

**BỘ GIÁO DỤC ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM**



ĐỒ ÁN TỐT NGHIỆP

**XÂY DỰNG ỨNG DỤNG WEBSITE THỰC HIỆN
ĐĂNG KÝ, HỒ SƠ SINH VIÊN DU HỌC NƯỚC NGOÀI
CỦA VIỆN ĐÀO TẠO QUỐC TẾ ĐẠI HỌC ĐẠI NAM**

SINH VIÊN THỰC HIỆN : VŨ ĐÌNH HÙNG

MÃ SINH VIÊN : 1451020120

KHOA : CÔNG NGHỆ THÔNG TIN

HÀ NỘI - 2024

BỘ GIÁO DỤC ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM



VŨ ĐÌNH HƯNG

**XÂY DỰNG ỨNG DỤNG WEBSITE THỰC HIỆN ĐĂNG
KÝ, HỒ SƠ SINH VIÊN DU HỌC NƯỚC NGOÀI CỦA VIỆN
ĐÀO TẠO QUỐC TẾ ĐẠI HỌC ĐẠI NAM**

CHUYÊN NGÀNH : CÔNG NGHỆ THÔNG TIN

MÃ SỐ : 74.80.201

GIẢNG VIÊN HƯỚNG DẪN: TS.ĐẬU HẢI PHONG

HÀ NỘI – 2024

LỜI CAM ĐOAN

Tôi xin cam đoan rằng báo cáo có tên: “Xây Dựng Ứng Dụng Website Thực Hiện Đăng Ký, Hồ Sơ Sinh Viên Du Học Nước Ngoài của Viện Đào Tạo Quốc Tế - Đại Học Đại Nam” là kết quả của công việc nghiêm túc và tự chủ của bản thân. Trong quá trình nghiên cứu và phát triển dự án này, tôi đã tuân thủ các nguyên tắc và chuẩn mực đạo đức nghiêm ngặt.

Mọi thông tin, số liệu và kết quả được trình bày trong báo cáo đều tuân thủ nguyên tắc trung thực và minh bạch nhất có thể. Tôi đã sử dụng tài liệu tham khảo và nguồn thông tin từ các nguồn uy tín, và những phần này đã được dẫn chứng rõ ràng và nêu rõ trong phần tài liệu tham khảo.

Đồng thời, tôi cam kết rằng nếu có bất kỳ sai sót nào trong thông tin trình bày, tôi sẽ chịu trách nhiệm hoàn toàn và tuân theo mọi quy định và kỷ luật do bộ môn và nhà trường áp đặt.

Tôi cũng xác nhận rằng quy trình kiểm tra, xác minh thông tin và kiểm soát chất lượng đã được thực hiện để đảm bảo tính chính xác và đáng tin cậy của mọi thông tin được trình bày trong báo cáo này.

Tôi xin chịu mọi trách nhiệm về lời cam đoan này.

Hà Nội, Ngày 23 tháng 05 năm 2024

Người cam đoan

LỜI CẢM ƠN

Tôi xin chân thành cảm ơn các thầy cô trong khoa CNTT cũng như các thầy cô giảng dạy trong trường đã giúp em có được những kiến thức tổng quan để hoàn thành bài báo cáo này. Đặc biệt, tôi muốn gửi lời cảm ơn sâu sắc nhất đến thầy Đậu Hải Phong người đã đồng hành và hướng dẫn tôi suốt quãng thời gian làm đồ án tốt nghiệp của tôi. Trong quá trình này, sự tận tâm và sự quan tâm từ thầy đã là nguồn động viên lớn, giúp tôi tiếp cận kiến thức một cách sâu sắc và tự tin hơn khi tiếp cận dự án thực tế của mình.

Tôi rất mong nhận được sự đánh giá và góp ý chân thành từ thầy để bài báo cáo của tôi có thể được cải thiện và hoàn thiện hơn. Lời hướng dẫn và ý kiến từ thầy sẽ giúp tôi nắm bắt rõ hơn những khía cạnh cần điều chỉnh, từ đó phát triển kỹ năng và hiểu biết của mình một cách toàn diện hơn.

Bài báo cáo của tôi có thể chưa tránh khỏi những sai sót do những hạn chế trong quá trình thực hiện. Mong rằng thầy có thể xem xét và góp ý để giúp tôi hoàn thiện bài báo cáo một cách tốt nhất. Sự chỉ dẫn và phản hồi từ thầy sẽ giúp tôi cải thiện kiến thức và kỹ năng một cách toàn diện hơn.

Em xin chân thành cảm ơn!

LỜI MỞ ĐẦU

Trong bối cảnh toàn cầu hóa ngày càng sâu rộng, nhu cầu du học và trao đổi quốc tế ngày càng trở nên phổ biến và thiết yếu. Viện Đào Tạo Quốc Tế - Đại Học Đại Nam đã nhận thấy tầm quan trọng của việc cung cấp một hệ thống quản lý đăng ký và hồ sơ sinh viên du học nước ngoài hiệu quả và tiện lợi. Trước tình hình đó, việc xây dựng một ứng dụng website đáp ứng nhu cầu này không chỉ là một bước tiến quan trọng trong việc nâng cao chất lượng dịch vụ đào tạo, mà còn góp phần tạo điều kiện thuận lợi cho sinh viên trong quá trình thực hiện các thủ tục liên quan.

Báo cáo này sẽ trình bày chi tiết quá trình xây dựng và triển khai ứng dụng website dành cho việc đăng ký và quản lý hồ sơ sinh viên du học nước ngoài. Thông qua đó, Viện Đào Tạo Quốc Tế - Đại Học Đại Nam hướng tới mục tiêu tối ưu hóa quy trình quản lý, tăng cường tính minh bạch và hiệu quả, đồng thời mang lại trải nghiệm tốt nhất cho người dùng. Hệ thống này không chỉ giúp giảm tải công việc hành chính, mà còn mở ra một kênh giao tiếp hiện đại, chuyên nghiệp giữa nhà trường và sinh viên, góp phần nâng cao uy tín và vị thế của Viện trên trường quốc tế.

Báo cáo sẽ bao gồm các phần chính như sau: phân tích yêu cầu và thiết kế hệ thống, triển khai và thử nghiệm, đánh giá và cải tiến. Hy vọng rằng, qua báo cáo này, quý độc giả sẽ có cái nhìn toàn diện về dự án và thấy rõ những lợi ích thiết thực mà ứng dụng mang lại cho cả nhà trường và sinh viên.

Em xin chân thành cảm ơn!

NHẬN XÉT

This image shows a full page of primary-ruled paper. It features multiple sets of horizontal dashed lines, each set consisting of three lines (top, middle, bottom) to guide letter height. The lines are evenly spaced across the entire page, providing a template for handwriting practice. There are no margins, text, or other markings on the paper.

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI.....	13
1.1. Giới thiệu về dự án và mục tiêu nghiên cứu	13
1.2. Lý do chọn đề tài và tầm quan trọng của đề tài	13
1.2.1. Lý do chọn đề tài	13
1.2.2. Tầm quan trọng của đề tài.....	13
1.3. Phạm vi nghiên cứu.....	14
1.4. Phương pháp nghiên cứu:.....	15
CHƯƠNG 2: KIẾN TRÚC HỆ THỐNG VÀ CÔNG NGHỆ SỬ DỤNG	16
2.1. Công nghệ sử dụng	16
2.1.1. Giới thiệu tổng quan về HTML.....	16
2.1.2. Giới thiệu tổng quan về CSS.....	16
2.1.3. Giới thiệu tổng quan về JavaScript	17
2.1.4. Giới thiệu tổng quan về ASP.NET Core framework.....	17
2.1.5. Lý do sử dụng các công nghệ trên.....	20
2.2. Kiến trúc của hệ thống	21
2.2.1. Giới thiệu về kiến trúc MVC.....	21
2.2.2. Lý do sử dụng kiến trúc MVC.....	25
2.3. Giới thiệu về hệ quản trị cơ sở dữ liệu của hệ thống.....	27
2.3.1. Giới thiệu về hệ quản trị cơ sở dữ liệu SQL Server	27
2.3.2. Thành phần chính của SQL Server.....	28
2.3.2. Giới thiệu về SSMS.....	28
2.4. Triển khai ứng dụng.....	30
2.4.1. Chuẩn bị Môi trường.....	30
2.4.2. Tạo dự án ASP.NET MVC	30
2.4.3. Thiết kế Cơ sở Dữ liệu:.....	31
2.4.4. Triển khai Mô hình MVC:	31
2.4.5. Kết nối với Cơ sở Dữ liệu:.....	32
2.4.6. Kiểm tra và Triển khai:	32

CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	34
3.1. <i>Đặc tả</i>	34
3.1.1. <i>Sinh viên</i>	34
3.1.2 <i>Admin</i>	38
3.2 <i>Thiết kế cơ sở dữ liệu</i>	45
3.2.1 <i>Sơ đồ ERD</i>	45
3.2.2 <i>Cài đặt cơ sở dữ liệu</i>	45
3.3 <i>Use case</i>	51
3.3.1 <i>Use case tổng quát</i>	51
3.3.2 <i>Quản lý Thông tin sinh viên của Sinh Viên</i>	52
3.3.3 <i>Quản lý thông tin gia đình của sinh viên</i>	53
3.3.4 <i>Quản lý khóa du học sinh viên</i>	54
3.3.5 <i>Quản lý hồ sơ du học sinh viên</i>	56
3.3.6 <i>Quản lý gửi phản hồi</i>	57
3.3.7 <i>Duyệt hồ sơ du học của Admin</i>	58
3.3.8 <i>Quản lý khóa du học của admin</i>	59
3.3.9 <i>Quản lý phản hồi của admin</i>	61
3.3.10 <i>Quản lý Thêm thông tin sinh viên của Admin</i>	62
3.3.11 <i>Quản lý xem thông tin sinh viên</i>	63
3.3.12 <i>Quản lý thông tin Admin</i>	65
3.4 <i>Biểu đồ hoạt động</i>	67
3.4.1 <i>Biểu đồ hoạt động đăng nhập</i>	67
3.4.2 <i>Biểu đồ hoạt động đăng xuất</i>	67
3.4.3 <i>Biểu đồ hoạt động thêm mới hồ sơ</i>	68
3.4.5 <i>Biểu đồ hoạt động sửa hồ sơ</i>	68
3.4.6 <i>Biểu đồ hoạt động đăng kí khóa du học</i>	69
3.4.7 <i>Biểu đồ hoạt động gửi đánh giá phản hồi</i>	69
3.4.8 <i>Biểu đồ hoạt động thêm sinh viên</i>	70
3.4.9 <i>Biểu đồ hoạt động xem chi tiết thông tin sinh viên</i>	71
3.4.10 <i>Biểu đồ hoạt động sửa thông tin sinh viên</i>	72

3.4.11 Biểu đồ hoạt động thêm khóa học	72
3.4.12 Biểu đồ hoạt động sửa thông tin khóa học	73
3.5 Biểu đồ tuần tự	73
3.5.1 Biểu đồ tuần tự đăng nhập	73
3.5.2 Biểu đồ tuần tự quản lý khóa du học	74
3.5.3 Biểu đồ tuần tự quản lý phản hồi đánh giá	74
3.5.4 Biểu đồ tuần tự quản lý sinh viên	75
3.5.5 Biểu đồ tuần tự quản lý thông tin admin	75
3.5.6 Biểu đồ tuần tự duyệt hồ sơ du học	76
3.6 Thiết kế giao diện.....	76
3.6.1 Giao diện đăng nhập	76
3.6.2 Giao diện trang chủ của Sinh viên	77
3.6.3 Giao diện chi tiết khóa du học.....	78
3.6.4 Giao diện quản lý hồ sơ dinh viên.....	79
3.6.5 Giao diện trang chủ Admin	80
3.6.6 Giao diện quản lý sinh viên	81
3.6.7 Giao diện quản lý khoá du học.....	82
3.6.8 Giao diện duyệt hồ sơ.....	83
3.6.9 Giao diện quản lý hồ sơ cá nhân admin.....	84
3.6.10 Giao diện quản lý hồ sơ thông tin sinh viên	85
3.6.11 Giao diện duyệt hồ sơ sinh viên.....	85
CHƯƠNG 4: KẾT QUẢ VÀ KIỂM THỬ.....	86
4.1 Kiểm thử đăng nhập	86
4.2 Kiểm thử chức năng khác	86
KẾT LUẬN	90
TÀI LIỆU THAM KHẢO	91

DANH MỤC HÌNH ẢNH

Hình 1. Hoạt động của mô hình MVC	23
Hình 2. Sơ đồ ERD.....	45
Hình 3. Use case tổng quát.....	51
Hình 4. Use case quản lý thông tin sinh viên	52
Hình 5. Use case quản lý thông tin gia đình sinh viên	53
Hình 6. Use case quản lý khóa du học sinh viên.....	54
Hình 7. Use case quản lý sơ đồ du học sinh viên.....	56
Hình 8. Use case quản lý gửi phản hồi.....	57
Hình 9. Use case duyệt hồ sơ du học admin	58
Hình 10. Use case quản lý khóa du học admin	59
Hình 11. Use case quản lý phản hồi admin	61
Hình 12. Use case quản lý thông tin sinh viên của admin	62
Hình 13. Use case quản lý xem thông tin sinh viên	63
Hình 14. Use case quản lý thông tin admin.....	65
Hình 15. Biểu đồ hoạt động đăng nhập	67
Hình 16. Biểu đồ hoạt động đăng xuất.....	67
Hình 17. Biểu đồ hoạt động thêm mới hồ sơ	68
Hình 18. Biểu đồ hoạt động sửa hồ sơ	68
Hình 19. Biểu đồ hoạt động đăng kí khóa du học	69
Hình 20. Biểu đồ hoạt động gửi đánh giá phản hồi.....	69
Hình 21. Biểu đồ hoạt động thêm sinh viên	70
Hình 22. Biểu đồ hoạt động xem chi tiết thông tin sinh viên.....	71
Hình 23. Biểu đồ hoạt động sửa thông tin sinh viên	72
Hình 24. Biểu đồ hoạt động thêm khóa học	72
Hình 25. Biểu đồ hoạt động thông tin khóa học.....	73

Hình 26. Biểu đồ tuần tự đăng nhập.....	73
Hình 27. Biểu đồ tuần tự quản lý khóa du học.....	74
Hình 28. Biểu đồ tuần tự phản hồi đánh giá.....	74
Hình 29. Biểu đồ tuần tự quản lý sinh viên.....	75
Hình 30. Biểu đồ tuần tự quản lý thông tin sinh viên admin	75
Hình 31. Biểu đồ tuần tự duyệt hồ sơ du học	76
Hình 32. Giao diện đăng nhập.....	76
Hình 33. Giao diện trang chủ của sinh viên	77
Hình 34. Giao diện chi tiết khóa du học.....	78
Hình 35. Giao diện quản lý hồ sơ sinh viên	79
Hình 36. Giao diện trang chủ admin	80
Hình 37. Giao diện quản lý sinh viên.....	81
Hình 38. Giao diện quản lý khóa học.....	82
Hình 39. Giao diện duyệt hồ sơ.....	83
Hình 40. Giao diện quản lý hồ sơ cá nhân admin	84
Hình 41. Giao diện quản lý hồ sơ thông tin sinh viên.....	85
Hình 42. Giao diện duyệt hồ sơ sinh viên	85

DANH MỤC BẢNG

Bảng 1. Bảng so sánh sự khác nhau giữa ASP.NET và ASP.NET Core	18
Bảng 2. Bảng mô tả use case quản lý thông tin sinh viên	52
Bảng 3. Bảng đặc tả use case quản lý thông tin gia đình sinh viên	53
Bảng 4. Bảng đặc tả use case quản lý khóa du học sinh viên	55
Bảng 5. Bảng đặc tả use case quản lý sơ đồ du học sinh viên	56
Bảng 6. Bảng đặc tả Use case quản lý gửi phản hồi	57
Bảng 7. Bảng đặc tả use case duyệt hồ sơ du học admin	58
Bảng 8. Bảng đặc tả use case quản lý khóa du học admin	59
Bảng 9. Bảng đặc tả use case quản lý phản hồi admin.....	61
Bảng 10. Bảng đặc tả use case quản lý thông tin sinh viên của admin	62
Bảng 11. Bảng đặc tả use case quản lý xem thông tin sinh viên.....	63
Bảng 12. Bảng đặc tả use case quản lý thông tin admin	66

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1. Giới thiệu về dự án và mục tiêu nghiên cứu

Dự án "Xây dựng ứng dụng website thực hiện đăng ký, hồ sơ sinh viên du học nước ngoài của Viện Đào tạo Quốc tế - Đại học Đại Nam" nhằm mục đích tối ưu hóa quá trình đăng ký và quản lý hồ sơ của sinh viên quốc tế. Với sứ mệnh giảm bớt thủ tục phức tạp và tạo điều kiện thuận lợi cho sinh viên trong quá trình nộp đơn du học, dự án đặt ra các mục tiêu chính là tối ưu hóa quy trình đăng ký thông qua một giao diện thân thiện và dễ sử dụng, đồng thời đảm bảo tính bảo mật cao cho thông tin cá nhân của sinh viên. Ngoài ra, hệ thống cũng được thiết kế để tăng cường hiệu suất quản lý hồ sơ, giúp nhân viên dễ dàng theo dõi và xử lý các yêu cầu của sinh viên một cách hiệu quả. Với những nỗ lực này, dự án hướng đến mục tiêu mang lại trải nghiệm tốt nhất cho cả sinh viên và nhân viên trong quá trình đăng ký và quản lý hồ sơ du học.

1.2. Lý do chọn đề tài và tầm quan trọng của đề tài

1.2.1. Lý do chọn đề tài

Trong bối cảnh toàn cầu hóa giáo dục, việc du học đã không còn là một xu hướng xa lạ mà đã trở thành một lựa chọn phổ biến đối với sinh viên trên khắp thế giới. Điều này đặt ra nhu cầu cấp thiết trong việc phát triển một hệ thống đăng ký và quản lý hồ sơ hiệu quả, nhằm đáp ứng nhu cầu ngày càng tăng của sinh viên quốc tế. Sự thuận tiện và tốc độ trong quá trình đăng ký không chỉ giúp viện đào tạo nâng cao lợi thế cạnh tranh mà còn góp phần thu hút và giữ chân sinh viên quốc tế, qua đó củng cố vị thế của trường trong lĩnh vực giáo dục quốc tế.

Hơn nữa, việc tự động hóa các quy trình đăng ký và quản lý hồ sơ không chỉ giảm bớt sự phụ thuộc vào công việc thủ công mà còn tăng cường hiệu suất làm việc, đồng thời giúp tiết kiệm chi phí hoạt động. Điều này không chỉ mang lại lợi ích cho viện đào tạo mà còn tạo điều kiện thuận lợi cho sinh viên, giúp họ tiếp cận các cơ hội giáo dục một cách dễ dàng và nhanh chóng hơn. Đây chính là bước tiến quan trọng trong việc hiện đại hóa hệ thống giáo dục, đáp ứng nhu cầu của thời đại số và tạo điều kiện cho sự phát triển không ngừng của cộng đồng học thuật quốc tế.

1.2.2. Tầm quan trọng của đề tài

Đề tài này có tầm quan trọng lớn trong việc nâng cao chất lượng giáo dục và trải nghiệm của sinh viên. Bằng việc cung cấp một giao diện người dùng dễ sử dụng và thuận tiện, hệ thống không chỉ giúp sinh viên có được những trải nghiệm tích cực trong quá trình đăng ký và quản lý hồ sơ du học mà còn góp phần nâng cao hình ảnh và uy tín của trường. Sự thuận lợi và dễ dàng truy cập này là yếu tố quan trọng trong việc tạo ra một môi trường học tập chất lượng cao, thu hút sinh viên từ khắp nơi trên thế giới.

Hệ thống cũng tăng cường tính hiệu quả và minh bạch trong quản lý, cho phép nhân viên dễ dàng theo dõi và kiểm soát các quy trình đăng ký và hồ sơ. Điều này tạo ra một môi trường làm việc minh bạch, nơi mọi quy trình đều được thực hiện một cách rõ ràng và công bằng, đảm bảo rằng mọi sinh viên đều có cơ hội bình đẳng trong việc tiếp cận giáo dục.

Cuối cùng, việc triển khai các biện pháp bảo mật thông tin hiện đại là một phần không thể thiếu, giúp đảm bảo an toàn cho thông tin cá nhân của sinh viên. Sự an toàn này không chỉ bảo vệ sinh viên khỏi các nguy cơ tiềm ẩn trên mạng mà còn xây dựng niềm tin và lòng tin từ phía sinh viên và phụ huynh, là nền tảng vững chắc cho một môi trường giáo dục an toàn và thân thiện. Đây là những yếu tố cốt lõi giúp định hình một hệ thống giáo dục hiện đại, đáp ứng nhu cầu của thế hệ sinh viên mới, đồng thời đảm bảo sự phát triển bền vững của trường trong tương lai.

