**Introduction**

The report outlines the prototype development of the Contract Monthly Claim System (CMCS) as a .NET Core web application for independent contractor lecturer approval and monthly claim submission. The prototype includes non-functional aspects, for example, database design using UML class diagram, project plan, and GUI layout using ASP.NET Core MVC. The system aims to enhance efficiency since it allows lecturers to make claims, upload supporting documents and check status, and the aforementioned are accredited and certified by coordinators and managers.

**1. Documentation: Design Decisions, Database Schema, and GUI Design**

**Design Decisions**

I have employed ASP.NET Core MVC for GUI because it has a proven pattern for developing web applications with separation of concern (Model-View-Controller) where it is simple to maintain and scale. MVC supports clean routing, razor views for dynamic UI, and non-obtrusive integration with Entity Framework for database access. For the database, I chose a relational model with SQL Server to be able to handle inter-entity relationships like users, claims, and documents efficiently. User interface choices are made with simplicity in mind: Bootstrap and responsive design for mobile compatibility, business-like color scheme (blues and grays), and simple navigation to minimize user errors.

**Database Structure**

Database is centered on main entities: Lecturer (user info), Claim (submission details like hours, rate, and status), Document (documents), and Approver (managers/coordinators). There are one-to-many (approver claims) and many-to-one (one lecturer can have multiple claims) relationships. This is for data consistency and status query tracking. Assumptions: All users are authenticated using ASP.NET Identity; claims are computed as (hours \* rate). Constraints: Limited 5MB file uploading; no real-time notifications in-prototype; local development environment over cloud hosting assumed.