1.DOCUMENTATION

DESIGN CHOICES

The GUI for the Contract Monthly Claim System is designed as a user-friendly and intuitive front-end prototype using WPF. The layout is simple and clean, focusing on clear headings and a straightforward button-based menu. A single window is used to present all key functionalities, avoiding the need for complex navigation. The use of distinct button groups for "Lecturer Options" and "Coordinator / Academic Manager Options" makes it immediately clear what actions are available to each user type. For the prototype, button clicks display simple MessageBox alerts to simulate the successful execution of an action without connecting to a real backend.

DATABASE STRUCTURE

The system's data requirements can be fulfilled with a simple, relational database model. A **UML Class Diagram** below illustrates the proposed structure, which would be implemented as three main tables:

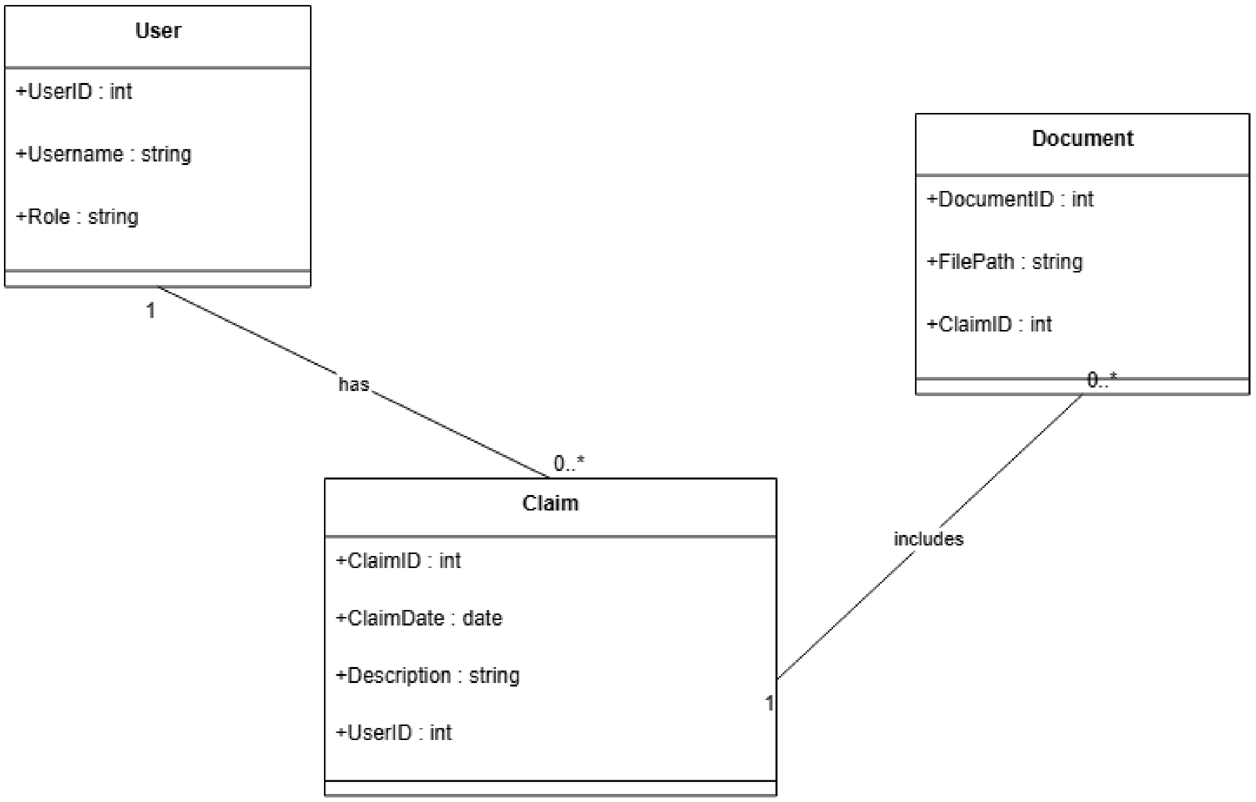
- **Users Table:** Stores user information, including a unique identifier (UserID), Username, and PasswordHash for security. A Role field is included to differentiate between lecturers and coordinators/academic managers.
- **Claims Table:** Serves as the central repository for all submitted claims. It is linked to the Users table via the UserID foreign key. Key attributes include ClaimID (primary key), DateSubmitted, and ClaimStatus (e.g., 'Pending', 'Verified', 'Approved').
- **Documents Table:** Stores metadata for supporting documents uploaded with each claim. It is linked to the Claims table via a ClaimID foreign key, allowing a single claim to have multiple associated documents.

ASSUMPTIONS AND CONSTRAINTS

- **Frontend-only:** This is a prototype and does not connect to a live database or perform actual data transactions.

- **Simple Logic:** All backend functionality (e.g., submitting, verifying, and approving claims) is simulated with basic MessageBox displays.
- **Role-based Access:** The design assumes that user roles (lecturer vs. manager) are predefined and simple. A future, more robust implementation would include a login screen to manage access based on these roles.
- **Static UI:** The current UI is static and does not dynamically update based on user roles or claim status.

2.UML CLASS DIAGRAM



3.PROJECT PLAN

Phase	**Task**	**Dependencies**	**Duration**
Phase 1: Design	• Define Requirements and Features. • Create GUI Wireframes. • Design Database Schema (UML).	• N/A • Requirements • Requirements	• 1-2 days • 1 day • 1 day
Phase 2: Development	• Set UP the WPF Project.	• GUI Wireframes,	• 1 day • 2 days

	• Implement Main Window UI (xaml). • Implement C# Code Behind. • Connect UI to Logic (Event Handlers).	Database Schema • GUI Wireframes • Main Window UI • UI & C# Code	• 3 days • 1 day
Phase 3: Testing & Review	• Test All Button Functionality. • Review and Refine GUI	• All UI and logic implemented • Testing Results	• 1-2 days • 1 day
Total Estimated Time			2 weeks