1.3. Phạm vi nghiên cứu

Dự án xây dựng website cho việc đăng ký và quản lý hồ sơ sinh viên du học nước ngoài tại Viện Đào Tạo Quốc Tế Đại Học Đại Nam đặt ra một phạm vi nghiên cứu rộng lớn, bao gồm việc phát triển một hệ thống quản lý thông tin sinh viên toàn diện. Hệ thống này sẽ cho phép quản trị viên thêm, sửa đổi và xóa thông tin chi tiết về sinh viên, từ thông tin cá nhân đến học vấn và liên hệ, đảm bảo rằng dữ liệu luôn được cập nhật và chính xác.

Tính năng tải lên hình ảnh sẽ được tích hợp để xác thực danh tính của sinh viên, tăng cường bảo mật và độ tin cậy của hệ thống. Sinh viên sẽ có khả năng tải lên các tài liệu quan trọng như bản sao bằng cấp và các tài liệu khác, cần thiết cho việc xét duyệt và xử lý hồ sơ của họ. Hệ thống cũng sẽ cung cấp chức năng xem và tải về các tài liệu đã nạp lên, giúp sinh viên dễ dàng quản lý và theo dõi hồ sơ của mình.

Về mặt giao diện người dùng, website sẽ được thiết kế để đảm bảo dễ sử dụng và thân thiện, mang lại trải nghiệm thuận tiện cho sinh viên và cung cấp các công cụ quản lý hiệu quả cho giảng viên. Đặc biệt, hệ thống sẽ được thiết kế với tính linh hoạt và khả năng mở rộng, cho phép nó thích ứng với sự phát triển của trường và tích hợp các tính năng mới, cũng như mở rộng cơ sở dữ liệu khi cần thiết, đảm bảo khả năng phục vụ lâu dài và ổn định cho viện đào tạo.

1.4. Phương pháp nghiên cứu:

- Tìm hiểu tài liệu chính thống của Framework, sau đó cài đặt môi trường phát triển trên để thực hành viết mã và hiểu rõ hơn về mỗi công nghệ.
- Nghiên cứu các dự án đã triển khai sử dụng các công nghệ này để hiểu cách triển khai và duy trì ứng dụng, đồng thời tham gia các diễn đàn và nhóm người dùng để học hỏi từ kinh nghiệm của người khác.
- Áp dụng kiến thức từ nghiên cứu để xây dựng các ứng dụng mẫu hoặc dự án thử nghiệm, sau đó đánh giá hiệu suất và tính ổn định của ứng dụng dựa trên các thử nghiệm thực tế.

CHƯƠNG 2: KIẾN TRÚC HỆ THỐNG VÀ CÔNG NGHỆ SỬ DỤNG

2.1. Công nghệ sử dụng

2.1.1. Giới thiệu tổng quan về HTML

HTML, viết tắt của HyperText Markup Language, là ngôn ngữ đánh dấu chuẩn để tạo và thiết kế các trang web. Nó được sử dụng để mô tả cấu trúc của thông tin trên một trang web và nó có khả năng kết hợp với các ngôn ngữ như CSS (Cascading Style Sheets) và JavaScript để tạo ra các trang web động và phong phú.

Dưới đây là một số điểm cơ bản về HTML:

- Cấu trúc cơ bản: Một tài liệu HTML bao gồm các thẻ (tags) như `<html>`, `<head>`, và `<body>` để định nghĩa cấu trúc cơ bản của trang web.
- Thẻ và thuộc tính: Các thẻ như `<p>`, `<h1>`, `<a>`, và `` được sử dụng để tạo ra văn bản, tiêu đề, liên kết, và hình ảnh. Mỗi thẻ có thể có các thuộc tính (attributes) như `class`, `id`, và `style` để định nghĩa thêm thông tin và kiểu dáng.
- HTML5: Là phiên bản mới nhất của HTML, nó bổ sung thêm nhiều thẻ và API mới để hỗ trợ tốt hơn cho các ứng dụng web động và đa phương tiện.
- Tương thích: HTML được thiết kế để hoạt động trên nhiều trình duyệt và thiết bị khác nhau, từ máy tính để bàn đến điện thoại di động.

2.1.2. Giới thiệu tổng quan về CSS

CSS, viết tắt của Cascading Style Sheets, là ngôn ngữ được sử dụng để định dạng và bố trí trang web. Nó hoạt động cùng với HTML để tạo ra ngoại hình và cảm giác cho website. Dưới đây là một số thông tin cơ bản về CSS:

- Selectors: Đây là cách chúng ta chọn các phần tử HTML để áp dụng các quy tắc CSS. Có thể là tên thẻ (như `div`, `h2`), class (bắt đầu bằng dấu chấm như `.class-name`), hoặc ID (bắt đầu bằng dấu # như `#id-name`).
- Properties và Values: Mỗi selector có thể có nhiều thuộc tính (properties), mỗi thuộc tính có một giá trị (value) tương ứng. Ví dụ, `color: red;` là một thuộc tính và giá trị để định nghĩa màu sắc cho văn bản.

- Box Model: Mọi phần tử trong CSS đều được xem như một hộp, và “Box Model” mô tả cách các hộp này tương tác với nhau. Nó bao gồm margin, border, padding, và content area.
- Layout: CSS cung cấp các công cụ như Flexbox và Grid để sắp xếp các phần tử trong một layout linh hoạt và đáp ứng.
- Responsive Design: CSS cho phép tạo ra các trang web đáp ứng, tức là trang web có thể thích ứng với kích thước màn hình khác nhau, từ máy tính để bàn đến điện thoại di động.

2.1.3. Giới thiệu tổng quan về JavaScript

JavaScript là một ngôn ngữ lập trình đa năng và linh hoạt, rất quan trọng trong việc phát triển web. Dưới đây là một số thông tin cơ bản về JavaScript:

- Định nghĩa: JavaScript là một ngôn ngữ lập trình phía client, cho phép tạo ra các trang web tương tác và động.
- Lịch sử: Được phát triển bởi Brendan Eich vào năm 1995, JavaScript đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới.
- Tính năng: JavaScript có thể tương tác với HTML và CSS, thêm các hiệu ứng chuyển động, hộp thoại thông báo, và xử lý dữ liệu từ máy chủ.
- Nền tảng: JavaScript chạy trên nhiều nền tảng khác nhau, từ các trình duyệt web như Chrome và Firefox đến máy chủ thông qua Node.js.
- Các khái niệm cơ bản: Bao gồm biến, kiểu dữ liệu, hàm, và cấu trúc điều khiển dòng chảy của chương trình.

2.1.4. Giới thiệu tổng quan về ASP.NET Core framework

ASP.NET Core là một phiên bản mới của ASP.NET, được phát hành bởi Microsoft và là mã nguồn mở trên GitHub. ASP.NET Core được sử dụng để phát triển các khuôn khổ website và có thể thích ứng với nhiều trình duyệt khác nhau như Windows, Mac, hoặc Linux, kể cả trên nền tảng MVC. Ban đầu, phiên bản này có tên là ASP.NET 5 nhưng sau đó được đổi tên thành ASP.NET Core. Được thiết kế để tối ưu hóa framework phát triển cho các ứng dụng chạy on-premise hay triển khai trên đám mây, ASP.NET Core bao gồm các thành phần theo hướng module nhằm tối thiểu hóa tài nguyên và tiết kiệm chi phí phát

triển. Bên cạnh đó, ASP.NET Core cũng là mã nguồn mở, phù hợp với xu thế của các ngôn ngữ lập trình hiện nay.

ASP.NET Core mang lại nhiều lợi ích vượt trội cho việc phát triển web. Với ASP.NET Core MVC, người dùng có thể xây dựng giao diện web (UI) và API web, cho phép tạo ra các ứng dụng web có thể thực hiện testing theo mô hình MVC. Nó hỗ trợ xây dựng các dịch vụ HTTP với nhiều định dạng và cung cấp đầy đủ hỗ trợ cho nội dung của dữ liệu trả về. Razor, ngôn ngữ tạo Views, mang lại hiệu quả cao, trong khi Tag Helper cho phép code phía server tham gia vào quá trình tạo và render các phần tử HTML. Model Binding tự động ánh xạ dữ liệu từ HTTP request đến các tham số của phương thức hành động, và Model Validation tự động thực hiện xác thực trên cả client và server. Về phát triển phía client, ASP.NET Core được thiết kế để tích hợp liên tục với nhiều frameworks như AngularJS, Bootstrap và KnockoutJS. Bên cạnh đó, ASP.NET Core còn được đánh giá cao bởi những lợi ích khác như khung mẫu gọn gàng hơn kiến trúc nhiều tầng, là mã nguồn mở, dễ dàng tạo ứng dụng đa nền tảng trên Windows, Mac và Linux, cấu hình sẵn sàng cho đám mây, và khả năng lưu trữ trên nhiều nền tảng như IIS, HTTP.sys, Kestrel, Nginx, Apache và Docker.

Dưới đây là bảng so sánh sự khác nhau giữa ASP.NET và ASP.NET Core:

Bảng 1. Bảng so sánh sự khác nhau giữa ASP.NET và ASP.NET Core

Đặc điểm	ASP.NET	ASP.NET Core
Mã nguồn	Đóng	Mở, có sẵn trên GitHub
Nền tảng	Chỉ chạy trên Windows	Chạy đa nền tảng: Windows, Mac, Linux
Cấu trúc	Truyền thống với Web Forms hoặc MVC	Hiện đại, được thiết kế lại hoàn toàn

Hiệu năng	Tốt nhưng không tối ưu cho cloud	Tối ưu hóa cho các ứng dụng cloud và microservices
Hỗ trợ đám mây	Hạn chế	Tối ưu hóa và sẵn sàng cho triển khai trên đám mây
Module hóa	Không rõ ràng	Thiết kế theo hướng module, tiết kiệm tài nguyên
Tích hợp phía client	Hạn chế, thường cần thêm công cụ ngoài	Tích hợp tốt với các framework như AngularJS, Bootstrap, KnockoutJS
Khả năng lưu trữ	IIS	Nhiều tùy chọn: IIS, HTTP.sys, Kestrel, Nginx, Apache, Docker
Triển khai	Phụ thuộc vào hệ điều hành Windows	Linh hoạt, có thể triển khai trên nhiều môi trường
Khả năng phát triển	Tập trung vào Windows	Hỗ trợ phát triển đa nền tảng
Cộng đồng và hỗ trợ	Lớn, lâu đời	Đang phát triển, nhưng tăng trưởng nhanh
Công nghệ phụ thuộc	.NET Framework	.NET Core, không phụ thuộc vào hệ điều hành
Mô hình phát triển	Web Forms, MVC	MVC, Razor Pages, API, Blazor

2.1.5. Lý do sử dụng các công nghệ trên

Việc lựa chọn HTML, CSS, JavaScript và ASP.NET Core để xây dựng website cho đồ án tốt nghiệp của tôi dựa trên những lý do sau:

HTML (HyperText Markup Language):

- Cấu trúc trang web: HTML là ngôn ngữ tiêu chuẩn để xây dựng cấu trúc của các trang web. Nó cho phép tạo và định dạng các phần tử như tiêu đề, đoạn văn, hình ảnh, liên kết, và biểu mẫu, tạo nên khung cơ bản cho trang web.
- Tiêu chuẩn web: HTML được chấp nhận rộng rãi và hỗ trợ bởi tất cả các trình duyệt web, đảm bảo tính tương thích và khả năng truy cập cao cho mọi người dùng.

CSS (Cascading Style Sheets):

- Thiết kế và giao diện: CSS được sử dụng để định dạng và bố trí các phần tử HTML. Nó cho phép tùy chỉnh màu sắc, phông chữ, khoảng cách, và bố cục, tạo ra giao diện người dùng hấp dẫn và dễ sử dụng.
- Tách biệt nội dung và trình bày: CSS giúp tách biệt phần nội dung (HTML) khỏi phần trình bày, làm cho mã nguồn sạch sẽ và dễ dàng bảo trì.

JavaScript:

- Tương tác và động: JavaScript là ngôn ngữ lập trình phía client, cho phép tạo ra các tính năng tương tác như xác thực biểu mẫu, tạo hiệu ứng động và thao tác DOM, cải thiện trải nghiệm người dùng và làm cho trang web trở nên sống động và phản hồi nhanh.
- Frameworks và Libraries: JavaScript có nhiều thư viện và frameworks như jQuery, React, và AngularJS, giúp phát triển các ứng dụng web phức tạp và hiệu quả hơn.

ASP.NET Core:

- Phát triển web hiện đại: ASP.NET Core là một framework mạnh mẽ và hiện đại để xây dựng các ứng dụng web và API. Nó hỗ trợ các mô hình như MVC (Model-View-Controller), Razor Pages, và nhiều mô hình khác, giúp phát triển các ứng dụng web linh hoạt và hiệu quả.
- Đa nền tảng: ASP.NET Core có thể chạy trên Windows, Mac và Linux, cho phép

triển khai ứng dụng trên nhiều môi trường khác nhau.

- Hiệu năng cao: ASP.NET Core được tối ưu hóa để cung cấp hiệu năng cao, giúp ứng dụng chạy nhanh và đáp ứng tốt hơn.
- Mã nguồn mở: ASP.NET Core là mã nguồn mở, cho phép kiểm tra, sửa đổi và tùy chỉnh mã nguồn để phù hợp với nhu cầu cụ thể. Nó cũng có một cộng đồng lớn hỗ trợ, giúp giải quyết các vấn đề nhanh chóng.
- Tích hợp tốt với các công nghệ hiện đại: ASP.NET Core dễ dàng tích hợp với các công nghệ và công cụ hiện đại như Docker, Kubernetes, và các dịch vụ đám mây (Azure, AWS), hỗ trợ xây dựng và triển khai ứng dụng web theo tiêu chuẩn công nghiệp hiện đại.

Kết hợp các công nghệ này, tôi có thể xây dựng một trang web mạnh mẽ, linh hoạt, tương tác cao và dễ dàng bảo trì cũng như mở rộng trong tương lai, đáp ứng tốt các yêu cầu của đồ án tốt nghiệp.

2.2. Kiến trúc của hệ thống

2.2.1. Giới thiệu về kiến trúc MVC

Mẫu kiến trúc Model - View - Controller (MVC) được sử dụng để chia ứng dụng thành ba thành phần chính: model, view và controller. Nền tảng ASP.NET MVC cho phép chúng ta tạo các ứng dụng web áp dụng mô hình MVC thay vì theo mẫu ASP.NET Web Forms. Đặc điểm nổi bật của nền tảng ASP.NET MVC là nhẹ (lightweight), dễ kiểm thử phần giao diện (so với ứng dụng Web Forms) và tích hợp các tính năng có sẵn của ASP.NET. Nền tảng này được định nghĩa trong namespace System.Web.Mvc và là một phần của namespace System.Web. MVC là một mẫu thiết kế (design pattern) chuẩn mà nhiều lập trình viên đã quen thuộc. Một số loại ứng dụng web sẽ thích hợp với kiến trúc MVC, trong khi một số khác vẫn phù hợp với ASP.NET Web Forms và cơ chế postbacks. Đôi khi, có những ứng dụng kết hợp cả hai kiến trúc trên.

Models:

- Các đối tượng Models là một phần của ứng dụng, các đối tượng này thiết lập logic của phần dữ liệu của ứng dụng. Thông thường, các đối tượng model lấy và lưu trạng thái của model trong CSDL. Ví dụ như, một đối tượng Product (sản phẩm) sẽ lấy dữ liệu

từ CSDL, thao tác trên dữ liệu và sẽ cập nhật dữ liệu trở lại vào bảng Products ở SQL Server.

- Trong các ứng dụng nhỏ, model thường là chỉ là một khái niệm nhằm phân biệt hơn là được cài đặt thực thụ, ví dụ, nếu ứng dụng chỉ đọc dữ liệu từ CSDL và gửi chúng đến view, ứng dụng không cần phải có tầng model và các lớp liên quan. Trong trường hợp này, dữ liệu được lấy như là một đối tượng model (hơn là tầng model)

Views:

Là các thành phần dùng để hiển thị giao diện người dùng (UI). Thông thường, view được tạo dựa vào thông tin dữ liệu model. Ví dụ như, view dùng để cập nhật bảng Products sẽ hiển thị các hộp văn bản, drop-down list, và các check box dựa trên trạng thái hiện tại của một đối tượng Product.

Controllers:

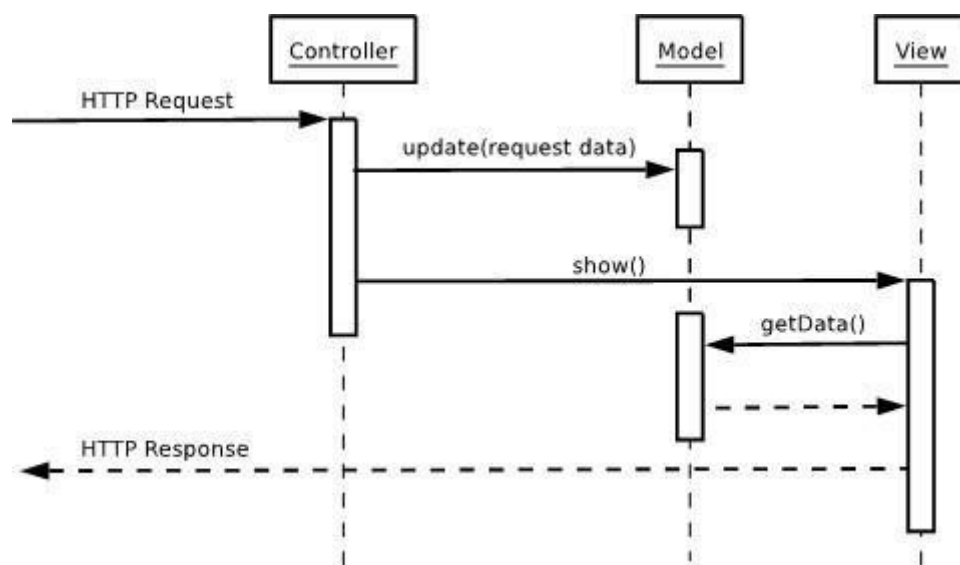
Controller là các thành phần dùng để quản lý tương tác người dùng, làm việc với model và chọn view để hiển thị giao diện người dùng. Trong một ứng dụng MVC, view chỉ được dùng để hiển thị thông tin, controller chịu trách nhiệm quản lý và đáp trả nội dung người dùng nhập và tương tác với người dùng. Ví dụ, controller sẽ quản lý các dữ liệu người dùng gửi lên (query-string values) và gửi các giá trị đó đến model, model sẽ lấy dữ liệu từ CSDL nhờ vào các giá trị này.

Mẫu MVC giúp bạn tạo được các ứng dụng mà chúng phân tách rạch ròi các khía cạnh của ứng dụng (logic về nhập liệu, logic xử lý tác vụ và logic về giao diện). Mẫu MVC chỉ ra mỗi loại logic kể trên nên được thiết lập ở đâu trên ứng dụng. Logic giao diện (UI logic) thuộc về views. Logic nhập liệu (input logic) thuộc về controller. Và logic tác vụ (Business logic – là logic xử lý thông tin, mục đích chính của ứng dụng) thuộc về model. Sự phân chia này giúp bạn giảm bớt được sự phức tạp của ứng dụng và chỉ tập trung vào mỗi khía cạnh cần được cài đặt ở mỗi thời điểm. Ví dụ như bạn chỉ cần tập trung vào giao diện (views) mà không phải quan tâm đến logic xử lý thông tin của ứng dụng.

Để quản lý sự phức tạp của ứng dụng, mẫu MVC giúp cho chúng ta có thể kiểm thử ứng dụng dễ dàng hơn hẳn so với khi áp dụng mẫu Web Forms. Ví dụ, trong một ứng dụng ASP.NET Web Forms, một lớp thường được sử dụng để hiển thị thông tin xuất ra cho

người dùng và đồng thời xử lý thông tin người dùng nhập. Việc xây dựng các bộ test tự động cho ứng dụng Web Forms là rất phức tạp, bởi để kiểm thử mỗi trang web, bạn phải khởi tạo đối tượng trang, khởi tạo tất cả các control được sử dụng trong trang và các lớp phụ thuộc trong ứng dụng. Và bởi vì có quá nhiều lớp cần được khởi tạo để chạy được trang, thật khó để có thể viết các test chỉ tập trung vào một khía cạnh nào đó của ứng dụng. Và vì thế, kiểm thử đối với các ứng dụng dựa trên nền tảng Web Forms sẽ khó khăn hơn nhiều so với khi áp dụng trên ứng dụng MVC. Hơn thế nữa, việc kiểm thử trên nền tảng Web Forms yêu cầu phải sử dụng đến web server. Nền tảng MVC phân tách các thành phần và sử dụng các interface (khái niệm giao diện trong lập trình hướng đối tượng), và nhờ đó có thể kiểm thử các thành phần riêng biệt trong tình trạng phân lập với các yếu tố còn lại của ứng dụng

Sự phân tách rạch ròi ba thành phần của ứng dụng MVC còn giúp cho việc lập trình diễn ra song song. Ví dụ như một lập trình viên làm việc với view, lập trình viên thứ hai lo cài đặt logic của controller và lập trình viên thứ ba có thể tập trung vào logic tác vụ của model tại cùng một thời điểm..



