**Dự án bài tập lớn nhóm 11-Quản lý sinh viên**

**Bài 1.Folder tài liệu:**

**a.**

**1. Giới Thiệu Chung**

**1.1 Mục Đích**

Tài liệu này nhằm định nghĩa các yêu cầu cho hệ thống web quản lý giảng viên và sinh viên (Lecturer and Student Management System - LSMS). Hệ thống được thiết kế để hỗ trợ quản lý thông tin giảng viên, sinh viên, lớp học, điểm số và đăng ký môn học trong một cơ sở giáo dục. Tài liệu này cung cấp hướng dẫn chi tiết cho nhóm phát triển và làm rõ các chức năng, yêu cầu cho các bên liên quan.

**1.2 Phạm Vi**

LSMS là một ứng dụng web cho phép:

* **Quản trị viên (Admin)**: Thêm, sửa, xóa thông tin giảng viên và sinh viên, quản lý tài khoản người dùng.
* **Giảng viên**: Tạo lớp học, nhập điểm cho sinh viên, quản lý thông tin môn học.
* **Sinh viên**: Xem điểm, xem thông tin giảng viên, đăng ký môn học. Hệ thống nhằm giảm thiểu công việc thủ công, tăng tính chính xác dữ liệu và cải thiện hiệu quả quản lý học vụ. LSMS sẽ được triển khai trên nền tảng web, truy cập qua trình duyệt và lưu trữ dữ liệu trên máy chủ tập trung.

**1.3 Định Nghĩa, Thuật Ngữ và Viết Tắt**

* **LSMS**: Lecturer and Student Management System (Hệ thống quản lý giảng viên và sinh viên).
* **Admin**: Quản trị viên với quyền quản lý toàn hệ thống.
* **Giảng viên**: Người dạy học, chịu trách nhiệm nhập điểm và quản lý lớp học.
* **Sinh viên**: Người học, có quyền xem điểm và đăng ký môn học.
* **Môn học**: Lớp học hoặc khóa học do giảng viên phụ trách.
* **GUI**: Giao diện người dùng đồ họa (Graphical User Interface).
* **RDBMS**: Hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System).

**2. Mục Tiêu Hệ Thống**

Hệ thống LSMS được phát triển để:

* Tự động hóa việc quản lý thông tin giảng viên, sinh viên, lớp học và điểm số.
* Cung cấp giao diện thân thiện để quản trị viên, giảng viên và sinh viên thực hiện các tác vụ liên quan.
* Đảm bảo dữ liệu được lưu trữ an toàn, truy cập nhanh chóng và chính xác.
* Hỗ trợ sinh viên đăng ký môn học và xem thông tin học tập một cách thuận tiện.
* Giảm thiểu lỗi thủ công và tăng hiệu quả quản lý trong cơ sở giáo dục.

**3. Các Yêu Cầu Chức Năng**

**3.1 Quản Trị Viên (Admin)**

* **1**: Đăng nhập vào hệ thống với tài khoản quản trị viên.
  + Yêu cầu: Hệ thống xác thực thông tin đăng nhập (tên người dùng/mật khẩu).
* **2**: Quản lý thông tin giảng viên.
  + Thêm, sửa, xóa thông tin giảng viên (họ tên, mã giảng viên, email, số điện thoại, chuyên môn).
  + Xem danh sách giảng viên.
* **3**: Quản lý thông tin sinh viên.
  + Thêm, sửa, xóa thông tin sinh viên (họ tên, mã sinh viên, email, số điện thoại, lớp).
  + Xem danh sách sinh viên.
* **4**: Quản lý tài khoản người dùng.
  + Tạo, cập nhật, vô hiệu hóa tài khoản cho giảng viên và sinh viên.
  + Phân quyền dựa trên vai trò (Admin, Giảng viên, Sinh viên).
* **5**: Xem báo cáo.
  + Tạo báo cáo về danh sách lớp học, sinh viên đăng ký môn học, hoặc điểm số.

**3.2 Giảng Viên**

* **1**: Đăng nhập vào hệ thống với tài khoản giảng viên.
  + Yêu cầu: Xác thực thông tin đăng nhập.
* **2**: Quản lý lớp học.
  + Tạo lớp học mới (mã môn học, tên môn học, số tín chỉ, thời gian, địa điểm).
  + Sửa, xóa thông tin lớp học.
  + Xem danh sách lớp học đang phụ trách.
* **3**: Nhập và quản lý điểm sinh viên.
  + Nhập điểm cho sinh viên trong lớp học (điểm quá trình, điểm thi, điểm tổng kết).
  + Sửa, xóa điểm đã nhập.
  + Xem danh sách sinh viên và điểm trong lớp học.
* **4**: Xem thông tin sinh viên trong lớp.
  + Xem chi tiết sinh viên (họ tên, mã sinh viên, email).

