[1. Giới thiệu 1](#_Toc7257)

[1.1. Mục đích 1](#_Toc19886)

[1.2. Phạm vi 2](#_Toc24354)

[1.3. Đối tượng sử dụng 2](#_Toc8025)

[1.4. Định nghĩa, từ viết tắt, và thuật ngữ 2](#_Toc4654)

[2. Tổng quan hệ thống 3](#_Toc11189)

[2.1. Mô tả hệ thống 3](#_Toc9146)

[2.2. Kiến trúc tổng quan 3](#_Toc24011)

[3. Các yếu tố thiết kế 4](#_Toc14701)

[3.1. Giả định và phụ thuộc 4](#_Toc30838)

[3.2. Ràng buộc chung 4](#_Toc14137)

[3.3. Mục tiêu và nguyên tắc 5](#_Toc29255)

[3.4. Phương pháp phát triển 5](#_Toc23116)

[4. Chiến lược kiến trúc 5](#_Toc6433)

[4.1. Sử dụng Clean Architecture 5](#_Toc28831)

[4.2. API RESTful 6](#_Toc1056)

[4.3. Server-Side Rendering (Next.js) 6](#_Toc14817)

[4.4. Quản lý cơ sở dữ liệu 6](#_Toc19782)

[4.5. Xác thực và phân quyền 6](#_Toc9421)

[5. Kiến trúc hệ thống 7](#_Toc6659)

[5.1. Phân hệ chính 7](#_Toc29747)

[5.2. Tương tác giữa các phân hệ 7](#_Toc28824)

[6. Chính sách và chiến thuật 8](#_Toc28985)

[6.1. Quy ước mã hóa 8](#_Toc2522)

[6.2. Công cụ và thư viện 8](#_Toc15511)

[6.3. Kiểm thử 9](#_Toc2544)

[6.4. Triển khai 9](#_Toc7910)

[7. Thiết kế hệ thống chi tiết 9](#_Toc8177)

[7.1. Mô hình phân cấp chức năng 9](#_Toc30778)

[7.2. Thiết kế PDM (Physical Data Model) 10](#_Toc19349)

[7.3. Thiết kế Diagram 11](#_Toc12692)

[7.3.1. Quản lý sinh viên tham gia khóa luận 11](#_Toc16727)

[7.3.2. Ra đề và kiểm duyệt đề tài 11](#_Toc26362)

[7.3.3. Đăng ký đề tài khóa luận 12](#_Toc26810)

[7.3.4. Quản lý tiến độ thực hiện đề tài 13](#_Toc30516)

[7.3.5. Chấm điểm đề tài 14](#_Toc3464)

[7.3.6. Thống kê kết quả 15](#_Toc20090)

[7.4. Thiết kế giao diện 16](#_Toc25672)

[7.4.1. Giao diện Admin 16](#_Toc11166)

[7.4.2. Giao diện Giảng viên 17](#_Toc17224)

[7.4.3. Giao diện Sinh viên 19](#_Toc23413)

[7.4.4. Giao diện Quản lý Tiến độ 20](#_Toc15582)

[7.4.5. Giao diện Chấm điểm 21](#_Toc1158)

[7.4.6. Giao diện Thống kê 22](#_Toc27858)

[8. Từ điển thuật ngữ 23](#_Toc11602)

[9. Tài liệu tham khảo 23](#_Toc6678)

**Đặc tả Thiết kế Phần mềm (SDS)**

**Hệ thống Xử lý Tiến trình Khóa luận Tốt nghiệp**

**1. Giới thiệu**

**1.1. Mục đích**

Tài liệu này cung cấp một mô tả chi tiết về thiết kế của hệ thống xử lý tiến trình khóa luận tốt nghiệp, nhằm mục đích hướng dẫn đội phát triển triển khai hệ thống đúng với các yêu cầu được xác định trong Đặc tả Yêu cầu Phần mềm (SRS). SDS mô tả kiến trúc hệ thống, các thành phần, giao diện, và quy trình triển khai, đồng thời cung cấp các biểu đồ tuần tự và mô hình dữ liệu để minh họa rõ ràng cách hệ thống hoạt động.

**1.2. Phạm vi**

Hệ thống được thiết kế để:

* Tự động hóa quy trình quản lý khóa luận tốt nghiệp tại Khoa Công nghệ Thông tin (CNTT), Trường Đại học Công Thương TP. Hồ Chí Minh (HUIT).
* Hỗ trợ các chức năng như quản lý sinh viên, giảng viên, đề tài, tiến độ, chấm điểm, và thống kê kết quả.
* Cung cấp giao diện người dùng thân thiện cho quản trị viên, giảng viên, và sinh viên thông qua ứng dụng web.

Hệ thống sử dụng:

* **Back-end**: ASP.NET Core Web API với Clean Architecture.
* **Front-end**: Next.js với Server-Side Rendering (SSR) và Static Site Generation (SSG).
* **Cơ sở dữ liệu**: SQL Server 2022.

**1.3. Đối tượng sử dụng**

**Quản trị viên (Admin)**: Quản lý khoa, ngành, chuyên ngành, bộ môn, giảng viên, sinh viên, và đợt đăng ký.

**Giảng viên**: Ra đề tài, duyệt đề tài, quản lý tiến độ, và chấm điểm.

**Sinh viên**: Đăng ký đề tài, lập nhóm, nộp báo cáo tiến độ, và xem điểm.

**Giáo vụ**: Tổng hợp danh sách, thống kê kết quả, và hỗ trợ quản lý quy trình.

**1.4. Định nghĩa, từ viết tắt, và thuật ngữ**

**ASP.NET Core Web API**: Framework để xây dựng API RESTful.

**Next.js**: Framework React cho ứng dụng web full-stack.

**Clean Architecture**: Kiến trúc phần mềm tách biệt logic nghiệp vụ khỏi phụ thuộc bên ngoài.

**PDM**: Physical Data Model (Mô hình dữ liệu vật lý).

**GVHD**: Giảng viên hướng dẫn.

**GVPB**: Giảng viên phản biện.

**JWT**: JSON Web Token, dùng để xác thực người dùng.

**2. Tổng quan hệ thống**

**2.1. Mô tả hệ thống**

Hệ thống là một ứng dụng web full-stack, sử dụng mô hình client-server, với các thành phần chính:

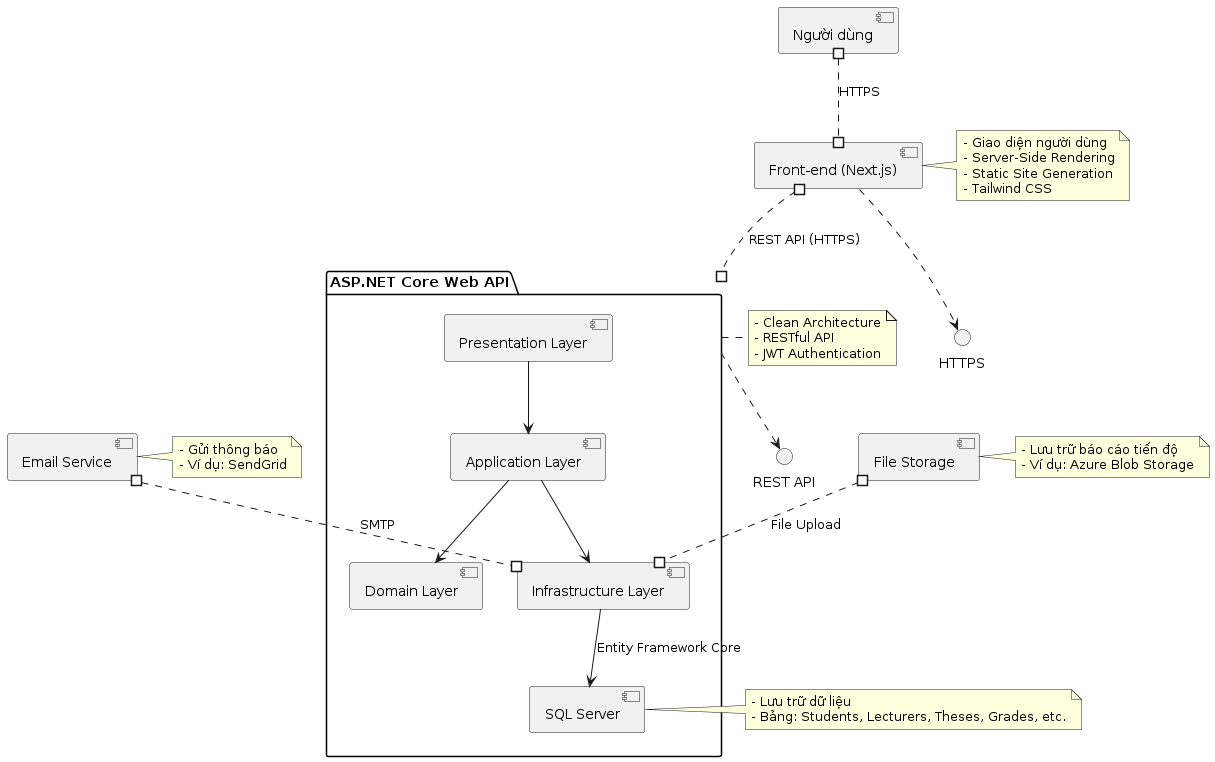
* **Front-end (Next.js)**: Cung cấp giao diện người dùng cho quản trị viên, giảng viên, và sinh viên.
* **Back-end (ASP.NET Core Web API)**: Xử lý logic nghiệp vụ, quản lý dữ liệu, và cung cấp API RESTful.
* **Cơ sở dữ liệu (SQL Server)**: Lưu trữ thông tin về sinh viên, giảng viên, đề tài, tiến độ, và điểm số.