Hình 1. Hoạt động của mô hình MVC

Lấy ví dụ một GUI Component (thành phần đồ họa người dùng) đơn giản là checkbox. Checkbox có thành phần Model để quản lý trạng thái của nó là check hay uncheck, thành phần View để thể hiện nó với trạng thái tương ứng lên màn hình, và thành

phần Controller để xử lý nhưng sự kiện khi có sự tương tác của người sử dụng hoặc các đối tượng khác lên checkbox.

Khi người sử dụng nhấn chuột vào Check box , thành phần Controller của Checkbox sẽ xử lý sự kiện này, yêu cầu thành phần Model thay đổi dữ liệu trạng thái. Sau khi thay đổi trạng thái, thành phần Model phát thông điệp đến thành phần View và Controller. Thành phần View của Checkbox nhận được thông điệp sẽ cập nhật lại thể hiện của Checkbox, phản ánh chính xác trạng thái Checkbox do Model lưu giữ. Thành phần Controller nhận được thông điệp do Model gửi tới sẽ có những tương tác phản hồi với người sử dụng nếu cần thiết.

Tìm hiểu thêm về Controller:

Controller có trách nhiệm chính là điều hướng các yêu cầu của người sử dụng. Như vậy trên toàn ứng dụng của ta, tất cả các request đều sẽ phải đi tới controller. Và tại đây, ứng với các tham số người sử dụng truyền mà ta đưa họ đến một tác vụ nào đó trên ứng dụng. Tại các tác vụ này, chúng sẽ thông qua lớp model để làm việc và trả kết quả trở về controller. Cuối cùng controller sẽ đẩy dữ liệu thao tác tới view. View là thành phần cuối cùng mà người sử dụng nhận được khi họ gửi request tới ứng dụng. Có thể hiểu controller, giống với kỹ thuật đa cấp với các tác vụ chạy ứng dụng phân cấp theo từng nhánh riêng biệt như: Module, action, ... Qua hình vẽ này, ta hiểu rằng. Để có thể thao tác với các action (hành động) ta cần phải đi qua file index.php. Lúc này file index đóng vai trò như một controller được dùng để điều hướng các request. Ứng với các request thì nó sẽ trả về một controller khác để xử lý tác vụ một cách cụ thể. Tại controller con, nó sẽ gọi các action riêng biệt. Ở đó, thông qua các action mà nó gọi tới các file xử lý riêng đơn.

Tìm hiểu thêm về Model:

- Model là thành phần chủ yếu được sử dụng để thao tác xử lý dữ liệu. Trong các framework, Model vẫn thường sử dụng theo phương thức Active Record. Một trong những design pattern. Chúng có tác dụng rút ngắn thời gian viết câu truy vấn cho người sử dụng. Biến những câu truy vấn phức tạp trở nên gọn gàng và thân thiện với người sử dụng thông qua các thư viện được định nghĩa sẵn.

- Model thường sẽ là các phương thức có trách nhiệm xử lý các tác vụ như: select, insert, update, delete các record trong database. Ứng với các lấy dữ liệu, model thường sử dụng mảng để gửi trả kết quả về. Vì mảng có thể cho phép model lưu trữ nhiều thông tin hơn, nên thường các record khi bóc tách chúng sẽ mang các dữ liệu của database một cách chi tiết.

- Khi sử dụng models, ta cũng cần tuân theo nguyên tắc chính của chúng là không xuất giá trị trực tiếp trong model. Mà tất cả những dữ liệu ấy, phải đưa vào mảng và trả về theo phương thức. Và tiếp tục ở view ta sẽ sử dụng nó để lấy dữ liệu ra.

Tìm hiểu thêm về View:

View là phần hiển thị thông tin tương phản khi gọi và nhận request. Trước đây, khi người lập trình chưa nghĩ tới view. Họ thường thao tác xử lý dữ liệu ngay trực tiếp trên ứng dụng và đổ cả dữ liệu ngay trên file PHP đó. Điều này làm cho ứng dụng trở nên cồng kềnh, và đặc biệt rất khó cho việc bảo trì nâng cấp sau này. Nhất là đối với designer, việc thay đổi giao diện của một website luôn làm cho họ cảm thấy đau đầu vì phải vọc thẳng vào core. Trước đây, để giải quyết tình huống này. Người ta thường sử dụng template để phân tách website thành 2 mảng riêng biệt. Một là giao diện và một là core. Việc chỉnh sửa giao diện trở nên đơn giản hơn đối với họ so với cách viết thập cẩm kia. Tuy nhiên, các thư viện này thực chất sẽ làm cho ứng dụng của chúng ta trở nên chậm chạp hơn bao giờ hết. Bởi chúng phải phiên dịch nhiều lần các kịch bản. Chẳng hạn: Để dễ thao tác, smarty sẽ dịch ngược các yêu cầu của bạn sang ngôn ngữ của nó. Sau đó chúng sẽ chuyển ngôn ngữ đó sang PHP và thao tác xử lý trên nó. Việc này sẽ làm ứng dụng chậm chạp, do cứ phải dịch qua, dịch lại một kịch bản. Trong khi, với sự kết hợp của PHP thuần, ứng dụng của bạn sẽ nhanh và ổn định hơn nhiều. Và view cũng là một phần trong việc nâng cấp những hạn chế ấy. Chúng giúp giảm thiểu tối đa quá trình biên dịch nhiều lần. Và làm cho ứng dụng trở nên mạnh mẽ và chuyên nghiệp hơn nhiều so với cách lập trình thuần.

2.2.2. Lý do sử dụng kiến trúc MVC

Kiến trúc Model-View-Controller (MVC) mang lại nhiều lợi ích quan trọng cho việc phát triển ứng dụng web, và dưới đây là những lý do chính để sử dụng kiến trúc này:

Tách biệt giữa các thành phần:

- Model: Đại diện cho dữ liệu và logic nghiệp vụ của ứng dụng.
- View: Chịu trách nhiệm hiển thị dữ liệu và giao diện người dùng.
- Controller: Xử lý các yêu cầu từ người dùng, tương tác với Model và chọn View phù hợp để hiển thị.
- Sự tách biệt này giúp mã nguồn dễ hiểu hơn, dễ bảo trì và nâng cấp, đồng thời cho phép các nhà phát triển làm việc trên các thành phần khác nhau một cách độc lập.

Dễ kiểm thử:

- Việc tách biệt rõ ràng giữa Model, View và Controller giúp cho việc kiểm thử từng phần trở nên dễ dàng hơn. Đặc biệt, phần giao diện (View) có thể được kiểm thử một cách độc lập so với các phần khác, nâng cao chất lượng và độ tin cậy của ứng dụng.

Tính linh hoạt:

- MVC cho phép các nhà phát triển linh hoạt trong việc phát triển và bảo trì ứng dụng. Khi có thay đổi trong logic nghiệp vụ hoặc giao diện người dùng, chỉ cần thay đổi trong Model hoặc View tương ứng mà không ảnh hưởng đến các thành phần khác, giúp tiết kiệm thời gian và công sức.

Khả năng mở rộng:

- Kiến trúc MVC hỗ trợ việc mở rộng ứng dụng một cách dễ dàng. Các tính năng mới có thể được thêm vào mà không cần phải sửa đổi toàn bộ ứng dụng, giúp ứng dụng phát triển một cách bền vững và lâu dài.

Tích hợp với các công nghệ hiện đại:

- ASP.NET MVC tích hợp các tính năng có sẵn của ASP.NET và dễ dàng tương thích với các công nghệ hiện đại như AJAX, jQuery và các framework JavaScript khác. Điều này giúp phát triển các ứng dụng web hiện đại, tương tác cao và thân thiện với người dùng.

Được cộng đồng chấp nhận rộng rãi:

- MVC là một mẫu thiết kế chuẩn mà nhiều lập trình viên đã quen thuộc, do đó, việc

tìm kiếm tài liệu hướng dẫn, hỗ trợ cộng đồng và các tài nguyên liên quan rất dễ dàng. Sự phổ biến này giúp đẩy nhanh quá trình học tập và áp dụng kiến trúc MVC trong các dự án thực tế.

Khả năng kết hợp:

- Một số loại ứng dụng web sẽ thích hợp với kiến trúc MVC, trong khi một số khác vẫn phù hợp với ASP.NET Web Forms và cơ chế postbacks. Đôi khi, có những ứng dụng kết hợp cả hai kiến trúc trên, tận dụng ưu điểm của cả hai mô hình, đáp ứng đa dạng nhu cầu phát triển ứng dụng.

Những lý do trên làm cho kiến trúc MVC trở thành một lựa chọn lý tưởng cho việc phát triển các ứng dụng web hiện đại, giúp tăng cường hiệu quả phát triển, bảo trì và mở rộng ứng dụng.

2.3. Giới thiệu về hệ quản trị cơ sở dữ liệu của hệ thống

2.3.1. Giới thiệu về hệ quản trị cơ sở dữ liệu SQL Server

SQL Server là một phần mềm do Microsoft phát triển, đang trở thành một trong những mối quan tâm hàng đầu khi nói về công nghệ thông tin. SQL Server, viết tắt của Microsoft SQL Server hoặc MS SQL Server, được thiết kế để quản lý và lưu trữ dữ liệu theo mô hình quan hệ RDBMS (Relational Database Management System).

SQL, viết tắt của Structured Query Language, là ngôn ngữ truy vấn cấu trúc dữ liệu. Đây là một loại ngôn ngữ lập trình được sử dụng để quản lý và truy vấn cơ sở dữ liệu quan hệ. SQL cho phép bạn thực hiện các hoạt động như lưu trữ, cập nhật, xóa, tìm kiếm, truy xuất dữ liệu, quản lý và tối ưu hóa hiệu suất của cơ sở dữ liệu.

Sự nhầm lẫn giữa hai khái niệm SQL Server và ngôn ngữ SQL là phổ biến. Đơn giản, SQL là một loại ngôn ngữ lập trình, trong khi SQL Server là phần mềm của Microsoft được tạo nên bởi loại ngôn ngữ này. SQL Server hỗ trợ nhiều loại ngôn ngữ lập trình SQL khác nhau, từ ANSI SQL (SQL tiêu chuẩn) đến SQL thông thường và cả T-SQL (Transaction-SQL) - được sử dụng chủ yếu cho các hệ thống cơ sở dữ liệu quan hệ cao cấp.

2.3.2. Thành phần chính của SQL Server

SQL Server bao gồm hai thành phần chính là Database Engine và SQLOS (SQL Server Operating System).

Database Engine: Database Engine là trung tâm của SQL Server, nơi mà tất cả các hoạt động quản lý và xử lý dữ liệu diễn ra. Được chia thành hai phần quan trọng:

- **Relation Engine:** Đây là thành phần chịu trách nhiệm xử lý các câu truy vấn SQL từ phía người dùng và ứng dụng. Relation Engine thực hiện các hoạt động truy xuất dữ liệu (SELECT), cập nhật dữ liệu (INSERT, UPDATE, DELETE), và các thao tác khác như định nghĩa và quản lý đối tượng cơ sở dữ liệu như bảng, view, procedure, function, và trigger.
- **Storage Engine:** Storage Engine quản lý cách lưu trữ và truy cập dữ liệu trong SQL Server. Nó bao gồm các tệp dữ liệu (data files), trang dữ liệu (data pages), chỉ mục (indexes), và các phần khác của cơ sở dữ liệu. Storage Engine đảm bảo rằng dữ liệu được lưu trữ và truy xuất một cách hiệu quả và an toàn.

SQLOS (SQL Server Operating System): SQLOS là hệ điều hành được tối ưu hóa cho SQL Server, chịu trách nhiệm quản lý tài nguyên hệ thống như bộ nhớ, lập lịch thực hiện nhiệm vụ, quản lý luồng dữ liệu và xử lý các yêu cầu đồng thời. SQLOS đảm bảo tính toàn vẹn của dữ liệu bằng cách quản lý các khóa dữ liệu và đồng bộ hóa truy cập đến dữ liệu từ các tiến trình và luồng thực thi khác nhau.

SQL Server không chỉ đơn thuần là một hệ quản trị cơ sở dữ liệu, mà còn cung cấp một loạt các tính năng mở rộng và công cụ bổ sung như SQL Server Reporting Services (SSRS) để tạo báo cáo, SQL Server Integration Services (SSIS) để tích hợp dữ liệu, và SQL Server Analysis Services (SSAS) để phân tích dữ liệu. Tính linh hoạt và mở rộng của SQL Server cho phép người dùng tổ chức và quản lý dữ liệu một cách hiệu quả và đáp ứng nhu cầu kinh doanh ngày càng phức tạp.

2.3.2. Giới thiệu về SSMS

SQL Server Management Studio (SSMS) là một công cụ quản lý cơ sở dữ liệu mạnh mẽ do Microsoft phát triển. Được thiết kế đặc biệt cho quản trị viên cơ sở dữ liệu và các

nhà phát triển ứng dụng, SSMS cung cấp một loạt các tính năng và công cụ để quản lý và phát triển cơ sở dữ liệu SQL Server.

Các tính năng chính của SQL Server Management Studio (SSMS) bao gồm:

- Quản lý đối tượng: SSMS cho phép người dùng tạo, sửa đổi và xóa các đối tượng cơ sở dữ liệu như bảng, view, stored procedure, function và trigger. Người dùng có thể quản lý các đối tượng này thông qua giao diện đồ họa hoặc dòng lệnh SQL.
- Truy vấn và Biên tập dữ liệu: SSMS cung cấp môi trường để thực thi các câu truy vấn SQL đến cơ sở dữ liệu. Người dùng có thể thực hiện các thao tác truy xuất dữ liệu, cập nhật, xóa và biên tập dữ liệu một cách dễ dàng từ giao diện của SSMS.
- Bảo mật và Phân quyền: SSMS cho phép người quản trị viên thiết lập và quản lý an toàn và phân quyền truy cập vào cơ sở dữ liệu. Người dùng có thể điều chỉnh các quyền truy cập, quản lý vai trò người dùng và thiết lập các chính sách bảo mật để bảo vệ dữ liệu.
- Giám sát và Tối ưu hóa: SSMS cung cấp công cụ để giám sát hoạt động của cơ sở dữ liệu, bao gồm thống kê về hiệu suất và sử dụng tài nguyên. Người dùng có thể tối ưu hóa truy vấn SQL và cấu trúc cơ sở dữ liệu để cải thiện hiệu suất hệ thống.
- Thiết kế và Quản lý bảo trì: SSMS hỗ trợ người dùng trong việc thiết kế và quản lý cấu trúc cơ sở dữ liệu như bảng, chỉ mục và ràng buộc. Nó cung cấp các công cụ để sao lưu, phục hồi và đồng bộ hóa dữ liệu để đảm bảo tính toàn vẹn và khả năng sẵn sàng của dữ liệu.
- Phát triển ứng dụng và tích hợp dữ liệu: SSMS hỗ trợ lập trình viên trong việc phát triển ứng dụng và tích hợp dữ liệu thông qua SQL Server Integration Services (SSIS). Nó cung cấp các công cụ để thiết kế, quản lý và triển khai các gói tích hợp dữ liệu để chuyển đổi và đồng bộ dữ liệu từ các nguồn khác nhau vào SQL Server.

SSMS là một công cụ không thể thiếu đối với bất kỳ ai làm việc với SQL Server, bao gồm quản trị viên cơ sở dữ liệu, nhà phát triển ứng dụng và các chuyên gia dữ liệu. Với giao diện thân thiện và tính năng đa dạng, SSMS giúp người dùng quản lý và tối ưu hóa cơ sở dữ liệu một cách hiệu quả và dễ dàng.

2.4. Triển khai ứng dụng

2.4.1. Chuẩn bị Môi trường

Để bắt đầu phát triển ứng dụng với ASP.NET và SQL Server, tôi cần chuẩn bị môi trường làm việc bằng cách thực hiện các bước sau:

- Cài đặt Visual Studio: Đầu tiên, tôi sẽ tải và cài đặt Visual Studio. Visual Studio là một môi trường phát triển tích hợp (IDE) mạnh mẽ của Microsoft, hỗ trợ phát triển các ứng dụng desktop, web và di động trên nền tảng Windows. Tôi sẽ chọn phiên bản phù hợp với nhu cầu và hệ điều hành của tôi để bắt đầu phát triển ứng dụng ASP.NET.
- Cài đặt SQL Server: Sau đó, tôi sẽ cài đặt SQL Server hoặc sử dụng SQL Server Management Studio (SSMS) để quản lý cơ sở dữ liệu. SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ mạnh mẽ do Microsoft phát triển, hỗ trợ lưu trữ và quản lý dữ liệu hiệu quả. SSMS là công cụ đồ họa và dòng lệnh để quản lý và tương tác với cơ sở dữ liệu SQL Server một cách thuận tiện và hiệu quả.

Chuẩn bị môi trường này sẽ giúp tôi có thể phát triển ứng dụng ASP.NET và liên kết với cơ sở dữ liệu SQL Server một cách dễ dàng và hiệu quả.

2.4.2. Tạo dự án ASP.NET MVC

Để bắt đầu phát triển dự án ASP.NET MVC, tôi sẽ làm theo các bước sau:

- Mở Visual Studio và tạo dự án mới: Tôi khởi động Visual Studio và chọn "File" > "New" > "Project" để bắt đầu tạo dự án mới.
- Chọn mẫu dự án ASP.NET MVC: Trong hộp thoại "New Project", tôi chọn mẫu dự án "ASP.NET Web Application". Tôi đảm bảo rằng tôi đã chọn môi trường làm việc phù hợp (ví dụ: .NET Framework hoặc .NET Core) và chọn mẫu "ASP.NET Core Web Application" nếu tôi đang làm việc với .NET Core.
- Thiết lập tên dự án, đường dẫn và các cài đặt khác: Tôi đặt tên cho dự án của mình và thiết lập đường dẫn lưu trữ dự án trên máy tính của mình. Tôi cũng có thể lựa chọn các cài đặt khác như framework target (đích đến), phiên bản .NET, và các mô-đun bổ sung như Docker, Azure và mô hình cấu trúc dự án.

Việc tạo dự án ASP.NET MVC sẽ tạo ra cấu trúc cơ bản cho một ứng dụng web

MVC, bao gồm các thư mục và tệp tin cần thiết để bắt đầu phát triển và triển khai ứng dụng web trên nền tảng ASP.NET.

2.4.3. Thiết kế Cơ sở Dữ liệu:

- Sử dụng SQL Server hoặc SSMS để thiết kế cơ sở dữ liệu: Tôi sẽ sử dụng SQL Server hoặc SQL Server Management Studio (SSMS) để thiết kế và quản lý cơ sở dữ liệu của mình. SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ mạnh mẽ của Microsoft, còn SSMS là công cụ đồ họa và dòng lệnh để tương tác với cơ sở dữ liệu SQL Server.
- Tạo các bảng cho sinh viên, thông tin gia đình, điểm số, v.v.: Tôi sẽ bắt đầu bằng việc thiết kế các bảng cơ sở dữ liệu cho các đối tượng như sinh viên, thông tin gia đình, điểm số, và các thông tin khác phù hợp với yêu cầu của ứng dụng của tôi.
- Thiết lập các mối quan hệ giữa các bảng: Sau khi tạo các bảng, tôi sẽ thiết lập các mối quan hệ giữa chúng để đảm bảo tính toàn vẹn dữ liệu và tối ưu hóa cơ sở dữ liệu. Các mối quan hệ này có thể bao gồm quan hệ một-nhiều, nhiều-nhiều và các ràng buộc khóa ngoại để liên kết các bảng với nhau.

Việc thiết kế cơ sở dữ liệu này sẽ cung cấp cho tôi nền tảng vững chắc để lưu trữ và quản lý dữ liệu cho ứng dụng ASP.NET MVC của tôi.

2.4.4. Triển khai Mô hình MVC:

Để triển khai mô hình MVC trong dự án của mình, tôi sẽ thực hiện các bước sau:

- Tạo các Model: Tôi sẽ định nghĩa các lớp Model để ánh xạ dữ liệu từ cơ sở dữ liệu. Các lớp Model này sẽ đại diện cho các đối tượng trong cơ sở dữ liệu như sinh viên, thông tin gia đình, điểm số, v.v.
- Tạo các Controller: Tôi sẽ xây dựng các Controller để xử lý các yêu cầu từ người dùng và tương tác với cơ sở dữ liệu. Controller sẽ chịu trách nhiệm điều phối dữ liệu giữa Model và View, xử lý logic nghiệp vụ và trả về kết quả cho người dùng.
- Tạo các View: Tôi sẽ thiết kế các giao diện người dùng bằng HTML, CSS và JavaScript. Tôi sử dụng Razor syntax để kết hợp dữ liệu từ Controller vào các View. Điều này giúp tạo ra các trang web tương tác mà người dùng có thể truy cập và sử dụng.

Triển khai mô hình MVC sẽ giúp tôi phân chia rõ ràng các phần của ứng dụng, từ việc lưu trữ dữ liệu, xử lý logic đến hiển thị giao diện người dùng, đảm bảo tính linh hoạt và bảo trì dễ dàng cho dự án ASP.NET MVC của tôi.

