SBONGAKONKE KWANELE JELE

ST10383731

PROG6212 PART 2

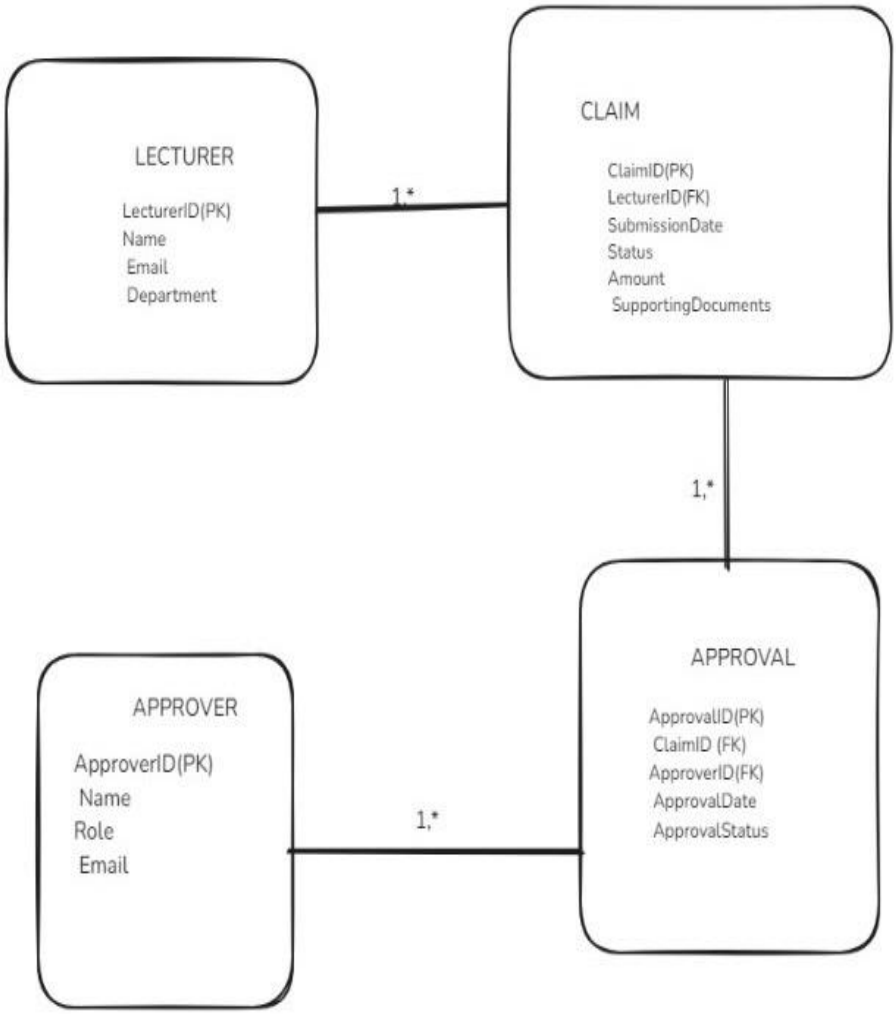
(UPDATED MY WHOLE REPORT AND PROVIDED SCREENSHOT'S OF MY GUI DESIGN AND PROVIDED LINK TO MY GITHUB)

Contract Monthly Claim System Prototype Design Report

1. Introduction

The Contract Monthly Claim System is a .NET web-based application designed to facilitate the submission and approval of monthly claims for independent contractor lecturers. The primary users of this system are lecturers, Programme Coordinators, and Academic Managers. The system allows lecturers to submit claims, upload supporting documents, and track claim statuses, while Programme Coordinators and Academic Managers can approve or reject claims. This report details the design decisions, including a UML class diagram, project plan, and GUI interface using MVC or Windows Presentation Forms (WPF) in .NET Core.

2. UML design



Classes:

1. Lecturer

- Attributes:LecturerID, Name, Email, Department

- Relationships: One-to-Many relationship with Claims.

2. Claim

- Attributes: ClaimID, LecturerID, SubmissionDate, Status, Amount, SupportingDocuments

- Relationships: Many-to-One relationship with Lecturer, One-to-Many relationship with Approvals.

3. Approval

- Attributes: ApprovalID, ClaimID, ApproverID, ApprovalDate, ApprovalStatus

- Relationships: Many-to-One relationship with Claim, Many-to-One relationship with Approver.

4. Approver

- Attributes: ApproverID, Name, Role, Email

- Relationships: One-to-Many relationship with Approvals.

UML Diagram Overview:

- Lecturer has a one-to-many relationship with Claim.

- Claim has a one-to-many relationship with Approval.

- **Approval** has a many-to-one relationship with **Claim** and **Approver.**

3. Project Plan

The development of the Contract Monthly Claim System is broken into phases. Below is a detailed project plan outlining the key tasks, dependencies, and timeline.

Phase 1: Requirement Analysis (2-3 Days)

- Gather and document requirements.

- Identify system users and their needs (lecturers, coordinators, managers).

Phase 2: UML and Database Design (3-4 Days)

- Create a UML class diagram representing the data relationships.