**3.3 Sinh Viên**

* **1**: Đăng nhập vào hệ thống với tài khoản sinh viên.
  + Yêu cầu: Xác thực thông tin đăng nhập.
* **2**: Xem điểm.
  + Xem điểm chi tiết (điểm quá trình, điểm thi, điểm tổng kết) của các môn học đã đăng ký.
* **3**: Xem thông tin giảng viên.
  + Xem thông tin giảng viên (họ tên, email, chuyên môn) của các môn học đã đăng ký.
* **4**: Đăng ký môn học.
  + Xem danh sách môn học khả dụng (mã môn học, tên môn học, giảng viên, thời gian).
  + Đăng ký hoặc hủy đăng ký môn học trước thời hạn quy định.

**3.4 Chức Năng Chung**

* **1**: Đăng nhập/đăng xuất.
  + Hỗ trợ đăng nhập an toàn cho tất cả người dùng (Admin, Giảng viên, Sinh viên).
  + Đăng xuất khỏi hệ thống sau khi sử dụng.
* **2**: Quản lý hồ sơ cá nhân.
  + Người dùng (Giảng viên, Sinh viên) có thể xem và cập nhật thông tin cá nhân (email, số điện thoại, mật khẩu).
* **3**: Thông báo.
  + Hệ thống gửi thông báo (qua giao diện hoặc email) về các sự kiện như điểm được cập nhật, môn học mới, hoặc thời hạn đăng ký.

**. Các Giao Diện Người Dùng (Mô Tả Sơ Bộ)**

**.1 Giao Diện Đăng Nhập**

* Màn hình hiển thị trường nhập tên người dùng, mật khẩu, và nút "Đăng nhập".
* Liên kết "Quên mật khẩu" để khôi phục tài khoản.

**.2 Giao Diện Quản Trị Viên**

* Bảng điều khiển (dashboard) hiển thị danh sách giảng viên, sinh viên, và thống kê (số lượng lớp học, môn học).
* Menu điều hướng đến các chức năng: Quản lý giảng viên, sinh viên, tài khoản, báo cáo.

**.3 Giao Diện Giảng Viên**

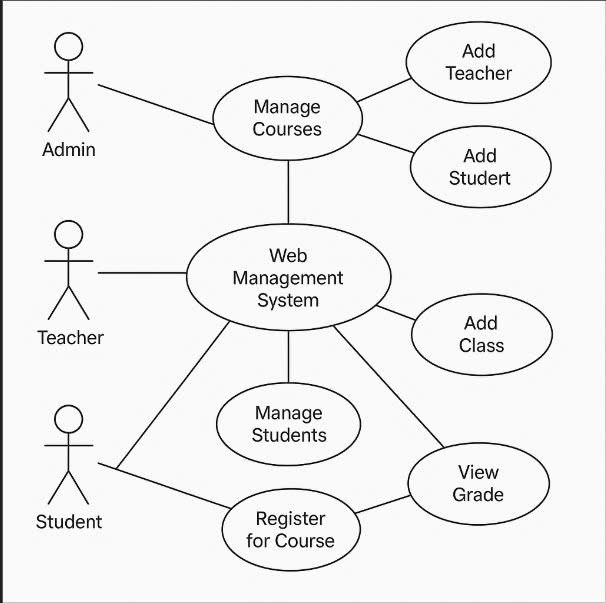
* Dashboard hiển thị danh sách lớp học đang phụ trách.
* Chức năng tạo/sửa lớp học, nhập điểm, xem danh sách sinh viên.

