|  |  |
| --- | --- |
|  |  |
| ĐINH VĂN DUY | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------** |
|  |
| ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC |
| NGÀNH CÔNG NGHỆ THÔNG TIN |
| **TÊN ĐỀ TÀI: XÂY DỰNG HỆ THỐNG HỖ TRỢ TỰ HỌC VÀ ĐÁNH GIÁ TRỰC TUYẾN** |
|  |
|  |
| **GVHD : TS. Đặng Trọng Hợp** |
| CÔNG NGHỆ THÔNG TIN | **Sinh viên : Đinh Văn Duy** |
| **Mã số sinh viên : 2021604611** |
|  |
|  |
|  |
| Hà Nội – Năm 2025 |
|  |

# MỤC LỤC

[MỤC LỤC i](#_Toc198308585)

[DANH MỤC TỪ VIẾT TẮT v](#_Toc198308586)

[DANH MỤC HÌNH ẢNH v](#_Toc198308587)

[DANH MỤC BẢNG BIỂU viii](#_Toc198308588)

[MỞ ĐẦU ix](#_Toc198308589)

[CHƯƠNG 1. TỔNG QUAN HỆ THỐNG HỖ TRỢ TỰ HỌC VÀ ĐÁNH GIÁ TRỰC TUYẾN 1](#_Toc198308590)

[1.1. Giới thiệu về hệ thống hỗ trợ tự học và đánh giá trực tuyến. 1](#_Toc198308591)

[1.2. Mô tả bài toán 1](#_Toc198308592)

[1.3. Mô hình và công nghệ Web 3](#_Toc198308593)

[1.3.1. Kiến trúc phân lớp (Layered Architecture) 3](#_Toc198308594)

[1.3.2. Mô hình kiến trúc MVC 4](#_Toc198308595)

[1.3.3. Ngôn ngữ lập trình C# và ASP.NET 5](#_Toc198308596)

[1.3.4. Cơ sở dữ liệu MS SQL 6](#_Toc198308597)

[1.3.5. Ngôn ngữ lập trình ReactJs 6](#_Toc198308598)

[1.4. Kết luận chương 1 7](#_Toc198308599)

[CHƯƠNG 2. PHÂN TÍCH THIÊT KẾ HỆ THỐNG 9](#_Toc198308600)

[2.1. Yêu cầu chức năng và phi chức năng 9](#_Toc198308601)

[2.1.1. Yêu cầu chức năng đối với sinh viên 9](#_Toc198308602)

[2.1.2. Yêu cầu đối với giảng viên 9](#_Toc198308603)

[2.1.3. Các yêu cầu phi chức năng 9](#_Toc198308604)

[2.2. Mô hình hóa chức năng 10](#_Toc198308605)

[2.2.1. Danh sách các Actor 10](#_Toc198308606)

[2.2.2. Biểu đồ Use Case 11](#_Toc198308607)

[2.3. Mô tả chi tiết các Use Case 11](#_Toc198308608)

[2.3.1. Use case đăng nhập hệ thống 11](#_Toc198308609)

[2.3.2. Use case xem danh sách học phần 13](#_Toc198308610)

[2.3.3. Use case truy cập bài học 14](#_Toc198308611)

[2.3.4. Use case làm bài tập lập trình 14](#_Toc198308612)

[2.3.5. Use case gửi code và chấm bài 15](#_Toc198308613)

[2.3.6. Use case theo dõi tiến độ học tập 17](#_Toc198308614)

[2.3.7. Use case tạo bài học và tài liệu dạy 18](#_Toc198308615)

[2.3.8. Use case tạo bài tập lập trình 19](#_Toc198308616)

[2.3.9. Use case xem kết quả bài làm của sinh viên 20](#_Toc198308617)

[2.3.10. Use case quản lý bài học và bài tập 22](#_Toc198308618)

[2.3.11. Use case quản lý sinh viên 23](#_Toc198308619)

[2.3.12. Use case thống kê và báo cáo 24](#_Toc198308620)

[2.3.13. Use case quản lý học phần 25](#_Toc198308621)

[2.4. Thiết kế chi tiết 28](#_Toc198308622)

[2.4.1. Mô hình hóa điều kiện 28](#_Toc198308623)

[2.4.2. Biểu đồ trình tự use case đăng nhập 29](#_Toc198308624)

[2.4.3. Biểu đồ trình tự use case làm bài tập lập trình 30](#_Toc198308625)

[2.4.4. Biểu đồ trình tự use case tạo bài tập lập trình 31](#_Toc198308626)

[2.4.5. Biểu đồ trình tự use case theo dõi kết quả học 32](#_Toc198308627)

[2.4.6. Biểu đồ lớp chi tiết 33](#_Toc198308628)

[2.5. Thiết kế cơ sở dữ liệu hợp lý 34](#_Toc198308629)

[2.5.1. Bảng Subjects(học phần) 34](#_Toc198308630)

[2.5.2. Bảng Student (sinh viên) 35](#_Toc198308631)

[2.5.3. Bảng ClassStudents (Lớp học chứa student) 35](#_Toc198308632)

[2.5.4. Bảng Classes (Bảng lớp học) 36](#_Toc198308633)

[2.5.5. Bảng Teachers (Giảng viên) 36](#_Toc198308634)

[2.5.6. Bảng Lessons (Bài học) 36](#_Toc198308635)

[2.5.7. Bảng LessonContents (Nội dung bài học) 37](#_Toc198308636)

[2.5.8. Bảng TestCases (Kiểm thử) 37](#_Toc198308637)

[2.5.9. Bảng CodingExercises (Bài tập code) 38](#_Toc198308638)

[2.5.10. Bảng Submission (Bài nộp) 38](#_Toc198308639)

[2.6. Thiết kế hình dung màn hình 39](#_Toc198308640)

[2.6.1. Thiết kế giao diện trang chủ 39](#_Toc198308641)

[2.6.2. Thiết kế hình dung màn hình chọn lớp học phần 39](#_Toc198308642)

[2.6.3. Thiết kế hình dung màn hình chọn bài học 40](#_Toc198308643)

[2.6.4. Thiết kế hình dung màn hình chọn bài tập thực hành 41](#_Toc198308644)

[2.6.5. Thiết kế hình dung màn h ình xem điểm số 42](#_Toc198308645)

[2.6.6. Thiết kế hình dung màn hình xem chi tiết điểm số 43](#_Toc198308646)

[2.7. Kết luận chương 2 44](#_Toc198308647)

[CHƯƠNG 3. CÀI ĐẶT VÀ KIỂM THỬ 45](#_Toc198308648)

[3.1. Giao diện và chức năng phía sinh viên 45](#_Toc198308649)

[3.1.1. Trang chủ 45](#_Toc198308650)

[3.1.2. Trang danh sách bài học 45](#_Toc198308651)

[3.1.3. Trang nội dung bài học 46](#_Toc198308652)

[3.1.4. Trang làm bài tập lập trình 46](#_Toc198308653)

[3.1.5. Trang xem điểm số 47](#_Toc198308654)

[3.1.6. Trang xem chi tiết điểm số 47](#_Toc198308655)

[3.2. Giao diện và chức năng phía người quản trị 47](#_Toc198308656)

[3.2.1. Trang chủ của Giảng viên 47](#_Toc198308657)

[3.2.2. Trang quản lý lớp học phần 48](#_Toc198308658)

[3.2.3. Trang quản lý học sinh 48](#_Toc198308659)

[3.2.4. Trang danh sách bài học trong lớp học phần 49](#_Toc198308660)

[3.2.5. Trang nội dung bài học 49](#_Toc198308661)

[3.2.6. Trang quản lý bài tập thực hành 50](#_Toc198308662)

[3.2.7. Trang tạo mới bài tập thực hành 50](#_Toc198308663)

[3.3. Kiểm thử 51](#_Toc198308664)

[3.3.1. Phạm vi kiểm thử 51](#_Toc198308665)

[3.3.2. Test case 52](#_Toc198308666)

[3.3.3. Báo cáo kiểm thử 53](#_Toc198308667)

[3.4. Tổng kết chương 3 53](#_Toc198308668)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 55](#_Toc198308669)

[TÀI LIỆU THAM KHẢO 57](#_Toc198308670)

# DANH MỤC TỪ VIẾT TẮT

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Từ viết tắt** | **Tiếng Anh** | **Nghĩa tiếng việt** |
| 1 | MVC | Model-View-Controller | Kiến trúc phần mềm Model-View-Controller |
| 2 | API | Application Programming Interface | Giao diện lập trình ứng dụng |
| 3 | UC | Use case | Ca sử dụng |
| 4 | ReactJS | React Javascript | Thư viện Javascript dùng để xây dựng giao diện |
| 5 | Asp.net | Active Server Page .NET | Trang máy chủ động trên nền tảng .NET |
| 6 | SEO | Search Engine Optimization | Là quá trình tối ưu hóa website để nâng cao thứ hạng tìm kiếm trên Google |

# DANH MỤC HÌNH ẢNH

[Hình 1. 1: Kiến trúc phần mềm Model-View-Controller 4](#_Toc198303049)

[Hình 2. 1: Biểu đồ Use case tổng quát của hệ thống 11](#_Toc198303050)

[Hình 2. 2: Biểu đồ lớp thiết kế hệ thống 28](#_Toc198303051)

[Hình 2. 3: Biểu đồ trình tự UC Đăng nhập 29](#_Toc198303052)

[Hình 2. 4: Biểu đồ trình tự UC Làm bài tập lập trình 30](#_Toc198303053)

[Hình 2. 5: Biểu đồ trình tự UC Tạo bài tập lập trình 31](#_Toc198303054)

[Hình 2. 6: Biểu đồ trình tự UC Theo dõi kết quả học 32](#_Toc198303055)

[Hình 2. 7: Biểu đồ lớp chi tiết. 33](#_Toc198303056)

[Hình 2. 8: Biểu đồ thực thể liên kết mức vật lý 34](#_Toc198303057)

[Hình 2. 9: Thiết kế giao diện trang chủ 39](#_Toc198303058)

[Hình 2. 10: Thiết kế hình dung màn hình chọn lớp học phần 40](#_Toc198303059)

[Hình 2. 11: Thiết kế hình dung màn hình chọn bài học 41](#_Toc198303060)

[Hình 2. 12: Thiết kế hình dung màn hình làm bài tập thực hành 42](#_Toc198303061)

[Hình 2. 13: Thiết kế hình dung màn hình xem điểm số 43](#_Toc198303062)

[Hình 2. 14: Thiết kế hình dung màn hình xem chi tiết điểm số 44](#_Toc198303063)

[Hình 3. 1: Giao diện trang chủ 45](#_Toc198303064)

[Hình 3. 2: Giao diện trang danh sách bài học*.* 45](#_Toc198303065)

[Hình 3. 3: Giao diện trang nội dung bài học. 46](#_Toc198303066)

[Hình 3. 4: Giao diện trang bài tập lập trình. 46](#_Toc198303067)

[Hình 3. 5: Giao diện trang xem điểm số 47](#_Toc198303068)

[Hình 3. 6: Giao diện trang xem điểm số chi tiết. 47](#_Toc198303069)

[Hình 3. 7: Giao diện trang chủ của giảng viên 48](#_Toc198303070)

[Hình 3. 8: Giao diện quản lý lớp học phần 48](#_Toc198303071)

[Hình 3. 9: Giao diện trang quản lý học sinh 49](#_Toc198303072)

[Hình 3. 10: Giao diện danh sách bài học 49](#_Toc198303073)

[Hình 3. 11: Giao diện nội dung bài học 50](#_Toc198303074)

[Hình 3. 12: Giao diện quản lý bài tập thực hành 50](#_Toc198303075)

[Hình 3. 13: Giao diện tạo bài tập thực hành 51](#_Toc198303076)

[Hình 3. 14: Test case làm bài tập thực hành 52](#_Toc198303077)

[Hình 3. 15: Test case hiển thị nội dung bài học 52](#_Toc198303078)

[Hình 3. 16: Test case xem điểm của sinh viên 52](#_Toc198303079)

# DANH MỤC BẢNG BIỂU

[Bảng 2. 1: Danh sách các Actor 10](#_Toc198303106)

[Bảng 2. 2: Use case Đăng nhập hệ thống 11](#_Toc198303107)

[Bảng 2. 3: Use case Xem danh sách học phần 13](#_Toc198303108)

[Bảng 2. 4: Use case truy cập bài học 14](#_Toc198303109)

[Bảng 2. 5:Use case làm bài tập lập trình 14](#_Toc198303110)

[Bảng 2. 6: Use case gửi code và chấm bài 15](#_Toc198303111)

[Bảng 2. 7: Use case Theo dõi tiến độ học tập 17](#_Toc198303112)

[Bảng 2. 8: Use case tạo bài học và tài liệu dạy 18](#_Toc198303113)

[Bảng 2. 9: Use case tạo bài tập lập trình 19](#_Toc198303114)

[Bảng 2. 10: Use case Xem kết quả của Sinh viên 20](#_Toc198303115)

[Bảng 2. 11: Use case Quản lý bài học và bài tập 22](#_Toc198303116)

[Bảng 2. 12: Use case quản lý sinh viên 23](#_Toc198303117)

[Bảng 2. 13: : Use case thống kê và báo cáo 24](#_Toc198303118)

[Bảng 2. 14: Use case quản lý học phần 25](#_Toc198303119)

[Bảng 2. 15: Bảng Subjects 34](#_Toc198303120)

[Bảng 2. 16: Bảng Student 35](#_Toc198303121)

[Bảng 2. 17: Bảng ClassStudent 35](#_Toc198303122)

[Bảng 2. 18: Bảng Classes 36](#_Toc198303123)

[Bảng 2. 19: Bảng Teachers 36](#_Toc198303124)

[Bảng 2. 20: Bảng Lessons 36](#_Toc198303125)

[Bảng 2. 21: Bảng LessonContents 37](#_Toc198303126)

[Bảng 2. 22: Bảng TestCases 37](#_Toc198303127)

[Bảng 2. 23: Bảng Submission 38](#_Toc198303128)

# MỞ ĐẦU

**Lý do chọn đề tài**

Hiện tại là thời kỳ chuyển đổi số rất mạnh mẽ, giáo dục trực tuyến là một phần không thể thiếu trong việc nâng cao chất lượng đào tạo, chất lượng học tập dành cho các sinh viên. Các trường đại học cần triển khai mạnh mẽ các hệ thống hỗ trợ học trực tuyến cho sinh viên. Tuy nhiên hầu hết các hệ thống chỉ hỗ trợ các bài tập dạng trắc nghiệm hoặc tự luận, chưa có tính năng thực hành lập trình trực tuyến trên nền tảng. Điều này gây khó khăn cho các sinh viên trong việc học tập các môn có liên quan tới lập trình, do phải sử dụng phần mềm (IDE) bên ngoài để viết mã nguồn.

Nhắm vào những yếu tố đó, em muốn phát triển mạnh mẽ một tính năng có thể trực tiếp lập trình trên chính trang website của trường đại học Công Nghiệp Hà Nội của mình. Chính vì vậy đề tài “Hệ thống hỗ trợ tự học và đánh giá trực tuyến” được thực hiện nhằm xây dựng một nền tảng tự học trực tuyến có tích hợp tính năng lập trình trực tuyến trên website.

**1. Mục tiêu nghiên cứu**

* + Xây dựng nền tảng học tập trực tuyến, cho phép sinh viên truy cập các học phần, bài học lý thuyết và thực hành lập trình.
  + Tích hợp trình soạn thảo code trực tuyến, hỗ trợ nhiều ngôn ngữ lập trình.
  + Cho phép sinh viên viết, chạy thử và nộp bài tập ngay trên nền tảng.
  + Tự động chấm điểm bài tập lập trình.
  + Hỗ trợ giảng viên trong quản lý và đánh giá sinh viên.
  + Ứng dụng công nghệ hiện đại để phát triển hệ thống: Docker, ASP.NET Core Web API, ReactJs,..

**2. Đối tượng nghiên cứu**

**-** Website học kết hợp của trường đại học Công Nghiệp Hà Nội.

- Sinh viên, giảng viên.

- Mô hình học tập kết hợp.

**3. Phạm vi nghiên cứu**

- Sinh viên: Sử dụng hệ thống để học lý thuyết, thực hành lập trình trực tuyến và nộp bài tập.

- Giảng viên: Quản lý bài học, tạo bài tập lập trình, theo dõi và đánh giá kết quả của sinh viên.

- Hệ thống chấm điểm tự động: Xử lý bài làm của sinh viên, kiểm tra tính đúng đắn của mã nguồn dựa trên test case, thời gian xử lý,..

- Lập trình trực tuyến: Hỗ trợ sinh viên viết code, chạy thử và nộp bài ngay trên trình duyệt mà không cần cài đặt phần mềm bên ngoài.

- Học trực tuyến: Sinh viên có thể truy cập học phần, bài giảng lý thuyết và tài liệu tham khảo.

**4. Nội dung nghiên cứu**

* + Nghiên cứu mô hình học kết hợp của trường Đại học Công Nghiệp Hà Nội.
  + Phân tích và thiết kế xây dựng theo các học phần.
  + Xác định các yêu cầu chức năng và phi chức năng của website.
  + Đề xuất mô hình hệ thống phù hợp với nhu cầu của sinh viên và giảng viên.
  + Thiết kế cơ sở dữ liệu để lưu trữ thông tin học phần, bài giảng, bài tập và kết quả nộp bài.
  + Xây dựng API trên nền tảng ASP.NET Core Web API.
  + Phát triển Frontend bằng ReactJS, tối ưu hiệu suất.
  + Tích hợp trình soạn thảo code online hỗ trợ nhiều ngôn ngữ lập trình để sinh viên có thể chọn ngôn ngữ tùy thích.

**5. Bố cục bản báo cáo**

Báo cáo gồm các phần sau đây:

# TỔNG QUAN HỆ THỐNG HỖ TRỢ TỰ HỌC VÀ ĐÁNH GIÁ TRỰC TUYẾN

## Giới thiệu về hệ thống hỗ trợ tự học và đánh giá trực tuyến.

Trong thời đại công nghệ số phát triển mạnh mẽ, việc ứng dụng công nghệ thông tin vào giáo dục ngày càng trở nên phổ biến và cần thiết. Đặc biệt là trong bối cảnh dịch bệnh Covid-19 vừa qua, hình thức học tập trực tuyến đã dần thay thế cho phương thức học truyền thống tại nhiều cơ sở giáo dục. Tuy nhiên, các hệ thống học trực tuyến hiện nay đa số chỉ hỗ trợ trình bày nội dung bài học mà thiếu đi sự tương tác thực hành, đặc biệt là đối với các môn học lập trình, đòi hỏi người học phải thường xuyên luyện tập và nhận được phản hồi tức thì.

Trước nhu cầu đó, đề tài "Xây dựng hệ thống hỗ trợ tự học và đánh giá trực tuyến" ra đời với mục tiêu xây dựng một nền tảng học trực tuyến tích hợp chức năng lập trình online và hệ thống chấm điểm tự động. Hệ thống không chỉ giúp sinh viên có thể tiếp cận nội dung bài học ở bất kỳ đâu, mà còn cho phép họ thực hành lập trình trực tiếp trên trình duyệt, nộp bài và nhận kết quả đánh giá ngay lập tức.

Với mô hình tổ chức theo từng học phần, mỗi học phần gồm nhiều bài học lý thuyết đi kèm bài tập thực hành, hệ thống này hỗ trợ giảng viên dễ dàng tạo nội dung giảng dạy và bài kiểm tra, đồng thời theo dõi tiến độ học tập của sinh viên. Qua đó, hệ thống góp phần nâng cao hiệu quả tự học và tạo môi trường học tập chủ động, linh hoạt cho người học.

## Mô tả bài toán

Bài toán đặt ra là xây dựng một hệ thống học tập và đánh giá trực tuyến, đặc biệt dành cho các môn học lập trình, cho phép người học không chỉ tiếp cận nội dung bài giảng mà còn có thể thực hành và được chấm điểm một cách tự động. Cụ thể, hệ thống cần đáp ứng các yêu cầu sau:

:Tổ chức học phần và bài học: Các môn học (học phần) được chia thành các bài học, mỗi bài học bao gồm phần lý thuyết (tài liệu, video, văn bản) và bài tập lập trình đi kèm.