**2.2. Kiến trúc tổng quan**

Hệ thống áp dụng **Clean Architecture**, chia thành các tầng:

* **Presentation Layer**: Giao diện người dùng (Next.js).
* **Application Layer**: Chứa logic nghiệp vụ, điều phối các yêu cầu (Use Cases).
* **Domain Layer**: Chứa các thực thể (Entities) và quy tắc nghiệp vụ cốt lõi.
* **Infrastructure Layer**: Tương tác với cơ sở dữ liệu (SQL Server) và các dịch vụ bên ngoài (email, file storage).

**Sơ đồ kiến trúc tổng quan**:



**3. Các yếu tố thiết kế**

**3.1. Giả định và phụ thuộc**

**Giả định**:

* Người dùng có kết nối internet ổn định.
* Sinh viên và giảng viên có tài khoản trên hệ thống để đăng nhập.
* Máy chủ chạy Windows/Linux hỗ trợ .NET Core 8 và SQL Server 2022.

**Phụ thuộc**:

* Thư viện Entity Framework Core để tương tác với SQL Server.
* Gói NuGet cho xác thực JWT (Microsoft.AspNetCore.Authentication.JwtBearer).
* Dịch vụ email (SendGrid) để gửi thông báo.
* Thư viện Next.js cho Server-Side Rendering và Static Site Generation.

**3.2. Ràng buộc chung**

Hệ thống phải sử dụng **ASP.NET Core 8** và **Next.js 14**.

Cơ sở dữ liệu phải là **SQL Server 2022**.

Hệ thống phải hỗ trợ tối đa 1000 người dùng đồng thời với thời gian phản hồi dưới 2 giây.

Giao diện phải responsive, hỗ trợ các thiết bị desktop và mobile.

Hệ thống phải tuân thủ các quy định bảo mật (HTTPS, mã hóa mật khẩu).

**3.3. Mục tiêu và nguyên tắc**

**Tính đơn giản (KISS)**: Giao diện và quy trình sử dụng dễ hiểu.

**Tính module hóa**: Mỗi thành phần (API, front-end, database) có thể phát triển độc lập.

**Tính bảo trì**: Áp dụng Clean Architecture để dễ dàng bảo trì và mở rộng.

**Hiệu suất**: Tối ưu thời gian tải trang và xử lý yêu cầu.

**3.4. Phương pháp phát triển**

**Phương pháp**: Agile với các sprint 2 tuần.

**Công cụ**:

* **IDE**: Visual Studio 2022, Visual Studio Code.
* **Quản lý mã nguồn**: Git, GitHub.
* **Kiểm thử**: Postman (API), Jest (front-end unit test).
* **Triển khai**: Docker, Azure (hoặc máy chủ nội bộ).

**4. Chiến lược kiến trúc**

**4.1. Sử dụng Clean Architecture**

**Lý do**: Tách biệt logic nghiệp vụ khỏi phụ thuộc bên ngoài, giúp hệ thống dễ bảo trì và mở rộng.

**Cách triển khai**:

* **Domain Layer**: Chứa các thực thể (Student, Lecturer, Thesis) và giao diện repository.
* **Application Layer**: Chứa các Use Case (RegisterThesis, GradeThesis) và DTOs (Data Transfer Objects).
* **Infrastructure Layer**: Triển khai repository sử dụng Entity Framework Core.
* **Presentation Layer**: API controllers (ASP.NET Core) và giao diện Next.js.

**4.2. API RESTful**

**Lý do**: Cung cấp giao diện chuẩn để front-end và các ứng dụng bên ngoài tương tác.

**Cách triển khai**:

* API sử dụng chuẩn RESTful với các phương thức HTTP (GET, POST, PUT, DELETE).
* Sử dụng JWT để xác thực và phân quyền.

**4.3. Server-Side Rendering (Next.js)**

**Lý do**: Cải thiện SEO và hiệu suất tải trang ban đầu.

**Cách triển khai**: Sử dụng SSR cho các trang động (danh sách đề tài, thống kê) và SSG cho các trang tĩnh (trang giới thiệu).

**4.4. Quản lý cơ sở dữ liệu**

**Lý do**: SQL Server cung cấp hiệu suất cao và khả năng quản lý dữ liệu phức tạp.

**Cách triển khai**:

* Sử dụng Entity Framework Core với Code-First approach.
* Áp dụng các index và stored procedures để tối ưu truy vấn.

**4.5. Xác thực và phân quyền**

**Lý do**: Đảm bảo chỉ người dùng được phép truy cập các chức năng tương ứng.

**Cách triển khai**:

* Sử dụng JWT cho xác thực.
* Phân quyền dựa trên vai trò (Admin, Lecturer, Student).

**5. Kiến trúc hệ thống**

**5.1. Phân hệ chính**

Hệ thống được chia thành các phân hệ sau:

* **Admin Module**:

Quản lý khoa, ngành, chuyên ngành, bộ môn, giảng viên, sinh viên, đợt đăng ký, và nhóm.

* **Thesis Proposal Module**:

Hỗ trợ giảng viên ra đề tài, yêu cầu duyệt, và duyệt đề tài.

* **Thesis Registration Module**:

Hỗ trợ sinh viên tạo nhóm và đăng ký đề tài.

* **Progress Management Module**:

Quản lý tiến độ thực hiện đề tài bởi GVHD và sinh viên.

* **Grading Module**:

Hỗ trợ chấm điểm bởi GVHD, GVPB, và hội đồng.

* **Statistics Module**:

Thống kê kết quả khóa luận của khoa CNTT.

**5.2. Tương tác giữa các phân hệ**

**Admin Module** cung cấp dữ liệu cơ bản (sinh viên, giảng viên, đề tài) cho các phân hệ khác.

**Thesis Proposal Module** tạo và duyệt đề tài, cung cấp danh sách đề tài cho **Thesis Registration Module**.

**Thesis Registration Module** gửi thông tin nhóm và đề tài đã đăng ký đến **Progress Management Module**.

**Progress Management Module** lưu trữ tiến độ và báo cáo, cung cấp dữ liệu cho **Grading Module**.

**Grading Module** gửi điểm số đến **Statistics Module** để tổng hợp.

**6. Chính sách và chiến thuật**

**6.1. Quy ước mã hóa**

**Back-end**:

* Sử dụng PascalCase cho tên class và phương thức.
* Tên API endpoints theo chuẩn RESTful (ví dụ: /api/theses, /api/students).

**Front-end**:

* Sử dụng camelCase cho biến và hàm.
* Tổ chức component theo thư mục (ví dụ: /components/admin, /pages/thesis).

**6.2. Công cụ và thư viện**

**Back-end**:

* Entity Framework Core: ORM để tương tác với SQL Server.
* AutoMapper: Ánh xạ giữa DTOs và Entities.
* MediatR: Xử lý các lệnh và truy vấn trong Application Layer.

**Front-end**:

* Tailwind CSS: Thiết kế giao diện responsive.
* Axios: Gửi yêu cầu HTTP đến API.
* React Query: Quản lý trạng thái và caching dữ liệu.

**6.3. Kiểm thử**

**Unit Test**: Sử dụng xUnit cho back-end và Jest cho front-end.

**Integration Test**: Kiểm thử API với Postman và kiểm thử tích hợp với SQL Server.

**End-to-End Test**: Sử dụng Cypress để kiểm thử luồng người dùng.

**6.4. Triển khai**

Hệ thống được đóng gói trong container Docker.

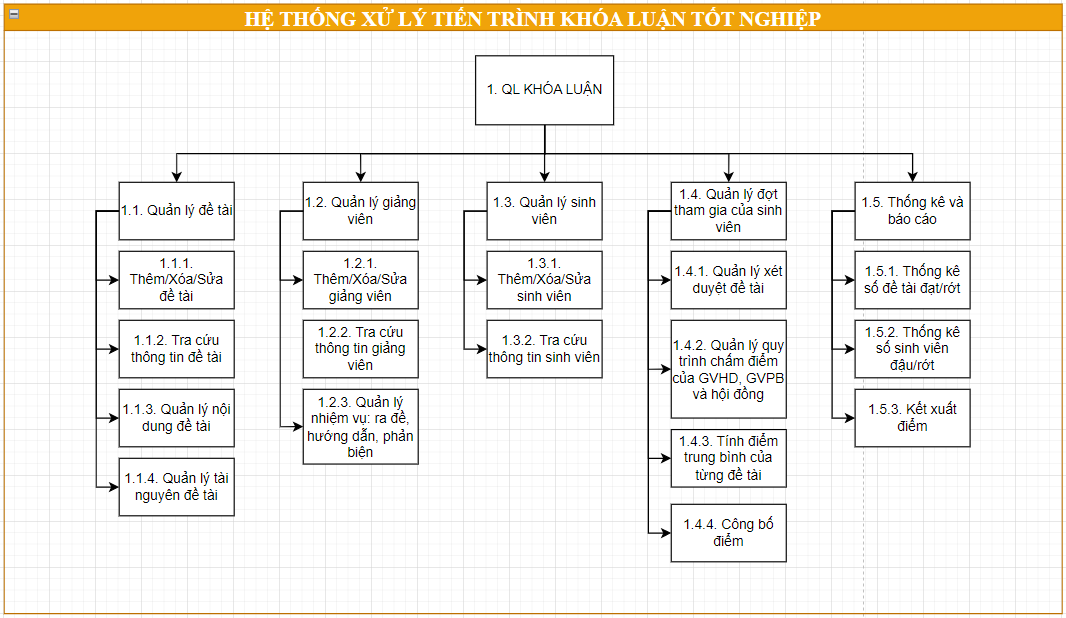
Triển khai trên Azure hoặc máy chủ nội bộ với Nginx làm reverse proxy.