**.4 Giao Diện Sinh Viên**

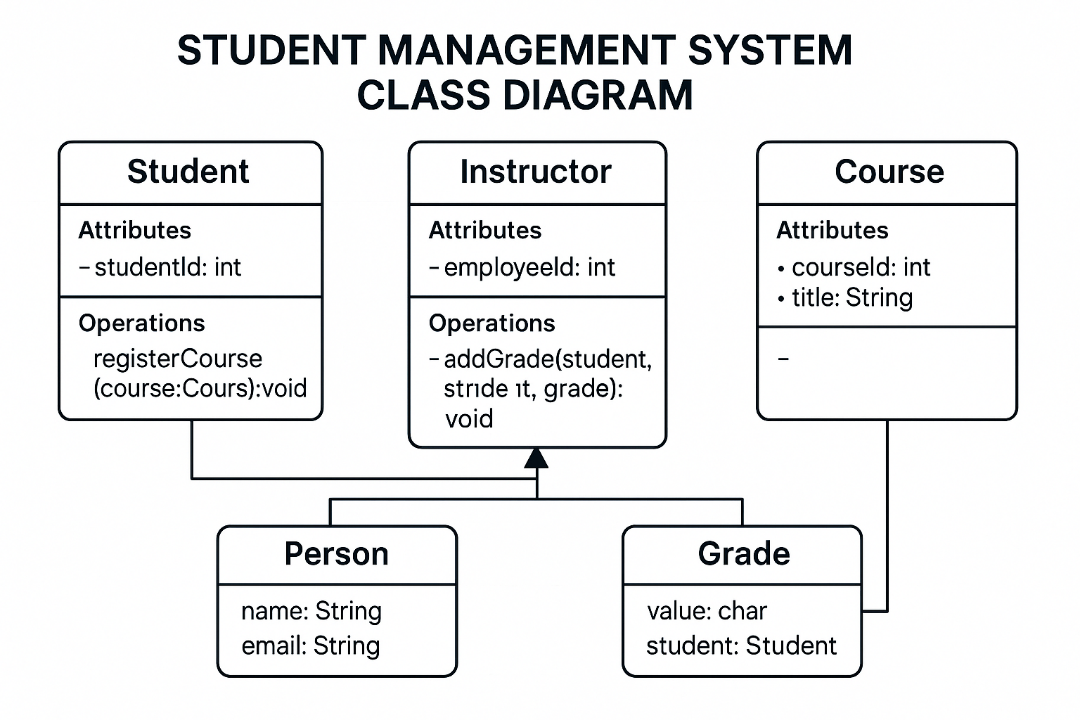
* Dashboard hiển thị điểm số, môn học đã đăng ký, và thông báo.
* Chức năng xem điểm chi tiết, xem thông tin giảng viên, đăng ký môn học.