Soạn thảo và thực hành lập trình trực tuyến: Hệ thống tích hợp trình soạn thảo code (code editor) cho phép sinh viên viết, chạy thử và nộp bài trực tiếp trên nền tảng, không cần cài đặt phần mềm bên ngoài.

Chấm điểm tự động: Bài làm của sinh viên được kiểm tra bằng hệ thống test case, trả về kết quả đúng/sai, điểm số, thời gian thực thi và số lượng test case vượt qua.

**Quản lý người dùng**

Sinh viên: Có thể đăng nhập, tham gia học phần, làm bài tập và theo dõi kết quả học tập.

Giảng viên: Có thể tạo học phần, bài học, bài tập, theo dõi tiến độ sinh viên và xem thống kê kết quả học tập.

Hệ thống quản trị: Quản trị viên có quyền quản lý học phần, người dùng, bài học, bài tập, kết quả và các báo cáo thống kê hệ thống.

**Yêu cầu về kỹ thuật**

Backend sử dụng ASP.NET Core Web API để cung cấp dữ liệu và xử lý nghiệp vụ.

Frontend sử dụng ReactJS để xây dựng giao diện tương tác hiện đại, dễ sử dụng.

Cơ sở dữ liệu SQL Server để lưu trữ toàn bộ thông tin người dùng, học phần, bài tập, kết quả.

Tích hợp Docker + Judge để xử lý và đánh giá bài nộp một cách an toàn, cách ly môi trường thực thi.

Mục tiêu của hệ thống là xây dựng một nền tảng học tập toàn diện, trực quan, dễ sử dụng, đồng thời nâng cao tính chủ động của sinh viên trong việc học và thực hành lập trình, tạo ra môi trường học tập thực tế và hiệu quả hơn trong giáo dục đại học.

## Mô hình và công nghệ Web

### Kiến trúc phân lớp (Layered Architecture)

Kiến trúc phân lớp là một mô hình tổ chức phần mềm phổ biến, trong đó hệ thống được chia thành nhiều lớp riêng biệt, mỗi lớp đảm nhiệm một vai trò cụ thể trong quy trình xử lý dữ liệu và nghiệp vụ. Việc chia nhỏ hệ thống thành các lớp giúp tăng tính mô-đun, dễ bảo trì, mở rộng và kiểm thử phần mềm.

Trong đề tài này, hệ thống được xây dựng theo kiến trúc phân lớp gồm các lớp chính sau:

**Lớp Presentation (Giao diện người dùng)**

Là phần giao tiếp giữa người dùng và hệ thống.

Trong đề tài, lớp này được phát triển bằng ReactJS, hiển thị các giao diện học tập, bài học, bài tập và cho phép sinh viên tương tác như gửi code, xem kết quả, theo dõi tiến độ.

**Lớp Business Logic (Xử lý nghiệp vụ)**

Chịu trách nhiệm xử lý các logic nghiệp vụ như tính điểm, kiểm tra kết quả bài làm, xử lý các yêu cầu từ frontend.

Lớp này được tổ chức thành các service trong ASP.NET Core, đảm bảo rằng mọi thao tác từ giao diện đều đi qua lớp xử lý nghiệp vụ.

**Lớp Data Access (Truy cập dữ liệu)**

Chịu trách nhiệm giao tiếp với cơ sở dữ liệu, thực hiện các thao tác thêm, sửa, xóa, truy vấn dữ liệu.

Lớp này sử dụng mô hình Repository và UnitOfWork để tách biệt logic truy cập dữ liệu khỏi logic xử lý nghiệp vụ, đồng thời tăng khả năng tái sử dụng và kiểm thử.

**Lớp Infrastructure (Hạ tầng)**

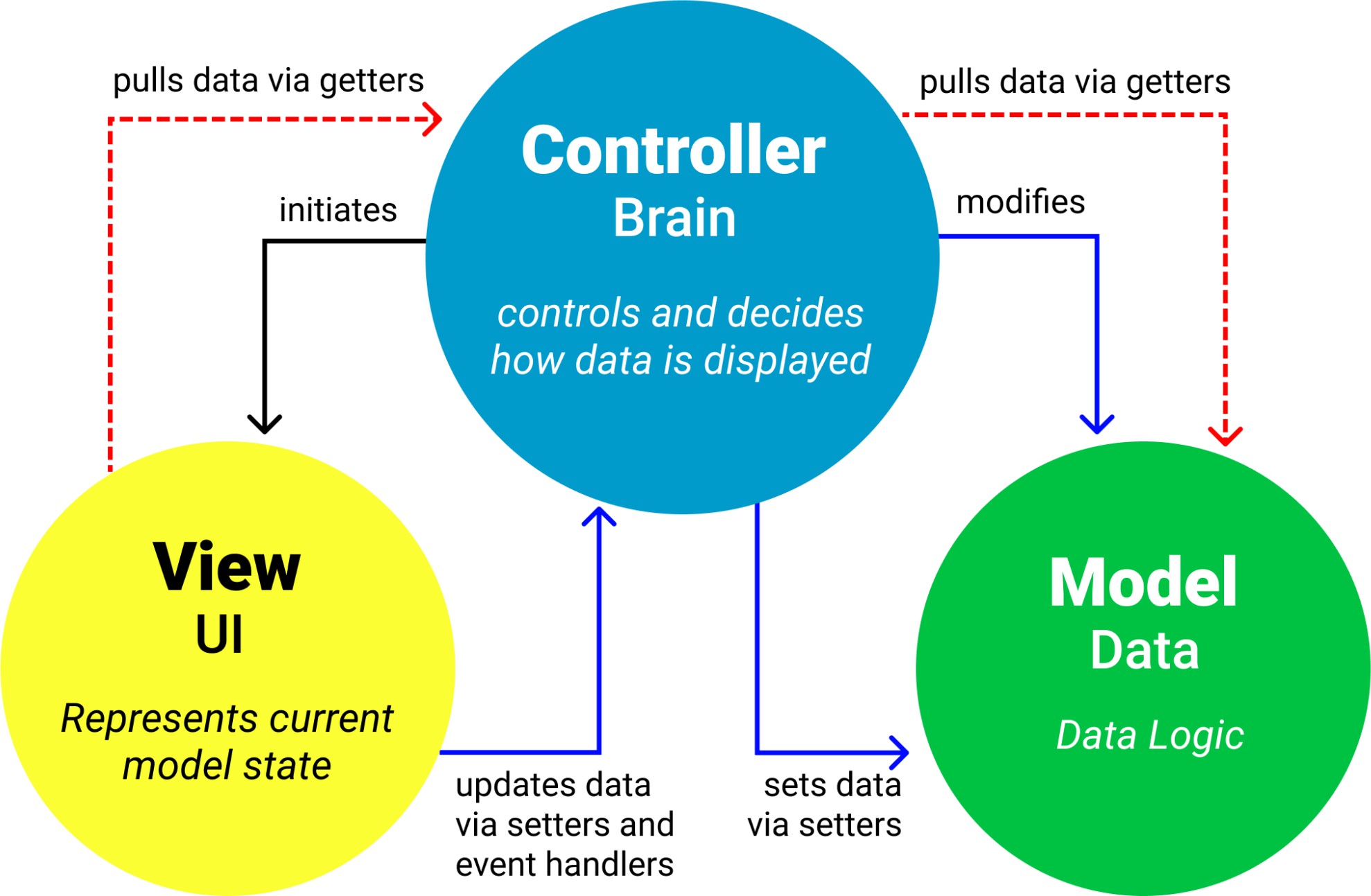
Bao gồm các thành phần hỗ trợ tích hợp với các dịch vụ bên ngoài như hệ thống chấm bài tự động (Judge API).

Giúp hệ thống mở rộng linh hoạt và dễ dàng tích hợp với các công nghệ khác trong tương lai.

Việc áp dụng kiến trúc phân lớp giúp hệ thống đảm bảo được tính rõ ràng về cấu trúc, dễ dàng bảo trì, mở rộng và tái sử dụng mã nguồn. Đồng thời giúp nhóm phát triển phân chia công việc hiệu quả hơn giữa các thành viên frontend và backend.

### Mô hình kiến trúc MVC

MVC [2] (Model-View-Controller) là một mô hình kiến trúc phần mềm được sử dụng để tổ chức và quản lý mã nguồn trong quá trình phát triển ứng dụng. Kiến trúc này tách biệt ứng dụng thành ba thành phần chính có thể tương tác được với nhau để tách biệt giữa cách thức mà thông tin được xử lý nội hàm và phần thông tin được trình bày và tiếp nhận từ phía người dùng.



Hình 1. 1: Kiến trúc phần mềm Model-View-Controller

- Model: là nơi lưu trữ thông tin, thực hiện các thao tác cập nhật và truy vấn dữ liệu mà không quan tâm đến cách dữ liệu được hiển thị hoặc tương tác với người dùng.

- View: là thành phần hiển thị giao diện người dùng. View chịu trách nhiệm hiển thị dữ liệu một cách đẹp mắt và dễ thao tác đối với người dùng.

- Controller: là thành phần tiếp nhận và xử lý sự kiện từ người dùng. Controller nhận yêu cầu từ người dùng thông qua View, sau đó thao tác với Model và điều hướng hiển thị trở lại View.

### Ngôn ngữ lập trình C# và ASP.NET

C# là một ngôn ngữ lập trình đa năng, được phát triển bởi Microsoft, thuộc dòng ngôn ngữ lập trình .NET. C# được sử dụng phổ biến trong phát triển phần mềm, đặc biệt là trong việc xây dựng các ứng dụng web với ASP.NET. ASP.NET là một framework phát triển ứng dụng web mạnh mẽ, được Microsoft phát triển, cho phép tạo ra các ứng dụng web, API, và các dịch vụ web hiệu quả.

Một số ưu điểm khiến ngôn ngữ lập trình C# và ASP.NET trở nên phổ biến:

- Hiệu suất cao và bảo mật tốt: C# và ASP.NET cung cấp hiệu suất vượt trội nhờ vào việc biên dịch mã nguồn sang mã máy (C#) và tối ưu hóa quá trình xử lý dữ liệu trong ASP.NET. Ngoài ra, ASP.NET còn tích hợp các tính năng bảo mật mạnh mẽ giúp bảo vệ các ứng dụng web khỏi các cuộc tấn công như SQL Injection, XSS, CSRF, v.v.

- Mã nguồn mở và miễn phí: C# và ASP.NET Core là mã nguồn mở, miễn phí và có thể sử dụng trên nhiều nền tảng như Windows, Linux và macOS. Điều này giúp tiết kiệm chi phí phát triển và triển khai ứng dụng.

- Tính linh hoạt: C# có thể được sử dụng để phát triển ứng dụng trên nhiều nền tảng, từ các ứng dụng máy tính để bàn, di động đến các ứng dụng web phức tạp. ASP.NET hỗ trợ phát triển ứng dụng web đa dạng từ các trang web động đến các API RESTful, WebSocket, và hơn thế nữa.

- Cộng đồng và thư viện hỗ trợ: C# và ASP.NET có một cộng đồng phát triển mạnh mẽ và số lượng thư viện hỗ trợ phong phú, giúp các nhà phát triển dễ dàng giải quyết các bài toán phức tạp và tối ưu hóa quá trình phát triển ứng dụng.

- Tính tương thích với các hệ quản trị cơ sở dữ liệu: ASP.NET có thể kết nối với nhiều hệ quản trị cơ sở dữ liệu như SQL Server, MySQL, PostgreSQL, SQLite và Oracle, giúp xây dựng các ứng dụng với cơ sở dữ liệu linh hoạt và mạnh mẽ.

Mặc dù có nhiều ưu điểm, C# và ASP.NET cũng có một số hạn chế, đặc biệt là về khối lượng tài nguyên hệ thống khi sử dụng cho các ứng dụng quy mô lớn. Tuy nhiên, những vấn đề này có thể được giải quyết nhờ vào các công cụ tối ưu hóa và cấu hình đúng đắn trong quá trình phát triển.

### Cơ sở dữ liệu MS SQL

MS SQL (Microsoft SQL Server) là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) được phát triển bởi Microsoft, sử dụng ngôn ngữ truy vấn có cấu trúc (SQL). MS SQL quản lý dữ liệu hiệu quả nhờ việc tổ chức dữ liệu dưới dạng các bảng và mối quan hệ giữa các bảng, giúp giảm thiểu sự lặp lại dữ liệu, tăng tính rõ ràng và hiệu quả của dữ liệu.

**Ưu điểm của MS SQL:**

Tốc độ cao, ổn định, dễ sử dụng: MS SQL là cơ sở dữ liệu tốc độ cao, ổn định, dễ sử dụng và hoạt động trên các hệ điều hành Windows, cung cấp nhiều tính năng tiện ích mạnh mẽ.

Độ bảo mật cao: MS SQL rất thích hợp cho các ứng dụng có truy cập CSDL trên Internet, với các tính năng bảo mật mạnh mẽ như mã hóa, kiểm tra quyền truy cập và chính sách bảo mật nâng cao.

Khả năng mở rộng: MS SQL hỗ trợ quy mô dữ liệu lớn và có khả năng mở rộng linh hoạt, cho phép xử lý các yêu cầu phức tạp và dữ liệu lớn trong môi trường doanh nghiệp.

**Nhược điểm của MS SQL:**

Chi phí cao: MS SQL yêu cầu giấy phép sử dụng, điều này có thể gây tốn kém cho các tổ chức nhỏ hoặc các dự án cần tiết kiệm chi phí.

Yêu cầu tài nguyên hệ thống lớn: MS SQL yêu cầu phần cứng và tài nguyên hệ thống mạnh mẽ, đặc biệt là khi xử lý các cơ sở dữ liệu lớn hoặc các tác vụ phức tạp.

### Ngôn ngữ lập trình ReactJs

ReactJS là một thư viện JavaScript được phát triển bởi Facebook, chuyên dùng để xây dựng giao diện người dùng (UI) cho các ứng dụng web. React cho phép phát triển ứng dụng đơn trang (SPA) với khả năng tái sử dụng các thành phần giao diện, tăng hiệu quả phát triển và dễ bảo trì. ReactJS nổi bật với tính năng Virtual DOM, giúp tối ưu hóa hiệu suất của ứng dụng bằng cách giảm thiểu số lần cập nhật giao diện thực sự trên trình duyệt.

**Ưu điểm của ReactJS:**

Dễ sử dụng và học hỏi: ReactJS có cú pháp đơn giản, dễ hiểu và rất dễ tiếp cận đối với lập trình viên JavaScript.

Tái sử dụng component: ReactJS cho phép tái sử dụng các component giao diện, giúp giảm thiểu sự trùng lặp mã nguồn và tăng tính mô-đun cho ứng dụng.

Tốc độ và hiệu suất cao: Nhờ vào Virtual DOM, ReactJS có khả năng cập nhật giao diện nhanh chóng và tối ưu hóa hiệu suất của ứng dụng web.

Cộng đồng mạnh mẽ: ReactJS có một cộng đồng phát triển lớn mạnh, với rất nhiều tài liệu, plugin và thư viện hỗ trợ.

**Nhược điểm của ReactJS:**

SEO khó khăn: Mặc dù có thể cải thiện SEO với các kỹ thuật như server-side rendering (SSR), nhưng ReactJS không tối ưu cho việc tối ưu hóa công cụ tìm kiếm (SEO) mặc định.

Tài liệu chưa đầy đủ: Một số phần của ReactJS có tài liệu chưa đầy đủ hoặc khó hiểu, đòi hỏi người học phải tự nghiên cứu hoặc tìm kiếm từ cộng đồng.

Cập nhật nhanh chóng: Với sự phát triển nhanh chóng của ReactJS, việc cập nhật và làm quen với các thay đổi mới trong thư viện có thể gây khó khăn cho lập trình viên.

## Kết luận chương 1

Chương 1 đã trình bày tổng quan về hệ thống hỗ trợ tự học và đánh giá trực tuyến, nhu cầu xây dựng hệ thống thi lập trình trực tuyến cho các trường đại học. Từ đó, đề tài nghiên cứu được xác định với mục tiêu phát triển một hệ thống cho phép sinh viên lập trình trực tiếp trên web, gửi bài và chấm điểm tự động bằng test case. Để đáp ứng các yêu cầu này, các công nghệ phù hợp được lựa chọn là ngôn ngữ lập trình ASP.NET Core, ReactJS, hệ quản trị CSDL MS SQL, và sử dụng Docker để triển khai hệ thống.

# PHÂN TÍCH THIÊT KẾ HỆ THỐNG

## Yêu cầu chức năng và phi chức năng

### Yêu cầu chức năng đối với sinh viên

* Đăng nhập: Sinh viên có thể đăng nhập vào hệ thống bằng mã sinh viên và mật khẩu đã được trường cấp trước đó.
* Xem danh sách học phần: Sinh viên có thể xem danh sách các học phần mà đã đăng ký học trước đó.
* Xem bài học và tài liệu: Sinh viên có thể xem các bài học, tài liệu hướng dẫn và các bài tập liên quan đến từng học phần.
* Làm bài tập lập trình: Sinh viên có thể viết mã nguồn và làm các bài tập lập trình trực tiếp trên hệ thống.
* Xem kết quả chấm điểm: Sinh viên có thể xem kết quả chấm điểm tự động sau khi nộp bài tập, bao gồm số test case đã pass, điểm số và thời gian thực thi.
* Quản lý thành tích học tập: xem các thành tích học tập, làm bài tập của mình qua các học phần.

### Yêu cầu đối với giảng viên

* Quản lý bài tập: Giảng viên có thể thêm, sửa, xóa bài tập lập trình, thiết lập các test case và ví dụ cho từng bài tập.
* Quản lý sinh viên: Quản trị viên có thể quản lý tài khoản sinh viên, xem thông tin và theo dõi tiến độ học tập của sinh viên.
* Chấm bài tự động: Quản trị viên có thể cấu hình hệ thống chấm bài tự động và theo dõi kết quả của sinh viên.
* Báo cáo và thống kê: Giảng viên có thể xem báo cáo thống kê kết quả làm bài của sinh viên, các vấn đề nổi bật trong quá trình học tập và đánh giá.

### Các yêu cầu phi chức năng

* Giao diện thân thiện với người dùng: Giao diện người dùng phải dễ sử dụng, trực quan, đẹp mắt và dễ dàng thao tác cho sinh viên và giảng viên.
* Tốc độ và hiệu suất cao: Hệ thống phải có tốc độ tải trang nhanh, có khả năng xử lý và tương tác mượt mà, đặc biệt trong việc chấm điểm tự động và xử lý bài tập lập trình.
* Độ khả dụng và tin cậy cao: Hệ thống phải đảm bảo luôn sẵn sàng và ổn định, tránh xảy ra lỗi trong quá trình làm bài và chấm điểm.
* Bảo mật và toàn vẹn dữ liệu: Dữ liệu của sinh viên và kết quả bài tập phải được bảo mật và đảm bảo tính toàn vẹn, không bị mất mát hoặc sai lệch.
* Khả năng mở rộng và bảo trì dễ dàng: Hệ thống phải có khả năng mở rộng để đáp ứng nhu cầu người dùng trong tương lai và dễ dàng bảo trì, cập nhật các tính năng mới.