2.4.5. Kết nối với Cơ sở Dữ liệu:

Để kết nối ứng dụng ASP.NET MVC của tôi với cơ sở dữ liệu SQL Server, tôi sẽ thực hiện các bước sau:

- Thiết lập kết nối đến cơ sở dữ liệu SQL Server từ ứng dụng ASP.NET MVC: Tôi sẽ định cấu hình kết nối đến cơ sở dữ liệu SQL Server từ tệp cấu hình của ứng dụng. Điều này bao gồm cung cấp các thông tin như tên máy chủ, tên cơ sở dữ liệu, tên người dùng và mật khẩu.
- Sử dụng ADO.NET hoặc Entity Framework để thực hiện các thao tác CRUD: Tôi có thể sử dụng ADO.NET để viết các truy vấn SQL trực tiếp và thực hiện các thao tác CRUD như tạo (Create), đọc (Read), cập nhật (Update) và xóa (Delete) dữ liệu từ cơ sở dữ liệu. Hoặc tôi cũng có thể sử dụng Entity Framework, một ORM (Object-Relational Mapping) framework của Microsoft, để tạo và quản lý các đối tượng dữ liệu (Model) và thực hiện các thao tác CRUD thông qua các phương thức của Entity Framework.

Việc kết nối và thao tác với cơ sở dữ liệu là một phần quan trọng trong việc phát triển ứng dụng web ASP.NET MVC, giúp tôi quản lý và tương tác dữ liệu một cách hiệu quả và an toàn.

2.4.6. Kiểm tra và Triển khai:

Để hoàn thành dự án ASP.NET MVC của tôi, tôi sẽ thực hiện các bước sau:

- Kiểm tra ứng dụng trên môi trường phát triển: Tôi sẽ kiểm tra ứng dụng trên môi trường phát triển để đảm bảo rằng tất cả các chức năng hoạt động đúng. Tôi sẽ kiểm tra từng tính năng, xử lý ngoại lệ và đảm bảo giao diện người dùng hoạt động một cách nhất quán trên các trình duyệt khác nhau.
- Triển khai ứng dụng lên môi trường sản phẩm: Sau khi hoàn thành kiểm tra, tôi sẽ triển khai ứng dụng lên môi trường sản phẩm. Tôi sẽ cấu hình Internet Information

Services (IIS) để chạy ứng dụng trên máy chủ web. Điều này bao gồm thiết lập các tham số cần thiết trong IIS như Pool Ứng dụng, Ứng dụng, và các cài đặt khác để đảm bảo ứng dụng hoạt động ổn định và an toàn trên môi trường sản phẩm.

Việc kiểm tra và triển khai là các bước cuối cùng nhưng vô cùng quan trọng trong quá trình phát triển ứng dụng, giúp đảm bảo rằng sản phẩm được cung cấp với chất lượng cao và đáp ứng được các yêu cầu của người dùng.

CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

3.1. Đặc tả

3.1.1. Sinh viên

Sinh viên là một trong những nhân vật chính trong hệ thống. Đặc tả cho sinh viên nhằm xác định rõ các chức năng, yêu cầu và luồng công việc liên quan đến quản lý thông tin cá nhân và các hoạt động liên quan của họ trong hệ thống.

a. Chức năng: Xem Thông tin Cá nhân của Sinh viên

Yêu cầu chức năng:

- Hệ thống phải xác thực danh tính sinh viên trước khi cho phép truy cập thông tin cá nhân.
- Hiện thị các thông tin cơ bản của sinh viên.

Luồng công việc:

- Sinh viên đăng nhập vào hệ thống.
- Sinh viên chọn mục "Xem Thông tin Cá nhân".
- Hệ thống hiển thị các thông tin cá nhân của sinh viên.

b. Chức năng: Sửa Thông tin Cá nhân của Sinh viên

Yêu cầu chức năng:

- Hệ thống phải xác thực danh tính sinh viên trước khi cho phép chỉnh sửa thông tin cá nhân.
- Sinh viên chỉ được phép chỉnh sửa một số thông tin nhất định.
- Gửi thông báo qua email đến sinh viên sau khi thông tin được cập nhật thành công.

Luồng công việc:

- Sinh viên đăng nhập vào hệ thống.
- Sinh viên chọn mục "Sửa Thông tin Cá nhân".
- Hệ thống hiển thị form cho phép chỉnh sửa các thông tin được phép.
- Sinh viên nhập thông tin mới và xác nhận thay đổi.

- Hệ thống kiểm tra và xác thực thông tin mới.
- Hệ thống cập nhật thông tin vào cơ sở dữ liệu và gửi thông báo qua email cho sinh viên.

c. Chức năng: Xem Thông tin Gia đình của Sinh viên

Yêu cầu chức năng:

- Hệ thống phải xác thực danh tính sinh viên trước khi cho phép truy cập thông tin gia đình.
- Hiện thị các thông tin cơ bản về gia đình của sinh viên.

Luồng công việc:

- Sinh viên đăng nhập vào hệ thống.
- Sinh viên chọn mục "Thông tin Gia đình".
- Hệ thống hiển thị các thông tin về gia đình của sinh viên.

d. Chức năng: Sửa Thông tin Gia đình của Sinh viên

Yêu cầu chức năng:

- Hệ thống phải xác thực danh tính sinh viên trước khi cho phép chỉnh sửa thông tin gia đình.
- Sinh viên chỉ được phép chỉnh sửa một số thông tin nhất định.
- Gửi thông báo qua email đến sinh viên sau khi thông tin được cập nhật thành công.

Luồng công việc:

- Sinh viên đăng nhập vào hệ thống.
- Sinh viên chọn mục "Sửa Thông tin Gia đình".
- Hệ thống hiển thị form cho phép chỉnh sửa các thông tin được phép.
- Sinh viên nhập thông tin mới và xác nhận thay đổi.
- Hệ thống kiểm tra và xác thực thông tin mới.
- Hệ thống cập nhật thông tin vào cơ sở dữ liệu và gửi thông báo qua email cho sinh viên.

e. Chức năng: Thêm mới Hồ sơ Du học của Sinh viên

Yêu cầu chức năng:

- Hệ thống phải xác thực danh tính sinh viên trước khi cho phép thêm mới hồ sơ du học.
- Sinh viên có thể nhập thông tin và tải lên các tài liệu cần thiết để tạo hồ sơ du học mới.

Luồng công việc:

- Sinh viên đăng nhập vào hệ thống.
- Sinh viên chọn mục "Thêm mới Hồ sơ Du học".
- Hệ thống hiển thị form cho phép nhập thông tin và tải lên các tài liệu cần thiết.
- Sinh viên tải lên các tài liệu.
- Sinh viên xác nhận và gửi hồ sơ.
- Hệ thống lưu trữ hồ sơ mới vào cơ sở dữ liệu.

f. Chức năng: Hiển thị Hồ sơ đã Nộp của Sinh viên

Yêu cầu chức năng:

- Hệ thống phải xác thực danh tính sinh viên trước khi cho phép xem các hồ sơ đã nộp.
- Hiển thị danh sách các hồ sơ du học mà sinh viên đã gửi.

Luồng công việc:

- Sinh viên đăng nhập vào hệ thống.
- Sinh viên chọn mục "Hiển thị Hồ sơ đã Nộp".
- Hệ thống hiển thị danh sách các hồ sơ đã gửi, bao gồm thông tin chi tiết của từng hồ sơ.

g. Chức năng: Hiển thị Danh sách Đăng kí Du học của Sinh viên

Yêu cầu chức năng:

- Hệ thống phải xác thực danh tính của người dùng trước khi cho phép xem danh sách đăng kí du học.
- Hiện thị danh sách các đăng kí du học đã gửi, kèm theo trạng thái của mỗi đăng kí.

Luồng công việc:

- Sinh viên đăng nhập vào hệ thống.
- Sinh viên chọn mục "Danh sách Đăng kí Du học".
- Hệ thống hiện thị danh sách các đăng kí du học đã gửi.
- Người dùng có thể xem chi tiết từng đăng kí để biết thêm thông tin về trạng thái của đăng kí.

h. Chức năng: Đăng ký Khóa Du học của Sinh viên

Yêu cầu chức năng:

- Hệ thống phải xác thực danh tính của người dùng trước khi cho phép đăng ký khóa du học.
- Người dùng có thể đăng ký cho một hoặc nhiều khóa học có sẵn trong danh sách.

Luồng công việc:

- Người dùng đăng nhập vào hệ thống.
- Người dùng duyệt qua danh sách khóa du học và chọn đăng ký.
- Người dùng chọn mục "Đăng ký".
- Hệ thống ghi nhận việc đăng ký của người dùng cho khóa du học đã chọn.
- Hệ thống gửi thông tin du học cho admin duyệt và hiện thị trạng thái duyệt ở danh sách đăng kí.

i. Chức năng: Gửi Phản hồi cho Admin

Yêu cầu chức năng:

- Hệ thống cần cung cấp tính năng cho người dùng gửi phản hồi và góp ý cho admin.

- Phản hồi cần được gửi đến admin để admin có thể xem phản hồi của sinh viên.

Luồng công việc:

- Người dùng đăng nhập vào hệ thống (nếu cần).
- Người dùng truy cập vào phần "Liên hệ" trên giao diện của ứng dụng.
- Người dùng nhập nội dung phản hồi vào một biểu mẫu hoặc ô văn bản.
- Sau khi hoàn thành, người dùng chọn nút "Gửi".
- Hệ thống ghi nhận phản hồi và gửi nó đến hộp thư hoặc hệ thống quản trị của admin.
- Admin nhận được thông báo về phản hồi mới và có thể xem nó thông qua giao diện quản trị.

3.1.2 Admin

a. Chức năng Xem Phản hồi của Sinh viên

Yêu cầu chức năng:

- Admin cần có khả năng xem phản hồi và góp ý được gửi từ sinh viên.

Luồng công việc:

- Admin đăng nhập vào hệ thống quản trị.
- Admin truy cập vào phần "Inbox".
- Hệ thống hiển thị danh sách các phản hồi từ sinh viên, bao gồm nội dung phản hồi, thông tin liên hệ của sinh viên, và thời gian gửi.

b. Chức năng Quản lý Sinh viên

Chức năng Thêm Sinh viên:

Yêu cầu chức năng:

- Admin có khả năng thêm mới thông tin sinh viên vào hệ thống.
- Form nhập thông tin cần bao gồm các trường cơ bản như: Họ và tên, Mã số sinh viên, Ngày sinh, Địa chỉ email, Số điện thoại, Địa chỉ nhà, Chương trình học, v.v.

Luồng công việc:

- Admin đăng nhập vào hệ thống quản trị.
- Admin truy cập vào mục "Student" và chọn "Thêm Sinh viên".
- Hệ thống hiển thị form nhập thông tin sinh viên mới.
- Admin nhập đầy đủ thông tin cần thiết và xác nhận.
- Hệ thống lưu trữ thông tin sinh viên mới vào cơ sở dữ liệu.

Chức năng Sửa Thông tin Sinh viên

Yêu cầu chức năng:

- Admin có khả năng chỉnh sửa thông tin của sinh viên đã tồn tại trong hệ thống.

Luồng công việc:

- Admin đăng nhập vào hệ thống quản trị.
- Admin truy cập vào mục "Student".
- Hệ thống hiển thị danh sách sinh viên hiện có.
- Admin chọn sinh viên cần chỉnh sửa và nhấp vào "Sửa".
- Hệ thống hiển thị form chỉnh sửa thông tin sinh viên với các trường thông tin hiện có.
- Admin thực hiện các thay đổi cần thiết và xác nhận.
- Hệ thống cập nhật thông tin sinh viên trong cơ sở dữ liệu.

Chức năng Xóa Sinh viên

Yêu cầu chức năng:

- Admin có khả năng xóa thông tin sinh viên khỏi hệ thống.

Luồng công việc:

- Admin đăng nhập vào hệ thống quản trị.
- Admin truy cập vào mục "Student".
- Hệ thống hiển thị danh sách sinh viên hiện có.
- Admin chọn sinh viên cần xóa và nhấp vào "Xóa".
- Hệ thống yêu cầu xác nhận việc xóa sinh viên.

- Admin xác nhận xóa sinh viên.
- Hệ thống xóa thông tin sinh viên khỏi cơ sở dữ liệu.

Chức năng Tìm kiếm và Hiện thị Danh sách Sinh viên

Yêu cầu chức năng:

- Admin có khả năng tìm kiếm sinh viên dựa trên các tiêu chí khác nhau như tên, mã số sinh viên, v.v.
- Hiện thị danh sách sinh viên phù hợp với các tiêu chí tìm kiếm.

Luồng công việc:

- Admin đăng nhập vào hệ thống quản trị.
- Admin truy cập vào mục "Student".
- Hệ thống hiển thị thanh tìm kiếm và các tiêu chí lọc.
- Admin nhập các tiêu chí tìm kiếm và nhấp vào "Tìm kiếm".
- Hệ thống hiển thị danh sách sinh viên phù hợp với các tiêu chí tìm kiếm.

Chức năng Hiện thị Chi tiết Thông tin Sinh viên

Yêu cầu chức năng:

- Admin có khả năng xem chi tiết thông tin của từng sinh viên.

Luồng công việc:

- Admin đăng nhập vào hệ thống quản trị.
- Admin truy cập vào mục "Student".
- Hệ thống hiển thị danh sách sinh viên hiện có.
- Admin chọn sinh viên muốn xem chi tiết và nhấp vào "Xem chi tiết".
- Hệ thống hiển thị thông tin chi tiết của sinh viên bao gồm các thông tin cá nhân, học tập, liên lạc, và các thông tin liên quan khác.

c. Chức năng Quản lý Khóa Du học

Chức năng Thêm Khóa Du học

Yêu cầu chức năng:

- Admin có khả năng thêm mới thông tin khóa du học vào hệ thống.

- Form nhập thông tin cần bao gồm các trường như: Tên khóa du học, thời gian du học, giá thành, v.v.

Luồng công việc:

- Admin đăng nhập vào hệ thống quản trị.
- Admin truy cập vào mục "Course" và chọn "Thêm Khóa du học".
- Hệ thống hiển thị form nhập thông tin khóa du học mới.
- Admin nhập đầy đủ thông tin cần thiết và xác nhận.
- Hệ thống lưu trữ thông tin khóa du học mới vào cơ sở dữ liệu.

Chức năng Sửa Thông tin Khóa Du học

Yêu cầu chức năng:

- Admin có khả năng chỉnh sửa thông tin của các khóa du học đã tồn tại trong hệ thống.

Luồng công việc:

- Admin đăng nhập vào hệ thống quản trị.
- Admin truy cập vào mục "Course".
- Hệ thống hiển thị danh sách các khóa du học hiện có.
- Admin chọn khóa du học cần chỉnh sửa và nhấp vào "Sửa".
- Hệ thống hiển thị form chỉnh sửa thông tin khóa du học với các trường thông tin hiện có.
- Admin thực hiện các thay đổi cần thiết và xác nhận.
- Hệ thống cập nhật thông tin khóa du học trong cơ sở dữ liệu.

Chức năng Xóa Khóa Du học

Yêu cầu chức năng:

- Admin có khả năng xóa thông tin khóa du học khỏi hệ thống.

Luồng công việc:

- Admin đăng nhập vào hệ thống quản trị.
- Admin truy cập vào mục "Course".

- Hệ thống hiển thị danh sách các khóa du học hiện có.
- Admin chọn khóa du học cần xóa và nhấp vào "Xóa".
- Hệ thống yêu cầu xác nhận việc xóa khóa du học.
- Admin xác nhận xóa khóa du học.
- Hệ thống xóa thông tin khóa du học khỏi cơ sở dữ liệu

Chức năng Tìm kiếm và Hiển thị Danh sách Khóa Du học

Yêu cầu chức năng:

- Admin có khả năng tìm kiếm khóa du học dựa trên các tiêu chí khác nhau như tên, v.v.
- Hiển thị danh sách các khóa du học phù hợp với các tiêu chí tìm kiếm.

Luồng công việc:

- Admin đăng nhập vào hệ thống quản trị.
- Admin truy cập vào mục "Course".
- Hệ thống hiển thị thanh tìm kiếm và các tiêu chí lọc.
- Admin nhập các tiêu chí tìm kiếm và nhấp vào "Tìm kiếm".
- Hệ thống hiển thị danh sách các khóa du học phù hợp với các tiêu chí tìm kiếm.

Chức năng Hiển thị Chi tiết Thông tin Khóa Du học

Yêu cầu chức năng:

- Admin có khả năng xem chi tiết thông tin của từng khóa du học.

Luồng công việc:

- Admin đăng nhập vào hệ thống quản trị.
- Admin truy cập vào mục "Course".
- Hệ thống hiển thị danh sách các khóa du học hiện có.
- Admin chọn khóa du học muốn xem chi tiết và nhấp vào "Xem chi tiết".
- Hệ thống hiển thị thông tin chi tiết của khóa du học bao gồm các thông tin về tên khóa, mô tả, thời gian, địa điểm, và các thông tin liên quan khác.
- Chức năng quản lý duyệt hồ sơ đăng ký du học

Chức năng Hiển thị Chi tiết Hồ sơ Đăng ký Du học

Yêu cầu chức năng:

- Admin có khả năng xem chi tiết từng hồ sơ đăng ký du học được gửi bởi sinh viên.
- Thông tin chi tiết bao gồm các phần như: Thông tin cá nhân, Thông tin học tập, và các tài liệu đính kèm (bảng điểm, chứng chỉ, v.v.).

Luồng công việc:

- Admin đăng nhập vào hệ thống quản trị.
- Admin truy cập vào mục "Register list".
- Hệ thống hiển thị danh sách các hồ sơ đăng ký du học.
- Admin chọn hồ sơ muốn xem chi tiết và nhấp vào "Xem chi tiết".
- Hệ thống hiển thị thông tin chi tiết của hồ sơ đăng ký, bao gồm các thông tin cá nhân, học tập, và các tài liệu đính kèm.

Chức năng Duyệt Hồ sơ Đăng ký Du học

Yêu cầu chức năng:

- Admin có khả năng duyệt hoặc từ chối hồ sơ đăng ký du học.
- Hệ thống cần cung cấp cơ chế ghi nhận lý do từ chối và gửi thông báo về trạng thái hồ sơ đến sinh viên.

Luồng công việc:

- Admin đăng nhập vào hệ thống quản trị.
- Admin truy cập vào mục "Register list".
- Hệ thống hiển thị danh sách các hồ sơ đăng ký du học.
- Admin chọn hồ sơ muốn duyệt hoặc từ chối và nhấp vào "Xem chi tiết".
- Hệ thống hiển thị thông tin chi tiết của hồ sơ đăng ký.
- Admin nhấp vào nút "Duyệt" hoặc "Từ chối".
- Hệ thống cập nhật trạng thái hồ sơ trong cơ sở dữ liệu và gửi thông báo về trạng thái mới (đã duyệt hoặc bị từ chối) đến sinh viên qua email hoặc hệ thống thông báo.

d. Chức năng Quản lý Hồ sơ của Admin

Chức năng Hiển thị chi tiết hồ sơ admin

Yêu cầu chức năng:

- Admin có khả năng xem chi tiết hồ sơ admin.
- Thông tin chi tiết bao gồm các phần như: Thông tin cá nhân, thông tin liên lạc, học vị.

Luồng công việc:

- Admin đăng nhập vào hệ thống quản trị.
- Admin truy cập vào mục "Personal".
- Hệ thống hiển thị thông tin chi tiết của admin.

Chức năng Sửa thông tin admin

Yêu cầu chức năng:

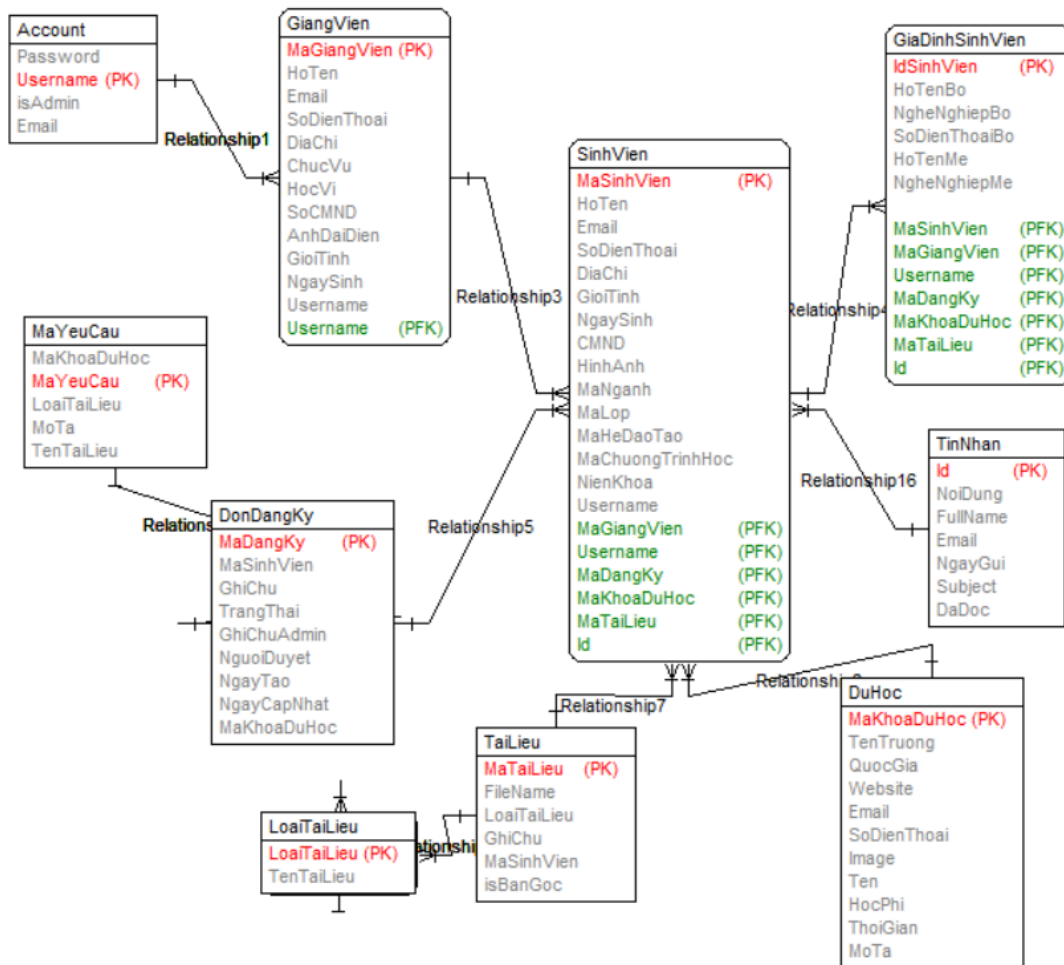
- Admin có khả năng chỉnh sửa thông tin của chính mình trong hệ thống.

