Contract Monthly Claim System (CMCS)

POE Part 3 – Automation & Final Implementation Presentation

Paballo Nthutang – ST10446382

System Overview

- **The CMCS system streamlines lecturer monthly claim processing.**

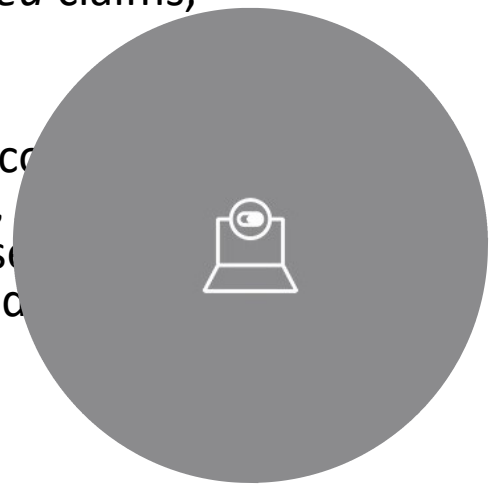
  The Contract Monthly Claim System (CMCS) digitizes and centralizes the entire claim submission workflow. Instead of relying on manual documents or email-based submissions, lecturers can submit claims directly through a user-friendly online interface. This reduces administrative delays, ensures data accuracy, and provides lecturers with a structured and transparent platform for managing their claims.

- **Automates submission, verification, approval, and HR processing.**

  Each role in the institution—Lecturer, Programme Coordinator, Academic Manager, and HR—interacts with an automated workflow. Claims move between workflow stages without manual forwarding. Coordinators and Managers receive structured dashboards showing exactly which claims require action, while HR automatically receives only *final approved* claims, ready for payment processing.

- **Includes real-time status updates and secure document uploads.**

  Using SignalR, changes in claim status are instantly pushed to all co
  users. Lecturers can immediately see whether a claim is *Pending*,
  Rejected, or *CoordinatorApproved*. All uploaded documents are s
  in the server's dedicated directory, ensuring both accessibility and



Lecturer Automation – Submission View

- **Auto-calculation using JavaScript (Hours × Hourly Rate).**

  As lecturers enter their hours worked and hourly rate, JavaScript dynamically calculates the total amount due. This allows users to see the payment they will receive before submitting the claim, reducing errors and providing instant clarity.

- **Validation ensures required fields are filled.**

  ASP.NET Core model validation ensures that all required fields—such as Hours Worked and Hourly Rate—are completed correctly before submission. Errors trigger friendly validation messages directly on the form.

- **Multiple document upload supported (PDF, DOCX, XLSX).**

  Lecturers can upload multiple supporting documents such as timetables, delivery evidence, attendance registers, and contracts. File restrictions ensure only acceptable formats and file sizes are uploaded.

- **Controller stores documents securely in wwwroot/uploads.**

  The server-side controller processes each uploaded file, assigns it a secure random filename, and saves it in a controlled directory. Metadata about the file (original stored name, file size) is stored in the database for auditing and retrieval.

- **SignalR triggers real-time 'Pending' status updates.**

  When a lecturer submits a claim, the system broadcasts a status-change n dashboards. The claim immediately appears in the Coordinator's pending l smooth workflow progression.

Key Code – Auto Calculation Logic

- **JavaScript calculates real-time total:**

  The script listens for changes on the Hours Worked and Hourly Rate fields. Every time a field is updated, the script recomputes the total.

- **total = hours * hourlyRate (updated on input changes).**

  The calculation is simple and efficient, ensuring immediate updates without needing a page refresh or server call.

- **Displayed to user before submission.**

  The updated value appears inside a highlighted Total Amount area, giv
  user helpful instant feedback.

Coordinator Automation

- **Coordinator dashboard shows all pending claims.**

  The interface lists only claims with the status **"Pending"**, making it easy f or Coordinators to identify tasks that need attention.
- **Claims sorted by status = 'Pending’.**

  EF Core queries filter claims so that only those awaiting Coordinator approval are displayed.
- **Coordinator can Approve or Reject.**

  Each claim row includes Approve/Reject action buttons, triggering an update in claim workflow.
- **Approval triggers automatic workflow to 'CoordinatorApproved’.**

  Upon approval, the system updates the database and forwards the claim to the Manager by changing its status.
- **SignalR updates lecturer dashboard instantly.**

  Lecturers immediately receive notifications when Coordinators ta action—no refresh required.

Manager Automation

- **Manager sees claims approved by Coordinator.**

  The Manager dashboard filters for claims with status **"CoordinatorApproved"**, ensuring clear workflow separation.

