PROG6212 – Part 1

Contract Monthly Claim System (CMCS)

Contract Monthly Claim System (CMCS) – Prototype Report

1. DOCUMENTATION

The Contract Monthly Claim System (CMCS) is a WPF prototype designed to streamline claim submissions for Independent Contractor (IC) lecturers and manage their approval workflow. The system routes each claim through two review stages where verification by the Programme Coordinator and final approval by the Academic Manager. The design prioritizes efficiency, accuracy, and transparency. At this stage, the prototype is non-functional and focuses only on structure, user flows, and interface layout.

Technology Choice: WPF was selected over MVC for Part 1 because it is suited for desktop-based prototypes, offers strong XAML layout support, and makes it easier to build role-specific interfaces without back-end complexity. MVC may be considered in later iterations if a web deployment is required.

Database Design: The schema centers on **Lecturers, Claims, Documents, Approvals, and Users** (for Coordinators and Managers). Each claim is tied to a lecturer, may include multiple documents, and must progress through structured approval steps. Primary keys ensure unique identification of entities, while foreign keys define relationships. Normalization ensures no redundant data (e.g., lecturer details stored once in Lecturer).

UI/UX Principles: Layouts were designed for clarity and accessibility. Role-based dashboards (Lecturer, Coordinator, Manager) . Navigation is straightforward with a consistent design language across pages. Visual hierarchy is achieved through clear sectioning, consistent typography, and buttons styled for recognizable actions. Accessibility is considered with readable labels, sufficient contrast, and predictable layouts. Consistency across dashboards helps users quickly adapt regardless of role.