Luồng công việc:

- Admin đăng nhập vào hệ thống quản trị.
- Admin truy cập vào mục "Personal".
- Hệ thống hiển thị thông tin của admin.
- Admin điền thông tin cần chỉnh sửa và nhấp vào "Sửa".
- Admin xác nhận và lưu thông tin của mình.
- Hệ thống cập nhật thông tin admin trong cơ sở dữ liệu.

3.2 Thiết kế cơ sở dữ liệu

3.2.1 Sơ đồ ERD



Hình 2. Sơ đồ ERD

3.2.2 Cài đặt cơ sở dữ liệu

Bảng EFMigrationsHistory

```
CREATE TABLE [dbo].[__EFMigrationsHistory](
    [MigrationId] [nvarchar](150) NOT NULL,
    [ProductVersion] [nvarchar](32) NOT NULL,
    CONSTRAINT [PK__EFMigrationsHistory] PRIMARY KEY CLUSTERED
(
```

Bảng Account

```

CREATE TABLE [dbo].[Account](
    [Username] [nvarchar](450) NOT NULL,
    [Password] [nvarchar](max) NOT NULL,
    [isAdmin] [bit] NOT NULL,
    [Email] [nvarchar](max) NOT NULL,
    CONSTRAINT [PK_Account] PRIMARY KEY CLUSTERED
(

```

Bảng DonDangKy

```

CREATE TABLE [dbo].[DonDangKy](
    [MaDangKy] [int] IDENTITY(1,1) NOT NULL,
    [MaSinhVien] [int] NOT NULL,
    [GhiChu] [nvarchar](200) NOT NULL,
    [TrangThai] [int] NOT NULL,
    [GhiChuAdmin] [nvarchar](max) NOT NULL,
    [NguoiDuyet] [nvarchar](500) NOT NULL,
    [NgayTao] [datetime] NOT NULL,
    [NgayCapNhat] [datetime] NOT NULL,
    [MaKhoaDuHoc] [int] NOT NULL,
    CONSTRAINT [PK_DonDangKy] PRIMARY KEY CLUSTERED
(

```

Bảng DuHoc

```

CREATE TABLE [dbo].[DuHoc](
    [MaKhoaDuHoc] [int] IDENTITY(1,1) NOT NULL,
    [TenTruong] [nvarchar](max) NOT NULL,

```

```

[QuocGia] [nvarchar](max) NOT NULL,
[Website] [nvarchar](max) NOT NULL,
[Email] [nvarchar](max) NOT NULL,
[SoDienThoai] [nvarchar](max) NOT NULL,
[Image] [nvarchar](max) NOT NULL,
[Ten] [nvarchar](max) NOT NULL,
[HocPhi] [int] NOT NULL,
[ThoiGian] [nvarchar](max) NOT NULL,
[MoTa] [nvarchar](max) NULL,
CONSTRAINT [PK_DuHoc] PRIMARY KEY CLUSTERED
(

```

Bảng GiaDinhSinhVien

```

CREATE TABLE [dbo].[GiaDinhSinhVien](
    [IdSinhVien] [int] NOT NULL,
    [HoTenBo] [nvarchar](max) NOT NULL,
    [NgheNghiepBo] [nvarchar](max) NOT NULL,
    [SoDienThoaiBo] [nvarchar](max) NOT NULL,
    [HoTenMe] [nvarchar](max) NOT NULL,
    [NgheNghiepMe] [nvarchar](max) NOT NULL,
    [SoDienThoaiMe] [nvarchar](max) NOT NULL,
    CONSTRAINT [PK_GiaDinhSinhVien] PRIMARY KEY CLUSTERED
(

```

Bảng GiangVien

```

CREATE TABLE [dbo].[GiangVien](

```

```

[MaGiangVien] [int] IDENTITY(1,1) NOT NULL,
[HoTen] [nvarchar](max) NOT NULL,
[Email] [nvarchar](max) NOT NULL,
[SoDienThoai] [nvarchar](max) NOT NULL,
[DiaChi] [nvarchar](max) NOT NULL,
[ChucVu] [nvarchar](max) NOT NULL,
[HocVi] [nvarchar](max) NOT NULL,
[SoCMND] [nvarchar](max) NOT NULL,
[AnhDaiDien] [nvarchar](max) NOT NULL,
[GioiTinh] [nvarchar](max) NOT NULL,
[NgaySinh] [datetime2](7) NOT NULL,
[Username] [nvarchar](max) NOT NULL,
CONSTRAINT [PK_GiangVien] PRIMARY KEY CLUSTERED
(

```

Bảng LoạiTaiLieu

```

CREATE TABLE [dbo].[LoạiTaiLieu](
    [LoạiTaiLieu] [nvarchar](50) NOT NULL,
    [TenTaiLieu] [nvarchar](100) NOT NULL,
    CONSTRAINT [PK_LoạiTaiLieu] PRIMARY KEY CLUSTERED
(

```

Bảng SinhVien

```

CREATE TABLE [dbo].[SinhVien](
    [MaSinhVien] [int] NOT NULL,
    [HoTen] [nvarchar](max) NOT NULL,

```



```

[Email] [nvarchar](max) NOT NULL,
[SoDienThoai] [nvarchar](max) NOT NULL,
[DiaChi] [nvarchar](max) NOT NULL,
[GioiTinh] [nvarchar](max) NOT NULL,
[NgaySinh] [datetime2](7) NOT NULL,
[CMND] [nvarchar](max) NOT NULL,
[HinhAnh] [nvarchar](max) NOT NULL,
[MaNganh] [nvarchar](max) NOT NULL,
[MaLop] [int] NOT NULL,
[MaHeDaoTao] [nvarchar](max) NOT NULL,
[MaChuongTrinhHoc] [nvarchar](max) NOT NULL,
[NienKhoa] [int] NOT NULL,
[Username] [nvarchar](max) NOT NULL,
CONSTRAINT [PK_SinhVien] PRIMARY KEY CLUSTERED
(

```

Bảng TaiLieu

```

CREATE TABLE [dbo].[TaiLieu](
    [MaTaiLieu] [int] IDENTITY(1,1) NOT NULL,
    [FileName] [nvarchar](max) NOT NULL,
    [LoaiTaiLieu] [nvarchar](50) NOT NULL,
    [GhiChu] [nvarchar](max) NOT NULL,
    [MaSinhVien] [int] NOT NULL,
    [isBanGoc] [bit] NOT NULL,
    [TenTaiLieu] [nvarchar](100) NULL,

```

CONSTRAINT [PK_TaiLieu] PRIMARY KEY CLUSTERED

(

Bảng TinNhan

CREATE TABLE [dbo].[TinNhan](

[Id] [int] IDENTITY(1,1) NOT NULL,

[NoiDung] [nvarchar](max) NOT NULL,

[FullName] [nvarchar](max) NOT NULL,

[Email] [nvarchar](max) NOT NULL,

[NgayGui] [datetime2](7) NOT NULL,

[Subject] [nvarchar](max) NOT NULL,

[DaDoc] [bit] NOT NULL,

CONSTRAINT [PK_TinNhan] PRIMARY KEY CLUSTERED

(

Bảng YeuCauDuHoc

CREATE TABLE [dbo].[YeuCauDuHoc](

[MaKhoaDuHoc] [int] NOT NULL,

[LoaiTaiLieu] [nvarchar](50) NOT NULL,

[MoTa] [nvarchar](max) NOT NULL,

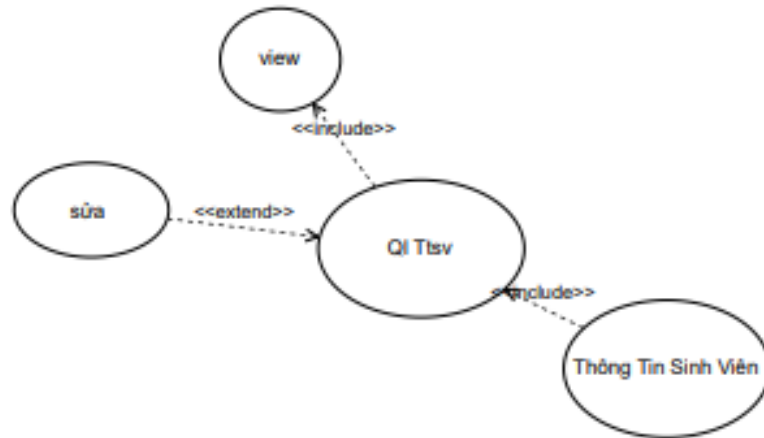
[TenTaiLieu] [nvarchar](100) NOT NULL,

[MaYeuCau] [int] IDENTITY(1,1) NOT NULL,

CONSTRAINT [PK_YeuCauDuHoc] PRIMARY KEY CLUSTERED

(

3.3.2 Quản lý Thông tin sinh viên của Sinh Viên



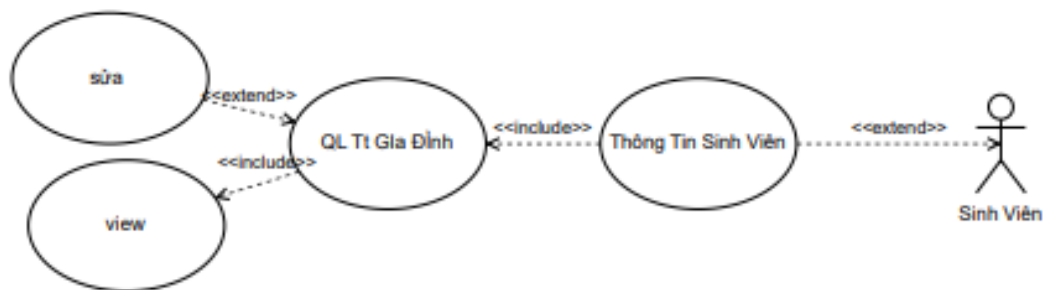
Hình 4. Use case quản lý thông tin sinh viên

Bảng 2. Bảng mô tả use case quản lý thông tin sinh viên

Test Case	Mô Tả Test Case	Kết Quả Mong Đợi
1	Hiện thị thông tin sinh viên của sinh viên	
	1. Mở ứng dụng. 2. Chuyển đến màn hình đăng nhập. 3. Nhập tên người dùng và mật khẩu hợp lệ. 4. Nhấn vào nút đăng nhập. 5. Khi đăng nhập hiện thị thông tin sinh viên.	Người dùng đăng nhập thành công và vào giao diện thông tin sinh viên
2	Sửa thông tin sinh viên	
	1. Sinh viên đăng nhập vào hệ thống 2. Bấm sửa thông tin thì sẽ hiện thị thông tin và	Sinh viên khi bấm lưu thông tin thì thông tin sẽ được

Test Case	Mô Tả Test Case	Kết Quả Mong Đợi
	cho phép sinh viên sửa thông tin của mình. 3. sinh viên thay đổi thông tin cần thay đổi và bấm lưu	thay đổi

3.3.3 Quản lý thông tin gia đình của sinh viên



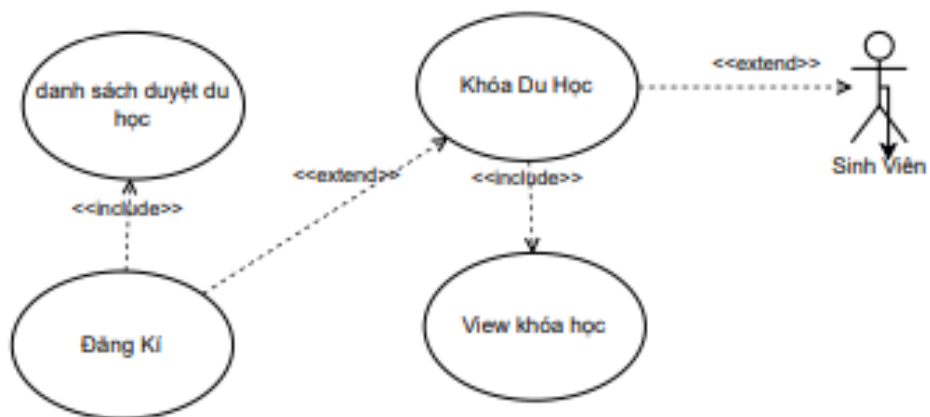
Hình 5. Use case quản lý thông tin gia đình sinh viên

Bảng 3. Bảng đặc tả use case quản lý thông tin gia đình sinh viên

Test Case	Mô Tả Test Case	Kết Quả Mong Đợi
1	Hiện thị thông tin gia đình	
	1. Mở ứng dụng. 2. Chuyển đến màn hình đăng nhập. 3. Nhập tên người dùng và mật khẩu hợp lệ. 4. Nhấn vào nút đăng nhập.	Người dùng đăng nhập thành công và vào giao diện thông tin gia đình

Test Case	Mô Tả Test Case	Kết Quả Mong Đợi
	5. Khi đăng nhập hiển thị thông tin gia đình.	
2	Sửa thông tin gia đình	
	1. Mở ứng dụng. 2. Chuyển đến màn hình đăng nhập. 3. Nhập tên người dùng và mật khẩu hợp lệ. 4. Nhấn vào nút đăng nhập. 5. Khi đăng nhập sẽ hiển thị thông tin gia đình 6. Sinh viên thực hiện thay đổi thông tin cần đổi của gia đình	Sinh viên khi bấm lưu thông tin thì thông tin sẽ được thay đổi

3.3.4 Quản lý khóa du học sinh viên



Hình 6. Use case quản lý khóa du học sinh viên

Bảng 4. Bảng đặc tả use case quản lý khóa du học sinh viên

Test Case	Mô Tả Test Case	Kết Quả Mong Đợi
1	Hiển thị danh sách khóa học	
	<ol style="list-style-type: none"> 1. Mở ứng dụng. 2. Chuyển đến màn hình đăng nhập. 3. Nhập tên người dùng và mật khẩu hợp lệ. 4. Nhấn vào nút đăng nhập. 5. Khi vào giao diện chính ta bấm vào khóa học lập tức sẽ hiển thị danh sách khóa học du học nước ngoài do trường giới thiệu. 	Khi bấm vào khóa học thì sẽ hiển thị khóa du học nước ngoài
2	Đăng kí khóa du học	
	<ol style="list-style-type: none"> 1. Mở ứng dụng. 2. Chuyển đến màn hình đăng nhập. 3. Nhập tên người dùng và mật khẩu hợp lệ. 4. Nhấn vào nút đăng nhập. 5. Khi vào giao diện chính ta bấm vào khóa học lập tức sẽ hiển thị danh sách khóa học du học nước ngoài do trường giới thiệu. 	Khi bấm vào khóa học thì sẽ hiển thị khóa du học nước ngoài

3.3.5 Quản lý hồ sơ du học sinh viên



Hình 7. Use case quản lý sơ đồ du học sinh viên

Bảng 5. Bảng đặc tả use case quản lý sơ đồ du học sinh viên

Test Case	Mô Tả Test Case	Kết Quả Mong Đợi
1	Hiển thị hồ sơ	
	1. Mở ứng dụng. 2. Chuyển đến màn hình đăng nhập. 3. Nhập tên người dùng và mật khẩu hợp lệ. 4. Nhấn vào nút đăng nhập. 5. Khi đăng nhập thành công thì ta sẽ chọn nút hồ sơ và sẽ hiển thị danh sách hồ sơ đã nộp và cần nộp	Hiển thị danh sách đã nộp và cần nộp hồ sơ
2	Thêm hồ sơ	
	1. Mở ứng dụng. 2. Chuyển đến màn hình đăng nhập.	Thêm mới các giấy tờ hồ sơ cần nộp để du học

Test Case	Mô Tả Test Case	Kết Quả Mong Đợi
	<p>3. Nhập tên người dùng và mật khẩu hợp lệ.</p> <p>4. Nhấn vào nút đăng nhập.</p> <p>5. Khi đăng nhập thành công thì ta sẽ chọn nút hồ sơ và sẽ hiển thị danh sách hồ sơ đã nộp và cần nộp</p> <p>6. bấm nút thêm mới thì sẽ hiển thị những giấy tờ cần nộp và ấn lưu lại</p>	

3.3.6 Quản lý gửi phản hồi



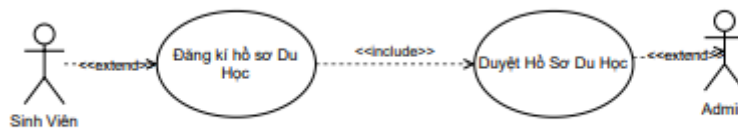
Hình 8. Use case quản lý gửi phản hồi

Bảng 6. Bảng đặc tả Use case quản lý gửi phản hồi

Test Case	Mô Tả Test Case	Kết Quả Mong Đợi
1	Gửi thư phản ánh đánh giá	
	<p>1. Mở ứng dụng.</p> <p>2. Chuyển đến màn hình đăng nhập.</p>	Thực hiện chức năng gửi phản ánh đánh giá cho admin xem

Test Case	Mô Tả Test Case	Kết Quả Mong Đợi
	3. Nhập tên người dùng và mật khẩu hợp lệ. 4. Nhấn vào nút đăng nhập. 5. Sinh viên chọn vào Contact sẽ hiển thị các thư mục để điền và ấn gửi những thông tin này sẽ được admin đọc	

3.3.7 Duyệt hồ sơ du học của Admin



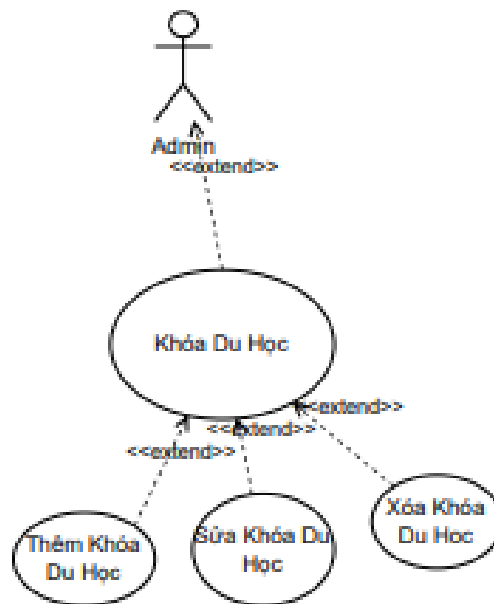
Hình 9. Use case duyệt hồ sơ du học admin

Bảng 7. Bảng đặc tả use case duyệt hồ sơ du học admin

Test Case	Mô Tả Test Case	Kết Quả Mong Đợi
1	Duyệt hồ sơ du học	
	1. Mở ứng dụng. 2. Chuyển đến màn hình đăng nhập. 3. Nhập tên người dùng và mật khẩu hợp lệ. 4. Nhấn vào nút đăng nhập. 5. Khi đăng nhập thành công vào giao diện admin thì chọn register list	Hiện thị danh danh sách hồ sơ cần duyệt

Test Case	Mô Tả Test Case	Kết Quả Mong Đợi
	6. Danh sách hồ sơ du học cần duyệt hiện ra	

3.3.8 Quản lý khóa du học của admin



Hình 10. Use case quản lý khóa du học admin

Bảng 8. Bảng đặc tả use case quản lý khóa du học admin

Test Case	Mô Tả Test Case	Kết Quả Mong Đợi
1	Hiện thị khóa du học	
	1. Mở ứng dụng. 2. Chuyển đến màn hình đăng nhập. 3. Nhập tên người dùng và mật khẩu hợp lệ. 4. Nhấn vào nút đăng nhập.	Hiện thị khóa du học list danh sách khóa đi du học của sinh viên DNU

