

## Contract Monthly Claim System (CMCS)

### 1. Introduction

This document provides the full documentation for the prototype development of the Contract Monthly Claim System (CMCS). The CMCS is a web-based application aimed at simplifying and automating the submission and approval of monthly claims for Independent Contractor (IC) lecturers. The system allows ICs to submit claims based on hours worked, which are then verified and approved by Programme Coordinators and Academic Managers. The system also supports document uploads and transparent claim tracking.

This documentation outlines the design choices, assumptions, database structure, and user interface (GUI) layout of the CMCS. The aim is to provide a clear overview of the system at the prototype stage, with functionality to be added in future iterations.

### 2. Design Choices

**The system's database is designed to efficiently manage multiple data entities, including:**

- Lecturer : Stores lecturer information such as name, ID, and hourly rate.
  - Claim: Contains claim details, including hours worked, date, and status (pending, approved, or rejected).
  - Supporting Documents : Stores references to files uploaded as part of the claim submission.
  - Programme Coordinator and Academic Manager : Manage the verification and approval process.
- These entities are structured to ensure modularity and scalability, making it easier to add new features as needed. For example, the separation of `Lecturer` and `Claim` entities ensures that claims can be submitted, processed, and tracked independently while maintaining a strong relationship with lecturer data.

### **GUI Design Choices**

The GUI has been designed to be intuitive and user-friendly. Key design principles include:

- Simplicity : Focus on essential tasks like submitting claims, uploading documents, and tracking claim status.

- Role-based Access : Different user interfaces for lecturers, programme coordinators, and academic managers, ensuring they only see the features relevant to their role.
- Consistency : Uniformity in layout, colours, and font to create a consistent and professional user experience.
- The system uses a clean and modern interface with straightforward navigation. Buttons and forms are clearly labeled to avoid confusion, and the design is responsive to ensure usability across devices.

