POE PART 1

[ST10395268]

**BCAD2**

PROG6212

9/17/2025

Dharshanee Moodley

Table of Contents

1   DOCUMENTATION

## 1.   Design Choices

a.  Architecture

The Model-View-Controller (MVC) pattern in ASP.NET Core is used in the system's design.  This method divides issues:

- Data and business logic are handled by the model.

- View uses Razor Pages to display data to the user.

- The controller controls application flow and user input.

Maintainability, scalability, and a distinct division of duties are all guaranteed by this framework (GeeksForGeeks, 2025).

b.  Technology Stack

- **Frontend:** Razor Pages, HTML5, CSS3, and Bootstrap

- **Backend:** ASP.NET Core MVC with C#.

- **Database:** Entity Framework Core (EF Core) is used with SQL Server.

- **Authentication:** Role-based access using ASP.NET Identity (Academic Manager, Program Coordinator, Lecturer) (GeeksForGeeks, 2025).

c.  User Roles

- **Lecturer:** Uploads supporting documentation and submits claims every month.

- **Programme Coordinator:** Examines, accepts, or rejects claims.

- **Academic Manager:** Final authority for approval and supervision (GeeksForGeeks, 2025).

## 2. Database Structure

The database is made to facilitate document management, user roles, and claim tracking. Important entities consist of:

- **Lecturer**
  - LecturerID (PK)
  - Name
  - Email
  - HourlyRate

- **Claim**
  - ClaimID (PK)
  - LecturerID (FK)
  - Month
  - HoursWorked
  - TotalAmount
  - Status (Pending, Approved, Rejected)

- **Document**
  - DocumentID (PK)
  - ClaimID (FK)
  - FilePath
  - UploadDate

- **Approval**
  - ApprovalID (PK)

- ClaimID (FK)

- ApprovedBy

- ApprovalDate

- Remarks

- **UserRole**

  - RoleID (PK)

  - RoleName

In EF Core, entity relationships are enforced through the use of navigation properties and foreign keys. A lecturer is associated with each claim, and each claim may have a single document and approval record (GeeksForGeeks, 2025).

3. **GUI Interface**

a. Lecturer Interface:

- **Claim Submission Form**

  - Selector for the month

  - Enter the number of hours worked.

  - Total amount automatically calculated

  - Uploading a file for supporting documentation

- **Claim History**

  - Submitted claims in a table with status indicators (GeeksForGeeks, 2025).

b. Coordinator/Manager Interface

- **Claim Review Dashboard**

- o   A list of unresolved claims

- o   View the paperwork and claim information.

- o   Accept/Reject with comments

Bootstrap is used for layout and styling in the GUI's responsive and user-friendly design (GeeksForGeeks, 2025).

## 4.  Assumptions and Constraints

Assumptions

- Every month, one claim is submitted by each lecturer.

- The database contains the predefined hourly rate.

- Each claim must be accompanied by supporting documentation.

- Both the manager and the coordinator must authorize claims (GeeksForGeeks, 2025).

Constraints

- Only PDF and picture files (up to 5MB) may be uploaded.

- Once accepted, claims cannot be changed.

- Certain features can only be accessed by authenticated users who possess the necessary roles (GeeksForGeeks, 2025).
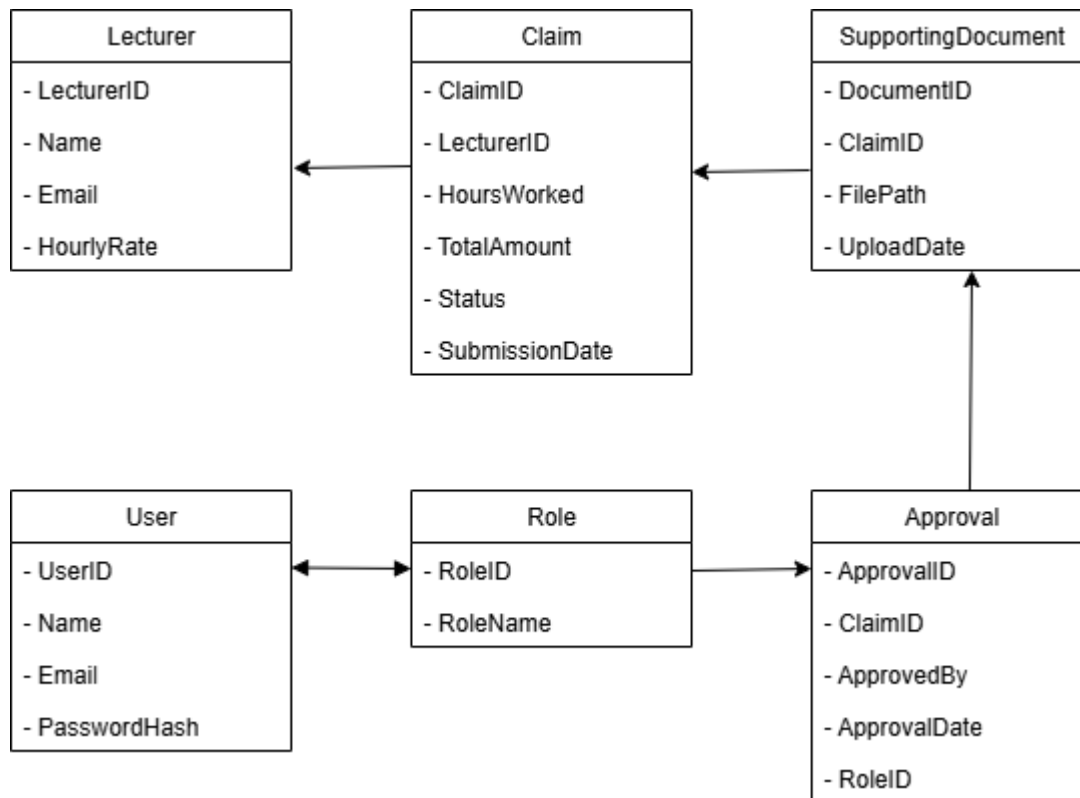
## 5.  Rationale

The layout gives priority to:

- **Efficiency:** Automating routine processes such as tracking documents and performing calculations.

