**1. Design Overview**

The Contract Monthly Claim System (CMCS) is designed to streamline the process of submitting, reviewing, and approving monthly claims made by independent contractor lecturers. The system provides two main user interfaces:

**Lecturer Interface:** Allows lecturers to submit claims, upload supporting documents, and view the status of their claims.

**Admin Interface:** Allows Programme Coordinators and Academic Managers to review and approve/reject submitted claims.

The system is designed to be user-friendly, with clearly defined actions and intuitive navigation. This prototype focuses on the front-end layout without implementing the full back-end functionality, but it simulates the expected interactions between the users and the system.

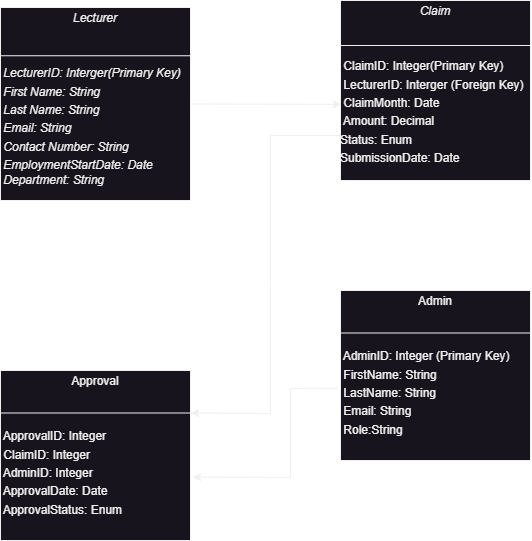
**Key Design Principles:**

**Separation of Concerns:** The system is built with distinct roles and responsibilities for lecturers and admins. Each has their own dedicated interface to perform their respective tasks.

**Consistency:** The design ensures that all users see reliable and consistent information. For instance, lecturers can view their submitted claims, while admins can track all pending claims.

**Simplicity:** The interfaces are simple and intuitive, with minimal steps required to perform key tasks such as submitting claims or approving them.

**The UML Class Diagram**



**Project Plan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Phase | Tasks | Duration | Start | End |
| 1. Requirements Gathering | 1.2 & 1.3 | 1 week | Week 1 | Week 1 |
| 1. UML & Database Design | 2.2 & 2.3 | 2 days | Week 2 | End Week 2 |
| 1. GUI Prototyping (WPF/MVC) | 3.1 , 3.2 & 3.3 | 2 weeks | Mid-Week 3 | Week 5 |
| 1. Documentation & Final Review | 4.1 | 1 week | Week 6 | Week 6 |

**KEYS:**

Task 1.1: Identify and gather requirements from stakeholders (lecturers, admins, etc.).

Task 1.2: Analyze the requirements and identify system constraints and assumptions.

Task 1.3: Initial brainstorming and selection of technology stack (e.g., .NET Core, SQL Server).

Task 2.1: Design the UML Class Diagram (lecturer, claim, admin, approval).

Task 2.2: Define database schema based on UML class diagram.

Task 2.3: Review and refine the class diagram and database schema with stakeholders.

Task 3.1: Create initial wireframes/mockups for both Lecturer and Admin UIs.

Task 3.2: Implement the Lecturer dashboard (submission form, claim history).

Task 3.3: Implement the Admin dashboard (claim approval interface, claim overview).

Task 4.1: Create technical documentation (UML, database schema, GUI design choices).