**7. Thiết kế hệ thống chi tiết**

**7.1. Mô hình phân cấp chức năng**

**Mô hình phân cấp chức năng** (Functional Decomposition Diagram) mô tả cách các chức năng chính của hệ thống được chia nhỏ thành các chức năng con.

**Sơ đồ phân cấp chức năng**:



**7.2. Thiết kế PDM (Physical Data Model)**

**PDM** mô tả cấu trúc cơ sở dữ liệu vật lý, bao gồm các bảng, cột, khóa chính, khóa ngoại, và các ràng buộc.

**Sơ đồ PDM**:



**Ràng buộc**:

* Khóa chính: Đảm bảo tính duy nhất cho mỗi bản ghi.
* Khóa ngoại: Liên kết giữa các bảng (ví dụ: FacultyId trong Departments tham chiếu đến Faculties).
* Ràng buộc kiểm tra: Score trong Grades phải nằm trong khoảng 0-10.
* Index: Tạo index trên Email trong Students và Lecturers để tối ưu truy vấn.

**7.2.1 Mô tả các lớp có trong hệ thống**

**Bảng dữ liệu TEACHER**

| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Null / Not Null** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- | --- |
| 1 | ID | CHAR(10) | Not null | PRIMARY KEY | Mã định danh giáo viên |
| 2 | Internal Code | VARCHAR(10) | Null |  | Mã nội bộ |
| 3 | Name | NVARCHAR(50) | Null |  | Tên giáo viên |
| 4 | Gender | NVARCHAR(3) | Null |  | Giới tính giáo viên |
| 5 | Date of Birth | DATE | Null |  | Ngày sinh giáo viên |
| 6 | PhoneNumber | CHAR(10) | Null |  | Số điện thoại giáo viên |
| 7 | Academic Title | NVARCHAR(30) | Null |  | Học vị |
| 8 | Email | NVARCHAR(50) | Null |  | Email giáo viên |
| 9 | Degree | NVARCHAR(50) | Null |  | Bằng cấp |
| 10 | Departmentld | NVARCHAR(50) | Null |  | Khoa giảng dạy |
| 11 | Date Created | DATE | Null |  | Ngày cấp |
| 12 | Created By | NVARCHAR(50) | Null |  | Tổ chức cấp |
| 13 | Last Modified Date | DATE | Null |  | Ngày sửa đổi cuối cùng |
| 14 | Last Modified By | NVARCHAR(50) | Null |  | Ngày sửa đổi cuối cùng bởi |
| 15 | Is Deleted | BOOLEAN | Not Null |  | Đã xóa |
| 16 | Type | NVARCHAR(20) | Null |  | Thuộc loại |
| 17 | User Id | INT | Null |  | Mã người dùng |

Bảng 1 TEACHER

**Bảng dữ liệu STUDENTS**

| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Null / Not Null** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- | --- |
| 1 | ID | CHAR(10) | Not null | PRIMARY KEY | Mã định danh học sinh |
| 2 | Internal Code | VARCHAR(10) | Null |  | Mã nội bộ học sinh |
| 3 | Name | NVARCHAR(50) | Null |  | Tên học sinh |
| 4 | Date of Birth | DATE | Null |  | Ngày sinh học sinh |
| 5 | Gender | NVARCHAR(3) | Null |  | Giới tính học sinh |
| 6 | Class | VARCHAR(10) | Null |  | Lớp |
| 7 | PhoneNumber | CHAR(10) | Null |  | Số điện thoại học sinh |
| 8 | Email | NVARCHAR(50) | Null |  | Email học sinh |
| 9 | Majors | VARCHAR(50) | Null |  | Chuyên ngành |
| 10 | Date Created | DATE | Null |  | Ngày cấp |
| 11 | Created By | NVARCHAR(50) | Null |  | Tổ chức cấp |
| 12 | Last Modified Date | DATE | Null |  | Ngày sửa đổi cuối cùng |

Bảng 2 STUDENTS

**Bảng dữ liệu JOBS**

| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Null / Not Null** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- | --- |
| 1 | ID | CHAR(10) | Not null | PRIMARY KEY | Mã định danh công việc |
| 2 | Name | NVARCHAR(50) | Null |  | Tên công việc |
| 3 | Instructions | NVARCHAR(50) | Null |  | Hướng dẫn sử dụng |
| 4 | Due | DATE | Null |  | Ngày quá hạn |
| 5 | Files | NVARCHAR(30) | Null |  | Tệp đính kèm |
| 6 | Teacher Id | INT | Null |  | Mã giáo viên |
| 7 | Thesis Id | INT | Null |  | Mã khóa luận |
| 8 | Date Created | DATE | Null |  | Ngày tạo |
| 9 | Created By | NVARCHAR(30) | Null |  | Người tạo |
| 10 | LastModified Date | DATE | Null |  | Ngày chỉnh sửa cuối |
| 11 | LastModified By | NVARCHAR(30) | Null |  | Người chỉnh sửa cuối |
| 12 | Is Deleted | BOOLEAN | Not Null |  | Đã xóa |

Bảng 3 JOBS

**Bảng dữ liệu REPORT SCHEDULES**

| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Null / Not Null** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- | --- |
| 1 | ID | CHAR(10) | Not null | PRIMARY KEY | Mã định danh lịch |
| 2 | Time Start | DATETIME | Not Null |  | Thời gian bắt đầu |
| 3 | Location | VARCHAR(50) | Not Null |  | Địa điểm |
| 4 | Type | VARCHAR(50) | Not Null |  | Loại hình |
| 5 | Thesis ID | INT | Null |  | Mã khóa luận |
| 6 | Teacher ID | INT | Null |  | Mã giáo viên |
| 7 | Date Created | DATE | Null |  | Ngày tạo |
| 8 | Created By | NVARCHAR(50) | Null |  | Người tạo |
| 9 | LastModified Date | DATE | Null |  | Ngày chỉnh sửa cuối cùng |
| 10 | LastModified By | NVARCHAR(50) | Null |  | Người chỉnh sửa cuối cùng |
| 11 | Is Deleted | BOOLEAN | Not Null |  | Đã xóa |
| 12 | TimeEnd | DATETIME | Null |  | Thời gian kết thúc |

Bảng 4 REPORT SCHEDULES

**Bảng dữ liệu FACULITIES**

| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Null / Not Null** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- | --- |
| 1 | ID | CHAR(10) | Not null | PRIMARY KEY | Mã định danh khoa |
| 2 | Interal Code | VARCHAR(10) | Not Null |  | Mã nội bộ khoa |
| 3 | Name | NVARCHAR(50) | Not Null |  | Tên khoa |
| 4 | Adress | NVARCHAR(50) | Not Null |  | Địa chỉ văn phòng |
| 5 | PhoneNumber | VARCHAR(10) | Null |  | Số điện thoại |
| 6 | Email | NVARCHAR(30) | Null |  | Địa chỉ Email |
| 7 | Date Created | DATE | Null |  | Ngày tạo |
| 8 | Created By | NVARCHAR(50) | Null |  | Người tạo |
| 9 | LastModified Date | DATE | Null |  | Ngày chỉnh sửa cuối cùng |
| 10 | LastModified By | NVARCHAR(50) | Null |  | Người chỉnh sửa cuối cùng |
| 11 | Is Deleted | BOOLEAN | Not Null |  | Đã xóa |

Bảng 5 FACULITIES

**Bảng dữ liệu THESISS**

| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Null / Not Null** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- | --- | --- | --- |
| 1 | ID | CHAR(10) | Not null | PRIMARY KEY | Mã định danh khoa |
| 2 | Interal Code | VARCHAR(10) | Null |  | Mã nội bộ khoa |
| 3 | Name | NVARCHAR(50) | Not Null |  | Tên khoa |
| 4 | Summary | TEXT | Null |  | Mô tả nội dung khóa luận |
| 5 | Min Quantity | INT | Null |  | Số lượn sinh viên tối thiểu |
| 6 | Max Quantity | INT | Null |  | Số lượng sinh viên tối đa |
| 7 | Status | VARCHAR(30) | Null |  | Trạng thái |
| 8 | Type | VARCHAR(30) | Null |  | Loại hình |
| 9 | Lecturer Thesis ID | INT | Not Null |  | Mã giảng viên hướng dẫn |
| 10 | Proposed Student ID | INT | Not Null |  | Mã sinh viên đề xuất |
| 11 | Date Created | DATE | Null |  | Ngày tạo |
| 12 | Created By | NVARCHAR(50) | Null |  | Người tạo |
| 13 | LastModified Date | DATE | Null |  | Ngày chỉnh sửa cuối cùng |
| 14 | LastModified By | NVARCHAR(50) | Null |  | Người chỉnh sửa cuối cùng |
| 15 | Is Deleted | BOOLEAN | Not Null |  | Đã xóa |