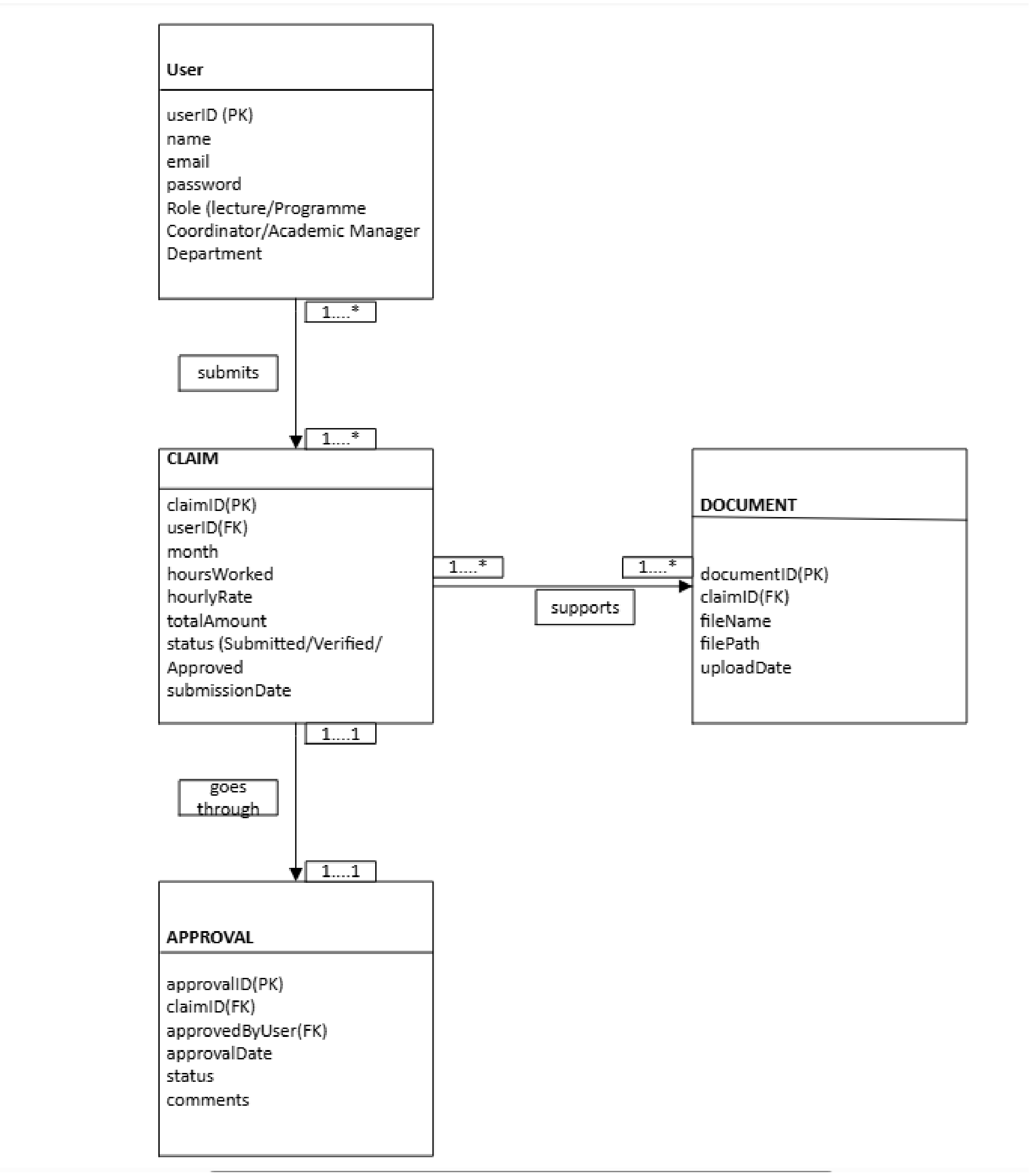
Assumptions & Constraints

- Lecturers are paid hourly, with variable rates depending on their contract.
- A claim is monthly and linked to one lecturer only.
- Supporting documents (PDF, Word, Excel) are uploaded and linked to a claim.

- The POE uses WPF for GUI
- Approvals are strictly ordered: Programme Coordinator then Academic Manager.
- Authentication and back-end logic are excluded for Part 1.
- Constraint: Prototype must not include working data or logic (no functional logic, database binding, or file uploads exist yet).
- Constraint: Prototype focuses only on the graphical layer, no business rules, authentication, or file persistence are implemented.

2. UML CLASS DIAGRAM FOR DATABASES

- Claim: ClaimID, LecturerID, ClaimPeriod, TotalHours, HourlyRate, TotalAmount, Status, Description
- Document: DocumentID, ClaimID, FileName, FilePath, UploadDate, DocumentType
- Approval: ApprovalID, ClaimID, ApproverID, Date, Status, Comments, ApprovalLevel
- User: UserID, Name, Email, Role (Coordinator/Manager), Department



Key relationships:

- Lecturer (1) → (M) Claim
- Claim (1) → (M) Document
- Claim (1) → (M) Approval
- User (1) → (M) Approval

3.Project Plan (Timeline & Dependencies)

- **Week 1** – Requirements analysis, personas, UML draft, wireframes.
- **Week 2** – WPF solution setup, main windows, basic navigation.
- **Week 3** – Build prototype screens Submit Claim, Review, Approval, Status.).
- **Week 4** – Style refinement, polish layouts, final UML.
- **Week 5** – Documentation, screenshots, GitHub cleanup, submission.

