**STUDENT NAME: MOTSHEUDI KARABO SELBY  
STUDENT NUMBER: ST10438285  
MODULE NAME: PROGRAMMING 2B  
MODULE CODE: PROG6212  
ASSESSMENT TYPE: ASSIGNMENT 1 [PART 1]  
  
Contract Monthly Claim System (CMCS)   
  
Introduction  
  
>** The purpose of this report is to present the prototype design of the Contract Monthly Claim System (CMCS). This system is intended to streamline the process of submitting, verifying, and approving monthly claims for independent contractor lecturers. Part 1 of the Portfolio of Evidence (POE) focuses on planning and prototype development, including design documentation, a UML class diagram, a project plan, GUI design, and version control strategy.

**1. Documentation**

**>** The CMCS prototype is designed with simplicity and efficiency in mind. The primary design choice was to adopt a role-based structure, enabling three types of users: Lecturers, Programme Coordinators, and Academic Managers. This ensures claims can be submitted, reviewed, and approved in a transparent workflow.

**>** The database structure consists of core entities such as Lecturer, Claim, Supporting Document, and Approval. Each lecturer can submit multiple claims, with claims linked to uploaded supporting documents. Programme Coordinators and Academic Managers are associated with approval records that capture verification actions. Relationships between these entities are defined through primary and foreign keys to enforce referential integrity.

**>** The GUI layout is structured to maximize usability. Lecturers have a dashboard with a button to submit claims and upload documents. Coordinators and managers have dashboards for reviewing and approving claims. All users can track claim statuses. A clean and intuitive interface design was prioritized, with consistent navigation and reliable status indicators.

**>** Assumptions: It is assumed that all users have access to stable internet connections and basic digital literacy.

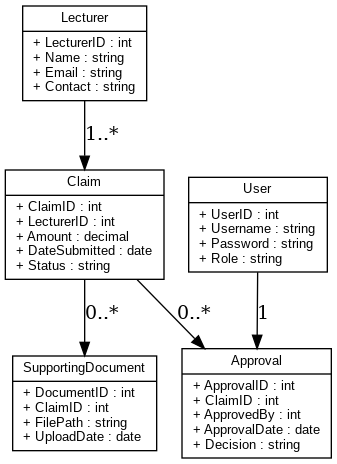
**>** Constraints: The prototype stage focuses solely on interface and structural design, with no backend functionality implemented at this stage.

**2. UML Class Diagram**

**The UML class diagram represents the database schema with the following entities:**

**>** Lecturer (LecturerID, Name, Email, Contact)  
**>** Claim (ClaimID, LecturerID, Amount, DateSubmitted, Status)  
**>** SupportingDocument (DocumentID, ClaimID, FilePath, UploadDate)  
**>** Approval (ApprovalID, ClaimID, ApprovedBy, ApprovalDate, Decision)  
**>** User (UserID, Username, Password, Role)

**Relationships:**

**>** Lecturer submits Claims.  
**>** Claims contain Supporting Documents.  
**>** Claims require Approval from Programme Coordinators and Academic Managers.  
**>** Users are linked to roles and actions.  
  


**3. Project Plan**

**The project plan consists of five phases:**

**>** Requirement Analysis (Week 1–2) – Gather functional requirements and identify user roles.

**>** Database and UML Design (Week 2–3) – Create class diagram and finalize schema.

**>** GUI Wireframing and Prototype (Week 3–4) – Design WPF/MVC layouts for user dashboards.

**>** Documentation Compilation (Week 4) – Prepare project report and integrate designs.

**>** Version Control and Submission (Week 5) – Push commits, finalize GitHub repository, and submit.

**>** Dependencies include database design completion before GUI development and documentation compilation as the final step.