- Design database tables and relationships using a relational database (e.g., SQL Server).

Phase 3: GUI Design and Prototype Development (5-6 Days)

- Develop wireframes for the user interface.

- Implement a basic prototype using either MVC or Windows Presentation Forms (.NET Core). This stage focuses on the layout and does not include any functional logic.

Phase 4: Documentation (2-3 Days)

- Write detailed documentation explaining the design choices.

- Document assumptions and constraints.

Phase 5: Version Control (Ongoing)

- Set up a GitHub repository for version control.

- Regularly commit design iterations and documentation.

The total time allocated for the prototype is approximately two weeks, ensuring a manageable and realistic timeline.

4. GUI Design

I have two GUI design's both pushed to github and screenshots provided

For the GUI design, the system will use (WPF) within .NET Core, based on user-friendliness and the system's requirements.

Lecturer Dashboard:

- Claim Submission Form: A simple form where lecturers can enter claim details such as the amount, date, and upload supporting documents. This form includes a "Submit" button to process the claim.

- Document Upload: Integration of a file upload feature allowing lecturers to attach supporting documents for their claims.

- Claim Status Tracker: An interface where lecturers can view the current status of their claims (e.g., Pending, Approved, Rejected).

Admin Dashboard (Programme Coordinator/Academic Manager):

- Claims Overview: A table showing all submitted claims. Each row provides details like claim ID, lecturer name, date, amount, and status.

- Approval Workflow: Coordinators and managers can approve or reject claims with a single click and can review the uploaded supporting documents.

Design Considerations:

- Simplicity: The UI is designed to be intuitive, minimizing the number of clicks needed for key actions like claim submission and approval.

- Consistency: The design uses a consistent layout and color scheme to make the experience seamless.

- Feedback: After actions such as submitting a claim, the system provides feedback (e.g., "Claim submitted successfully")

5. Assumptions and Constraints

Assumptions:

- Lecturers have unique login credentials, and role-based access is enforced (lecturers, coordinators, and managers).

- Claims can only be made for the current month.

- Supporting documents are required for each claim.

Constraints:

- The system should provide secure file storage for uploaded documents.

- The system must handle concurrent claims and approvals from multiple users without performance degradation.

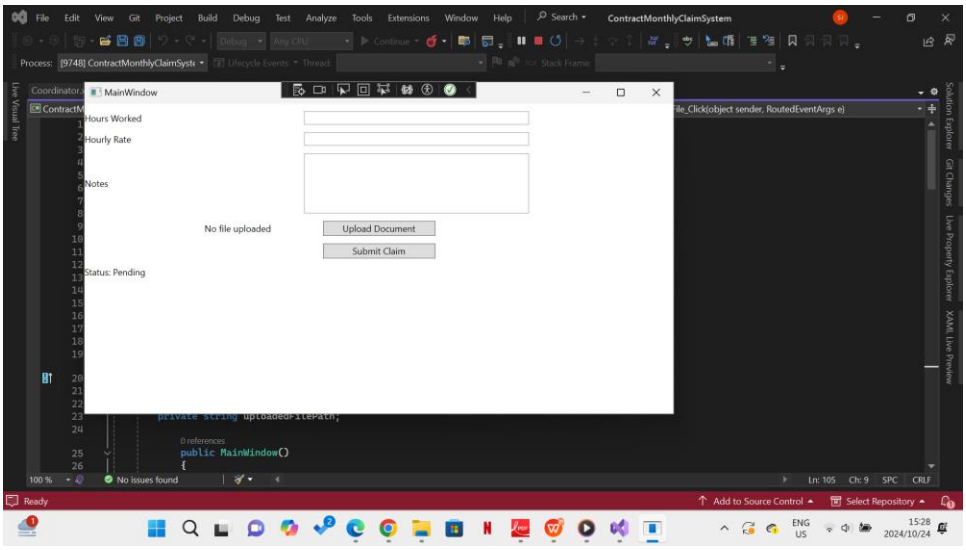
- Role-based permissions must ensure that only Programme Coordinators and Academic Managers can approve claims.

