**Documentation**

**Design Choices and Rationale**

The development of the Contract Monthly Claim System (CMCS) prototype will be based on the ASP.NET Core MVC (Model-View-Controller) framework. This choice supports the requirement for a web-based application which fits with MVC's architecture. MVC separates the system into three layers, dividing the application into Model (data and business rules), View (user interface), and Controller (user interaction handling) This promotes modularity, maintainability, and scalability for future functionality (Esposito & Saltarello, 2018).

The web-based deployment of the system allows universal accessibility for programme coordinators, academic managers and lecturers, irrespective of device or operating system. This removes the need for specific software installations, accelerated user adoption, and consistent user experience across multiple platforms (Microsoft, 2023).

**Database Structure**

The database is structured to manage and track claims with precision while preserving data integrity. The design centres around a few key entities, primary entities include:

**User:**

This will be used as a base table for all user types, it will have information like UserID, Name, Email, and a UserType attributed to differentiate between Lecturers, ProgrammeCoordinators, and AcademicManagers.

**Claim:**

This entity will store all claim-related data this will include ClaimID, a foreign key to the Lecturer who submitted it, the DateSubmitted, and current Status (e.g., Pending, Approved, Rejected).

**ClaimLineItem:**

Provides a detailed breakdown of individual claim entries, linked to the Claim table. Attributes include DateWorked, Hours, and a Description of work done.

**SupportingDocument:**

This Manages uploaded files attached to claims, storing DocumentID, FileName, and a reference to the associated Claim.

This structure supports the submission of multiple claims per lecturer, each with multiple line items and supporting documents, this will still maintain traceability, and an audit trail for accountability (Elmasri & Navathe, 2020).

**GUI Layout and User Experience // complete**

The graphical user interface (GUI) will be designed to be intuitive and user centric. Navigation will be simple, with the layout being consistent across all pages. Prototype will include the following views:

**Login Page:**

A secure straightforward entry point for all users.

**Lecturer Dashboard:**

When logged in, lecturers will see a summary of their submitted claims, status, and button to initiate a new claim submission.

**Claim Submission Form:**

This form will guide the user to enter required information, this will include hours, activities and to upload supporting documents with a clear drag-and-drop or file picker tab.

**Admin/Manager Dashboard:**

Displays pending claims for review, with action buttons for Approve, Reject, or Request More Information, showing a transparent review process. // complete

**Assumptions and Constraints**

The prototype is a non-functional visual representation. The focus is on design and planning stage of development, with the functionality to be brought in the next stage. The project is restricted to the .NET Core ecosystem due to the requirements. Data validation and authentication is to be handled by the application logic in later stages of development. The system will assume that lecturers are responsible for time and work records and claims will be scrutinised before approval. (Microsoft, 2023)

References

Nielsen, J., 2012. Usability 101: Introduction to Usability. [online] Nielsen Norman Group. Available at: https://www.nngroup.com/articles/usability-101-introduction-to-usability/ [Accessed 20 September 2025].

Esposito, D. & Saltarello, A., 2018. Modern Web Development: Understanding Domains, Technologies, and User Experience. 2nd ed. Microsoft Press

Elmasri, R. & Navathe, S.B., 2020. Fundamentals of Database Systems. 7th ed. Harlow: Pearson.

Microsoft, 2023. ASP.NET Core MVC Overview. [online] Microsoft Docs. Available at: https://learn.microsoft.com/en-us/aspnet/core/mvc/overview [Accessed 20 September 2025].