	5.Admin chọn vào Course và danh sách khóa du học sẽ hiện ra	
2	Thêm khóa học	
	1. Mở ứng dụng. 2. Chuyển đến màn hình đăng nhập. 3. Nhập tên người dùng và mật khẩu hợp lệ. 4. Nhấn vào nút đăng nhập. 5. Admin chọn vào Course và chọn add 6. khi chọn add list 1 các thông tin khóa du học và hồ sơ cần có trong khóa học	Thêm được khóa học và các giấy tờ cần thiết trong khóa học
3	Sửa khóa học	
	1. Mở ứng dụng. 2. Chuyển đến màn hình đăng nhập. 3. Nhập tên người dùng và mật khẩu hợp lệ. 4. Nhấn vào nút đăng nhập. 5. Admin chọn vào Course và chọn vào edit 6. Khi chọn vào edit hiển thị thông tin cần sửa và hồ sơ cần thiết khi đi du học	Sửa được khóa học và các giấy tờ cần thiết trong khóa học
4	Xóa khóa học	
	1. Mở ứng dụng. 2. Chuyển đến màn hình đăng nhập. 3. Nhập tên người dùng và mật khẩu hợp lệ. 4. Nhấn vào nút đăng nhập.	Xóa khóa học thành công

	5. Admin chọn vào Course và chọn vào xóa lập tức khóa du họcj sẽ được xóa đi	
--	--	--

3.3.9 Quản lý phản hồi của admin

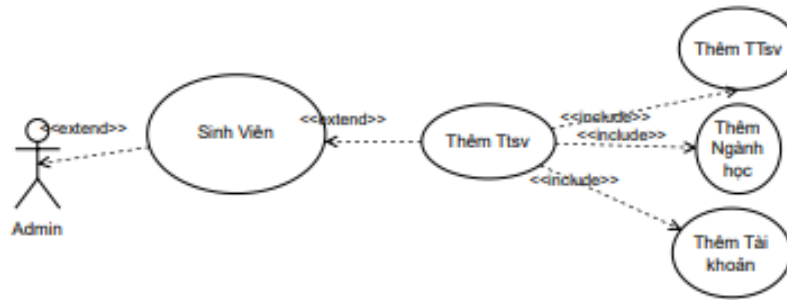


Hình 11. Use case quản lý phản hồi admin

Bảng 9. Bảng đặc tả use case quản lý phản hồi admin

Test Case	Mô Tả Test Case	Kết Quả Mong Đợi
1	Đánh giá của sinh viên gửi về admin	
	<ol style="list-style-type: none"> 1. Mở ứng dụng. 2. Chuyển đến màn hình đăng nhập. 3. Nhập tên người dùng và mật khẩu hợp lệ. 4. Nhấn vào nút đăng nhập. 5. Khi đăng nhập thành công chọn tin nhắn thì sẽ hiện ra những thông tin của sinh viên phản hồi 	Nhận được những đánh giá của sinh viên

3.3.10 Quản lý Thêm thông tin sinh viên của Admin



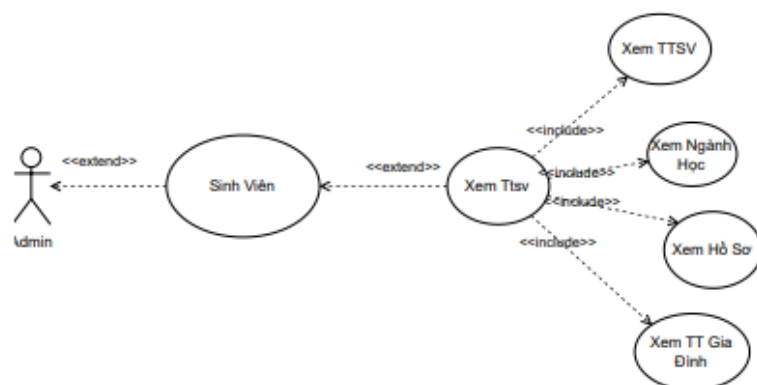
Hình 12. Use case quản lý thông tin sinh viên của admin

Bảng 10. Bảng đặc tả use case quản lý thông tin sinh viên của admin

Test Case	Mô Tả Test Case	Kết Quả Mong Đợi
1	Thêm thông tin sinh viên	
	1. Mở ứng dụng. 2. Chuyển đến màn hình đăng nhập. 3. Nhập tên người dùng và mật khẩu hợp lệ. 4. Nhấn vào nút đăng nhập. 5. Khi đăng nhập thành công ta chọn add student thì hiện thị các ô để ta thêm thông tin sinh viên	Thêm thông tin sinh viên thành công
2	Thêm ngành học	
	1. Mở ứng dụng. 2. Chuyển đến màn hình đăng nhập.	Thêm thông tin ngành học thành công
3	Thêm tài khoản	

Test Case	Mô Tả Test Case	Kết Quả Mong Đợi
	1. Mở ứng dụng. 2. Chuyển đến màn hình đăng nhập. 3. Nhập tên người dùng và mật khẩu hợp lệ. 4. Nhấn vào nút đăng nhập. 5. Khi đăng nhập thành công ta chọn add student thì hiện thị các ô để ta thêm thông tin tài khoản	Thêm thông tin tài khoản thành công dùng để đăng nhập

3.3.11 Quản lý xem thông tin sinh viên



Hình 13. Use case quản lý xem thông tin sinh viên

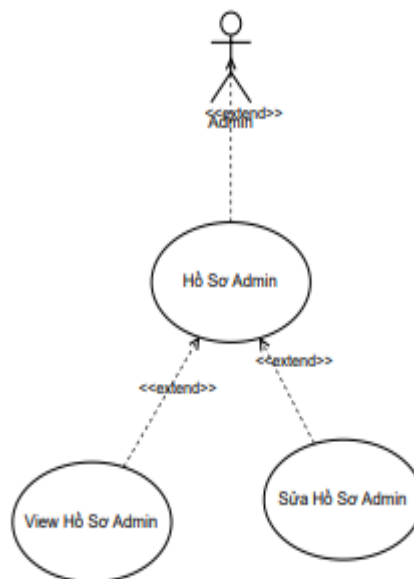
Bảng 11. Bảng đặc tả use case quản lý xem thông tin sinh viên

Test Case	Mô Tả Test Case	Kết Quả Mong Đợi
1	Xem thông tin sinh viên	
	1. Mở ứng dụng.	Hiện thị thông tin sinh viên

Test Case	Mô Tả Test Case	Kết Quả Mong Đợi
	2. Chuyển đến màn hình đăng nhập. 3. Nhập tên người dùng và mật khẩu hợp lệ. 4. Nhấn vào nút đăng nhập. 5. Khi đăng nhập thành công ta chọn hiển thị lập tức các thông tin sinh viên sẽ hiện lên	
2	Xem thông tin ngành học	
	1. Mở ứng dụng. 2. Chuyển đến màn hình đăng nhập. 3. Nhập tên người dùng và mật khẩu hợp lệ. 4. Nhấn vào nút đăng nhập. 5. Khi đăng nhập thành công ta chọn hiển thị lập tức các thông tin các ngành học	Hiện thị thành công các ngành học
3	Xem hồ sơ cần có để du học	
	1. Mở ứng dụng. 2. Chuyển đến màn hình đăng nhập. 3. Nhập tên người dùng và mật khẩu hợp lệ. 4. Nhấn vào nút đăng nhập. 5. Khi đăng nhập thành công ta chọn hiển thị lập tức các thông tin hồ sơ mà sinh viên đã nộp	Hiện thị thành công hồ sơ mà sinh viên đã nộp

Test Case	Mô Tả Test Case	Kết Quả Mong Đợi
4	Hiển thị thông tin gia đình	
	1. Mở ứng dụng. 2. Chuyển đến màn hình đăng nhập. 3. Nhập tên người dùng và mật khẩu hợp lệ. 4. Nhấn vào nút đăng nhập. 5. Khi đăng nhập thành công ta chọn hiển thị lập tức các thông tin gia đình	Hiển thị thành công thông tin gia đình

3.3.12 Quản lý thông tin Admin



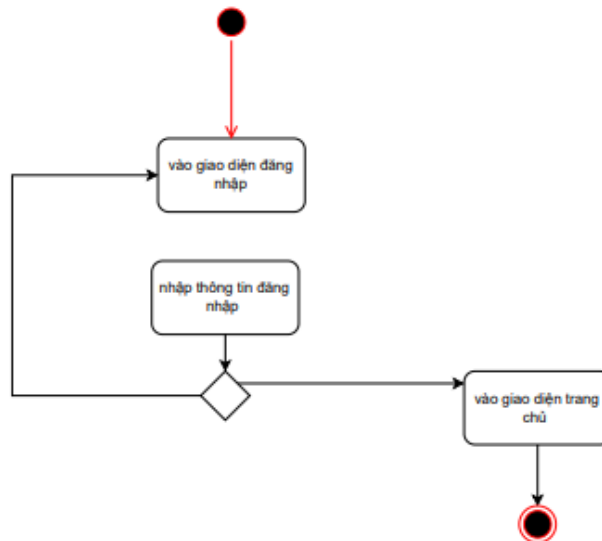
Hình 14. Use case quản lý thông tin admin

Bảng 12. Bảng đặc tả use case quản lý thông tin admin

Test Case	Mô Tả Test Case	Kết Quả Mong Đợi
1	Hiển thị hồ sơ admin	
	<ol style="list-style-type: none"> 1. Mở ứng dụng. 2. Chuyển đến màn hình đăng nhập. 3. Nhập tên người dùng và mật khẩu hợp lệ. 4. Nhấn vào nút đăng nhập. 5. Sau khi đăng nhập chọn nút personal và hiển thị hồ sơ admin 	Hiển thị hồ sơ admin
2	Sửa hồ sơ admin	
	<ol style="list-style-type: none"> 1. Mở ứng dụng. 2. Chuyển đến màn hình đăng nhập. 3. Nhập tên người dùng và mật khẩu hợp lệ. 4. Nhấn vào nút đăng nhập. 5. sau khi đăng nhập chọn nút personal và hiển thị hồ sơ admin sau đó ta sửa lại hồ sơ và bấm lưu lại 	Chỉnh sửa và cập nhập hồ sơ thành công

3.4 Biểu đồ hoạt động

3.4.1 Biểu đồ hoạt động đăng nhập



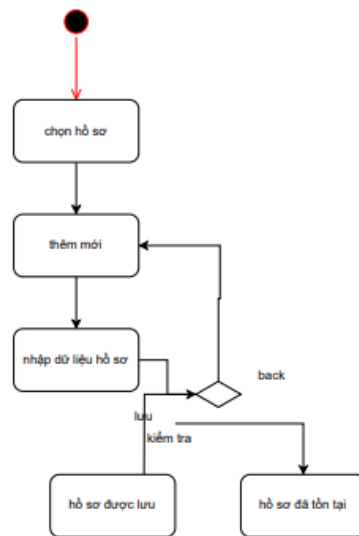
Hình 15. Biểu đồ hoạt động đăng nhập

3.4.2 Biểu đồ hoạt động đăng xuất



Hình 16. Biểu đồ hoạt động đăng xuất

3.4.3 Biểu đồ hoạt động thêm mới hồ sơ



Hình 17. Biểu đồ hoạt động thêm mới hồ sơ

3.4.5 Biểu đồ hoạt động sửa hồ sơ



Hình 18. Biểu đồ hoạt động sửa hồ sơ

3.4.6 Biểu đồ hoạt động đăng kí khóa du học



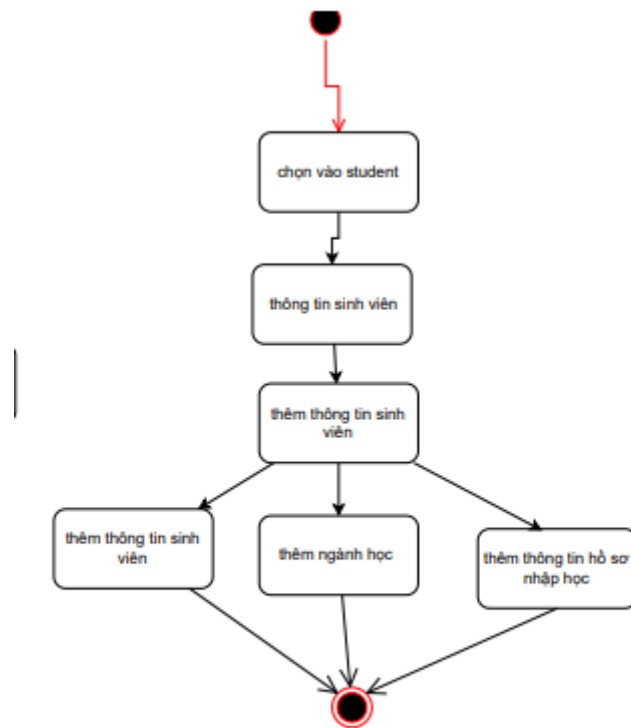
Hình 19. Biểu đồ hoạt động đăng kí khóa du học

3.4.7 Biểu đồ hoạt động gửi đánh giá phản hồi



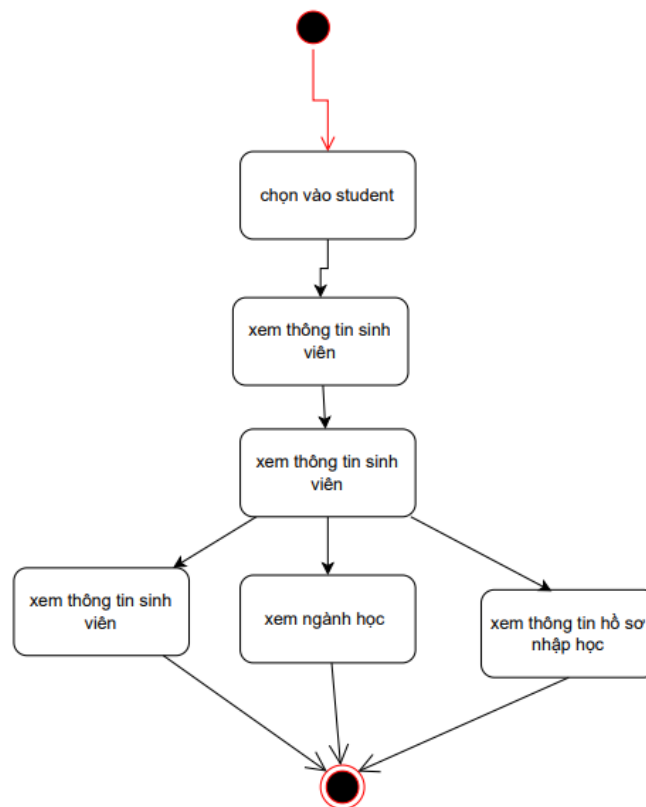
Hình 20. Biểu đồ hoạt động gửi đánh giá phản hồi

3.4.8 Biểu đồ hoạt động thêm sinh viên



Hình 21. Biểu đồ hoạt động thêm sinh viên

3.4.9 Biểu đồ hoạt động xem chi tiết thông tin sinh viên



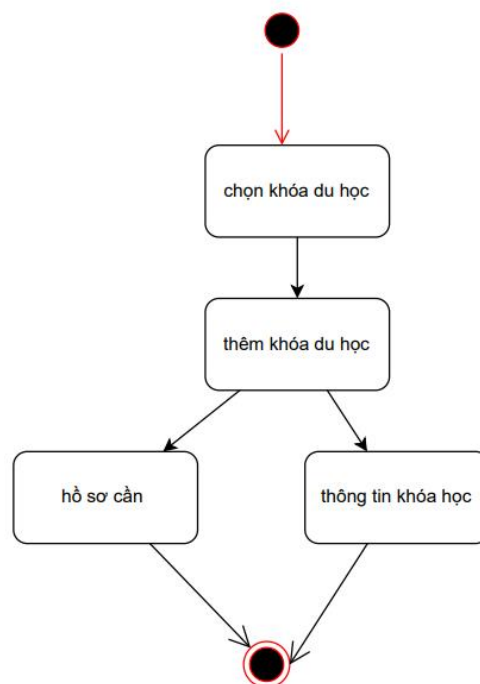
Hình 22. Biểu đồ hoạt động xem chi tiết thông tin sinh viên

3.4.10 Biểu đồ hoạt động sửa thông tin sinh viên



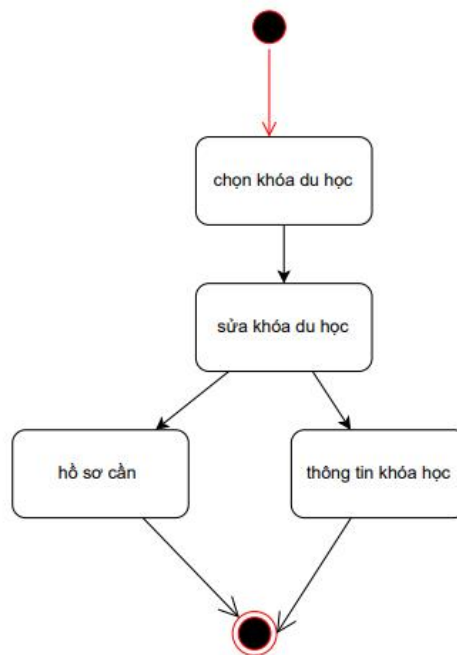
Hình 23. Biểu đồ hoạt động sửa thông tin sinh viên

3.4.11 Biểu đồ hoạt động thêm khóa học



Hình 24. Biểu đồ hoạt động thêm khóa học

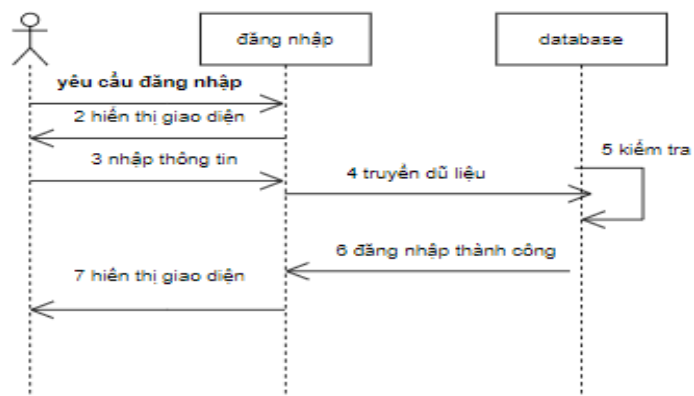
3.4.12 Biểu đồ hoạt động sửa thông tin khóa học



Hình 25. Biểu đồ hoạt động thông tin khóa học

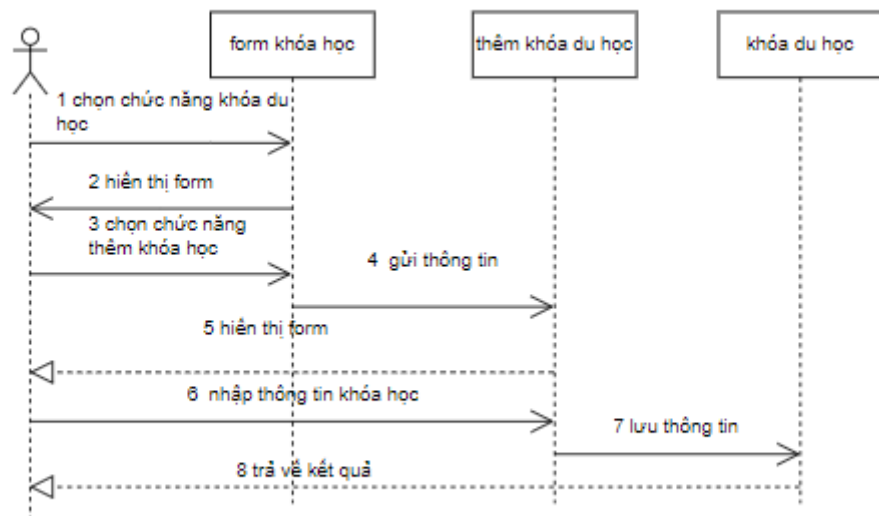
3.5 Biểu đồ tuần tự

3.5.1 Biểu đồ tuần tự đăng nhập



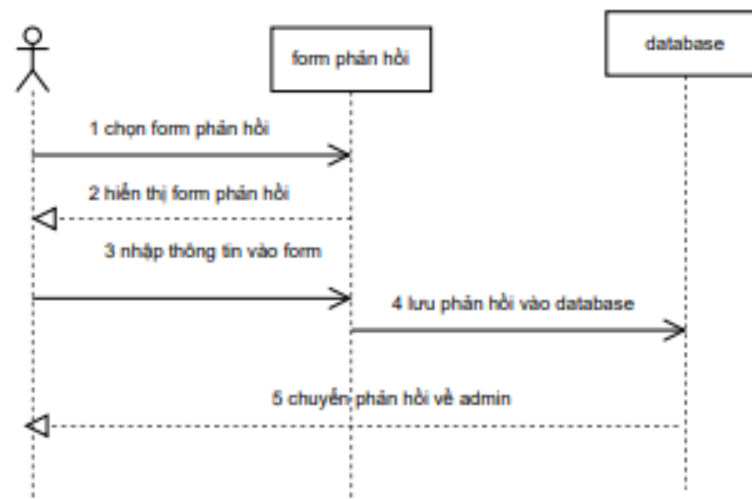
Hình 26. Biểu đồ tuần tự đăng nhập

3.5.2 Biểu đồ tuần tự quản lý khóa du học



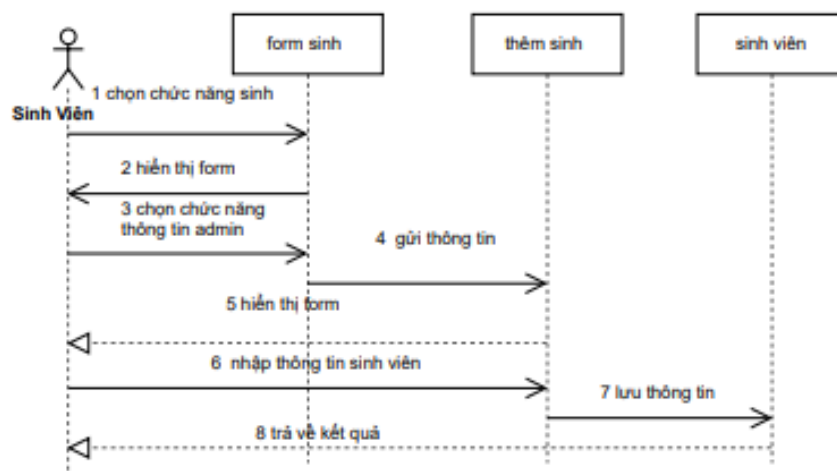
Hình 27. Biểu đồ tuần tự quản lý khóa du học

