POE-PART 1

Xola Mgangxela-ST10439697

PROG6212

9/17/25

Table of Contents

Documentation

Design Choice

My design choice is Model-View-Controller (MVC) for the CMCS prototype because it enables a clear separation of the data, the user interface, and the control logic. This renders the system more maintainable, testable, and easier to extend in the future.

Database Structure

The database is structured around the core entities and relationships required for the claim submission and approval process. The database will consist of 5 main entities, which are the following:

1. Lecturers' Table
   This table will store the personal and professional information of each Independent Contractor lecturer. Like the LecturerID, Name, Surname, Email, Department, and hourly rate. Each lecturer will be linked to multiple claims uploaded by the lecturer.
2. Claims Table
   This table will store details of each claim submitted by a lecturer. Key attributes include ClaimID, LecturerID, HoursWorked, HourlyRate, TotalAmount, DateSubmitted, and ClaimStatus. Each claim is associated with one lecturer but may have multiple supporting documents.
3. Supporting Documents Table
   This table basically manages all the files uploaded by lecturers to justify their claims; the attributes include DocumentID, ClaimID, FileName, FileType, and FilePath. Each claim can have zero or many supporting documents attached.
4. Manager Table
   This table will store details of Programme Coordinators and Academic Managers responsible for reviewing claims, the attributes include ManagerID, Name, Email, and Role.
5. Approvals Table
   This table serves as a bridge between claims and managers, recording the approval workflow. The attributes include Approval ID, Claim ID, Manager ID, Approval Date, and Approval Status. This structure allows a claim to be reviewed by more than one manager if necessary.

How the database would work is, a lecturer submits a claim, then the claim is stored in the Claims Table and linked to the lecturer in the Lecturers Table. Supporting documents are uploaded and stored in the Supporting Documents Table and linked to the claim. A manager reviews the claim, and then it is recorded in the Approvals Table, which links the claim to the manager in the Managers Table. The status of the claim, which can be either Pending, Approved, or Rejected, is updated in the Claims Table to reflect its progress.

GUI Layout

My GUI design is simple, professional, and user-friendly. The left sidebar for navigation offers a consistent structure through which lecturers can readily gain access to the most critical operations, such as submitting claims, viewing past claims, uploading supporting documents, and logging off. Having icons present with text in the sidebar is to provide the user with clarity and make the interface seem approachable, whereas the red font used for the logout highlights how necessary signing out is after usage.

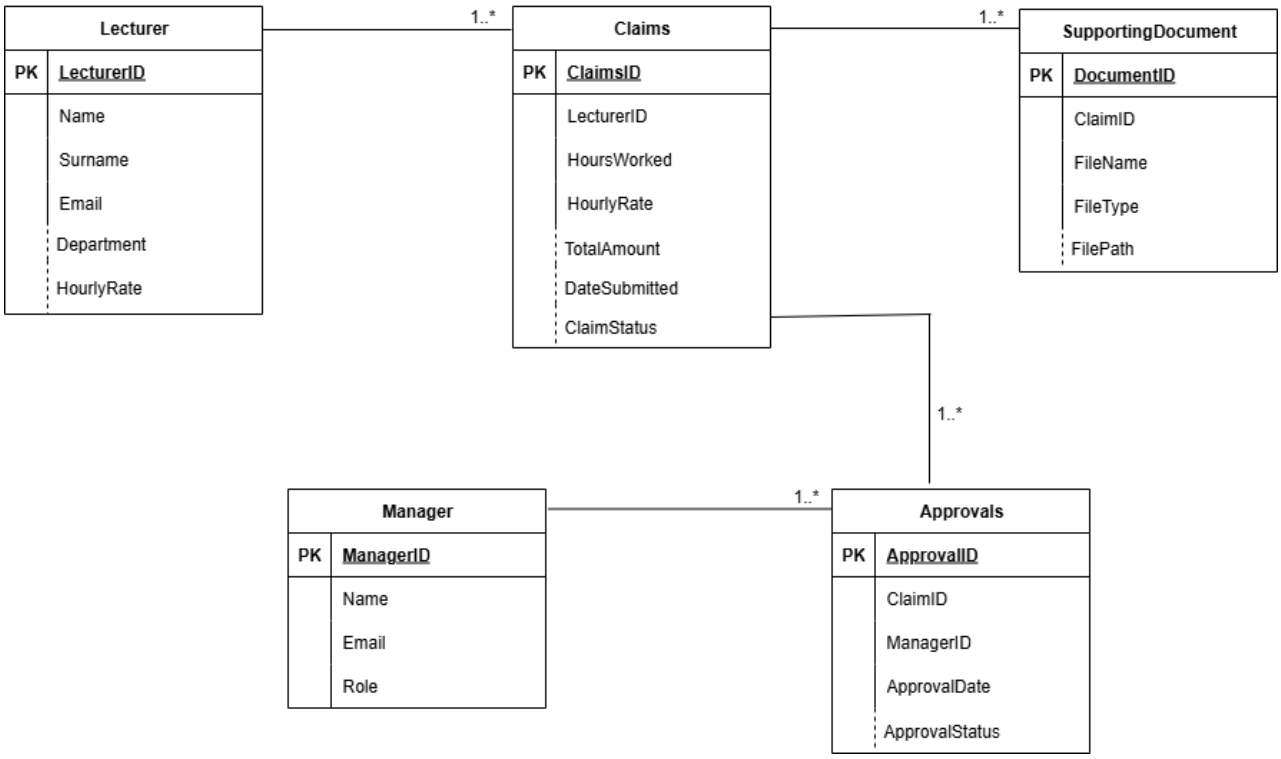
The login page similarly follows a minimalist and clean style format, leaving things in the centre of the screen with proper form labels and a bold blue login button, which contributes to the atmosphere of seriousness and concentration but keeps things readily accessible. Upon logging in, the initial page to display is the dashboard page, which contains three prominent action cards, namely Submit New Claim, View Past Claims, and Upload Supporting Documents, each accompanied by a short description. Such a layout reduces cognitive load by bringing the users directly to the most significant activities straight away, while the interface looks neat and organized.

