Programming 2B PROG6212

POE PART 1 Report

1. Documentation

A) ***Design Decisions***

The Contract Monthly Claim System (CMCS) aims to automate and streamline claim submission processing for independent contractor lecturers. The system provides a step-by-step workflow that enables lecturers to submit claims, attach supporting documents, and track the progress of their approval. This obviates paper work manually, brings in transparency, and speeds up claim processing.

As far as system architecture is concerned, the project employs .NET Core MVC as the primary framework. This was the decision due to its scalability, maintainability, and adherence to industry standards. .NET Core MVC provides for modular development, strict separation of concerns, and native integration with relational databases. Otherwise, in the event of a desktop solution being required, Windows Presentation Foundation (WPF) has also been considered. WPF provides a rich GUI experience but does not provide portability on par with that of an MVC system based on the web. The utilization of .NET Core MVC therefore allows for broader availability and longer-term upkeep.

With regard to database structure, a relational database (SQL Server) was employed. SQL Server supports high data storage consistency, provides normalization to reduce redundancy, and provides reliable query performance. This choice guarantees lecturer claims, approvals, and related documents are safely stored and retrievable effectively.

The GUI structure has been put together taking user experience into account. With direct access to essential functions like Upload Documents, View Status, and Submit Claim, the dashboard serves as the initial point of access. Important details like the lecturer's ID, hours worked, hourly rate, and the uploading of supporting documentation are gathered by the claim form. The Approval Screen supports role-based processing, allowing Programme Coordinators and Academic Managers to view and approve claims with efficiency.

B) ***Assumptions***

The CMCS structure is based on several assumptions that guide its functioning. Each lecturer is assumed to have a unique identification number to facilitate proper claim tracing and avoidance of duplication. Hourly rates are fixed in advance, either lecturer by lecturer or by the nature of the contract. This prevents conflicts during claim processing. Claims are assumed to follow a monthly cycle with one claim being processed per lecturer per month. Finally, authentication methods such as login are assumed but not included within the scope at the prototype level.

C) ***Constraints***

The system operates under some constraints as well. The system allows only a single active claim submission for each lecturer in one month, keeping it in line with the policies of administration. The file upload for attachments is capped at 5MB per file, finding a balance between efficiency in storage and use. Finally, the system has to run efficiently on a Windows platform, as per the .NET Core and WPF development frameworks.

**Conclusion**

Through its combination of structured workflows, database dependability, and user-friendly interfaces, the CMCS has been thoughtfully designed to improve efficiency in handling lecturer claims. The system addresses realistic assumptions and constraints while achieving scalability, consistency, and maintainability through the use of SQL Server and.NET Core MVC. The fundamental design choices needed to create a workable and long-lasting prototype that satisfies institutional requirements are reflected in this documentation.