**b.**

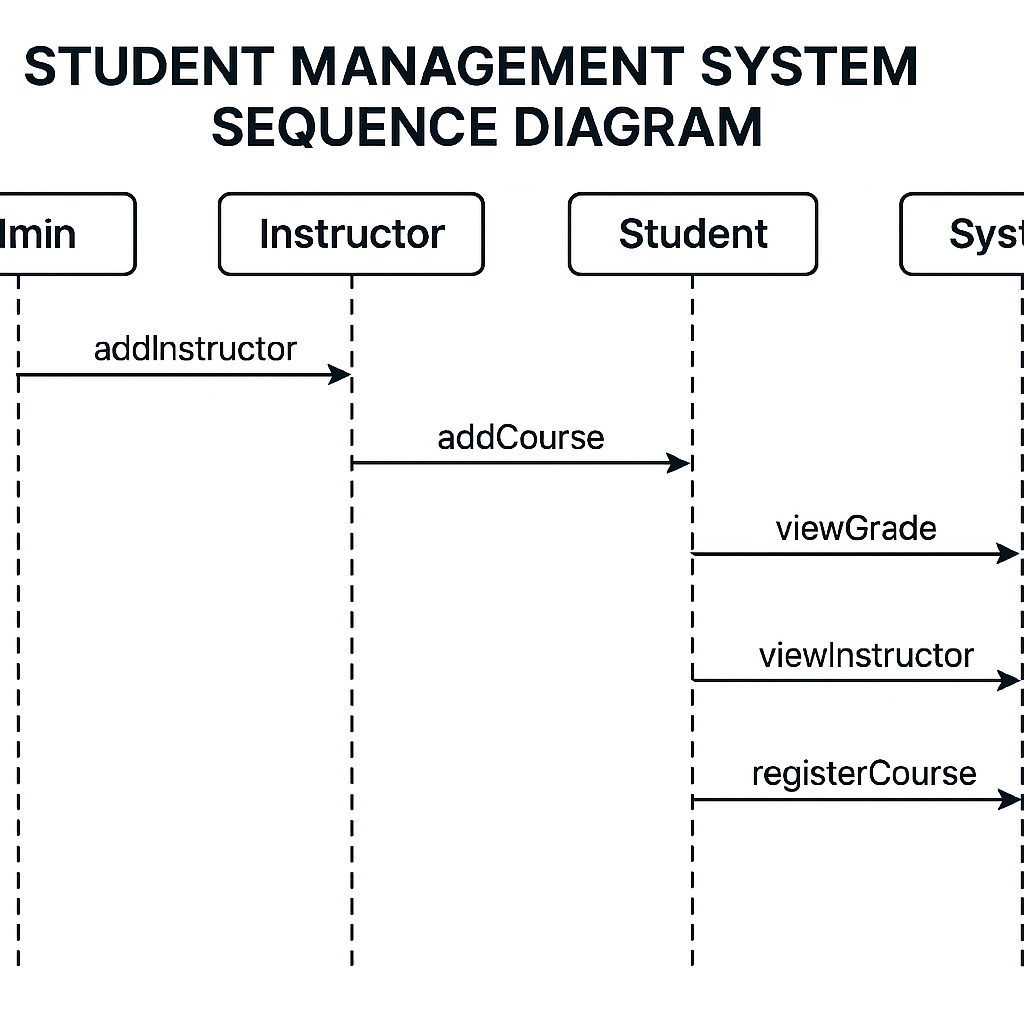
**Use Case Diagram:**

****

**Class diagram:**

****

**Sequence diagram:**

****

**c.Thiết kế kiến trúc phần mềm**

**1. Tổng Quan Kiến Trúc**

Hệ thống quản lý giảng viên và sinh viên (Lecturer and Student Management System - LSMS) là một ứng dụng web được phát triển bằng **C#** với framework **ASP.NET Core** và sử dụng **SQL Server** làm cơ sở dữ liệu. Hệ thống được thiết kế theo mô hình **MVC (Model-View-Controller)** kết hợp với **kiến trúc tầng (Layered Architecture)** để đảm bảo tính mô-đun, dễ bảo trì, khả năng mở rộng và hiệu năng cao.

**1.1 Mục Tiêu Kiến Trúc**

* Tận dụng **ASP.NET Core** để xây dựng ứng dụng web hiệu năng cao, bảo mật và dễ triển khai.
* Sử dụng **SQL Server** để quản lý dữ liệu có cấu trúc, đảm bảo tính toàn vẹn và hiệu suất truy vấn.
* Đảm bảo **bảo mật** dữ liệu người dùng và phân quyền chặt chẽ.
* Hỗ trợ giao diện thân thiện, tương thích đa nền tảng (PC và thiết bị di động).
* Dễ dàng mở rộng để thêm các chức năng mới trong tương lai.

**2. Cấu Trúc Tổng Thể**

**2.1 Các Module Chính**

Hệ thống được chia thành các module chính, mỗi module chịu trách nhiệm cho một nhóm chức năng:

1. **Module Quản Lý Người Dùng (User Management)**:
   * Quản lý thông tin và tài khoản của quản trị viên, giảng viên, sinh viên.
   * Chức năng: Thêm, sửa, xóa người dùng; quản lý phân quyền.
2. **Module Quản Lý Lớp Học (Course Management)**:
   * Quản lý thông tin lớp học/môn học (tạo, sửa, xóa).
   * Chức năng: Giảng viên tạo lớp, sinh viên đăng ký môn học.
3. **Module Quản Lý Điểm (Grade Management)**:
   * Quản lý điểm số của sinh viên (nhập, sửa, xem điểm).
   * Chức năng: Giảng viên nhập điểm, sinh viên xem điểm.
4. **Module Báo Cáo (Reporting)**:
   * Tạo báo cáo về danh sách lớp học, điểm số, hoặc sinh viên đăng ký.
   * Chức năng: Quản trị viên xem và xuất báo cáo dưới dạng PDF/Excel.
5. **Module Thông Báo (Notification)**:
   * Gửi thông báo qua giao diện hoặc email (ví dụ: điểm được cập nhật, môn học mới).
   * Chức năng: Gửi thông báo cho giảng viên và sinh viên.
6. **Module Xác Thực và Phân Quyền (Authentication & Authorization)**:
   * Quản lý đăng nhập, đăng xuất và phân quyền người dùng.
   * Chức năng: Xác thực người dùng, kiểm tra quyền truy cập.