| **Task ID** | **Task Description** | **Dependencies** | **Duration** | **Start Day** | **End Day** |
| --- | --- | --- | --- | --- | --- |
| 1 | Requirements Analysis | None | 1 day | Day 1 | Day 1 |
| 2 | Database Design & UML Diagram | Task 1 | 2 days | Day 2 | Day 3 |
| 3 | GUI Layout Design (WPF/MVC) | Task 1 | 2 days | Day 2 | Day 3 |
| 4 | GUI Prototype Development | Task 3 | 2 days | Day 4 | Day 5 |
| 5 | Documentation of Design Choices | Tasks 2 & 4 | 2 days | Day 4 | Day 5 |
| 6 | Review and Refinement | Tasks 2, 4, 5 | 1 day | Day 6 | Day 6 |
| 7 | Submission of Part 1 | Task 6 | 0 day | Day 7 | Day 7 |

**Gantt Chart**

Day: 1 2 3 4 5 6 7

Task 1: █

Task 2: █ █

Task 3: █ █

Task 4: █ █

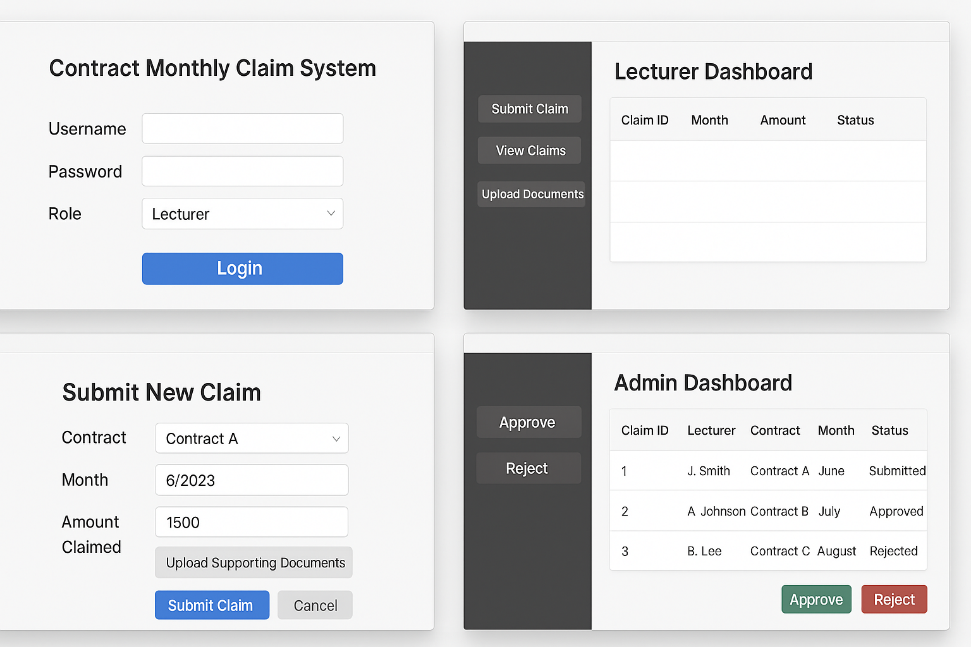
Task 5: █ █

Task 6: █

Task 7: █

**Legend:** █ = Task duration

**4. GUI Design (Prototype)**

**>** The GUI is designed as a non-functional prototype that visually represents the final application.  
**>** **Lecturer Dashboard**: “Submit Claim” button, document upload option, and status tracker.  
**> Coordinator Dashboard**: Claim verification interface with claim details.  
**> Manager Dashboard**: Final approval interface.  
**> Claim Status Page**: Transparent view of pending, approved, or rejected claims.  
**>** The layout emphasizes user-friendliness with intuitive buttons, consistent fonts, and accessible menus.  
  


**5. Version Control**

**A GitHub repository was created for CMCS. Five commits will be made to demonstrate version control best practices:**

**>** Initial commit – Project setup.  
**>** Added UML class diagram – Included design structure.  
**>** GUI wireframe commit – Uploaded non-functional GUI prototype.  
**>** Project plan commit – Added timeline and Gantt chart.  
**>** Final documentation commit – Report and supporting files.  
**>** Commit messages will be clear and descriptive, reflecting incremental progress.

**Conclusion**

The CMCS prototype effectively demonstrates the planning, design, and user interface layout of the system. While no functionality is implemented at this stage, the design provides a solid foundation for subsequent development. The clear database schema, intuitive GUI, and realistic project plan ensure that the system can be developed in a structured and achievable manner in later stages of the POE.