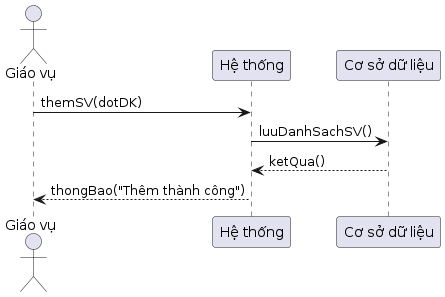
Bảng 6 THESISS

**7.3. Thiết kế Diagram**

**Mô hình Diagram** bao gồm các biểu đồ tuần tự (Sequence Diagrams) cho từng chức năng chính.

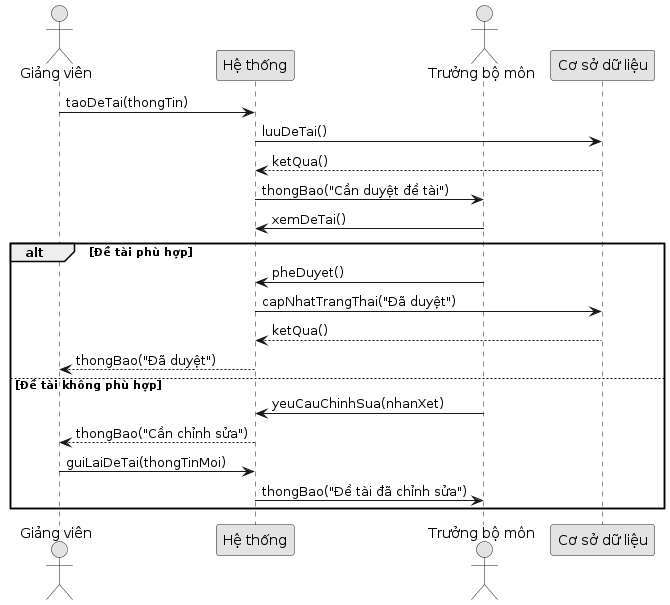
**7.3.1. Quản lý sinh viên tham gia khóa luận**

**Biểu đồ tuần tự**:



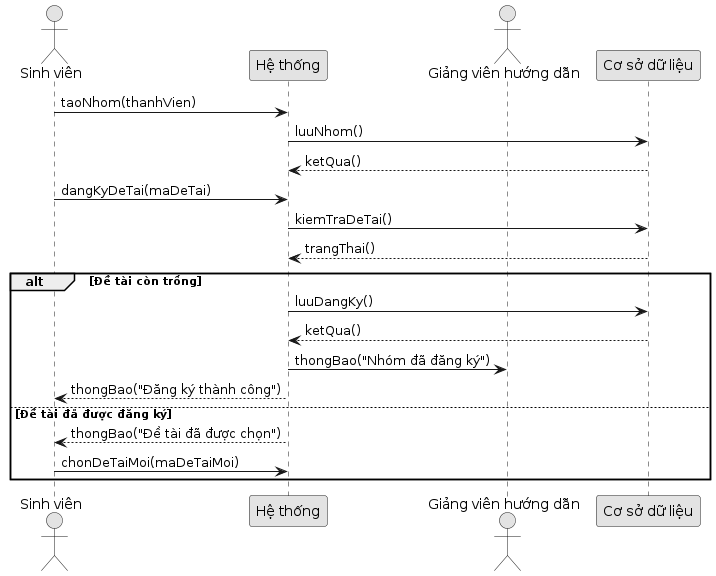
**7.3.2. Ra đề và kiểm duyệt đề tài**

**Biểu đồ tuần tự**:



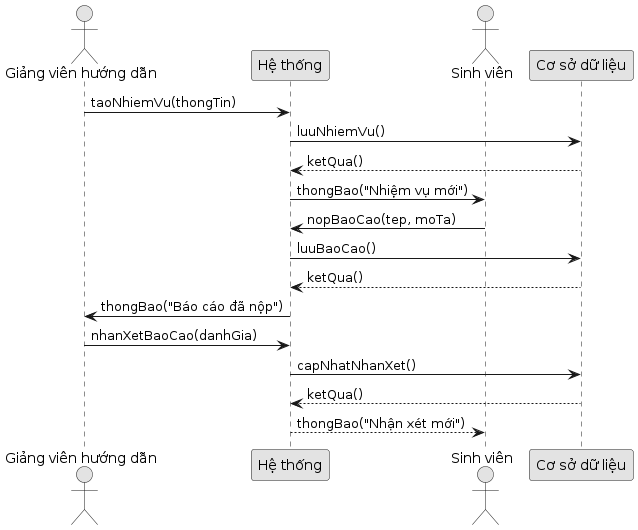
**7.3.3. Đăng ký đề tài khóa luận**

**Biểu đồ tuần tự**:



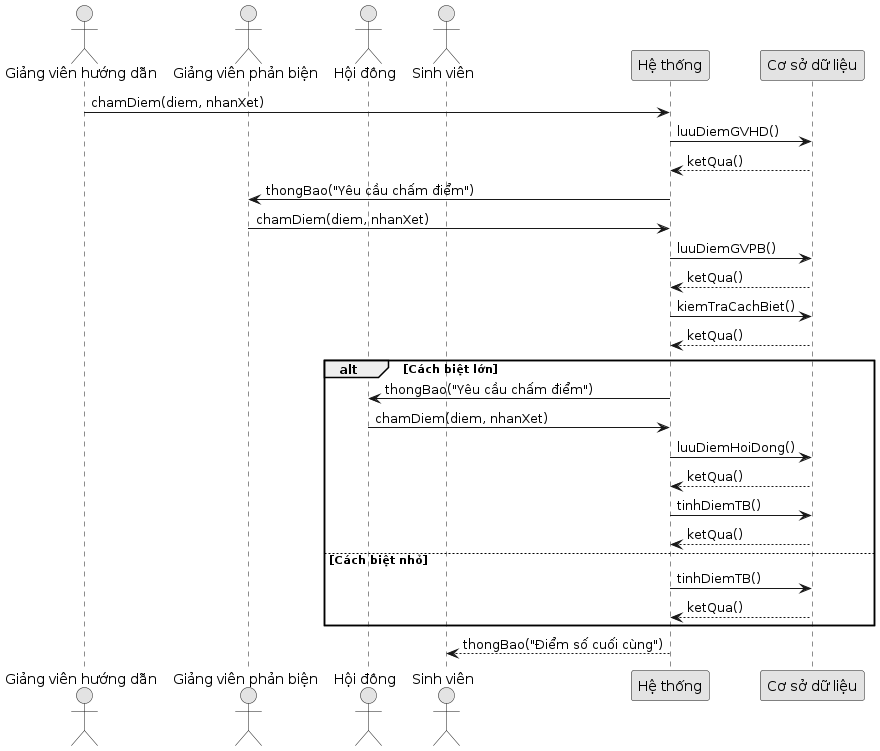
**7.3.4. Quản lý tiến độ thực hiện đề tài**

**Biểu đồ tuần tự**:



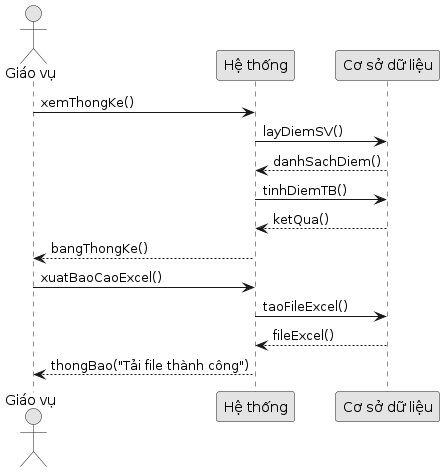
**7.3.5. Chấm điểm đề tài**

**Biểu đồ tuần tự**:



**7.3.6. Thống kê kết quả**

**Biểu đồ tuần tự**:



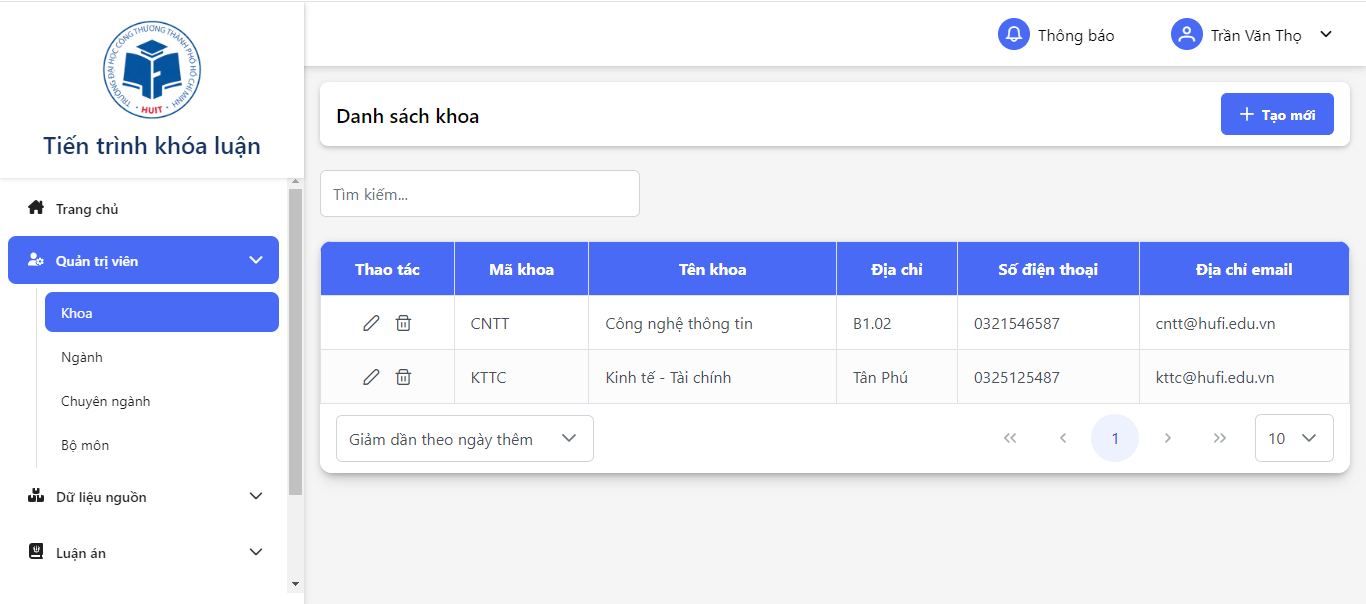
**7.4. Thiết kế giao diện**

Dựa trên mô tả trong đồ án, giao diện được thiết kế sử dụng **Next.js** với **Tailwind CSS** để đảm bảo tính responsive và thân thiện với người dùng.