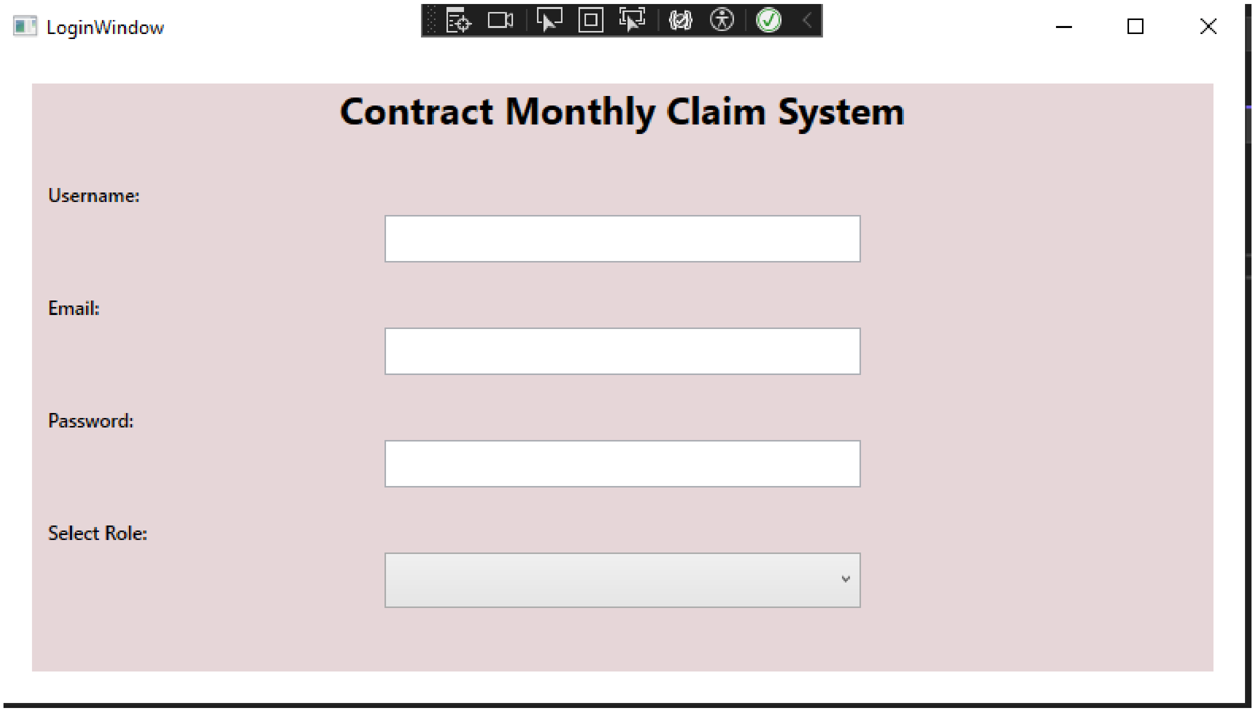
Dependencies: Wireframes → UML → WPF shell → Screens → Styling → Docs.

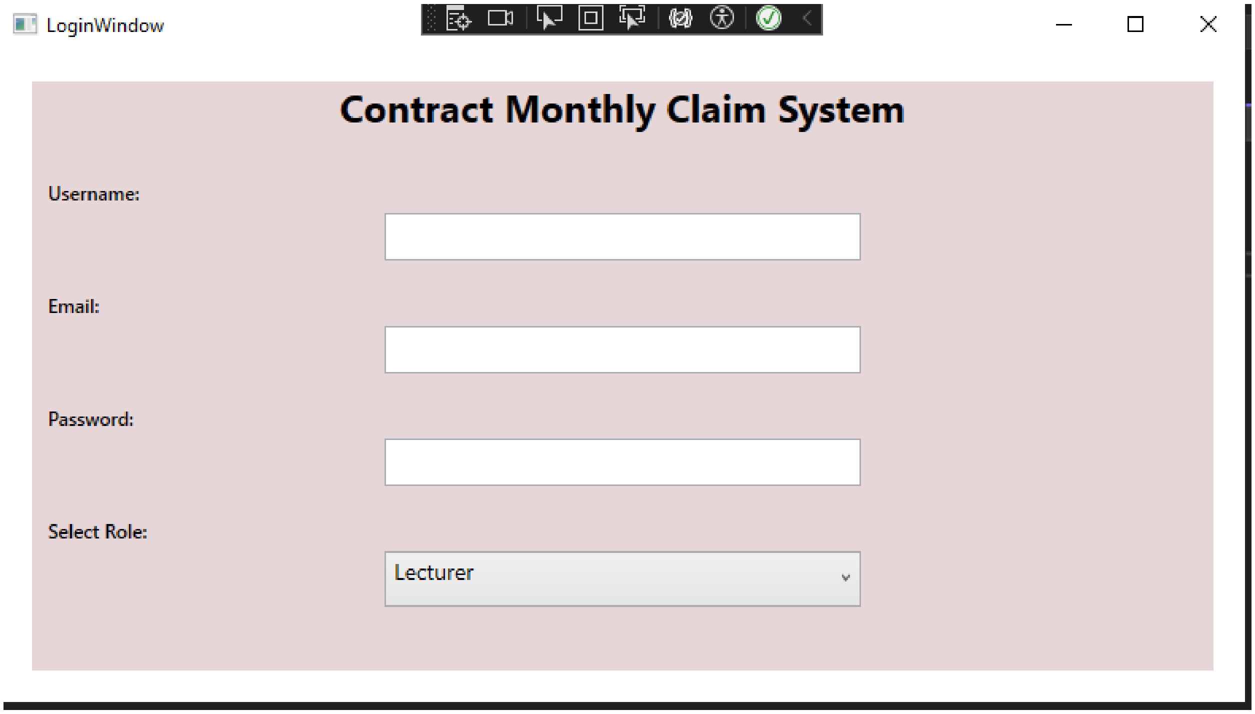
Phase	Task	Timeline	Dependencies	Responsible	Deliverables
Phase 1: Planning & Design	Requirements Gathering	3 days (Week 1)	None	Analyst	Requirement document
	UML Diagram Creation	3 days (Week 1–2)	Requirements Gathering	Analyst	UML diagrams
	UI/UX Wireframes	2 days (Week 2)	Requirements Gathering	Analyst / Designer	Wireframes
	Database Schema Design	3 days (Week 2)	UML & Wireframes	Analyst / DB Designer	Database schema
Phase 2: Prototype Development	GUI Framework Setup	3 days (Week 3)	Database schema	Developer	GUI skeleton
	Static UI Components	2 days (Week 3)	GUI Framework	Developer	UI components
	Navigation Structure	1 day (Week 3–4)	UI Components	Developer	Navigation menus
	Basic Styling	1 day (Week 4)	Navigation	Developer / Designer	Styled prototype

