Overall System Architecture

The CMCS is designed as a web-based application using ASP.NET Core MVC framework. This architecture was chosen for its scalability, separation of concerns, and ability to create responsive web interfaces. The system consists of three main layers:

- Presentation Layer: MVC views for user interaction
- Business Logic Layer: Controllers and services handling data processing and business rules
- Data Access Layer: Models and database context for data persistence

Database Design Choices and Rationale

The CMCS database is designed to efficiently store and manage key data entities, including Lecturers, Claims, Approver, and Document. The database adheres to a normalized structure to avoid redundancy and ensure scalability as more lecturers and claims are added to the system. I opted for a relational database using Entity Framework Core with SQL Server. Key entities include:

+ Lecturers
    - LecturerID (Primary Key)
    - Name
    - Email
    - Department: The department or faculty the lecturer belongs to.
    - SubmittedClaims:
    - *A one-to-many relationship with the Claim entity, as each lecturer can submit multiple claims*.
+ Claim Entity
    - ClaimID (Primary Key)
    - LecturerID (Foreign Key): Links to the Lecturer entity, identifying which lecturer submitted the claim.
    - ApproverID (Foreign Key): Links to the Approver entity, identifying which Programme Coordinator or Academic Manager approved or rejected the claim.
    - DateSubmitted
    - DateApproved
    - Status: The status of the claim (e.g., Submitted, Pending, Approved, Rejected).
    - TotalAmount: The total amount being claimed by the lecturer.
    - SupportingDocuments

        - *A one-to-many relationship with the Document entity, as each claim can have multiple supporting documents attached.*
- Approver Entity
    - ApproverID (Primary Key)
    - Name
    - Email
    - Role: The role of the approver, which can be either "Programme Coordinator" or "Academic Manager".
    - ApprovedClaims
- Document Entity
    - DocumentID (Primary Key)
    - ClaimID (Foreign Key) - Links to the Claim entity.
    - FilePath
    - UploadDate

This design allows for efficient data retrieval and maintains data integrity through relationships.

GUI Layout and User Interaction Flow

The GUI is designed to be intuitive and user-friendly, with a responsive layout suitable for both desktop and mobile devices. Key interfaces include:

- Dashboard: Personalized view for each user type
- Claim Submission Form: For lecturers to input claim details and upload documents
- Approval Interface: For coordinators and managers to review and process claims
- Claim Status Tracker: For all users to monitor claim progress

The user flow is designed to minimize clicks and provide clear navigation between different application sections.

Assumptions Made During the Design Process

- Reliable internet access for all users
- Moderate concurrent users (up to 1000)
- Limited document upload formats (PDF, DOC, JPEG)
- Standard approval workflow

Constraints and Limitations

- Web browser-based, not optimized for mobile apps
- No real-time notifications in prototype

Design Choices Rationale

- ASP.NET Core MVC: Offers scalability and clear separation of concerns.
- Relational Database: Ensures data integrity and efficient querying.
- Responsive Design: Accommodates various devices and screen sizes.
- Role-based Access: Enhances security and user experience.

Note: *There is room for future enhancements based evolving requirements during coding.*