PROG6212

ST10456400

Mongezi Thomo

POE PART 1

Table of Contents

Introduction

This project aims to design a prototype for the Contract Monthly Claim System. The system will streamline how lecturers submit their claims, while enabling Programme Coordinators and Academic Managers to verify and approve claims. I will develop this prototype using **.NET Core MVC** for GUI demonstration. The focus is on project planning, UML diagrams, and GUI.

Design Choices

1. **Separation of Roles (Lecturer, Coordinator, Academic Manager)**
    - Ensures accountability at each level.
    - Reduces fraud since one person cannot claim, verify, and approve their own work.
2. **Register & Login Pages**
    - Implemented to enforce security and personalized access.
    - Chosen over a shared login because it enables audit trails (tracking who submitted/approved what).
3. **Dashboard Layouts**
    - **Lecturer Dashboard:** Minimalist UI with claim submission form and upload button.
    - **Coordinator Dashboard:** List view of claims with verify/reject actions.
    - **Academic Manager Dashboard:** Automated claim list, final approve/reject buttons.
4. **Database Structure (Relational Design)**
    - Chosen because claims have a natural **one-to-many** relationship with lecturers and can be verified at multiple levels.

Database Structure

Tables

1. **Users**
    - UserID (PK)
    - FullName
    - Email
    - PasswordHash
    - Role (Lecturer, Coordinator, AcademicManager)
2. **Claims**
    - ClaimID (PK)
    - UserID (FK → Users.UserID)
    - HoursWorked
    - HourlyRate
    - TotalAmount
    - SubmissionDate
    - CoordinatorStatus (Pending, Verified, Rejected)
    - ManagerStatus (Pending, Approved, Rejected)
3. **SupportingDocuments**
    - DocumentID (PK)
    - ClaimID (FK → Claims.ClaimID)
    - FilePath
    - UploadDate

GUI Layout

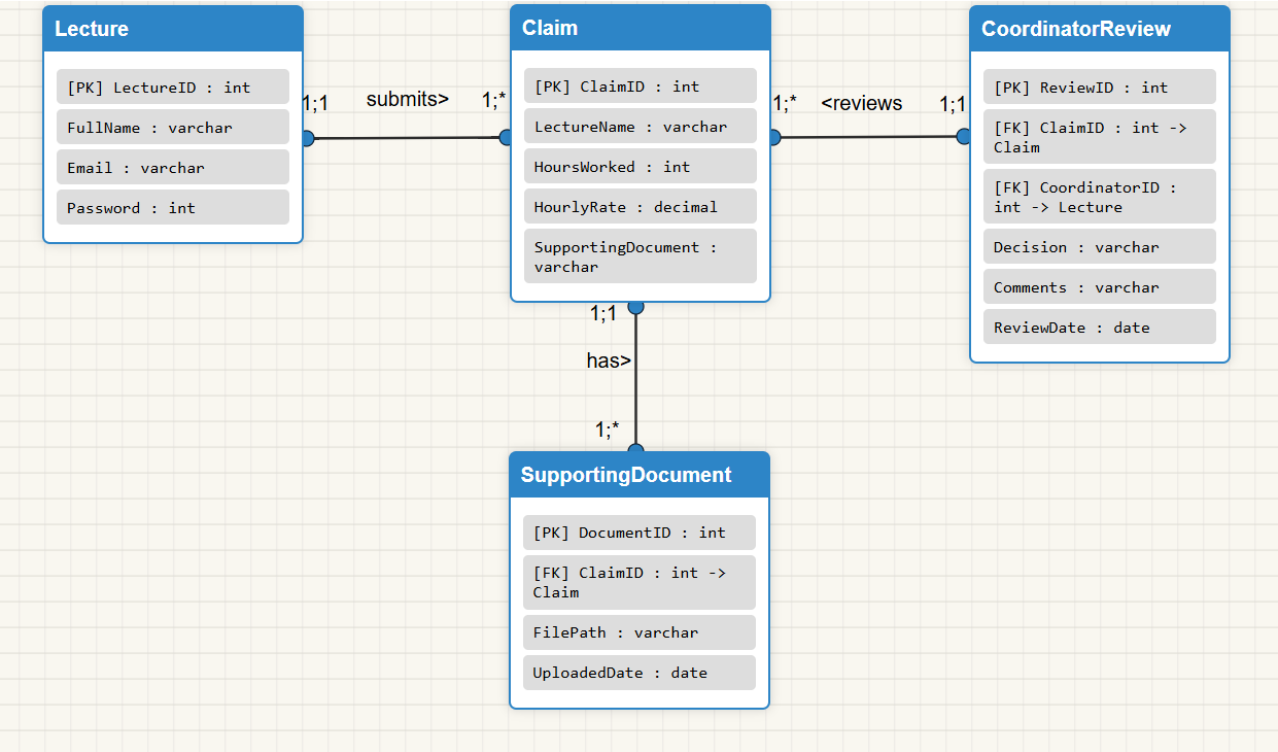
- **Lecturer GUI:** Form-based with claim submission fields and file upload.
- **Coordinator GUI:** Tabular view of submitted claims with verify/reject.
- **Academic Manager GUI:** Automated, read-only view of coordinator-approved claims with final action buttons.

Each GUI uses **tables for readability** and **color-coded buttons** for clarity (green approve, red reject).

Assumptions

1. Each lecturer must **register and login** before submitting a claim.
2. Claims move **sequentially** → Lecturer → Coordinator → Academic Manager.
3. Academic Managers **do not enter claims manually**; they only view, approve, or reject.

UML Class Diagram



Conclusion

The development of the Contract Monthly Claim System (CMCS) prototype shows how technology can make the process of managing lecturer claims more efficient and organized. By giving specific roles to Lecturers, Coordinators, and Academic Managers, the system creates a clear flow of responsibilities. This not only reduces errors but also ensures that claims are verified and approved in a transparent way.

When designing the system, I focused on creating secure login and registration pages, structuring the database so that the relationships between data are clear, and designing dashboards that are simple and easy to use.

It could later be expanded to include features like payroll integration, notifications for users, and reporting tools for management. With these improvements, the system has the potential to grow into a complete solution for managing contract claims in academic institutions.

References

ChatGPT, 2022. *ChatGPT*. [Online]
Available at: https://chatgpt.com
[Accessed 01 September 2024].