3.5.3 Biểu đồ tuần tự quản lý phản hồi đánh giá



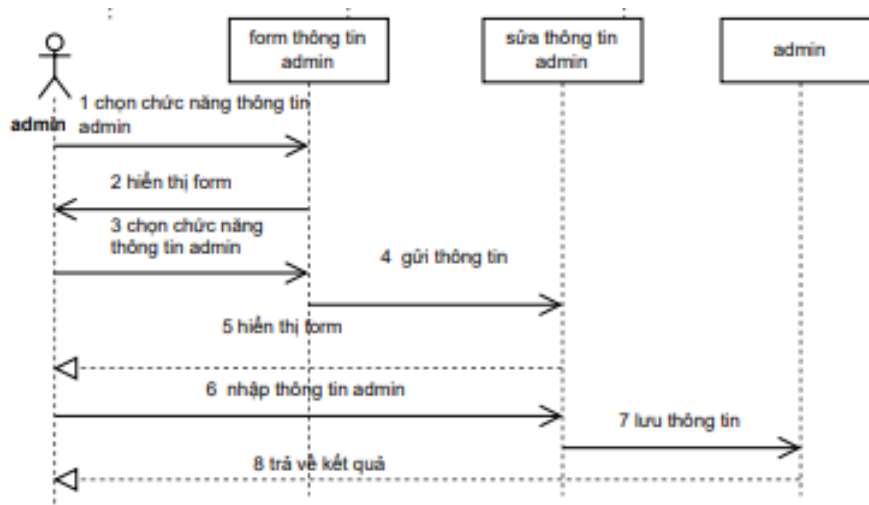
Hình 28. Biểu đồ tuần tự phản hồi đánh giá

3.5.4 Biểu đồ tuần tự quản lý sinh viên



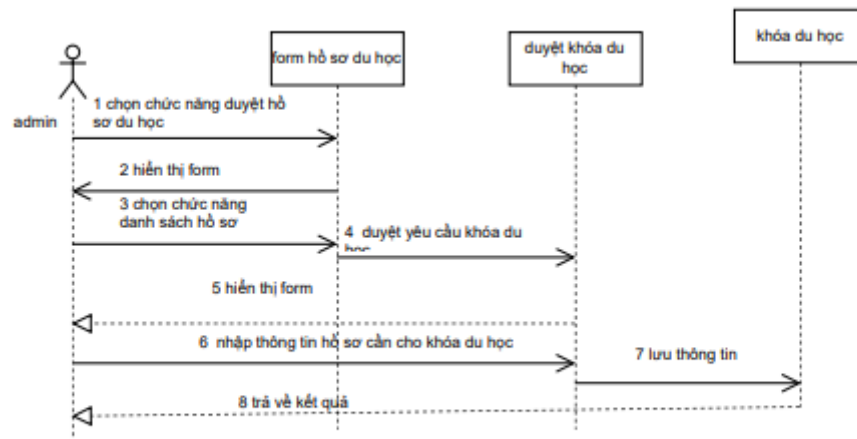
Hình 29. Biểu đồ tuần tự quản lý sinh viên

3.5.5 Biểu đồ tuần tự quản lý thông tin admin



Hình 30. Biểu đồ tuần tự quản lý thông tin sinh viên admin

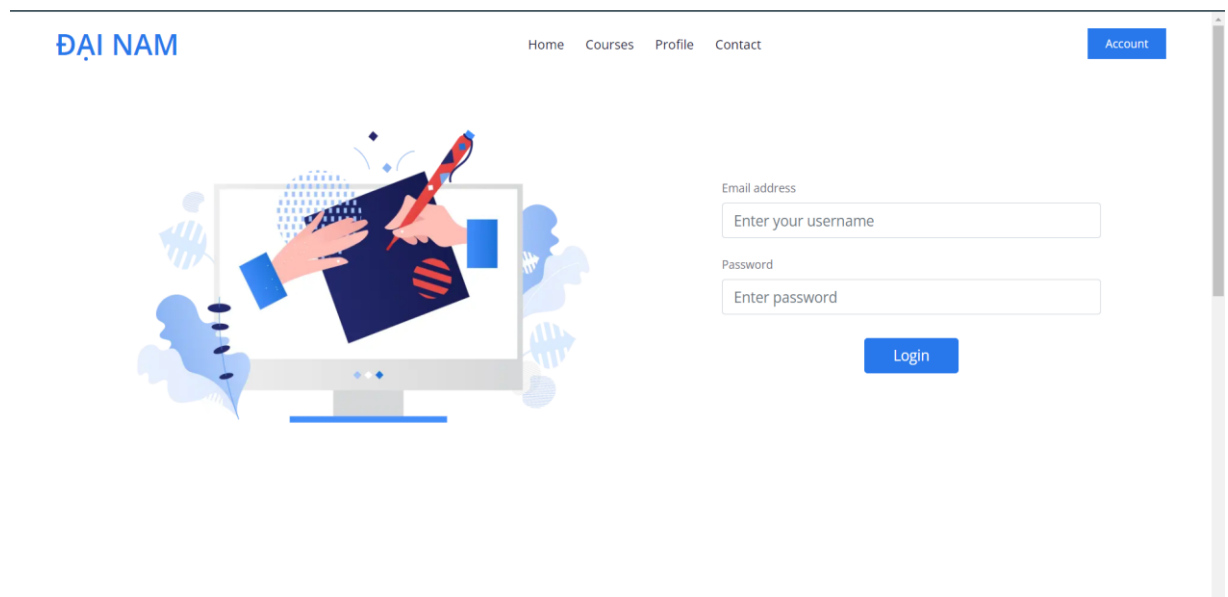
3.5.6 Biểu đồ tuần tự duyệt hồ sơ du học



Hình 31. Biểu đồ tuần tự duyệt hồ sơ du học

3.6 Thiết kế giao diện

3.6.1 Giao diện đăng nhập



Hình 32. Giao diện đăng nhập

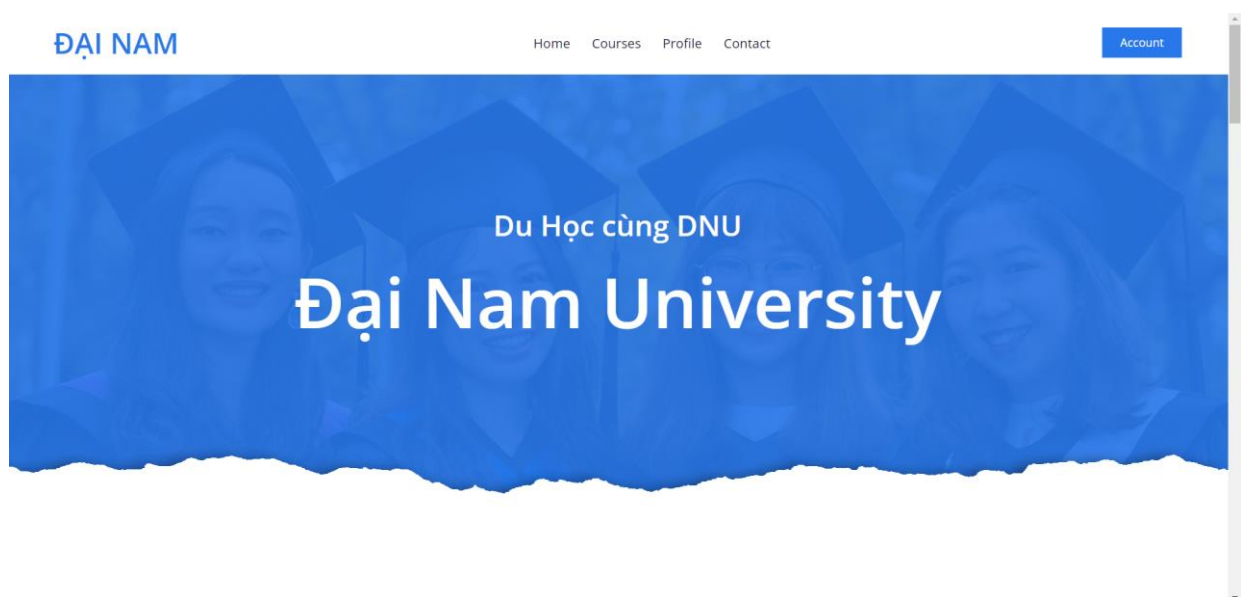
Trang đăng nhập gồm các thành phần sau:

- Thanh điều hướng: Ở phía trên cùng của giao diện có một thanh điều hướng với các tùy chọn Home, Courses, Profile, Contact, và một nút Account.

- Mục đăng nhập: Bên phải của màn hình là khu vực đăng nhập với các trường để nhập Email address và Password, cùng với nút Login bên dưới.
- Logo website: Bên trái của màn hình là logo của website

Giao diện này được thiết kế theo phong cách hiện đại và sạch sẽ, với mục tiêu cung cấp một trải nghiệm người dùng thuận tiện và dễ dàng trong việc đăng nhập và điều hướng. Nếu bạn cần thêm thông tin chi tiết hoặc muốn tối ưu hóa giao diện này cho dự án.

3.6.2 Giao diện trang chủ của Sinh viên



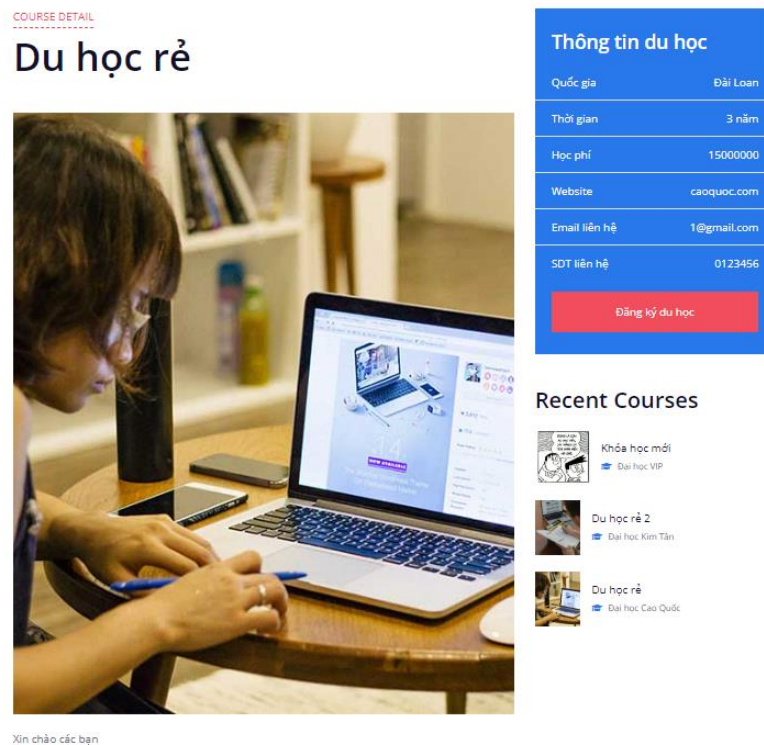
Hình 33. Giao diện trang chủ của sinh viên

Trang chủ gồm các thành phần sau:

- Logo: Ở góc trên bên trái có logo của website
- Thanh điều hướng: Bao gồm các tùy chọn như Home, Courses, Profile, Contact, và Account.
- Nội dung chính: Trung tâm của hình ảnh có dòng chữ lớn màu trắng “Du Học cùng DNU Đại Nam University”, liên quan đến các chương trình hoặc khóa học quốc tế tại Đại Nam University.
- Phong nền: Màu nền là một gradient của các sắc độ màu xanh

Trang chủ này được thiết kế với một phong cách hiện đại và chuyên nghiệp, nhấn mạnh đến việc cung cấp thông tin về cơ hội du học tại Đại Nam University.

3.6.3 Giao diện chi tiết khóa du học



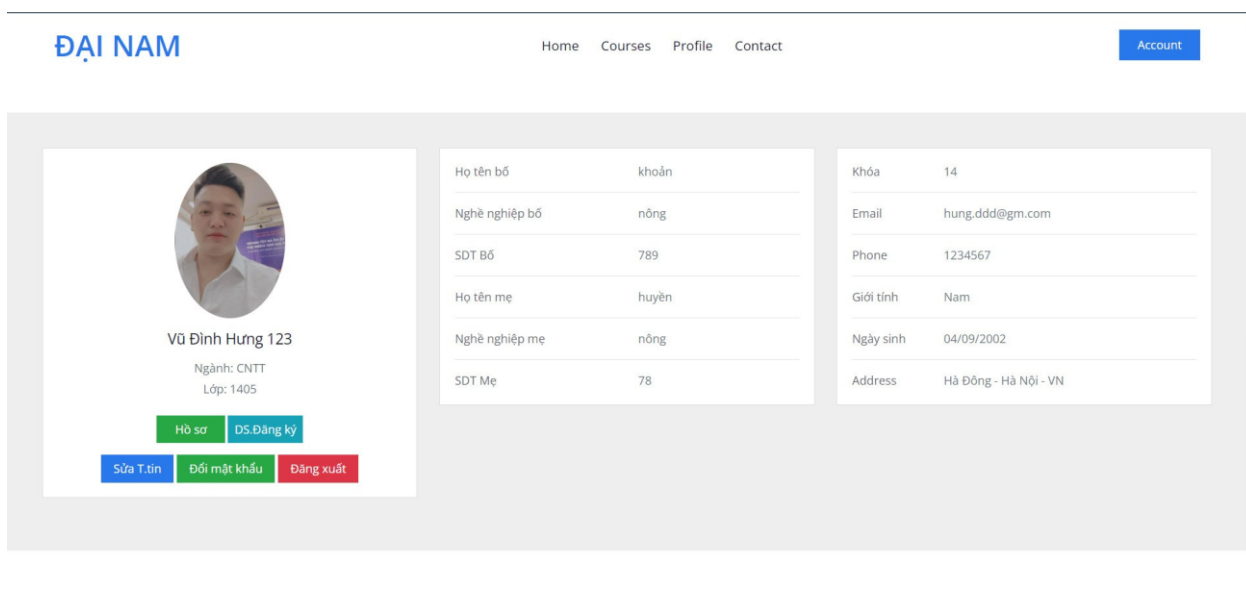
Hình 34. Giao diện chi tiết khóa du học

Dưới đây là mô tả chi tiết trang chi tiết khóa du học:

- Thanh bên: Bên phải có một thanh bên với các mục thông tin được gắn nhãn “Thông tin du học”, “Học phí”, “Website”, và “Số điện thoại” kèm theo các chi tiết tương ứng như học phí, địa chỉ website, và số điện thoại.
- Khóa học gần đây: Dưới thanh bên này có một mục khác với tiêu đề “Recent Courses” liệt kê các khóa học và hình thu nhỏ của chúng.

Hình ảnh này cung cấp cái nhìn tổng quan về cách thông tin về các khóa học du học được trình bày trực tuyến, có thể hữu ích cho những người quan tâm đến các chương trình như vậy.

3.6.4 Giao diện quản lý hồ sơ sinh viên



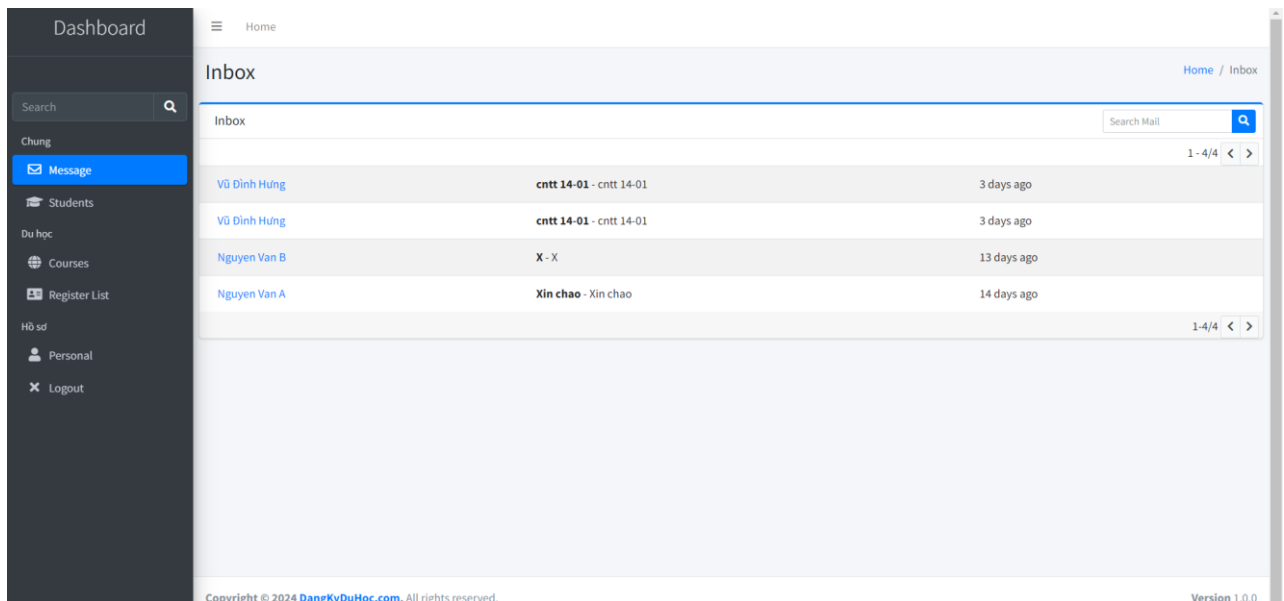
Hình 35. Giao diện quản lý hồ sơ sinh viên

Trang quản lý hồ sơ sinh viên của tôi được thiết kế để giúp sinh viên như bạn có thể dễ dàng theo dõi và cập nhật thông tin cá nhân và học thuật của mình. Khi mở trang này, bạn sẽ thấy:

- Thanh điều hướng: Ở phía trên cùng của trang, tôi có các tùy chọn như Home, Courses, Profile, Contact, và Account để tôi có thể dễ dàng điều hướng qua các phần khác nhau của trang web.
- Thông tin cá nhân: Tôi có một avatar đại diện và bên dưới là tên của mình, cùng với ID duy nhất của mình trong hệ thống.
- Chức năng quản lý: Tôi có các nút chức năng như “Hồ sơ”, “Đổi mật khẩu”, “Sửa TT”, “Bổ sung TT”, và “Đăng xuất” để tôi có thể quản lý thông tin của mình một cách linh hoạt.
- Thông tin chi tiết: Tôi có thể xem các trường thông tin bao gồm ‘Họ tên Đệm’, ‘Khoản’, ‘Rút ra’, ‘Ngày nhập học’, ‘Học phí’, ‘Ngày sinh’, và các chi tiết cá nhân khác như email, số điện thoại, dân tộc, địa chỉ.

Tôi cảm thấy rất hài lòng với cách thông tin của mình được tổ chức một cách rõ ràng và dễ dàng truy cập, giúp tôi quản lý hồ sơ của mình một cách hiệu quả và tiện lợi.

