**A logo of a company

Description automatically generated**

Ryan Millard

ST10383326

## Programming 2b

PROG6212

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

NWEG5122

POE Part 1

10 September 2024

Contents

[PrOgramming 2b 1](#_Toc176888010)

[Design Choices and Structure 3](#_Toc176888011)

[Assumptions and Constraints 5](#_Toc176888012)

[Database Design 6](#_Toc176888013)

[Project Plan 7](#_Toc176888014)

[GUI Layout 9](#_Toc176888015)

# Design Choices and Structure

**Layered architecture:**

* **Presentation:** Razor Pages, Bootstrap
  + Excellent performance, scalability, and reusability.
    - Through partials,SSR, other Razor features
  + Cleaner architecture
  + Concern separation
  + Simplifies maintenance
  + **Layout:**
    - Top navbar
    - Pages
      * Home
      * Dashboard
        + Lecturers:

New claim form

Submitted claims table

* + - * + Managers/Co-ordinators:

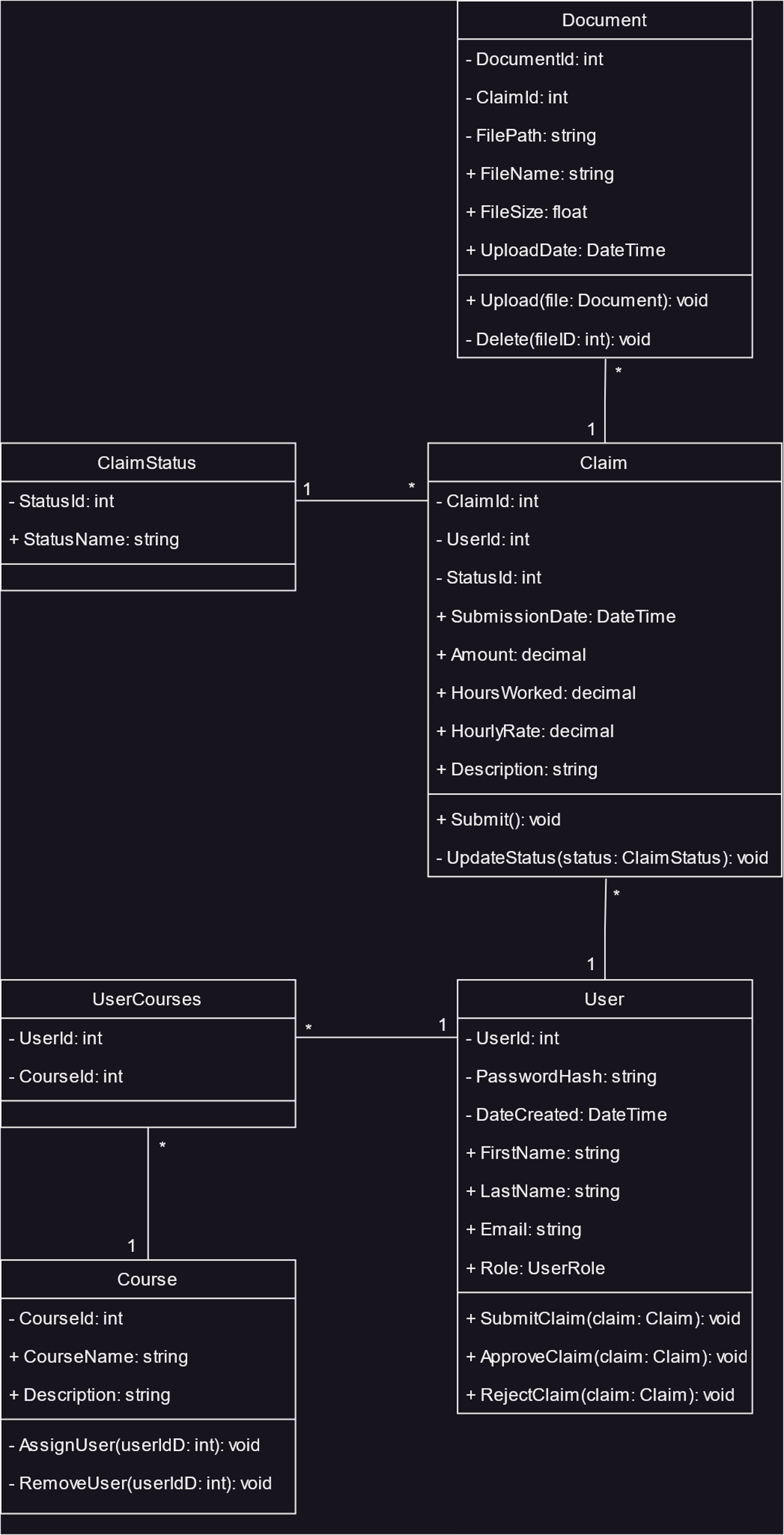
Pending/approved claims table

* + - * Login
        + Full Name, email, password
    - Footer
* **Business Logic:** Services and domain models
  + Claim processing,etc.
  + Dependency injection decouples logic and UI
* **Data Access:** Entity Framework Core
  + Safe against attacks
  + Reusable queries
* **Database:** SQL Server
  + Integrates with ASP.NET and Entity
  + **Entities:**
    - **Users:** **Lecturers**, **coordinators,** and **managers**.
    - **Claims:** Claim details submitted by lecturers.
    - **ClaimStatus:** Approval status of each claim.
    - **Documents:** Metadata about claim documents.
    - **Courses:** Courses lecturers teach.