**7.4.1. Giao diện Admin**

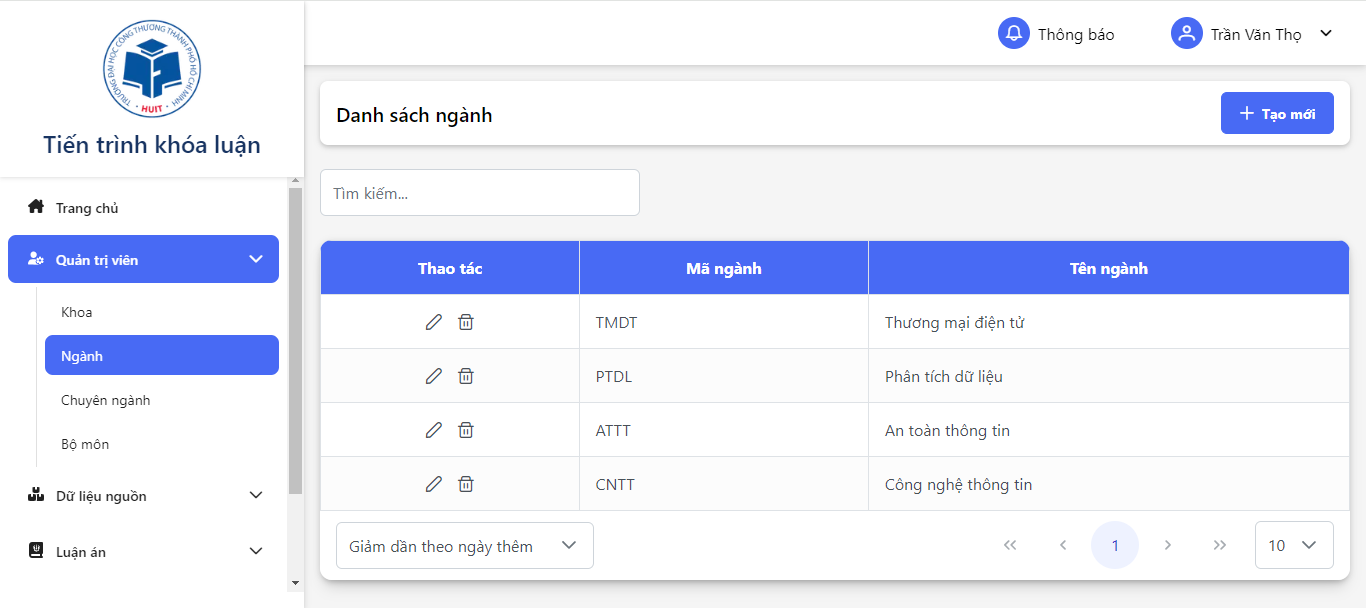
**Quản lý Khoa**:

* **Mô tả**: Hiển thị danh sách khoa, cho phép thêm, sửa, xóa.
* **Component**: FacultyManagement.tsx
* **API**: GET /api/faculties, POST /api/faculties, PUT /api/faculties/{id}, DELETE /api/faculties/{id}.
* **Giao diện**: Bảng dữ liệu với các cột (ID, Tên khoa, Mô tả) và nút hành động.



**Quản lý Ngành, Chuyên ngành, Bộ môn, Giảng viên, Sinh viên, Đợt đăng ký, Nhóm**:

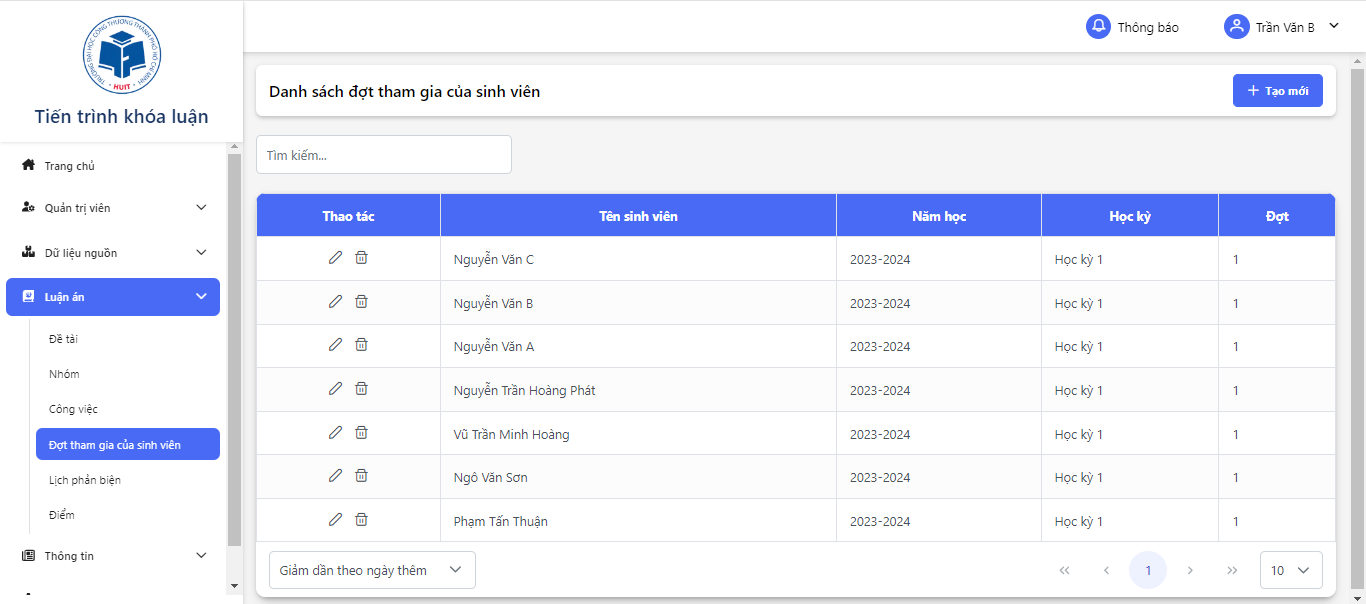
Tương tự giao diện quản lý khoa, với các bảng dữ liệu và form nhập liệu tương ứng.