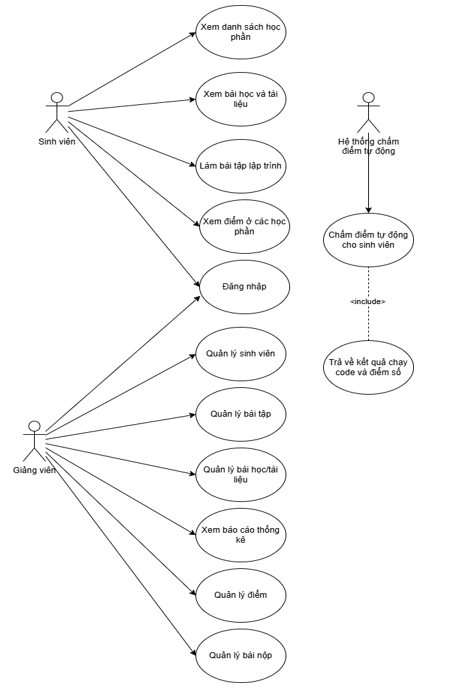
## Mô hình hóa chức năng

### Danh sách các Actor

Bảng 2. 1: Danh sách các Actor

|  |  |  |
| --- | --- | --- |
| **STT** | **Actor** | **Ghi chú** |
| 1 | Sinh viên | Là người sử dụng chính hệ thống để tham gia học các học phần và làm bài tập lập trình. |
| 2 | Giảng viên | Là người quản lý hệ thống, tạo các bài tập ở các học phần và các bài test lập trình. |
| 3 | Hệ thống chấm điểm tự động | Là hệ thống tích hợp vào để tự động chấm điểm các bài tập lập trình của sinh viên. |

### Biểu đồ Use Case



Hình 2. 1: Biểu đồ Use case tổng quát của hệ thống

## Mô tả chi tiết các Use Case

### Use case đăng nhập hệ thống

Bảng 2. 2: Use case Đăng nhập hệ thống

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên use case | | Đăng nhập hệ thống | | |
| Tác nhân | | Sinh viên, giảng viên | | |
| Mô tả | | Use case này cho phép sinh viên, giảng viên sử dụng tài khoản để đăng nhập vào hệ thống | | |
| Sự kiện kích hoạt | | Sinh viên truy cập trang đăng nhập và gửi thông tin tài khoản | | |
| Tiền điều kiện | | Sinh viên đã có tài khoản | | |
| Luồng sự kiện | Luồng sự kiện chính | # | **Hành động của tác nhân** | **Hệ thống phản hồi** |
| 1 | Sinh viên, giảng viên truy cập trang đăng nhập | Hiển thị form đăng nhập |
| 2 | Sinh viên, giảng viêng nhập mã và mật khẩu, ấn “Đăng nhập” | Kiểm tra thông tin hợp lệ và cho phép truy cập hệ thống |
| Luồng sự kiện rẽ nhánh | 2.1 | Nhập sai mật khẩu | Hiển thị thông báo lỗi "Mật khẩu không đúng" |
| 1  .. 2 |  | Tại bất kỳ bước nào trong luồng sự kiện chính, nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và UC kết thúc. |
| Hậu điều kiện | | Sinh viên được đăng nhập vào hệ thống và chuyển đến giao diện trang chính | | |

### Use case xem danh sách học phần

Bảng 2. 3: Use case Xem danh sách học phần

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên use case | | Xem danh sách học phần | | |
| Tác nhân | | Sinh viên | | |
| Mô tả | | Sau khi đăng nhập, sinh viên có thể xem danh sách các học phần mà mình tham gia | | |
| Sự kiện kích hoạt | | Sinh viên chọn mục "Học phần" trên giao diện | | |
| Tiền điều kiện | | Sinh viên đã đăng nhập vào hệ thống | | |
| Luồng sự kiện | Luồng sự kiện chính | # | **Tác nhân hành động** | **Hệ thống phản hồi** |
| 1 | Truy cập mục "Học phần" |  |
| 2 |  | Truy vấn và hiển thị danh sách học phần tương ứng với sinh viên |
| Luồng sự kiện rẽ nhánh | 21 |  | Nếu chưa có học phần nào Hiển thị thông báo "Bạn chưa được phân vào học phần nào" |
| 1  2 |  | Tại bất kỳ bước nào trong luồng sự kiện chính, nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và UC kết thúc. |
| Hậu điều kiện | | Sinh viên có thể chọn học phần để vào xem nội dung | | |

### Use case truy cập bài học

Bảng 2. 4: Use case truy cập bài học

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên use case | | Truy cập bài học | | |
| Tác nhân | | Sinh viên | | |
| Mô tả | | Sinh viên chọn học phần và truy cập các bài học lý thuyết | | |
| Sự kiện kích hoạt | | Sinh viên chọn học phần cụ thể | | |
| Tiền điều kiện | | Học phần có ít nhất một bài học | | |
| Luồng sự kiện | Luồng sự kiện chính | # | **Hành động của tác nhân** | **Hệ thống phản hồi** |
| 1 | Nhấp vào bài học cụ thể |  |
| 2 |  | Hiển thị nội dung bài học với các tài liệu |
| 1  .. 2 |  | Tại bất kỳ bước nào trong luồng sự kiện chính, nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và UC kết thúc. |
| Hậu điều kiện | | Không | | |

### Use case làm bài tập lập trình

Bảng 2. 5:Use case làm bài tập lập trình

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên use case | | Làm bài tập lập trình | | |
| Tác nhân | | Sinh viên | | |
| Mô tả | | Sinh viên thực hiện các bài tập lập trình trực tiếp trên hệ thống sau khi học xong lý thuyết | | |
| Sự kiện kích hoạt | | Sinh viên chọn một bài tập từ bài học | | |
| Tiền điều kiện | | Sinh viên đã đăng nhập và chọn một bài học có bài tập lập trình | | |
| Luồng sự kiện | Luồng sự kiện chính | # | **Hành động của tác nhân** | **Hệ thống phản hồi** |
| 1 | Chọn bài học và nhấn vào bài tập | Hiển thị giao diện bài tập lập trình gồm: đề bài, ví dụ, khung viết code |
|  | 2 | Đọc đề, viết code vào ô soạn thảo | Cho phép chọn ngôn ngữ lập trình, hiển thị nút “Gửi bài” |
| Ll  Luồng sự kiện rẽ nhánh | 2.1 |  | Sinh viên không code gì mà bấm “gửi bài” hiển thị cảnh báo “Hãy viết code mới được gửi” |
| 2.2 |  | Nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và UC kết thúc. |
| Hậu điều kiện  Hậu đi | | Sinh viên hoàn thành việc viết code, sẵn sàng gửi bài để hệ thống chấm điểm | | |

### Use case gửi code và chấm bài

Bảng 2. 6: Use case gửi code và chấm bài

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên use case | | Gửi code và chấm bài | | |
| Tác nhân | | Sinh viên | | |
| Mô tả | | Sau khi làm bài, sinh viên gửi code lên hệ thống để được chấm điểm tự động | | |
| Sự kiện kích hoạt | | Sinh viên nhấn nút “Gửi bài” sau khi hoàn thành phần code | | |
| Tiền điều kiện | | Bài làm đã được viết và có thể biên dịch | | |
| Luồng sự kiện | Luồng sự kiện chính |  | **Hành động của tác nhân** | **Hệ thống phản hồi** |
| 1 | Nhấn nút “Gửi bài” |  |
| 2 |  | Gửi code đến Piston API để thực thi |
| 3 |  | Nhận kết quả chấm (pass/fail theo test case) |
| 4 |  | Hiển thị mã đã nộp và kết quả chấm (test case, điểm, thời gian...). |
|  | Luồng sự kiện rẽ nhánh | 1 | Chưa làm bài nào | Hiển thị thông báo “Chưa có bài nộp nào” |
|  |  |  |  | Nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và UC kết thúc. |
| Hậu điều kiện | | Không | | |

### Use case theo dõi tiến độ học tập

Bảng 2. 7: Use case Theo dõi tiến độ học tập

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên use case | | Theo dõi tiến độ học tập | | |
| Tác nhân | | Sinh viên | | |
| Mô tả | | Hệ thống cung cấp thông tin tổng quan về các bài học đã hoàn thành, số bài tập đã làm, điểm trung bình,... | | |
| Sự kiện kích hoạt | | Sinh viên chọn mục “Tiến độ học tập” trên giao diện | | |
| Tiền điều kiện | | Sinh viên đã thực hiện ít nhất một hoạt động học hoặc làm bài tập | | |
| Luồng sự kiện | Luồng sự kiện chính |  | **Hành động của tác nhân** | **Hệ thống phản hồi** |
| 1 | Truy cập mục “Tiến độ học tập” |  |
| 2 |  | Lấy dữ liệu về các bài học đã mở, bài tập đã làm, điểm trung bình |
| 3 |  | Hiển thị bảng tiến độ học tập dưới dạng biểu đồ hoặc danh sách |
| Luồng sự kiện rẽ nhánh | 1 |  | Nếu sinh viên chưa làm bài tập nào Hiển thị thông báo “Bạn chưa bắt đầu học phần nào”` |
|  |  | 2 |  | Nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và UC kết thúc. |
| Hậu điều kiện | | Sinh viên có thể biết được mình đang học tới đâu và cần cải thiện phần nào | | |

### Use case tạo bài học và tài liệu dạy

Bảng 2. 8: Use case tạo bài học và tài liệu dạy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên use case | | Tạo bài học và tài liệu | | |
| Tác nhân | | Giảng viên | | |
| Mô tả | | Giảng viên tạo các bài học mới, biên soạn nội dung tài liệu, thêm vào hệ thống để sinh viên học tập | | |
| Sự kiện kích hoạt | | Giảng viên đăng nhập và truy cập vào mục quản lý bài học | | |
| Tiền điều kiện | | Giảng viên đã đăng nhập vào hệ thống | | |
| Luồng sự kiện | Luồng sự kiện chính | # | **Hành động của tác nhân** | **Hệ thống phản hồi** |
| 11 | Chọn thêm bài học | Hiển thị form nhập thông tin bài học (tiêu đề, mô tả, tài liệu PDF, video, v.v.) |
| 2 | Nhập đầy đủ thông tin và tải lên tài liệu |  |
| 22 |  | Lưu bài học vào hệ thống, hiển thị trong danh sách bài học cho sinh viên |
| Luồng sự kiện rẽ nhánh | 2.1 |  | Nếu quên nhập thông tin gì đó hệ thống hiển thị “Vui lòng nhập .. cho bài học” |
| 3.2 |  | Nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và UC kết thúc. |
| Hậu điều kiện | | Bài học mới được tạo thành công và có thể hiển thị cho sinh viên | | |

### Use case tạo bài tập lập trình

Bảng 2. 9: Use case tạo bài tập lập trình

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên use case | | Tạo bài tập lập trình | | |
| Tác nhân | | Giảng viên | | |
| Mô tả | | Giảng viên tạo các bài tập lập trình, thêm đề bài, ví dụ, test case chấm điểm | | |
| Sự kiện kích hoạt | | Giảng viên chọn bài học và nhấn “Thêm bài tập lập trình” | | |
| Tiền điều kiện | | Bài học đã tồn tại trong hệ thống | | |
| Luồng sự kiện | Luồng sự kiện chính |  | **Hành động của tác nhân** | **Hệ thống phản hồi** |
| 1 | Chọn bài học cần thêm bài tập | Hiển thị giao diện thêm bài tập (tiêu đề, mô tả, ví dụ input/output) |
| 2 | Nhập nội dung đề bài, ví dụ minh họa và test case | Lưu bài tập vào hệ thống, liên kết với bài học tương ứng |
| Luồng sự kiện rẽ nhánh | 2.1 |  | Nếu Không nhập test case, Cảnh báo “Vui lòng nhập ít nhất một test case để chấm điểm” |
| 2.2 |  | Nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và UC kết thúc. |
| Hậu điều kiện | | Bài tập lập trình đã sẵn sàng cho sinh viên làm và hệ thống có thể tự động chấm điểm | | |

### Use case xem kết quả bài làm của sinh viên

Bảng 2. 10: Use case Xem kết quả của Sinh viên

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên use case | | Xem kết quả làm bài của sinh viên | | |
| Tác nhân | | Giảng viên | | |
| Mô tả | | Giảng viên theo dõi, đánh giá kết quả bài tập sinh viên đã nộp | | |
| Sự kiện kích hoạt | | Giảng viên truy cập trang quản lý kết quả bài tập | | |
| Tiền điều kiện | | Có sinh viên đã nộp bài | | |
| Luồng sự kiện | Luồng sự kiện chính | # | **Hành động của tác nhân** | **Hệ thống phản hồi** |
| 11 | Truy cập phần “Kết quả bài làm” |  |
| 22 |  | Hiển thị danh sách sinh viên và các bài tập đã nộp |
|  | 33 | Chọn sinh viên hoặc bài cụ thể để xem chi tiết |  |
|  | 44 |  | Hiển thị mã nguồn, kết quả test case, thời gian nộp |
| Luồng sự kiện rẽ nhánh | 11 | Chọn một sinh viên chưa nộp bài nào |  |
| 22 |  | Hiển thị thông báo “Chưa có bài nộp” |
|  |  | 33 |  | Nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và UC kết thúc. |
| Hậu điều kiện | | Giảng viên có thể đánh giá tiến độ và năng lực của sinh viên | | |

### Use case quản lý bài học và bài tập

Bảng 2. 11: Use case Quản lý bài học và bài tập

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên use case | | Quản lý bài học và bài tập | | |
| Tác nhân | | Giảng viên | | |
| Mô tả | | Giảng viên có thể sửa, xóa bài học hoặc bài tập không còn phù hợp | | |
| Sự kiện kích hoạt | | Giảng viên truy cập phần danh sách bài học hoặc bài tập | | |
| Tiền điều kiện | | Bài học hoặc bài tập đã tồn tại | | |
| Luồng sự kiện | Luồng sự kiện chính |  | **Hành động của tác nhân** | **Hệ thống phản hồi** |
| 1 | Chọn bài học hoặc bài tập cần chỉnh sửa |  |
| 2 |  | Hiển thị nội dung chi tiết |
| 3 | Thực hiện cập nhật nội dung hoặc xóa bài |  |
| 4 |  | Lưu thay đổi hoặc xóa dữ liệu tương ứng |
|  | Luồng sự kiện rẽ nhánh | 1 |  | Nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và UC kết thúc. |
| Hậu điều kiện | | Dữ liệu bài học/bài tập được cập nhật chính xác trong hệ thống | | |

### Use case quản lý sinh viên

Bảng 2. 12: Use case quản lý sinh viên

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên use case | | Quản lý sinh viên | | |
| Tác nhân | | Giảng viên | | |
| Mô tả | | Xem danh sách sinh viên, thông tin tài khoản, theo dõi tiến độ học tập và kết quả làm bài của từng sinh viên | | |
| Sự kiện kích hoạt | | Giảng viên truy cập mục "Quản lý Sinh viên" từ trang quản trị | | |
| Tiền điều kiện | | Sinh viên đã đăng ký tài khoản và tham gia vào hệ thống | | |
| Luồng sự kiện | Luồng sự kiện chính | # | **Hành động của tác nhân** | **Hệ thống phản hồi** |
| 11 | Truy cập trang quản lý sinh viên |  |
| 22 |  | Hiển thị danh sách sinh viên đang hoạt động |
| 33 | Chọn sinh viên cụ thể để xem chi tiết |  |
| 44 |  | Hiển thị thông tin cá nhân, bài học đã học, số bài đã làm và điểm số |
|  | Luồng sự kiện rẽ nhánh | 11 |  | Nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và UC kết thúc. |
| Hậu điều kiện | | Thông tin sinh viên được hiển thị, giảng viên có thể đánh giá quá trình học tập | | |

### Use case thống kê và báo cáo

Bảng 2. 13: : Use case thống kê và báo cáo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên use case | | Thống kê và báo cáo | | |
| Tác nhân | | Giảng viên | | |
| Mô tả | | Xem thống kê về số lượng bài nộp, tỷ lệ hoàn thành, điểm trung bình theo từng học phần hoặc bài tập | | |
| Sự kiện kích hoạt | | Truy cập chức năng thống kê từ trang quản trị | | |
| Tiền điều kiện | | Có dữ liệu bài nộp từ sinh viên | | |
| Luồng sự kiện | Luồng sự kiện chính | # | **Hành động của tác nhân** | **Hệ thống phản hồi** |
| 11 | Chọn học phần hoặc bài tập cụ thể |  |
| 22 |  | Truy vấn và hiển thị các số liệu thống kê gồm: tổng số sinh viên, số bài nộp, số bài hoàn thành, điểm trung bình |
|  |  | 33 | Chọn xuất báo cáo |  |
|  | 4 |  | Tạo file PDF hoặc Excel và cho phép tải xuống |
|  | Luồng sự kiện rẽ nhánh | 1 | Chọn học phần chưa có sinh viên tham gia |  |
|  |  | 2 |  | Thông báo “Không có dữ liệu để thống kê” |
|  |  | 3 |  | Nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và UC kết thúc. |
| Hậu điều kiện | | Không | | |

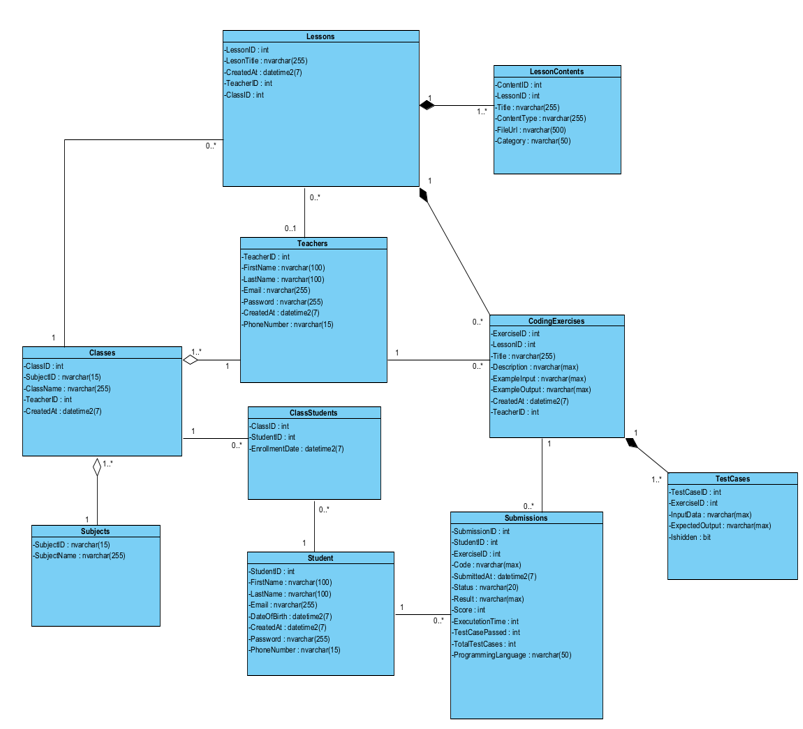
### Use case quản lý học phần

Bảng 2. 14: Use case quản lý học phần

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên use case | | Quản lý học phần | | |
| Tác nhân | | Giảng viên | | |
| Mô tả | | Tạo, cập nhật, xóa các học phần (môn học) và liên kết với bài học, bài tập tương ứng | | |
| Sự kiện kích hoạt | | Giảng viên truy cập chức năng quản lý học phần | | |
| Tiền điều kiện | | Tài khoản có quyền quản trị | | |
| Luồng sự kiện | Luồng sự kiện chính | # | **Hành động của tác nhân** | **Hệ thống phản hồi** |
| 1 | Chọn menu “Quản lý học phần”. |  |
|  | 2 |  | Hiển thị danh sách các học phần lên màn hình |
|  | 3 | Nếu chọn “Thêm học phần” thực hiện các bước 4, 5 |  |
|  |  | 4 | Nhập thông tin học phần và chọn Thêm mới |  |
|  |  | 5 |  | Lưu thông tin Học phần và hiển thị danh sách học phần mới lên màn hình, UC kết thúc. |
|  |  | 6 | Nếu chọn 1 Học phần đã có để cập nhật thông tin, thực hiện bước 7,8. |  |
|  |  | 7 | Thực hiện sửa thông tin Học phần chọn cập nhật. Hoặc chọn xóa học phần |  |
|  |  | 8 |  | Cập nhật thông tin học phần. Hiển thị danh sách Học phần mới lên màn hình, UC kết thúc. |
|  | Luồng sự kiện rẽ nhánh | 1 |  | Nếu không có học phần nào, hiển thị thông báo “không có học phần nào”. |
|  | 2 |  | Nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và UC kết thúc. |
| Hậu điều kiện | | Học phần được tạo và có thể gán bài học, bài tập | | |

