Documentation

### Design Rationale Report: Contract Monthly Claim System

## **Introduction**

The Contract Monthly Claim System (CMCS) is designed to streamline the monthly claim submission process for Independent Contractor lecturers. This report outlines the design choices, database structure, and GUI layout for the prototype, demonstrating a thoughtful approach to meeting the system requirements while ensuring usability, security, and maintainability.

## **Design Choices**

I selected the Model-View-Controller (MVC) architecture with .NET Core for several strategic reasons. This pattern provides clear separation of concerns, which enhances maintainability and allows parallel development of components. The MVC framework's built-in support for dependency injection promotes testability, crucial for validating the complex calculation requirements of the system. I chose SQL Server as the database management system for its robust transactional support and integration capabilities with the .NET ecosystem, ensuring data integrity throughout the claim approval workflow.

The design incorporates role-based access control with three distinct user roles: Lecturers, Programme Coordinators, and Academic Managers. Each role has tailored interfaces and permissions that reflect their responsibilities in the claim process. This approach enhances security while providing an intuitive experience for each user type.

## **Database Structure**

The database schema consists of six core tables designed with normalization principles to eliminate redundancy. The Users table serves as the central authentication hub, with relationships to the Lecturers table that stores contractor-specific information. The Claims table acts as the primary container for monthly submissions, linked to ClaimItems that detail individual work entries. This relationship allows flexible tracking of multiple work items per claim while maintaining referential integrity.

The Approvals table implements the multi-level review process, capturing each stage of the approval workflow with timestamps and comments. Finally, the Documents table manages supporting materials with appropriate metadata and storage references. This structure supports efficient querying for reporting needs while maintaining comprehensive audit trails of all claim-related activities.

## **GUI Layout**

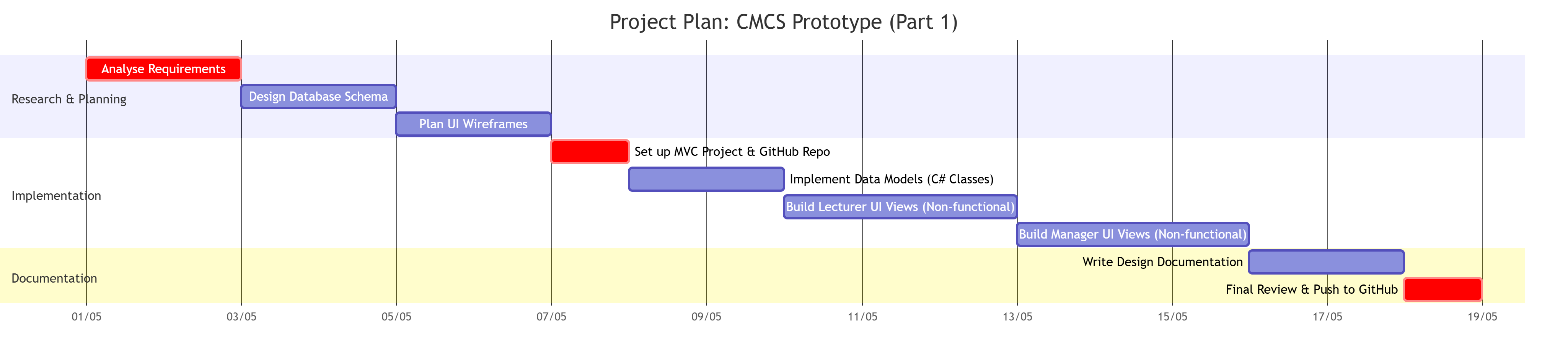
The user interface follows a responsive, role-adaptive design that prioritizes intuitive navigation. Lecturers encounter a dashboard highlighting claim status and quick submission options, with a form-based interface for entering work items that includes real-time calculation of totals. The interface includes validation cues and helpful tooltips to prevent submission errors.

Approvers (Programmer Coordinators and Academic Managers) receive a task-oriented interface focused on their review responsibilities. Their dashboard presents claims requiring attention in priority order, with clear visual indicators of status and aging. The review interface provides consolidated claim information with approver commentary fields and simple approval/rejection controls.

All interfaces maintain consistent navigation patterns and visual design language across roles, reducing cognitive load for users who may interact with multiple parts of the system. The layout uses a clean, professional aesthetic with careful attention to information hierarchy, ensuring important data and actions remain prominent while secondary information is accessible but not distracting.

## **Conclusion**

The CMCS prototype design balances functional requirements with usability considerations, creating a foundation for a system that will genuinely streamline the claim process. The architecture supports future expansion while maintaining performance and security standards expected in enterprise applications.



UML DIAGRAM