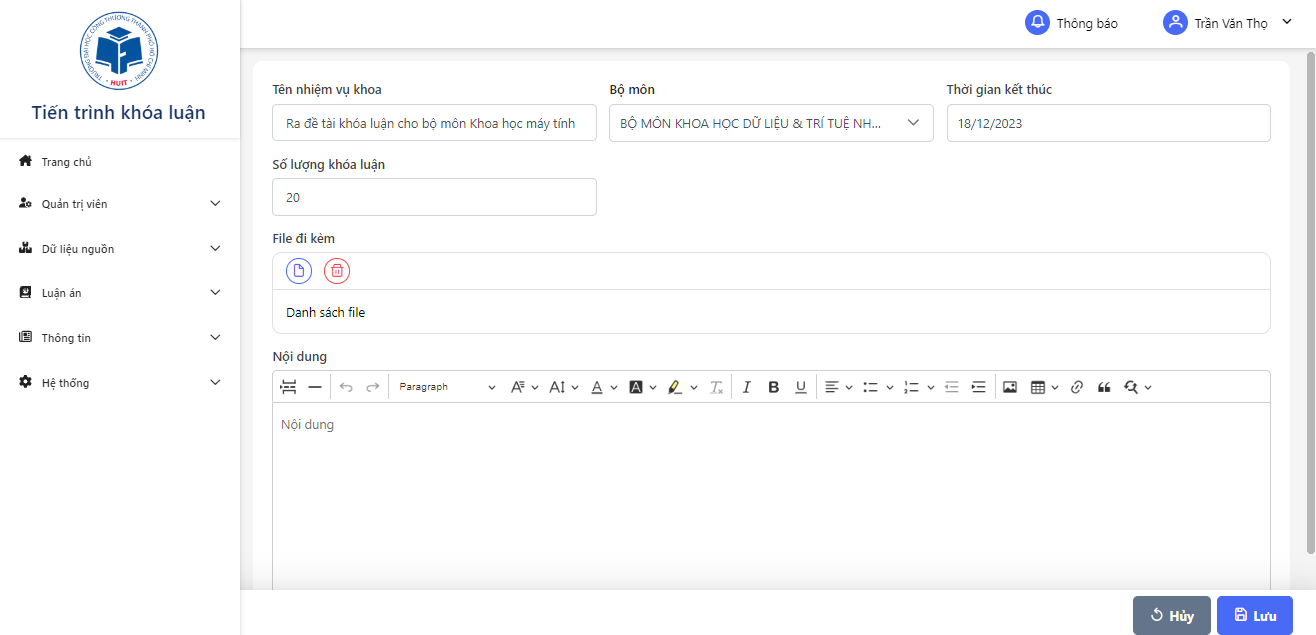


**7.4.2. Giao diện Giảng viên**

**Ra đề tài**:

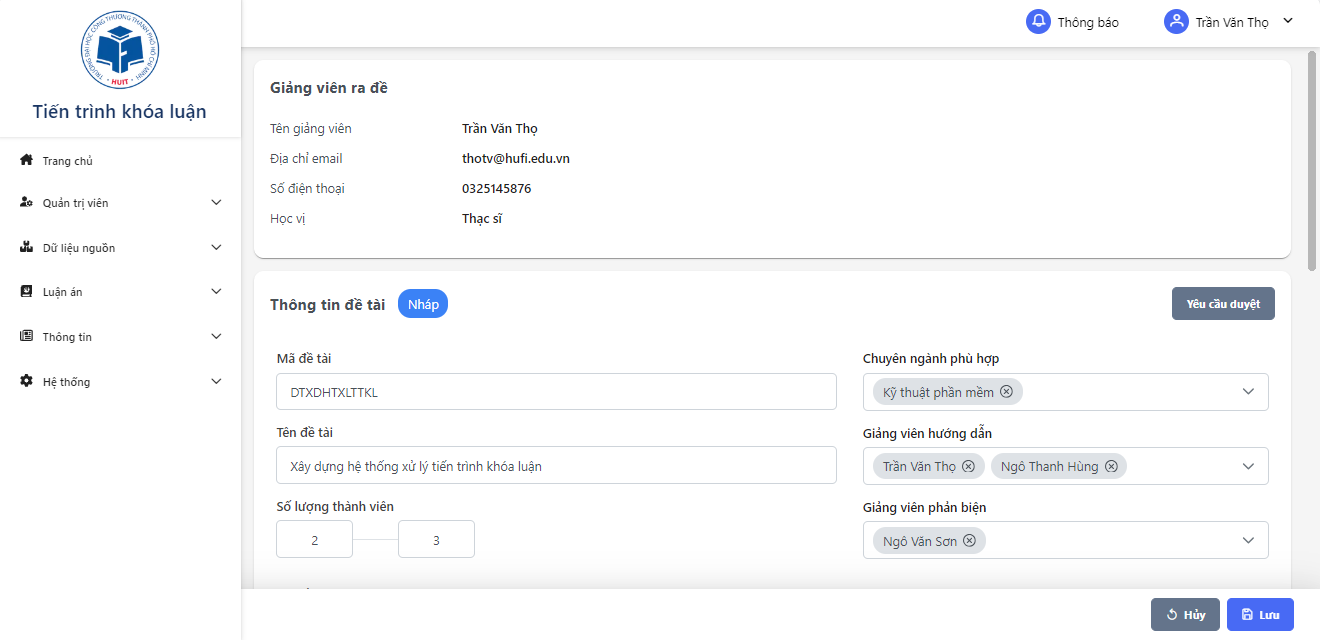
* **Mô tả**: Form nhập thông tin đề tài (tên, mô tả, số lượng sinh viên, chuyên ngành).
* **Component**: ThesisProposalForm.tsx
* **API**: POST /api/theses, PUT /api/theses/{id}/request-approval.

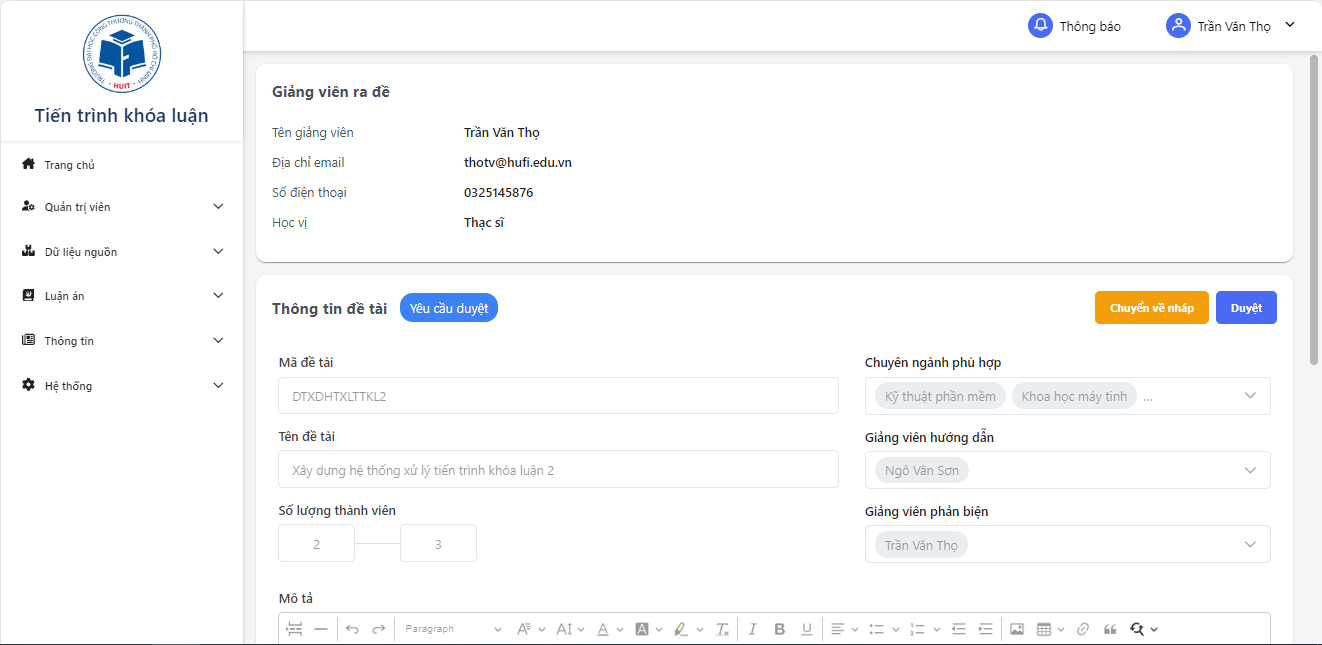




**Duyệt đề tài**:

* **Mô tả**: Danh sách đề tài chờ duyệt, với nút duyệt/từ chối và form nhập nhận xét.
* **Component**: ThesisApproval.tsx
* **API**: GET /api/theses/pending, PUT /api/theses/{id}/approve, PUT /api/theses/{id}/reject.