**2.2 Kiến Trúc Tầng**

Hệ thống sử dụng **kiến trúc tầng** với 4 tầng chính:

1. **Tầng Giao Diện Người Dùng :**
   * Chịu trách nhiệm hiển thị giao diện và tương tác với người dùng.
   * Công nghệ: **ASP.NET Core Razor Pages** hoặc **Blazor** (cho giao diện động), kết hợp với HTML, CSS, JavaScript và Bootstrap.
   * Bao gồm các giao diện cho quản trị viên, giảng viên và sinh viên.
2. **Tầng Ứng Dụng (Application Layer)**:
   * Xử lý logic nghiệp vụ và điều phối yêu cầu từ giao diện đến tầng dữ liệu.
   * Chứa các **Controller** trong mô hình MVC, xử lý các yêu cầu như thêm người dùng, nhập điểm, đăng ký môn học.
3. **Tầng Mô Hình (Domain Layer)**:
   * Định nghĩa các mô hình dữ liệu và logic cốt lõi (ví dụ: User, Course, Grade).
   * Công nghệ: **C# Classes** và **Entity Framework Core** (ORM) để ánh xạ dữ liệu.
4. **Tầng Dữ Liệu (Data Layer)**:
   * Quản lý lưu trữ, truy xuất và thao tác với cơ sở dữ liệu SQL Server.
   * Công nghệ: **SQL Server** và **Entity Framework Core** (hoặc Dapper cho truy vấn hiệu năng cao).

**3. Giao Tiếp Giữa Các Thành Phần**

**3.1 Luồng Dữ Liệu**

* **Người dùng (Client)** gửi yêu cầu qua trình duyệt đến tầng giao diện.
* **Tầng Giao Diện** gửi yêu cầu đến **Tầng Ứng Dụng**
* **Tầng Ứng Dụng** tương tác với **Tầng Mô Hình** để xử lý logic và truy xuất dữ liệu từ
* Kết quả được trả về ngược lại qua các tầng và hiển thị trên giao diện người dùng.

**3.2 Cơ Sở Dữ Liệu (SQL Server)**

Cơ sở dữ liệu được thiết kế theo mô hình quan hệ với các bảng chính:

* **Users** : Lưu thông tin người dùng.
* **Courses** : Lưu thông tin môn học.
* **Grades** : Lưu điểm số.
* **Enrollments** : Lưu thông tin đăng ký môn học.
* **Notifications** : Lưu thông báo.

**4 Sơ Đồ Luồng Dữ Liệu (Ví dụ: Nhập điểm)**

1. Giảng viên gửi yêu cầu nhập điểm qua giao diện .
2. Controller nhận yêu cầu qua POST /grades.
3. Controller gọi IGradeRepository để cập nhật điểm.
4. Repository sử dụng Entity Framework Core để lưu dữ liệu vào bảng Grades trong SQL Server.
5. SignalR gửi thông báo thời gian thực đến sinh viên.
6. View hiển thị thông báo thành công cho giảng viên.

**Bài 2:Bảng kanban.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nhiệm vụ** | **Mô tả** | **Thời gian (giờ)** | **Vai trò** | **Trạng thái** | **Ưu tiên** |
| Khởi tạo | Thiết lập GitHub/GitLab, cấu hình CI/CD cơ bản. | 2 | DevOps | Done | Cao |
| Viết tài liệu SRS | Soạn thảo đặc tả yêu cầu phần mềm. | 10 | System Analyst | Done | Cao |
| Thiết kế giao diện đăng nhập | Tạo trang đăng nhập (Razor/Blazor, Bootstrap). | 6 | Frontend | To Do | Trung |
| Triển khai Entity Framework Core | Cấu hình EF Core để ánh xạ bảng SQL Server. | 10 | Backend | In Progress (50%) | Cao |
| Kiểm thử giao diện đăng nhập | Kiểm thử trên Chrome, Firefox, Edge, di động. | 3 | Tester | Testing | Trung |
| Thiết kế giao diện chi tiết | Thiết kế mockup (dashboard, nhập điểm, đăng ký môn). | 20 | UI/UX Designer | Backlog | Thấp |
| Phát triển module quản lý lịch học | Quản lý lịch học cho giảng viên, sinh viên. | 30 | Backend, Frontend | Backlog | Thấp |
| Tích hợp hệ thống LMS | Kết nối LSMS với LMS bên ngoài qua API. | 50 | Backend | Backlog | Thấp |

**Bài 5:Bảng đánh giá đồng đẳng,đóng góp của các thành viên:**

|  |  |
| --- | --- |
| **Họ và tên sinh viên** | **Đánh giá theo thang điểm 10** |
| **Bùi Gia Bảo** | **9** |
| **Lê Tài Phước** | **8** |
| **Vũ Đức Linh** | **8** |
| **Nguyễn Tuấn Phong** | **8** |
| **Bùi Đăng Tuyến** | **7.5** |