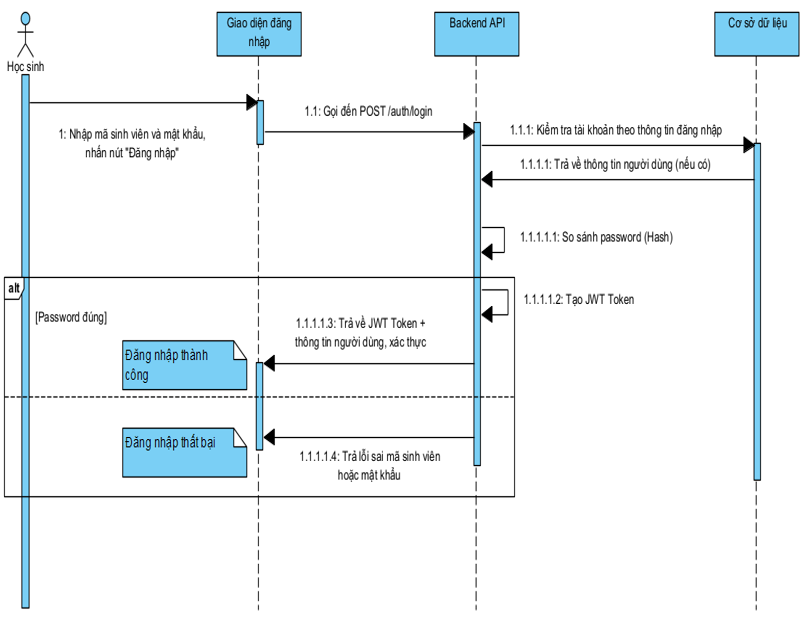
## Thiết kế chi tiết

### Mô hình hóa điều kiện



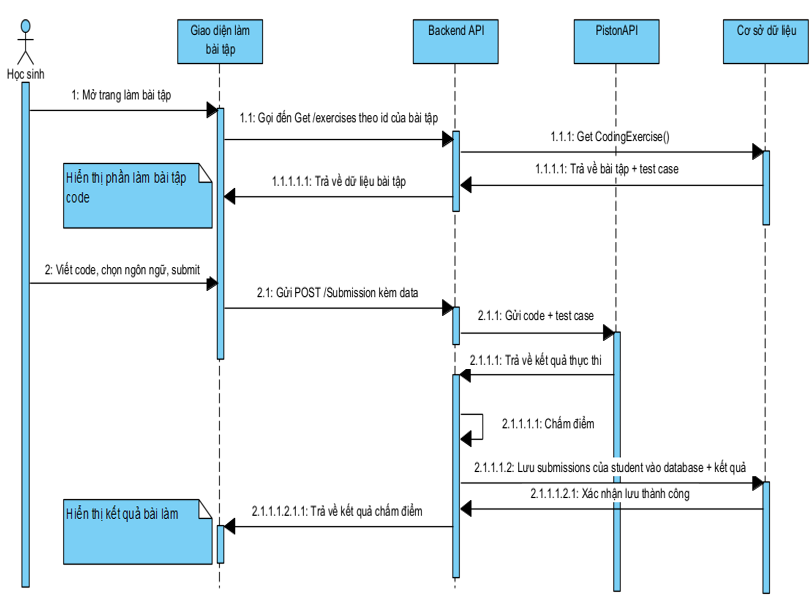
Hình 2. 2: Biểu đồ lớp thiết kế hệ thống

### Biểu đồ trình tự use case đăng nhập



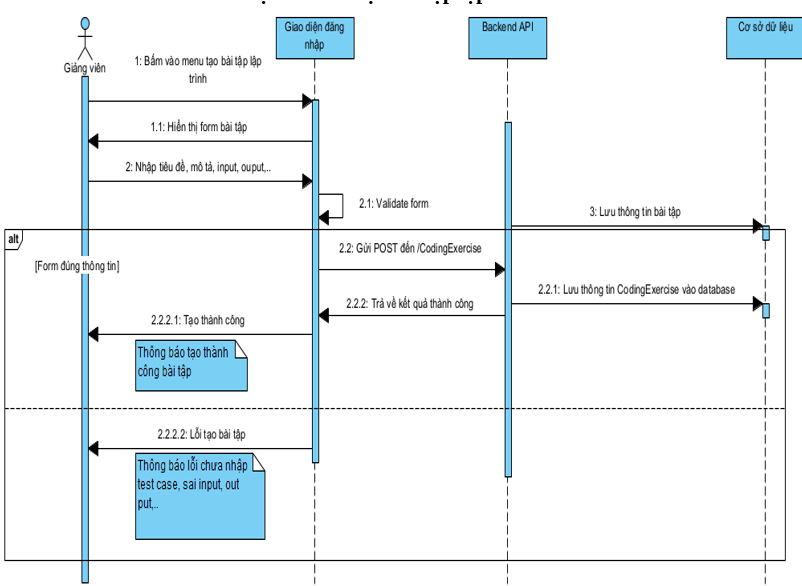
Hình 2. 3: Biểu đồ trình tự UC Đăng nhập

### Biểu đồ trình tự use case làm bài tập lập trình



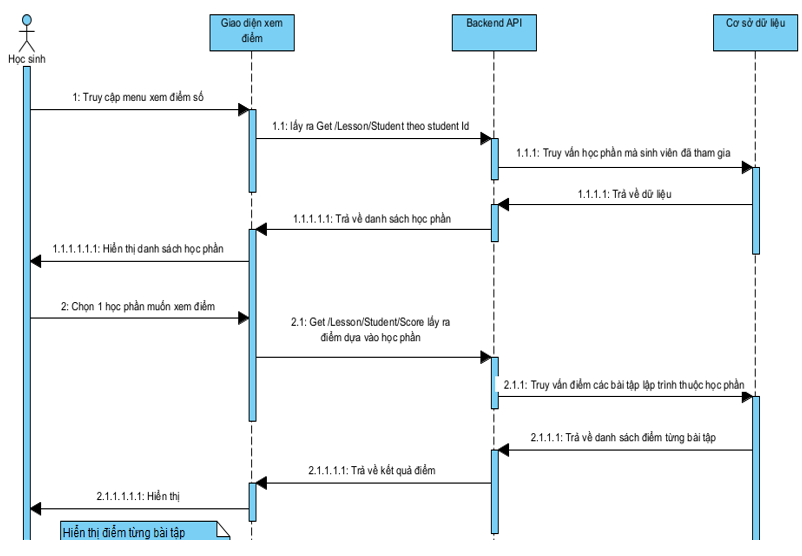
Hình 2. 4: Biểu đồ trình tự UC Làm bài tập lập trình

### Biểu đồ trình tự use case tạo bài tập lập trình



Hình 2. 5: Biểu đồ trình tự UC Tạo bài tập lập trình

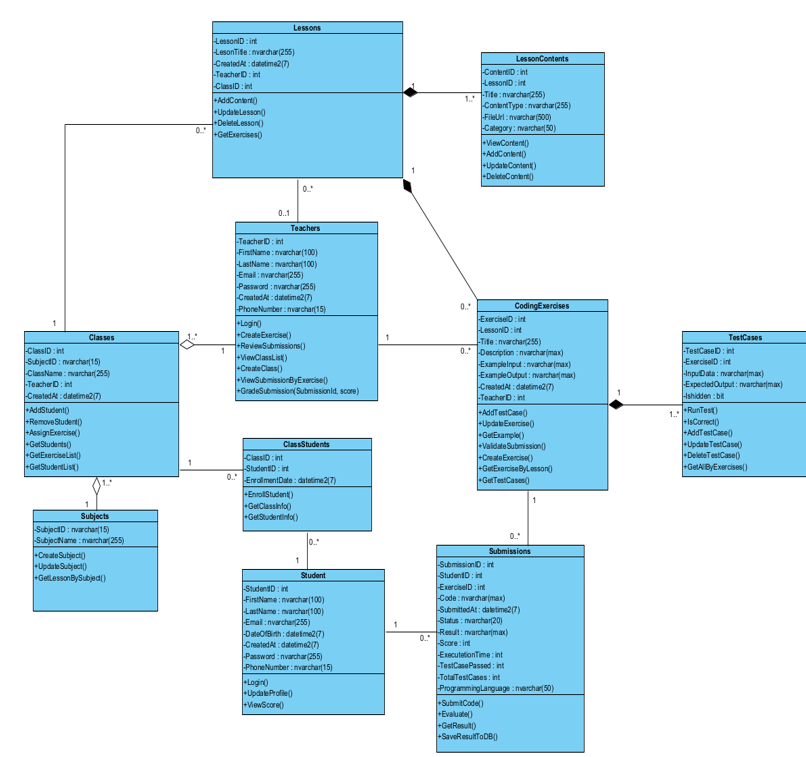
### Biểu đồ trình tự use case theo dõi kết quả học



Hình 2. 6: Biểu đồ trình tự UC Theo dõi kết quả học

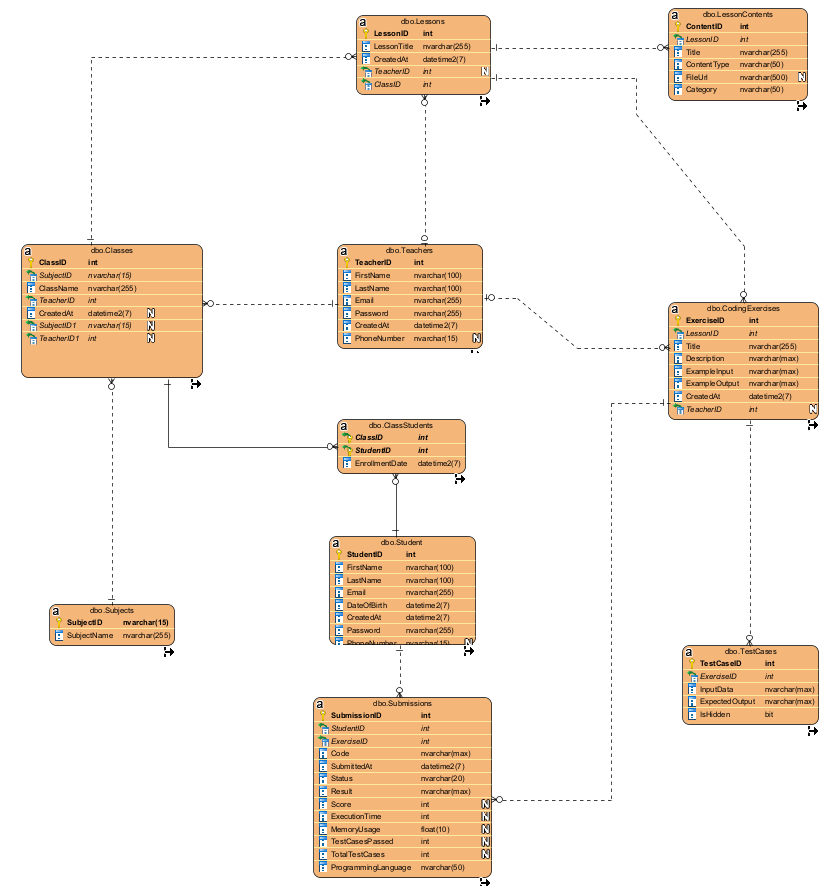
### Biểu đồ lớp chi tiết

Hình 2.7 là biểu đồ lớp sau khi thêm các phương thức từ biểu đồ tuần tự.



Hình 2. 7: Biểu đồ lớp chi tiết.

## Thiết kế cơ sở dữ liệu hợp lý

****

Hình 2. 8: Biểu đồ thực thể liên kết mức vật lý

### Bảng Subjects(học phần)

Bảng 2. 15: Bảng Subjects

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Giải thích** | **Ghi chú** |
| 1 | SubjectId | nvarchar(15) | Mã học phần | Khóa chính |
| 2 | SubjectName | nvarchar(255) | Tên học phần |  |

### Bảng Student (sinh viên)

Bảng 2. 16: Bảng Student

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Giải thích** | **Ghi chú** |
| 1 | StudentID | int | Mã sinh viên | Khóa chính |
| 2 | FirstName | Nvarchar(100) | Họ |  |
| 3 | LastName | Nvarchar(100) | Tên |  |
| 4 | Email | Nvarchar(255) | Email |  |
| 5 | DateOfBirth | Datetime2(7) | Ngày sinh |  |
| 6 | CreatedAt | Datetime2(7) | Thời gian tạo |  |
| 7 | Password | Nvarchar(255) | Mật khẩu |  |
| 8 | PhoneNumber | Nvarchar(15) | Số điện thoại |  |

### Bảng ClassStudents (Lớp học chứa student)

Bảng 2. 17: Bảng ClassStudent

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Giải thích** | **Ghi chú** |
| 1 | ClassID | int | Mã lớp học | Khóa chính |
| 2 | StudentID | int | Mã sinh viên | Khóa chính |
| 3 | EnrollmentDate | Datetime2(7) | Ngày tham gia học |  |

### Bảng Classes (Bảng lớp học)

Bảng 2. 18: Bảng Classes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Giải thích** | **Ghi chú** |
| 1 | ClassID | int | Mã lớp học | Khóa chính |
| 2 | SubjectID | Nvarchar(15) | Mã học phần | Khóa ngoại |
| 3 | ClassName | Nvarchar(255) | Tên lớp học |  |
| 4 | TeacherID | Int | Mã giảng viên | Khóa ngoại |
| 5 | CreatedAt | Datetime2(7) | Ngày tạo lớp |  |

### Bảng Teachers (Giảng viên)

Bảng 2. 19: Bảng Teachers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Giải thích** | **Ghi chú** |
| 1 | TeacherID | Int | Mã giảng viên | Khóa chính |
| 2 | FirstName | Nvarchar(100) | Họ |  |
| 3 | LastName | Nvarchar(100) | Tên |  |
| 4 | Email | Nvarchar(255) | Email |  |
| 5 | Password | Nvarchar(255) | Password |  |
| 6 | CreatedAt | Datetime2(7) | Ngày tạo |  |
| 7 | PhoneNumber | Nvarchar(15) | Số điện thoại |  |

### Bảng Lessons (Bài học)

Bảng 2. 20: Bảng Lessons

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Giải thích** | **Ghi chú** |
| 1 | LessonID | Int | Mã bài học | Khóa chính |
| 2 | LessonTitle | Nvarchar(255) | Tên tiêu đề |  |
| 3 | CreateAt | Datetime2(7) | Ngày tạo bài học |  |
| 4 | TeacherID | int | Mã giảng viên | Khóa ngoại |
| 5 | ClassID | int | Mã lớp học | Khóa ngoại |

### Bảng LessonContents (Nội dung bài học)

Bảng 2. 21: Bảng LessonContents

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Giải thích** | **Ghi chú** |
| 1 | ContentID | int | Mã nội dung bài học | Khóa chính |
| 2 | LessonID | Int(11) | Mã bài học | Khóa ngoại |
| 3 | Title | Nvarchar(255) | Tên tiêu đề |  |
| 4 | ContentType | Nvarchar(255) | Thể loại (File,PDF) |  |
| 5 | FileUrl | Nvarchar(500) | Đường dẫn |  |
| 6 | Category | Nvarchar(50) | Thể loại |  |

### Bảng TestCases (Kiểm thử)

Bảng 2. 22: Bảng TestCases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Giải thích** | **Ghi chú** |
| 1 | TestCaseID | Int | Mã test case | Khóa chính |
| 2 | ExerciseID | Int | Mã bài tập | Khóa ngoại |
| 3 | InputData | Nvarchar(max) | Đầu vào |  |
| 4 | ExpectedOutput | Nvarchar(max) | Mong đợi đầu ra |  |
| 5 | IsHidden | bit | Ẩn/hiện test case |  |

### Bảng CodingExercises (Bài tập code)

*Bảng 2.28: Bảng CodingExercises*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Giải thích** | **Ghi chú** |
| 1 | ExerciseID | Int | Mã bài tập | Khóa chính |
| 2 | LessonID | Int | Mã học phần | Khóa ngoại |
| 3 | Title | Nvarchar(255) | Tiêu đề |  |
| 4 | Description | Nvarchar(max) | Mô tả bài tập |  |
| 5 | ExampleInput | Nvarchar(max) | Ví dụ đầu vào |  |
| 6 | ExampleOutput | Nvarchar(max) | Ví dụ đầu ra |  |
| 7 | CreatedAt | Datetime2(7) | Ngày tạo |  |
| 8 | TeacherID | int | Mã giảng viên | Khóa ngoại |

### Bảng Submission (Bài nộp)

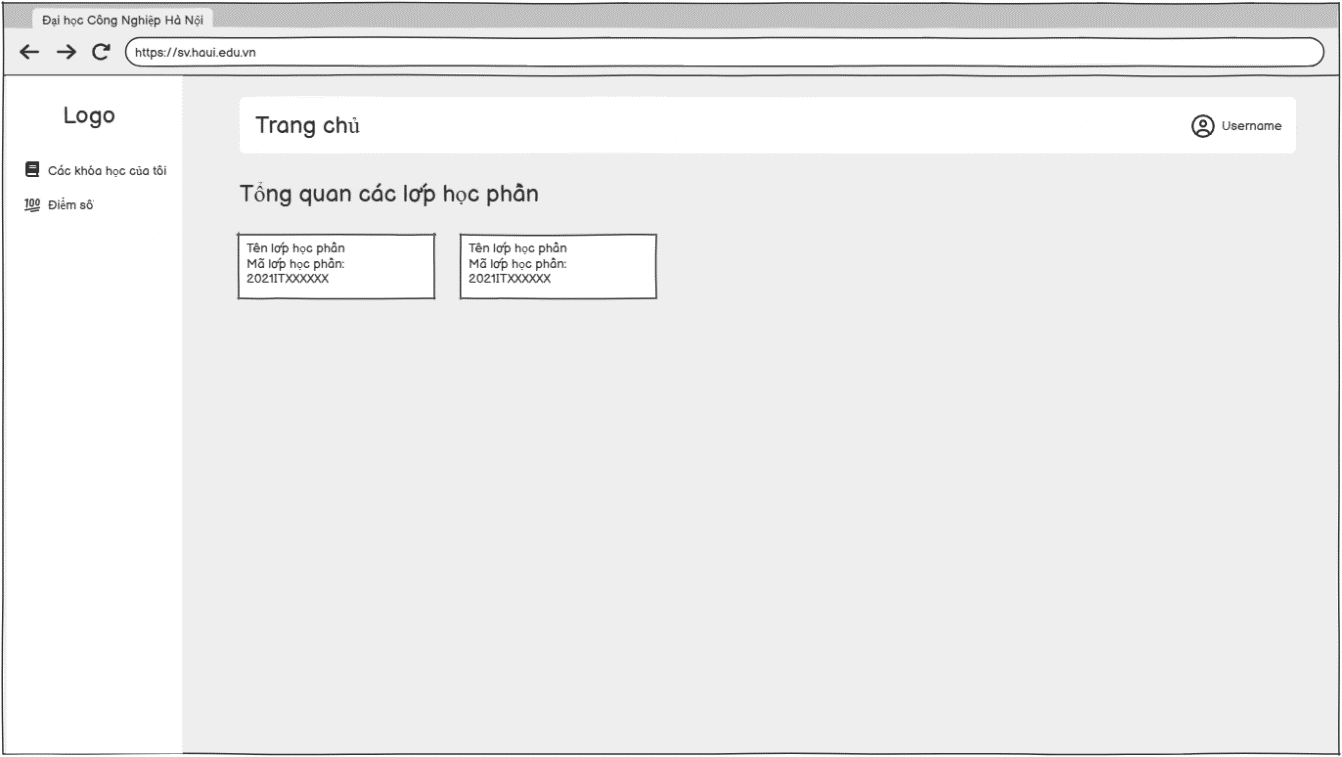
Bảng 2. 23: Bảng Submission

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Giải thích** | **Ghi chú** |
| 1 | SubmissionID | Int | Mã bài nộp | Khóa chính |
| 2 | StudentID | Int | Mã sinh viên | Khóa ngoại |
| 3 | ExerciseID | Int | Mã bài tập | Khóa ngoại |
| 4 | Code | Nvarchar(max) | Đoạn code |  |
| 5 | SubmittedAt | Datetime2(7) | Ngày nộp |  |
| 6 | Status | varchar(20) | Trạng thái |  |
| 7 | Result | Nvarchar(max) | Kết quả trả về |  |
| 8 | Score | Int | Điểm của sinh viên |  |
| 9 | ExcutetionTime | Int | Thời gian thực thi |  |
| 10 | TestCasePassed | Int | Số test case vượt |  |
| 11 | TotalTestCases | Int | Tổng test case |  |
| 12 | ProgramingLanguage | varchar(50) | Ngôn ngữ lập trình |  |