- **Status must be 'CoordinatorApproved’.**

  Only claims already vetted by the Coordinator appear for managerial review.

- **Approval sets final status to 'Approved’.**

  Once a Manager approves a claim, it becomes eligible for HR processing.

- **Reject sets status to 'Rejected’.**

  Managers may also reject claims; the lecturer is notified ins

HR Automation

- **HR dashboard lists all 'Approved' claims.**

  HR officers see only final-approved claims, ensuring the list is clean and ready for payment action.

- **Excel export using ClosedXML for payment processing.**

  HR can export all approved claims into a formatted Excel spreadsheet for financial reconciliation or external payments.

- **Generates Excel report with hours, rate, totals, and dates.**

  The exported report provides all key data: lecturer name, hours worked, hourly rate, total payment, and submission date.

Excel Export Logic

- **ClosedXML creates worksheets programmatically.**

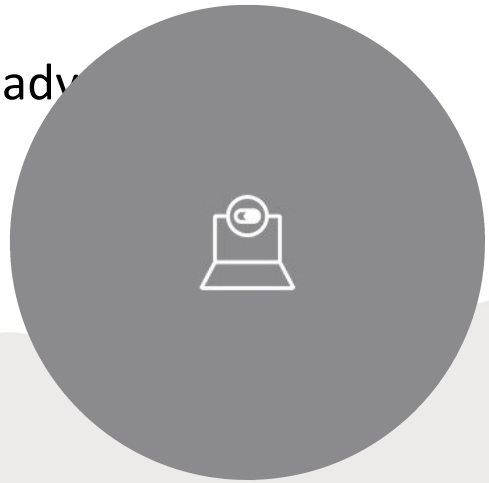
  The HR action method initializes a new Excel workbook in-memory using ClosedXML libraries.

- **One row per approved claim.**

  Each approved claim is written to the worksheet as a line item, creating a structured payment report.

- **Readable, well-formatted payment report for HR.**

  The final exported Excel file is clear, organized, and ready for finance administration.



Real-Time Status Updates (SignalR)

- **StatusChanged event pushed to all clients.**

  Whenever a claim is updated, the server broadcasts the new status to all connected dashboards.
- **Lecturer can see status change instantly (Pending → Approved).**

  The lecturer dashboard dynamically updates UI badges without page reload.
- **Coordinator/Manager dashboards refresh badges dynamically.**

  Workflows progress smoothly as each role gets instant updates.

Security & Data Integrity

- **Identity Roles: Lecturer, Coordinator, AcademicManager, HR.**

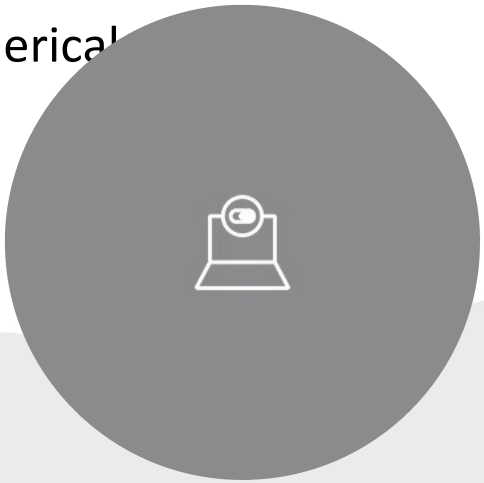
  Role-based authorization ensures that users only access the sections relevant to their duties.

- **File upload restricted by type and size.**

  Only certain extensions (.pdf, .docx, .xlsx) and file sizes are allowed to prevent security risks.

- **Server-side validation ensures accurate and consistent data.**

  ASP.NET model binding and validation attributes enforce numerical accuracy and mandatory fields.



Overall Workflow

1. **Lecturer submits claim (auto-calculated).**
   Auto-total, uploads, and real-time status notifications.
2. **Coordinator reviews & approves.**
   Verification of hours and documentation.
3. **Manager final-approves for HR.**
   Final compliance and correctness check.
4. **HR processes and exports claims.**
   Excel export ensures easy financial processing.
5. **Real-time notifications ensure transparency.**
   SignalR maintains live communication system-wide.

Conclusion

The CMCS system successfully automates the full monthly claim workflow using ASP.NET Core MVC, EF Core, JavaScript, and SignalR. Lecturers benefit from real-time total calculations, validation, and secure document uploads, while Coordinators and Managers use automated status-driven dashboards to verify and approve claims efficiently. HR processing is streamlined through ClosedXML Excel exports that summarize all approved claims. Role-based security, server-side validation, and structured database models ensure data accuracy and system integrity. Overall, the system meets all POE requirements and delivers a fast, transparent, and fully automated claim management process.