Name: Percy Dube

Module: PROG6212

ST10383359

Contract Monthly Claim System (CMCS)
Part 1 UML, Project Plan, and Prototype Report

Design Choices

The system was built using ASP.NET Core MVC to follow best practices and meet module requirements. MVC provides a clear separation of concerns, ensuring scalability as the system evolves.

- **Styling**: Internal CSS and Bootstrap 5 were used for a consistent and responsive UI.
- **Authentication**: A simple session-based login system manages user access.
- **Roles**: The system is role-driven, with Lecturers, Coordinators, and Managers each having distinct dashboards and permissions:
  Lecturers - Submit monthly claims, upload supporting documents, and track claim statuses.
  Coordinators - Verify and approve/reject lecturer claims.
  Managers - Oversee all lecturer claims and make approval/rejection decisions.

These roles influenced both the user interface layout (role-specific navigation, dashboards, and actions) and the data structure (users, claims, and approvals).

Database Structure

Although the prototype currently uses JSON-based persistence instead of a relational database, the system is designed with a logical database structure in mind for scalability (to be implemented in later stages with Entity Framework and SQL).

The key entities include:

- **User** → Stores login details, roles, and basic information.

- **Lecturer** → Includes lecturer name, email, and other staff details (in future versions).
- **Claim** → Records claim submissions with lecturer name, hours worked, hourly rate, total amount, notes, file references, and status (`Pending`, `Approved`, `Rejected`).
- **SupportingDocument** → Uploaded files linked to claims (`.pdf`, `.docx`, `.xlsx`).
- **Approval** → Stores decisions and timestamps when claims are approved/rejected by Coordinators or Managers.

GUI Layout

The prototype includes role-based dashboards with distinct functionality for each user type:

- **Login / Register Pages** - Simple forms for account creation and login, with role selection during registration.
- **Lecturer Dashboard:**
  Welcome page with quick navigation buttons.
  Submit Claim form (with file upload).
  Track Claims page showing history, statuses, and options to delete or clear claims.
- **Coordinator Dashboard:**
  Claims List view with stats (Total, Pending, Approved, Rejected).
  Actions to approve or reject pending claims.
- **Manager Dashboard:**
  All Lecturer Claims table for system-wide visibility.
  Approval and rejection actions for managers.
- **Navigation Bar** - Displays a role-aware Back button, dynamic menu options, and user greeting based on session values.

Each page includes alerts, tables, and Bootstrap components to provide an intuitive and professional interface.

Assumptions and Constraints

It is assumed that all lecturers are Independent Contractors with variable modules and hourly rates. Claims must always include supporting documents, and the Academic Manager and Program Coordinator must approve them in order. Constraints include the

lack of real authentication, data persistence, or business logic, as this phase is limited to front-end design only.

Project Plan

Phase	Tasks	Deliverables	Timeline
1. Planning	Define system scope and user roles (Lecturer, Coordinator, Manager). Identify required pages (Login, Home, Claims, Approvals, Lecturers).	Requirements outline	Week 1
2. Design	Draft GUI wireframes, prepare UML class diagram (database structure for future parts). Decide layout and styling rules.	UML diagram, wireframes	Week 2
3. Prototype Build	Build ASP.NET Core MVC project structure. Implement Razor views (Login, Home, Claims, Approvals, Lecturers). Add internal CSS for layout & styling.	Front-end prototype (non-functional)	Weeks 3–4
4. Testing & Review	Check navigation works across pages. Validate consistency of design. Ensure mock modals/tables reflect real system flow.	UI prototype screenshots + feedback	Week 5
5. Documentation	Write design documentation (choices, assumptions,	PoE Part 1 submission	Week 6

	constraints). Add UML diagram & project plan.		