## Thiết kế hình dung màn hình

### Thiết kế giao diện trang chủ

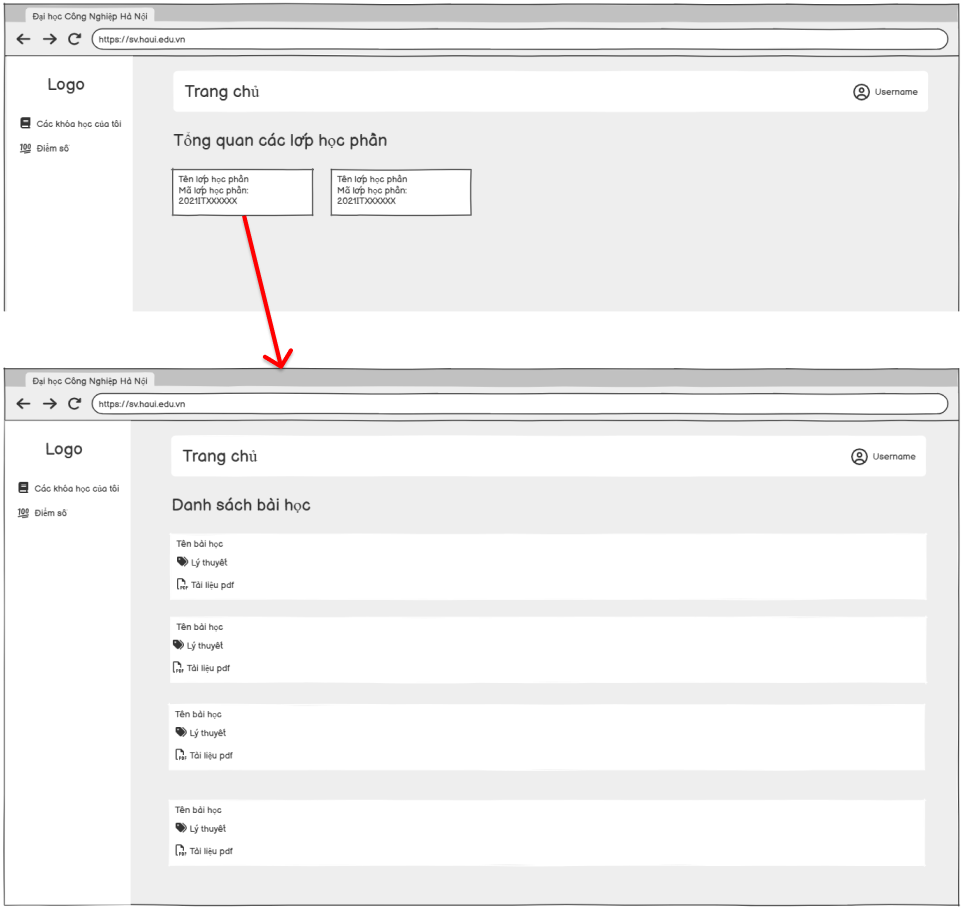
Giao diện trang chủ gồm header chứa logo, menu chính, thông tin các lớp học phần của sinh viên.



Hình 2. 9: Thiết kế giao diện trang chủ

### Thiết kế hình dung màn hình chọn lớp học phần

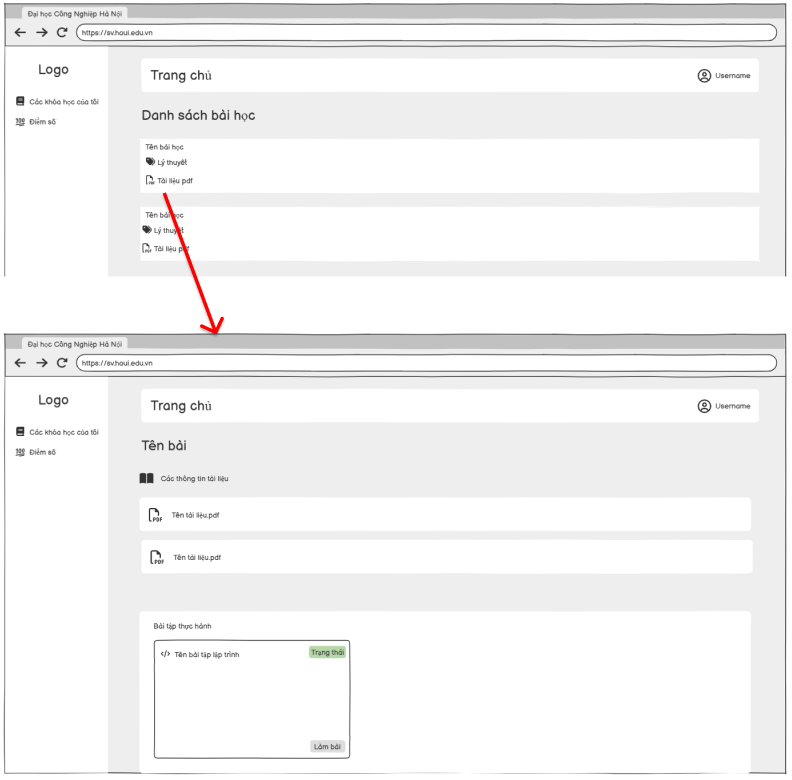
Khi chọn vào lớp học phần sẽ hiển thị danh sách bài học



Hình 2. 10: Thiết kế hình dung màn hình chọn lớp học phần

### Thiết kế hình dung màn hình chọn bài học

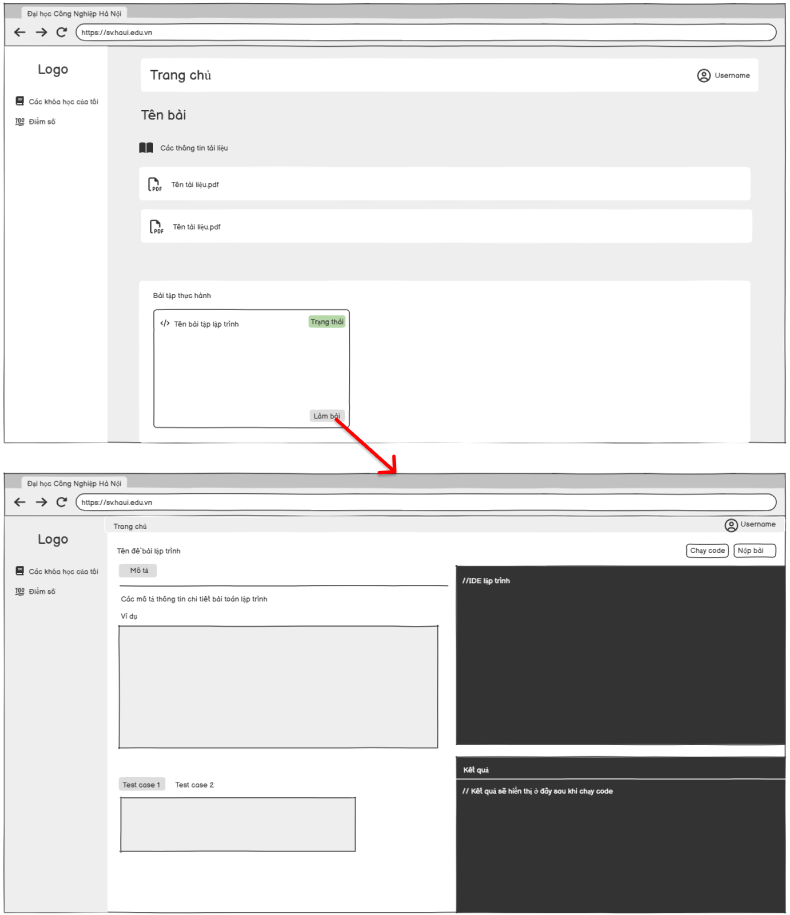
Sau khi chọn vào bài học trong lớp học phần sẽ hiển thị ra các nội dung bài học, các tài liệu và thông tin bài tập thực hành



Hình 2. 11: Thiết kế hình dung màn hình chọn bài học

### Thiết kế hình dung màn hình chọn bài tập thực hành

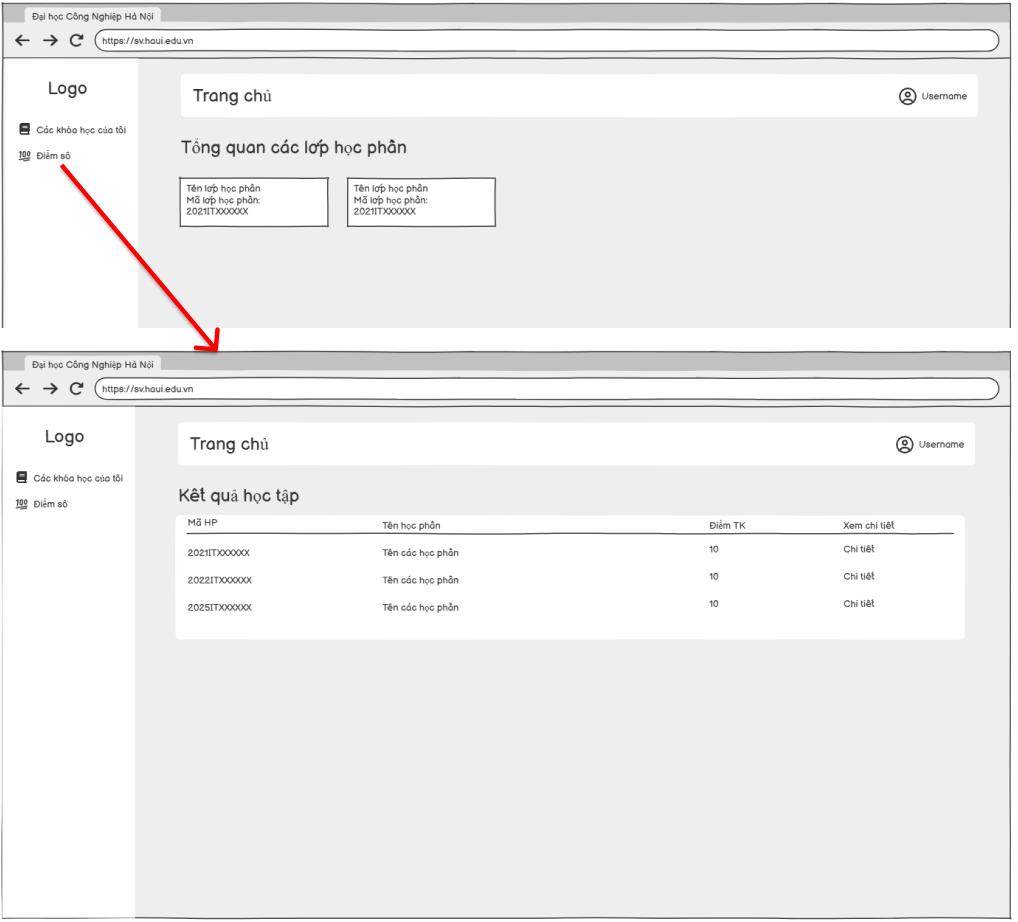
Khi chọn vào bài tập thực hành để luyện tập sẽ hiển thị ra giao diện coding system kèm theo các thông tin của bài học thực hành đó.



Hình 2. 12: Thiết kế hình dung màn hình làm bài tập thực hành

### Thiết kế hình dung màn h ình xem điểm số

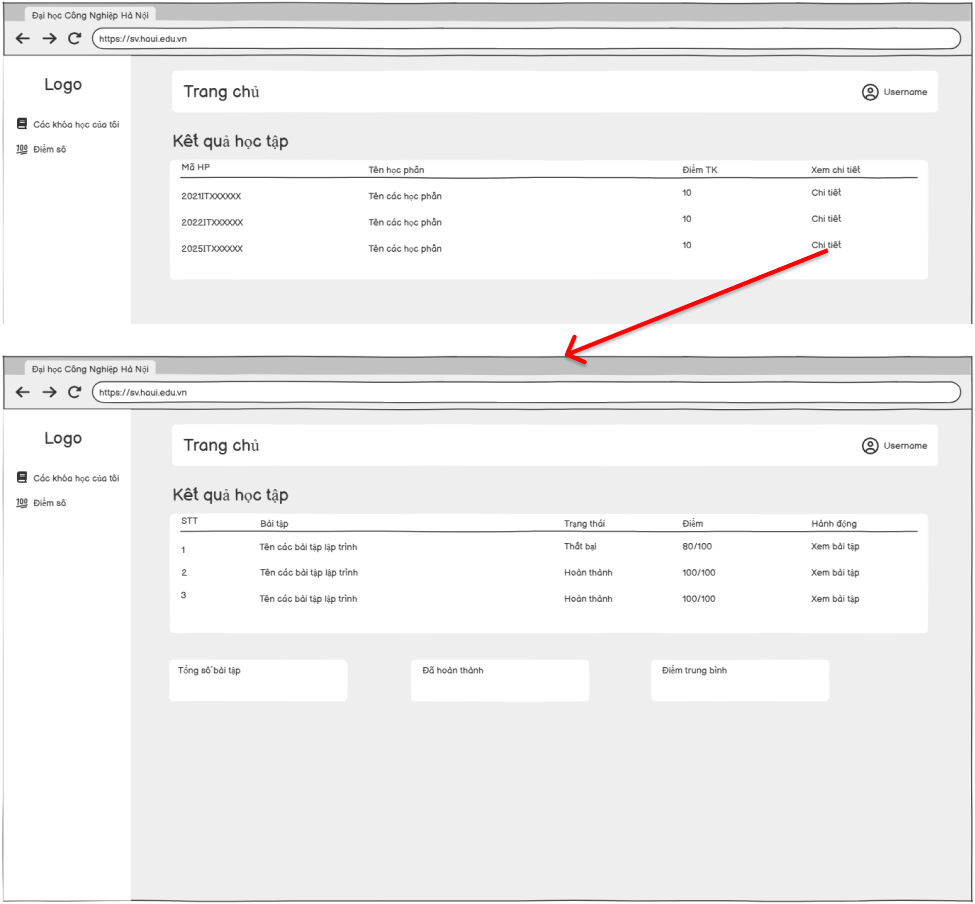
Khi chọn vào điểm số trên thanh menu sẽ hiển thị ra thông tin kết quả học thập dựa theo các lớp học phần.

****

Hình 2. 13: Thiết kế hình dung màn hình xem điểm số

### Thiết kế hình dung màn hình xem chi tiết điểm số

Khi chọn vào chi tiết sẽ hiển thị ra chi tiết kết quả học tập từng bài học của lớp học phần đó.

****

Hình 2. 14: Thiết kế hình dung màn hình xem chi tiết điểm số

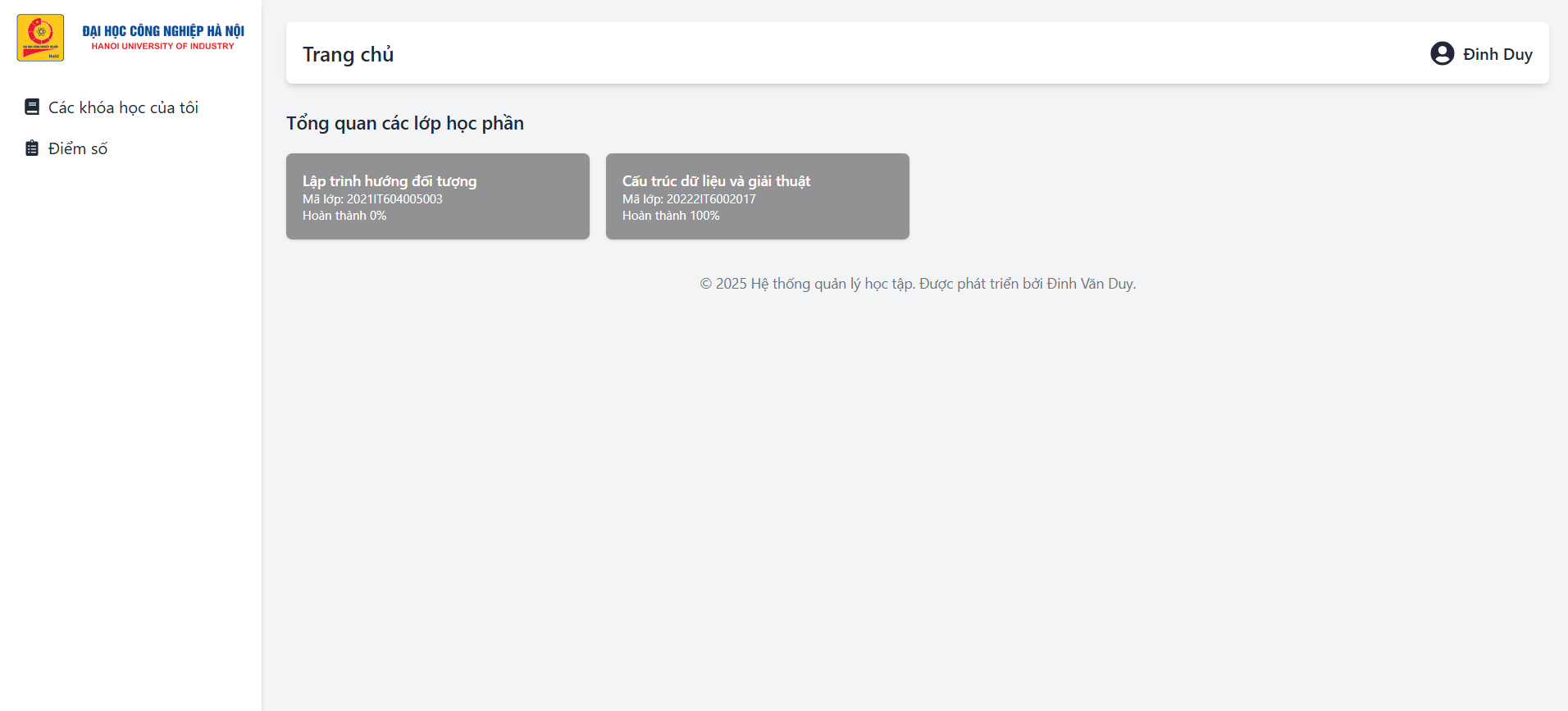
## Kết luận chương 2

Chương 2 đã trình bày về các yêu cầu chức năng, phi chức năng của hệ thống, lập biểu đồ use case tổng quát của hệ thống, đặc tả chi tiết các use case, vẽ các biểu đồ hành vi, biểu đồ lớp, thiết kế cơ sở dữ liệu mức vật lý. Từ đó bước vào giai đoạn tiếp theo của đề tài là cài đặt và kiểm thử chương trình.