### 3. Assumptions and Constraints

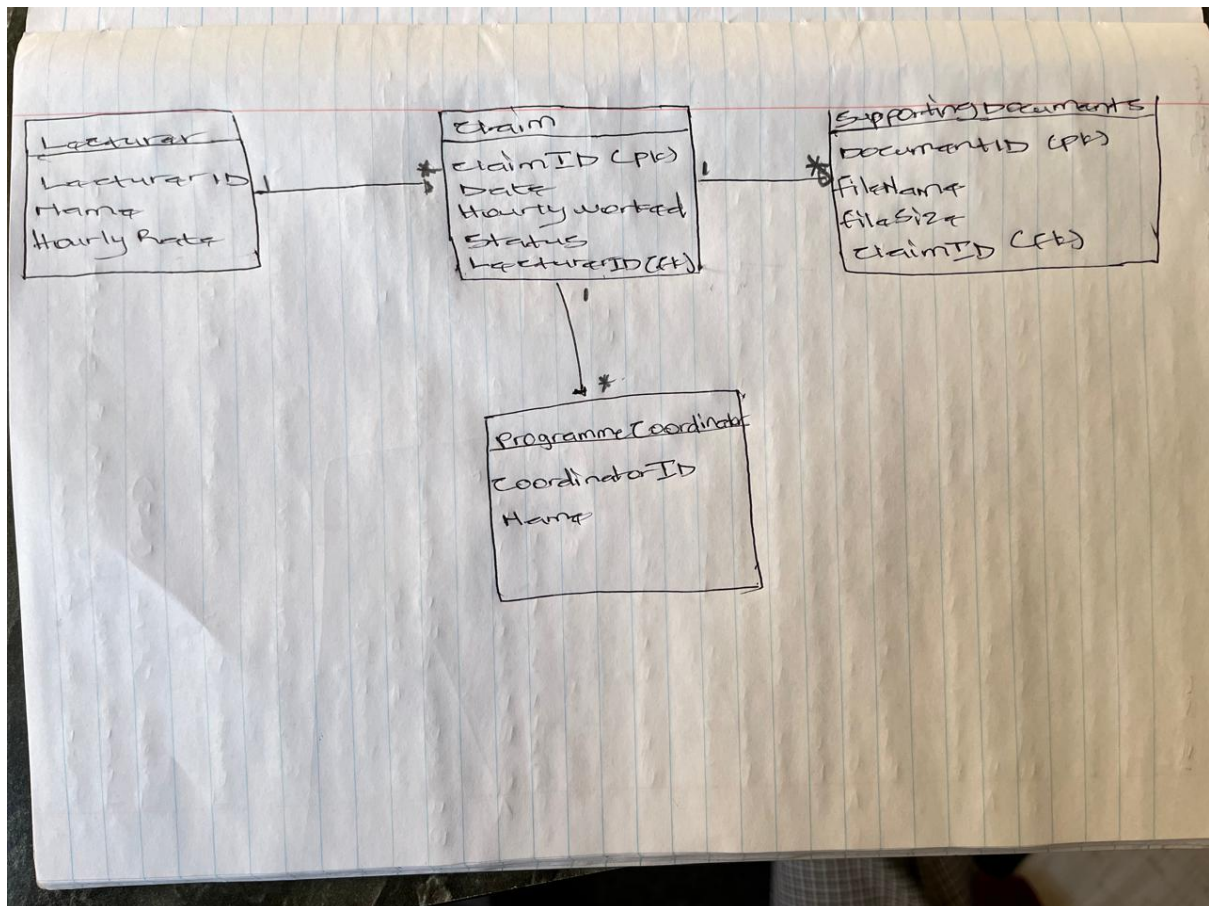
#### **Assumptions**

- Only authenticated lecturers can submit claims.
- Claims can be submitted once a month, with a grace period for late submissions.
- Programme Coordinators and Academic Managers are responsible for approving claims within a defined period.
- Supporting documents (e.g., invoices) are required for claim approval.
- Claims and supporting documents are stored securely and follow the institution's data protection policies.

#### **Constraints**

- File uploads for supporting documents are limited to 10 MB per claim.
- Lecturers are restricted to submitting a claim only once per month to avoid duplicate claims.
- The system is expected to handle concurrent access from multiple users without performance issues.

### 4. UML Class Diagram for Databases



## 5. Project Plan

### Week 1:

- Finalize system requirements and gather input from stakeholders.
- Design the database structure and UML class diagram.

### Week 2:

- Create wireframes for the GUI.
- Develop a front-end prototype using Windows Presentation Foundation (WPF) or MVC in .NET Core.

### Week 3:

- Implement the lecturer submission form.
- Design the interfaces for coordinators and managers to review claims.

### Week 4:

- Write documentation and prepare the final report for submission.

- Perform final touch-ups on the non-functional prototype.

### **Dependencies**

- Database: The design of the database needs to be completed before the GUI is fully developed to ensure proper data representation.
- Frontend Development: The GUI design is dependent on the finalization of wireframes and feedback from stakeholders.

### **Timeline**

Week 1: UML diagram completion.

Week 2: Wireframes and GUI design.

Week 3: Front-end prototype and user roles creation.

Week 4: Final report, GitHub repository setup, and documentation submission.

## **6. GUI Design**

The graphical user interface (GUI) is designed using WPF or MVC in .NET Core. At this stage, the prototype focuses on the visual layout without functional implementation. The primary user interface elements include:

### **Lecturer Interface**

- Form for Claim Submission: Allows lecturers to input the hours worked, select the relevant claim date, and upload supporting documents.
- Submit Button: Initiates the submission of a non-functional claim.
- Claim Status Tracking: Displays the current status of submitted claims, including whether they are pending, approved, or rejected.

### **Coordinator/Manager Interface**

- Claim Review Page: Displays a list of claims that require approval or rejection.
- Approve/Reject Buttons: Allows programme coordinators and managers to process claims.

The GUI is built to be responsive and clean, ensuring a smooth experience across different devices and platforms. Placeholder data will be used for demonstration purposes in the prototype.

## 7. Version Control

All changes to the project are tracked using GitHub for version control. The repository will include five key commits at major milestones:

1. Initial project setup.
2. Completion of the UML class diagram and database structure.
3. Completion of wireframes and GUI layout design.
4. GUI implementation for claim submission and tracking.
5. Final documentation and project wrap-up.

## 8. Conclusion

The Contract Monthly Claim System (CMCS) prototype provides a visual representation of the planned system, focusing on its core elements: claim submission, approval, and document uploads. This prototype lays the groundwork for future iterations where functionality will be added, including backend processing and data persistence. The project demonstrates the importance of careful planning, user-centric design, and database modelling in developing efficient software solutions.