The overall style is minimalist, with a white background that helps avoid unnecessary distractions. The use of consistent elements reinforces a professional and formal appearance, making it suitable for an academic and administrative system. This design ensures that lecturers can navigate the system efficiently, making it suitable for handling monthly contract claims.

Assumptions and Constraints

Every lecturer may submit multiple claims, but each claim belongs to one lecturer only. Claims must be made based on hours worked, with these being multiplied by the hourly rate of the lecturer. Only a Programme Coordinator or Academic Manager can accept or reject the claim. Verification must be supported by documents connected to a claim. The prototype is just going to show them the design and user interface; no backend function or payment integration will be added here. (Usmani, 2024)

UML Class Diagram for Database



(Satzinger, et al., 2022)

Relationships

The lecturer can submit multiple claims. One Lecturer → Many Claims.

Each claim belongs to one lecturer and may include multiple supporting documents. One Claim → Many Supporting Documents, One Claim → Many Approvals.

Each supporting document is linked to a single claim, but a claim can have many documents.

A manager can review many claims.

Connects Managers to Claims. Each claim can have multiple approvals, and each manager can approve multiple claims.

Project Plan

Phase	Tasks	Dependencies	Timeline
1. Requirement Review	-Looked over the PoE brief and instructions. -Understood what Part 1 required (non-functional GUI only)	None	08-11 Aug 2025
2. Planning and Research	-Planned how to approach the GUI development. -Brainstormed how you wanted the GUI to look. -Picked colours, fonts, and overall styling ideas. -Researched examples on Google for contract monthly claim systems for inspiration.	Phase 1 completion	12-15 Aug 2025
3. Environment and Project Setup	-Created a new MVC project in Visual Studio. -Set up folders for Views, Controllers, and Models. -Prepared the workspace.	Phase 2 completion	16-18 Aug 2025
4. Controller Creation	-Created AccountController, ClaimController, and DocumentController. -Noted HomeController was already present.	Phase 3 completion	19-22 Aug 2025
5. View Folder Setup	-Created view folders for Account, Claim, and Document.	Phase 4 completion	23-25 Aug 2025
6. Razor View Creation	-Account folder: Created Login view -Claim folder: Created History and Submit views -Documents folder: Created Upload view. -Home folder: Used existing Index and Privacy views.	Phase 5 completion	26-30 Aug 2025
7. Layout and Styling Implementation	-Added code for page layout and structure -Applied chosen colours, fonts, font sizes, and icons - Added placeholders -Structured the pages for user-friendliness and consistency	Phase 6 completion	01-08 Sep 2025
8. Review and Final Adjustment	-Complied and reviewed all reviews, all views, and navigation. -Checked consistency of design and placeholders. -Captured screenshots for documentation. -Prepared files for submission.	Phase 7 completion	09-13 Sep 2025
9. Submission Prep	-Packaged the non-functional GUI for submission. -Completed final PoE documentation for Part 1.	Phase 8 completion	14-16 Sep 2025

(Oui, 2025)

GitHub Link

https://github.com/MgangxelaXola-ST10439697/POE-Part1_PROG6212.git

AI Declaration

Throughout my writing in this assignment, I have used ChatGPT to help explain the questions in depth and what is required of me for Part 1. I used AI to help me write and explain the GUI layout of my system. This helped me articulate the reasoning behind my design choices more clearly and ensured my documentation was detailed and professional. I also used ChatGPT to help me with the attributes for my database design.

References

Oui, T., 2025. *Planning-INSY6212.* Joburg: ARC Student Management System.

Satzinger, J. W., Jackson, R. B. & Burd, S. D., 2022. The Domain Model Class Diagram. In: *Systems Analysis and Design.* USA: Cengage Learning, pp. 103-113.

Usmani, F., 2024. *Assumption and Constraints in Project Management.* [Online] Available at: https://pmstudycircle.com/assumptions-and-constraints-in-project-management/ [Accessed 14 Sep 2025].