Phase 3: Documentation	Technical Documentation	2 days (Week 5)	All development tasks	Analyst	GitHub repository (≥5 commits)
	User Guides	1 day (Week 5)	Prototype	Analyst / Team	User guide
	Testing Documentation	1 day (Week 5)	Prototype	Tester / Analyst	Test cases & results

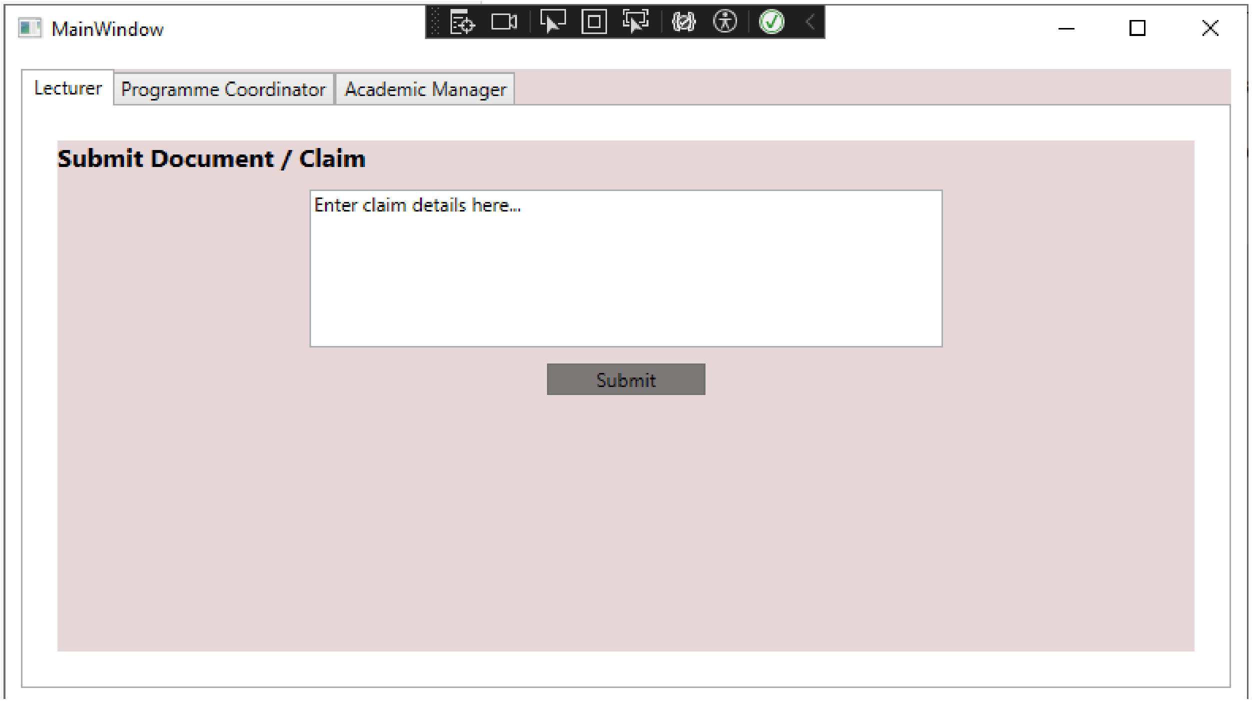
4. GUI/WPF Layout (Prototype Only)