- **Accuracy:** Role-based approvals guarantee accountability and minimize errors.

- **User Experience:** Interfaces that are responsive and clean for every role.

- **Scalability:** Future improvements like analytics and reporting are made possible via MVC and EF Core (GeeksForGeeks, 2025).

## 2  UML CLASS DIAGRAM FOR DATABASES

| Lecturer |
| --- |
| - LecturerID |
| - Name |
| - Email |
| - HourlyRate |

| Claim |
| --- |
| - ClaimID |
| - LecturerID |
| - HoursWorked |
| - TotalAmount |
| - Status |
| - SubmissionDate |

| SupportingDocument |
| --- |
| - DocumentID |
| - ClaimID |
| - FilePath |
| - UploadDate |

| User |
| --- |
| - UserID |
| - Name |
| - Email |
| - PasswordHash |

| Role |
| --- |
| - RoleID |
| - RoleName |

| Approval |
| --- |
| - ApprovalID |
| - ClaimID |
| - ApprovedBy |
| - ApprovalDate |
| - RoleID |

Satzinger, J., Jackson, R. and Burd, S., 2023. *Systems Analysis and Design in a Changing World. 7th ed.* Massachusetts: Cengage.

3    PROJECT PLAN

**CMCS Prototype Development**

**Project Scope:**

Create a working prototype of a .NET Core MVC web application to handle independent contractor lecturers' monthly claims.  The prototype consists of:

- Database design UML class diagram
- GUI wireframes
- MVC structure configuration
- Project records (PM Study Circle, 2024).

**Work Breakdown Structure (WBS)**

| Task No. | Task Description | Duration | Dependencies |
|----------|-----------------|----------|--------------|
| 1 | Collecting and analyzing requirements | 2 days | - |
| 2 | Describe the roles and use cases of the system. | 1 day | Task 1 |
| 3 | Create a database UML class diagram. | 2 days | Task 2 |
| 4 | Create GUI Wireframes for Managers, Coordinators, and Lecturers | 3 days | Task 2 |

| 5 | Configure the Project Structure for.NET Core MVC | 2 days | Task 1 |
| 6 | Build Models Using UML Diagrams | 2 days | Task 3, Task 5 |
| 7 | Make placeholder views (a GUI that isn't working). | 3 days | Task 4, Task 5 |
| 8 | Make Controllers Using False Information | 2 days | Task 6, Task 7 |
| 9 | Assemble Design Documentation | 2 days | Task 3-8 |
| 10 | Evaluate and Complete the Prototype | 1 day | Task 9 |

Gido, J., Clements, J., Baker, R., Harinarain, N. and Eresia-Eke, C. 2022. *Successful Project Management in South Africa.* 2nd ed. Andover: Cengage.

## Timeline Overview (2.5 Weeks)

| Week | Milestones |
| --- | --- |
| Week 1 | Tasks 1–4: UML, GUI Design, Roles, and Requirements |
| Week 2 | Tasks 5–8: Controllers, Models, Views, and MVC Setup |
| Week 3 | Tasks 9–10: Review, Submission, and Documentation |

Gido, J., Clements, J., Baker, R., Harinarain, N. and Eresia-Eke, C. 2022. *Successful Project Management in South Africa.* 2nd ed. Andover: Cengage.

## Key Dependencies

- Understanding roles and use cases is essential for Task 3 (UML Design).

- Both MVC and UML setup are required for Task 6 (Model Creation).

- MVC structure and wireframes are necessary for Task 7 (GUI Views).

- All of the earlier design work is consolidated in Task 9 (Documentation) (PM Study Circle, 2024).

**Tools & Resources**

- **IDE:** Visual Studio 2022

- **Framework:** ASP.NET Core MVC

- **Design Tools:** Draw.io or Lucidchart for wireframes and UML

- **Documentation:** Markdown or Microsoft Word (PM Study Circle, 2024).

# 4    REFERENCE LIST

GeeksForGeeks, 2025. *Best Practices For Documenting Database Design.* [online] Available at: https://www.geeksforgeeks.org/dbms/best-practices-for-documenting-database-design/ [Accessed 17 September 2025].

Gido, J., Clements, J., Baker, R., Harinarain, N. and Eresia-Eke, C. 2022. *Successful Project Management in South Africa.* 2nd ed. Andover: Cengage.

PM Study Circle, 2024. *Project Plan Examples: How to Write an Effective Project Plan (2025).* [online] Available at: https://pmstudycircle.com/project-plan-examples/ [Accessed 17 September 2025].

Satzinger, J., Jackson, R. and Burd, S., 2023. *Systems Analysis and Design in a Changing World. 7th ed.* Massachusetts: Cengage.

.