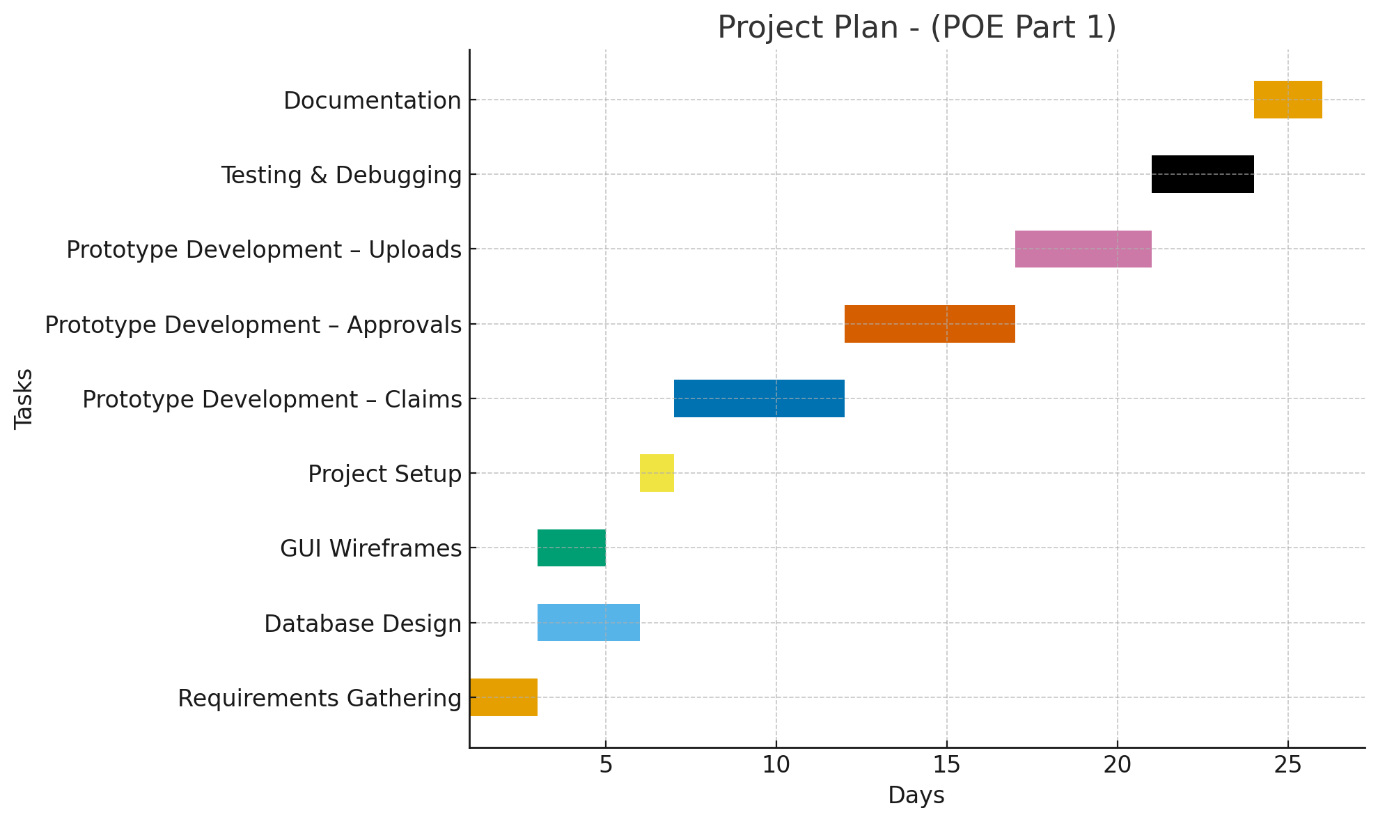
1. UML Class Diagram

A diagram of a computer flow

Description automatically generated with medium confidence

1. Project Plan

**(Gantt Chart)**



**(TABLE)**

|  |  |  |  |
| --- | --- | --- | --- |
| **TASK** | **DESCRIPTION** | **DURATION** | **DEPENDENCY** |
| Requirement Gathering | Identify claim process, roles,and data requirements | 2 days | None |
| Database Design | Design UML class diagram, create schema | 3 days | Requirement Gathering |
| GUI Wireframes | Draw GUI layouts (dashboard,claim,form,approval screen ) | 2 days | Requirement Gathering |
| Project Setup | Configure.NET core MVC project | 1 day | Database & Wireframes |
| Prototype Development -Claims | Create forms for claim submission & display | 5 days | Project Setup |
| Prototype Development-Approvals | Add approval interface and workflow logic | 5 days | Claims Development |
| Prototype Development- Uploads | Integrate document upload functionality | 4 days | Approvals Development |
| Testing & Debuging | Validate prototype,fix issues | 3 days | All Development Tasks |
| Documentation | Write design choices,user guide,assumptions | 2 days | All Tasks |

1. GUI

**A screenshot of a computer

Description automatically generated**