/blog/event-project-management-guide/?utm_source=chatgpt.com [Accessed 15 August 2025]