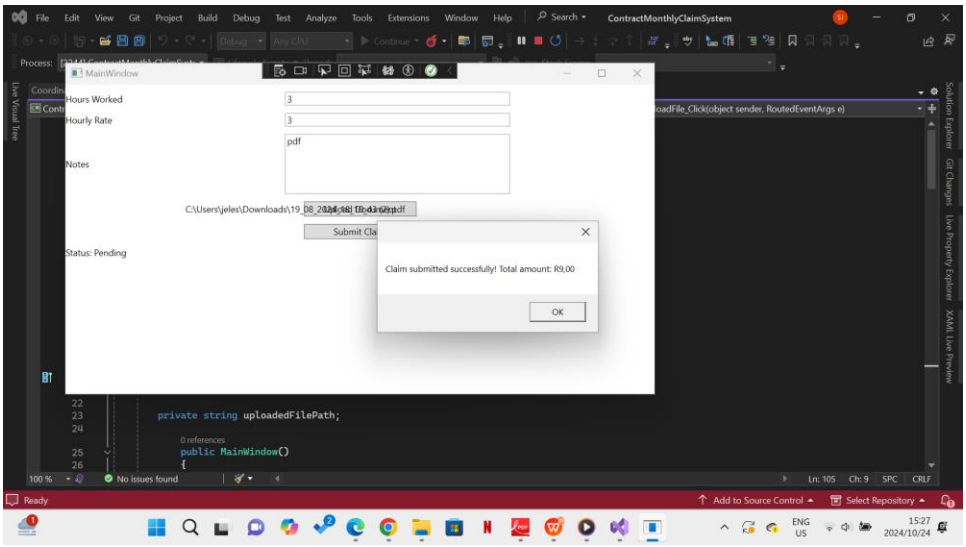
LINK TO MY GITHUB

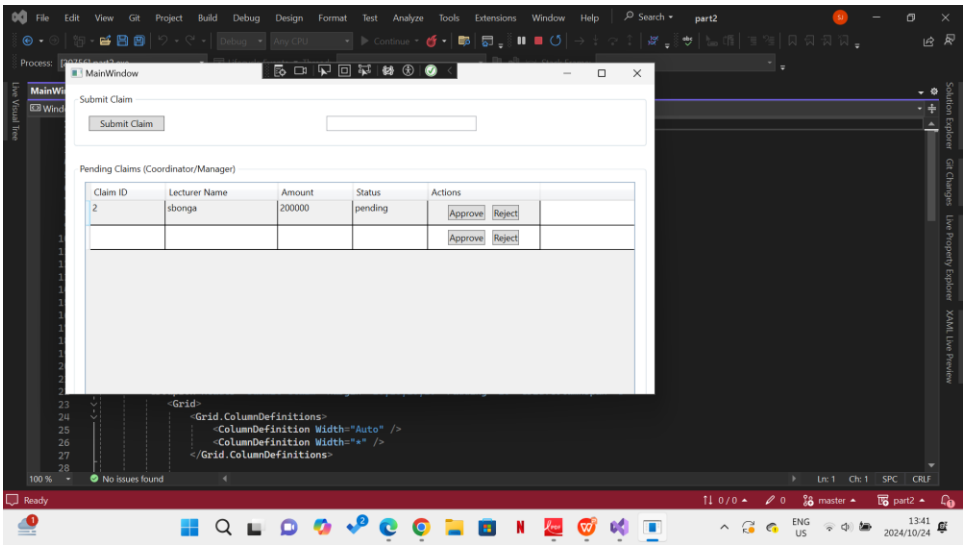
1. https://github.com/st10383731/-POE-PART2/tree/master/ContractMonthlyClaimSystem

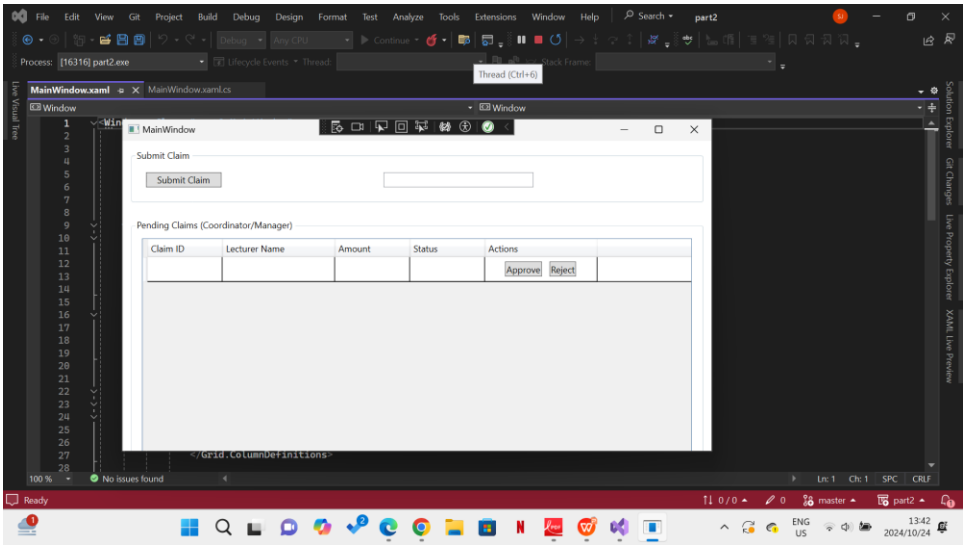
https://github.com/st10383731/-POE-PART2?tab=readme-ov-file#contractmonthlyclaimsystem

2.https://github.com/st10383731/part2/blob/master/part2/MainWindow.xaml.cs









6. Conclusion

In this prototype, the focus is on designing the structure and interface of the Contract Monthly Claim System. The UML class diagram accurately reflects the entities and relationships necessary to manage the claims, while the GUI design is centered on user-friendliness and role-specific functionality. The project plan outlines realistic milestones, ensuring that the system can be developed efficiently. The version control plan via GitHub

will ensure that the codebase is well-maintained and documented throughout the development cycle.