# CÀI ĐẶT VÀ KIỂM THỬ

## Giao diện và chức năng phía sinh viên

### Trang chủ

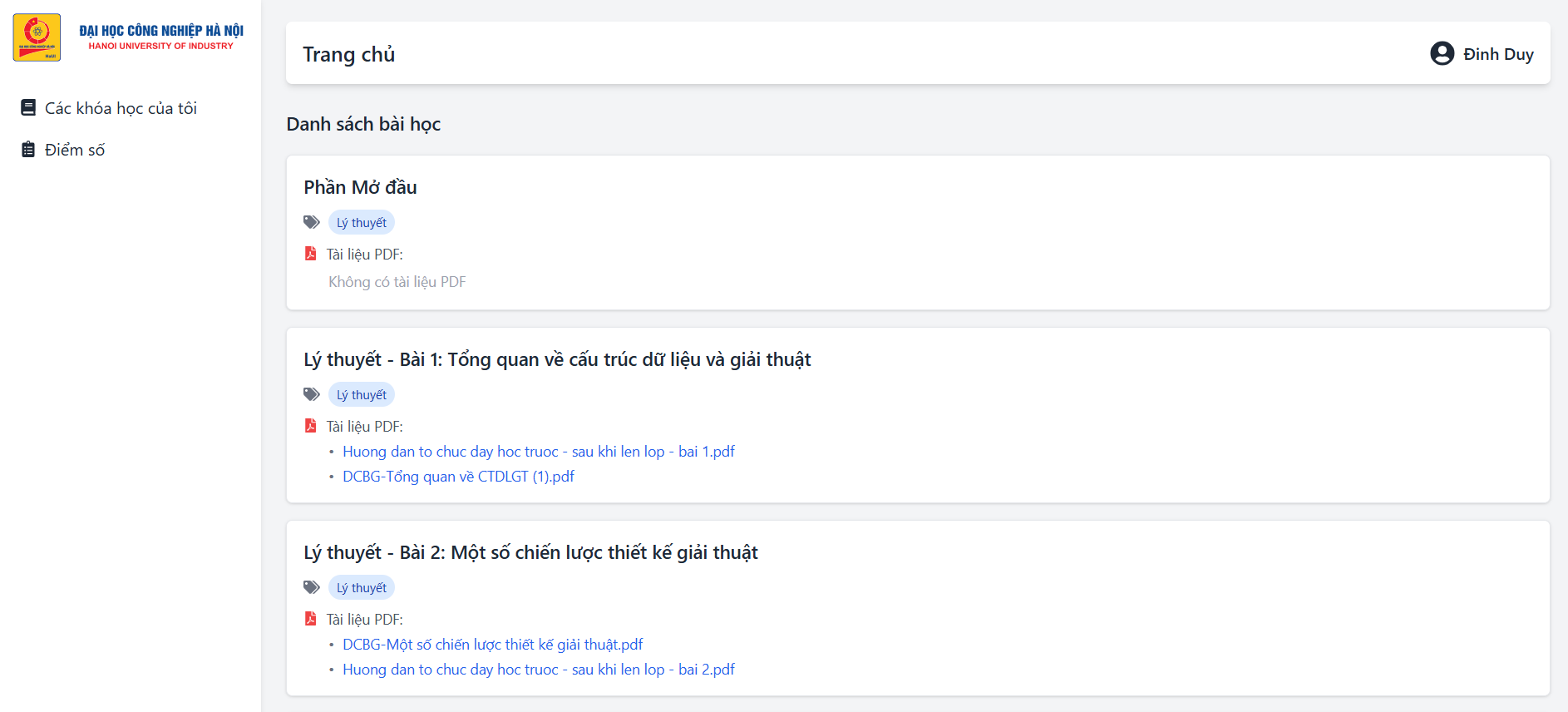
****

Hình 3. 1: Giao diện trang chủ

Trang chủ là phần mà sinh viên sẽ nhìn thấy đầu tiên khi truy cập vào hệ thống. Trang chủ sẽ hiển thị thông tin các lớp học phần mà sinh viên đã tham gia.

### Trang danh sách bài học

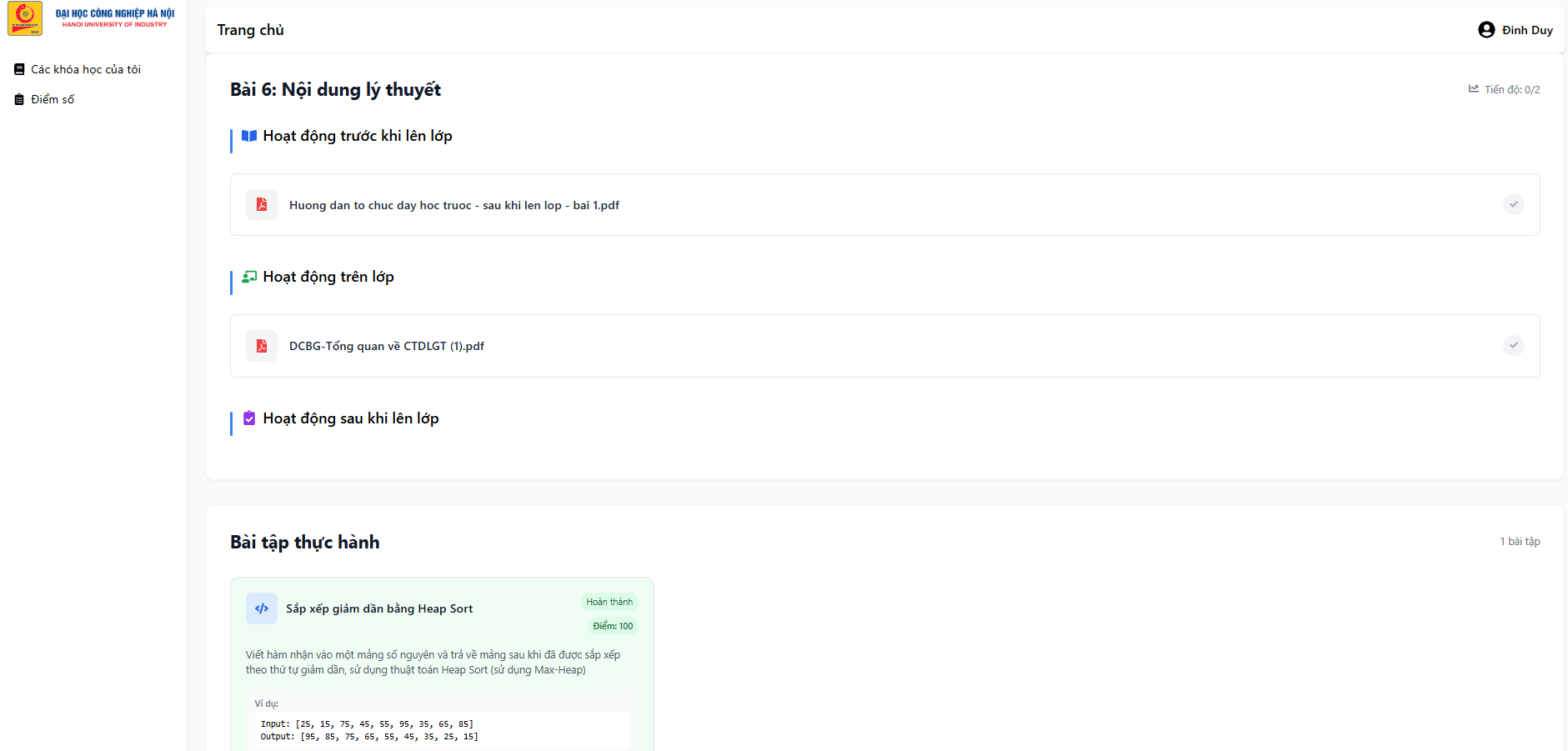
Sau khi bấm vào lớp học phần sẽ hiển thị ra danh sách các bài học có trong lớp học phần đó.



Hình 3. 2: Giao diện trang danh sách bài học*.*

### Trang nội dung bài học

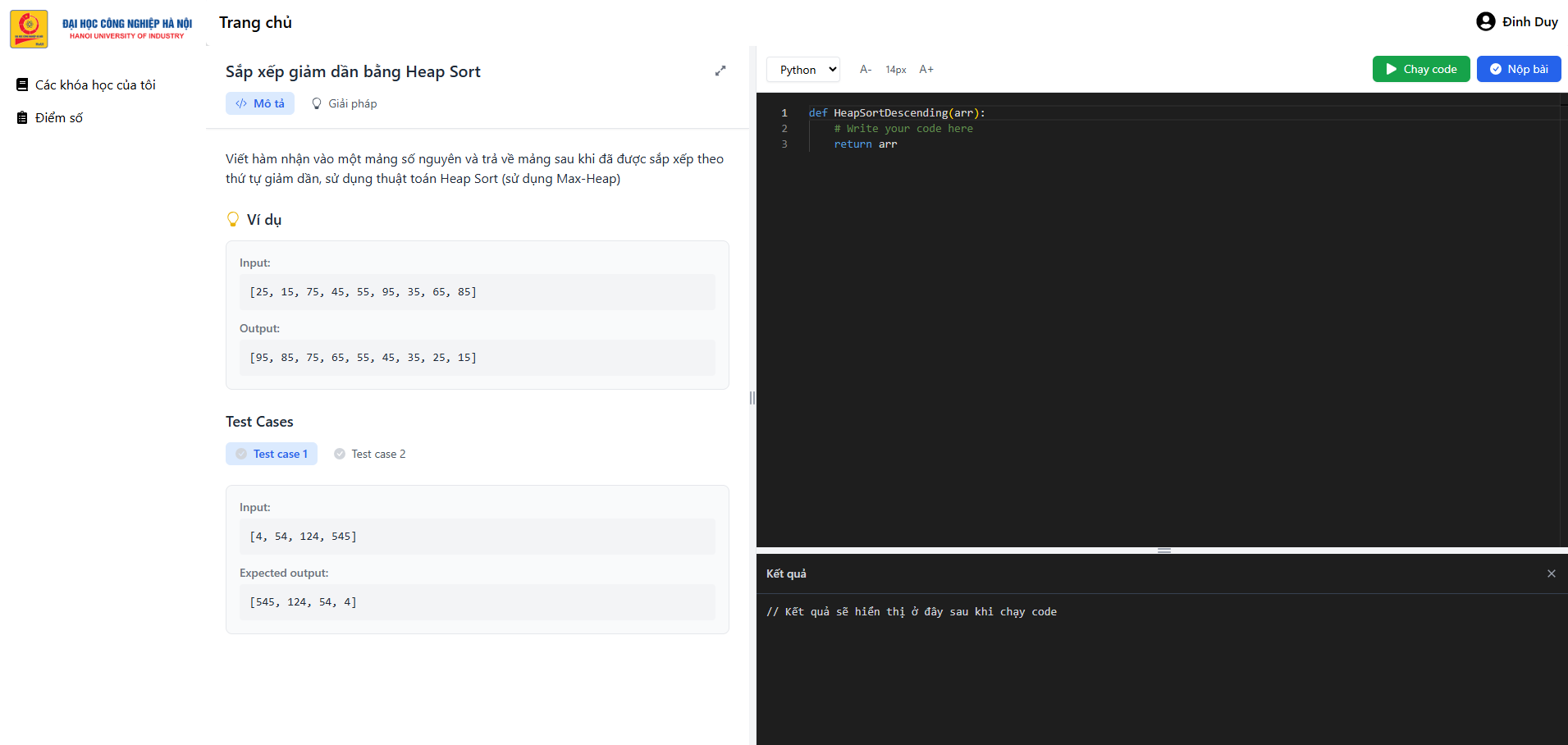
Sinh viên có thể nhìn thấy các nội dung bài học sau khi bấm vào bài học trong lớp học phần để học, nó sẽ hiển thị các nội dung như là nội dung bài tập, file tài liệu, bài tập thực hành.



Hình 3. 3: Giao diện trang nội dung bài học.

### Trang làm bài tập lập trình

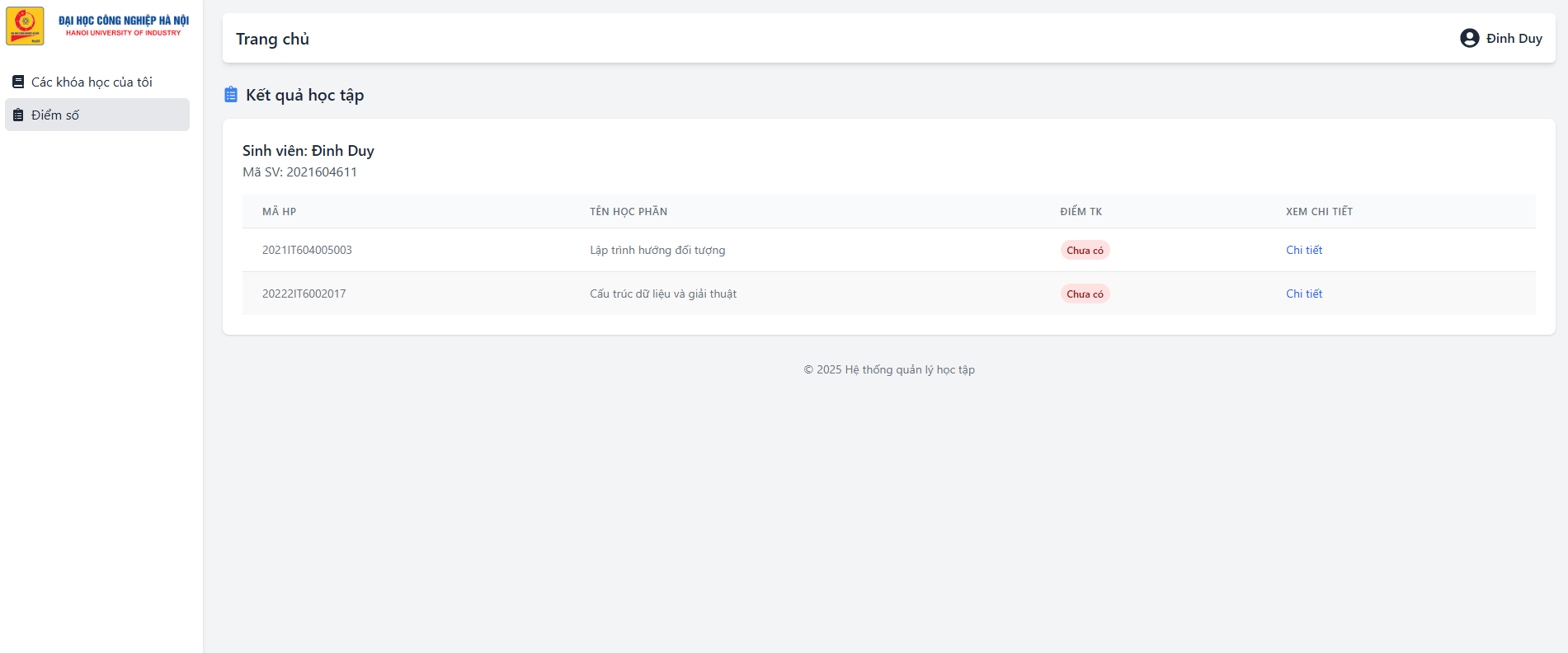
Trang làm bài tập lập trình sẽ hiển thị tất cả các nội dung liên quan đến bài tập code của giảng viên giao cho, cũng như có thể lập trình trực tiếp ngay trên web site



Hình 3. 4: Giao diện trang bài tập lập trình.

### Trang xem điểm số

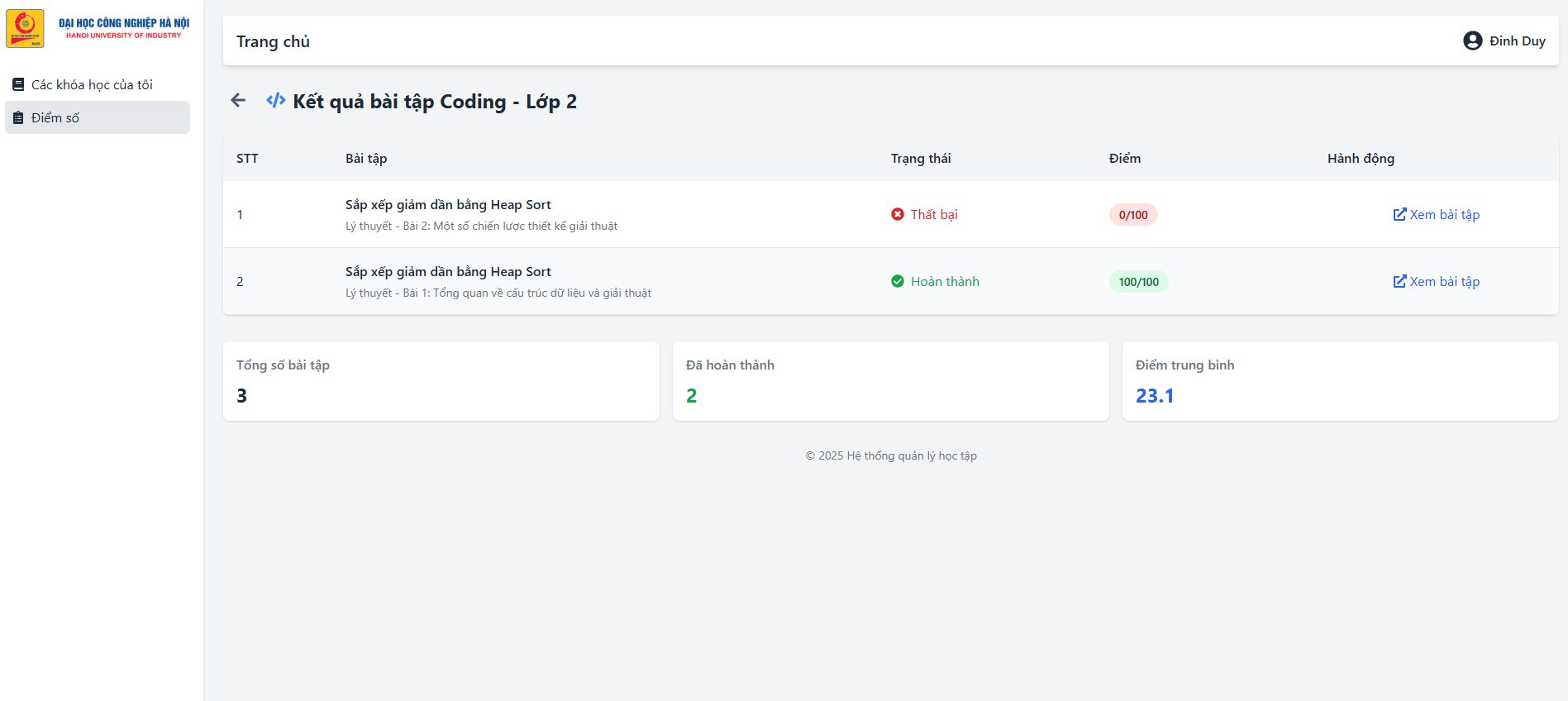
Tại đây sẽ hiển thị thông tin điểm theo lớp học phần của sinh viên.



Hình 3. 5: Giao diện trang xem điểm số

### Trang xem chi tiết điểm số

Trang này sẽ hiển thị thông tin tất cả các điểm số ở từng bài tập lập trình của sinh viên.