- **Login (Mock)** – entry point with role selection and buttons/links to role dashboards.
- **Lecturer Dashboard** – “New Claim”, “My Claims”, “Upload Documents”.
- **Submit Claim Page** – Month selector, Hours worked, Hourly rate, line items (Date, Hours, Description), total preview, Upload area (disabled for Part 1).
- **Coordinator Dashboard** – List of submitted claims, claim details, approve/reject buttons (disabled).
- **Manager Approval** – Claims pending final approval, approve/reject interface, reports placeholder.
- **Claim Status Page** – Claim history with status badges (Draft, Submitted, Approved, Rejected).
  GUI Layout (non-functional prototype)



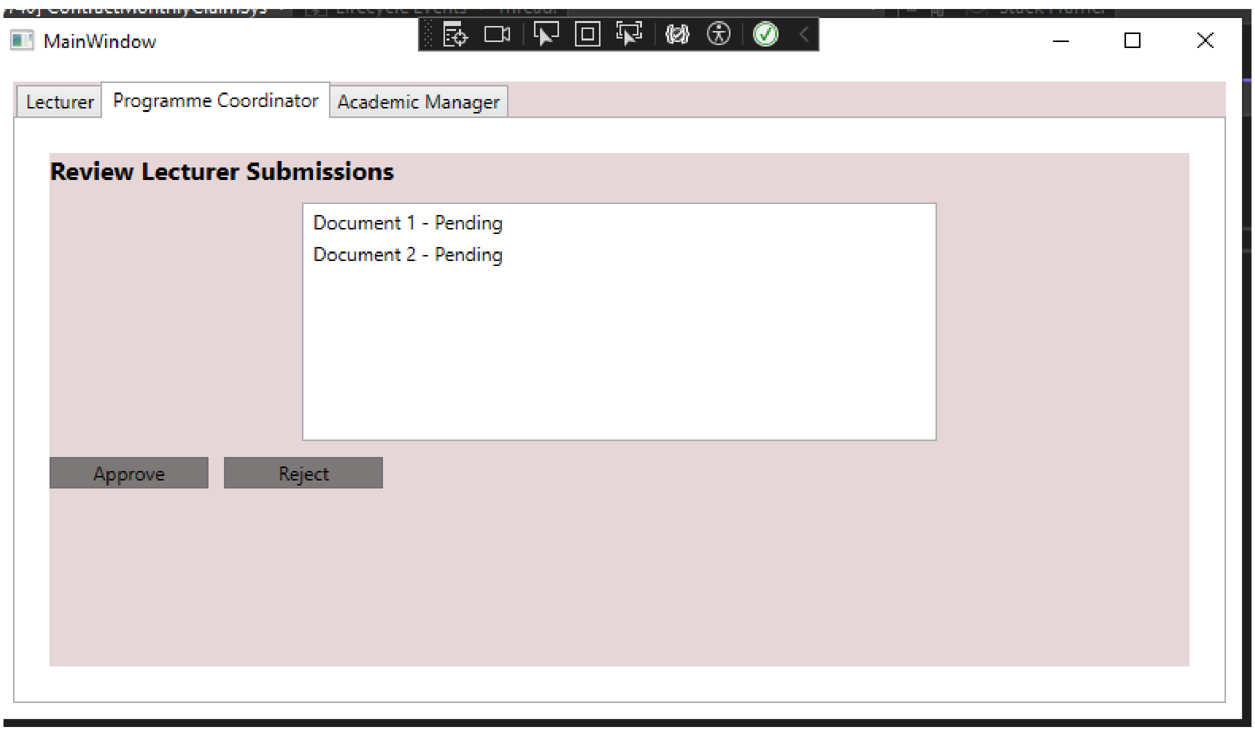


Select Role: Lecturer



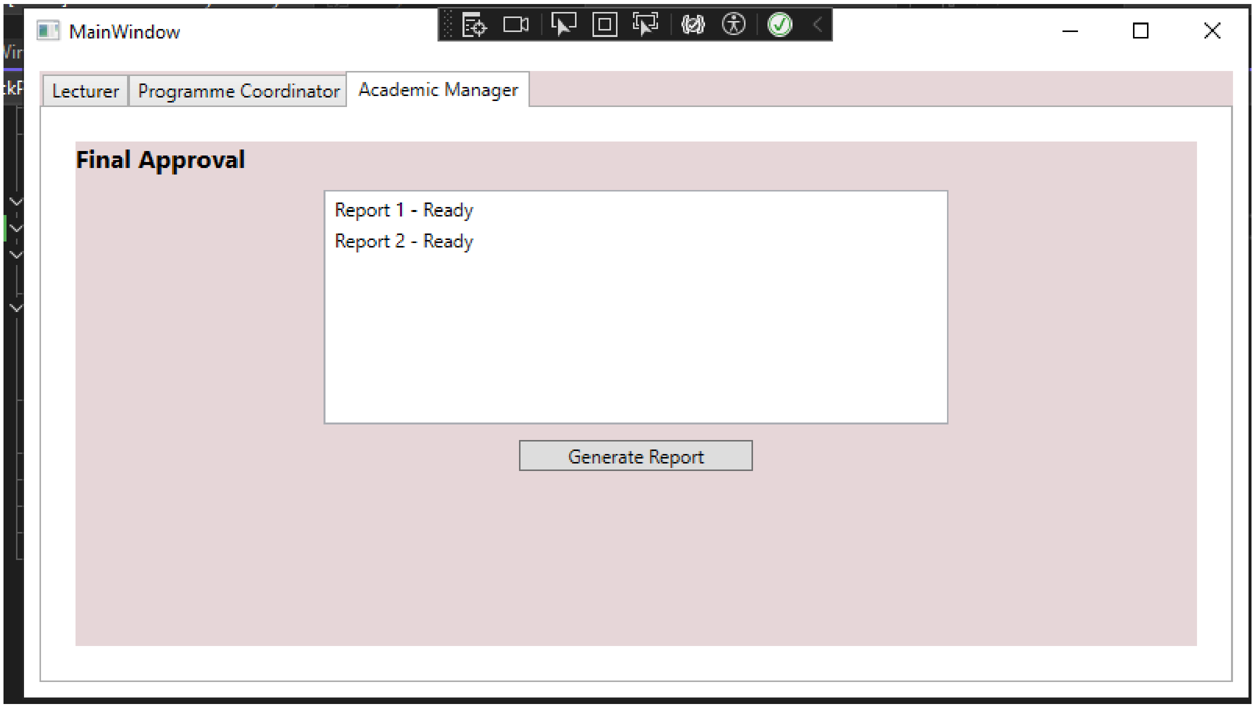
Submit Document / Claim

Enter claim details here...



Review Lecturer Submissions

Document 1 - Pending
Document 2 - Pending



Report 1 - Ready

Report 2 - Ready

Version Control Strategy

GitHub repository with minimum five commits:

- https://github.com/ST10451618/prog6212part1.git

Success Criteria

Documentation: Clear, concise 400-500 word report

UML Diagram: Complete class diagram with all relationships

Project Plan: Realistic timeline with clear dependencies

GUI: Non-functional but complete visual prototype, with a Login Page and another page for claims