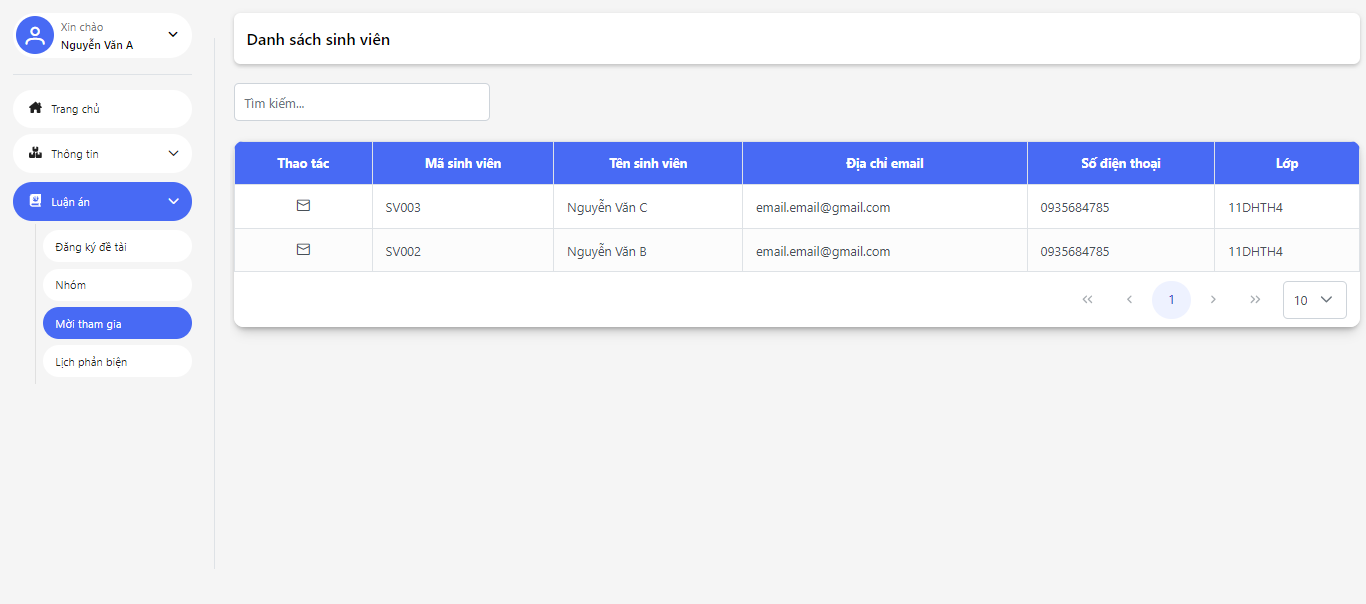




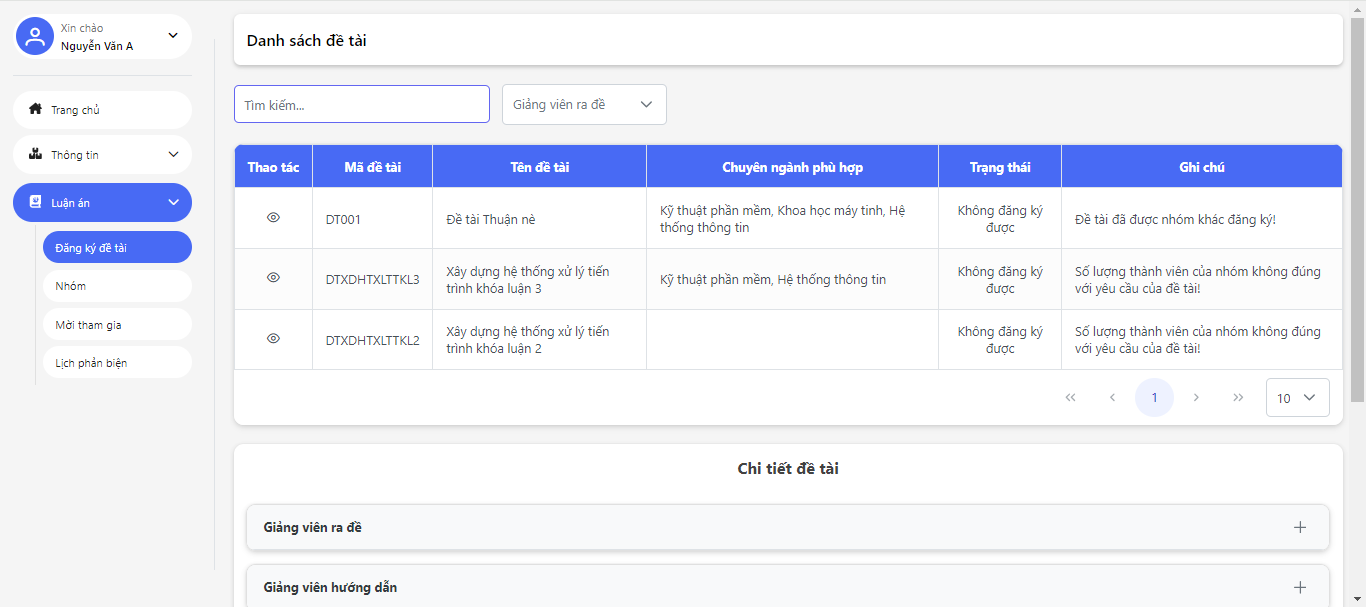
**7.4.3. Giao diện Sinh viên**

**Tạo nhóm**:

* **Mô tả**: Form mời thành viên và tạo nhóm.
* **Component**: GroupCreation.tsx
* **API**: POST /api/groups, POST /api/group-members.

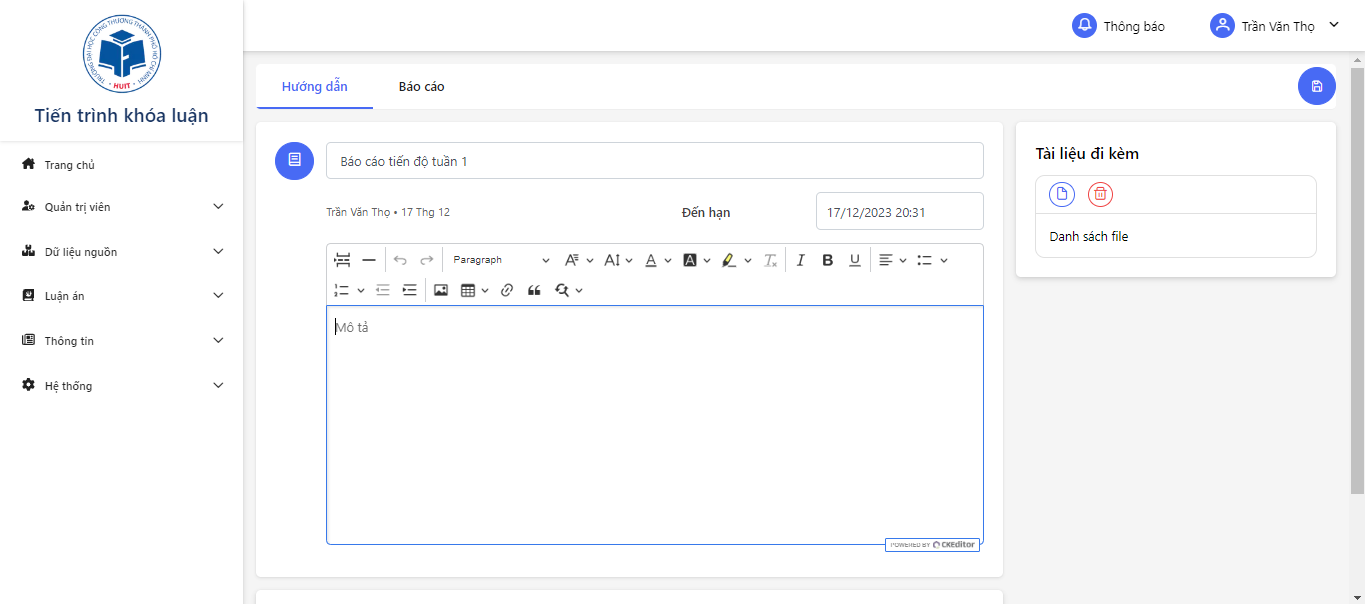


**Đăng ký đề tài**:

* **Mô tả**: Danh sách đề tài có thể đăng ký, với bộ lọc theo chuyên ngành.
* **Component**: ThesisRegistration.tsx
* **API**: GET /api/theses/available, POST /api/thesis-registrations.

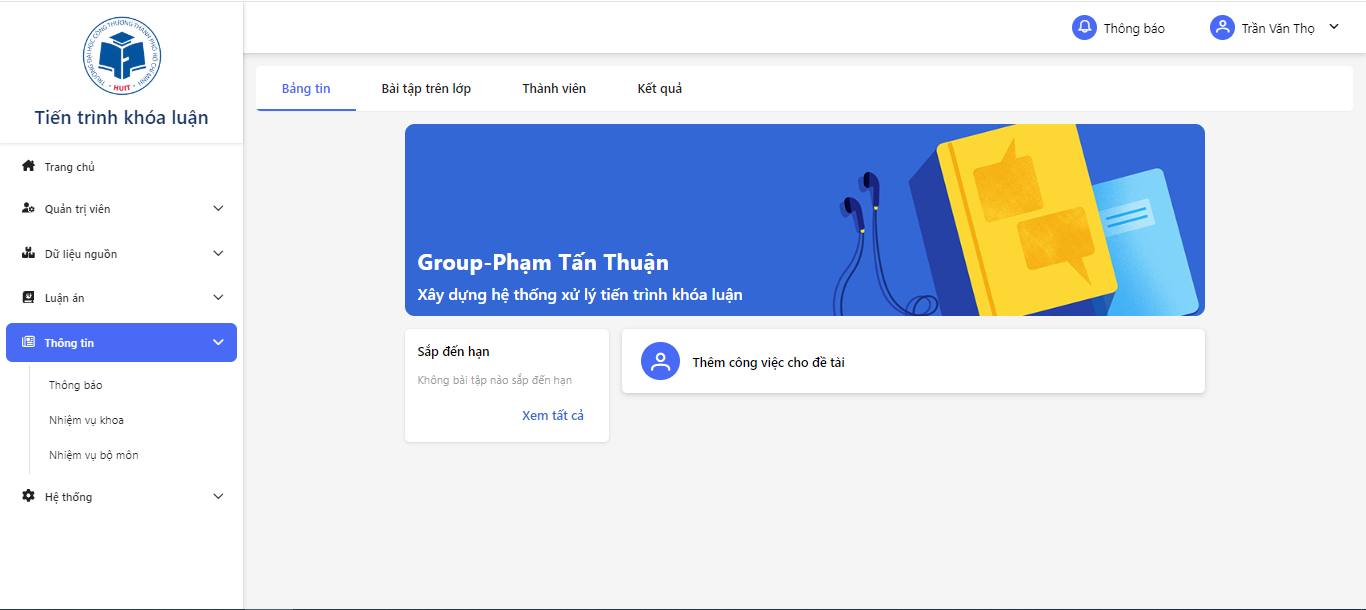
**7.4.4. Giao diện Quản lý Tiến độ**

**Tạo nhiệm vụ (GVHD)**:

* **Mô tả**: Form nhập tiêu đề, mô tả, và hạn nộp nhiệm vụ.
* **Component**: TaskCreation.tsx
* **API**: POST /api/tasks.  
  