  + **Relationships:**
    - One-to-many between **Users** and **Claims**
    - One-to-many between **Claims** and **Documents**
    - Many-to-many between **Users** and **Courses**
  + **Key design decisions:**
    - Separate **ClaimStatus** table allows for easy future extensions of status types.
    - Storing document metadata separately from **Claims** allows for multiple documents per claim.
    - **Users** and **Courses** join table, **UserCourses**, tracks courses lecturers teach.

# Assumptions and Constraints

* **User volume:** <1000 active users, primarily lecturers, is assumed.
* **Claim frequency:** Most claims expected monthly.
* **Document storage:** 2-3 documents/claim, max size: 10MB/each.
* **Workflow:** Two-step approval (coordinator->manager).
* **Browser support:** Modern browsers (Chrome, Firefox, Edge).

## Database Design



# Project Plan

**Phases and Milestones**

* **Planning (Week 1)**:
  + Scope project and specifications.
  + Finalize design and database structure.
* **Design & Prototyping (Week 2-3)**:
  + Develop:
    - View Mock-ups (Dashboard, Home, Login).
    - Razor Page layouts using Bootstrap.
* **Development (Week 4-8)**:
  + Implement
    - Entity Framework Core Database schema.
    - Role-based Razor Pages (Lecturer, Coordinator, Manager).
    - Claim submission and approval workflows.
* **Testing & Debugging (Week 9-10)**:
  + Unit testing.
  + Integration testing for Razor Pages and database interactions.
  + User Acceptance Testing for role-based functionality.
* **Finalization & Submission (Week 11)**:
  + Final bug fixes.
  + Documentation.
  + Deployment.