3.6.5 Giao diện trang chủ Admin

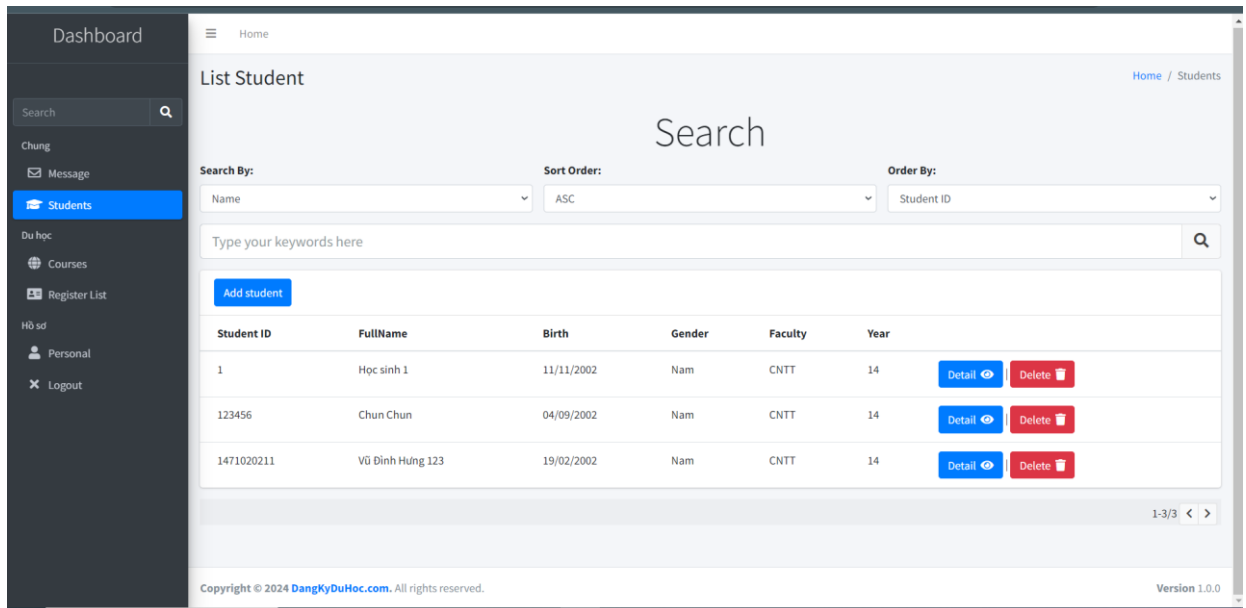


Hình 36. Giao diện trang chủ admin

Khi tôi sử dụng giao diện quản lý admin, tôi cảm thấy rất hài lòng với cách mọi thứ được tổ chức. Dưới đây là mô tả trang admin của tôi:

- Thanh điều hướng bên trái: Tôi có một menu điều hướng ở bên trái màn hình với các tùy chọn như Dashboard, Search, Messages, Students, Courses, Regular user list, và Personal. Điều này giúp tôi dễ dàng chuyển đổi giữa các nhiệm vụ khác nhau mà không mất nhiều thời gian.
- Inbox: Trong phần chính của màn hình, tôi thấy một danh sách các tin nhắn với các tiêu đề như “Web Hosting”, “Nguyễn Văn A”, và “Lý Quốc B”. Tôi cũng có thể soạn tin nhắn mới và sử dụng các chức năng khác thông qua các biểu tượng được cung cấp.
- Thiết kế và màu sắc: Giao diện có thiết kế hiện đại và sạch sẽ với gam màu xanh và trắng, tạo cảm giác chuyên nghiệp và dễ sử dụng.

3.6.6 Giao diện quản lý sinh viên



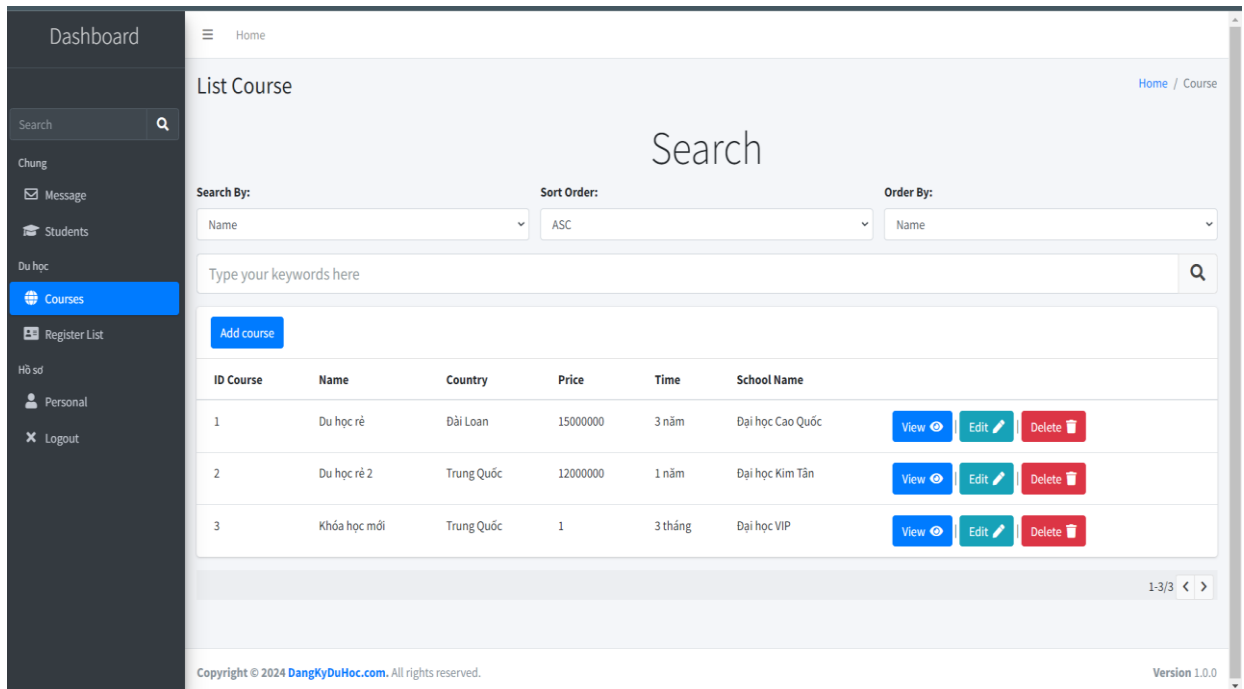
Hình 37. Giao diện quản lý sinh viên

Trang quản lý sinh viên của tôi cung cấp một cái nhìn tổng quan và chi tiết về thông tin sinh viên. Bạn có thể thấy:

- Thanh điều hướng: Bao gồm các tùy chọn như Dashboard, Students, Teachers, và nhiều hơn nữa để dễ dàng chuyển đổi giữa các chức năng.
- Bảng thông tin: Hiển thị danh sách sinh viên với các cột như Student ID, Fullname, Birth, Gender, Faculty và Course, cùng với các nút ‘Detail’ và ‘Delete’ cho mỗi sinh viên.
- Chức năng tìm kiếm và sắp xếp: Cho phép tôi tìm kiếm sinh viên theo từ khóa và sắp xếp thông tin theo các tiêu chí khác nhau.

Giao diện này giúp tôi quản lý và theo dõi thông tin sinh viên một cách hiệu quả và có tổ chức.

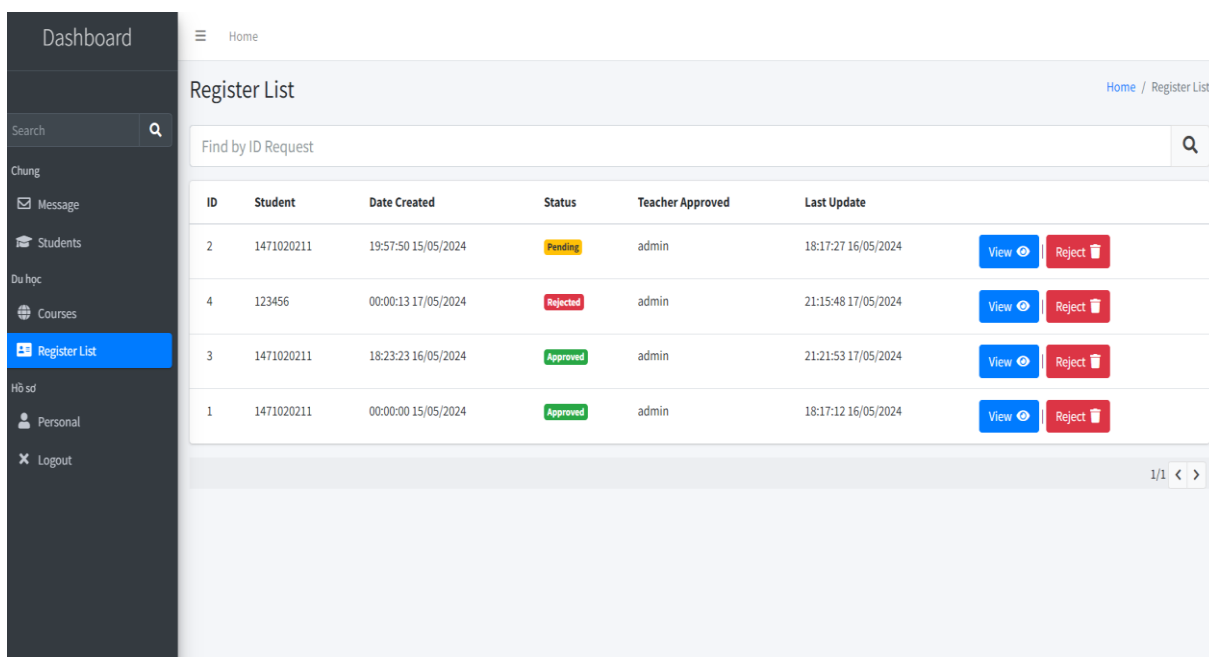
3.6.7 Giao diện quản lý khoá du học



Hình 38. Giao diện quản lý khóa học

- Trang quản lý sinh khóa du học của tôi cung cấp một cái nhìn tổng quan và chi tiết về thông tin sinh viên. Bạn có thể thấy:
- Thanh điều hướng: Bao gồm các tùy chọn như Dashboard, Students và nhiều hơn nữa để dễ dàng chuyển đổi giữa các chức năng.
- Bảng thông tin: Hiển thị danh sách khóa học với các cột như ID Coures, Name, Country..., cùng với các nút 'Detail' và 'Delete' cho mỗi khóa học.
- Chức năng tìm kiếm và sắp xếp: Cho phép tôi tìm kiếm khóa học theo từ khóa và sắp xếp thông tin theo các tiêu chí khác nhau.
- Giao diện này giúp tôi quản lý và theo dõi thông tin khóa học một cách hiệu quả và có tổ chức.

3.6.8 Giao diện duyệt hồ sơ



Hình 39. Giao diện duyệt hồ sơ

Giao diện duyệt hồ sơ của tôi được thiết kế để quản lý và xem xét các hồ sơ đăng ký một cách hiệu quả. Tôi có thể thấy một bảng danh sách với các thông tin như “Student”, “Essay Created”, “Status”, “Teacher Approved”, và “Last Update”. Mỗi bản ghi sinh viên đều có các tùy chọn hành động như ‘View’, ‘Edit’, và ‘More’ để tôi có thể dễ dàng quản lý và cập nhật trạng thái của hồ sơ. Thiết kế giao diện rõ ràng và dễ sử dụng, giúp tôi duyệt hồ sơ một cách nhanh chóng và chính xác.

3.6.9 Giao diện quản lý hồ sơ cá nhân admin

The screenshot displays the 'admin' profile management page. On the left, a dark sidebar contains navigation options: Search, Message, Students, Courses, Register List, Personal (highlighted), and Logout. The main area shows the profile for 'admin'. A profile card on the left lists details: 'Mã giảng viên' (1), 'Họ tên' (Vũ Đình Hùng), 'Số CMND' (0123456), 'Chức vụ' (Trưởng Khoa), and 'Giới tính' (Nam). To the right, a form allows editing personal information with fields for Email (hung@gmail.com123232323), Số điện thoại (01234561), Địa chỉ (HD - HN1), Học vị (Thạc sĩ 1), Ngày sinh (01/01/0001), and Mật khẩu mới (with a hint to enter a new password). A 'Choose File' button and 'No file chosen' text are visible above the profile card.

Hình 40. Giao diện quản lý hồ sơ cá nhân admin

Giao diện quản lý hồ sơ cá nhân admin của tôi được thiết kế để giúp người quản trị dễ dàng nhập và chỉnh sửa thông tin cá nhân. Tôi có thể thấy các trường thông tin như ‘Full name’, ‘Email’, ‘Phone number’, ‘Address’, và các nút lệnh như ‘Save’ và ‘Cancel’ để áp dụng hoặc hủy bỏ các thay đổi. Thiết kế giao diện này nhằm mục đích tạo sự thuận tiện và hiệu quả trong việc quản lý thông tin cá nhân trong hệ thống.

3.6.10 Giao diện quản lý hồ sơ thông tin sinh viên

Profile

Hồ sơ Sửa thông tin

Bằng cấp 3

Ghi chú: đây là bằng cấp 3
Loại: **Bản gốc**

Ảnh nền trắng

Ghi chú: ảnh nền
Loại: **Bản gốc**

Family

Họ tên bố

Nghề nghiệp bố

SĐT bố

Họ tên mẹ

Nghề nghiệp mẹ

SĐT mẹ

Ngành học

Mã ngành CNTT

Mã lớp 1

Mã hệ đào tạo DH

Mã sinh viên 123321

Email hung@gmail.com

CCCD 1234

Birth 19/02/2002

Gender Nam

Phone 038858732

Hình 41. Giao diện quản lý hồ sơ thông tin sinh viên

3.6.11 Giao diện duyệt hồ sơ sinh viên

Course Information

Country Đài Loan

Time 3 năm

Price 15000000

Website caoquoc.com

Email Contact 1@gmail.com

Phone Contact 0123456

Student Information

ID Student 1471020211

Fullname Vũ Đình Hưn 123

Email hung.ddd@gm.com

Phone 123456

Address Hà Đông - Hà Nội - VN

View

Status

Approved

Note

Thiếu bài TOCFLA2

Save

Information

Bằng cấp 3

Description

Note: Đã xin ở trường

Hình 42. Giao diện duyệt hồ sơ sinh viên

CHƯƠNG 4: KẾT QUẢ VÀ KIỂM THỬ

4.1 Kiểm thử đăng nhập

TC	Các bước thực hiện	Kết quả dự kiến	Kết quả đạt được
TC1	Tài khoản trống Mật khẩu trống Bấm đăng nhập	Show message box : Tài khoản hoặc mật khẩu không đúng!	Passed
TC2	Nhập tài khoản Không nhập mật khẩu Bấm đăng nhập	Show message box : Tài khoản hoặc mật khẩu không đúng!	Passed
TC3	Bỏ trống tài khoản Nhập mật khẩu Bấm đăng nhập	Show message box : Tài khoản hoặc mật khẩu không đúng!	Passed
TC4	Nhập sai định dạng tài khoản Nhập mật khẩu đúng	Show message box : Tài khoản hoặc mật khẩu không đúng!	Passed
TC5	Nhập tài khoản đúng Nhập mật khẩu đúng Bấm đăng nhập	Show message box: Xin chào [displayname] and show main form	Passed

4.2 Kiểm thử chức năng khác

TC	Tên	Các bước thực hiện	Kết quả dự kiến	Kết quả đạt được
TC1	Xem hồ sơ du học	Tại giao diện Main, click hồ sơ	Show toàn bộ thông tin hồ sơ đã tải lên	Passed

TC	Tên	Các bước thực hiện	Kết quả dự kiến	Kết quả đạt được
TC2	Xem danh sách Khóa du học	1. Tại giao diện main 2. Click vào COURSES	Show toàn bộ danh sách khóa du học	Passed
TC3	Đăng kí khóa du học	1. Trong giao diện danh mục khóa học, chọn khóa học mong muốn	Show danh sách khóa du học đã đăng kí và trạng thái duyệt	Passed
TC4	Xem thông tin sinh viên	1. Đăng nhập vào giao diện 2. Click vào profile	Show thông tin sinh viên và thông tin gia đình	Passed
TC5	Sửa thông tin sinh viên	1. Click button 'Sửa thông tin sinh viên' 2. Điền thông tin muốn sửa và lưu	Sửa thông tin thành công và lưu vào cơ sở dữ liệu	Passed
TC6	Gửi phản hồi đánh giá	1. Click inbox 2. Gửi phản hồi đánh giá và gửi	Hiển thị cho admin xem	Passed
TC7	Quản lý student	1. Click menustrip 'Quản trị' 2. Chọn 'Student'	Show danh sách sinh viên và chức năng thêm, xem chi tiết, xóa	Passed
TC8	Thêm sinh viên	1. Click menustrip 'Quản trị' 2. Chọn 'add student'	Thêm sinh viên vào cơ sở dữ liệu và hiển thị trong danh sách	Passed

TC	Tên	Các bước thực hiện	Kết quả dự kiến	Kết quả đạt được
		3. Điền thông tin sinh viên		
TC9	Sửa thông tin sinh viên	1. Click menustrip 'Quản trị' 2. Chọn 'Detail' và chọn 'Sửa' 3. Điền thông tin cần sửa và lưu	Lưu thông tin sinh viên vào cơ sở dữ liệu	Passed
TC10	Xóa thông tin sinh viên	1. Click menustrip 'Tài khoản' 2. Chọn sinh viên cần xóa	Hiển thị thông báo xác nhận xóa sinh viên khỏi cơ sở dữ liệu	Passed
TC11	Quản lý khóa du học	1. Click menustrip 'Quản trị' 2. Chọn 'Course'	Show danh sách khóa du học và chức năng thêm, xem chi tiết, xóa	Passed
TC12	Thêm khóa du học	1. Click menustrip 'Quản trị', 2. Chọn 'add Course', 3. Điền thông tin khóa du học	Thêm khóa du học vào cơ sở dữ liệu và hiển thị trong danh sách	Passed
TC13	Sửa thông tin khóa du học	1. Click menustrip 'Quản trị' 2. Chọn 'edit' và chọn 'Sửa'	Lưu thông tin khóa du học vào cơ sở dữ liệu	Passed

TC	Tên	Các bước thực hiện	Kết quả dự kiến	Kết quả đạt được
		3. Điền thông tin cần sửa và lưu		
TC14	Xóa khóa du học	1. Click menustrip 'Tài khoản' 2. Chọn khóa du học cần xóa	Hiện thị thông báo xác nhận xóa khóa du học khỏi cơ sở dữ liệu	Passed
TC15	Duyệt hồ sơ đăng ký du học	1. Click menustrip 'Tài khoản' 2. Chọn 'Register List' 3. Chọn 'view' và duyệt theo yêu cầu	Duyệt hồ sơ đăng ký du học thành công	Passed

KẾT LUẬN

Sau quá trình nghiên cứu, phát triển và triển khai, ứng dụng website dành cho việc đăng ký và quản lý hồ sơ sinh viên du học nước ngoài của Viện Đào Tạo Quốc Tế - Đại Học Đại Nam đã hoàn thành với nhiều kết quả đáng khích lệ. Ứng dụng không chỉ đáp ứng đầy đủ các yêu cầu ban đầu mà còn mang lại nhiều tiện ích vượt trội, góp phần cải thiện quy trình quản lý và nâng cao chất lượng dịch vụ đào tạo.

Việc ứng dụng công nghệ thông tin vào công tác quản lý đã giúp tiết kiệm thời gian, giảm thiểu sai sót và tạo điều kiện thuận lợi cho sinh viên trong quá trình đăng ký và quản lý hồ sơ. Đồng thời, hệ thống cũng giúp Viện Đào Tạo Quốc Tế theo dõi và quản lý thông tin sinh viên một cách khoa học, minh bạch và hiệu quả hơn. Điều này không chỉ nâng cao trải nghiệm của người dùng mà còn góp phần xây dựng hình ảnh chuyên nghiệp, hiện đại của Viện Đào Tạo Quốc Tế - Đại Học Đại Nam.

Qua quá trình thực hiện dự án, tôi đã rút ra nhiều bài học quý báu về quản lý dự án công nghệ thông tin, đặc biệt là trong lĩnh vực giáo dục. Những kinh nghiệm này sẽ là nền tảng quan trọng cho các dự án tương lai, không chỉ trong phạm vi Viện mà còn có thể mở rộng ra các lĩnh vực khác.

Tôi hy vọng rằng, với sự hoàn thiện không ngừng, ứng dụng này sẽ tiếp tục được nâng cấp và phát triển, đáp ứng tốt hơn nữa nhu cầu của sinh viên và nhà trường. Đồng thời, mong rằng các kinh nghiệm và kiến thức thu được từ dự án sẽ góp phần thúc đẩy sự phát triển chung của công nghệ thông tin trong giáo dục, mang lại những giá trị thiết thực cho xã hội.

TÀI LIỆU THAM KHẢO

Tiếng Anh:

- [1] Troelsen, Andrew, & Japikse, Philip. (2017). "Pro C# 7: With .NET and .NET Core". Apress.
- [2] Sharp, John. (2018). "Microsoft Visual C# Step by Step". Microsoft Press.
- [3] Microsoft Documentation. (2022). "SQL Server Documentation". Microsoft.
- [4] Udemy. (2022). "C# Basics for Beginners: Learn C# Fundamentals by Coding". Udemy.

Tiếng Việt:

- [1] Trần Văn Thắng. (2021). "Quản lý cơ sở dữ liệu với SQL Server". Nhà xuất bản Lao động - Xã hội.
- [2] Nguyễn Quang Huy. (2021). "Phân tích và thiết kế hệ thống thông tin". Nhà xuất bản Giáo dục Việt Nam.