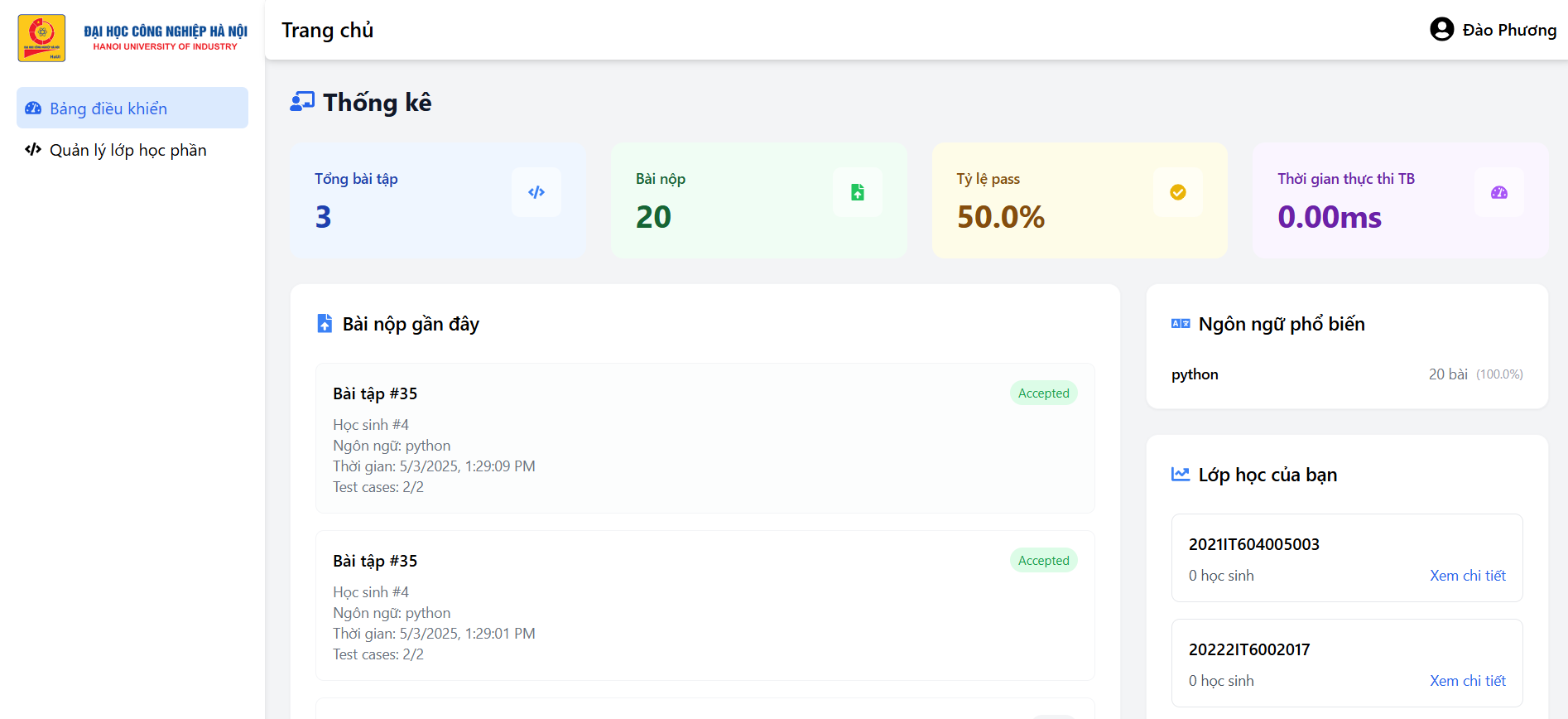


Hình 3. 6: Giao diện trang xem điểm số chi tiết.

## Giao diện và chức năng phía người quản trị

### Trang chủ của Giảng viên

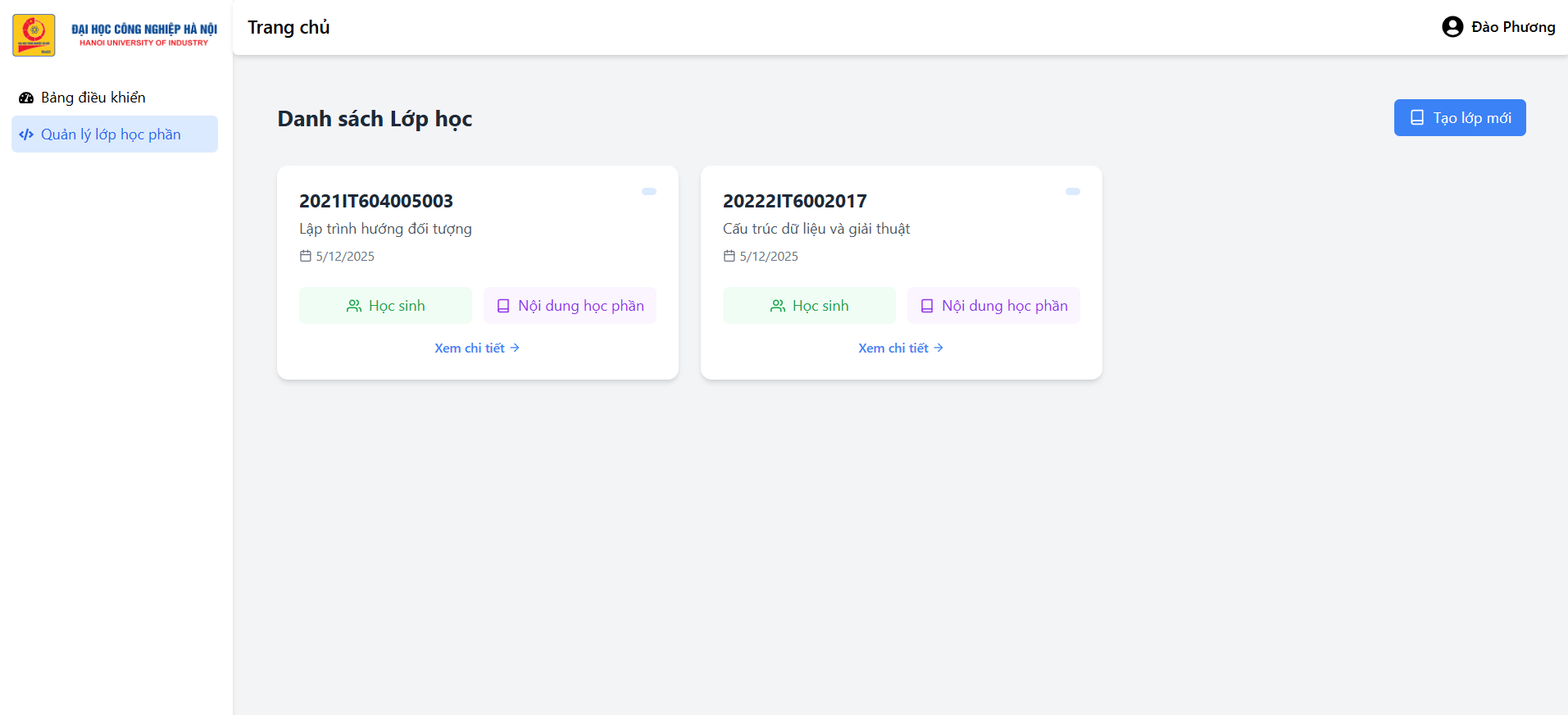
Trang chủ admin hiển hiển thị thống kê khái quát gồm: Tổng số bài tập, bài nộp, tỷ lệ pass, các bài nộp gần nhất, lớp học của giảng viên.



Hình 3. 7: Giao diện trang chủ của giảng viên

### Trang quản lý lớp học phần

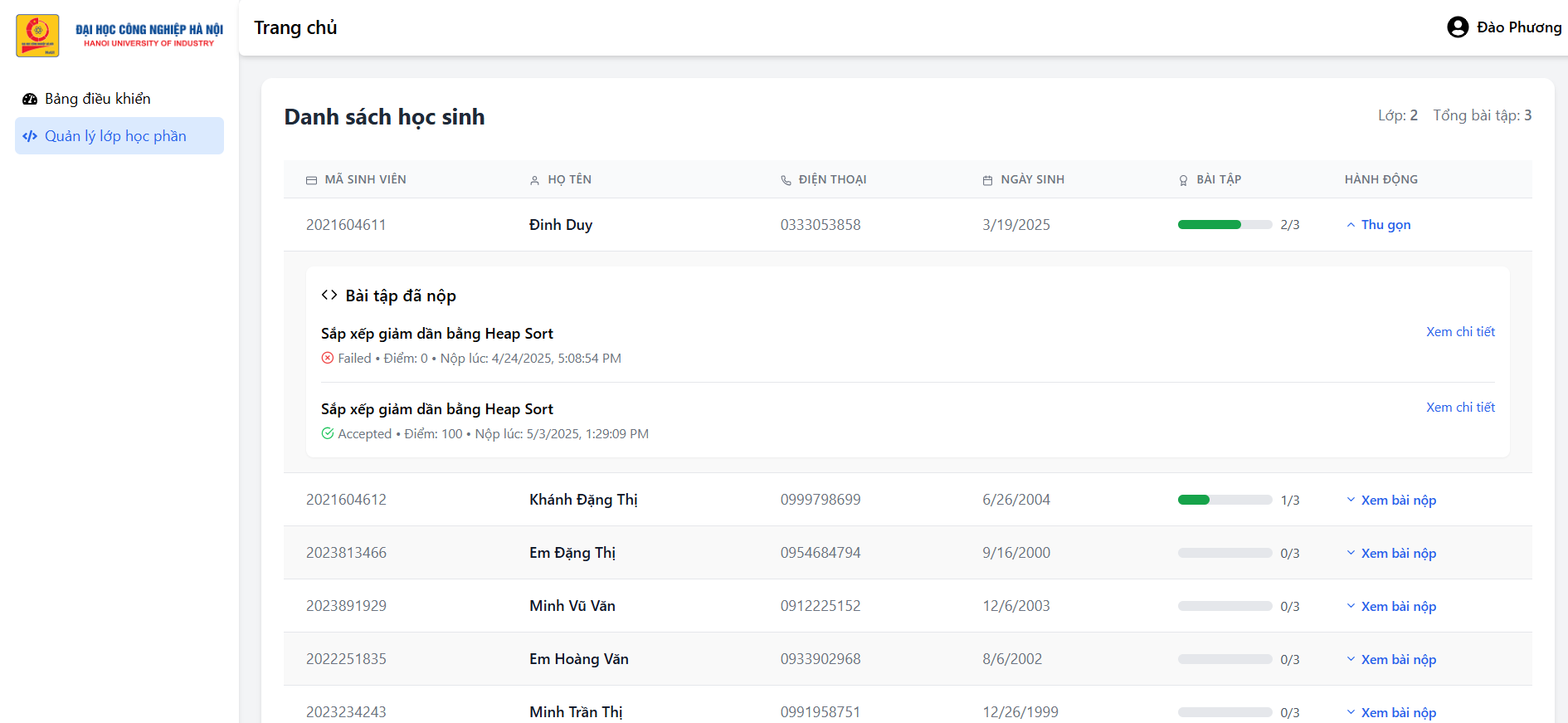
Tại đây sẽ hiển thị danh sách lớp học phần mà giảng viên quản lý, bao gồm quản lý cả học sinh, nội dung học phần trong từng lớp học phần



Hình 3. 8: Giao diện quản lý lớp học phần

### Trang quản lý học sinh

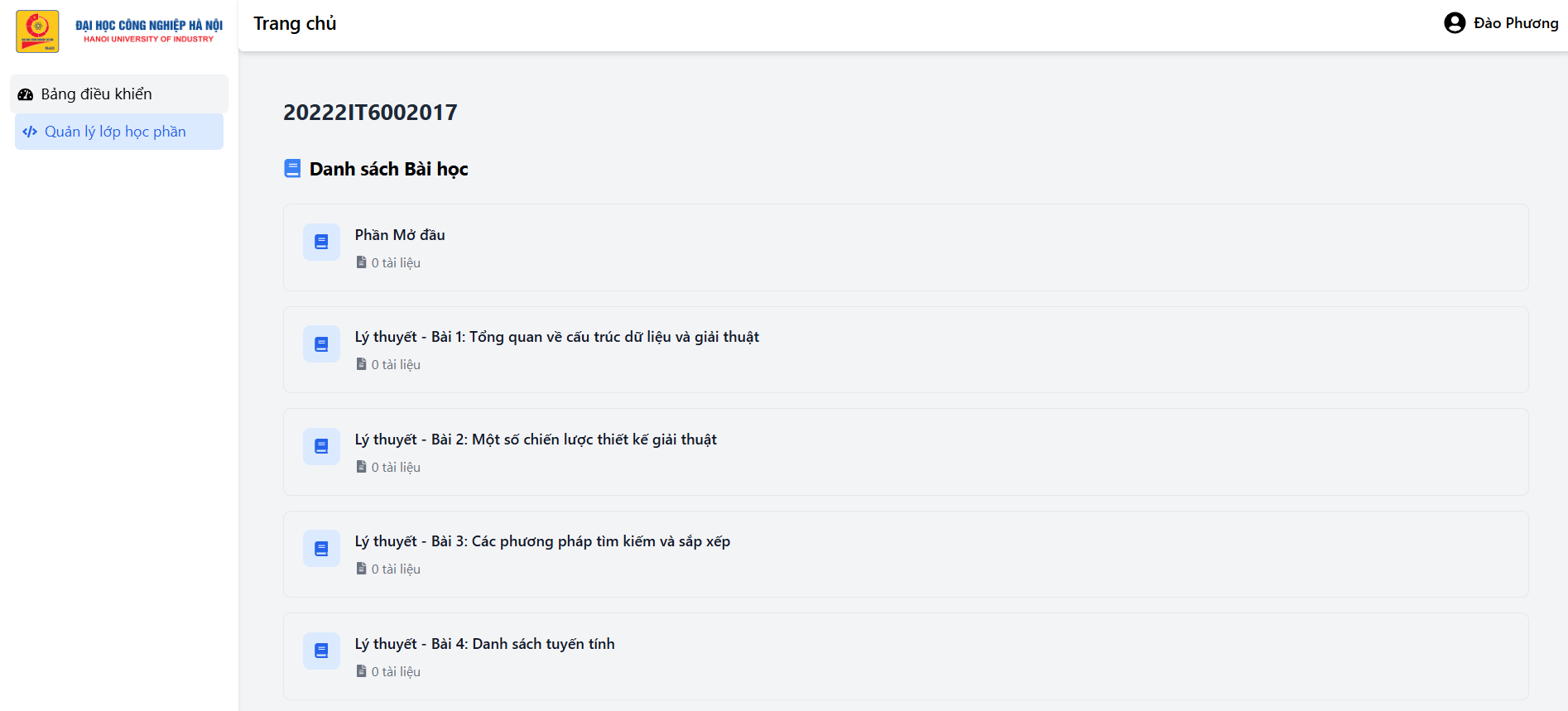
Tại trang quản lý học sinh sẽ hiển thị thông tin cơ bản của các học sinh kèm theo số lượng bài tập học sinh đã hoàn thành. Có thể xem chi tiết các bài tập mà học sinh đã nộp và số điểm của bài tập đó.



Hình 3. 9: Giao diện trang quản lý học sinh

### Trang danh sách bài học trong lớp học phần

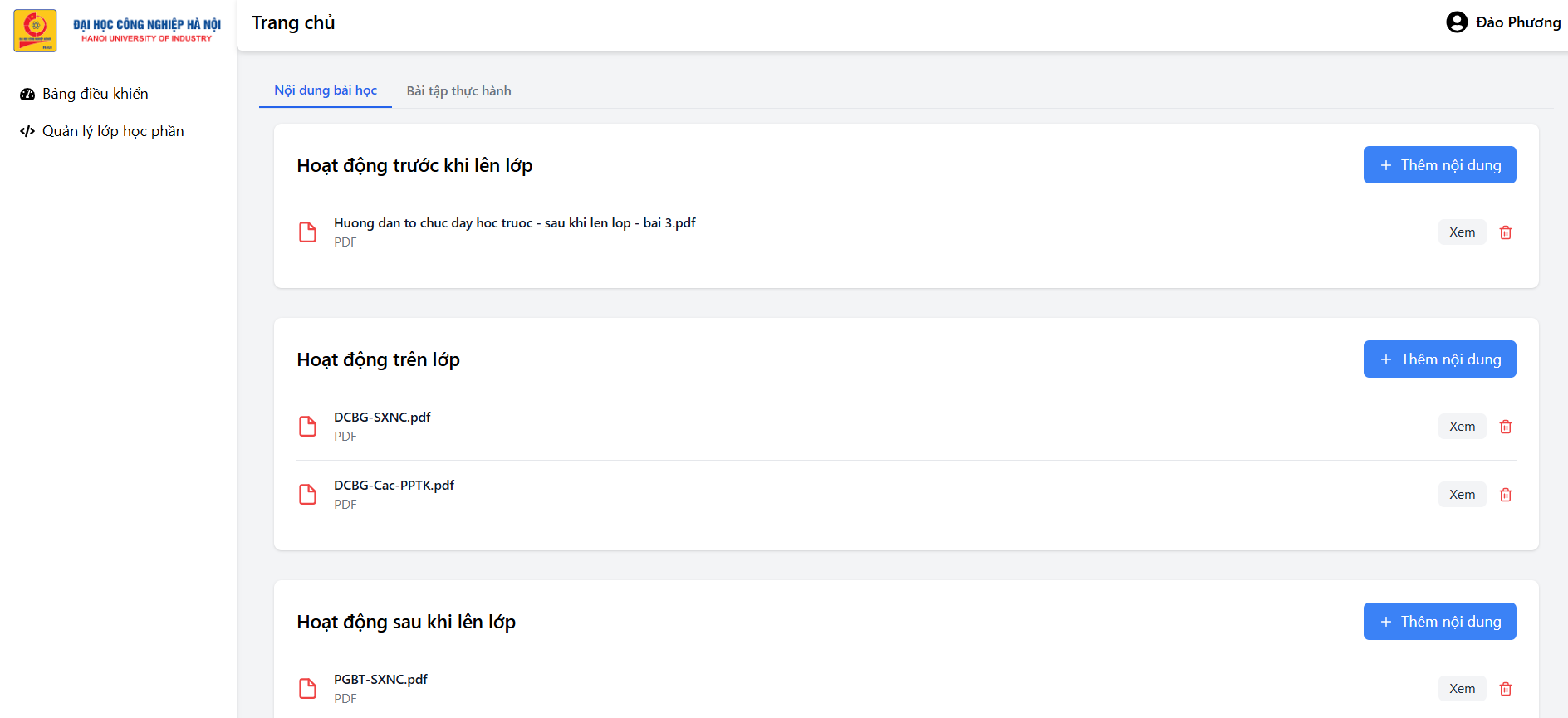
Trang này sẽ quản lý tất cả các bài học trong lớp học phần, sẽ hiển thị theo từng bài học và số lượng các tài liệu.



Hình 3. 10: Giao diện danh sách bài học

### Trang nội dung bài học

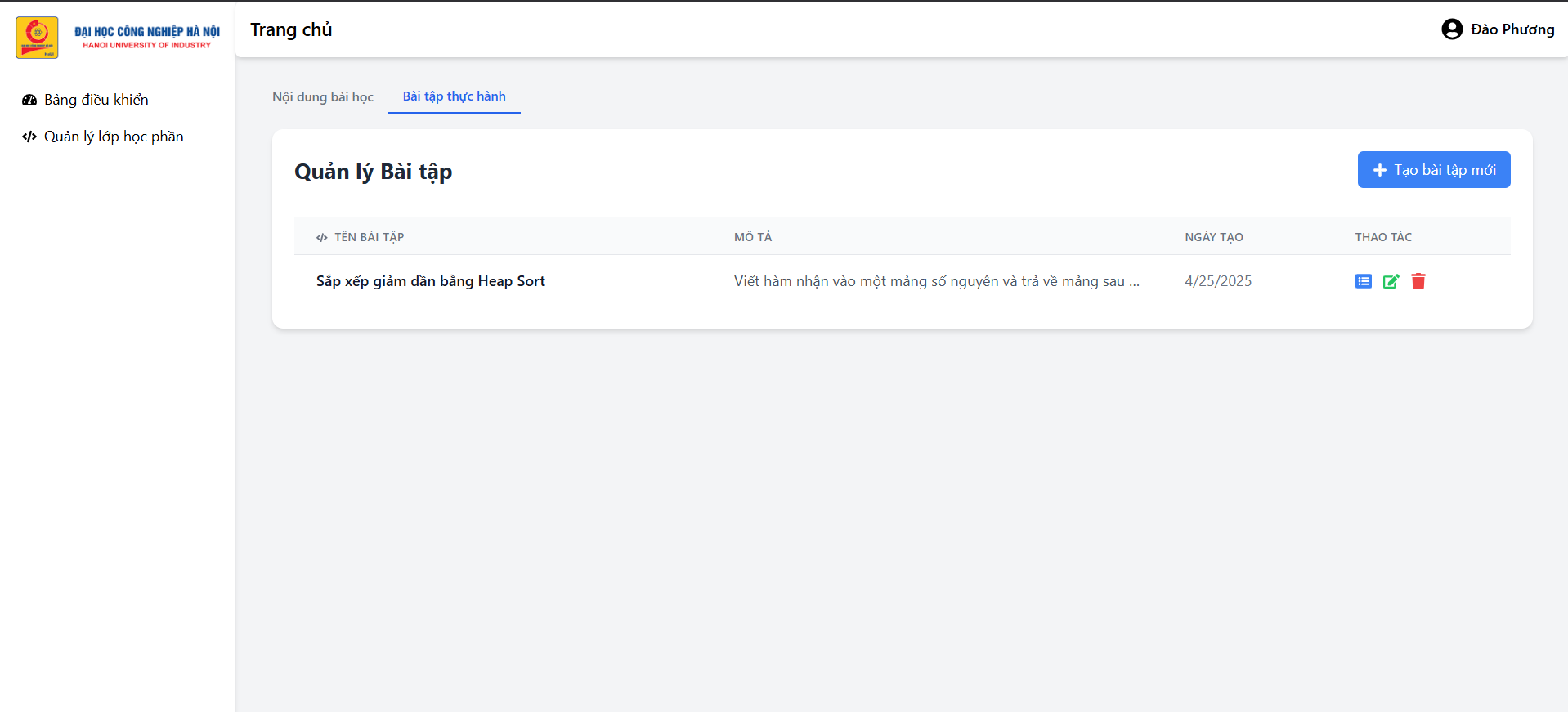
Trang nội dung bài học hiển thị thông tin các bài học theo 3 loại “Hoạt động trước khi lên lớp”, “Hoạt động trên lớp”, “Hoạt động sau khi lên lớp”. Trang này sẽ có thể thao tác xem, thêm sửa và xóa các nội dung trong bài học.