**Dependencies**

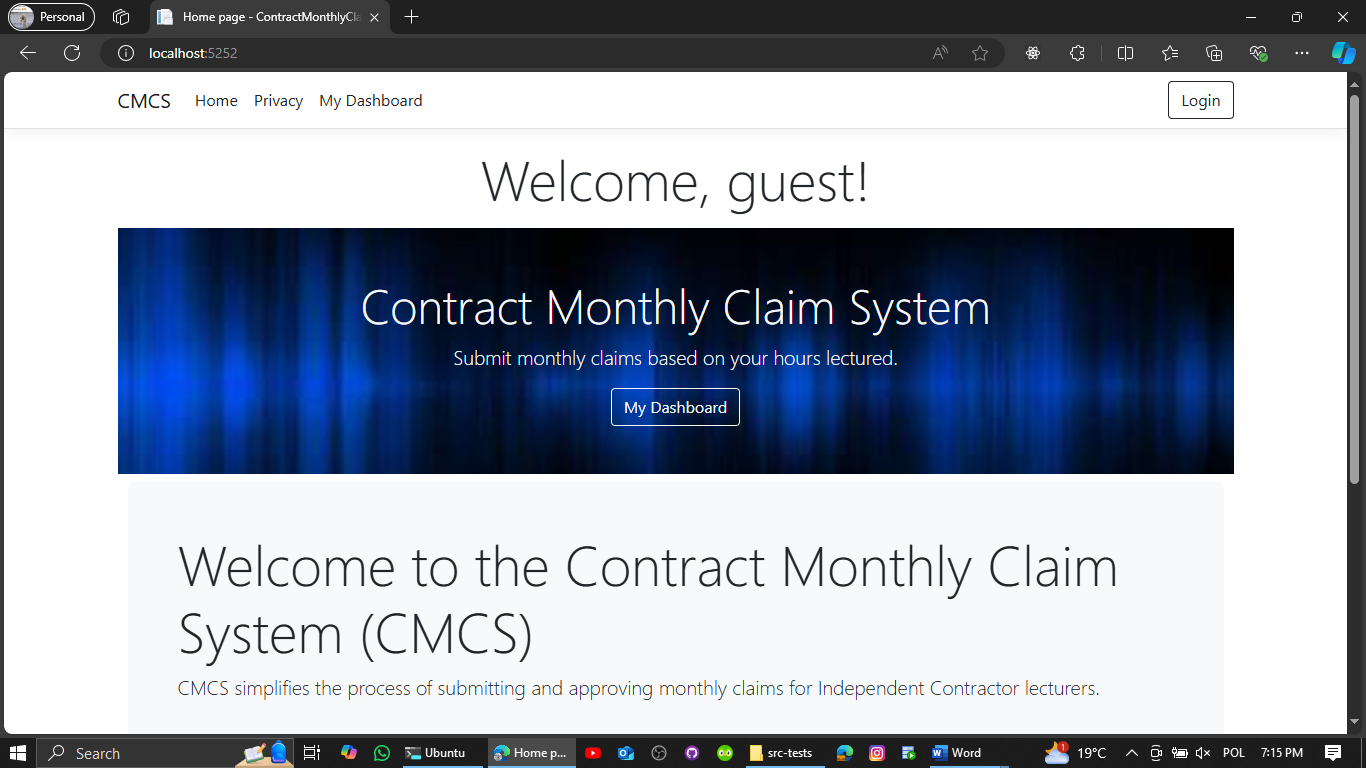
* **Tech Stack**:
  + SQL Server,ASP.NET Core, Entity Framework Core, Razor Pages, Bootstrap.
* **Dependencies**:
  + Environment setup (SQL Server,ASP.NET Core).
  + Database migration tools (EF Core Migrations).
  + Browser testing (modern browsers).

**Tasks**:

* **Week 1**: Scope and database schema.
* **Week 2**: Mock-ups.
* **Week 3**: BootstrapCSS.
* **Week 4-6**: Razor Pages.
* **Week 7-8**: Claim processing and document handling.
* **Week 9-10**: Testing, bug fixes.
* **Week 11**: Final submission preparations.

# GUI Layout

* **Home**

A screenshot of a computer

Description automatically generated

* **Login**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

* **Privacy**

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

* **Role-based Dashboard (Updated according to role)**
  + **Lecturer View:**
    - Dashboard showing claim history and status.
    - Claim submission form with clear input validation.
    - Document upload area with drag-and-drop functionality.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

* + **Coordinator/Manager View:**
    - Overview of pending claims requiring approval.
    - Detailed view of individual claims with approve/reject options.
    - Reporting tools for analysing claim patterns.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* **Consistent navigation:**
  + Minimal top navigation bar.
* **Clean, modern and consistent aesthetic:**
  + [Bootstrap](https://getbootstrap.com/) for consistent styling and layout.
  + Razor Pages for reusable code.

# GitHub Repository Link

[Ryan-Millard/Contract-Monthly-Claim-System---PROG6212-POE (github.com)](https://github.com/Ryan-Millard/Contract-Monthly-Claim-System---PROG6212-POE)