**Nộp báo cáo (Sinh viên)**:

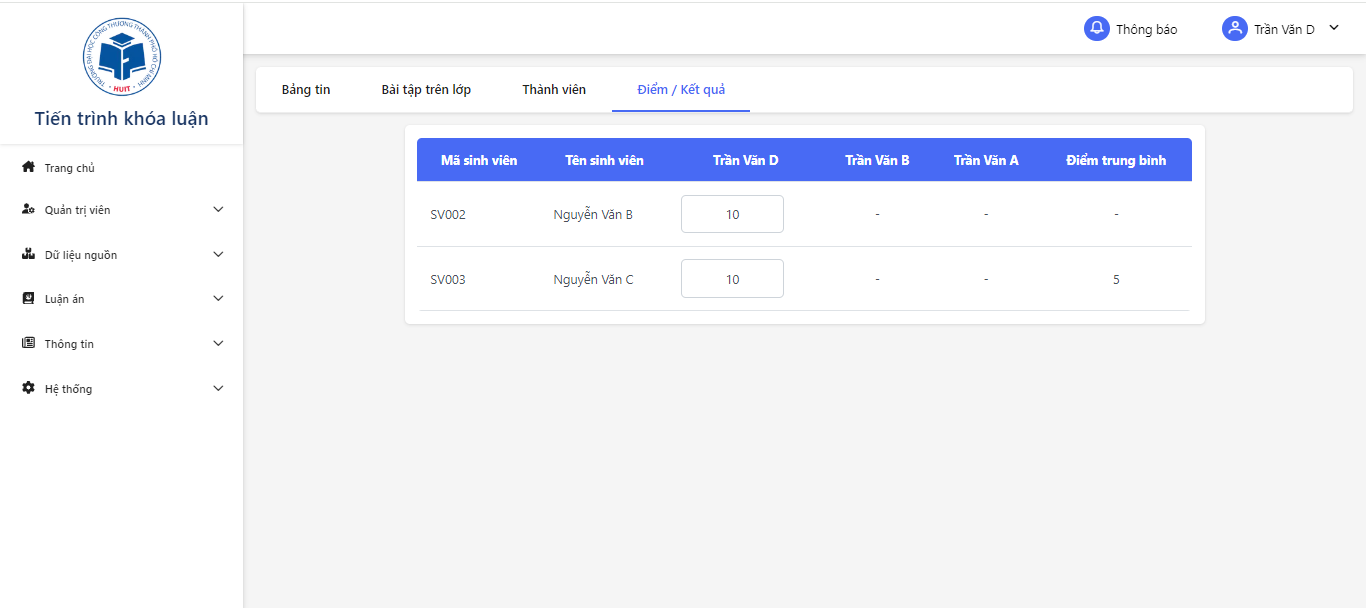
* **Mô tả**: Form tải lên file và nhập mô tả.
* **Component**: TaskSubmission.tsx
* **API**: POST /api/task-submissions.



**7.4.5. Giao diện Chấm điểm**

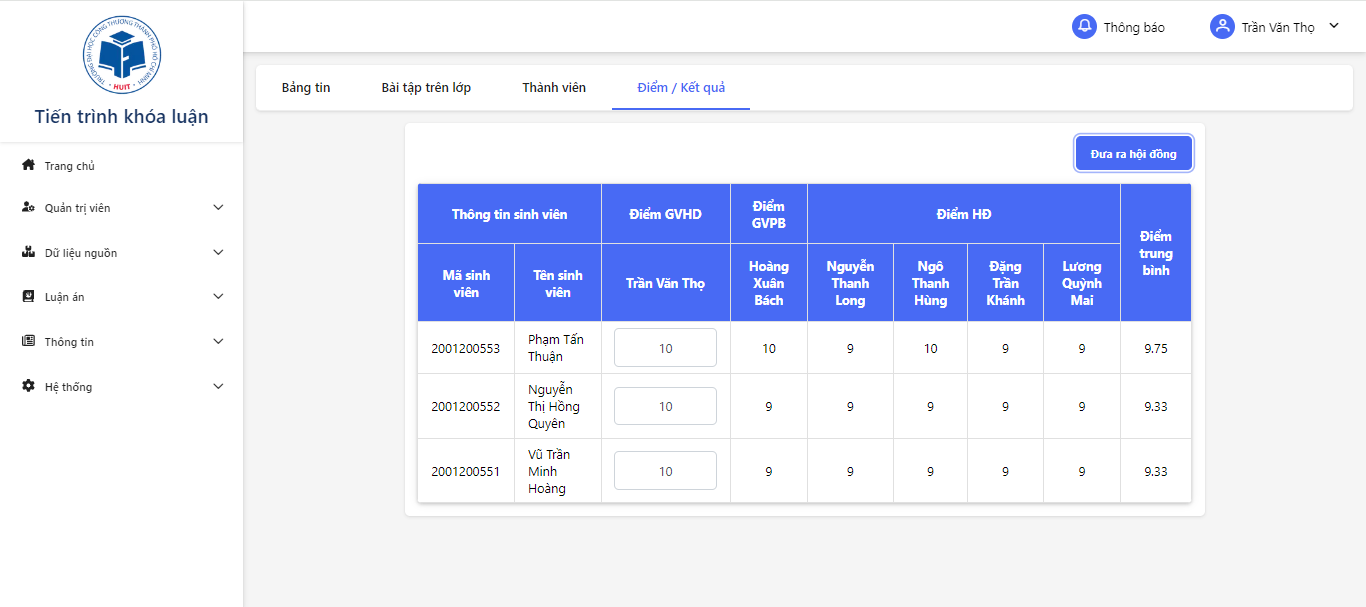
**Chấm điểm (GVHD/GVPB)**:

* **Mô tả**: Form nhập điểm số và nhận xét cho từng sinh viên.
* **Component**: ThesisGrading.tsx
* **API**: POST /api/grades.



**Đưa ra hội đồng**:

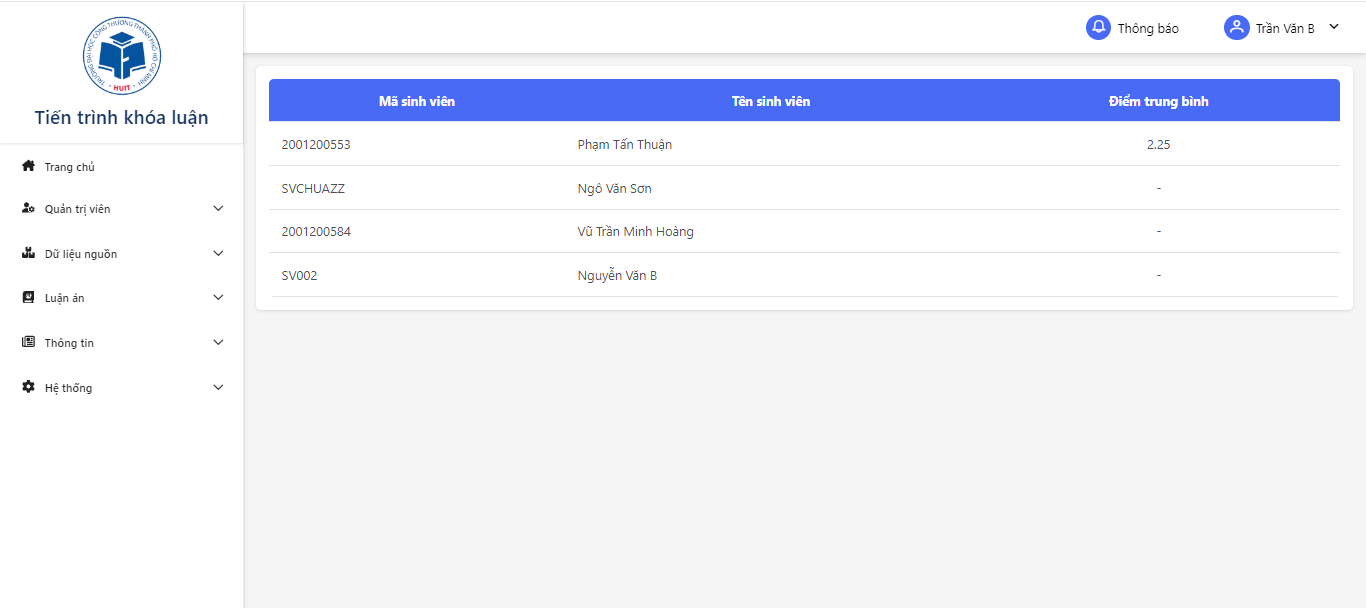
* **Mô tả**: Giao diện hiển thị điểm số và nút yêu cầu đưa ra hội đồng.
* **Component**: CouncilAssignment.tsx
* **API**: POST /api/councils.



**7.4.6. Giao diện Thống kê**

**Thống kê kết quả**:

* **Mô tả**: Bảng hiển thị điểm trung bình của sinh viên, với nút xuất Excel.
* **Component**: StatisticsDashboard.tsx
* **API**: GET /api/statistics, GET /api/statistics/export.



**8. Từ điển thuật ngữ**

* **Đợt đăng ký**: Khoảng thời gian mà sinh viên có thể đăng ký đề tài.
* **Nhóm**: Tập hợp sinh viên cùng thực hiện một đề tài.
* **GVHD**: Giảng viên hướng dẫn, chịu trách nhiệm giám sát tiến độ đề tài.
* **GVPB**: Giảng viên phản biện, đánh giá đề tài độc lập.
* **Hội đồng**: Nhóm giảng viên chấm điểm đề tài khi có sự cách biệt điểm số.

**9. Tài liệu tham khảo**

Martin, R. C. (2012, August 13). The Clean Architecture. The Clean Code Blog. Truy cập ngày 19 tháng 4 năm 2025, từ https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html

Microsoft. (n.d.). ASP.NET Core documentation. Truy cập ngày 19 tháng 4 năm 2025, từ https://learn.microsoft.com/en-us/aspnet/core/

Microsoft. (n.d.). SQL Server documentation. Truy cập ngày 19 tháng 4 năm 2025, từ https://learn.microsoft.com/en-us/sql/

Vercel. (n.d.). Next.js Documentation. Truy cập ngày 19 tháng 4 năm 2025, từ https://nextjs.org/docs