Hình 3. 11: Giao diện nội dung bài học

### Trang quản lý bài tập thực hành

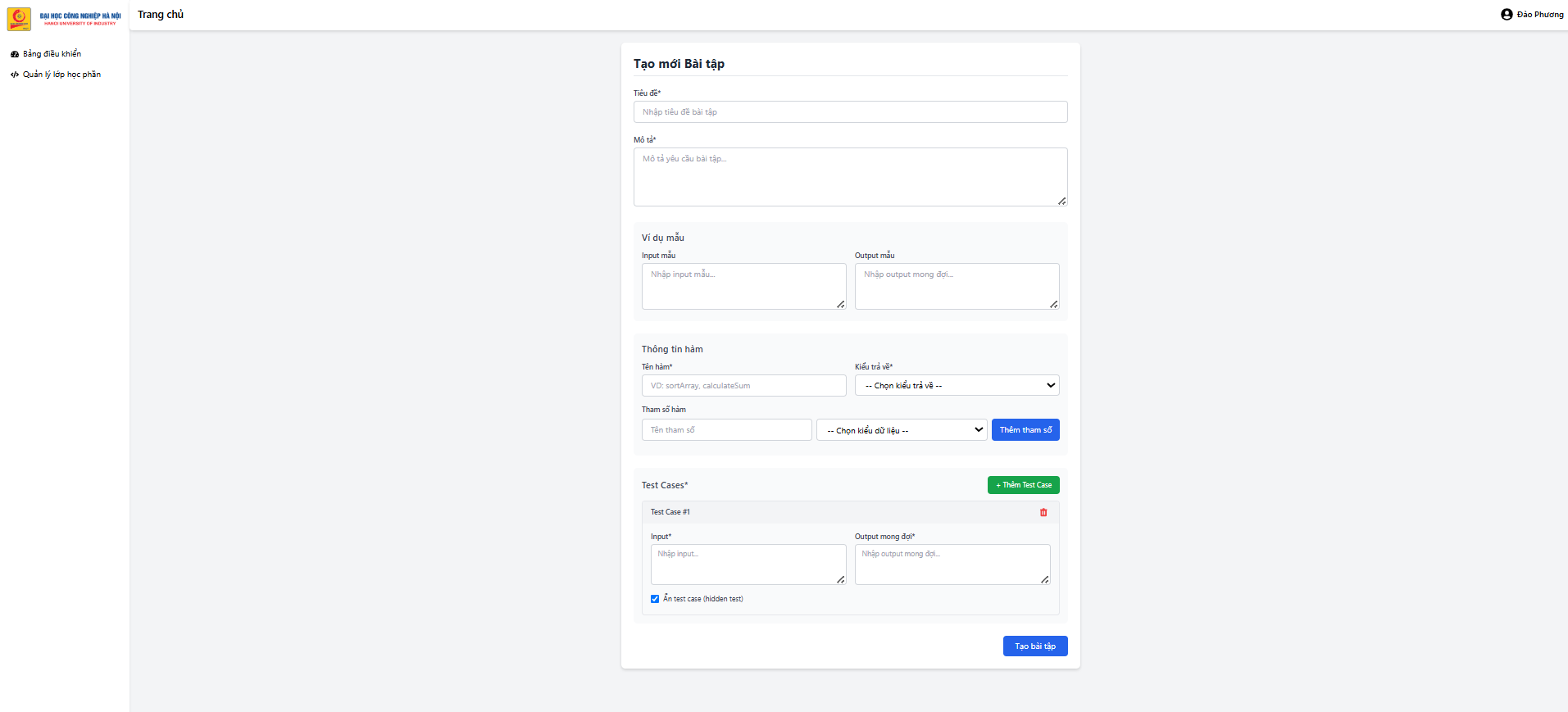
Tại đây giảng viên có thể thêm, sửa, tạo và xóa các bài tập thực hành theo từng nội dung bài học.



Hình 3. 12: Giao diện quản lý bài tập thực hành

### Trang tạo mới bài tập thực hành

Từ trang này giảng viên có thể tạo mới các bài tập thực hành, đây cũng là phần chính trong việc tạo một bài tập code giúp sinh viên có thể làm bài tập.



Hình 3. 13: Giao diện tạo bài tập thực hành

## Kiểm thử

Kiểm thử là một phần không thể thiếu trong tất cả các ứng dụng, chương trình hiện nay. Vì thế em sẽ tiến hành kiểm thử trên phần mềm của mình. Phạm vi kiểm thử của chương trình chính là cả frontend và backend, em sẽ trình bày nội dung kiểm thử qua những phần chức năng chính trong chương trình.

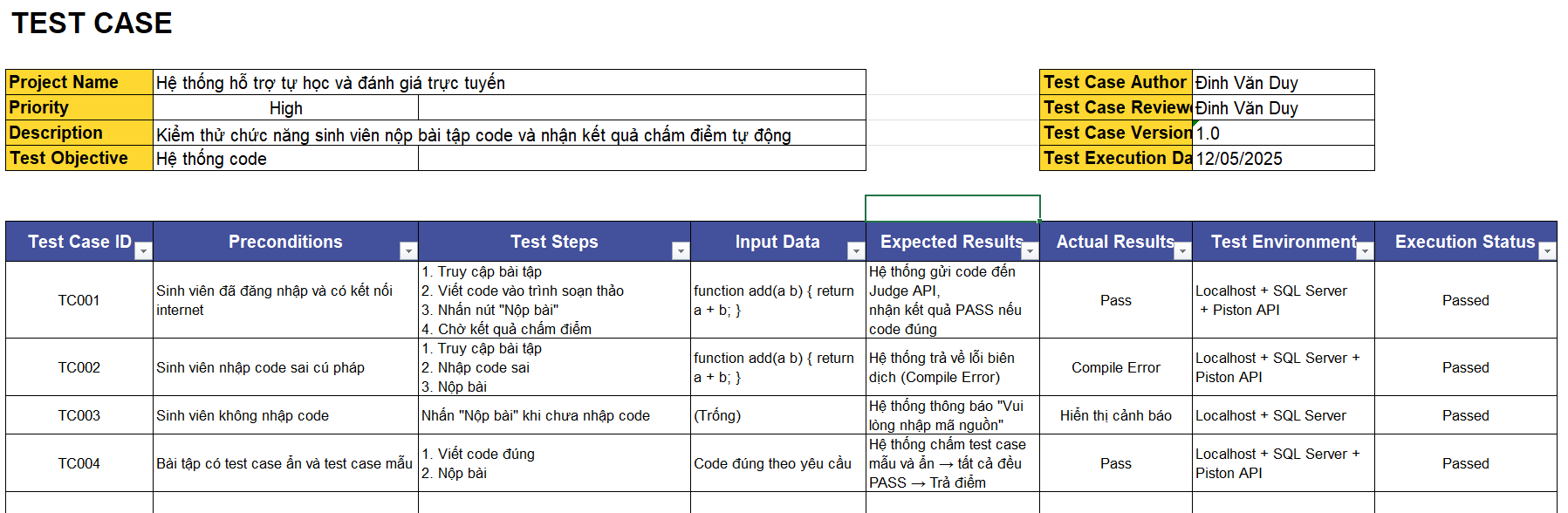
### Phạm vi kiểm thử

Những chức năng sẽ được kiểm thử:

* Làm bài tập thực hành:
* Làm bài tập thực hành, kiểm tra các test case.
* Kiểm tra syntax trong đoạn code của sinh viên.
* Hiển thị nội dung bài học:
* Hiển thị nội dung bài học có tài liệu.
* Hiển thị nội dung bài học không có tài liệu.
* Hiển thị điểm của các sinh viên đã nộp

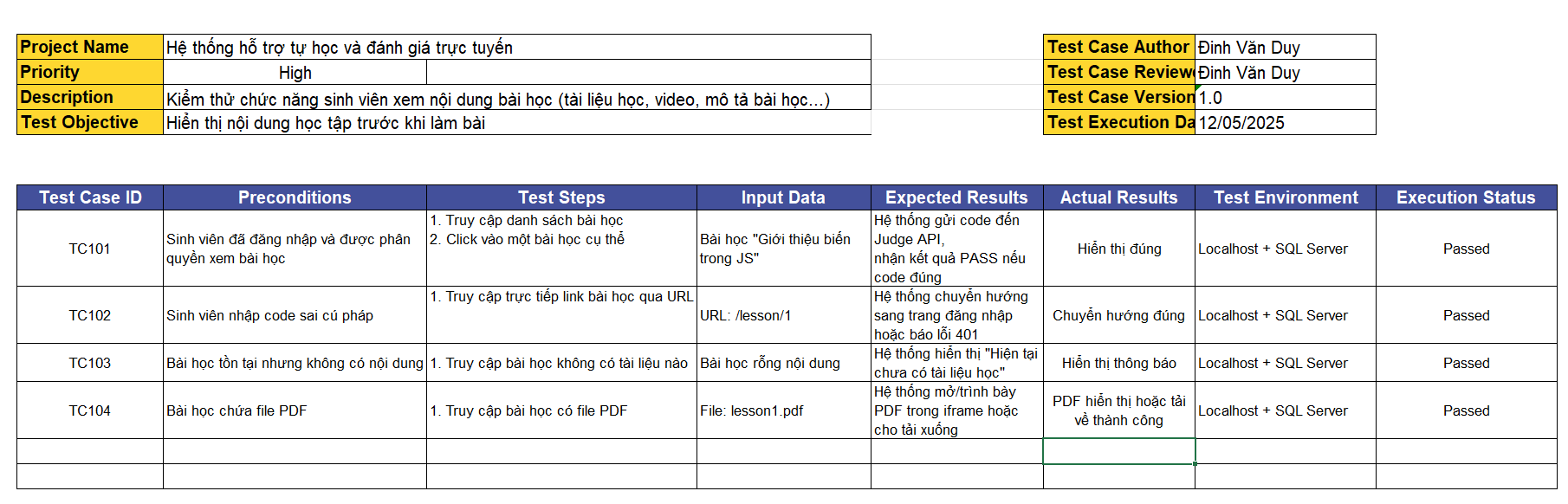
### Test case

#### Test case kiểm tra chức năng làm bài tập thực hành

**

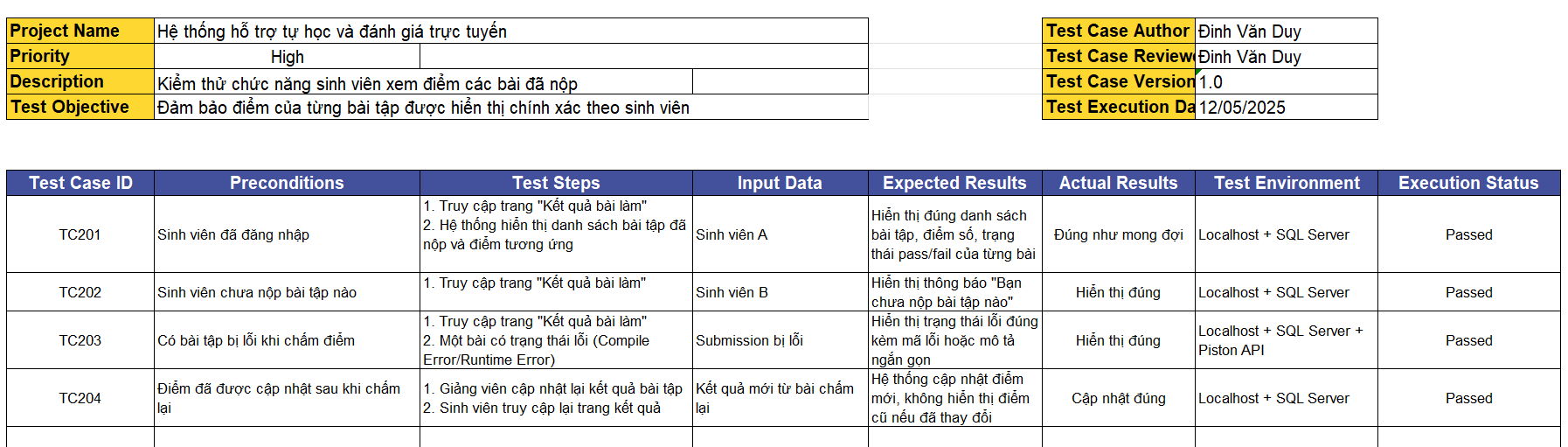
Hình 3. 14: Test case làm bài tập thực hành

#### Test case hiển thị nội dung bài học

**

Hình 3. 15: Test case hiển thị nội dung bài học

#### Test case Xem điểm của sinh viên

**

Hình 3. 16: Test case xem điểm của sinh viên

### Báo cáo kiểm thử

● Về giao diện người dùng:

Giao diện người dùng của hệ thống được thiết kế thân thiện, dễ quan sát và dễ thao tác. Hệ thống đảm bảo khả năng hiển thị tốt trên nhiều kích thước màn hình và thiết bị khác nhau, đáp ứng yêu cầu về khả năng tương thích và trải nghiệm người dùng.

● Về chức năng của chương trình:

Các chức năng được triển khai đúng theo thiết kế ban đầu, có tính logic cao và dễ sử dụng. Toàn bộ các yêu cầu từ phía người dùng được hệ thống xử lý chính xác, trơn tru và nhanh chóng. Trong quá trình kiểm thử, không phát hiện lỗi nghiêm trọng nào ảnh hưởng đến trải nghiệm sử dụng.

● Về hiệu năng:

Hệ thống phản hồi nhanh trong các tình huống kiểm thử thông thường. Các thao tác như gửi bài, chấm điểm, truy xuất dữ liệu được xử lý kịp thời, không xuất hiện hiện tượng treo máy hoặc gián đoạn. Tuy nhiên, do chưa thực hiện kiểm thử với số lượng người dùng truy cập lớn, nên khả năng chịu tải và hiệu suất thực tế vẫn cần được đánh giá thêm trong các giai đoạn tiếp theo.

● Về bảo mật:

Hệ thống có các cơ chế xác thực tài khoản cơ bản, đảm bảo người dùng không thể truy cập trái phép vào khu vực dành cho đối tượng khác (ví dụ: sinh viên không thể truy cập trang quản trị của giảng viên). Tuy nhiên, trong quá trình kiểm thử, chưa thực hiện các biện pháp kiểm thử bảo mật chuyên sâu như tấn công SQL Injection, XSS hay kiểm thử mã hóa dữ liệu. Do đó, cần bổ sung các kiểm thử bảo mật ở giai đoạn tiếp theo nếu triển khai thực tế.

## Tổng kết chương 3

Chương 3 đã trình bày chi tiết về giao diện người dùng và các chức năng chính dành cho sinh viên và giảng viên trong hệ thống hỗ trợ tự học và đánh giá trực tuyến. Sau quá trình cài đặt hệ thống là phần kiểm thử, tập trung vào các chức năng trọng tâm như: nộp bài tập lập trình, chấm điểm tự động, hiển thị kết quả, quản lý bài học và nội dung học tập. Các test case được xây dựng và thực thi nhằm đánh giá tính chính xác, hiệu quả và độ ổn định của hệ thống.

Kết quả kiểm thử cho thấy hệ thống hoạt động ổn định, giao diện thân thiện và các chức năng đáp ứng tốt yêu cầu đề ra. Bên cạnh đó, một số đánh giá sơ bộ về hiệu năng, bảo mật và khả năng mở rộng của hệ thống cũng được đưa ra ở phần báo cáo kiểm thử.

Kết thúc chương 3, quá trình xây dựng và kiểm thử hệ thống hỗ trợ tự học và đánh giá trực tuyến đã hoàn thành. Phần tiếp theo của báo cáo sẽ trình bày kết luận và định hướng phát triển hệ thống trong tương lai.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

**1. Kết luận**

Sau quá trình nghiên cứu và thực hiện đồ án tốt nghiệp, hệ thống hỗ trợ tự học và đánh giá trực tuyến dành cho sinh viên ngành lập trình đã đạt được các kết quả như sau:

Nắm vững kiến thức về lập trình web hiện đại với ASP.NET Core Web API, ReactJS, Entity Framework, SQL Server, cũng như các công nghệ hỗ trợ như Docker, Piston API, Git và GitHub.

Hiểu rõ quy trình xây dựng hệ thống phần mềm từ khâu phân tích yêu cầu, thiết kế kiến trúc hệ thống, đến triển khai backend và frontend riêng biệt.

Ứng dụng mô hình phân lớp rõ ràng, chia tách các tầng: Presentation, Application, Domain và Infrastructure.Lập kế hoạch, thực hiện kiểm thử và báo cáo kiểm thử cho một website.

Thực hiện kiểm thử tính năng, kiểm thử chức năng gửi và chấm code thông qua Piston API, đánh giá hiệu quả và độ chính xác của hệ thống.

Sử dụng Git và GitHub để quản lý mã nguồn, làm việc có tổ chức và dễ dàng mở rộng về sau.

**2. Vấn đề tồn tại và hướng phát triển**

Hiện tại, hệ thống mới chỉ được triển khai ở mức cơ bản, chủ yếu phục vụ mục tiêu kiểm chứng kỹ thuật và đáp ứng bài toán trong khuôn khổ đồ án, nên vẫn còn một số hạn chế:

Hệ thống chưa được triển khai thực tế cho số lượng người dùng lớn, nên chưa thể đánh giá được hiệu năng và tính ổn định khi có nhiều sinh viên làm bài cùng lúc.

Giao diện và trải nghiệm người dùng còn đơn giản, chưa tối ưu toàn diện cho thiết bị di động hoặc trình duyệt cũ.

**3. Hướng phát triển trong tương lai**

Triển khai hệ thống trên môi trường thực tế (cloud server hoặc trường đại học), tiến hành thử nghiệm với nhiều sinh viên để đánh giá khả năng chịu tải và mức độ ổn định.

Tối ưu hiệu suất hệ thống, đặc biệt là phần xử lý chấm điểm tự động để giảm thời gian phản hồi.

Cải tiến giao diện và nâng cao trải nghiệm người dùng, hướng đến tiêu chuẩn sản phẩm thương mại.

# TÀI LIỆU THAM KHẢO

**Tiếng Việt**

1. Website Hệ thống học kết hợp của Trường Đại học Công Nghiệp Hà Nội

Từ: https://qlht.haui.edu.vn/my/

1. Tài liệu C# của Microsoft

Từ: https://learn.microsoft.com/vi-vn/dotnet/csharp/tour-of-csharp/

1. Tài liệu ReactJS *.*

Từ: https://react.dev/learn

1. MSSQL (2024), *Hệ quản trị cơ sở dữ liệu MSSQL.*

Từ: https://learn.microsoft.com/vi-vn/sql/?view=sql-server-ver16

1. Nguyễn Trung Phú (Chủ biên), Trần Thị Phương Nhung, Đỗ Thị Minh Nguyệt (2019), *Giáo trình thiết kế web*, Nhà xuất bản Thống kê.

**Tiếng Anh**

1. Json Web Token.

Từ: https://jwt.io/introduction

1. Postman documentation.  
   Từ: https://learning.postman.com/docs/introduction/overview/
2. Docker.

Từ: https://docs.docker.com/